Resnet50 Model

Action Classes - 10

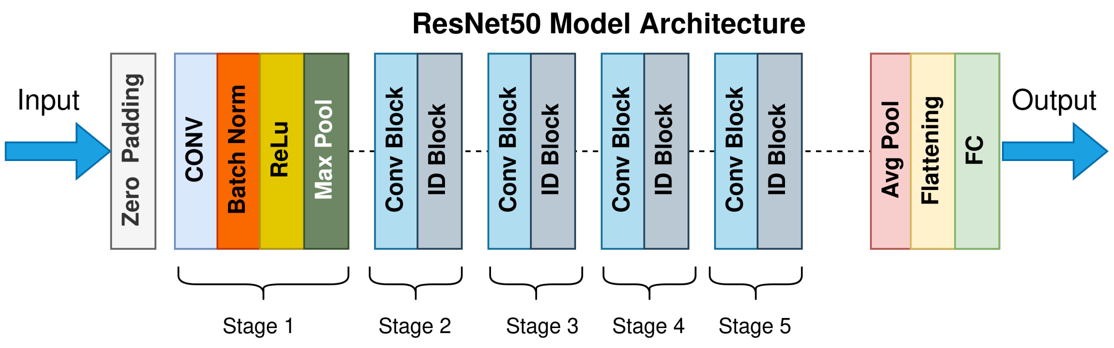
In [1]:
```
from keras import models
from keras.layers import Dense,Flatten
from keras import backend as K
import numpy as np
import matplotlib.pyplot as plt

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 16:01:21.304756: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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 16:01:21.417137: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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 16:01:21.417393: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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_funct
```

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 2734 images belonging to 10 classes.
Found 679 images belonging to 10 classes.
Found 679 images belonging to 10 classes.
```

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

Out[5]:
```
{'ApplyLipstick': 0,
 'Archery': 1,
 'Biking': 2,
 'Diving': 3,
 'Kayaking': 4,
 'MilitaryParade': 5,
 'ShavingBeard': 6,
 'SkateBoarding': 7,
 'TennisSwing': 8,
 'Typing': 9}
```

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).
```

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

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 16:01:22.586230: I tensorflow/core/platform/cpu_feature_guard.
cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Netwo
rk Library (oneDNN) to use the following CPU instructions in performance-
critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropria
te compiler flags.
2022-08-31 16:01:22.587030: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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 16:01:22.587482: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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 16:01:22.587782: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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 16:01:23.719694: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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 16:01:23.719868: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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 16:01:23.719975: I tensorflow/stream_executor/cuda/cuda_gpu_ex
ecutor.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 16:01:23.720103: I tensorflow/core/common_runtime/gpu/gpu_devi
ce.cc:1532] Created device /job:localhost/replica:0/task:0/device:GPU:0 w
ith 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]:
```python
resnet50model = resnet.ResNet50(include_top=False,
                       input_shape=(224,224,3),
                       pooling='avg',classes=10,
                       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]:
```python
model = models.Sequential()

flatten = Flatten()  # adding Flatten Layer
dense_layer_1 = Dense(32, activation='relu')  # Adding a Dense layer
prediction_layer = Dense(10, 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, 10)                330       
                                                                 
=================================================================
Total params: 23,653,610
Trainable params: 65,898
Non-trainable params: 23,587,712
_________________________________________________________________
```

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

In [12]:
```python
model.save("./models/action-class-10-resnet50.h5")
```

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

```
Epoch 1/20
2022-08-31 16:01:31.917691: I tensorflow/stream_executor/cuda/cuda_dnn.c
c:384] Loaded cuDNN version 8401
2022-08-31 16:01:33.369265: I tensorflow/core/platform/default/subproces
s.cc:304] Start cannot spawn child process: No such file or directory
```

```
274/274 [==============================] - 40s 122ms/step - loss: 0.2983
- accuracy: 0.9203 - val_loss: 0.2041 - val_accuracy: 0.9190
Epoch 2/20
274/274 [==============================] - 30s 109ms/step - loss: 0.0253
- accuracy: 0.9960 - val_loss: 0.2053 - val_accuracy: 0.9161
Epoch 3/20
274/274 [==============================] - 30s 109ms/step - loss: 0.0096
- accuracy: 0.9993 - val_loss: 0.2389 - val_accuracy: 0.9146
Epoch 4/20
274/274 [==============================] - 30s 110ms/step - loss: 0.0042
- accuracy: 1.0000 - val_loss: 0.2154 - val_accuracy: 0.9264
Epoch 5/20
274/274 [==============================] - 32s 117ms/step - loss: 0.0027
- accuracy: 1.0000 - val_loss: 0.3568 - val_accuracy: 0.8925
Epoch 6/20
274/274 [==============================] - 36s 131ms/step - loss: 0.0016
- accuracy: 1.0000 - val_loss: 0.1963 - val_accuracy: 0.9323
Epoch 7/20
274/274 [==============================] - 48s 173ms/step - loss: 0.0011
- accuracy: 1.0000 - val_loss: 0.2177 - val_accuracy: 0.9264
Epoch 8/20
274/274 [==============================] - 54s 195ms/step - loss: 8.4735e
-04 - accuracy: 1.0000 - val_loss: 0.2215 - val_accuracy: 0.9249
Epoch 9/20
274/274 [==============================] - 53s 194ms/step - loss: 6.3168e
-04 - accuracy: 1.0000 - val_loss: 0.2126 - val_accuracy: 0.9249
Epoch 10/20
274/274 [==============================] - 59s 216ms/step - loss: 4.9911e
-04 - accuracy: 1.0000 - val_loss: 0.2304 - val_accuracy: 0.9264
Epoch 11/20
274/274 [==============================] - 64s 232ms/step - loss: 4.0519e
-04 - accuracy: 1.0000 - val_loss: 0.2359 - val_accuracy: 0.9278
Epoch 12/20
274/274 [==============================] - 62s 226ms/step - loss: 3.2403e
-04 - accuracy: 1.0000 - val_loss: 0.2428 - val_accuracy: 0.9234
Epoch 13/20
274/274 [==============================] - 65s 235ms/step - loss: 2.6514e
-04 - accuracy: 1.0000 - val_loss: 0.2398 - val_accuracy: 0.9264
Epoch 14/20
274/274 [==============================] - 50s 180ms/step - loss: 2.1628e
-04 - accuracy: 1.0000 - val_loss: 0.2397 - val_accuracy: 0.9264
Epoch 15/20
274/274 [==============================] - 68s 249ms/step - loss: 1.7888e
-04 - accuracy: 1.0000 - val_loss: 0.2455 - val_accuracy: 0.9249
Epoch 16/20
274/274 [==============================] - 63s 229ms/step - loss: 1.5102e
-04 - accuracy: 1.0000 - val_loss: 0.2409 - val_accuracy: 0.9264
Epoch 17/20
274/274 [==============================] - 65s 239ms/step - loss: 1.2464e
-04 - accuracy: 1.0000 - val_loss: 0.2484 - val_accuracy: 0.9264
Epoch 18/20
274/274 [==============================] - 64s 233ms/step - loss: 1.0649e
-04 - accuracy: 1.0000 - val_loss: 0.2579 - val_accuracy: 0.9249
Epoch 19/20
274/274 [==============================] - 68s 246ms/step - loss: 8.6565e
-05 - accuracy: 1.0000 - val_loss: 0.2656 - val_accuracy: 0.9249
Epoch 20/20
274/274 [==============================] - 65s 235ms/step - loss: 7.3058e
-05 - accuracy: 1.0000 - val_loss: 0.2732 - val_accuracy: 0.9234
```

In [14]:
```
model.save("./models/action-class-10-trained-resnet50.h5")
```

Evaluate and Predict

In [15]:
```
model = models.load_model("./models/action-class-10-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, 10)                330

=================================================================
Total params: 23,653,610
Trainable params: 65,898
Non-trainable params: 23,587,712
_________________________________________________________________
```

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

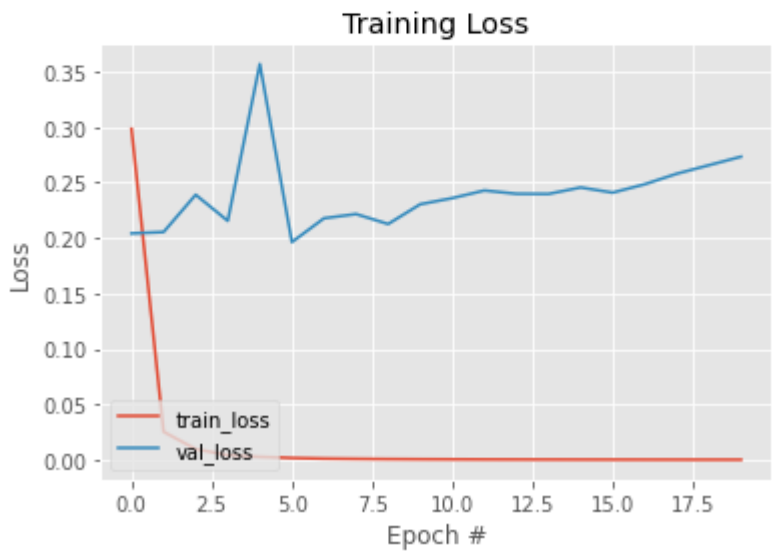
```
68/68 [==============================] - 14s 170ms/step - loss: 0.2732 -
accuracy: 0.9234
```

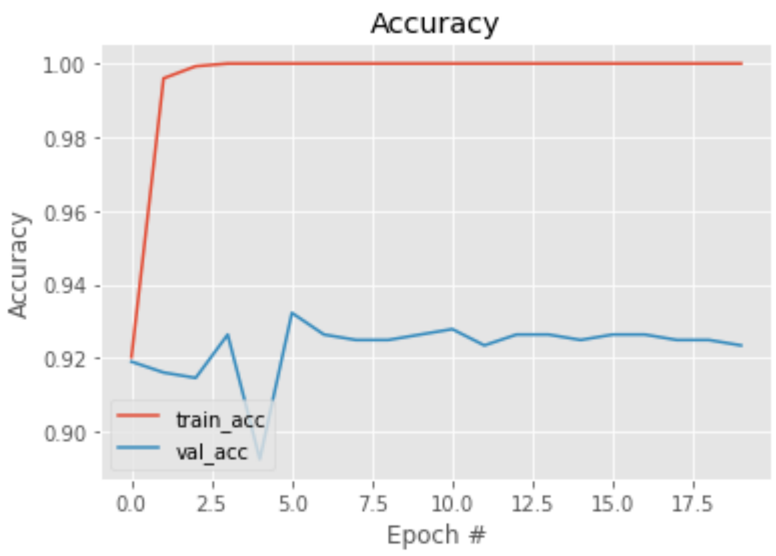
Out[16]:
```
[0.27323681116104126, 0.923416793346405]
```

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: 92.29013234376907
Avg Val Loss: 23.943122550845146
```