```
# download functions for course
!wget https://raw.githubusercontent.com/mrdbourke/tensorflow-deep-learning/main/extras/helper
```

```
--2022-09-29 00:46:15--  https://raw.githubusercontent.com/mrdbourke/tensorflow-deep-lea
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443
HTTP request sent, awaiting response... 200 OK
Length: 10246 (10K) [text/plain]
Saving to: ‘helper_functions.py’

helper_functions.py 100%[===================>]  10.01K  --.-KB/s    in 0s

2022-09-29 00:46:16 (21.7 MB/s) - ‘helper_functions.py’ saved [10246/10246]
```

```
from helper_functions import create_tensorboard_callback, plot_loss_curves, walk_through_dir,
```

```
#download data for project
!wget https://storage.googleapis.com/ztm_tf_course/food_vision/10_food_classes_10_percent.zip

unzip_data('10_food_classes_10_percent.zip')
```

```
--2022-09-29 00:46:20--  https://storage.googleapis.com/ztm_tf_course/food_vision/10_foo
Resolving storage.googleapis.com (storage.googleapis.com)... 74.125.20.128, 108.177.98.1
Connecting to storage.googleapis.com (storage.googleapis.com)|74.125.20.128|:443... conn
HTTP request sent, awaiting response... 200 OK
Length: 168546183 (161M) [application/zip]
Saving to: ‘10_food_classes_10_percent.zip’

10_food_classes_10_ 100%[===================>] 160.74M   102MB/s    in 1.6s

2022-09-29 00:46:22 (102 MB/s) - ‘10_food_classes_10_percent.zip’ saved [168546183/16854
```

```
#review directory of data
walk_through_dir('10_food_classes_10_percent')
```

```
There are 2 directories and 0 images in '10_food_classes_10_percent'.
There are 10 directories and 0 images in '10_food_classes_10_percent/test'.
There are 0 directories and 250 images in '10_food_classes_10_percent/test/sushi'.
There are 0 directories and 250 images in '10_food_classes_10_percent/test/chicken_wings
There are 0 directories and 250 images in '10_food_classes_10_percent/test/steak'.
There are 0 directories and 250 images in '10_food_classes_10_percent/test/hamburger'.
There are 0 directories and 250 images in '10_food_classes_10_percent/test/chicken_curry
There are 0 directories and 250 images in '10_food_classes_10_percent/test/fried_rice'.
There are 0 directories and 250 images in '10_food_classes_10_percent/test/pizza'.
There are 0 directories and 250 images in '10_food_classes_10_percent/test/ice_cream'.
There are 0 directories and 250 images in '10_food_classes_10_percent/test/grilled_salmo
There are 0 directories and 250 images in '10_food_classes_10_percent/test/ramen'.
```

```
There are 10 directories and 0 images in '10_food_classes_10_percent/train'.
There are 0 directories and 75 images in '10_food_classes_10_percent/train/sushi'.
There are 0 directories and 75 images in '10_food_classes_10_percent/train/chicken_wings
There are 0 directories and 75 images in '10_food_classes_10_percent/train/steak'.
There are 0 directories and 75 images in '10_food_classes_10_percent/train/hamburger'.
There are 0 directories and 75 images in '10_food_classes_10_percent/train/chicken_curry
There are 0 directories and 75 images in '10_food_classes_10_percent/train/fried_rice'.
There are 0 directories and 75 images in '10_food_classes_10_percent/train/pizza'.
There are 0 directories and 75 images in '10_food_classes_10_percent/train/ice_cream'.
There are 0 directories and 75 images in '10_food_classes_10_percent/train/grilled_salmo
There are 0 directories and 75 images in '10_food_classes_10_percent/train/ramen'.
```

```
train_dir = '10_food_classes_10_percent/train'
test_dir = '10_food_classes_10_percent/test'


import tensorflow as tf

IMG_SIZE = (224,224)
BATCH_SIZE = 32

#prep train and test sets
train_data_10_percent = tf.keras.preprocessing.image_dataset_from_directory(directory=train_d
                                                                            image_size=IMG_S
                                                                            label_mode='cate
                                                                            batch_size=BATCH

test_data = tf.keras.preprocessing.image_dataset_from_directory(directory=test_dir,
                                                                image_size=IMG_SIZE,
                                                                label_mode='categorical',
                                                                batch_size=BATCH_SIZE)
```

```
Found 750 files belonging to 10 classes.
Found 2500 files belonging to 10 classes.
```

```
#review classes in dataset
train_data_10_percent.class_names
```

```
['chicken_curry',
 'chicken_wings',
 'fried_rice',
 'grilled_salmon',
 'hamburger',
 'ice_cream',
 'pizza',
 'ramen',
 'steak',
 'sushi']
```

Test the base model (EfficientNet B0)

```
#Model 0

#create base model with our own output layer
base_model = tf.keras.applications.EfficientNetB0(include_top=False)

#freeze base model
base_model.trainable = False

#create inputs into our model
inputs = tf.keras.layers.Input(shape=(224,224,3), name='input_layer')

#normalize inputs if needed
#x = tf.keras.layers.experimental.preprocessing.Rescaling(1./255)(inputs)

#pass inputs to base model
x = base_model(inputs)
print(f'Shape after passing inputs through base model: {x.shape}')
#average pool the putputs of the base
x = tf.keras.layers.GlobalAveragePooling2D(name='global_average_pooling_layer')(x)
print(f'Shape after GlobalAveragePooling2D: {x.shape}')

#create the output layer
outputs = tf.keras.layers.Dense(10, activation='softmax', name='output_layer')(x)

#combine the inputs with the outputs
model_0 = tf.keras.Model(inputs, outputs)

#compile
model_0.compile(loss='categorical_crossentropy',
                    optimizer=tf.keras.optimizers.Adam(),
                    metrics=['accuracy'])
#fit
history_0 = model_0.fit(train_data_10_percent,
                            epochs=5,
                            validation_data=test_data,
                            steps_per_epoch=len(train_data_10_percent),
                            validation_steps=int(0.25 *len(test_data)),
                           callbacks=[create_tensorboard_callback(dir_name='transfer_learnin

    Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0_r
    16711680/16705208 [==============================] - 0s 0us/step
    16719872/16705208 [==============================] - 0s 0us/step
    Shape after passing inputs through base model: (None, 7, 7, 1280)
    Shape after GlobalAveragePooling2D: (None, 1280)
    Saving TensorBoard log files to: transfer_learning/model_0/20220929-004632
    Epoch 1/5
    24/24 [==============================] - 23s 256ms/step - loss: 1.9281 - accuracy: 0.386
    Epoch 2/5
    24/24 [==============================] - 4s 151ms/step - loss: 1.1741 - accuracy: 0.7307
    Epoch 3/5
    24/24 [==============================] - 4s 150ms/step - loss: 0.8549 - accuracy: 0.7987
```

```
Epoch 4/5
24/24 [==============================] - 5s 185ms/step - loss: 0.6959 - accuracy: 0.8267
Epoch 5/5
24/24 [==============================] - 4s 149ms/step - loss: 0.5898 - accuracy: 0.8667
```

```
#evaluate on full test data
model_0.evaluate(test_data)
```

```
79/79 [==============================] - 6s 75ms/step - loss: 0.6166 - accuracy: 0.8348
[0.6165868043899536, 0.8348000049591064]
```

Review the layers and structure of the base model

```
#check the layers in our base model
for layer_number, layer in enumerate(base_model.layers):
  print(layer_number, layer.name)
```

```
178 block6b_expand_activation
179 block6b_dwconv
180 block6b_bn
181 block6b_activation
182 block6b_se_squeeze
183 block6b_se_reshape
184 block6b_se_reduce
185 block6b_se_expand
186 block6b_se_excite
187 block6b_project_conv
188 block6b_project_bn
189 block6b_drop
190 block6b_add
191 block6c_expand_conv
192 block6c_expand_bn
193 block6c_expand_activation
194 block6c_dwconv
195 block6c_bn
196 block6c_activation
197 block6c_se_squeeze
198 block6c_se_reshape
199 block6c_se_reduce
200 block6c_se_expand
201 block6c_se_excite
202 block6c_project_conv
203 block6c_project_bn
204 block6c_drop
205 block6c_add
206 block6d_expand_conv
207 block6d_expand_bn
208 block6d_expand_activation
209 block6d_dwconv
210 block6d_bn
211 block6d_activation
212 block6d_se_squeeze
```

```
213 block6d_se_reshape
214 block6d_se_reduce
215 block6d_se_expand
216 block6d_se_excite
217 block6d_project_conv
218 block6d_project_bn
219 block6d_drop
220 block6d_add
221 block7a_expand_conv
222 block7a_expand_bn
223 block7a_expand_activation
224 block7a_dwconv
225 block7a_bn
226 block7a_activation
227 block7a_se_squeeze
228 block7a_se_reshape
229 block7a_se_reduce
230 block7a_se_expand
231 block7a_se_excite
232 block7a_project_conv
233 block7a_project_bn
234 top_conv
235 top_bn
```

```
base_model.summary()
```

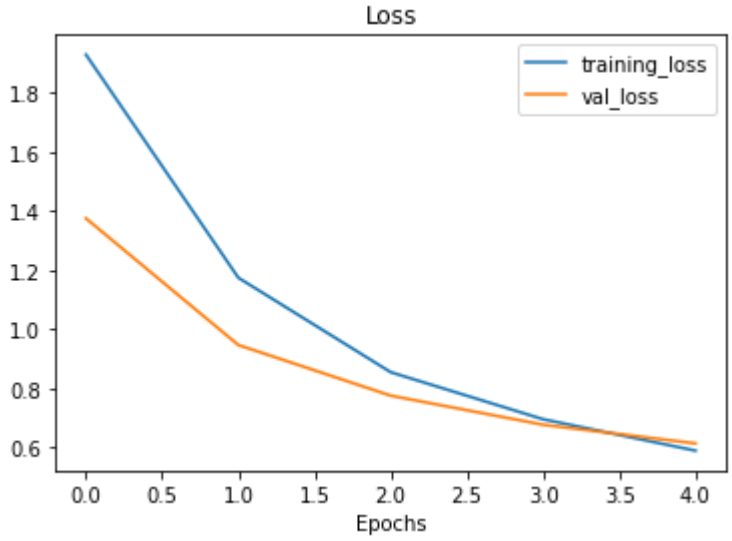
```
                                1152)                                 block6d_se_expand[

 block6d_project_conv (Conv2D)  (None, None, None,   221184      ['block6d_se_excite[0
                                192)

 block6d_project_bn (BatchNorma  (None, None, None,   768        ['block6d_project_co
 lization)                      192)

 block6d_drop (Dropout)         (None, None, None,   0          ['block6d_project_bn
                                192)

 block6d_add (Add)              (None, None, None,   0          ['block6d_drop[0][0]
                                192)                             'block6c_add[0][0]'

 block7a_expand_conv (Conv2D)   (None, None, None,   221184      ['block6d_add[0][0]'
                                1152)

 block7a_expand_bn (BatchNormal  (None, None, None,   4608       ['block7a_expand_conv
 ization)                       1152)

 block7a_expand_activation (Act  (None, None, None,   0          ['block7a_expand_bn[0
 ivation)                       1152)

 block7a_dwconv (DepthwiseConv2  (None, None, None,   10368      ['block7a_expand_act
 D)                             1152)                            ']

 block7a_bn (BatchNormalization  (None, None, None,   4608       ['block7a_dwconv[0][0
 )                              1152)

 block7a_activation (Activation  (None, None, None,   0          ['block7a_bn[0][0]']
 )                              1152)
```

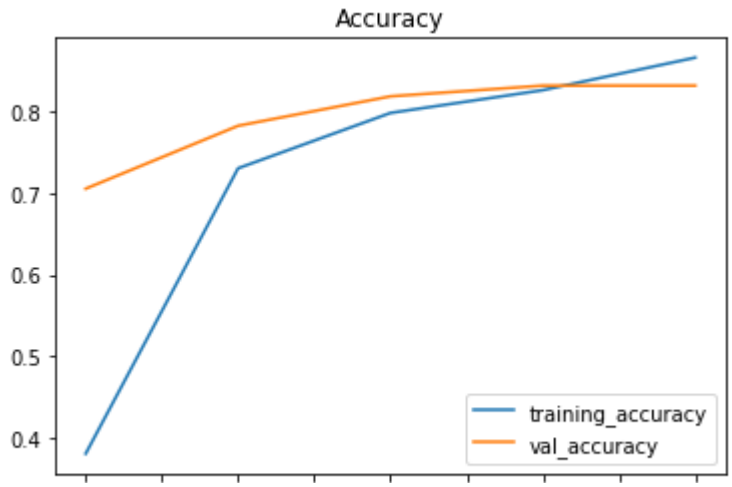
```
block7a_se_squeeze (GlobalAver  (None, 1152)        0           ['block7a_activation
agePooling2D)

block7a_se_reshape (Reshape)   (None, 1, 1, 1152)   0           ['block7a_se_squeeze

block7a_se_reduce (Conv2D)     (None, 1, 1, 48)     55344       ['block7a_se_reshape

block7a_se_expand (Conv2D)     (None, 1, 1, 1152)   56448       ['block7a_se_reduce[0

block7a_se_excite (Multiply)   (None, None, None,   0           ['block7a_activation
                               1152)                            'block7a_se_expand[0

block7a_project_conv (Conv2D)  (None, None, None,   368640      ['block7a_se_excite[0
                               320)

block7a_project_bn (BatchNorma  (None, None, None,   1280       ['block7a_project_co
lization)                      320)

top_conv (Conv2D)              (None, None, None,   409600      ['block7a_project_bn
                               1280)

top_bn (BatchNormalization)    (None, None, None,   5120        ['top_conv[0][0]']
                               1280)

top_activation (Activation)    (None, None, None,   0           ['top_bn[0][0]']
                               1280)
```

```
model_0.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_layer (InputLayer)    [(None, 224, 224, 3)]     0

 efficientnetb0 (Functional)  (None, None, None, 1280)  4049571

 global_average_pooling_laye  (None, 1280)             0
 r (GlobalAveragePooling2D)

 output_layer (Dense)        (None, 10)                12810

=================================================================
Total params: 4,062,381
Trainable params: 12,810
Non-trainable params: 4,049,571
_________________________________________________________________
```

```
plot_loss_curves(history_0)
```





Model 0 trained on 10% of the data set with training only allowed on the output layer and it still performed very well

Now we will create several different models:

Model_1: feature extraction transfer learning on 1% of data with augmentation

Model_2: feature extraction transfer learning on 10% with augmentation

Model_3: fine-tuning transfer learning on 10% with augmentation

Model_4: fine-tuning transfer learning on 100% with augmentation

```
#download 1% of data
!wget https://storage.googleapis.com/ztm_tf_course/food_vision/10_food_classes_1_percent.zip

unzip_data('10_food_classes_1_percent.zip')
```

```
--2022-09-29 00:47:25--  https://storage.googleapis.com/ztm_tf_course/food_vision/10_foo
Resolving storage.googleapis.com (storage.googleapis.com)... 142.250.99.128, 142.250.107
Connecting to storage.googleapis.com (storage.googleapis.com)|142.250.99.128|:443... cor
HTTP request sent, awaiting response... 200 OK
```

```
Length: 133612354 (127M) [application/zip]
Saving to: ‘10_food_classes_1_percent.zip’

10_food_classes_1_p 100%[===================>] 127.42M   203MB/s    in 0.6s

2022-09-29 00:47:26 (203 MB/s) - ‘10_food_classes_1_percent.zip’ saved [133612354/133612
```

```
train_dir_1_percent = '10_food_classes_1_percent/train'
test_dir = '10_food_classes_1_percent/test'

walk_through_dir('10_food_classes_1_percent')
```

```
There are 2 directories and 0 images in '10_food_classes_1_percent'.
There are 10 directories and 0 images in '10_food_classes_1_percent/test'.
There are 0 directories and 250 images in '10_food_classes_1_percent/test/sushi'.
There are 0 directories and 250 images in '10_food_classes_1_percent/test/chicken_wings
There are 0 directories and 250 images in '10_food_classes_1_percent/test/steak'.
There are 0 directories and 250 images in '10_food_classes_1_percent/test/hamburger'.
There are 0 directories and 250 images in '10_food_classes_1_percent/test/chicken_curry
There are 0 directories and 250 images in '10_food_classes_1_percent/test/fried_rice'.
There are 0 directories and 250 images in '10_food_classes_1_percent/test/pizza'.
There are 0 directories and 250 images in '10_food_classes_1_percent/test/ice_cream'.
There are 0 directories and 250 images in '10_food_classes_1_percent/test/grilled_salmor
There are 0 directories and 250 images in '10_food_classes_1_percent/test/ramen'.
There are 10 directories and 0 images in '10_food_classes_1_percent/train'.
There are 0 directories and 7 images in '10_food_classes_1_percent/train/sushi'.
There are 0 directories and 7 images in '10_food_classes_1_percent/train/chicken_wings'
There are 0 directories and 7 images in '10_food_classes_1_percent/train/steak'.
There are 0 directories and 7 images in '10_food_classes_1_percent/train/hamburger'.
There are 0 directories and 7 images in '10_food_classes_1_percent/train/chicken_curry'
There are 0 directories and 7 images in '10_food_classes_1_percent/train/fried_rice'.
There are 0 directories and 7 images in '10_food_classes_1_percent/train/pizza'.
There are 0 directories and 7 images in '10_food_classes_1_percent/train/ice_cream'.
There are 0 directories and 7 images in '10_food_classes_1_percent/train/grilled_salmon
There are 0 directories and 7 images in '10_food_classes_1_percent/train/ramen'.
```

```
#process data
IMG_SIZE = (224,224)
BATCH_SIZE = 32
train_data_1_percent = tf.keras.preprocessing.image_dataset_from_directory(directory=train_di
                                                                           image_size=IMG_S
                                                                           label_mode='cate
                                                                           batch_size=BATCH
test_data = tf.keras.preprocessing.image_dataset_from_directory(directory=test_dir,
                                                                image_size=IMG_SIZE,
                                                                label_mode='categorical',
                                                                batch_size=BATCH_SIZE)
```

```
Found 70 files belonging to 10 classes.
Found 2500 files belonging to 10 classes.
```

```
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.layers.experimental import preprocessing


#augment data
data_augmentation = keras.Sequential([
    preprocessing.RandomFlip('horizontal'),
    preprocessing.RandomRotation(0.2),
    preprocessing.RandomZoom(0.2),
    preprocessing.RandomHeight(0.2),
    preprocessing.RandomWidth(0.2),
    #preprocessing.Rescale(1./255) # not needed for EfficentNet

], name='data_augmentation')


#show random example of image and augmented image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import os
import random
target_class = random.choice(train_data_1_percent.class_names)
target_dir = '10_food_classes_1_percent/train/' + target_class
random_image = random.choice(os.listdir(target_dir))
random_image_path = target_dir + '/' + random_image

img = mpimg.imread(random_image_path)
plt.imshow(img)
plt.title(f'Original random image from class: {target_class}')
plt.axis(False);

augmented_img = data_augmentation(tf.expand_dims(img, axis=0), training=True)
plt.figure()
plt.imshow(tf.squeeze(augmented_img)/255.)
```

```
<matplotlib.image.AxesImage at 0x7f1b0ddd2ad0>
```

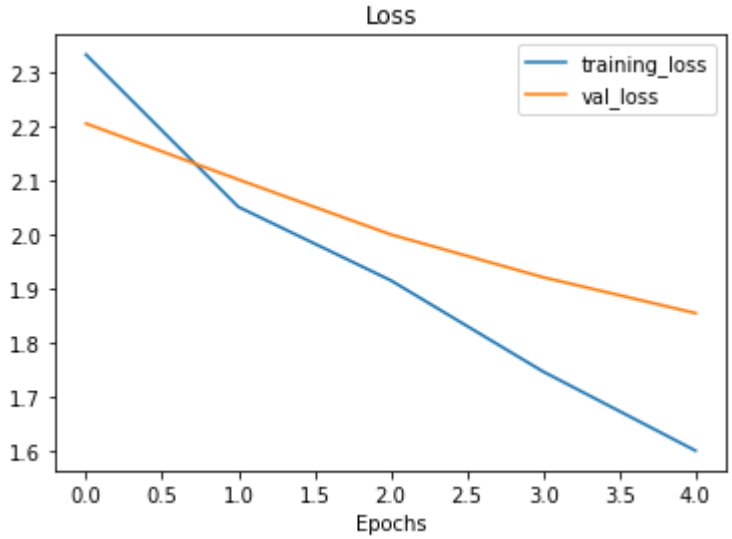
Original random image from class: grilled_salmon

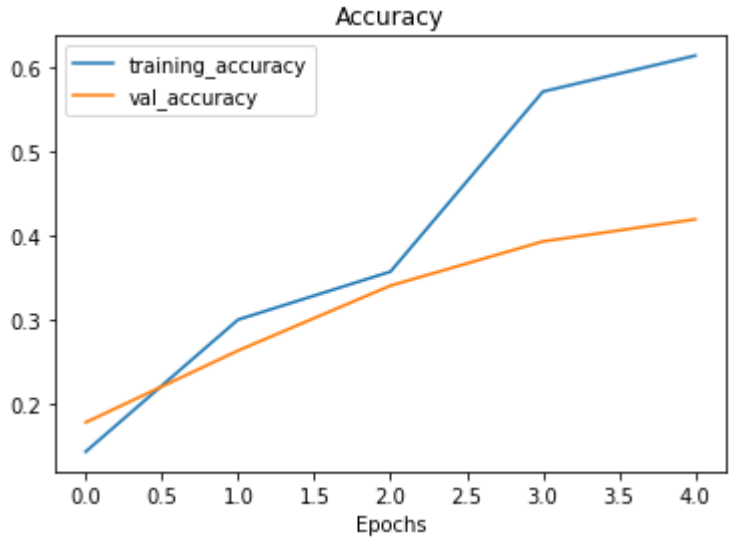
```
#create model 1 (EfficientNet B0 with 1% of data and augmentation)
input_shape = (224,224,3)
base_model = tf.keras.applications.EfficientNetB0(include_top=False)
base_model.trainable = False

inputs = layers.Input(shape=input_shape, name = 'input_layer')
x = data_augmentation(inputs)
x = base_model(x, training=False)
x = layers.GlobalAveragePooling2D(name='global_average_pooling')(x)

outputs = layers.Dense(10, activation='softmax', name='output_layer')(x)

model_1 = keras.Model(inputs, outputs)

model_1.compile(loss='categorical_crossentropy',
                optimizer=tf.keras.optimizers.Adam(),
                metrics=['accuracy'])

history_1 = model_1.fit(train_data_1_percent,
                        epochs=5,
                        steps_per_epoch=len(train_data_1_percent),
                        validation_data=test_data,
                        validation_steps=int(0.25*len(test_data)),
                        callbacks=[create_tensorboard_callback(dir_name='transfer_learning',
                                                               experiment_name='model_1')])
```

```
Saving TensorBoard log files to: transfer_learning/model_1/20220929-004731
Epoch 1/5
3/3 [==============================] - 11s 2s/step - loss: 2.3324 - accuracy: 0.1429 - v
Epoch 2/5
3/3 [==============================] - 4s 2s/step - loss: 2.0507 - accuracy: 0.3000 - va
Epoch 3/5
3/3 [==============================] - 3s 1s/step - loss: 1.9148 - accuracy: 0.3571 - va
Epoch 4/5
3/3 [==============================] - 4s 2s/step - loss: 1.7460 - accuracy: 0.5714 - va
```

```
Epoch 5/5
3/3 [==============================] - 3s 1s/step - loss: 1.5997 - accuracy: 0.6143 - va
```

```
plot_loss_curves(history_1)
```






```
model_1.evaluate(test_data)
```

```
79/79 [==============================] - 6s 73ms/step - loss: 1.8314 - accuracy: 0.4488
[1.8314275741577148, 0.4487999975681305]
```

Summary of Model 1: Promising results for 1% of training data. We will continue with more training

```
train_dir_10_percent = '10_food_classes_10_percent/train'
test_dir = '10_food_classes_10_percent/test'
```

```
IMG_SIZE=(224,224)
train_data_10_percent = tf.keras.preprocessing.image_dataset_from_directory(train_dir_10_perc
                                                                            label_mode='catego
                                                                            image_size=IMG_SIZ

test_data = tf.keras.preprocessing.image_dataset_from_directory(test_dir,
                                                                label_mode='categorical',
                                                                image_size=IMG_SIZE)
```

```
     Found 750 files belonging to 10 classes.
     Found 2500 files belonging to 10 classes.
```

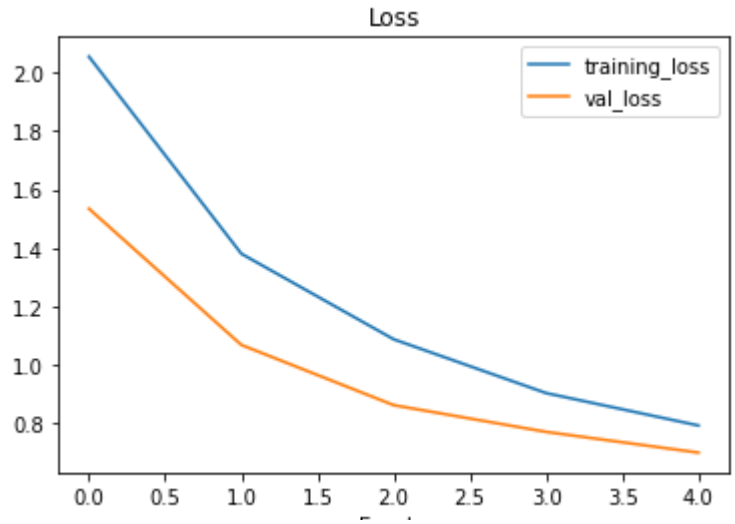
```
data_augmentation = keras.Sequential([
    preprocessing.RandomFlip('horizontal'),
    preprocessing.RandomRotation(0.2),
    preprocessing.RandomZoom(0.2),
    preprocessing.RandomHeight(0.2),
    preprocessing.RandomWidth(0.2),
    #preprocessing.Rescale(1./255) # not needed for EfficentNet

], name='data_augmentation')


#create model 2 (EfficientNet B0 training on 10% of augmented data)
input_shape = (224,224,3)
base_model = tf.keras.applications.EfficientNetB0(include_top=False)
base_model.trainable = False

inputs = layers.Input(shape=input_shape, name = 'input_layer')
x = data_augmentation(inputs)
x = base_model(x, training=False)
x = layers.GlobalAveragePooling2D(name='global_average_pooling_layer')(x)

outputs = layers.Dense(10, activation='softmax', name='output_layer')(x)

model_2 = keras.Model(inputs, outputs)

model_2.compile(loss='categorical_crossentropy',
                optimizer=tf.keras.optimizers.Adam(),
                metrics=['accuracy'])




#add checkpoints to model 2
checkpoint_path = 'ten_percent_model_checkpoints_weights/checkpoint.ckpt'

checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(filepath=checkpoint_path,
                                                         save_weights_only=True,
                                                         save_best_only=False,
```

```
                                                save_freq='epoch',
```

```
initial_epochs = 5

history_2 = model_2.fit(train_data_10_percent,
                        epochs=initial_epochs,
                        validation_data=test_data,
                        validation_steps=int(0.25*len(test_data)),
                        callbacks=[create_tensorboard_callback(dir_name='transfer_learning',
                                                               experiment_name='model_2'),
                                   checkpoint_callback])
```

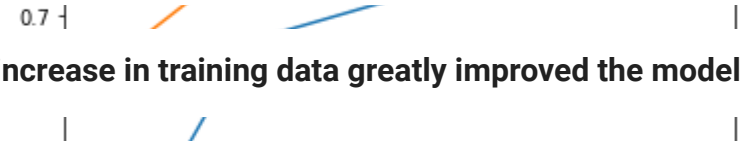
```
Saving TensorBoard log files to: transfer_learning/model_2/20220929-004811
Epoch 1/5
24/24 [==============================] - ETA: 0s - loss: 2.0549 - accuracy: 0.3093
Epoch 1: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 18s 493ms/step - loss: 2.0549 - accuracy: 0.309
Epoch 2/5
24/24 [==============================] - ETA: 0s - loss: 1.3799 - accuracy: 0.6653
Epoch 2: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 10s 415ms/step - loss: 1.3799 - accuracy: 0.665
Epoch 3/5
24/24 [==============================] - ETA: 0s - loss: 1.0867 - accuracy: 0.7240
Epoch 3: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 9s 355ms/step - loss: 1.0867 - accuracy: 0.7240
Epoch 4/5
24/24 [==============================] - ETA: 0s - loss: 0.9025 - accuracy: 0.7680
Epoch 4: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 9s 368ms/step - loss: 0.9025 - accuracy: 0.7680
Epoch 5/5
24/24 [==============================] - ETA: 0s - loss: 0.7914 - accuracy: 0.8027
Epoch 5: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 8s 325ms/step - loss: 0.7914 - accuracy: 0.8027
```

```
plot_loss_curves(history_2)
```




```
model_2.evaluate(test_data)
```

```
79/79 [==============================] - 6s 72ms/step - loss: 0.7026 - accuracy: 0.8180
[0.7025765180587769, 0.8180000185966492]
```



The increase in training data greatly improved the model


```
#confirming that model_2 layers are not trainable
for i , layer in enumerate(model_2.layers[2].layers):
  print(i, layer.name, layer.trainable)
```

```
177 block6b_expand_bn False
178 block6b_expand_activation False
179 block6b_dwconv False
180 block6b_bn False
181 block6b_activation False
182 block6b_se_squeeze False
183 block6b_se_reshape False
184 block6b_se_reduce False
185 block6b_se_expand False
186 block6b_se_excite False
187 block6b_project_conv False
188 block6b_project_bn False
189 block6b_drop False
190 block6b_add False
191 block6c_expand_conv False
192 block6c_expand_bn False
193 block6c_expand_activation False
194 block6c_dwconv False
195 block6c_bn False
196 block6c_activation False
197 block6c_se_squeeze False
198 block6c_se_reshape False
199 block6c_se_reduce False
200 block6c_se_expand False
201 block6c_se_excite False
202 block6c_project_conv False
203 block6c_project_bn False
```

```
204 block6c_drop False
205 block6c_add False
206 block6d_expand_conv False
207 block6d_expand_bn False
208 block6d_expand_activation False
209 block6d_dwconv False
210 block6d_bn False
211 block6d_activation False
212 block6d_se_squeeze False
213 block6d_se_reshape False
214 block6d_se_reduce False
215 block6d_se_expand False
216 block6d_se_excite False
217 block6d_project_conv False
218 block6d_project_bn False
219 block6d_drop False
220 block6d_add False
221 block7a_expand_conv False
222 block7a_expand_bn False
223 block7a_expand_activation False
224 block7a_dwconv False
225 block7a_bn False
226 block7a_activation False
227 block7a_se_squeeze False
228 block7a_se_reshape False
229 block7a_se_reduce False
230 block7a_se_expand False
231 block7a_se_excite False
232 block7a_project_conv False
233 block7a_project_bn False
234 top_conv False
235 top_bn False
```

```
# change base model to make the last 10 layers trainable
base_model.trainable = True

for layer in base_model.layers[:-10]:
  layer.trainable=False


#compile updated base model
base_model.compile(loss='categorical_crossentropy',
                   optimizer=tf.optimizers.Adam(learning_rate=0.0001), # when fine-tuning, lo
                   metrics=['accuracy'])


#confirm change to some trainable layers
for layer_number, layer in enumerate(model_2.layers[2].layers):
  print(layer_number,layer.name,layer.trainable)
```

```
177 block6b_expand_bn False
178 block6b_expand_activation False
179 block6b_dwconv False
180 block6b_bn False
181 block6b_activation False
182 block6b_se_squeeze False
```

```
183 block6b_se_reshape False
184 block6b_se_reduce False
185 block6b_se_expand False
186 block6b_se_excite False
187 block6b_project_conv False
188 block6b_project_bn False
189 block6b_drop False
190 block6b_add False
191 block6c_expand_conv False
192 block6c_expand_bn False
193 block6c_expand_activation False
194 block6c_dwconv False
195 block6c_bn False
196 block6c_activation False
197 block6c_se_squeeze False
198 block6c_se_reshape False
199 block6c_se_reduce False
200 block6c_se_expand False
201 block6c_se_excite False
202 block6c_project_conv False
203 block6c_project_bn False
204 block6c_drop False
205 block6c_add False
206 block6d_expand_conv False
207 block6d_expand_bn False
208 block6d_expand_activation False
209 block6d_dwconv False
210 block6d_bn False
211 block6d_activation False
212 block6d_se_squeeze False
213 block6d_se_reshape False
214 block6d_se_reduce False
215 block6d_se_expand False
216 block6d_se_excite False
217 block6d_project_conv False
218 block6d_project_bn False
219 block6d_drop False
220 block6d_add False
221 block7a_expand_conv False
222 block7a_expand_bn False
223 block7a_expand_activation False
224 block7a_dwconv False
225 block7a_bn False
226 block7a_activation False
227 block7a_se_squeeze True
228 block7a_se_reshape True
229 block7a_se_reduce True
230 block7a_se_expand True
231 block7a_se_excite True
232 block7a_project_conv True
233 block7a_project_bn True
234 top_conv True
```

```
model_3 = model_2 #model 3 is model 2 with the new base. Training will begin where model 2 le
```

```
history_3 = model_3.fit(train_data_10_percent,
                        validation_data=test_data,
                        epochs=10,
                        validation_steps=int(0.25*len(test_data)),
                        initial_epoch=history_2.epoch[-1], #start training from end of model
                        steps_per_epoch=len(train_data_10_percent),
                        callbacks=[create_tensorboard_callback(dir_name='transfer_learning',
                                                               experiment_name='model_3'),
                                   checkpoint_callback])
```

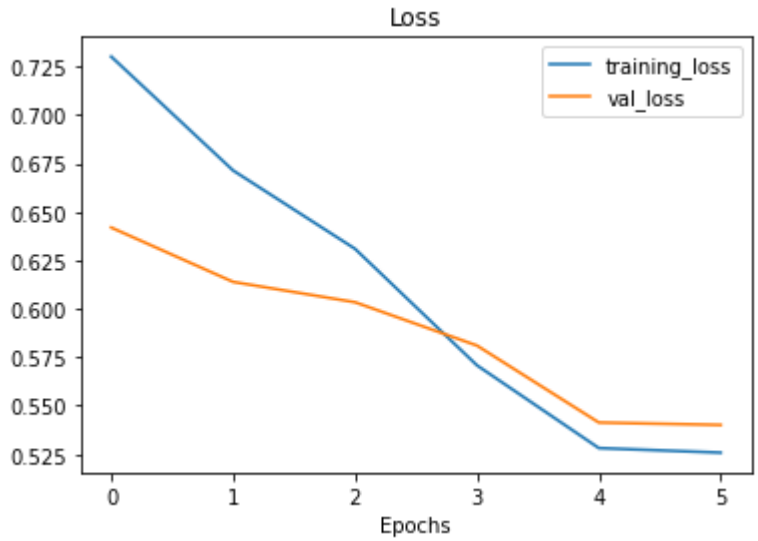
```
Saving TensorBoard log files to: transfer_learning/model_3/20220929-004929
Epoch 5/10
24/24 [==============================] - ETA: 0s - loss: 0.7301 - accuracy: 0.7987
Epoch 5: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 9s 343ms/step - loss: 0.7301 - accuracy: 0.7987
Epoch 6/10
24/24 [==============================] - ETA: 0s - loss: 0.6713 - accuracy: 0.8320
Epoch 6: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 9s 348ms/step - loss: 0.6713 - accuracy: 0.8320
Epoch 7/10
24/24 [==============================] - ETA: 0s - loss: 0.6309 - accuracy: 0.8253
Epoch 7: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 9s 350ms/step - loss: 0.6309 - accuracy: 0.8253
Epoch 8/10
24/24 [==============================] - ETA: 0s - loss: 0.5708 - accuracy: 0.8733
Epoch 8: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 9s 346ms/step - loss: 0.5708 - accuracy: 0.8733
Epoch 9/10
24/24 [==============================] - ETA: 0s - loss: 0.5281 - accuracy: 0.8720
Epoch 9: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 9s 341ms/step - loss: 0.5281 - accuracy: 0.8720
Epoch 10/10
24/24 [==============================] - ETA: 0s - loss: 0.5258 - accuracy: 0.8693
Epoch 10: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
24/24 [==============================] - 9s 353ms/step - loss: 0.5258 - accuracy: 0.8693
```

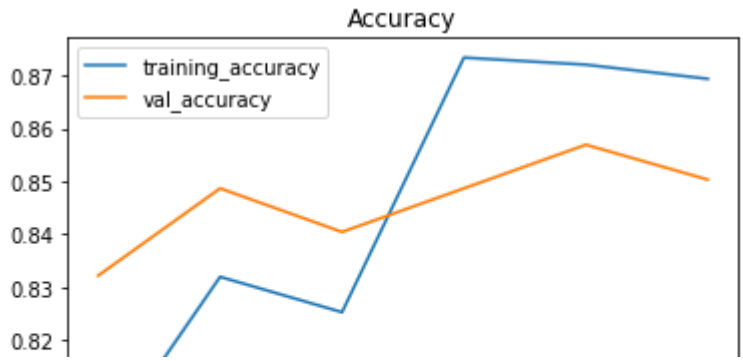
```
model_3.evaluate(test_data)
```

```
79/79 [==============================] - 6s 72ms/step - loss: 0.5525 - accuracy: 0.8364
[0.5525273680686951, 0.8363999724388123]
```

Model 3 (with more trainable layers) did improve the accuracy

```
plot_loss_curves(history_3)
```

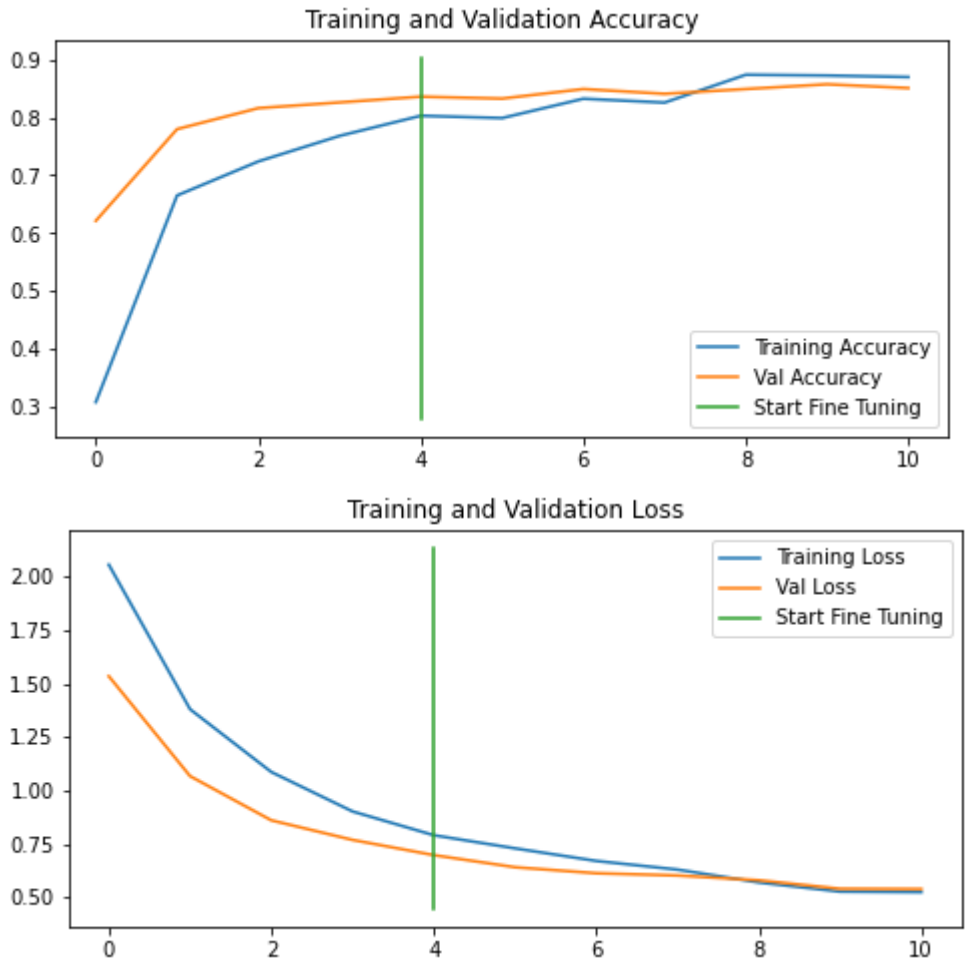





```
def compare_historys(original_history, new_history, initial_epochs=5):

  """
  Compares two model history objects.
  """

  acc = original_history.history['accuracy']
  loss = original_history.history['loss']

  val_acc = original_history.history['val_accuracy']
  val_loss = original_history.history['val_loss']

  #combine original history with new history
  total_acc = acc + new_history.history['accuracy']
  total_loss = loss + new_history.history['loss']

  total_val_acc = val_acc + new_history.history['val_accuracy']
  total_val_loss = val_loss + new_history.history['val_loss']

  #create plots
  plt.figure(figsize=(8,8))
  plt.subplot(2,1,1)
  plt.plot(total_acc, label='Training Accuracy')
  plt.plot(total_val_acc, label='Val Accuracy')
  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.figure(figsize=(8,8))
  plt.subplot(2,1,2)
  plt.plot(total_loss, label='Training Loss')
  plt.plot(total_val_loss, label='Val Loss')
  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')
```

```
compare_historys(history_2, history_3)
```



Model 3 improves on the accuracy and loss of model 2 in epochs 5-10.

```
#download all data
!wget https://storage.googleapis.com/ztm_tf_course/food_vision/10_food_classes_all_data.zip

unzip_data('10_food_classes_all_data.zip')
```

```
    --2022-09-29 00:50:38--  https://storage.googleapis.com/ztm_tf_course/food_vision/10_foo
    Resolving storage.googleapis.com (storage.googleapis.com)... 173.194.202.128, 173.194.20
```

```
Connecting to storage.googleapis.com (storage.googleapis.com)|173.194.202.128|:443... co
HTTP request sent, awaiting response... 200 OK
Length: 519183241 (495M) [application/zip]
Saving to: ‘10_food_classes_all_data.zip’

10_food_classes_all 100%[===================>] 495.13M   171MB/s    in 2.9s

2022-09-29 00:50:41 (171 MB/s) - ‘10_food_classes_all_data.zip’ saved [519183241/519183
```

```
train_dir_all = '10_food_classes_all_data/train'
test_dir = '10_food_classes_all_data/test'

walk_through_dir('10_food_classes_all_data')
```

```
There are 2 directories and 0 images in '10_food_classes_all_data'.
There are 10 directories and 0 images in '10_food_classes_all_data/test'.
There are 0 directories and 250 images in '10_food_classes_all_data/test/sushi'.
There are 0 directories and 250 images in '10_food_classes_all_data/test/chicken_wings'
There are 0 directories and 250 images in '10_food_classes_all_data/test/steak'.
There are 0 directories and 250 images in '10_food_classes_all_data/test/hamburger'.
There are 0 directories and 250 images in '10_food_classes_all_data/test/chicken_curry'
There are 0 directories and 250 images in '10_food_classes_all_data/test/fried_rice'.
There are 0 directories and 250 images in '10_food_classes_all_data/test/pizza'.
There are 0 directories and 250 images in '10_food_classes_all_data/test/ice_cream'.
There are 0 directories and 250 images in '10_food_classes_all_data/test/grilled_salmon
There are 0 directories and 250 images in '10_food_classes_all_data/test/ramen'.
There are 10 directories and 0 images in '10_food_classes_all_data/train'.
There are 0 directories and 750 images in '10_food_classes_all_data/train/sushi'.
There are 0 directories and 750 images in '10_food_classes_all_data/train/chicken_wings
There are 0 directories and 750 images in '10_food_classes_all_data/train/steak'.
There are 0 directories and 750 images in '10_food_classes_all_data/train/hamburger'.
There are 0 directories and 750 images in '10_food_classes_all_data/train/chicken_curry
There are 0 directories and 750 images in '10_food_classes_all_data/train/fried_rice'.
There are 0 directories and 750 images in '10_food_classes_all_data/train/pizza'.
There are 0 directories and 750 images in '10_food_classes_all_data/train/ice_cream'.
There are 0 directories and 750 images in '10_food_classes_all_data/train/grilled_salmor
There are 0 directories and 750 images in '10_food_classes_all_data/train/ramen'.
```

```
IMG_SIZE=(224,224)
train_data_all = tf.keras.preprocessing.image_dataset_from_directory(train_dir_all,
                                                                     label_mode='catego
                                                                     image_size=IMG_SIZ

test_data = tf.keras.preprocessing.image_dataset_from_directory(test_dir,
                                                                label_mode='categorical',
                                                                image_size=IMG_SIZE)
```

```
Found 7500 files belonging to 10 classes.
Found 2500 files belonging to 10 classes.
```

```
model_4 = model_3 #model 4 is the same as model 3 but it will train on all of the data. Again

history_4 = model_4.fit(train_data_all,
                        validation_data=test_data,
                        epochs=10,
                        validation_steps=int(0.25*len(test_data)),
                        initial_epoch=history_2.epoch[-1], #start training from end of model
                        steps_per_epoch=len(train_data_all),
                        callbacks=[create_tensorboard_callback(dir_name='transfer_learning',
                                                               experiment_name='model_4'),
                                   checkpoint_callback])
```

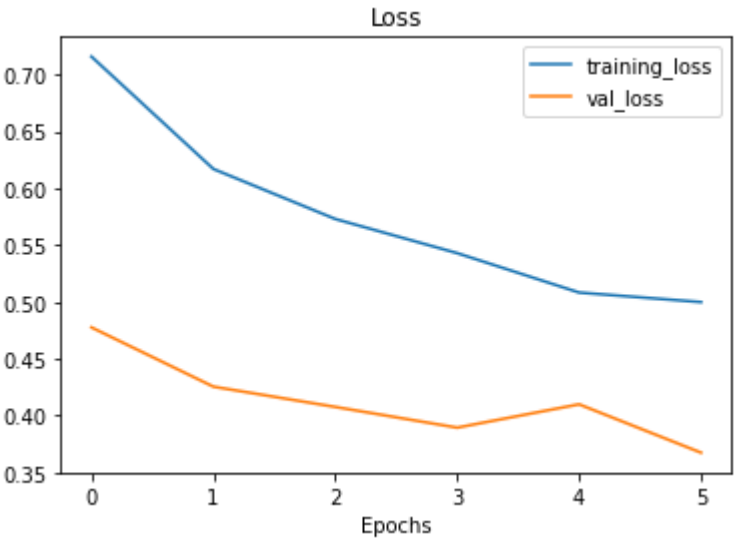
```
Saving TensorBoard log files to: transfer_learning/model_4/20220929-005047
Epoch 5/10
235/235 [==============================] - ETA: 0s - loss: 0.7159 - accuracy: 0.7771
Epoch 5: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
235/235 [==============================] - 53s 221ms/step - loss: 0.7159 - accuracy: 0.7
Epoch 6/10
235/235 [==============================] - ETA: 0s - loss: 0.6170 - accuracy: 0.8052
Epoch 6: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
235/235 [==============================] - 46s 195ms/step - loss: 0.6170 - accuracy: 0.8
Epoch 7/10
235/235 [==============================] - ETA: 0s - loss: 0.5730 - accuracy: 0.8219
Epoch 7: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
235/235 [==============================] - 42s 179ms/step - loss: 0.5730 - accuracy: 0.8
Epoch 8/10
235/235 [==============================] - ETA: 0s - loss: 0.5430 - accuracy: 0.8291
Epoch 8: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
235/235 [==============================] - 42s 176ms/step - loss: 0.5430 - accuracy: 0.8
Epoch 9/10
235/235 [==============================] - ETA: 0s - loss: 0.5083 - accuracy: 0.8361
Epoch 9: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
235/235 [==============================] - 36s 149ms/step - loss: 0.5083 - accuracy: 0.8
Epoch 10/10
235/235 [==============================] - ETA: 0s - loss: 0.4999 - accuracy: 0.8383
Epoch 10: saving model to ten_percent_model_checkpoints_weights/checkpoint.ckpt
235/235 [==============================] - 40s 168ms/step - loss: 0.4999 - accuracy: 0.8
```

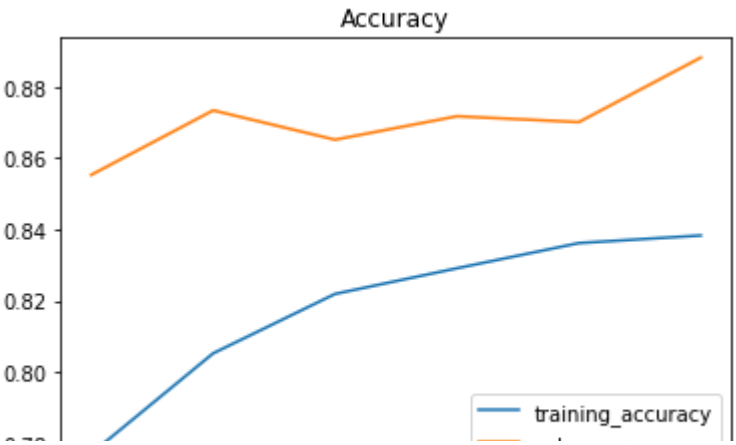
```
model_4.evaluate(test_data)
```

```
79/79 [==============================] - 6s 73ms/step - loss: 0.3597 - accuracy: 0.8884
[0.35968881845474243, 0.8884000182151794]
```

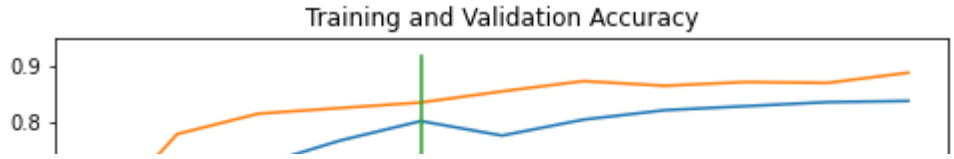
Model 4 has the highest accuracy of all of our models

```
plot_loss_curves(history_4)
```






```
compare_historys(history_2, history_4)
```




```
!tensorboard dev upload --logdir ./transfer_learning \
 --name 'Transfer Learning Experiments with 10 Food 101 Classes' \
 --description 'Transfer Learning Experiments' \
 --one_shot
```

```
***** TensorBoard Uploader *****

This will upload your TensorBoard logs to https://tensorboard.dev/ from
the following directory:

./transfer_learning

This TensorBoard will be visible to everyone. Do not upload sensitive
data.

Your use of this service is subject to Google's Terms of Service
<https://policies.google.com/terms> and Privacy Policy
<https://policies.google.com/privacy>, and TensorBoard.dev's Terms of Service
<https://tensorboard.dev/policy/terms/>.

This notice will not be shown again while you are logged into the uploader.
To log out, run `tensorboard dev auth revoke`.

Continue? (yes/NO)
```

https://tensorboard.dev/experiment/Ue2wa4MzToql0evG9qk40g/

Double-click (or enter) to edit

y

Executing (2s) Cell › system() › _system_compat() › _run_command() › _monitor_process() › _poll_process()