

```
#Beating DeepFood
!nvidia-smi -L
```

```
GPU 0: Tesla T4 (UUID: GPU-51661545-ab33-e106-2349-e28dbb924b12)
```

```
# !pip install tensorflow==2.4
import tensorflow as tf
tf.__version__
```

```
'2.4.0'
```

```
#load helper functions from github
!wget https://raw.githubusercontent.com/mrdbourke/tensorflow-deep-learning/main/extras/helper
```

```
--2022-02-04 01:29:11-- https://raw.githubusercontent.com/mrdbourke/tensorflow-deep-learning/main/extras/helper
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.1'
```

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

```
2022-02-04 01:29:11 (74.4 MB/s) - 'helper_functions.py.1' saved [10246/10246]
```



```
from helper_functions import create_tensorboard_callback,plot_loss_curves,compare_historys
```

```
import tensorflow as tf
import tensorflow_datasets as tfds
```

```
datasets_list = tfds.list_builders()
print("food101" in datasets_list)
```

```
True
```

```
#load food101 dataset
```

```
(train_data,test_data),ds_info = tfds.load(name = "food101",
                                             split = ["train","validation"],
                                             shuffle_files = True,
                                             as_supervised = True,
                                             with_info = True)
```

Downloading and preparing dataset food101/2.0.0 (download: 4.65 GiB, generated: Unknown

DI Completed...: 100% 1/1 [04:14<00:00, 183.99s/ url]

DI Size...: 100% 4764/4764 [04:14<00:00, 25.49 MiB/s]

Extraction completed...: 100% 1/1 [04:14<00:00, 254.71s/ file]

Shuffling and writing examples to /root/tensorflow_datasets/food101/2.0.0.incompleteDG51

100% 75749/75750 [00:25<00:00, 1377.48 examples/s]

Shuffling and writing examples to /root/tensorflow_datasets/food101/2.0.0.incompleteDG51

100% 25249/25250 [00:08<00:00, 3300.60 examples/s]

ds_info.features

```
FeaturesDict({
  'image': Image(shape=(None, None, 3), dtype=tf.uint8),
  'label': ClassLabel(shape=(), dtype=tf.int64, num_classes=101),
})
```

#get class names

```
class_names = ds_info.features["label"].names
class_names[:10]
```

```
['apple_pie',
 'baby_back_ribs',
 'baklava',
 'beef_carpaccio',
 'beef_tartare',
 'beet_salad',
 'beignets',
 'bibimbap',
 'bread_pudding',
 'breakfast_burrito']
```

```
take_one = train_data.take(1)
```

```
for image,label in take_one:
    print(image.shape,image.dtype,label)
```

```
(512, 512, 3) <dtype: 'uint8'> tf.Tensor(18, shape=(), dtype=int64)
```

image

```
<tf.Tensor: shape=(512, 512, 3), dtype=uint8, numpy=
array([[ [170, 193, 165],
        [175, 198, 170],
```

```

[187, 209, 186],
...,
[253, 255, 249],
[253, 255, 249],
[253, 255, 249]],

[[169, 185, 159],
 [169, 184, 161],
 [171, 189, 167],
 ...,
 [253, 255, 249],
 [253, 255, 249],
 [253, 255, 249]],

[[166, 171, 151],
 [161, 166, 146],
 [158, 165, 147],
 ...,
 [253, 255, 249],
 [253, 255, 249],
 [253, 255, 249]],

...,

[[143, 131, 105],
 [143, 131, 105],
 [141, 132, 103],
 ...,
 [152, 131, 104],
 [153, 129, 103],
 [151, 127, 101]],

[[139, 129, 102],
 [138, 128, 101],
 [137, 128, 99],
 ...,
 [150, 126, 100],
 [151, 125, 100],
 [152, 126, 99]],

[[133, 123, 96],
 [134, 124, 97],
 [136, 127, 98],
 ...,
 [149, 125, 99],
 [152, 126, 99],
 [156, 130, 103]]], dtype=uint8)>

```

```

#plot image tensor
import matplotlib.pyplot as plt
import numpy

plt.imshow(image)

```

```
plt.title(class_names[label.numpy()])
plt.axis(False)
```

```
(-0.5, 511.5, 511.5, -0.5)
```

```
chicken_curry
```



```
#TF likes batched, normalized/scaled datasets
#right now the loaded data is in tuples of (image,label) for both train and test
def preprocess_img (image,label,img_shape = 224):
    image = tf.image.resize(image, [img_shape, img_shape])
    return tf.cast(image, tf.float32), label
```

```
#preprocess a sample
```

```
preprocessed_img = preprocess_img(image, label)[0]
preprocessed_img
```

```
<tf.Tensor: shape=(224, 224, 3), dtype=float32, numpy=
array([[[170.5051 , 188.59184 , 163.11736 ],
        [187.9898 , 206.82143 , 186.91837 ],
        [208.97958 , 225.69388 , 212.7653  ],
        ...,
        [253.      , 255.      , 249.      ],
        [253.      , 255.      , 249.      ],
        [253.      , 255.      , 249.      ]],

       [[155.02551 , 149.47958 , 134.38776 ],
        [149.11224 , 149.47958 , 137.11734 ],
        [153.97449 , 162.87245 , 152.42857 ],
        ...,
        [253.      , 255.      , 249.      ],
        [253.      , 255.      , 249.      ],
        [253.      , 255.      , 249.      ]],

       [[126.99999 , 106.21939 ,  99.79082 ],
        [119.591835, 108.19388 , 102.12245 ],
        [118.66837 , 121.42857 , 113.42857 ],
        ...,
        [253.      , 255.      , 249.      ],
        [253.      , 255.      , 249.      ],
        [253.      , 255.      , 249.      ]]])
```

```
...,
[[139.71938 , 125.71938 , 98.71938 ],
 [137.28569 , 125.6429 , 98.07143 ],
 [138.19385 , 125.719406, 99.85715 ],
 ...,
 [149.47444 , 131.6887 , 112.117226],
 [144.07144 , 123.77037 , 102.785614],
 [154.05106 , 134.47958 , 110.90811 ]],

[[142.00511 , 130.00511 , 104.00512 ],
 [142.79074 , 133.6581 , 104.72443 ],
 [138.01526 , 129.01526 , 100.015274],
 ...,
 [150.39792 , 128.39792 , 107.25503 ],
 [150.3419 , 128.347 , 105.19391 ],
 [152.61731 , 130.80608 , 105.66822 ]],

[[136.67331 , 126.67331 , 99.6733 ],
 [137.04091 , 128.04091 , 99.04091 ],
 [135.41338 , 128.41338 , 99.41338 ],
 ...,
 [154.41856 , 129.69922 , 107.13792 ],
 [151.8316 , 127.8316 , 103.0254 ],
 [152.09708 , 126.097084, 99.92352 ]]], dtype=float32)>
```

```
#batch and prepare dataset(parallelize) pipeline using image_dataset_from_directory
#AUTOTUNE will utilize all CPU/GPU power
#shuffle train_data and turn into batches
#we mapped our preprocess_img func to full train_data and used AUTOTUNE to make the process f

train_data = train_data.map(map_func=preprocess_img,num_parallel_calls=tf.data.AUTOTUNE)
train_data = train_data.shuffle(buffer_size=1000).batch(batch_size = 32).prefetch(buffer_size

test_data = test_data.map(map_func=preprocess_img,num_parallel_calls=tf.data.AUTOTUNE).batch(
```

```
train_data,test_data
```

```
(<PrefetchDataset shapes: ((None, 224, 224, 3), (None,)), types: (tf.float32, tf.int64):
 <PrefetchDataset shapes: ((None, 224, 224, 3), (None,)), types: (tf.float32, tf.int64):
```



```
#create tensorboard callback
from helper_functions import create_tensorboard_callback

checkpoint_path="model_checkpoints/cp.pkt"
model_checkpoint=tf.keras.callbacks.ModelCheckpoint(checkpoint_path,
.....monitor="val_acc",
.....save_best_only=True,
.....save_weights_only=True,
```

```
.....verbose=0)
```

```
#setup mixed precision training
```

```
from tensorflow.keras import mixed_precision
mixed_precision.set_global_policy("mixed_float16")
```

```
INFO:tensorflow:Mixed precision compatibility check (mixed_float16): OK
Your GPU will likely run quickly with dtype policy mixed_float16 as it has compute capat
INFO:tensorflow:Mixed precision compatibility check (mixed_float16): OK
Your GPU will likely run quickly with dtype policy mixed_float16 as it has compute capat
```

```
#build feature extraction model
```

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

```
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 = base_model(inputs,training=False)
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(len(class_names))(x)
outputs = layers.Activation("softmax",dtype = tf.float32,name = "softmax_float32")(x) # tur
model = tf.keras.Model(inputs,outputs)
```

```
model.compile(loss="sparse_categorical_crossentropy",
              optimizer = tf.keras.optimizers.Adam(),
              metrics = ["accuracy"]) #since labels are in int we use sparse , if it were on
```

```
model.summary()
```

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0\_16711680/16705208 [=====] - 0s 0us/step
Model: "model"
```

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	[(None, 224, 224, 3)]	0
efficientnetb0 (Functional)	(None, None, None, 1280)	4049571
global_average_pooling2d (Gl	(None, 1280)	0
dense (Dense)	(None, 101)	129381
softmax_float32 (Activation)	(None, 101)	0
Total params: 4,178,952		
Trainable params: 129,381		

Non-trainable params: 4,049,571

```
for layer in model.layers:
    print(layer.name, layer.trainable, layer.dtype, layer.dtype_policy)

input_layer True float32 <Policy "float32">
efficientnetb0 False float32 <Policy "mixed_float16">
global_average_pooling2d True float32 <Policy "mixed_float16">
dense True float32 <Policy "mixed_float16">
softmax_float32 True float32 <Policy "float32">
```

```
for layer in model.layers[1].layers:
    print(layer.name, layer.trainable, layer.dtype, layer.dtype_policy)

block6b_expand_activation False float32 <Policy "mixed_float16">
block6b_dwconv False float32 <Policy "mixed_float16">
block6b_bn False float32 <Policy "mixed_float16">
block6b_activation False float32 <Policy "mixed_float16">
block6b_se_squeeze False float32 <Policy "mixed_float16">
block6b_se_reshape False float32 <Policy "mixed_float16">
block6b_se_reduce False float32 <Policy "mixed_float16">
block6b_se_expand False float32 <Policy "mixed_float16">
block6b_se_excite False float32 <Policy "mixed_float16">
block6b_project_conv False float32 <Policy "mixed_float16">
block6b_project_bn False float32 <Policy "mixed_float16">
block6b_drop False float32 <Policy "mixed_float16">
block6b_add False float32 <Policy "mixed_float16">
block6c_expand_conv False float32 <Policy "mixed_float16">
block6c_expand_bn False float32 <Policy "mixed_float16">
block6c_expand_activation False float32 <Policy "mixed_float16">
block6c_dwconv False float32 <Policy "mixed_float16">
block6c_bn False float32 <Policy "mixed_float16">
block6c_activation False float32 <Policy "mixed_float16">
block6c_se_squeeze False float32 <Policy "mixed_float16">
block6c_se_reshape False float32 <Policy "mixed_float16">
block6c_se_reduce False float32 <Policy "mixed_float16">
block6c_se_expand False float32 <Policy "mixed_float16">
block6c_se_excite False float32 <Policy "mixed_float16">
block6c_project_conv False float32 <Policy "mixed_float16">
block6c_project_bn False float32 <Policy "mixed_float16">
block6c_drop False float32 <Policy "mixed_float16">
block6c_add False float32 <Policy "mixed_float16">
block6d_expand_conv False float32 <Policy "mixed_float16">
block6d_expand_bn False float32 <Policy "mixed_float16">
block6d_expand_activation False float32 <Policy "mixed_float16">
block6d_dwconv False float32 <Policy "mixed_float16">
block6d_bn False float32 <Policy "mixed_float16">
block6d_activation False float32 <Policy "mixed_float16">
block6d_se_squeeze False float32 <Policy "mixed_float16">
block6d_se_reshape False float32 <Policy "mixed_float16">
block6d_se_reduce False float32 <Policy "mixed_float16">
block6d_se_expand False float32 <Policy "mixed_float16">
```

```

block6d_se_excite False float32 <Policy "mixed_float16">
block6d_project_conv False float32 <Policy "mixed_float16">
block6d_project_bn False float32 <Policy "mixed_float16">
block6d_drop False float32 <Policy "mixed_float16">
block6d_add False float32 <Policy "mixed_float16">
block7a_expand_conv False float32 <Policy "mixed_float16">
block7a_expand_bn False float32 <Policy "mixed_float16">
block7a_expand_activation False float32 <Policy "mixed_float16">
block7a_dwconv False float32 <Policy "mixed_float16">
block7a_bn False float32 <Policy "mixed_float16">
block7a_activation False float32 <Policy "mixed_float16">
block7a_se_squeeze False float32 <Policy "mixed_float16">
block7a_se_reshape False float32 <Policy "mixed_float16">
block7a_se_reduce False float32 <Policy "mixed_float16">
block7a_se_expand False float32 <Policy "mixed_float16">
block7a_se_excite False float32 <Policy "mixed_float16">
block7a_project_conv False float32 <Policy "mixed_float16">
block7a_project_bn False float32 <Policy "mixed_float16">
top_conv False float32 <Policy "mixed_float16">
top_bn False float32 <Policy "mixed_float16">

```

```

history_101_food_classes_features_extract = model.fit(train_data,
                                                        epochs = 3,
                                                        steps_per_epoch = len(train_data),
                                                        validation_data = test_data,
                                                        validation_steps = int(0.15 * len(test_data)),
                                                        callbacks = [create_tensorboard_callback,
                                                                    model_checkpoint])

```

Saving TensorBoard log files to: training_logs/efficientnetB0_101_food_classes/20220204-

Epoch 1/3
 2368/2368 [=====] - 187s 72ms/step - loss: 2.3229 - accuracy: 0.60
 WARNING:tensorflow:Can save best model only with val_acc available, skipping.
 WARNING:tensorflow:Can save best model only with val_acc available, skipping.

Epoch 2/3
 2368/2368 [=====] - 169s 70ms/step - loss: 1.3081 - accuracy: 0.70
 WARNING:tensorflow:Can save best model only with val_acc available, skipping.
 WARNING:tensorflow:Can save best model only with val_acc available, skipping.

Epoch 3/3
 2368/2368 [=====] - 176s 73ms/step - loss: 1.1483 - accuracy: 0.74
 WARNING:tensorflow:Can save best model only with val_acc available, skipping.
 WARNING:tensorflow:Can save best model only with val_acc available, skipping.

```
model.evaluate(test_data)
```

790/790 [=====] - 46s 59ms/step - loss: 1.0915 - accuracy: 0.76
 [1.091526746749878, 0.7068910598754883]

TO DOs


```
#save model to a file
model.save("drive/MyDrive/tensorflow_saved_models/101_food_milestone_model")
```

```
INFO:tensorflow:Assets written to: drive/MyDrive/tensorflow_saved_models/101_food_milestone_model
INFO:tensorflow:Assets written to: drive/MyDrive/tensorflow_saved_models/101_food_milestone_model
```

```
#load model
loaded_model = tf.keras.models.load_model("drive/MyDrive/tensorflow_saved_models/101_food_milestone_model")
```

```
#evaluate loaded model
loaded_loss,loaded_accuracy = loaded_model.evaluate(test_data)
loaded_loss, loaded_accuracy
```

```
790/790 [=====] - 50s 63ms/step - loss: 1.0915 - accuracy: 0.7068910598754883
```

```
for layer in loaded_model.layers[1].layers:
    print(layer.trainable,layer.name,layer.dtype_policy)

True block5c_se_reshape <Policy "mixed_float16">
True block5c_se_reduce <Policy "mixed_float16">
True block5c_se_expand <Policy "mixed_float16">
True block5c_se_excite <Policy "mixed_float16">
True block5c_project_conv <Policy "mixed_float16">
True block5c_project_bn <Policy "mixed_float16">
True block5c_drop <Policy "mixed_float16">
True block5c_add <Policy "mixed_float16">
True block6a_expand_conv <Policy "mixed_float16">
True block6a_expand_bn <Policy "mixed_float16">
True block6a_expand_activation <Policy "mixed_float16">
True block6a_dwconv_pad <Policy "mixed_float16">
True block6a_dwconv <Policy "mixed_float16">
True block6a_bn <Policy "mixed_float16">
True block6a_activation <Policy "mixed_float16">
True block6a_se_squeeze <Policy "mixed_float16">
True block6a_se_reshape <Policy "mixed_float16">
True block6a_se_reduce <Policy "mixed_float16">
True block6a_se_expand <Policy "mixed_float16">
True block6a_se_excite <Policy "mixed_float16">
True block6a_project_conv <Policy "mixed_float16">
True block6a_project_bn <Policy "mixed_float16">
True block6b_expand_conv <Policy "mixed_float16">
True block6b_expand_bn <Policy "mixed_float16">
True block6b_expand_activation <Policy "mixed_float16">
True block6b_dwconv <Policy "mixed_float16">
True block6b_bn <Policy "mixed_float16">
True block6b_activation <Policy "mixed_float16">
True block6b_se_squeeze <Policy "mixed_float16">
True block6b_se_reshape <Policy "mixed_float16">
True block6b_se_reduce <Policy "mixed_float16">
True block6b_se_expand <Policy "mixed_float16">
True block6b_se_excite <Policy "mixed_float16">
```

```

True block6b_se_excite <Policy "mixed_float16">
True block6b_project_conv <Policy "mixed_float16">
True block6b_project_bn <Policy "mixed_float16">
True block6b_drop <Policy "mixed_float16">
True block6b_add <Policy "mixed_float16">
True block6c_expand_conv <Policy "mixed_float16">
True block6c_expand_bn <Policy "mixed_float16">
True block6c_expand_activation <Policy "mixed_float16">
True block6c_dwconv <Policy "mixed_float16">
True block6c_bn <Policy "mixed_float16">
True block6c_activation <Policy "mixed_float16">
True block6c_se_squeeze <Policy "mixed_float16">
True block6c_se_reshape <Policy "mixed_float16">

True block6c_se_reduce <Policy "mixed_float16">
True block6c_se_expand <Policy "mixed_float16">
True block6c_se_excite <Policy "mixed_float16">
True block6c_project_conv <Policy "mixed_float16">
True block6c_project_bn <Policy "mixed_float16">
True block6c_drop <Policy "mixed_float16">
True block6c_add <Policy "mixed_float16">
True block6d_expand_conv <Policy "mixed_float16">
True block6d_expand_bn <Policy "mixed_float16">
True block6d_expand_activation <Policy "mixed_float16">
True block6d_dwconv <Policy "mixed_float16">
True block6d_bn <Policy "mixed_float16">
True block6d_activation <Policy "mixed_float16">
True block6d_se_squeeze <Policy "mixed_float16">

```

#download pretrained model

```
!wget https://storage.googleapis.com/ztn_tf_course/food_vision/07_efficientnetb0_feature_extr
```

```

--2022-02-04 02:11:19--  https://storage.googleapis.com/ztn_tf_course/food_vision/07_efficientnetb0_feature_extract_model_mixed_precision.zip
Resolving storage.googleapis.com (storage.googleapis.com)... 142.251.107.128, 173.194.211.128
Connecting to storage.googleapis.com (storage.googleapis.com)|142.251.107.128|:443... cc
HTTP request sent, awaiting response... 200 OK
Length: 16976857 (16M) [application/zip]
Saving to: '07_efficientnetb0_feature_extract_model_mixed_precision.zip'

```

```
07_efficientnetb0_f 100%[=====] 16.19M 63.1MB/s in 0.3s
```

```
2022-02-04 02:11:20 (63.1 MB/s) - '07_efficientnetb0_feature_extract_model_mixed_precision.zip'
```

```
!mkdir downloaded_gs_model
```

```
!unzip 07_efficientnetb0_feature_extract_model_mixed_precision.zip -d downloaded_gs_model
```

```

Archive: 07_efficientnetb0_feature_extract_model_mixed_precision.zip
  creating: downloaded_gs_model/07_efficientnetb0_feature_extract_model_mixed_precision/
  creating: downloaded_gs_model/07_efficientnetb0_feature_extract_model_mixed_precision/
 inflating: downloaded_gs_model/07_efficientnetb0_feature_extract_model_mixed_precision/
 inflating: downloaded_gs_model/07_efficientnetb0_feature_extract_model_mixed_precision/

```

```

inflating: downloaded_gs_model/07_efficientnetb0_feature_extract_model_mixed_precision_1.0.h5
creating: downloaded_gs_model/07_efficientnetb0_feature_extract_model_mixed_precision_1.0.h5

```

```
downloaded_model = tf.keras.models.load_model("/content/downloaded_gs_model/07_efficientnetb0_
```

```
downloaded_model.summary()
```

Model: "model"

Layer (type)	Output Shape	Param #
=====		
input_layer (InputLayer)	[(None, 224, 224, 3)]	0

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

pooling_layer (GlobalAverage)	(None, 1280)	0

dense (Dense)	(None, 101)	129381

softmax_float32 (Activation)	(None, 101)	0
=====		
Total params: 4,178,952		
Trainable params: 4,136,929		
Non-trainable params: 42,023		

```
downloaded_model.evaluate(test_data)
```

```

790/790 [=====] - 51s 61ms/step - loss: 1.0881 - accuracy: 0.7064950466156006
[1.0881279706954956, 0.7064950466156006]

```

```
base_model.trainable = True
```

```

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

```

```

downloaded_model.compile(loss = "sparse_categorical_crossentropy",
                        optimizer = tf.keras.optimizers.Adam(0.0001),
                        metrics = ["accuracy"])

```

```
from tensorflow.keras.callbacks import EarlyStopping
```

```
early_stopping = EarlyStopping(monitor = 'val_loss',patience=3)
```

```

checkpoint_path = "fine_tuning_model_checkpoints/cp.pkt"
model_checkpoint = tf.keras.callbacks.ModelCheckpoint(checkpoint_path,
                                                    monitor = "val_acc",

```

```
save_best_only = True,
save_weights_only = True,
verbose = 0)
```

```
for layer in downloaded_model.layers[1].layers:
    print(layer.trainable, layer.name, layer.dtype_policy)
```

```
fine_tune_epochs = 5+95
```

```
history_fine_tuned_downloaded_model = downloaded_model.fit(train_data,
                                                            epochs = fine_tune_epochs,
                                                            steps_per_epoch = len(train_data),
                                                            validation_data = test_data,
                                                            validation_steps = int(0.15 * len(
initial_epoch=history_101_food_cla
callbacks = [create_tensorboard_ca

model_checkpoint, ear
```

Saving TensorBoard log files to: training_logs/efficientnetB0_101_food_classes/20220204-
Epoch 3/100

2368/2368 [=====] - 331s 130ms/step - loss: 0.9833 - accuracy:

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

Epoch 4/100

2368/2368 [=====] - 297s 125ms/step - loss: 0.5854 - accuracy:

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

Epoch 5/100

2368/2368 [=====] - 295s 124ms/step - loss: 0.3299 - accuracy:

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

Epoch 6/100

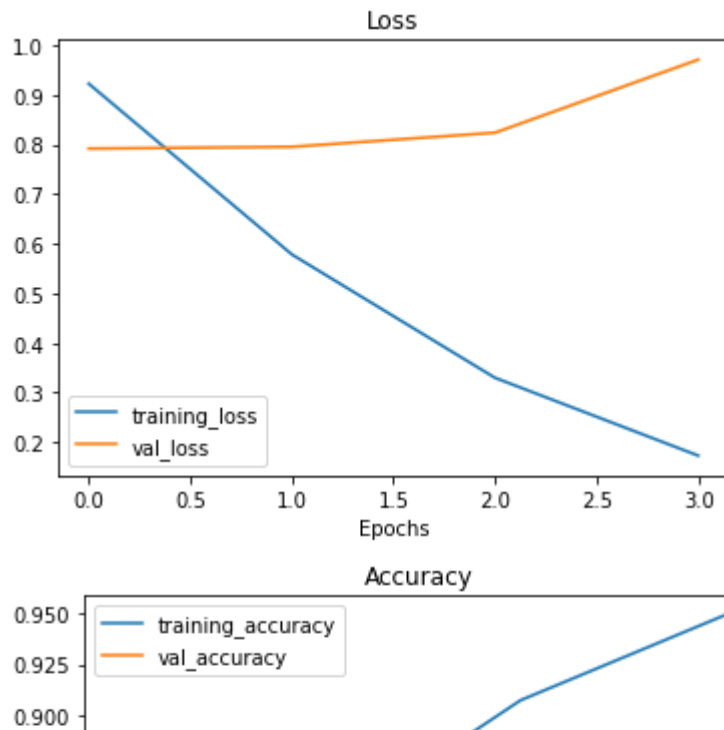
2368/2368 [=====] - 299s 126ms/step - loss: 0.1679 - accuracy:

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

WARNING:tensorflow:Can save best model only with val_acc available, skipping.

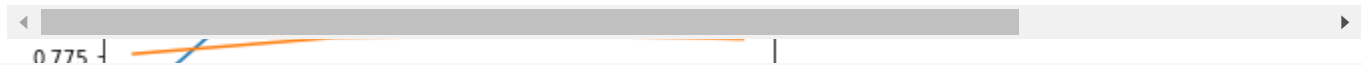


```
plot_loss_curves(history_fine_tuned_downloaded_model)
```



```
downloaded_model.save("drive/MyDrive/tensorflow_saved_models/101_food_milestone_model+all_lay
```

```
INFO:tensorflow:Assets written to: drive/MyDrive/tensorflow_saved_models/101_food_milest
INFO:tensorflow:Assets written to: drive/MyDrive/tensorflow_saved_models/101_food_milest
```



```
downloaded_model.evaluate(test_data)
```

```
790/790 [=====] - 51s 64ms/step - loss: 0.9969 - accuracy: 0.77
[0.9968550205230713, 0.7767128944396973]
```



```
!tensorboard dev upload --logdir ./training_logs \
--name "Transfer learning experiments" \
--description "A series of different transfer learning experiments with varying amounts of
--one_shot # exits the uploader when upload has finished
```

```
2022-02-04 03:22:50.338829: I tensorflow/stream_executor/platform/default/dso_loader.cc
```

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

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

```
./training_logs
```

```
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
```

