Inception_v3 Model

Action Classes - 20

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 inception_v3
```

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

```
Num GPUs Available:  1

2022-08-25 18:43:27.092055: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing NUMA node zero
2022-08-25 18:43:27.202297: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing NUMA node zero
2022-08-25 18:43:27.202615: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing 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.
```

In [4]:

```
image_size = (299,299)
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 5118 images belonging to 20 classes.
Found 1270 images belonging to 20 classes.
Found 1270 images belonging to 20 classes.
```

In [5]:

```
train_batches.class_indices
```

Out[5]:

```
{'ApplyLipstick': 0,
 'Archery': 1,
 'BabyCrawling': 2,
 'Basketball': 3,
 'Biking': 4,
 'Diving': 5,
 'Fencing': 6,
 'IceDancing': 7,
 'Kayaking': 8,
 'MilitaryParade': 9,
 'PizzaTossing': 10,
 'PullUps': 11,
 'ShavingBeard': 12,
 'SkateBoarding': 13,
 'SumoWrestling': 14,
 'Surfing': 15,
 'TennisSwing': 16,
 'Typing': 17,
 'WritingOnBoard': 18,
 'YoYo': 19}
```

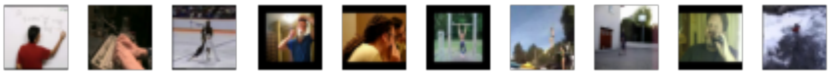
In [6]:

```
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])
```



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

Initialize model

In [8]:

```
inception_v3_model_top = inception_v3.InceptionV3(include_top=True,
                   input_shape=(299,299,3),
                   pooling='avg',
                   weights='imagenet')

for (i,layer) in enumerate(inception_v3_model_top.layers):
    print((i, layer.name, layer.output_shape))
```

```
2022-08-25 18:43:28.085783: 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 instructio
ns in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the app
ropriate compiler flags.
2022-08-25 18:43:28.086793: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing NUMA node zero
2022-08-25 18:43:28.087051: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing NUMA node zero
2022-08-25 18:43:28.087154: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing NUMA node zero
2022-08-25 18:43:29.211873: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing NUMA node zero
2022-08-25 18:43:29.212106: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing NUMA node zero
2022-08-25 18:43:29.212335: I tensorflow/stream_executor/cuda/cuda_
gpu_executor.cc:975] successful NUMA node read from SysFS had negat
ive value (-1), but there must be at least one NUMA node, so return
ing NUMA node zero
2022-08-25 18:43:29.212449: I tensorflow/core/common_runtime/gpu/gp
u_device.cc:1532] Created device /job:localhost/replica:0/task:0/de
vice: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, 299, 299, 3)])
(1, 'conv2d', (None, 149, 149, 32))
(2, 'batch_normalization', (None, 149, 149, 32))
(3, 'activation', (None, 149, 149, 32))
(4, 'conv2d_1', (None, 147, 147, 32))
(5, 'batch_normalization_1', (None, 147, 147, 32))
(6, 'activation_1', (None, 147, 147, 32))
(7, 'conv2d_2', (None, 147, 147, 64))
(8, 'batch_normalization_2', (None, 147, 147, 64))
(9, 'activation_2', (None, 147, 147, 64))
(10, 'max_pooling2d', (None, 73, 73, 64))
(11, 'conv2d_3', (None, 73, 73, 80))
(12, 'batch_normalization_3', (None, 73, 73, 80))
(13, 'activation_3', (None, 73, 73, 80))
(14, 'conv2d_4', (None, 71, 71, 192))
(15, 'batch_normalization_4', (None, 71, 71, 192))
(16, 'activation_4', (None, 71, 71, 192))
```

```
(17, 'max_pooling2d_1', (None, 35, 35, 192))
(18, 'conv2d_8', (None, 35, 35, 64))
(19, 'batch_normalization_8', (None, 35, 35, 64))
(20, 'activation_8', (None, 35, 35, 64))
(21, 'conv2d_6', (None, 35, 35, 48))
(22, 'conv2d_9', (None, 35, 35, 96))
(23, 'batch_normalization_6', (None, 35, 35, 48))
(24, 'batch_normalization_9', (None, 35, 35, 96))
(25, 'activation_6', (None, 35, 35, 48))
(26, 'activation_9', (None, 35, 35, 96))
(27, 'average_pooling2d', (None, 35, 35, 192))
(28, 'conv2d_5', (None, 35, 35, 64))
(29, 'conv2d_7', (None, 35, 35, 64))
(30, 'conv2d_10', (None, 35, 35, 96))
(31, 'conv2d_11', (None, 35, 35, 32))
(32, 'batch_normalization_5', (None, 35, 35, 64))
(33, 'batch_normalization_7', (None, 35, 35, 64))
(34, 'batch_normalization_10', (None, 35, 35, 96))
(35, 'batch_normalization_11', (None, 35, 35, 32))
(36, 'activation_5', (None, 35, 35, 64))
(37, 'activation_7', (None, 35, 35, 64))
(38, 'activation_10', (None, 35, 35, 96))
(39, 'activation_11', (None, 35, 35, 32))
(40, 'mixed0', (None, 35, 35, 256))
(41, 'conv2d_15', (None, 35, 35, 64))
(42, 'batch_normalization_15', (None, 35, 35, 64))
(43, 'activation_15', (None, 35, 35, 64))
(44, 'conv2d_13', (None, 35, 35, 48))
(45, 'conv2d_16', (None, 35, 35, 96))
(46, 'batch_normalization_13', (None, 35, 35, 48))
(47, 'batch_normalization_16', (None, 35, 35, 96))
(48, 'activation_13', (None, 35, 35, 48))
(49, 'activation_16', (None, 35, 35, 96))
(50, 'average_pooling2d_1', (None, 35, 35, 256))
(51, 'conv2d_12', (None, 35, 35, 64))
(52, 'conv2d_14', (None, 35, 35, 64))
(53, 'conv2d_17', (None, 35, 35, 96))
(54, 'conv2d_18', (None, 35, 35, 64))
(55, 'batch_normalization_12', (None, 35, 35, 64))
(56, 'batch_normalization_14', (None, 35, 35, 64))
(57, 'batch_normalization_17', (None, 35, 35, 96))
(58, 'batch_normalization_18', (None, 35, 35, 64))
(59, 'activation_12', (None, 35, 35, 64))
(60, 'activation_14', (None, 35, 35, 64))
(61, 'activation_17', (None, 35, 35, 96))
(62, 'activation_18', (None, 35, 35, 64))
(63, 'mixed1', (None, 35, 35, 288))
(64, 'conv2d_22', (None, 35, 35, 64))
(65, 'batch_normalization_22', (None, 35, 35, 64))
(66, 'activation_22', (None, 35, 35, 64))
(67, 'conv2d_20', (None, 35, 35, 48))
(68, 'conv2d_23', (None, 35, 35, 96))
(69, 'batch_normalization_20', (None, 35, 35, 48))
(70, 'batch_normalization_23', (None, 35, 35, 96))
(71, 'activation_20', (None, 35, 35, 48))
(72, 'activation_23', (None, 35, 35, 96))
(73, 'average_pooling2d_2', (None, 35, 35, 288))
(74, 'conv2d_19', (None, 35, 35, 64))
(75, 'conv2d_21', (None, 35, 35, 64))
(76, 'conv2d_24', (None, 35, 35, 96))
```

```
(77, 'conv2d_25', (None, 35, 35, 64))
(78, 'batch_normalization_19', (None, 35, 35, 64))
(79, 'batch_normalization_21', (None, 35, 35, 64))
(80, 'batch_normalization_24', (None, 35, 35, 96))
(81, 'batch_normalization_25', (None, 35, 35, 64))
(82, 'activation_19', (None, 35, 35, 64))
(83, 'activation_21', (None, 35, 35, 64))
(84, 'activation_24', (None, 35, 35, 96))
(85, 'activation_25', (None, 35, 35, 64))
(86, 'mixed2', (None, 35, 35, 288))
(87, 'conv2d_27', (None, 35, 35, 64))
(88, 'batch_normalization_27', (None, 35, 35, 64))
(89, 'activation_27', (None, 35, 35, 64))
(90, 'conv2d_28', (None, 35, 35, 96))
(91, 'batch_normalization_28', (None, 35, 35, 96))
(92, 'activation_28', (None, 35, 35, 96))
(93, 'conv2d_26', (None, 17, 17, 384))
(94, 'conv2d_29', (None, 17, 17, 96))
(95, 'batch_normalization_26', (None, 17, 17, 384))
(96, 'batch_normalization_29', (None, 17, 17, 96))
(97, 'activation_26', (None, 17, 17, 384))
(98, 'activation_29', (None, 17, 17, 96))
(99, 'max_pooling2d_2', (None, 17, 17, 288))
(100, 'mixed3', (None, 17, 17, 768))
(101, 'conv2d_34', (None, 17, 17, 128))
(102, 'batch_normalization_34', (None, 17, 17, 128))
(103, 'activation_34', (None, 17, 17, 128))
(104, 'conv2d_35', (None, 17, 17, 128))
(105, 'batch_normalization_35', (None, 17, 17, 128))
(106, 'activation_35', (None, 17, 17, 128))
(107, 'conv2d_31', (None, 17, 17, 128))
(108, 'conv2d_36', (None, 17, 17, 128))
(109, 'batch_normalization_31', (None, 17, 17, 128))
(110, 'batch_normalization_36', (None, 17, 17, 128))
(111, 'activation_31', (None, 17, 17, 128))
(112, 'activation_36', (None, 17, 17, 128))
(113, 'conv2d_32', (None, 17, 17, 128))
(114, 'conv2d_37', (None, 17, 17, 128))
(115, 'batch_normalization_32', (None, 17, 17, 128))
(116, 'batch_normalization_37', (None, 17, 17, 128))
(117, 'activation_32', (None, 17, 17, 128))
(118, 'activation_37', (None, 17, 17, 128))
(119, 'average_pooling2d_3', (None, 17, 17, 768))
(120, 'conv2d_30', (None, 17, 17, 192))
(121, 'conv2d_33', (None, 17, 17, 192))
(122, 'conv2d_38', (None, 17, 17, 192))
(123, 'conv2d_39', (None, 17, 17, 192))
(124, 'batch_normalization_30', (None, 17, 17, 192))
(125, 'batch_normalization_33', (None, 17, 17, 192))
(126, 'batch_normalization_38', (None, 17, 17, 192))
(127, 'batch_normalization_39', (None, 17, 17, 192))
(128, 'activation_30', (None, 17, 17, 192))
(129, 'activation_33', (None, 17, 17, 192))
(130, 'activation_38', (None, 17, 17, 192))
(131, 'activation_39', (None, 17, 17, 192))
(132, 'mixed4', (None, 17, 17, 768))
(133, 'conv2d_44', (None, 17, 17, 160))
(134, 'batch_normalization_44', (None, 17, 17, 160))
(135, 'activation_44', (None, 17, 17, 160))
(136, 'conv2d_45', (None, 17, 17, 160))
```

```
(137, 'batch_normalization_45', (None, 17, 17, 160))
(138, 'activation_45', (None, 17, 17, 160))
(139, 'conv2d_41', (None, 17, 17, 160))
(140, 'conv2d_46', (None, 17, 17, 160))
(141, 'batch_normalization_41', (None, 17, 17, 160))
(142, 'batch_normalization_46', (None, 17, 17, 160))
(143, 'activation_41', (None, 17, 17, 160))
(144, 'activation_46', (None, 17, 17, 160))
(145, 'conv2d_42', (None, 17, 17, 160))
(146, 'conv2d_47', (None, 17, 17, 160))
(147, 'batch_normalization_42', (None, 17, 17, 160))
(148, 'batch_normalization_47', (None, 17, 17, 160))
(149, 'activation_42', (None, 17, 17, 160))
(150, 'activation_47', (None, 17, 17, 160))
(151, 'average_pooling2d_4', (None, 17, 17, 768))
(152, 'conv2d_40', (None, 17, 17, 192))
(153, 'conv2d_43', (None, 17, 17, 192))
(154, 'conv2d_48', (None, 17, 17, 192))
(155, 'conv2d_49', (None, 17, 17, 192))
(156, 'batch_normalization_40', (None, 17, 17, 192))
(157, 'batch_normalization_43', (None, 17, 17, 192))
(158, 'batch_normalization_48', (None, 17, 17, 192))
(159, 'batch_normalization_49', (None, 17, 17, 192))
(160, 'activation_40', (None, 17, 17, 192))
(161, 'activation_43', (None, 17, 17, 192))
(162, 'activation_48', (None, 17, 17, 192))
(163, 'activation_49', (None, 17, 17, 192))
(164, 'mixed5', (None, 17, 17, 768))
(165, 'conv2d_54', (None, 17, 17, 160))
(166, 'batch_normalization_54', (None, 17, 17, 160))
(167, 'activation_54', (None, 17, 17, 160))
(168, 'conv2d_55', (None, 17, 17, 160))
(169, 'batch_normalization_55', (None, 17, 17, 160))
(170, 'activation_55', (None, 17, 17, 160))
(171, 'conv2d_51', (None, 17, 17, 160))
(172, 'conv2d_56', (None, 17, 17, 160))
(173, 'batch_normalization_51', (None, 17, 17, 160))
(174, 'batch_normalization_56', (None, 17, 17, 160))
(175, 'activation_51', (None, 17, 17, 160))
(176, 'activation_56', (None, 17, 17, 160))
(177, 'conv2d_52', (None, 17, 17, 160))
(178, 'conv2d_57', (None, 17, 17, 160))
(179, 'batch_normalization_52', (None, 17, 17, 160))
(180, 'batch_normalization_57', (None, 17, 17, 160))
(181, 'activation_52', (None, 17, 17, 160))
(182, 'activation_57', (None, 17, 17, 160))
(183, 'average_pooling2d_5', (None, 17, 17, 768))
(184, 'conv2d_50', (None, 17, 17, 192))
(185, 'conv2d_53', (None, 17, 17, 192))
(186, 'conv2d_58', (None, 17, 17, 192))
(187, 'conv2d_59', (None, 17, 17, 192))
(188, 'batch_normalization_50', (None, 17, 17, 192))
(189, 'batch_normalization_53', (None, 17, 17, 192))
(190, 'batch_normalization_58', (None, 17, 17, 192))
(191, 'batch_normalization_59', (None, 17, 17, 192))
(192, 'activation_50', (None, 17, 17, 192))
(193, 'activation_53', (None, 17, 17, 192))
(194, 'activation_58', (None, 17, 17, 192))
(195, 'activation_59', (None, 17, 17, 192))
(196, 'mixed6', (None, 17, 17, 768))
```

```
(197, 'conv2d_64', (None, 17, 17, 192))
(198, 'batch_normalization_64', (None, 17, 17, 192))
(199, 'activation_64', (None, 17, 17, 192))
(200, 'conv2d_65', (None, 17, 17, 192))
(201, 'batch_normalization_65', (None, 17, 17, 192))
(202, 'activation_65', (None, 17, 17, 192))
(203, 'conv2d_61', (None, 17, 17, 192))
(204, 'conv2d_66', (None, 17, 17, 192))
(205, 'batch_normalization_61', (None, 17, 17, 192))
(206, 'batch_normalization_66', (None, 17, 17, 192))
(207, 'activation_61', (None, 17, 17, 192))
(208, 'activation_66', (None, 17, 17, 192))
(209, 'conv2d_62', (None, 17, 17, 192))
(210, 'conv2d_67', (None, 17, 17, 192))
(211, 'batch_normalization_62', (None, 17, 17, 192))
(212, 'batch_normalization_67', (None, 17, 17, 192))
(213, 'activation_62', (None, 17, 17, 192))
(214, 'activation_67', (None, 17, 17, 192))
(215, 'average_pooling2d_6', (None, 17, 17, 768))
(216, 'conv2d_60', (None, 17, 17, 192))
(217, 'conv2d_63', (None, 17, 17, 192))
(218, 'conv2d_68', (None, 17, 17, 192))
(219, 'conv2d_69', (None, 17, 17, 192))
(220, 'batch_normalization_60', (None, 17, 17, 192))
(221, 'batch_normalization_63', (None, 17, 17, 192))
(222, 'batch_normalization_68', (None, 17, 17, 192))
(223, 'batch_normalization_69', (None, 17, 17, 192))
(224, 'activation_60', (None, 17, 17, 192))
(225, 'activation_63', (None, 17, 17, 192))
(226, 'activation_68', (None, 17, 17, 192))
(227, 'activation_69', (None, 17, 17, 192))
(228, 'mixed7', (None, 17, 17, 768))
(229, 'conv2d_72', (None, 17, 17, 192))
(230, 'batch_normalization_72', (None, 17, 17, 192))
(231, 'activation_72', (None, 17, 17, 192))
(232, 'conv2d_73', (None, 17, 17, 192))
(233, 'batch_normalization_73', (None, 17, 17, 192))
(234, 'activation_73', (None, 17, 17, 192))
(235, 'conv2d_70', (None, 17, 17, 192))
(236, 'conv2d_74', (None, 17, 17, 192))
(237, 'batch_normalization_70', (None, 17, 17, 192))
(238, 'batch_normalization_74', (None, 17, 17, 192))
(239, 'activation_70', (None, 17, 17, 192))
(240, 'activation_74', (None, 17, 17, 192))
(241, 'conv2d_71', (None, 8, 8, 320))
(242, 'conv2d_75', (None, 8, 8, 192))
(243, 'batch_normalization_71', (None, 8, 8, 320))
(244, 'batch_normalization_75', (None, 8, 8, 192))
(245, 'activation_71', (None, 8, 8, 320))
(246, 'activation_75', (None, 8, 8, 192))
(247, 'max_pooling2d_3', (None, 8, 8, 768))
(248, 'mixed8', (None, 8, 8, 1280))
(249, 'conv2d_80', (None, 8, 8, 448))
(250, 'batch_normalization_80', (None, 8, 8, 448))
(251, 'activation_80', (None, 8, 8, 448))
(252, 'conv2d_77', (None, 8, 8, 384))
(253, 'conv2d_81', (None, 8, 8, 384))
(254, 'batch_normalization_77', (None, 8, 8, 384))
(255, 'batch_normalization_81', (None, 8, 8, 384))
(256, 'activation_77', (None, 8, 8, 384))
```

```
(257, 'activation_81', (None, 8, 8, 384))
(258, 'conv2d_78', (None, 8, 8, 384))
(259, 'conv2d_79', (None, 8, 8, 384))
(260, 'conv2d_82', (None, 8, 8, 384))
(261, 'conv2d_83', (None, 8, 8, 384))
(262, 'average_pooling2d_7', (None, 8, 8, 1280))
(263, 'conv2d_76', (None, 8, 8, 320))
(264, 'batch_normalization_78', (None, 8, 8, 384))
(265, 'batch_normalization_79', (None, 8, 8, 384))
(266, 'batch_normalization_82', (None, 8, 8, 384))
(267, 'batch_normalization_83', (None, 8, 8, 384))
(268, 'conv2d_84', (None, 8, 8, 192))
(269, 'batch_normalization_76', (None, 8, 8, 320))
(270, 'activation_78', (None, 8, 8, 384))
(271, 'activation_79', (None, 8, 8, 384))
(272, 'activation_82', (None, 8, 8, 384))
(273, 'activation_83', (None, 8, 8, 384))
(274, 'batch_normalization_84', (None, 8, 8, 192))
(275, 'activation_76', (None, 8, 8, 320))
(276, 'mixed9_0', (None, 8, 8, 768))
(277, 'concatenate', (None, 8, 8, 768))
(278, 'activation_84', (None, 8, 8, 192))
(279, 'mixed9', (None, 8, 8, 2048))
(280, 'conv2d_89', (None, 8, 8, 448))
(281, 'batch_normalization_89', (None, 8, 8, 448))
(282, 'activation_89', (None, 8, 8, 448))
(283, 'conv2d_86', (None, 8, 8, 384))
(284, 'conv2d_90', (None, 8, 8, 384))
(285, 'batch_normalization_86', (None, 8, 8, 384))
(286, 'batch_normalization_90', (None, 8, 8, 384))
(287, 'activation_86', (None, 8, 8, 384))
(288, 'activation_90', (None, 8, 8, 384))
(289, 'conv2d_87', (None, 8, 8, 384))
(290, 'conv2d_88', (None, 8, 8, 384))
(291, 'conv2d_91', (None, 8, 8, 384))
(292, 'conv2d_92', (None, 8, 8, 384))
(293, 'average_pooling2d_8', (None, 8, 8, 2048))
(294, 'conv2d_85', (None, 8, 8, 320))
(295, 'batch_normalization_87', (None, 8, 8, 384))
(296, 'batch_normalization_88', (None, 8, 8, 384))
(297, 'batch_normalization_91', (None, 8, 8, 384))
(298, 'batch_normalization_92', (None, 8, 8, 384))
(299, 'conv2d_93', (None, 8, 8, 192))
(300, 'batch_normalization_85', (None, 8, 8, 320))
(301, 'activation_87', (None, 8, 8, 384))
(302, 'activation_88', (None, 8, 8, 384))
(303, 'activation_91', (None, 8, 8, 384))
(304, 'activation_92', (None, 8, 8, 384))
(305, 'batch_normalization_93', (None, 8, 8, 192))
(306, 'activation_85', (None, 8, 8, 320))
(307, 'mixed9_1', (None, 8, 8, 768))
(308, 'concatenate_1', (None, 8, 8, 768))
(309, 'activation_93', (None, 8, 8, 192))
(310, 'mixed10', (None, 8, 8, 2048))
(311, 'avg_pool', (None, 2048))
```

In [9]:

```
inception_v3_model = inception_v3.InceptionV3(include_top=False,
                   input_shape=(299,299,3),
                   pooling='avg',classes=20,
                   weights='imagenet')

for (i,layer) in enumerate(inception_v3_model.layers):
    layer.trainable = False
    print((i, layer.name, layer.output_shape))
```

```
(0, 'input_2', [(None, 299, 299, 3)])
(1, 'conv2d_94', (None, 149, 149, 32))
(2, 'batch_normalization_94', (None, 149, 149, 32))
(3, 'activation_94', (None, 149, 149, 32))
(4, 'conv2d_95', (None, 147, 147, 32))
(5, 'batch_normalization_95', (None, 147, 147, 32))
(6, 'activation_95', (None, 147, 147, 32))
(7, 'conv2d_96', (None, 147, 147, 64))
(8, 'batch_normalization_96', (None, 147, 147, 64))
(9, 'activation_96', (None, 147, 147, 64))
(10, 'max_pooling2d_4', (None, 73, 73, 64))
(11, 'conv2d_97', (None, 73, 73, 80))
(12, 'batch_normalization_97', (None, 73, 73, 80))
(13, 'activation_97', (None, 73, 73, 80))
(14, 'conv2d_98', (None, 71, 71, 192))
(15, 'batch_normalization_98', (None, 71, 71, 192))
(16, 'activation_98', (None, 71, 71, 192))
(17, 'max_pooling2d_5', (None, 35, 35, 192))
(18, 'conv2d_102', (None, 35, 35, 64))
(19, 'batch_normalization_102', (None, 35, 35, 64))
```

In [10]:

```
model = models.Sequential()

dense_layer_1 = Dense(32, activation='relu')
prediction_layer = Dense(20, activation='softmax')

model.add(inception_v3_model)
model.add(dense_layer_1)
model.add(prediction_layer)

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 2048)              21802784

 dense (Dense)               (None, 32)                65568

 dense_1 (Dense)             (None, 20)                660

=================================================================
Total params: 21,869,012
Trainable params: 66,228
Non-trainable params: 21,802,784
_________________________________________________________________
```

In [11]:

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

In [12]:

```
model.save("./models/action-class-20-model-inception_v3_model.h5")
```

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

```
Epoch 1/20

2022-08-25 18:43:37.777515: I tensorflow/stream_executor/cuda/cuda_
dnn.cc:384] Loaded cuDNN version 8401
2022-08-25 18:43:39.291845: I tensorflow/core/platform/default/subp
rocess.cc:304] Start cannot spawn child process: No such file or di
rectory

512/512 [==============================] - 67s 116ms/step - loss:
0.9075 - accuracy: 0.7685 - val_loss: 0.4676 - val_accuracy: 0.8614
Epoch 2/20
512/512 [==============================] - 58s 113ms/step - loss:
0.1785 - accuracy: 0.9642 - val_loss: 0.4016 - val_accuracy: 0.8795
Epoch 3/20
512/512 [==============================] - 62s 121ms/step - loss:
0.0937 - accuracy: 0.9807 - val_loss: 0.3684 - val_accuracy: 0.8724
Epoch 4/20
512/512 [==============================] - 64s 125ms/step - loss:
0.0547 - accuracy: 0.9898 - val_loss: 0.3751 - val_accuracy: 0.8803
Epoch 5/20
512/512 [==============================] - 71s 138ms/step - loss:
0.0364 - accuracy: 0.9947 - val_loss: 0.4237 - val_accuracy: 0.8803
Epoch 6/20
512/512 [==============================] - 95s 185ms/step - loss:
0.0210 - accuracy: 0.9986 - val_loss: 0.3898 - val_accuracy: 0.8772
Epoch 7/20
512/512 [==============================] - 110s 215ms/step - loss:
0.0176 - accuracy: 0.9980 - val_loss: 0.4505 - val_accuracy: 0.8709
Epoch 8/20
512/512 [==============================] - 110s 213ms/step - loss:
0.0105 - accuracy: 0.9986 - val_loss: 0.3452 - val_accuracy: 0.9071
Epoch 9/20
512/512 [==============================] - 125s 245ms/step - loss:
0.0095 - accuracy: 0.9984 - val_loss: 0.3736 - val_accuracy: 0.8890
Epoch 10/20
512/512 [==============================] - 124s 242ms/step - loss:
0.0039 - accuracy: 1.0000 - val_loss: 0.3875 - val_accuracy: 0.8945
Epoch 11/20
512/512 [==============================] - 122s 238ms/step - loss:
0.0028 - accuracy: 1.0000 - val_loss: 0.3508 - val_accuracy: 0.9055
Epoch 12/20
512/512 [==============================] - 130s 254ms/step - loss:
0.0022 - accuracy: 1.0000 - val_loss: 0.3704 - val_accuracy: 0.8992
Epoch 13/20
512/512 [==============================] - 130s 254ms/step - loss:
0.0017 - accuracy: 1.0000 - val_loss: 0.3765 - val_accuracy: 0.8984
Epoch 14/20
512/512 [==============================] - 134s 262ms/step - loss:
0.0012 - accuracy: 1.0000 - val_loss: 0.4296 - val_accuracy: 0.8921
Epoch 15/20
512/512 [==============================] - 124s 243ms/step - loss:
9.5946e-04 - accuracy: 1.0000 - val_loss: 0.3828 - val_accuracy: 0.
9063
Epoch 16/20
512/512 [==============================] - 136s 266ms/step - loss:
0.0290 - accuracy: 0.9920 - val_loss: 0.5124 - val_accuracy: 0.8787
Epoch 17/20
512/512 [==============================] - 134s 262ms/step - loss:
```

```
0.0017 - accuracy: 1.0000 - val_loss: 0.4760 - val_accuracy: 0.8866
Epoch 18/20
512/512 [==============================] - 140s 273ms/step - loss:
6.9804e-04 - accuracy: 1.0000 - val_loss: 0.4257 - val_accuracy: 0.
8945
Epoch 19/20
512/512 [==============================] - 127s 247ms/step - loss:
5.4271e-04 - accuracy: 1.0000 - val_loss: 0.4470 - val_accuracy: 0.
8921
Epoch 20/20
512/512 [==============================] - 128s 249ms/step - loss:
4.4166e-04 - accuracy: 1.0000 - val_loss: 0.4414 - val_accuracy: 0.
8984
```

In [14]:

```
model.save("./models/action-class-20-trained-inception_v3_model.h5")
```

Evaluate and Predict

In [15]:

```
model = models.load_model("./models/action-class-20-trained-inception
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 inception_v3 (Functional)   (None, 2048)              21802784  
                                                                 
 dense (Dense)               (None, 32)                65568     
                                                                 
 dense_1 (Dense)             (None, 20)                660       
                                                                 
=================================================================
Total params: 21,869,012
Trainable params: 66,228
Non-trainable params: 21,802,784
_________________________________________________________________
```

In [16]:

```
model.evaluate(test_batches)
```

```
127/127 [==============================] - 20s 132ms/step - loss:
0.4414 - accuracy: 0.8984
```

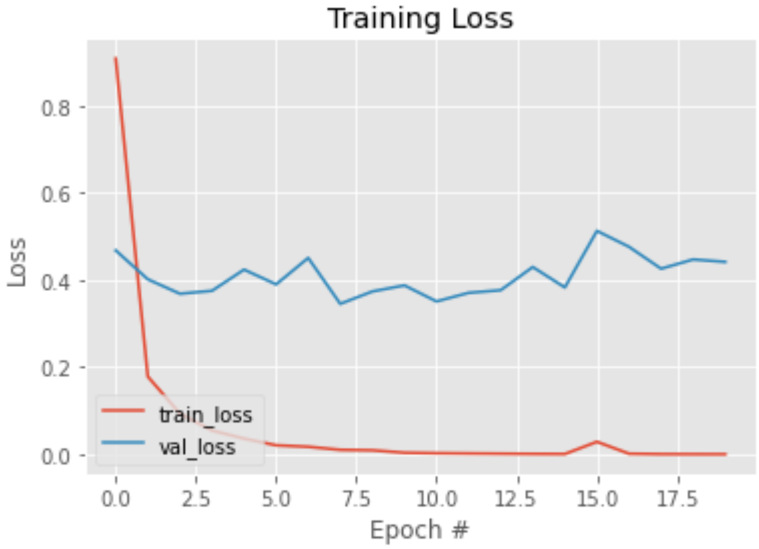
Out[16]:

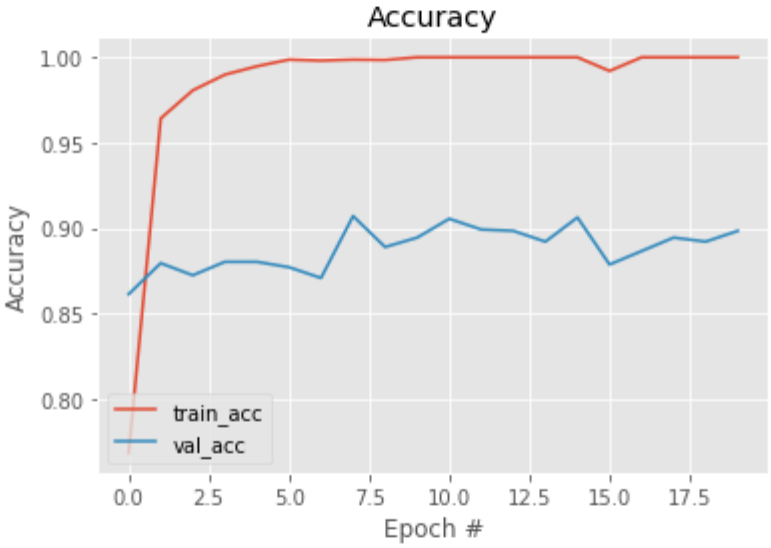
```
[0.44139841198921204, 0.8984252214431763]
```

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_ac
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))
print("Avg Val Loss: " + str(sum(fit.history["val_loss"])/20))
```

```
Avg Val Acc: 0.8882283449172974
Avg Val Loss: 0.40977955162525176
```