

<https://www.kaggle.com/datasets/bongsang/eye-disease-deep-learning-dataset>

<https://www.kaggle.com/code/andreamontemurro/grapevine-disease-classification/notebook>

https://keras.io/examples/vision/image_classification_from_scratch/

```
In [ ]: import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
```

2023-06-17 18:57:12.755529: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

2023-06-17 18:57:13.258055: W tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open shared object file: No such file or directory

2023-06-17 18:57:13.258076: I tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.

2023-06-17 18:57:14.896441: W tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot open shared object file: No such file or directory

2023-06-17 18:57:14.896861: W tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror: libnvinfer_plugin.so.7: cannot open shared object file: No such file or directory

2023-06-17 18:57:14.896873: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with TensorRT, please make sure the missing libraries mentioned above are installed properly.

```
In [ ]: ls Grape
```

```
Grape/
'Grape__Leaf_blight_(Isariopsis_Leaf_Spot)'/
Grape__Black_rot/
'Grape__Esca_(Black_Measles)'/
```

```
In [ ]: import os
```

```
num_skipped = 0
for folder_name in ("Grape__Black_rot", "Grape__Esca_(Black_Measles)", "Grape__healthy"):
    folder_path = os.path.join("Grape", folder_name)
    for fname in os.listdir(folder_path):
        fpath = os.path.join(folder_path, fname)
        try:
            fobj = open(fpath, "rb")
            is_jfif = tf.compat.as_bytes("JFIF") in fobj.peek(10)
        finally:
            fobj.close()
```

```

if not is_jfif:
    num_skipped += 1
    # Delete corrupted image
    os.remove(fpath)

print("Deleted %d images" % num_skipped)

```

Deleted 0 images

```

In [ ]: image_size = (180, 180)
batch_size = 128

train_ds, val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    "Grape",
    validation_split=0.2,
    subset="both",
    seed=1337,
    image_size=image_size,
    batch_size=batch_size,
)

```

Found 5164 files belonging to 5 classes.

Using 4132 files for training.

Using 1032 files for validation.

```

2023-06-17 18:57:20.291713: W tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2023-06-17 18:57:20.292294: W tensorflow/compiler/xla/stream_executor/cuda/cuda_driver.cc:265] failed call to cuInit: UNKNOWN ERROR (303)
2023-06-17 18:57:20.292420: I tensorflow/compiler/xla/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (LEOSER 4): /proc/driver/nvidia/version does not exist
2023-06-17 18:57:20.294638: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

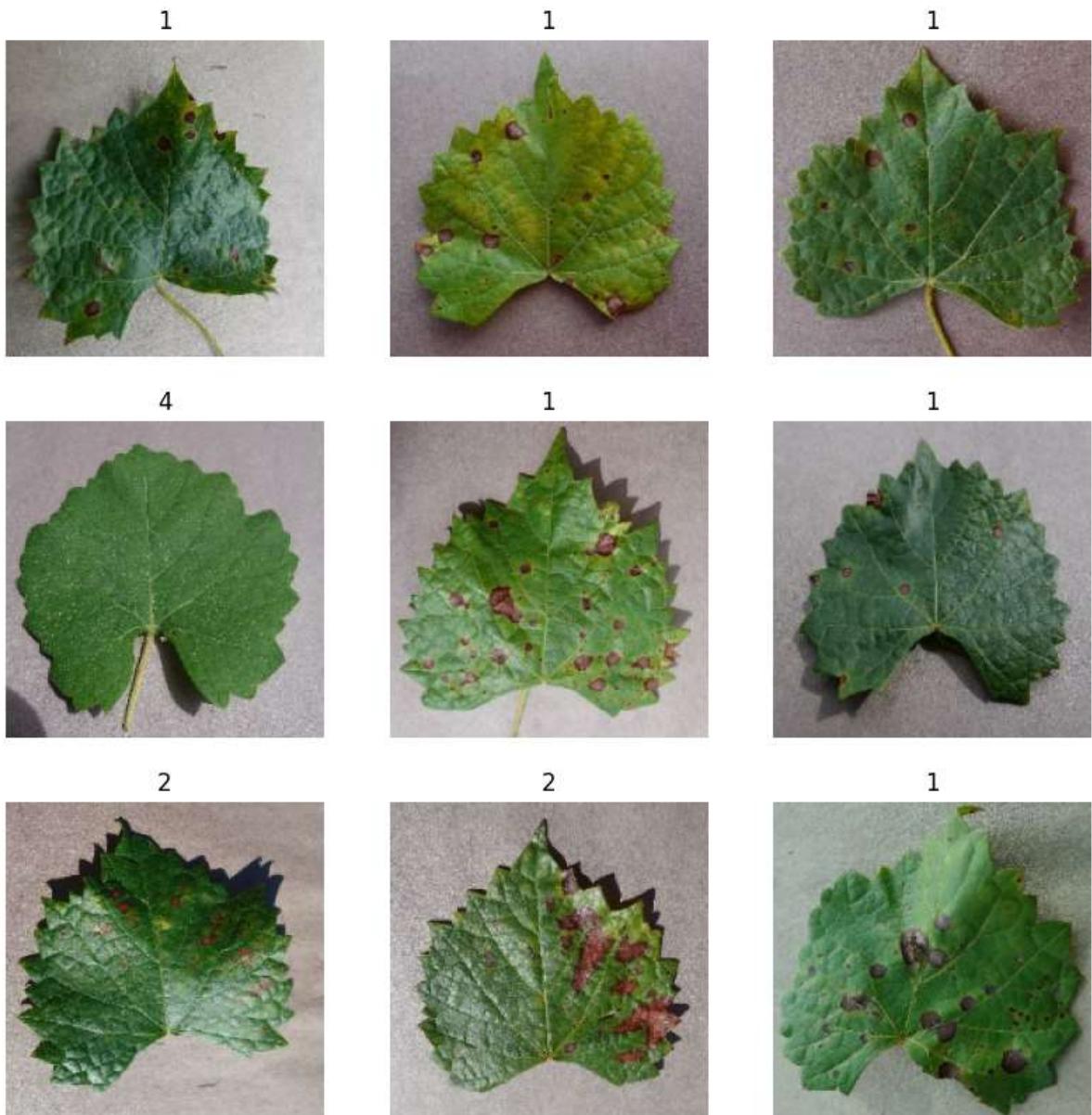
```

```

In [ ]: 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.title(int(labels[i])))
        plt.axis("off")

```



```
In [ ]: data_augmentation = keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
        layers.RandomRotation(0.1),
    ]
)
```

```
In [ ]: plt.figure(figsize=(10, 10))
for images, _ in train_ds.take(1):
    for i in range(9):
        augmented_images = data_augmentation(images)
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(augmented_images[0].numpy().astype("uint8"))
        plt.axis("off")
```


WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:5 out of the last 5 calls to <function pfor.<locals>.f at 0x7fe9286fae60> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:6 out of the last 6 calls to <function pfor.<locals>.f at 0x7fe928761630> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.function has reduce_retracing=True option that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/guide/function#controlling_retracing and https://www.tensorflow.org/api_docs/python/tf/function for more details.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.



```
In [ ]: augmented_train_ds = train_ds.map(  
        lambda x, y: (data_augmentation(x, training=True), y))
```

WARNING:tensorflow:From /home/trabalhof/mlpytorch/lib/python3.10/site-packages/tensorflow/python/autograph/pyct/static_analysis/liveness.py:83: Analyzer.lambda_check (from tensorflow.python.autograph.pyct.static_analysis.liveness) is deprecated and will be removed after 2023-09-23.

Instructions for updating:

Lambda fuctions will be no more assumed to be used in the statement where they are used, or at least in the same block. <https://github.com/tensorflow/tensorflow/issues/56089>

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

```
In [ ]: # Apply `data_augmentation` to the training images.
train_ds = train_ds.map(
    lambda img, label: (data_augmentation(img), label),
    num_parallel_calls=tf.data.AUTOTUNE,
)
# Prefetching samples in GPU memory helps maximize GPU utilization.
train_ds = train_ds.prefetch(tf.data.AUTOTUNE)
val_ds = val_ds.prefetch(tf.data.AUTOTUNE)
```

WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.

WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.

```
In [ ]: def make_model(input_shape, num_classes):
    inputs = keras.Input(shape=input_shape)

    # Entry block
    x = layers.Rescaling(1.0 / 255)(inputs)
    x = layers.Conv2D(128, 3, strides=2, padding="same")(x)
    x = layers.BatchNormalization()(x)
```

```
x = layers.Activation("relu")(x)

previous_block_activation = x # Set aside residual

for size in [256, 512, 728]:
    x = layers.Activation("relu")(x)
    x = layers.SeparableConv2D(size, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)

    x = layers.Activation("relu")(x)
    x = layers.SeparableConv2D(size, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)

    x = layers.MaxPooling2D(3, strides=2, padding="same")(x)

    # Project residual
    residual = layers.Conv2D(size, 1, strides=2, padding="same")(
        previous_block_activation
    )
    x = layers.add([x, residual]) # Add back residual
    previous_block_activation = x # Set aside next residual

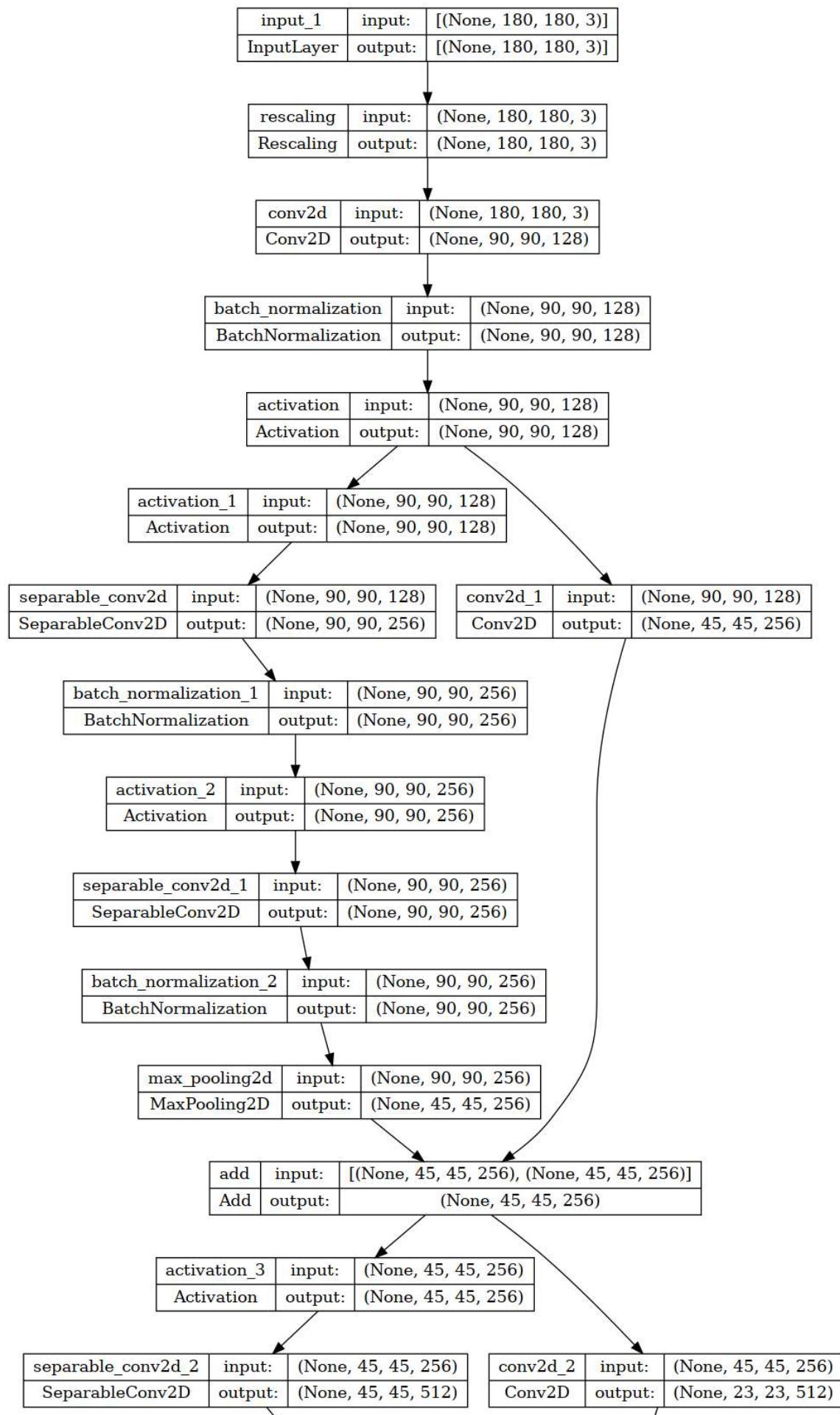
x = layers.SeparableConv2D(1024, 3, padding="same")(x)
x = layers.BatchNormalization()(x)
x = layers.Activation("relu")(x)

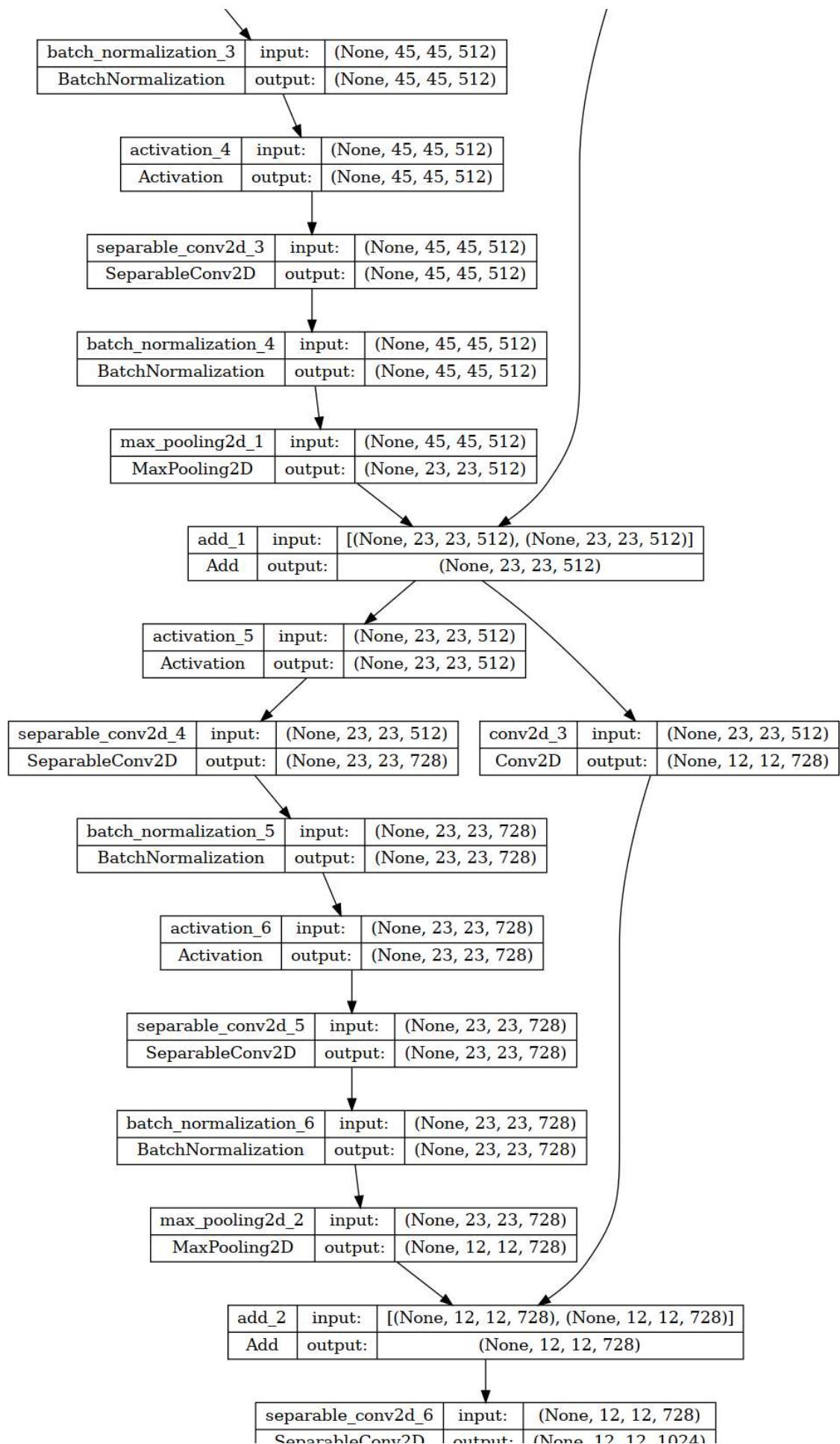
x = layers.GlobalAveragePooling2D()(x)
if num_classes == 2:
    activation = "sigmoid"
    units = 1
else:
    activation = "softmax"
    units = num_classes

x = layers.Dropout(0.5)(x)
outputs = layers.Dense(units, activation=activation)(x)
return keras.Model(inputs, outputs)

model = make_model(input_shape=image_size + (3,), num_classes=2)
keras.utils.plot_model(model, show_shapes=True)
```

Out[]:





```
[ SeparableConv2D | Output: (None, 14, 14, 1024) ]
```

```
In [ ]: epochs = 25

callbacks = [
    keras.callbacks.ModelCheckpoint("save_at_{epoch}.keras"),
]
model.compile(
    optimizer=keras.optimizers.Adam(1e-3),
    loss="binary_crossentropy",
    metrics=["accuracy"],
)
model.fit(
    train_ds,
    epochs=epochs,
    callbacks=callbacks,
    validation_data=val_ds,
)
```

dense	input:	(None, 1024)
Dense	output:	(None, 1)

```
Epoch 1/25
33/33 [=====] - 599s 18s/step - loss: -16.7049 - accuracy: 0.1718 - val_loss: -1.5713 - val_accuracy: 0.2946
Epoch 2/25
33/33 [=====] - 577s 17s/step - loss: -51.4359 - accuracy: 0.2800 - val_loss: -3.5793 - val_accuracy: 0.2946
Epoch 3/25
33/33 [=====] - 564s 17s/step - loss: -92.0042 - accuracy: 0.2899 - val_loss: -5.1098 - val_accuracy: 0.2946
Epoch 4/25
33/33 [=====] - 532s 16s/step - loss: -147.7040 - accuracy: 0.2916 - val_loss: -6.6079 - val_accuracy: 0.2946
Epoch 5/25
33/33 [=====] - 549s 16s/step - loss: -221.5514 - accuracy: 0.2916 - val_loss: -8.8180 - val_accuracy: 0.2946
Epoch 6/25
33/33 [=====] - 549s 16s/step - loss: -305.4582 - accuracy: 0.2916 - val_loss: -17.9489 - val_accuracy: 0.2946
Epoch 7/25
33/33 [=====] - 550s 17s/step - loss: -404.7810 - accuracy: 0.2916 - val_loss: -32.5065 - val_accuracy: 0.2946
Epoch 8/25
33/33 [=====] - 554s 17s/step - loss: -510.1067 - accuracy: 0.2916 - val_loss: -71.3656 - val_accuracy: 0.2946
Epoch 9/25
33/33 [=====] - 547s 16s/step - loss: -630.2923 - accuracy: 0.2916 - val_loss: -149.6425 - val_accuracy: 0.2946
Epoch 10/25
33/33 [=====] - 517s 16s/step - loss: -756.0745 - accuracy: 0.2916 - val_loss: -210.7875 - val_accuracy: 0.2946
Epoch 11/25
33/33 [=====] - 516s 15s/step - loss: -884.2916 - accuracy: 0.2916 - val_loss: -403.6058 - val_accuracy: 0.2946
Epoch 12/25
33/33 [=====] - 513s 15s/step - loss: -1033.2500 - accuracy: 0.2916 - val_loss: -584.5427 - val_accuracy: 0.2946
Epoch 13/25
33/33 [=====] - 514s 15s/step - loss: -1191.0635 - accuracy: 0.2916 - val_loss: -602.5728 - val_accuracy: 0.2946
Epoch 14/25
33/33 [=====] - 514s 15s/step - loss: -1349.8439 - accuracy: 0.2916 - val_loss: -1013.5120 - val_accuracy: 0.2946
Epoch 15/25
33/33 [=====] - 515s 15s/step - loss: -1530.5654 - accuracy: 0.2916 - val_loss: -960.7450 - val_accuracy: 0.2946
Epoch 16/25
33/33 [=====] - 511s 15s/step - loss: -1717.6173 - accuracy: 0.2916 - val_loss: -1745.5770 - val_accuracy: 0.2946
Epoch 17/25
33/33 [=====] - 530s 16s/step - loss: -1917.0107 - accuracy: 0.2916 - val_loss: -1816.2173 - val_accuracy: 0.2946
Epoch 18/25
33/33 [=====] - 511s 15s/step - loss: -2119.7461 - accuracy: 0.2916 - val_loss: -2423.8699 - val_accuracy: 0.2946
Epoch 19/25
33/33 [=====] - 508s 15s/step - loss: -2325.9153 - accuracy:
```

```

cy: 0.2916 - val_loss: -3481.8213 - val_accuracy: 0.2946
Epoch 20/25
33/33 [=====] - 518s 16s/step - loss: -2565.9158 - accuracy: 0.2916 - val_loss: -3346.4502 - val_accuracy: 0.2946
Epoch 21/25
33/33 [=====] - 555s 17s/step - loss: -2791.9211 - accuracy: 0.2916 - val_loss: -3880.9907 - val_accuracy: 0.2946
Epoch 22/25
33/33 [=====] - 576s 17s/step - loss: -3063.1543 - accuracy: 0.2916 - val_loss: -3826.8711 - val_accuracy: 0.2946
Epoch 23/25
33/33 [=====] - 554s 17s/step - loss: -3305.9336 - accuracy: 0.2916 - val_loss: -4092.1609 - val_accuracy: 0.2946
Epoch 24/25
33/33 [=====] - 547s 16s/step - loss: -3587.7175 - accuracy: 0.2916 - val_loss: -3846.2183 - val_accuracy: 0.2946
Epoch 25/25
33/33 [=====] - 551s 16s/step - loss: -3881.2117 - accuracy: 0.2916 - val_loss: -4630.5454 - val_accuracy: 0.2946

```

Out[]: <keras.callbacks.History at 0x7fe929577d30>

```
In [ ]: img = keras.preprocessing.image.load_img(
    "/home/trabalhof/Grape/Grape__Black_rot/14.jpg", target_size=image_size
)
img_array = keras.preprocessing.image.img_to_array(img)
img_array = tf.expand_dims(img_array, 0) # Create batch axis
```

```
In [ ]: predictions = model.predict(img_array)
score = float(predictions[0])
print(f"This image is {100 * (1 - score):.2f}% Grape__Black_rot , {100 * score:.2f}% healthy")
1/1 [=====] - 2s 2s/step
This image is 0.00% Grape__Black_rot , 100.00% Grape__Esca_(Black_Measles) .100.00% Grape__healthy and 100.00% Grape__Leaf_blight_(Isariopsis_Leaf_Spot)
1/1 [=====] - 2s 2s/step
This image is 0.00% Grape__Black_rot , 100.00% Grape__Esca_(Black_Measles) .100.00% Grape__healthy and 100.00% Grape__Leaf_blight_(Isariopsis_Leaf_Spot)
```