Untitled0

December 4, 2022

[1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

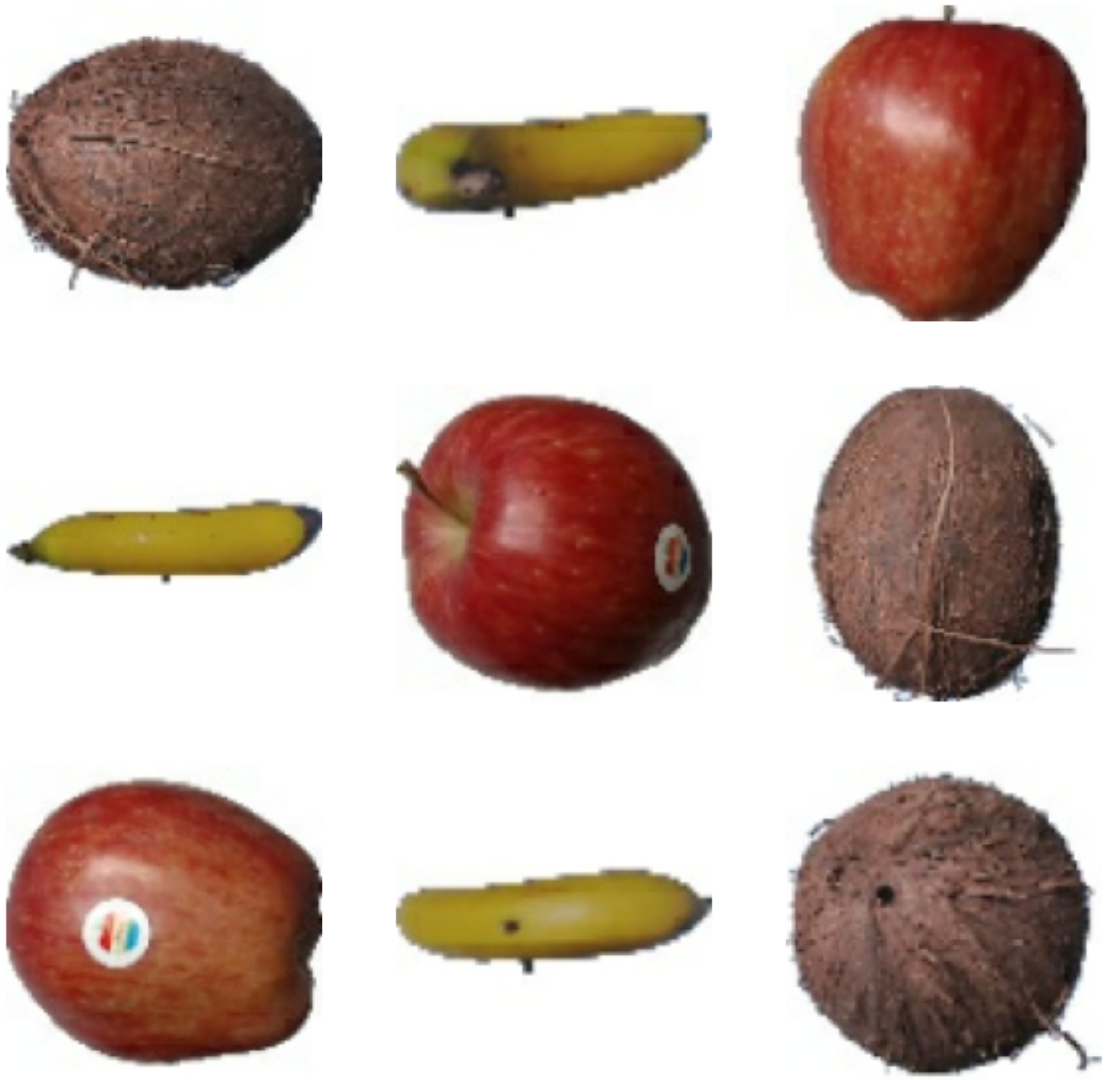
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[1]:
```
import tensorflow as tf
import numpy as np

batch_size = 32
num_classes = 3
epochs = 20

train_ds = tf.keras.utils.image_dataset_from_directory(
    directory='/content/drive/MyDrive/ML/DataSetFruits/',
    validation_split=0.2,
    subset="training",
    labels='inferred',
    label_mode='categorical',
    seed=123,
    shuffle=True,
    batch_size=batch_size,
    image_size=(100,100))

val_ds = tf.keras.utils.image_dataset_from_directory(
    directory='/content/drive/MyDrive/ML/DataSetFruits/',
    validation_split=0.2,
    subset="validation",
    labels='inferred',
    label_mode='categorical',
    seed=123,
    shuffle=True,
    batch_size=batch_size,
    image_size=(100,100))
```

```
Found 1472 files belonging to 3 classes.
Using 1178 files for training.
Found 1472 files belonging to 3 classes.
```

```
Using 294 files for validation.
```

[3]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
  for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(images[i].numpy().astype("uint8"))
    plt.axis("off")
```



The goal is to differentiate between apples, bananas, and cocos

1 Basic Sequential Model

[190]:
```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(100, 100, 3)),
  tf.keras.layers.Dense(512, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(512, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(num_classes, activation='softmax'),
])
```

Let's check the summary

[191]:
```
model.summary()
```

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_7 (Flatten)         (None, 30000)             0

 dense_21 (Dense)            (None, 512)               15360512

 dropout_20 (Dropout)        (None, 512)               0

 dense_22 (Dense)            (None, 512)               262656

 dropout_21 (Dropout)        (None, 512)               0

 dense_23 (Dense)            (None, 3)                 1539

=================================================================
Total params: 15,624,707
Trainable params: 15,624,707
Non-trainable params: 0
_________________________________________________________________
```

Now we compile and then fit to our dataset

[189]:
```
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['accuracy'])

history = model.fit(train_ds,
                    batch_size=batch_size,
                    epochs=epochs,
                    verbose=1,
                    validation_data=val_ds)
```

```
Epoch 1/20
30/37 [=======================>…] - ETA: 1s - loss: 4634.1978 - accuracy:
0.6240


      ␣
↪---------------------------------------------------------------------------

       KeyboardInterrupt                         Traceback (most recent call␣
↪last)

       <ipython-input-189-ccdc65fc173d> in <module>
         3               metrics=['accuracy'])
         4
   ----> 5 history = model.fit(train_ds,
         6                     batch_size=batch_size,
         7                     epochs=epochs,


       /usr/local/lib/python3.8/dist-packages/keras/utils/traceback_utils.py in␣
↪error_handler(*args, **kwargs)
        62     filtered_tb = None
        63     try:
   ---> 64       return fn(*args, **kwargs)
        65     except Exception as e:  # pylint: disable=broad-except
        66       filtered_tb = _process_traceback_frames(e.__traceback__)


       /usr/local/lib/python3.8/dist-packages/keras/engine/training.py in␣
↪fit(self, x, y, batch_size, epochs, verbose, callbacks, validation_split,␣
↪validation_data, shuffle, class_weight, sample_weight, initial_epoch,␣
↪steps_per_epoch, validation_steps, validation_batch_size, validation_freq,␣
↪max_queue_size, workers, use_multiprocessing)
      1407                 _r=1):
      1408               callbacks.on_train_batch_begin(step)
   -> 1409               tmp_logs = self.train_function(iterator)
      1410               if data_handler.should_sync:
      1411                 context.async_wait()


       /usr/local/lib/python3.8/dist-packages/tensorflow/python/util/
↪traceback_utils.py in error_handler(*args, **kwargs)
       148     filtered_tb = None
       149     try:
   --> 150       return fn(*args, **kwargs)
       151     except Exception as e:
       152       filtered_tb = _process_traceback_frames(e.__traceback__)
```

```
       /usr/local/lib/python3.8/dist-packages/tensorflow/python/eager/
↪def_function.py in __call__(self, *args, **kwds)
       913
       914       with OptionalXlaContext(self._jit_compile):
   --> 915         result = self._call(*args, **kwds)
       916
       917       new_tracing_count = self.experimental_get_tracing_count()


       /usr/local/lib/python3.8/dist-packages/tensorflow/python/eager/
↪def_function.py in _call(self, *args, **kwds)
       945       # In this case we have created variables on the first call, so␣
↪we run the
       946       # defunned version which is guaranteed to never create␣
↪variables.
   --> 947       return self._stateless_fn(*args, **kwds)  # pylint:␣
↪disable=not-callable
       948     elif self._stateful_fn is not None:
       949       # Release the lock early so that multiple threads can perform␣
↪the call


       /usr/local/lib/python3.8/dist-packages/tensorflow/python/eager/function.
↪py in __call__(self, *args, **kwargs)
      2451       (graph_function,
      2452        filtered_flat_args) = self._maybe_define_function(args,␣
↪kwargs)
   -> 2453     return graph_function._call_flat(
      2454         filtered_flat_args, captured_inputs=graph_function.
↪captured_inputs)  # pylint: disable=protected-access
      2455


       /usr/local/lib/python3.8/dist-packages/tensorflow/python/eager/function.
↪py in _call_flat(self, args, captured_inputs, cancellation_manager)
      1858         and executing_eagerly):
      1859       # No tape is watching; skip to running the function.
   -> 1860       return self._build_call_outputs(self._inference_function.call(
      1861           ctx, args, cancellation_manager=cancellation_manager))
      1862     forward_backward = self._select_forward_and_backward_functions(


       /usr/local/lib/python3.8/dist-packages/tensorflow/python/eager/function.
↪py in call(self, ctx, args, cancellation_manager)
       495       with _InterpolateFunctionError(self):
       496         if cancellation_manager is None:
```

```
--> 497           outputs = execute.execute(
    498               str(self.signature.name),
    499               num_outputs=self._num_outputs,


/usr/local/lib/python3.8/dist-packages/tensorflow/python/eager/execute.
↪py in quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
     52   try:
     53     ctx.ensure_initialized()
---> 54     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name,␣
↪op_name,
     55                                         inputs, attrs, num_outputs)
     56   except core._NotOkStatusException as e:


KeyboardInterrupt:
```

Let's go ahead and graph it

[ ]:
```
history.history.keys()
```

[ ]:
```
import matplotlib.pyplot as plt

# Plot training & validation accuracy values
plt.plot(history.history['val_accuracy'])
plt.plot(history.history['accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

[ ]:
```
score = model.evaluate(val_ds, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

2 CNN

[48]:
```
num_filters = 8
filter_size = 3
pool_size = 2

model = tf.keras.models.Sequential(
    [
        tf.keras.Input(shape=(100, 100, 3)),
```

```
        tf.keras.layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
        tf.keras.layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        tf.keras.layers.MaxPooling2D(pool_size=(2, 2)),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(num_classes, activation="softmax"),
    ]
)
```

[49]:
```
model.summary()
```

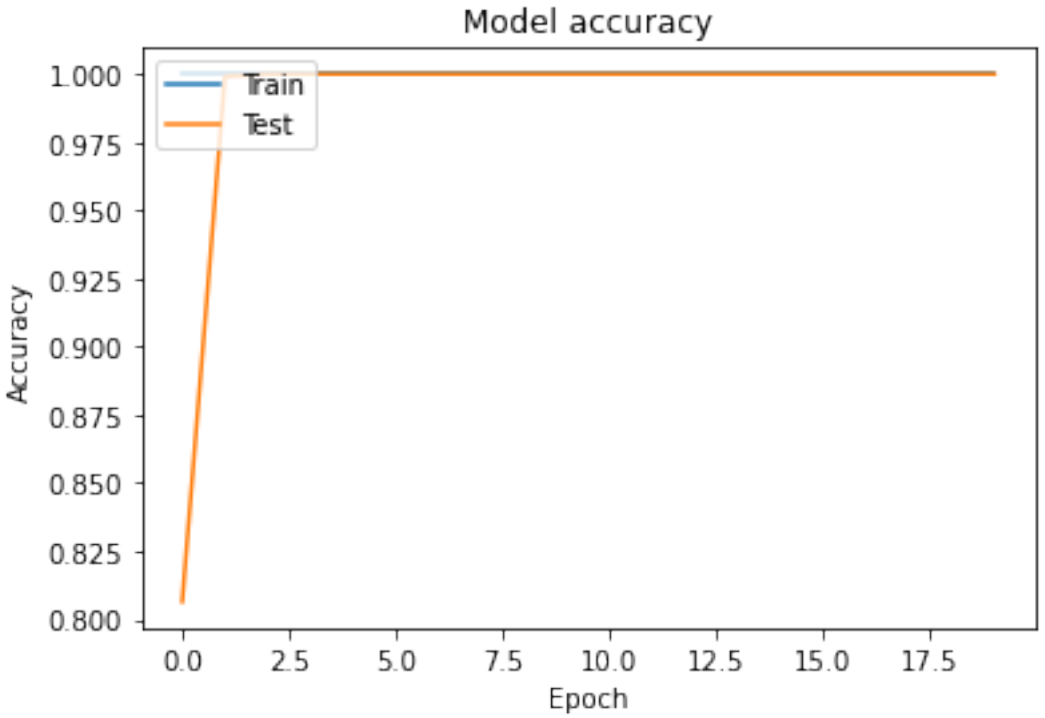
```
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 98, 98, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 49, 49, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 47, 47, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 23, 23, 64)       0
 2D)

 flatten_5 (Flatten)         (None, 33856)             0

 dropout_9 (Dropout)         (None, 33856)             0

 dense_13 (Dense)            (None, 3)                 101571

=================================================================
Total params: 120,963
Trainable params: 120,963
Non-trainable params: 0
_________________________________________________________________
```

[50]:
```
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

history = model.fit(train_ds,
                    batch_size=batch_size,
                    epochs=epochs,
                    verbose=1,
                    validation_data=val_ds)
```

```
Epoch 1/20
37/37 [==============================] - 21s 552ms/step - loss: 42.4354 -
accuracy: 0.8065 - val_loss: 1.1198e-05 - val_accuracy: 1.0000
Epoch 2/20
37/37 [==============================] - 18s 488ms/step - loss: 6.4997e-04 -
accuracy: 0.9992 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 3/20
37/37 [==============================] - 18s 487ms/step - loss: 2.0239e-10 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 4/20
37/37 [==============================] - 18s 488ms/step - loss: 9.1077e-10 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 5/20
37/37 [==============================] - 18s 490ms/step - loss: 1.0120e-10 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 6/20
37/37 [==============================] - 18s 489ms/step - loss: 2.0239e-10 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 7/20
37/37 [==============================] - 18s 486ms/step - loss: 2.8335e-09 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 8/20
37/37 [==============================] - 18s 487ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 9/20
37/37 [==============================] - 18s 490ms/step - loss: 3.0359e-10 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 10/20
37/37 [==============================] - 19s 506ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 11/20
37/37 [==============================] - 18s 484ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 12/20
37/37 [==============================] - 18s 486ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 13/20
37/37 [==============================] - 18s 485ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 14/20
37/37 [==============================] - 18s 484ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 15/20
37/37 [==============================] - 18s 488ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 16/20
37/37 [==============================] - 18s 485ms/step - loss: 3.0359e-10 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
```

```
Epoch 17/20
37/37 [==============================] - 18s 482ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 18/20
37/37 [==============================] - 18s 486ms/step - loss: 0.0000e+00 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 19/20
37/37 [==============================] - 18s 485ms/step - loss: 4.0479e-10 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 20/20
37/37 [==============================] - 20s 544ms/step - loss: 1.1941e-08 -
accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
```

[51]:
```
# Plot training & validation accuracy values
plt.plot(history.history['val_accuracy'])
plt.plot(history.history['accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```



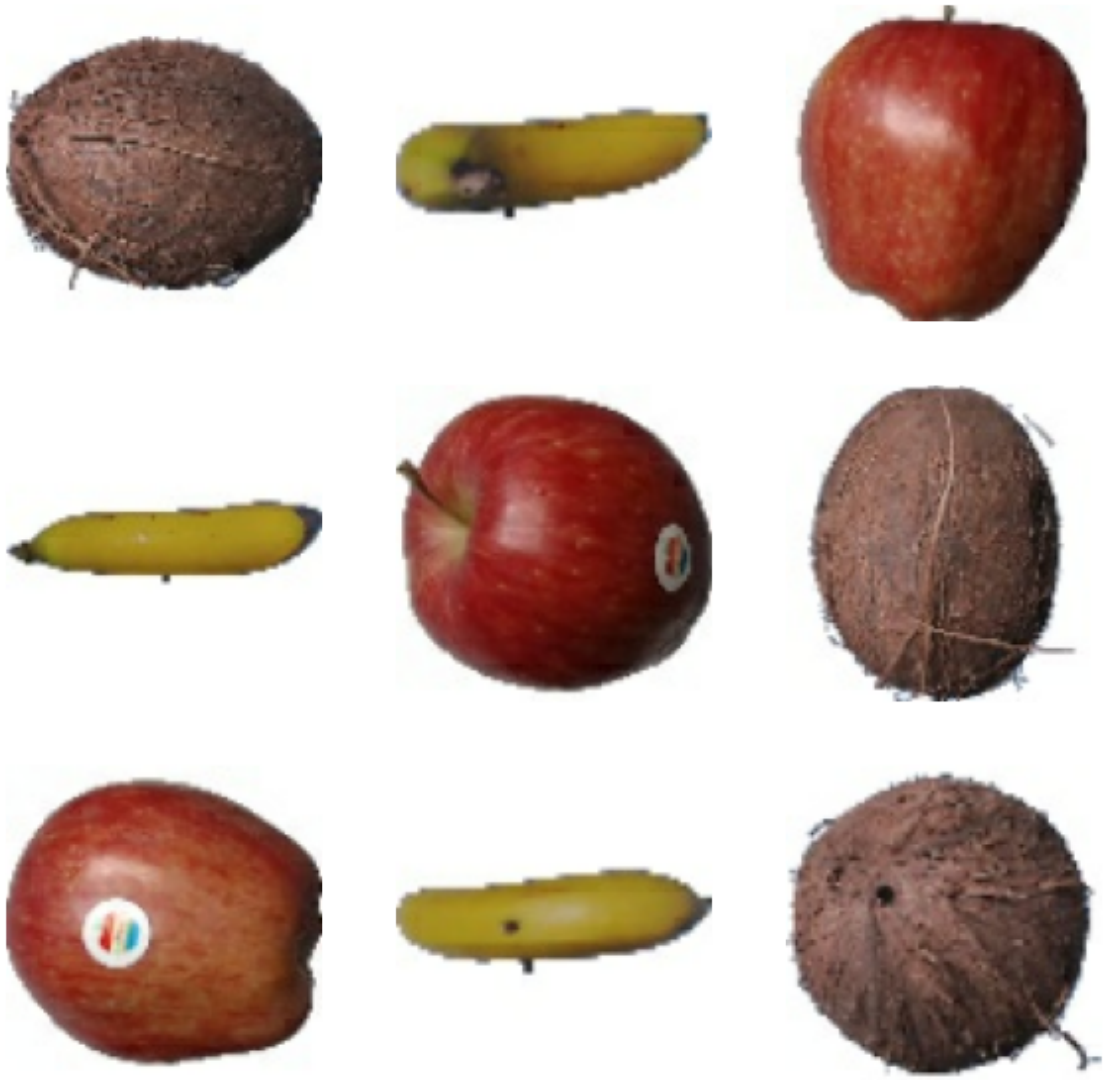
[52]:
```
score = model.evaluate(val_ds, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.0
Test accuracy: 1.0
```

3 Pretrained Model and Transfer Learning

[4]:
```
train_ds = tf.keras.utils.image_dataset_from_directory(
    directory='/content/drive/MyDrive/ML/DataSetFruits/',
    validation_split=0.2,
    subset="training",
    labels='inferred',
    label_mode='categorical',
    seed=123,
    shuffle=True,
    batch_size=batch_size,
    image_size=(100,100))

val_ds = tf.keras.utils.image_dataset_from_directory(
    directory='/content/drive/MyDrive/ML/DataSetFruits/',
    validation_split=0.2,
    subset="validation",
    labels='inferred',
    label_mode='categorical',
    seed=123,
    shuffle=True,
    batch_size=batch_size,
    image_size=(100,100))

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
  for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(images[i].numpy().astype("uint8"))
    plt.axis("off")
```

```
Found 1472 files belonging to 3 classes.
Using 1178 files for training.
Found 1472 files belonging to 3 classes.
Using 294 files for validation.
```



[5]:
```
val_batches = tf.data.experimental.cardinality(val_ds)
test_ds = val_ds.take(val_batches // 5)
val_ds = val_ds.skip(val_batches // 5)
```

[6]:
```
print('Number of validation batches: %d' % tf.data.experimental.
 ↪cardinality(val_ds))
print('Number of test batches: %d' % tf.data.experimental.cardinality(test_ds))
```

```
Number of validation batches: 8
Number of test batches: 2
```

[7]:
```
AUTOTUNE = tf.data.AUTOTUNE
```

```
train_dataset = train_ds.prefetch(buffer_size=AUTOTUNE)
validation_dataset = val_ds.prefetch(buffer_size=AUTOTUNE)
test_dataset = test_ds.prefetch(buffer_size=AUTOTUNE)
```

[8]:
```
data_augmentation = tf.keras.Sequential([
  tf.keras.layers.RandomFlip('horizontal'),
  tf.keras.layers.RandomRotation(0.2),
])
```

[9]:
```
for image, _ in train_dataset.take(1):
  plt.figure(figsize=(10, 10))
  first_image = image[0]
  for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    augmented_image = data_augmentation(tf.expand_dims(first_image, 0))
    plt.imshow(augmented_image[0] / 255)
    plt.axis('off')
```

[10]:
```
tf.keras.applications.mobilenet_v2.MobileNetV2(
    input_shape=None,
    alpha=1.0,
    include_top=True,
    weights='imagenet',
    input_tensor=None,
    pooling=None,
    classes=1000,
    classifier_activation='softmax'
)

preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input
```

[11]:
```
rescale = tf.keras.layers.Rescaling(1./127.5, offset=-1)
```

[12]:
```
IMG_SIZE = (100, 100)
```

[13]:
```
IMG_SHAPE = IMG_SIZE + (3,)
base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SHAPE,
                                               include_top=False,
                                               weights='imagenet')
```

```
WARNING:tensorflow:`input_shape` is undefined or non-square, or `rows` is not in
[96, 128, 160, 192, 224]. Weights for input shape (224, 224) will be loaded as
the default.
```

[14]:
```
image_batch, label_batch = next(iter(train_ds))
feature_batch = base_model(image_batch)
print(feature_batch.shape)
```

```
(32, 4, 4, 1280)
```

[15]:
```
base_model.trainable = False
```

[16]:
```
base_model.summary()
```

```
Model: "mobilenetv2_1.00_224"
__________________________________________________________________________________
__________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================
==================
 input_2 (InputLayer)           [(None, 100, 100, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 50, 50, 32)   864
['input_2[0][0]']

 bn_Conv1 (BatchNormalization)  (None, 50, 50, 32)   128         ['Conv1[0][0]']

 Conv1_relu (ReLU)              (None, 50, 50, 32)   0
['bn_Conv1[0][0]']

 expanded_conv_depthwise (Depth  (None, 50, 50, 32)  288
['Conv1_relu[0][0]']
 wiseConv2D)

 expanded_conv_depthwise_BN (Ba  (None, 50, 50, 32)  128
['expanded_conv_depthwise[0][0]']
 tchNormalization)
```

```
 expanded_conv_depthwise_relu (  (None, 50, 50, 32)  0
['expanded_conv_depthwise_BN[0][0
 ReLU)                                                                ]']

 expanded_conv_project (Conv2D)  (None, 50, 50, 16)  512
['expanded_conv_depthwise_relu[0]
                                                                      [0]']

 expanded_conv_project_BN (Batc  (None, 50, 50, 16)  64
['expanded_conv_project[0][0]']
 hNormalization)

 block_1_expand (Conv2D)        (None, 50, 50, 96)   1536
['expanded_conv_project_BN[0][0]'
                                                                      ]


 block_1_expand_BN (BatchNormal  (None, 50, 50, 96)  384
['block_1_expand[0][0]']
 ization)

 block_1_expand_relu (ReLU)     (None, 50, 50, 96)   0
['block_1_expand_BN[0][0]']

 block_1_pad (ZeroPadding2D)    (None, 51, 51, 96)   0
['block_1_expand_relu[0][0]']

 block_1_depthwise (DepthwiseCo  (None, 25, 25, 96)  864
['block_1_pad[0][0]']
 nv2D)

 block_1_depthwise_BN (BatchNor  (None, 25, 25, 96)  384
['block_1_depthwise[0][0]']
 malization)

 block_1_depthwise_relu (ReLU)  (None, 25, 25, 96)   0
['block_1_depthwise_BN[0][0]']

 block_1_project (Conv2D)       (None, 25, 25, 24)   2304
['block_1_depthwise_relu[0][0]']

 block_1_project_BN (BatchNorma  (None, 25, 25, 24)  96
['block_1_project[0][0]']
 lization)

 block_2_expand (Conv2D)        (None, 25, 25, 144)  3456
['block_1_project_BN[0][0]']

 block_2_expand_BN (BatchNormal  (None, 25, 25, 144)  576
```

```
['block_2_expand[0][0]']
 ization)

 block_2_expand_relu (ReLU)     (None, 25, 25, 144)  0
['block_2_expand_BN[0][0]']

 block_2_depthwise (DepthwiseCo  (None, 25, 25, 144)  1296
['block_2_expand_relu[0][0]']
 nv2D)

 block_2_depthwise_BN (BatchNor  (None, 25, 25, 144)  576
['block_2_depthwise[0][0]']
 malization)

 block_2_depthwise_relu (ReLU)  (None, 25, 25, 144)  0
['block_2_depthwise_BN[0][0]']

 block_2_project (Conv2D)       (None, 25, 25, 24)   3456
['block_2_depthwise_relu[0][0]']

 block_2_project_BN (BatchNorma  (None, 25, 25, 24)  96
['block_2_project[0][0]']
 lization)

 block_2_add (Add)              (None, 25, 25, 24)   0
['block_1_project_BN[0][0]',
'block_2_project_BN[0][0]']

 block_3_expand (Conv2D)        (None, 25, 25, 144)  3456
['block_2_add[0][0]']

 block_3_expand_BN (BatchNormal  (None, 25, 25, 144)  576
['block_3_expand[0][0]']
 ization)

 block_3_expand_relu (ReLU)     (None, 25, 25, 144)  0
['block_3_expand_BN[0][0]']

 block_3_pad (ZeroPadding2D)    (None, 27, 27, 144)  0
['block_3_expand_relu[0][0]']

 block_3_depthwise (DepthwiseCo  (None, 13, 13, 144)  1296
['block_3_pad[0][0]']
 nv2D)

 block_3_depthwise_BN (BatchNor  (None, 13, 13, 144)  576
['block_3_depthwise[0][0]']
 malization)
```

```
 block_3_depthwise_relu (ReLU)  (None, 13, 13, 144)  0
['block_3_depthwise_BN[0][0]']

 block_3_project (Conv2D)       (None, 13, 13, 32)   4608
['block_3_depthwise_relu[0][0]']

 block_3_project_BN (BatchNorma  (None, 13, 13, 32)  128
['block_3_project[0][0]']
 lization)

 block_4_expand (Conv2D)        (None, 13, 13, 192)  6144
['block_3_project_BN[0][0]']

 block_4_expand_BN (BatchNormal  (None, 13, 13, 192)  768
['block_4_expand[0][0]']
 ization)

 block_4_expand_relu (ReLU)     (None, 13, 13, 192)  0
['block_4_expand_BN[0][0]']

 block_4_depthwise (DepthwiseCo  (None, 13, 13, 192)  1728
['block_4_expand_relu[0][0]']
 nv2D)

 block_4_depthwise_BN (BatchNor  (None, 13, 13, 192)  768
['block_4_depthwise[0][0]']
 malization)

 block_4_depthwise_relu (ReLU)  (None, 13, 13, 192)  0
['block_4_depthwise_BN[0][0]']

 block_4_project (Conv2D)       (None, 13, 13, 32)   6144
['block_4_depthwise_relu[0][0]']

 block_4_project_BN (BatchNorma  (None, 13, 13, 32)  128
['block_4_project[0][0]']
 lization)

 block_4_add (Add)              (None, 13, 13, 32)   0
['block_3_project_BN[0][0]',
'block_4_project_BN[0][0]']

 block_5_expand (Conv2D)        (None, 13, 13, 192)  6144
['block_4_add[0][0]']

 block_5_expand_BN (BatchNormal  (None, 13, 13, 192)  768
['block_5_expand[0][0]']
```

```
 ization)

 block_5_expand_relu (ReLU)     (None, 13, 13, 192)  0
['block_5_expand_BN[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 13, 13, 192)  1728
['block_5_expand_relu[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 13, 13, 192)  768
['block_5_depthwise[0][0]']
 malization)

 block_5_depthwise_relu (ReLU)  (None, 13, 13, 192)  0
['block_5_depthwise_BN[0][0]']

 block_5_project (Conv2D)       (None, 13, 13, 32)   6144
['block_5_depthwise_relu[0][0]']

 block_5_project_BN (BatchNorma  (None, 13, 13, 32)  128
['block_5_project[0][0]']
 lization)

 block_5_add (Add)              (None, 13, 13, 32)   0
['block_4_add[0][0]',
'block_5_project_BN[0][0]']

 block_6_expand (Conv2D)        (None, 13, 13, 192)  6144
['block_5_add[0][0]']

 block_6_expand_BN (BatchNormal  (None, 13, 13, 192)  768
['block_6_expand[0][0]']
 ization)

 block_6_expand_relu (ReLU)     (None, 13, 13, 192)  0
['block_6_expand_BN[0][0]']

 block_6_pad (ZeroPadding2D)    (None, 15, 15, 192)  0
['block_6_expand_relu[0][0]']

 block_6_depthwise (DepthwiseCo  (None, 7, 7, 192)   1728
['block_6_pad[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor  (None, 7, 7, 192)   768
['block_6_depthwise[0][0]']
 malization)
```

```
 block_6_depthwise_relu (ReLU)  (None, 7, 7, 192)    0
['block_6_depthwise_BN[0][0]']

 block_6_project (Conv2D)       (None, 7, 7, 64)     12288
['block_6_depthwise_relu[0][0]']

 block_6_project_BN (BatchNorma  (None, 7, 7, 64)    256
['block_6_project[0][0]']
 lization)

 block_7_expand (Conv2D)        (None, 7, 7, 384)    24576
['block_6_project_BN[0][0]']

 block_7_expand_BN (BatchNormal  (None, 7, 7, 384)   1536
['block_7_expand[0][0]']
 ization)

 block_7_expand_relu (ReLU)     (None, 7, 7, 384)    0
['block_7_expand_BN[0][0]']

 block_7_depthwise (DepthwiseCo  (None, 7, 7, 384)   3456
['block_7_expand_relu[0][0]']
 nv2D)

 block_7_depthwise_BN (BatchNor  (None, 7, 7, 384)   1536
['block_7_depthwise[0][0]']
 malization)

 block_7_depthwise_relu (ReLU)  (None, 7, 7, 384)    0
['block_7_depthwise_BN[0][0]']

 block_7_project (Conv2D)       (None, 7, 7, 64)     24576
['block_7_depthwise_relu[0][0]']

 block_7_project_BN (BatchNorma  (None, 7, 7, 64)    256
['block_7_project[0][0]']
 lization)

 block_7_add (Add)              (None, 7, 7, 64)     0
['block_6_project_BN[0][0]',
'block_7_project_BN[0][0]']

 block_8_expand (Conv2D)        (None, 7, 7, 384)    24576
['block_7_add[0][0]']

 block_8_expand_BN (BatchNormal  (None, 7, 7, 384)   1536
['block_8_expand[0][0]']
 ization)
```

```
 block_8_expand_relu (ReLU)     (None, 7, 7, 384)    0
['block_8_expand_BN[0][0]']

 block_8_depthwise (DepthwiseCo  (None, 7, 7, 384)   3456
['block_8_expand_relu[0][0]']
 nv2D)

 block_8_depthwise_BN (BatchNor  (None, 7, 7, 384)   1536
['block_8_depthwise[0][0]']
 malization)

 block_8_depthwise_relu (ReLU)  (None, 7, 7, 384)    0
['block_8_depthwise_BN[0][0]']

 block_8_project (Conv2D)       (None, 7, 7, 64)     24576
['block_8_depthwise_relu[0][0]']

 block_8_project_BN (BatchNorma  (None, 7, 7, 64)    256
['block_8_project[0][0]']
 lization)

 block_8_add (Add)              (None, 7, 7, 64)     0
['block_7_add[0][0]',
'block_8_project_BN[0][0]']

 block_9_expand (Conv2D)        (None, 7, 7, 384)    24576
['block_8_add[0][0]']

 block_9_expand_BN (BatchNormal  (None, 7, 7, 384)   1536
['block_9_expand[0][0]']
 ization)

 block_9_expand_relu (ReLU)     (None, 7, 7, 384)    0
['block_9_expand_BN[0][0]']

 block_9_depthwise (DepthwiseCo  (None, 7, 7, 384)   3456
['block_9_expand_relu[0][0]']
 nv2D)

 block_9_depthwise_BN (BatchNor  (None, 7, 7, 384)   1536
['block_9_depthwise[0][0]']
 malization)

 block_9_depthwise_relu (ReLU)  (None, 7, 7, 384)    0
['block_9_depthwise_BN[0][0]']

 block_9_project (Conv2D)       (None, 7, 7, 64)     24576
```

```
['block_9_depthwise_relu[0][0]']

 block_9_project_BN (BatchNorma  (None, 7, 7, 64)    256
['block_9_project[0][0]']
 lization)

 block_9_add (Add)              (None, 7, 7, 64)     0
['block_8_add[0][0]',
'block_9_project_BN[0][0]']

 block_10_expand (Conv2D)       (None, 7, 7, 384)    24576
['block_9_add[0][0]']

 block_10_expand_BN (BatchNorma  (None, 7, 7, 384)   1536
['block_10_expand[0][0]']
 lization)

 block_10_expand_relu (ReLU)    (None, 7, 7, 384)    0
['block_10_expand_BN[0][0]']

 block_10_depthwise (DepthwiseC  (None, 7, 7, 384)   3456
['block_10_expand_relu[0][0]']
 onv2D)

 block_10_depthwise_BN (BatchNo  (None, 7, 7, 384)   1536
['block_10_depthwise[0][0]']
 rmalization)

 block_10_depthwise_relu (ReLU)  (None, 7, 7, 384)   0
['block_10_depthwise_BN[0][0]']

 block_10_project (Conv2D)      (None, 7, 7, 96)     36864
['block_10_depthwise_relu[0][0]']

 block_10_project_BN (BatchNorm  (None, 7, 7, 96)    384
['block_10_project[0][0]']
 alization)

 block_11_expand (Conv2D)       (None, 7, 7, 576)    55296
['block_10_project_BN[0][0]']

 block_11_expand_BN (BatchNorma  (None, 7, 7, 576)   2304
['block_11_expand[0][0]']
 lization)

 block_11_expand_relu (ReLU)    (None, 7, 7, 576)    0
['block_11_expand_BN[0][0]']
```

```
 block_11_depthwise (DepthwiseC  (None, 7, 7, 576)   5184
['block_11_expand_relu[0][0]']
 onv2D)

 block_11_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304
['block_11_depthwise[0][0]']
 rmalization)

 block_11_depthwise_relu (ReLU)  (None, 7, 7, 576)   0
['block_11_depthwise_BN[0][0]']

 block_11_project (Conv2D)      (None, 7, 7, 96)     55296
['block_11_depthwise_relu[0][0]']

 block_11_project_BN (BatchNorm  (None, 7, 7, 96)    384
['block_11_project[0][0]']
 alization)

 block_11_add (Add)             (None, 7, 7, 96)     0
['block_10_project_BN[0][0]',
'block_11_project_BN[0][0]']

 block_12_expand (Conv2D)       (None, 7, 7, 576)    55296
['block_11_add[0][0]']

 block_12_expand_BN (BatchNorma  (None, 7, 7, 576)   2304
['block_12_expand[0][0]']
 lization)

 block_12_expand_relu (ReLU)    (None, 7, 7, 576)    0
['block_12_expand_BN[0][0]']

 block_12_depthwise (DepthwiseC  (None, 7, 7, 576)   5184
['block_12_expand_relu[0][0]']
 onv2D)

 block_12_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304
['block_12_depthwise[0][0]']
 rmalization)

 block_12_depthwise_relu (ReLU)  (None, 7, 7, 576)   0
['block_12_depthwise_BN[0][0]']

 block_12_project (Conv2D)      (None, 7, 7, 96)     55296
['block_12_depthwise_relu[0][0]']

 block_12_project_BN (BatchNorm  (None, 7, 7, 96)    384
['block_12_project[0][0]']
```

```
 alization)

 block_12_add (Add)             (None, 7, 7, 96)     0
['block_11_add[0][0]',
'block_12_project_BN[0][0]']

 block_13_expand (Conv2D)       (None, 7, 7, 576)    55296
['block_12_add[0][0]']

 block_13_expand_BN (BatchNorma  (None, 7, 7, 576)   2304
['block_13_expand[0][0]']
 lization)

 block_13_expand_relu (ReLU)    (None, 7, 7, 576)    0
['block_13_expand_BN[0][0]']

 block_13_pad (ZeroPadding2D)   (None, 9, 9, 576)    0
['block_13_expand_relu[0][0]']

 block_13_depthwise (DepthwiseC  (None, 4, 4, 576)   5184
['block_13_pad[0][0]']
 onv2D)

 block_13_depthwise_BN (BatchNo  (None, 4, 4, 576)   2304
['block_13_depthwise[0][0]']
 rmalization)

 block_13_depthwise_relu (ReLU)  (None, 4, 4, 576)   0
['block_13_depthwise_BN[0][0]']

 block_13_project (Conv2D)      (None, 4, 4, 160)    92160
['block_13_depthwise_relu[0][0]']

 block_13_project_BN (BatchNorm  (None, 4, 4, 160)   640
['block_13_project[0][0]']
 alization)

 block_14_expand (Conv2D)       (None, 4, 4, 960)    153600
['block_13_project_BN[0][0]']

 block_14_expand_BN (BatchNorma  (None, 4, 4, 960)   3840
['block_14_expand[0][0]']
 lization)

 block_14_expand_relu (ReLU)    (None, 4, 4, 960)    0
['block_14_expand_BN[0][0]']

 block_14_depthwise (DepthwiseC  (None, 4, 4, 960)   8640
```

```
['block_14_expand_relu[0][0]']
 onv2D)

 block_14_depthwise_BN (BatchNo  (None, 4, 4, 960)   3840
['block_14_depthwise[0][0]']
 rmalization)

 block_14_depthwise_relu (ReLU)  (None, 4, 4, 960)   0
['block_14_depthwise_BN[0][0]']

 block_14_project (Conv2D)      (None, 4, 4, 160)    153600
['block_14_depthwise_relu[0][0]']

 block_14_project_BN (BatchNorm  (None, 4, 4, 160)   640
['block_14_project[0][0]']
 alization)

 block_14_add (Add)             (None, 4, 4, 160)    0
['block_13_project_BN[0][0]',
'block_14_project_BN[0][0]']

 block_15_expand (Conv2D)       (None, 4, 4, 960)    153600
['block_14_add[0][0]']

 block_15_expand_BN (BatchNorma  (None, 4, 4, 960)   3840
['block_15_expand[0][0]']
 lization)

 block_15_expand_relu (ReLU)    (None, 4, 4, 960)    0
['block_15_expand_BN[0][0]']

 block_15_depthwise (DepthwiseC  (None, 4, 4, 960)   8640
['block_15_expand_relu[0][0]']
 onv2D)

 block_15_depthwise_BN (BatchNo  (None, 4, 4, 960)   3840
['block_15_depthwise[0][0]']
 rmalization)

 block_15_depthwise_relu (ReLU)  (None, 4, 4, 960)   0
['block_15_depthwise_BN[0][0]']

 block_15_project (Conv2D)      (None, 4, 4, 160)    153600
['block_15_depthwise_relu[0][0]']

 block_15_project_BN (BatchNorm  (None, 4, 4, 160)   640
['block_15_project[0][0]']
 alization)
```

```
 block_15_add (Add)             (None, 4, 4, 160)    0
['block_14_add[0][0]',
'block_15_project_BN[0][0]']

 block_16_expand (Conv2D)       (None, 4, 4, 960)    153600
['block_15_add[0][0]']

 block_16_expand_BN (BatchNorma  (None, 4, 4, 960)   3840
['block_16_expand[0][0]']
 lization)

 block_16_expand_relu (ReLU)    (None, 4, 4, 960)    0
['block_16_expand_BN[0][0]']

 block_16_depthwise (DepthwiseC  (None, 4, 4, 960)   8640
['block_16_expand_relu[0][0]']
 onv2D)

 block_16_depthwise_BN (BatchNo  (None, 4, 4, 960)   3840
['block_16_depthwise[0][0]']
 rmalization)

 block_16_depthwise_relu (ReLU)  (None, 4, 4, 960)   0
['block_16_depthwise_BN[0][0]']

 block_16_project (Conv2D)      (None, 4, 4, 320)    307200
['block_16_depthwise_relu[0][0]']

 block_16_project_BN (BatchNorm  (None, 4, 4, 320)   1280
['block_16_project[0][0]']
 alization)

 Conv_1 (Conv2D)                (None, 4, 4, 1280)   409600
['block_16_project_BN[0][0]']

 Conv_1_bn (BatchNormalization)  (None, 4, 4, 1280)  5120
['Conv_1[0][0]']

 out_relu (ReLU)                (None, 4, 4, 1280)   0
['Conv_1_bn[0][0]']

==================================================================================
==================
Total params: 2,257,984
Trainable params: 0
Non-trainable params: 2,257,984
----------------------------------------------------------------------------------
```

```
------------------
```

[30]:
```
global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
feature_batch_average = global_average_layer(feature_batch)
print(feature_batch_average.shape)
```

```
(32, 1280)
```

[31]:
```
prediction_layer = tf.keras.layers.Dense(3)
prediction_batch = prediction_layer(feature_batch_average)
print(prediction_batch.shape)
```

```
(32, 3)
```

[32]:
```
inputs = tf.keras.Input(shape=(100, 100, 3))
x = data_augmentation(inputs)
x = preprocess_input(x)
x = base_model(x, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.2)(x)
outputs = prediction_layer(x)
model = tf.keras.Model(inputs, outputs)
```

[41]:
```
base_learning_rate = 0.0001
model.compile(optimizer=tf.keras.optimizers.
 ↪Adam(learning_rate=base_learning_rate),
              loss=tf.keras.losses.CategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

[42]:
```
model.summary()
```

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 100, 100, 3)]     0

 sequential (Sequential)     (None, 100, 100, 3)       0

 tf.math.truediv_1 (TFOpLamb  (None, 100, 100, 3)      0
 da)

 tf.math.subtract_1 (TFOpLam  (None, 100, 100, 3)      0
 bda)

 mobilenetv2_1.00_224 (Funct  (None, 4, 4, 1280)       2257984
 ional)
```

```
 global_average_pooling2d_2   (None, 1280)              0
 (GlobalAveragePooling2D)

 dropout_1 (Dropout)          (None, 1280)              0

 dense_1 (Dense)              (None, 3)                 3843

=================================================================
Total params: 2,261,827
Trainable params: 3,843
Non-trainable params: 2,257,984
_________________________________________________________________
```

[43]:
```
len(model.trainable_variables)
```

[43]: 2

[44]:
```
initial_epochs = 10

print(val_ds)

loss0, accuracy0 = model.evaluate(val_ds)
```

```
<SkipDataset element_spec=(TensorSpec(shape=(None, 100, 100, 3),
dtype=tf.float32, name=None), TensorSpec(shape=(None, 3), dtype=tf.float32,
name=None))>
8/8 [==============================] - 4s 257ms/step - loss: 1.5204 - accuracy:
0.4043
```

[45]:
```
print("initial loss: {:.2f}".format(loss0))
print("initial accuracy: {:.2f}".format(accuracy0))
```

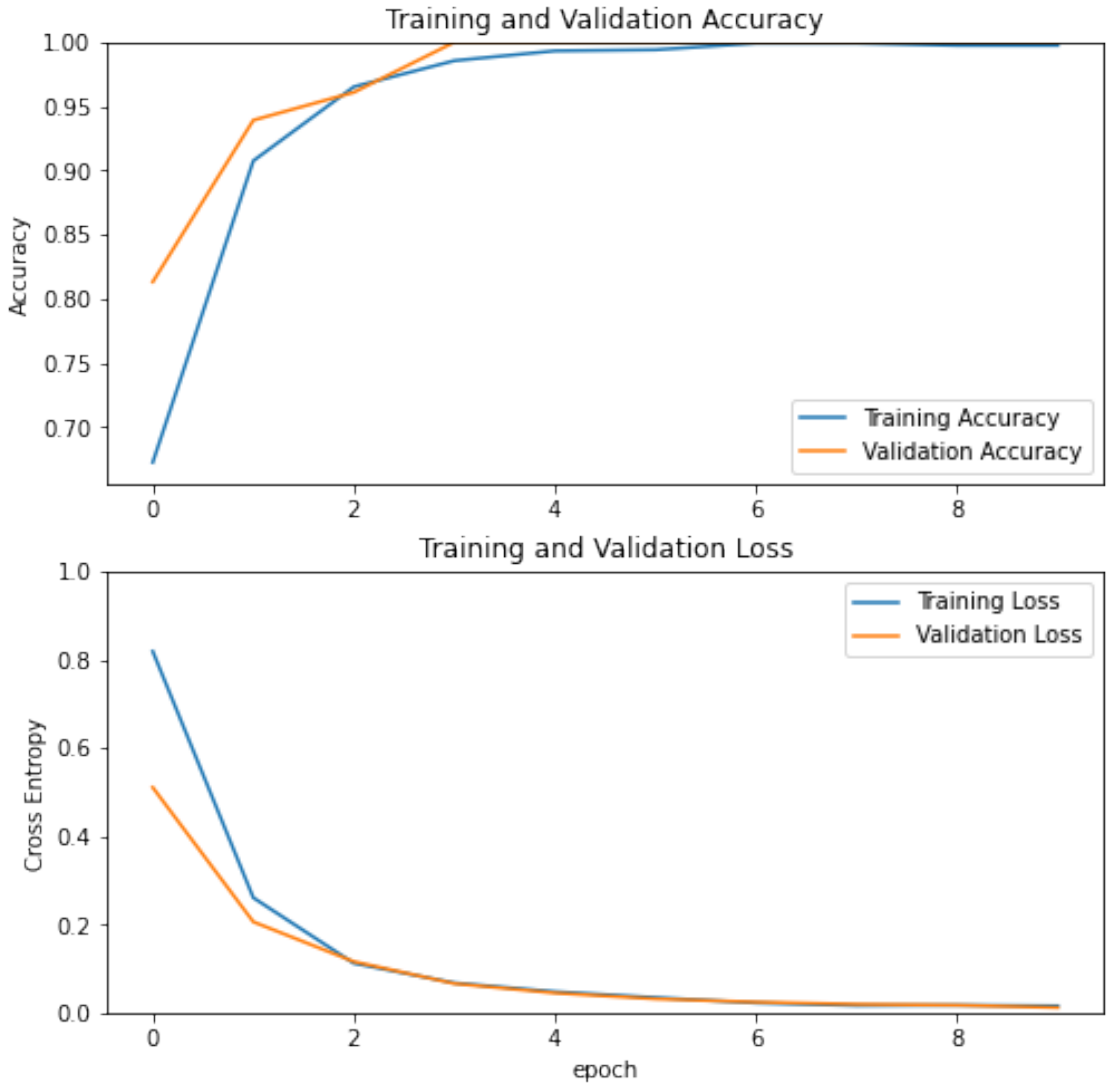
```
initial loss: 1.52
initial accuracy: 0.40
```

[46]:
```
history = model.fit(train_ds,
                    epochs=initial_epochs,
                    validation_data=val_ds)
```

```
Epoch 1/10
37/37 [==============================] - 19s 414ms/step - loss: 0.8191 -
accuracy: 0.6723 - val_loss: 0.5107 - val_accuracy: 0.8130
Epoch 2/10
37/37 [==============================] - 17s 450ms/step - loss: 0.2607 -
accuracy: 0.9075 - val_loss: 0.2060 - val_accuracy: 0.9391
Epoch 3/10
37/37 [==============================] - 16s 420ms/step - loss: 0.1120 -
accuracy: 0.9652 - val_loss: 0.1159 - val_accuracy: 0.9609
```

```
Epoch 4/10
37/37 [==============================] - 16s 429ms/step - loss: 0.0675 -
accuracy: 0.9856 - val_loss: 0.0660 - val_accuracy: 1.0000
Epoch 5/10
37/37 [==============================] - 17s 457ms/step - loss: 0.0477 -
accuracy: 0.9932 - val_loss: 0.0447 - val_accuracy: 1.0000
Epoch 6/10
37/37 [==============================] - 16s 418ms/step - loss: 0.0344 -
accuracy: 0.9941 - val_loss: 0.0314 - val_accuracy: 1.0000
Epoch 7/10
37/37 [==============================] - 18s 490ms/step - loss: 0.0222 -
accuracy: 0.9992 - val_loss: 0.0241 - val_accuracy: 1.0000
Epoch 8/10
37/37 [==============================] - 16s 414ms/step - loss: 0.0163 -
accuracy: 0.9992 - val_loss: 0.0192 - val_accuracy: 1.0000
Epoch 9/10
37/37 [==============================] - 16s 414ms/step - loss: 0.0171 -
accuracy: 0.9975 - val_loss: 0.0171 - val_accuracy: 1.0000
Epoch 10/10
37/37 [==============================] - 16s 412ms/step - loss: 0.0156 -
accuracy: 0.9975 - val_loss: 0.0117 - val_accuracy: 1.0000
```

[47]:
```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.ylabel('Accuracy')
plt.ylim([min(plt.ylim()),1])
plt.title('Training and Validation Accuracy')

plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.ylabel('Cross Entropy')
plt.ylim([0,1])
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.show()
```



[48]:
```
base_model.trainable = True
```

[49]:
```
# Let's take a look to see how many layers are in the base model
print("Number of layers in the base model: ", len(base_model.layers))

# Fine-tune from this layer onwards
fine_tune_at = 100

# Freeze all the layers before the `fine_tune_at` layer
for layer in base_model.layers[:fine_tune_at]:
  layer.trainable = False
```

```
Number of layers in the base model:  154
```

[58]:
```
model.compile(loss=tf.keras.losses.CategoricalCrossentropy(from_logits=True),
              optimizer = tf.keras.optimizers.
 ↪RMSprop(learning_rate=base_learning_rate/10),
              metrics=['accuracy'])
```

[59]:
```
model.summary()
```

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 100, 100, 3)]     0

 sequential (Sequential)     (None, 100, 100, 3)       0

 tf.math.truediv_1 (TFOpLamb  (None, 100, 100, 3)      0
 da)

 tf.math.subtract_1 (TFOpLam  (None, 100, 100, 3)      0
 bda)

 mobilenetv2_1.00_224 (Funct  (None, 4, 4, 1280)       2257984
 ional)

 global_average_pooling2d_2   (None, 1280)             0
 (GlobalAveragePooling2D)

 dropout_1 (Dropout)         (None, 1280)              0

 dense_1 (Dense)             (None, 3)                 3843

=================================================================
Total params: 2,261,827
Trainable params: 1,865,283
Non-trainable params: 396,544
_________________________________________________________________
```

[60]:
```
len(model.trainable_variables)
```

[60]:
```
56
```

[61]:
```
fine_tune_epochs = 10
total_epochs =  initial_epochs + fine_tune_epochs

history_fine = model.fit(train_ds,
                         epochs=total_epochs,
                         initial_epoch=history.epoch[-1],
```

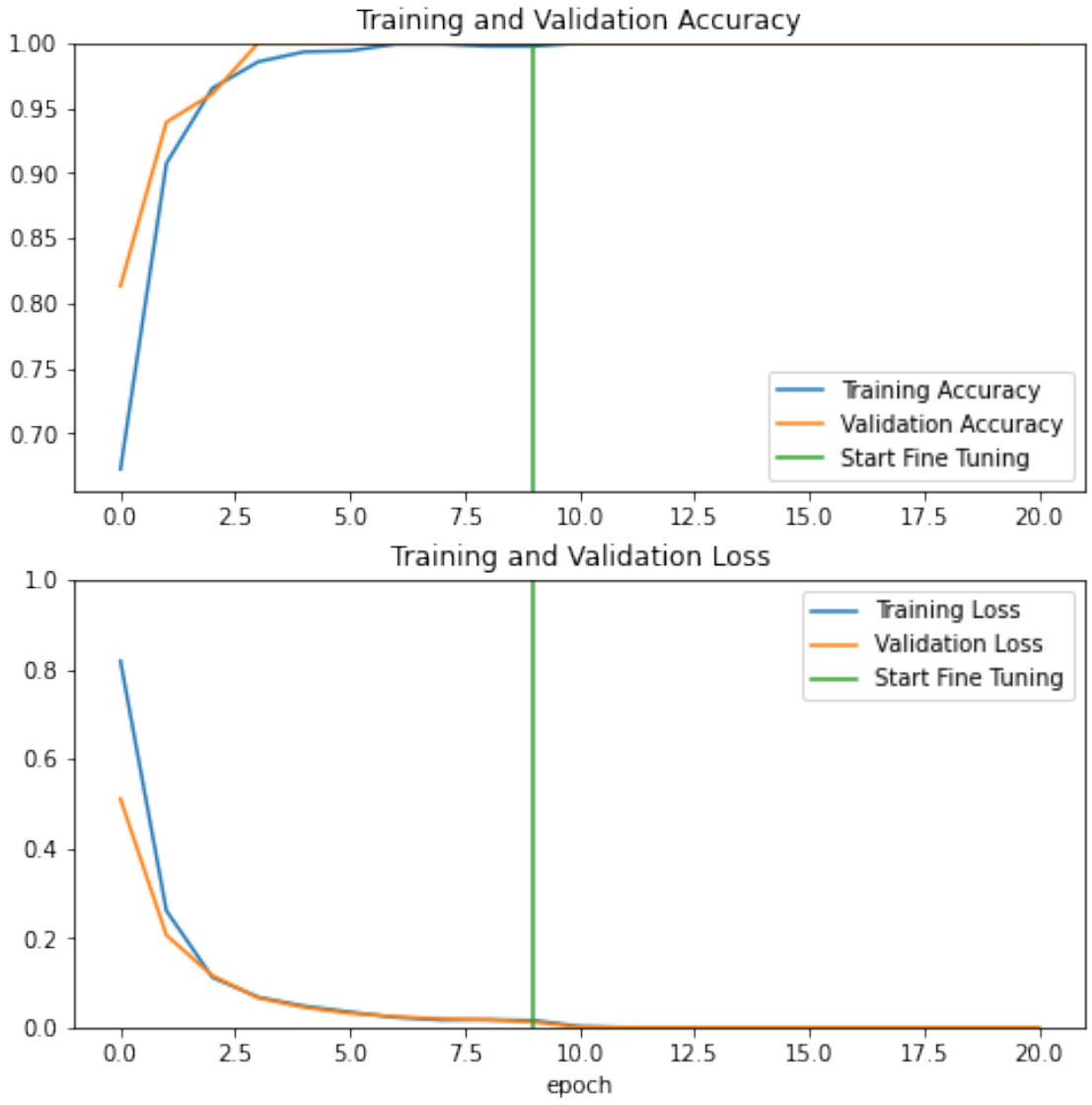
```
                                validation_data=val_ds)
```

```
Epoch 10/20
37/37 [==============================] - 58s 691ms/step - loss: 0.0025 -
accuracy: 1.0000 - val_loss: 4.2921e-04 - val_accuracy: 1.0000
Epoch 11/20
37/37 [==============================] - 26s 682ms/step - loss: 1.2336e-04 -
accuracy: 1.0000 - val_loss: 1.1158e-04 - val_accuracy: 1.0000
Epoch 12/20
37/37 [==============================] - 28s 746ms/step - loss: 3.3883e-05 -
accuracy: 1.0000 - val_loss: 1.5006e-05 - val_accuracy: 1.0000
Epoch 13/20
37/37 [==============================] - 25s 674ms/step - loss: 5.0725e-05 -
accuracy: 1.0000 - val_loss: 3.2375e-06 - val_accuracy: 1.0000
Epoch 14/20
37/37 [==============================] - 26s 682ms/step - loss: 2.5234e-06 -
accuracy: 1.0000 - val_loss: 6.1263e-07 - val_accuracy: 1.0000
Epoch 15/20
37/37 [==============================] - 25s 679ms/step - loss: 1.7295e-06 -
accuracy: 1.0000 - val_loss: 2.7988e-08 - val_accuracy: 1.0000
Epoch 16/20
37/37 [==============================] - 25s 676ms/step - loss: 1.0980e-07 -
accuracy: 1.0000 - val_loss: 1.8141e-08 - val_accuracy: 1.0000
Epoch 17/20
37/37 [==============================] - 25s 681ms/step - loss: 4.5538e-08 -
accuracy: 1.0000 - val_loss: 1.5031e-08 - val_accuracy: 1.0000
Epoch 18/20
37/37 [==============================] - 27s 736ms/step - loss: 9.0266e-08 -
accuracy: 1.0000 - val_loss: 2.0732e-09 - val_accuracy: 1.0000
Epoch 19/20
37/37 [==============================] - 25s 679ms/step - loss: 1.2245e-08 -
accuracy: 1.0000 - val_loss: 5.7013e-09 - val_accuracy: 1.0000
Epoch 20/20
37/37 [==============================] - 25s 677ms/step - loss: 2.0239e-09 -
accuracy: 1.0000 - val_loss: 3.6281e-09 - val_accuracy: 1.0000
```

[62]:
```
acc += history_fine.history['accuracy']
val_acc += history_fine.history['val_accuracy']

loss += history_fine.history['loss']
val_loss += history_fine.history['val_loss']
```

[64]:
```
plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.ylim([min(plt.ylim()), 1.0])
```

```
plt.plot([initial_epochs-1,initial_epochs-1],
          plt.ylim(), label='Start Fine Tuning')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.ylim([0, 1.0])
plt.plot([initial_epochs-1,initial_epochs-1],
         plt.ylim(), label='Start Fine Tuning')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.show()
```



[65]:
```
loss, accuracy = model.evaluate(test_dataset)
print('Test accuracy :', accuracy)
```

```
2/2 [==============================] - 1s 240ms/step - loss: 0.0000e+00 -
accuracy: 1.0000
Test accuracy : 1.0
```

4 Analysis

The best performing model was the CNN application to the sequential model - this is most likely due to the nature of the dataset itself, since it's small and with limited classes. The Pretrained Model and Transfer Learning model had a similar result but took significantly more time to process - this model is probably better for less focused datasets since it takes more time but is more accurate. They both did significantly better than the basic sequential model - which is why the sequential model has these additional parameters and options to better fine tune it towards a particular dataset.