

Resnet50 Model

Action Classes - 15

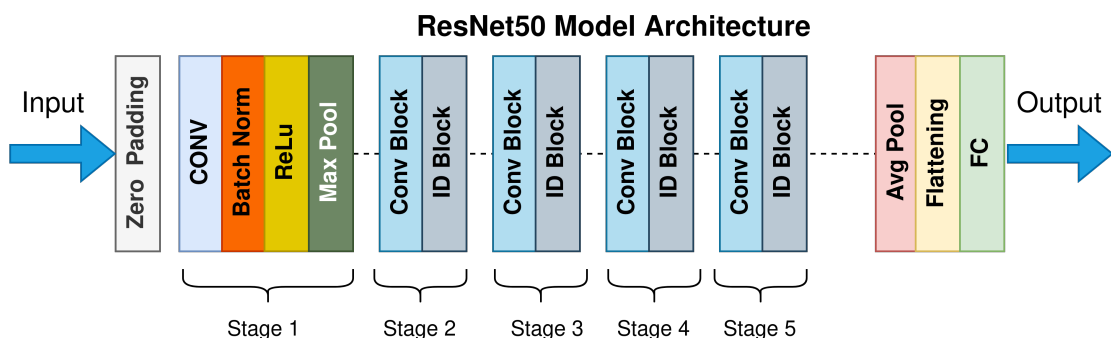
```
In [1]: from keras import models
from keras.layers import Dense, Flatten
from keras import backend as K
import numpy as np
import matplotlib.pyplot as plt

from keras.applications import resnet
```

```
In [2]: import tensorflow as tf
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))
```

Num GPUs Available: 1

2022-08-31 16:27:04.894499: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
 2022-08-31 16:27:04.928802: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
 2022-08-31 16:27:04.929106: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero



Dataset

```
In [3]: from keras.preprocessing.image import ImageDataGenerator

dataset_path = "./frames/"
# will contain the categories in respective folders

# Data generators
# train_datagen = ImageDataGenerator(rescale=1/255, validation_split=0.2)
train_datagen = ImageDataGenerator(dtype = 'float32', preprocessing_func
```

```
In [4]: image_size = (224,224)
        batch_size = 10

        train_batches = train_datagen.flow_from_directory(
            dataset_path,
            target_size = image_size,
            batch_size = batch_size,
            class_mode = "categorical",
            subset = "training"
        )

        validation_batches = train_datagen.flow_from_directory(
            dataset_path,
            target_size = image_size,
            batch_size = batch_size,
            class_mode = "categorical",
            subset = "validation"
        )

        test_batches = train_datagen.flow_from_directory(
            dataset_path,
            target_size = image_size,
            batch_size = batch_size,
            class_mode = "categorical",
            subset = "validation"
        )
```

Found 3767 images belonging to 15 classes.
Found 934 images belonging to 15 classes.
Found 934 images belonging to 15 classes.

```
In [5]: train_batches.class_indices
```

```
Out[5]: {'ApplyLipstick': 0,
         'Archery': 1,
         'BabyCrawling': 2,
         'Biking': 3,
         'Diving': 4,
         'Fencing': 5,
         'Kayaking': 6,
         'MilitaryParade': 7,
         'PizzaTossing': 8,
         'ShavingBeard': 9,
         'SkateBoarding': 10,
         'SumoWrestling': 11,
         'TennisSwing': 12,
         'Typing': 13,
         'WritingOnBoard': 14}
```

```
In [6]: from matplotlib import pyplot as plt

        def plot_images(images_arr):
            fig, axes = plt.subplots(1,10)
            axes = axes.flatten()
            for img, ax in zip(images_arr, axes):
                ax.imshow(img)
                ax.axis('off')
            plt.tight_layout()
            plt.show()
```

```
In [7]: imgs, labels = train_batches[0]
        plot_images(imgs)
        print(labels[:10])
```

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).

Clipping input data to the valid range for imshow with RGB data ([0..1] f or floats or [0..255] for integers).



```
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

```
In [8]: resnet50modeltop = resnet.ResNet50(include_top=True,
        input_shape=(224,224,3),
        pooling='avg',
        weights='imagenet')

        for (i,layer) in enumerate(resnet50modeltop.layers):
            print((i, layer.name, layer.output_shape))
```

```
2022-08-31 16:27:06.086318: 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.
2022-08-31 16:27:06.086962: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-08-31 16:27:06.087261: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-08-31 16:27:06.087578: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-08-31 16:27:06.551973: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-08-31 16:27:06.552183: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-08-31 16:27:06.552299: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:975] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2022-08-31 16:27:06.552432: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1532] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 3368 MB memory: -> device: 0, name: NVIDIA GeForce GTX 1050 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1
```

```
(0, 'input_1', [(None, 224, 224, 3)])
(1, 'conv1_pad', (None, 230, 230, 3))
(2, 'conv1_conv', (None, 112, 112, 64))
(3, 'conv1_bn', (None, 112, 112, 64))
(4, 'conv1_relu', (None, 112, 112, 64))
(5, 'pool1_pad', (None, 114, 114, 64))
(6, 'pool1_pool', (None, 56, 56, 64))
(7, 'conv2_block1_1_conv', (None, 56, 56, 64))
(8, 'conv2_block1_1_bn', (None, 56, 56, 64))
(9, 'conv2_block1_1_relu', (None, 56, 56, 64))
(10, 'conv2_block1_2_conv', (None, 56, 56, 64))
(11, 'conv2_block1_2_bn', (None, 56, 56, 64))
(12, 'conv2_block1_2_relu', (None, 56, 56, 64))
(13, 'conv2_block1_0_conv', (None, 56, 56, 256))
(14, 'conv2_block1_3_conv', (None, 56, 56, 256))
(15, 'conv2_block1_0_bn', (None, 56, 56, 256))
(16, 'conv2_block1_3_bn', (None, 56, 56, 256))
(17, 'conv2_block1_add', (None, 56, 56, 256))
(18, 'conv2_block1_out', (None, 56, 56, 256))
(19, 'conv2_block2_1_conv', (None, 56, 56, 64))
(20, 'conv2_block2_1_bn', (None, 56, 56, 64))
(21, 'conv2_block2_1_relu', (None, 56, 56, 64))
(22, 'conv2_block2_2_conv', (None, 56, 56, 64))
(23, 'conv2_block2_2_bn', (None, 56, 56, 64))
(24, 'conv2_block2_2_relu', (None, 56, 56, 64))
(25, 'conv2_block2_3_conv', (None, 56, 56, 256))
(26, 'conv2_block2_3_bn', (None, 56, 56, 256))
(27, 'conv2_block2_add', (None, 56, 56, 256))
(28, 'conv2_block2_out', (None, 56, 56, 256))
(29, 'conv2_block3_1_conv', (None, 56, 56, 64))
(30, 'conv2_block3_1_bn', (None, 56, 56, 64))
(31, 'conv2_block3_1_relu', (None, 56, 56, 64))
(32, 'conv2_block3_2_conv', (None, 56, 56, 64))
(33, 'conv2_block3_2_bn', (None, 56, 56, 64))
(34, 'conv2_block3_2_relu', (None, 56, 56, 64))
(35, 'conv2_block3_3_conv', (None, 56, 56, 256))
(36, 'conv2_block3_3_bn', (None, 56, 56, 256))
(37, 'conv2_block3_add', (None, 56, 56, 256))
(38, 'conv2_block3_out', (None, 56, 56, 256))
(39, 'conv3_block1_1_conv', (None, 28, 28, 128))
(40, 'conv3_block1_1_bn', (None, 28, 28, 128))
(41, 'conv3_block1_1_relu', (None, 28, 28, 128))
(42, 'conv3_block1_2_conv', (None, 28, 28, 128))
(43, 'conv3_block1_2_bn', (None, 28, 28, 128))
(44, 'conv3_block1_2_relu', (None, 28, 28, 128))
(45, 'conv3_block1_0_conv', (None, 28, 28, 512))
(46, 'conv3_block1_3_conv', (None, 28, 28, 512))
(47, 'conv3_block1_0_bn', (None, 28, 28, 512))
(48, 'conv3_block1_3_bn', (None, 28, 28, 512))
(49, 'conv3_block1_add', (None, 28, 28, 512))
(50, 'conv3_block1_out', (None, 28, 28, 512))
(51, 'conv3_block2_1_conv', (None, 28, 28, 128))
(52, 'conv3_block2_1_bn', (None, 28, 28, 128))
(53, 'conv3_block2_1_relu', (None, 28, 28, 128))
(54, 'conv3_block2_2_conv', (None, 28, 28, 128))
(55, 'conv3_block2_2_bn', (None, 28, 28, 128))
(56, 'conv3_block2_2_relu', (None, 28, 28, 128))
(57, 'conv3_block2_3_conv', (None, 28, 28, 512))
(58, 'conv3_block2_3_bn', (None, 28, 28, 512))
(59, 'conv3_block2_add', (None, 28, 28, 512))
(60, 'conv3_block2_out', (None, 28, 28, 512))
(61, 'conv3_block3_1_conv', (None, 28, 28, 128))
(62, 'conv3_block3_1_bn', (None, 28, 28, 128))
```

```
(63, 'conv3_block3_1_relu', (None, 28, 28, 128))
(64, 'conv3_block3_2_conv', (None, 28, 28, 128))
(65, 'conv3_block3_2_bn', (None, 28, 28, 128))
(66, 'conv3_block3_2_relu', (None, 28, 28, 128))
(67, 'conv3_block3_3_conv', (None, 28, 28, 512))
(68, 'conv3_block3_3_bn', (None, 28, 28, 512))
(69, 'conv3_block3_add', (None, 28, 28, 512))
(70, 'conv3_block3_out', (None, 28, 28, 512))
(71, 'conv3_block4_1_conv', (None, 28, 28, 128))
(72, 'conv3_block4_1_bn', (None, 28, 28, 128))
(73, 'conv3_block4_1_relu', (None, 28, 28, 128))
(74, 'conv3_block4_2_conv', (None, 28, 28, 128))
(75, 'conv3_block4_2_bn', (None, 28, 28, 128))
(76, 'conv3_block4_2_relu', (None, 28, 28, 128))
(77, 'conv3_block4_3_conv', (None, 28, 28, 512))
(78, 'conv3_block4_3_bn', (None, 28, 28, 512))
(79, 'conv3_block4_add', (None, 28, 28, 512))
(80, 'conv3_block4_out', (None, 28, 28, 512))
(81, 'conv4_block1_1_conv', (None, 14, 14, 256))
(82, 'conv4_block1_1_bn', (None, 14, 14, 256))
(83, 'conv4_block1_1_relu', (None, 14, 14, 256))
(84, 'conv4_block1_2_conv', (None, 14, 14, 256))
(85, 'conv4_block1_2_bn', (None, 14, 14, 256))
(86, 'conv4_block1_2_relu', (None, 14, 14, 256))
(87, 'conv4_block1_0_conv', (None, 14, 14, 1024))
(88, 'conv4_block1_3_conv', (None, 14, 14, 1024))
(89, 'conv4_block1_0_bn', (None, 14, 14, 1024))
(90, 'conv4_block1_3_bn', (None, 14, 14, 1024))
(91, 'conv4_block1_add', (None, 14, 14, 1024))
(92, 'conv4_block1_out', (None, 14, 14, 1024))
(93, 'conv4_block2_1_conv', (None, 14, 14, 256))
(94, 'conv4_block2_1_bn', (None, 14, 14, 256))
(95, 'conv4_block2_1_relu', (None, 14, 14, 256))
(96, 'conv4_block2_2_conv', (None, 14, 14, 256))
(97, 'conv4_block2_2_bn', (None, 14, 14, 256))
(98, 'conv4_block2_2_relu', (None, 14, 14, 256))
(99, 'conv4_block2_3_conv', (None, 14, 14, 1024))
(100, 'conv4_block2_3_bn', (None, 14, 14, 1024))
(101, 'conv4_block2_add', (None, 14, 14, 1024))
(102, 'conv4_block2_out', (None, 14, 14, 1024))
(103, 'conv4_block3_1_conv', (None, 14, 14, 256))
(104, 'conv4_block3_1_bn', (None, 14, 14, 256))
(105, 'conv4_block3_1_relu', (None, 14, 14, 256))
(106, 'conv4_block3_2_conv', (None, 14, 14, 256))
(107, 'conv4_block3_2_bn', (None, 14, 14, 256))
(108, 'conv4_block3_2_relu', (None, 14, 14, 256))
(109, 'conv4_block3_3_conv', (None, 14, 14, 1024))
(110, 'conv4_block3_3_bn', (None, 14, 14, 1024))
(111, 'conv4_block3_add', (None, 14, 14, 1024))
(112, 'conv4_block3_out', (None, 14, 14, 1024))
(113, 'conv4_block4_1_conv', (None, 14, 14, 256))
(114, 'conv4_block4_1_bn', (None, 14, 14, 256))
(115, 'conv4_block4_1_relu', (None, 14, 14, 256))
(116, 'conv4_block4_2_conv', (None, 14, 14, 256))
(117, 'conv4_block4_2_bn', (None, 14, 14, 256))
(118, 'conv4_block4_2_relu', (None, 14, 14, 256))
(119, 'conv4_block4_3_conv', (None, 14, 14, 1024))
(120, 'conv4_block4_3_bn', (None, 14, 14, 1024))
(121, 'conv4_block4_add', (None, 14, 14, 1024))
(122, 'conv4_block4_out', (None, 14, 14, 1024))
(123, 'conv4_block5_1_conv', (None, 14, 14, 256))
(124, 'conv4_block5_1_bn', (None, 14, 14, 256))
(125, 'conv4_block5_1_relu', (None, 14, 14, 256))
```

```

(126, 'conv4_block5_2_conv', (None, 14, 14, 256))
(127, 'conv4_block5_2_bn', (None, 14, 14, 256))
(128, 'conv4_block5_2_relu', (None, 14, 14, 256))
(129, 'conv4_block5_3_conv', (None, 14, 14, 1024))
(130, 'conv4_block5_3_bn', (None, 14, 14, 1024))
(131, 'conv4_block5_add', (None, 14, 14, 1024))
(132, 'conv4_block5_out', (None, 14, 14, 1024))
(133, 'conv4_block6_1_conv', (None, 14, 14, 256))
(134, 'conv4_block6_1_bn', (None, 14, 14, 256))
(135, 'conv4_block6_1_relu', (None, 14, 14, 256))
(136, 'conv4_block6_2_conv', (None, 14, 14, 256))
(137, 'conv4_block6_2_bn', (None, 14, 14, 256))
(138, 'conv4_block6_2_relu', (None, 14, 14, 256))
(139, 'conv4_block6_3_conv', (None, 14, 14, 1024))
(140, 'conv4_block6_3_bn', (None, 14, 14, 1024))
(141, 'conv4_block6_add', (None, 14, 14, 1024))
(142, 'conv4_block6_out', (None, 14, 14, 1024))
(143, 'conv5_block1_1_conv', (None, 7, 7, 512))
(144, 'conv5_block1_1_bn', (None, 7, 7, 512))
(145, 'conv5_block1_1_relu', (None, 7, 7, 512))
(146, 'conv5_block1_2_conv', (None, 7, 7, 512))
(147, 'conv5_block1_2_bn', (None, 7, 7, 512))
(148, 'conv5_block1_2_relu', (None, 7, 7, 512))
(149, 'conv5_block1_0_conv', (None, 7, 7, 2048))
(150, 'conv5_block1_3_conv', (None, 7, 7, 2048))
(151, 'conv5_block1_0_bn', (None, 7, 7, 2048))
(152, 'conv5_block1_3_bn', (None, 7, 7, 2048))
(153, 'conv5_block1_add', (None, 7, 7, 2048))
(154, 'conv5_block1_out', (None, 7, 7, 2048))
(155, 'conv5_block2_1_conv', (None, 7, 7, 512))
(156, 'conv5_block2_1_bn', (None, 7, 7, 512))
(157, 'conv5_block2_1_relu', (None, 7, 7, 512))
(158, 'conv5_block2_2_conv', (None, 7, 7, 512))
(159, 'conv5_block2_2_bn', (None, 7, 7, 512))
(160, 'conv5_block2_2_relu', (None, 7, 7, 512))
(161, 'conv5_block2_3_conv', (None, 7, 7, 2048))
(162, 'conv5_block2_3_bn', (None, 7, 7, 2048))
(163, 'conv5_block2_add', (None, 7, 7, 2048))
(164, 'conv5_block2_out', (None, 7, 7, 2048))
(165, 'conv5_block3_1_conv', (None, 7, 7, 512))
(166, 'conv5_block3_1_bn', (None, 7, 7, 512))
(167, 'conv5_block3_1_relu', (None, 7, 7, 512))
(168, 'conv5_block3_2_conv', (None, 7, 7, 512))
(169, 'conv5_block3_2_bn', (None, 7, 7, 512))
(170, 'conv5_block3_2_relu', (None, 7, 7, 512))
(171, 'conv5_block3_3_conv', (None, 7, 7, 2048))
(172, 'conv5_block3_3_bn', (None, 7, 7, 2048))
(173, 'conv5_block3_add', (None, 7, 7, 2048))
(174, 'conv5_block3_out', (None, 7, 7, 2048))
(175, 'avg_pool', (None, 2048))
(176, 'predictions', (None, 1000))

```

```

In [9]: resnet50model = resnet.ResNet50(include_top=False,
                                         input_shape=(224,224,3),
                                         pooling='avg', classes=15,
                                         weights='imagenet')

for (i, layer) in enumerate(resnet50model.layers):
    layer.trainable = False
    print((i, layer.name, layer.output_shape, layer.trainable))

```

```
(0, 'input_2', [(None, 224, 224, 3)], False)
(1, 'conv1_pad', (None, 230, 230, 3), False)
(2, 'conv1_conv', (None, 112, 112, 64), False)
(3, 'conv1_bn', (None, 112, 112, 64), False)
(4, 'conv1_relu', (None, 112, 112, 64), False)
(5, 'pool1_pad', (None, 114, 114, 64), False)
(6, 'pool1_pool', (None, 56, 56, 64), False)
(7, 'conv2_block1_1_conv', (None, 56, 56, 64), False)
(8, 'conv2_block1_1_bn', (None, 56, 56, 64), False)
(9, 'conv2_block1_1_relu', (None, 56, 56, 64), False)
(10, 'conv2_block1_2_conv', (None, 56, 56, 64), False)
(11, 'conv2_block1_2_bn', (None, 56, 56, 64), False)
(12, 'conv2_block1_2_relu', (None, 56, 56, 64), False)
(13, 'conv2_block1_0_conv', (None, 56, 56, 256), False)
(14, 'conv2_block1_3_conv', (None, 56, 56, 256), False)
(15, 'conv2_block1_0_bn', (None, 56, 56, 256), False)
(16, 'conv2_block1_3_bn', (None, 56, 56, 256), False)
(17, 'conv2_block1_add', (None, 56, 56, 256), False)
(18, 'conv2_block1_out', (None, 56, 56, 256), False)
(19, 'conv2_block2_1_conv', (None, 56, 56, 64), False)
(20, 'conv2_block2_1_bn', (None, 56, 56, 64), False)
(21, 'conv2_block2_1_relu', (None, 56, 56, 64), False)
(22, 'conv2_block2_2_conv', (None, 56, 56, 64), False)
(23, 'conv2_block2_2_bn', (None, 56, 56, 64), False)
(24, 'conv2_block2_2_relu', (None, 56, 56, 64), False)
(25, 'conv2_block2_3_conv', (None, 56, 56, 256), False)
(26, 'conv2_block2_3_bn', (None, 56, 56, 256), False)
(27, 'conv2_block2_add', (None, 56, 56, 256), False)
(28, 'conv2_block2_out', (None, 56, 56, 256), False)
(29, 'conv2_block3_1_conv', (None, 56, 56, 64), False)
(30, 'conv2_block3_1_bn', (None, 56, 56, 64), False)
(31, 'conv2_block3_1_relu', (None, 56, 56, 64), False)
(32, 'conv2_block3_2_conv', (None, 56, 56, 64), False)
(33, 'conv2_block3_2_bn', (None, 56, 56, 64), False)
(34, 'conv2_block3_2_relu', (None, 56, 56, 64), False)
(35, 'conv2_block3_3_conv', (None, 56, 56, 256), False)
(36, 'conv2_block3_3_bn', (None, 56, 56, 256), False)
(37, 'conv2_block3_add', (None, 56, 56, 256), False)
(38, 'conv2_block3_out', (None, 56, 56, 256), False)
(39, 'conv3_block1_1_conv', (None, 28, 28, 128), False)
(40, 'conv3_block1_1_bn', (None, 28, 28, 128), False)
(41, 'conv3_block1_1_relu', (None, 28, 28, 128), False)
(42, 'conv3_block1_2_conv', (None, 28, 28, 128), False)
(43, 'conv3_block1_2_bn', (None, 28, 28, 128), False)
(44, 'conv3_block1_2_relu', (None, 28, 28, 128), False)
(45, 'conv3_block1_0_conv', (None, 28, 28, 512), False)
(46, 'conv3_block1_3_conv', (None, 28, 28, 512), False)
(47, 'conv3_block1_0_bn', (None, 28, 28, 512), False)
(48, 'conv3_block1_3_bn', (None, 28, 28, 512), False)
(49, 'conv3_block1_add', (None, 28, 28, 512), False)
(50, 'conv3_block1_out', (None, 28, 28, 512), False)
(51, 'conv3_block2_1_conv', (None, 28, 28, 128), False)
(52, 'conv3_block2_1_bn', (None, 28, 28, 128), False)
(53, 'conv3_block2_1_relu', (None, 28, 28, 128), False)
(54, 'conv3_block2_2_conv', (None, 28, 28, 128), False)
(55, 'conv3_block2_2_bn', (None, 28, 28, 128), False)
(56, 'conv3_block2_2_relu', (None, 28, 28, 128), False)
(57, 'conv3_block2_3_conv', (None, 28, 28, 512), False)
(58, 'conv3_block2_3_bn', (None, 28, 28, 512), False)
(59, 'conv3_block2_add', (None, 28, 28, 512), False)
(60, 'conv3_block2_out', (None, 28, 28, 512), False)
(61, 'conv3_block3_1_conv', (None, 28, 28, 128), False)
(62, 'conv3_block3_1_bn', (None, 28, 28, 128), False)
```



```
(63, 'conv3_block3_1_relu', (None, 28, 28, 128), False)
(64, 'conv3_block3_2_conv', (None, 28, 28, 128), False)
(65, 'conv3_block3_2_bn', (None, 28, 28, 128), False)
(66, 'conv3_block3_2_relu', (None, 28, 28, 128), False)
(67, 'conv3_block3_3_conv', (None, 28, 28, 512), False)
(68, 'conv3_block3_3_bn', (None, 28, 28, 512), False)
(69, 'conv3_block3_add', (None, 28, 28, 512), False)
(70, 'conv3_block3_out', (None, 28, 28, 512), False)
(71, 'conv3_block4_1_conv', (None, 28, 28, 128), False)
(72, 'conv3_block4_1_bn', (None, 28, 28, 128), False)
(73, 'conv3_block4_1_relu', (None, 28, 28, 128), False)
(74, 'conv3_block4_2_conv', (None, 28, 28, 128), False)
(75, 'conv3_block4_2_bn', (None, 28, 28, 128), False)
(76, 'conv3_block4_2_relu', (None, 28, 28, 128), False)
(77, 'conv3_block4_3_conv', (None, 28, 28, 512), False)
(78, 'conv3_block4_3_bn', (None, 28, 28, 512), False)
(79, 'conv3_block4_add', (None, 28, 28, 512), False)
(80, 'conv3_block4_out', (None, 28, 28, 512), False)
(81, 'conv4_block1_1_conv', (None, 14, 14, 256), False)
(82, 'conv4_block1_1_bn', (None, 14, 14, 256), False)
(83, 'conv4_block1_1_relu', (None, 14, 14, 256), False)
(84, 'conv4_block1_2_conv', (None, 14, 14, 256), False)
(85, 'conv4_block1_2_bn', (None, 14, 14, 256), False)
(86, 'conv4_block1_2_relu', (None, 14, 14, 256), False)
(87, 'conv4_block1_0_conv', (None, 14, 14, 1024), False)
(88, 'conv4_block1_3_conv', (None, 14, 14, 1024), False)
(89, 'conv4_block1_0_bn', (None, 14, 14, 1024), False)
(90, 'conv4_block1_3_bn', (None, 14, 14, 1024), False)
(91, 'conv4_block1_add', (None, 14, 14, 1024), False)
(92, 'conv4_block1_out', (None, 14, 14, 1024), False)
(93, 'conv4_block2_1_conv', (None, 14, 14, 256), False)
(94, 'conv4_block2_1_bn', (None, 14, 14, 256), False)
(95, 'conv4_block2_1_relu', (None, 14, 14, 256), False)
(96, 'conv4_block2_2_conv', (None, 14, 14, 256), False)
(97, 'conv4_block2_2_bn', (None, 14, 14, 256), False)
(98, 'conv4_block2_2_relu', (None, 14, 14, 256), False)
(99, 'conv4_block2_3_conv', (None, 14, 14, 1024), False)
(100, 'conv4_block2_3_bn', (None, 14, 14, 1024), False)
(101, 'conv4_block2_add', (None, 14, 14, 1024), False)
(102, 'conv4_block2_out', (None, 14, 14, 1024), False)
(103, 'conv4_block3_1_conv', (None, 14, 14, 256), False)
(104, 'conv4_block3_1_bn', (None, 14, 14, 256), False)
(105, 'conv4_block3_1_relu', (None, 14, 14, 256), False)
(106, 'conv4_block3_2_conv', (None, 14, 14, 256), False)
(107, 'conv4_block3_2_bn', (None, 14, 14, 256), False)
(108, 'conv4_block3_2_relu', (None, 14, 14, 256), False)
(109, 'conv4_block3_3_conv', (None, 14, 14, 1024), False)
(110, 'conv4_block3_3_bn', (None, 14, 14, 1024), False)
(111, 'conv4_block3_add', (None, 14, 14, 1024), False)
(112, 'conv4_block3_out', (None, 14, 14, 1024), False)
(113, 'conv4_block4_1_conv', (None, 14, 14, 256), False)
(114, 'conv4_block4_1_bn', (None, 14, 14, 256), False)
(115, 'conv4_block4_1_relu', (None, 14, 14, 256), False)
(116, 'conv4_block4_2_conv', (None, 14, 14, 256), False)
(117, 'conv4_block4_2_bn', (None, 14, 14, 256), False)
(118, 'conv4_block4_2_relu', (None, 14, 14, 256), False)
(119, 'conv4_block4_3_conv', (None, 14, 14, 1024), False)
(120, 'conv4_block4_3_bn', (None, 14, 14, 1024), False)
(121, 'conv4_block4_add', (None, 14, 14, 1024), False)
(122, 'conv4_block4_out', (None, 14, 14, 1024), False)
(123, 'conv4_block5_1_conv', (None, 14, 14, 256), False)
(124, 'conv4_block5_1_bn', (None, 14, 14, 256), False)
(125, 'conv4_block5_1_relu', (None, 14, 14, 256), False)
```

```
(126, 'conv4_block5_2_conv', (None, 14, 14, 256), False)
(127, 'conv4_block5_2_bn', (None, 14, 14, 256), False)
(128, 'conv4_block5_2_relu', (None, 14, 14, 256), False)
(129, 'conv4_block5_3_conv', (None, 14, 14, 1024), False)
(130, 'conv4_block5_3_bn', (None, 14, 14, 1024), False)
(131, 'conv4_block5_add', (None, 14, 14, 1024), False)
(132, 'conv4_block5_out', (None, 14, 14, 1024), False)
(133, 'conv4_block6_1_conv', (None, 14, 14, 256), False)
(134, 'conv4_block6_1_bn', (None, 14, 14, 256), False)
(135, 'conv4_block6_1_relu', (None, 14, 14, 256), False)
(136, 'conv4_block6_2_conv', (None, 14, 14, 256), False)
(137, 'conv4_block6_2_bn', (None, 14, 14, 256), False)
(138, 'conv4_block6_2_relu', (None, 14, 14, 256), False)
(139, 'conv4_block6_3_conv', (None, 14, 14, 1024), False)
(140, 'conv4_block6_3_bn', (None, 14, 14, 1024), False)
(141, 'conv4_block6_add', (None, 14, 14, 1024), False)
(142, 'conv4_block6_out', (None, 14, 14, 1024), False)
(143, 'conv5_block1_1_conv', (None, 7, 7, 512), False)
(144, 'conv5_block1_1_bn', (None, 7, 7, 512), False)
(145, 'conv5_block1_1_relu', (None, 7, 7, 512), False)
(146, 'conv5_block1_2_conv', (None, 7, 7, 512), False)
(147, 'conv5_block1_2_bn', (None, 7, 7, 512), False)
(148, 'conv5_block1_2_relu', (None, 7, 7, 512), False)
(149, 'conv5_block1_0_conv', (None, 7, 7, 2048), False)
(150, 'conv5_block1_3_conv', (None, 7, 7, 2048), False)
(151, 'conv5_block1_0_bn', (None, 7, 7, 2048), False)
(152, 'conv5_block1_3_bn', (None, 7, 7, 2048), False)
(153, 'conv5_block1_add', (None, 7, 7, 2048), False)
(154, 'conv5_block1_out', (None, 7, 7, 2048), False)
(155, 'conv5_block2_1_conv', (None, 7, 7, 512), False)
(156, 'conv5_block2_1_bn', (None, 7, 7, 512), False)
(157, 'conv5_block2_1_relu', (None, 7, 7, 512), False)
(158, 'conv5_block2_2_conv', (None, 7, 7, 512), False)
(159, 'conv5_block2_2_bn', (None, 7, 7, 512), False)
(160, 'conv5_block2_2_relu', (None, 7, 7, 512), False)
(161, 'conv5_block2_3_conv', (None, 7, 7, 2048), False)
(162, 'conv5_block2_3_bn', (None, 7, 7, 2048), False)
(163, 'conv5_block2_add', (None, 7, 7, 2048), False)
(164, 'conv5_block2_out', (None, 7, 7, 2048), False)
(165, 'conv5_block3_1_conv', (None, 7, 7, 512), False)
(166, 'conv5_block3_1_bn', (None, 7, 7, 512), False)
(167, 'conv5_block3_1_relu', (None, 7, 7, 512), False)
(168, 'conv5_block3_2_conv', (None, 7, 7, 512), False)
(169, 'conv5_block3_2_bn', (None, 7, 7, 512), False)
(170, 'conv5_block3_2_relu', (None, 7, 7, 512), False)
(171, 'conv5_block3_3_conv', (None, 7, 7, 2048), False)
(172, 'conv5_block3_3_bn', (None, 7, 7, 2048), False)
(173, 'conv5_block3_add', (None, 7, 7, 2048), False)
(174, 'conv5_block3_out', (None, 7, 7, 2048), False)
(175, 'avg_pool', (None, 2048), False)
```

```
In [10]: model = models.Sequential()

flatten = Flatten() # adding Flatten Layer
dense_layer_1 = Dense(32, activation='relu') # Adding a Dense layer
prediction_layer = Dense(15, activation='softmax')

model.add(resnet50model)
model.add(flatten)
model.add(dense_layer_1)
model.add(prediction_layer)

model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2048)	23587712
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 32)	65568
dense_1 (Dense)	(None, 15)	495
Total params: 23,653,775		
Trainable params: 66,063		
Non-trainable params: 23,587,712		

```
In [11]: model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy'],
)
```

```
In [12]: model.save("./models/action-class-15-resnet50.h5")
```

```
In [13]: fit = model.fit(train_batches, epochs=20, validation_data=validation_batches)
```

Epoch 1/20

```
2022-08-31 16:27:16.140097: I tensorflow/stream_executor/cuda/cuda_dnn.cc:384] Loaded cuDNN version 8401
2022-08-31 16:27:17.618666: I tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child process: No such file or directory
```

```

377/377 [=====] - 56s 126ms/step - loss: 0.3637
- accuracy: 0.9044 - val_loss: 0.2988 - val_accuracy: 0.9143
Epoch 2/20
377/377 [=====] - 41s 110ms/step - loss: 0.0325
- accuracy: 0.9942 - val_loss: 0.3690 - val_accuracy: 0.8994
Epoch 3/20
377/377 [=====] - 42s 112ms/step - loss: 0.0089
- accuracy: 0.9997 - val_loss: 0.2462 - val_accuracy: 0.9208
Epoch 4/20
377/377 [=====] - 51s 136ms/step - loss: 0.0038
- accuracy: 1.0000 - val_loss: 0.2743 - val_accuracy: 0.9218
Epoch 5/20
377/377 [=====] - 58s 154ms/step - loss: 0.0024
- accuracy: 1.0000 - val_loss: 0.2799 - val_accuracy: 0.9261
Epoch 6/20
377/377 [=====] - 77s 204ms/step - loss: 0.0015
- accuracy: 1.0000 - val_loss: 0.2832 - val_accuracy: 0.9251
Epoch 7/20
377/377 [=====] - 73s 193ms/step - loss: 0.0011
- accuracy: 1.0000 - val_loss: 0.2824 - val_accuracy: 0.9293
Epoch 8/20
377/377 [=====] - 79s 209ms/step - loss: 8.2758e
-04 - accuracy: 1.0000 - val_loss: 0.2772 - val_accuracy: 0.9315
Epoch 9/20
377/377 [=====] - 77s 203ms/step - loss: 6.3000e
-04 - accuracy: 1.0000 - val_loss: 0.2924 - val_accuracy: 0.9283
Epoch 10/20
377/377 [=====] - 83s 219ms/step - loss: 4.9511e
-04 - accuracy: 1.0000 - val_loss: 0.2899 - val_accuracy: 0.9336
Epoch 11/20
377/377 [=====] - 80s 213ms/step - loss: 3.8374e
-04 - accuracy: 1.0000 - val_loss: 0.3002 - val_accuracy: 0.9261
Epoch 12/20
377/377 [=====] - 80s 211ms/step - loss: 3.0551e
-04 - accuracy: 1.0000 - val_loss: 0.2892 - val_accuracy: 0.9315
Epoch 13/20
377/377 [=====] - 83s 220ms/step - loss: 2.3718e
-04 - accuracy: 1.0000 - val_loss: 0.2887 - val_accuracy: 0.9293
Epoch 14/20
377/377 [=====] - 74s 195ms/step - loss: 1.9346e
-04 - accuracy: 1.0000 - val_loss: 0.3030 - val_accuracy: 0.9315
Epoch 15/20
377/377 [=====] - 80s 214ms/step - loss: 1.5707e
-04 - accuracy: 1.0000 - val_loss: 0.3038 - val_accuracy: 0.9315
Epoch 16/20
377/377 [=====] - 80s 212ms/step - loss: 1.2265e
-04 - accuracy: 1.0000 - val_loss: 0.3095 - val_accuracy: 0.9293
Epoch 17/20
377/377 [=====] - 73s 194ms/step - loss: 1.0048e
-04 - accuracy: 1.0000 - val_loss: 0.2942 - val_accuracy: 0.9315
Epoch 18/20
377/377 [=====] - 74s 195ms/step - loss: 8.0115e
-05 - accuracy: 1.0000 - val_loss: 0.3129 - val_accuracy: 0.9293
Epoch 19/20
377/377 [=====] - 62s 165ms/step - loss: 6.5001e
-05 - accuracy: 1.0000 - val_loss: 0.3129 - val_accuracy: 0.9304
Epoch 20/20
377/377 [=====] - 60s 158ms/step - loss: 5.2101e
-05 - accuracy: 1.0000 - val_loss: 0.3133 - val_accuracy: 0.9336

```

```
In [14]: model.save("./models/action-class-15-trained-resnet50.h5")
```

Evaluate and Predict

```
In [15]: model = models.load_model("./models/action-class-15-trained-resnet50.h5")
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 2048)	23587712
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 32)	65568
dense_1 (Dense)	(None, 15)	495

Total params: 23,653,775
 Trainable params: 66,063
 Non-trainable params: 23,587,712

```
In [16]: model.evaluate(test_batches)
```

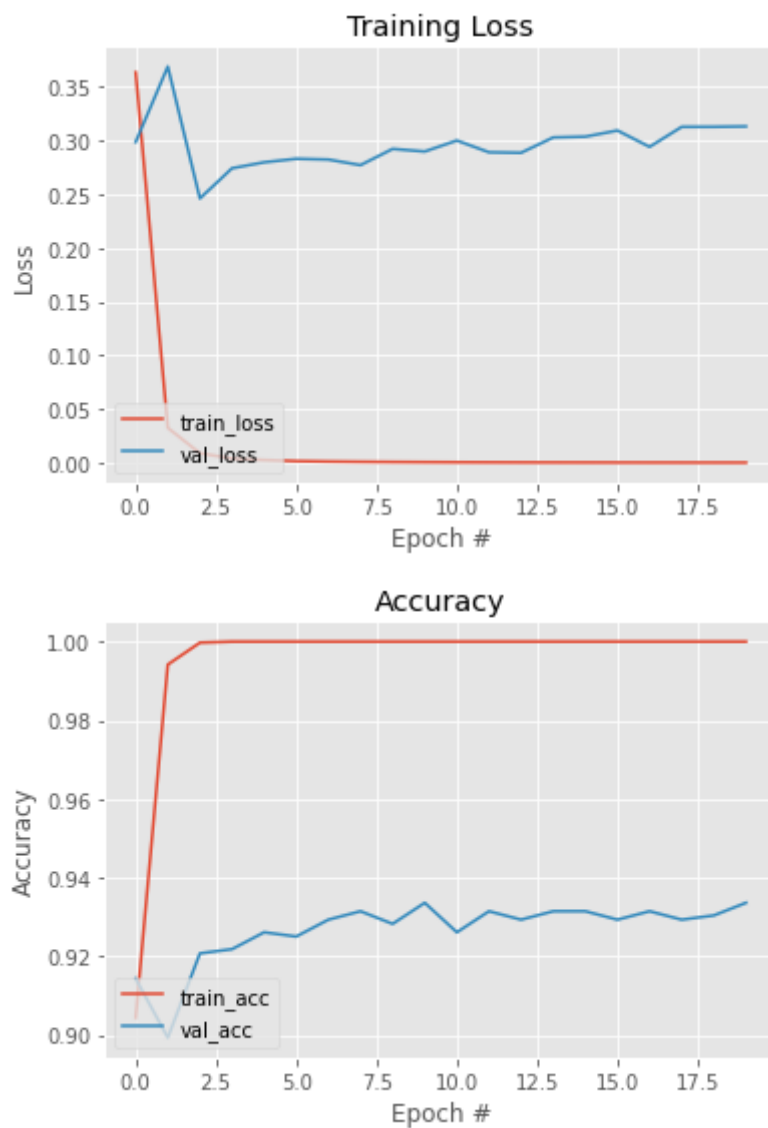
94/94 [=====] - 10s 92ms/step - loss: 0.3133 - accuracy: 0.9336

```
Out[16]: [0.31331899762153625, 0.9336188435554504]
```

```
In [17]: plt.style.use("ggplot")
plt.figure()
```

```
plt.plot(np.arange(0, 20), fit.history["loss"], label="train_loss")
plt.plot(np.arange(0, 20), fit.history["val_loss"], label="val_loss")
plt.title("Training Loss")
plt.xlabel("Epoch #")
plt.ylabel("Loss")
plt.legend(loc="lower left")
plt.show()
```

```
plt.plot(np.arange(0, 20), fit.history["accuracy"], label="train_acc")
plt.plot(np.arange(0, 20), fit.history["val_accuracy"], label="val_acc")
plt.title("Accuracy")
plt.xlabel("Epoch #")
plt.ylabel("Accuracy")
plt.legend(loc="lower left")
plt.show()
```



```
In [18]: print("Avg Val Acc: " + str(sum(fit.history["val_accuracy"])/20*100))  
print("Avg Val Loss: " + str(sum(fit.history["val_loss"])/20*100))  
  
Avg Val Acc: 92.67130583524704  
Avg Val Loss: 29.605583623051647
```