

Ved Nigam

```
In [ ]: # Importing required libraries
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
import seaborn as sb
import matplotlib.pyplot as plt

from tensorflow import keras
from keras import layers, models, Model
from keras.models import Sequential

# Creating a tensorflow device using mac gpu
tf.config.list_physical_devices()
with tf.device('/GPU'):
    a = tf.random.normal(shape=(2,), dtype= tf.float32)
    b = tf.nn.relu(a)
```

```
In [ ]: # loaded data in
path = "./data/"
data_dir = tf.keras.utils.image_dataset_from_directory(directory=path)
```

Found 1610 files belonging to 13 classes.

```
In [ ]: # Splitting data into train and test along with some data exploration
batch_size = 32
img_height = 180
img_width = 180

train_ds = tf.keras.utils.image_dataset_from_directory(
    path,
    validation_split=0.2,
    subset="training",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)

val_ds = tf.keras.utils.image_dataset_from_directory(
    path,
    validation_split=0.2,
    subset="validation",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)

class_names = train_ds.class_names
print(class_names)
```

```

Found 1610 files belonging to 13 classes.
Using 1288 files for training.
Found 1610 files belonging to 13 classes.
Using 322 files for validation.
['american_football', 'baseball', 'basketball', 'billiard_ball', 'bowling_ball', 'cricket_ball', 'football', 'golf_ball', 'rugby_ball', 'shuttlecock', 'table_tennis_ball', 'tennis_ball', 'volleyball']

```

```
In [ ]: # Need to make graph that shows distribution of data
```

```
In [ ]: # Configure dataset for performance
AUTOTUNE = tf.data.AUTOTUNE

train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

```
In [ ]: # Showing some sample images
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(class_names[labels[i]])
        plt.axis("off")
```

```

2023-04-22 16:01:50.299219: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indica
te an error and you can ignore this message): INVALID_ARGUMENT: You must feed
a value for placeholder tensor 'Placeholder/_0' with dtype string and shape
[1288]

```

```
[[{{node Placeholder/_0}}]]
```

```

2023-04-22 16:01:50.299584: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indica
te an error and you can ignore this message): INVALID_ARGUMENT: You must feed
a value for placeholder tensor 'Placeholder/_4' with dtype int32 and shape [1
288]

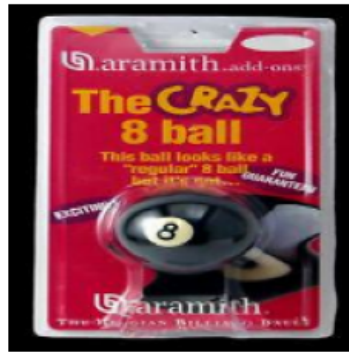
```

```
[[{{node Placeholder/_4}}]]
```

basketball



billiard\_ball



billiard\_ball



table\_tennis\_ball



football



billiard\_ball



rugby\_ball



bowling\_ball



cricket\_ball



```
In [ ]: # Defining model
num_classes = len(class_names)

model = Sequential([
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

```
In [ ]: # Compiling model
model.compile(optimizer='adam',
```

```
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)  
metrics=['accuracy'])  
  
model.summary()
```

Model: "sequential\_9"

Layer (type)	Output Shape	Param #
rescaling_4 (Rescaling)	(None, 180, 180, 3)	0
conv2d_27 (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d_22 (MaxPooling2D)	(None, 90, 90, 16)	0
conv2d_28 (Conv2D)	(None, 90, 90, 32)	4640
max_pooling2d_23 (MaxPooling2D)	(None, 45, 45, 32)	0
conv2d_29 (Conv2D)	(None, 45, 45, 64)	18496
max_pooling2d_24 (MaxPooling2D)	(None, 22, 22, 64)	0
flatten_12 (Flatten)	(None, 30976)	0
dense_27 (Dense)	(None, 128)	3965056
dense_28 (Dense)	(None, 13)	1677
Total params: 3,990,317		
Trainable params: 3,990,317		
Non-trainable params: 0		

Layer (type)	Output Shape	Param #
rescaling_4 (Rescaling)	(None, 180, 180, 3)	0
conv2d_27 (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d_22 (MaxPooling2D)	(None, 90, 90, 16)	0
conv2d_28 (Conv2D)	(None, 90, 90, 32)	4640
max_pooling2d_23 (MaxPooling2D)	(None, 45, 45, 32)	0
conv2d_29 (Conv2D)	(None, 45, 45, 64)	18496
max_pooling2d_24 (MaxPooling2D)	(None, 22, 22, 64)	0
flatten_12 (Flatten)	(None, 30976)	0
dense_27 (Dense)	(None, 128)	3965056

dense\_28 (Dense)

(None, 13)

1677

```
=====
Total params: 3,990,317
Trainable params: 3,990,317
Non-trainable params: 0
```

```
In [ ]: # Training model
epochs=10
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs
)

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(epochs)

# Showing the model's performance

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```

Epoch 1/10

```
2023-04-22 16:01:52.292196: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indica
te an error and you can ignore this message): INVALID_ARGUMENT: You must feed
a value for placeholder tensor 'Placeholder/_4' with dtype int32 and shape [1
288]
```

```
[[{{node Placeholder/_4}}]]
```

```
2023-04-22 16:01:52.292534: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indica
te an error and you can ignore this message): INVALID_ARGUMENT: You must feed
a value for placeholder tensor 'Placeholder/_4' with dtype int32 and shape [1
288]
```

```
[[{{node Placeholder/_4}}]]
```

```
2023-04-22 16:01:52.612208: I tensorflow/core/grappler/optimizers/custom_grap
h_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
```

41/41 [=====] - ETA: 0s - loss: 2.6153 - accuracy: 0.1273

2023-04-22 16:01:55.356417: I tensorflow/core/common\_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID\_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/\_0' with dtype string and shape [322]

[[{{node Placeholder/\_0}}]]

2023-04-22 16:01:55.356685: I tensorflow/core/common\_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID\_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/\_0' with dtype string and shape [322]

[[{{node Placeholder/\_0}}]]

2023-04-22 16:01:55.402411: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.

41/41 [=====] - 4s 67ms/step - loss: 2.6153 - accuracy: 0.1273 - val\_loss: 2.4984 - val\_accuracy: 0.0901

Epoch 2/10

41/41 [=====] - 1s 36ms/step - loss: 2.3549 - accuracy: 0.2019 - val\_loss: 2.3198 - val\_accuracy: 0.2267

Epoch 3/10

41/41 [=====] - 1s 35ms/step - loss: 2.0548 - accuracy: 0.3292 - val\_loss: 2.2828 - val\_accuracy: 0.2981

Epoch 4/10

41/41 [=====] - 1s 35ms/step - loss: 1.6280 - accuracy: 0.4969 - val\_loss: 2.5008 - val\_accuracy: 0.2671

Epoch 5/10

41/41 [=====] - 1s 35ms/step - loss: 1.1359 - accuracy: 0.6405 - val\_loss: 2.5452 - val\_accuracy: 0.3043

Epoch 6/10

41/41 [=====] - 1s 34ms/step - loss: 0.6876 - accuracy: 0.7849 - val\_loss: 2.9470 - val\_accuracy: 0.2857

Epoch 7/10

41/41 [=====] - 1s 34ms/step - loss: 0.4866 - accuracy: 0.8556 - val\_loss: 2.9360 - val\_accuracy: 0.3230

Epoch 8/10

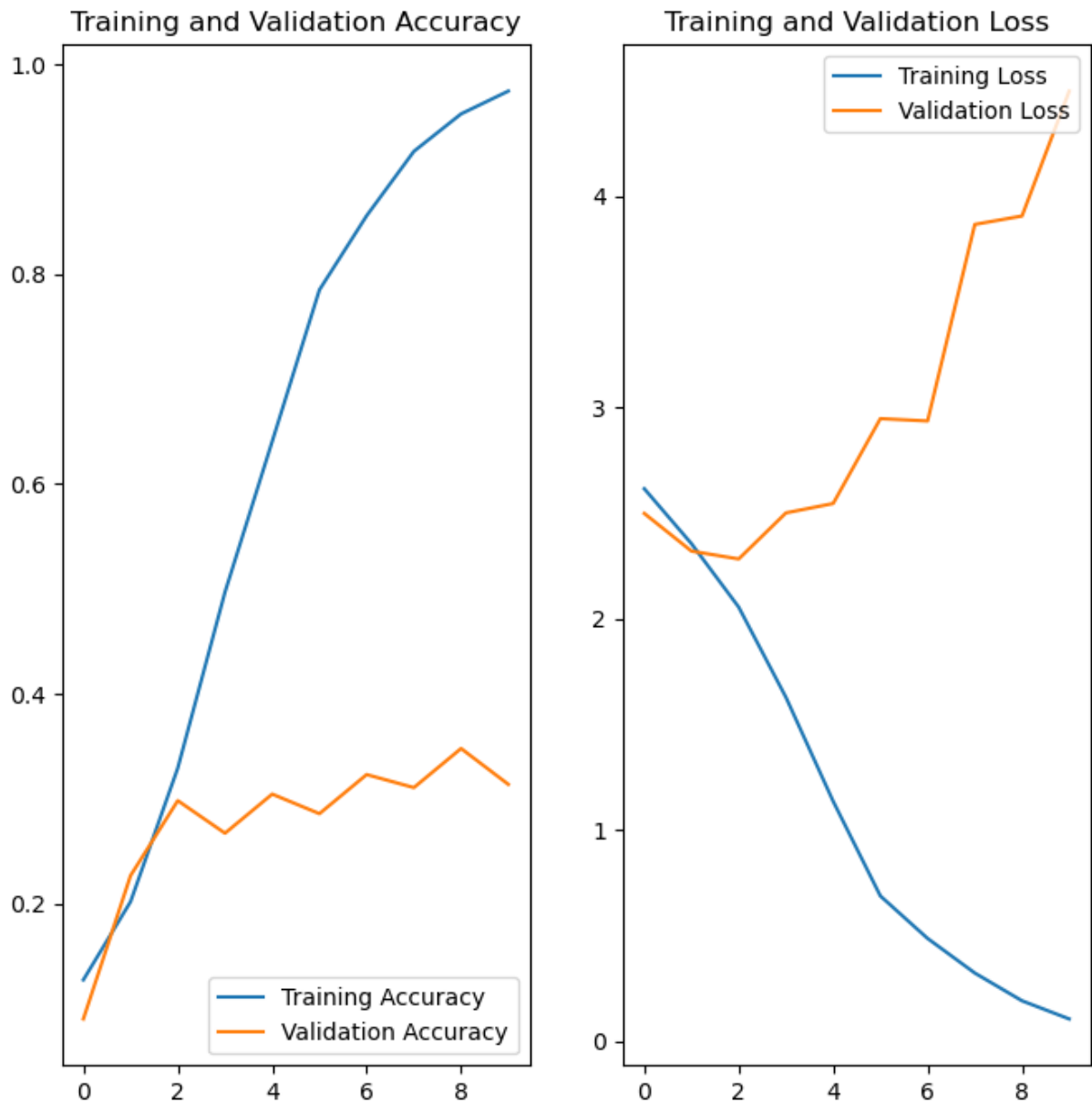
41/41 [=====] - 1s 34ms/step - loss: 0.3228 - accuracy: 0.9169 - val\_loss: 3.8663 - val\_accuracy: 0.3106

Epoch 9/10

41/41 [=====] - 1s 34ms/step - loss: 0.1914 - accuracy: 0.9526 - val\_loss: 3.9063 - val\_accuracy: 0.3478

Epoch 10/10

41/41 [=====] - 1s 34ms/step - loss: 0.1054 - accuracy: 0.9744 - val\_loss: 4.4975 - val\_accuracy: 0.3137



```
In [ ]: # Testing the model on images
basketball_path = "./data/basketball/basketball_140.jpg"

basketball_img = tf.keras.utils.load_img(
    basketball_path, target_size=(img_height, img_width)
)

img_array = tf.keras.utils.img_to_array(basketball_img)
img_array = np.array([img_array])

predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])

print(
    "This image most likely belongs to {} with a {:.2f} percent confidence."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)
```



```

shuttlecock_path = "./data/shuttlecock/shuttlecock_183.jpg"

shuttlecock_img = tf.keras.utils.load_img(
    shuttlecock_path, target_size=(img_height, img_width)
)

img_array = tf.keras.utils.img_to_array(shuttlecock_img)
img_array = np.array([img_array])

predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])

print(
    "This image most likely belongs to {} with a {:.2f} percent confidence."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)

billiard_ball_path = "./data/billiard_ball/billiard_ball_1016.jpg"

billiard_ball_img = tf.keras.utils.load_img(
    billiard_ball_path, target_size=(img_height, img_width)
)

img_array = tf.keras.utils.img_to_array(billiard_ball_img)
img_array = np.array([img_array])

predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])

print(
    "This image most likely belongs to {} with a {:.2f} percent confidence."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)

```

WARNING:tensorflow:5 out of the last 13 calls to <function Model.make\_predict\_function.<locals>.predict\_function at 0x7fa932bc3b50> 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](https://www.tensorflow.org/guide/function#controlling_retracing) and [https://www.tensorflow.org/api\\_docs/python/tf/function](https://www.tensorflow.org/api_docs/python/tf/function) for more details.

1/1 [=====] - 0s 110ms/step

This image most likely belongs to basketball with a 99.70 percent confidence.

1/1 [=====] - 0s 17ms/step

This image most likely belongs to shuttlecock with a 99.97 percent confidence.

1/1 [=====] - 0s 21ms/step

This image most likely belongs to billiard\_ball with a 99.98 percent confidence.

2023-04-22 16:02:09.447870: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.

```
In [ ]: # Building CNN model for comparison on performance
c_model = models.Sequential()
c_model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(180, 180, 3)))
c_model.add(layers.MaxPooling2D((2, 2)))
c_model.add(layers.Conv2D(64, (3, 3), activation='relu'))
c_model.add(layers.MaxPooling2D((2, 2)))
c_model.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

```
In [ ]: # Adding layers based on the amount of output classes
c_model.add(layers.Flatten())
c_model.add(layers.Dense(64, activation='relu'))
c_model.add(layers.Dense(13))

c_model.summary()
```

Model: "sequential\_10"

Layer (type)	Output Shape	Param #
conv2d_30 (Conv2D)	(None, 178, 178, 32)	896
max_pooling2d_25 (MaxPooling2D)	(None, 89, 89, 32)	0
conv2d_31 (Conv2D)	(None, 87, 87, 64)	18496
max_pooling2d_26 (MaxPooling2D)	(None, 43, 43, 64)	0
conv2d_32 (Conv2D)	(None, 41, 41, 64)	36928
flatten_13 (Flatten)	(None, 107584)	0
dense_29 (Dense)	(None, 64)	6885440
dense_30 (Dense)	(None, 13)	845
Total params: 6,942,605		
Trainable params: 6,942,605		
Non-trainable params: 0		

Layer (type)	Output Shape	Param #
conv2d_30 (Conv2D)	(None, 178, 178, 32)	896
max_pooling2d_25 (MaxPooling2D)	(None, 89, 89, 32)	0
conv2d_31 (Conv2D)	(None, 87, 87, 64)	18496
max_pooling2d_26 (MaxPooling2D)	(None, 43, 43, 64)	0
conv2d_32 (Conv2D)	(None, 41, 41, 64)	36928
flatten_13 (Flatten)	(None, 107584)	0
dense_29 (Dense)	(None, 64)	6885440
dense_30 (Dense)	(None, 13)	845
Total params: 6,942,605		
Trainable params: 6,942,605		
Non-trainable params: 0		

```
In [ ]: # Compiling model
        c_model.compile(optimizer='adam',
```

```
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metrics=['accuracy'])
```

```
history = c_model.fit(train_ds, epochs=10, validation_data=val_ds)
```

Epoch 1/10

2023-04-22 16:02:10.061959: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.  
40/41 [=====>.] - ETA: 0s - loss: 78.3346 - accuracy: 0.0748

2023-04-22 16:02:13.303489: I tensorflow/core/grappler/optimizers/custom\_graph\_optimizer\_registry.cc:114] Plugin optimizer for device\_type GPU is enabled.  
41/41 [=====] - 4s 72ms/step - loss: 76.4532 - accuracy: 0.0745 - val\_loss: 2.5696 - val\_accuracy: 0.0590

Epoch 2/10

Epoch 2/10

41/41 [=====] - 2s 52ms/step - loss: 2.4215 - accuracy: 0.1693 - val\_loss: 2.6253 - val\_accuracy: 0.1056

Epoch 3/10

41/41 [=====] - 2s 49ms/step - loss: 2.1035 - accuracy: 0.3129 - val\_loss: 2.7968 - val\_accuracy: 0.1646

Epoch 4/10

41/41 [=====] - 2s 49ms/step - loss: 1.7013 - accuracy: 0.4752 - val\_loss: 3.3264 - val\_accuracy: 0.1522

Epoch 5/10

41/41 [=====] - 2s 49ms/step - loss: 1.2083 - accuracy: 0.6623 - val\_loss: 5.5896 - val\_accuracy: 0.1398

Epoch 6/10

41/41 [=====] - 2s 49ms/step - loss: 0.9703 - accuracy: 0.7275 - val\_loss: 6.0235 - val\_accuracy: 0.1894

Epoch 7/10

41/41 [=====] - 2s 49ms/step - loss: 0.7829 - accuracy: 0.7865 - val\_loss: 5.7410 - val\_accuracy: 0.1894

Epoch 8/10

41/41 [=====] - 2s 49ms/step - loss: 0.6031 - accuracy: 0.8346 - val\_loss: 6.8587 - val\_accuracy: 0.1739

Epoch 9/10

41/41 [=====] - 2s 49ms/step - loss: 0.6945 - accuracy: 0.8253 - val\_loss: 5.5972 - val\_accuracy: 0.1739

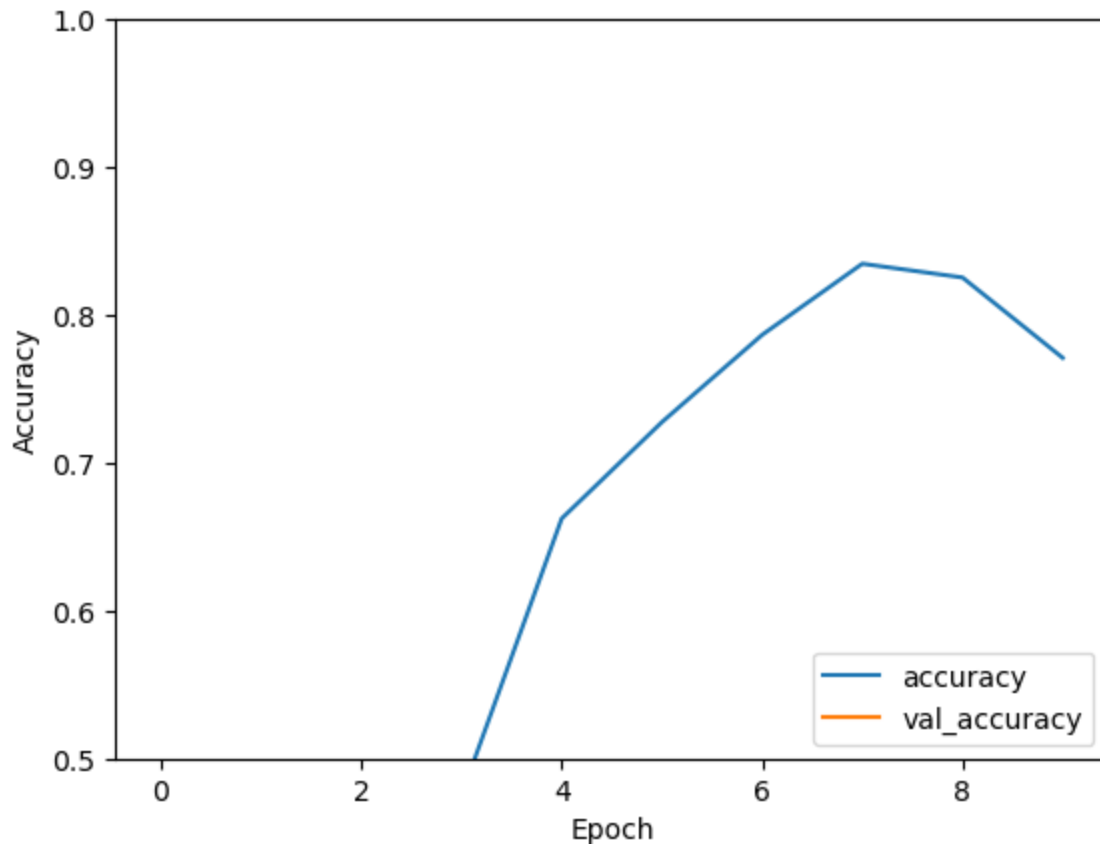
Epoch 10/10

41/41 [=====] - 2s 49ms/step - loss: 0.8480 - accuracy: 0.7710 - val\_loss: 8.2497 - val\_accuracy: 0.1584

```
In [ ]: plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.5, 1])
plt.legend(loc='lower right')
```

```
test_loss, test_acc = c_model.evaluate(val_ds)
```

11/11 [=====] - 0s 16ms/step - loss: 8.2497 - accuracy: 0.1584



```
In [ ]: _URL = 'https://storage.googleapis.com/mledu-datasets/cats_and_dogs_filtered'
path_to_zip = tf.keras.utils.get_file('cats_and_dogs.zip', origin=_URL, extr
PATH = os.path.join(os.path.dirname(path_to_zip), 'cats_and_dogs_filtered')

train_dir = os.path.join(PATH, 'train')
validation_dir = os.path.join(PATH, 'validation')

BATCH_SIZE = 32
IMG_SIZE = (160, 160)

train_dataset = tf.keras.utils.image_dataset_from_directory(train_dir,
                                                            shuffle=True,
                                                            batch_size=BATCH
                                                            image_size=IMG_S
```

Found 2000 files belonging to 2 classes.

```
In [ ]: validation_dataset = tf.keras.utils.image_dataset_from_directory(validation_
                                                shuffle=True
                                                batch_size=
                                                image_size=
```

Found 1000 files belonging to 2 classes.

```
In [ ]: class_names = train_dataset.class_names

plt.figure(figsize=(10, 10))
for images, labels in train_dataset.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
```

```
plt.imshow(images[i].numpy().astype("uint8"))
plt.title(class_names[labels[i]])
plt.axis("off")
```

2023-04-22 16:02:32.926063: I tensorflow/core/common\_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID\_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/\_4' with dtype int32 and shape [2000]

[[{{node Placeholder/\_4}}]]

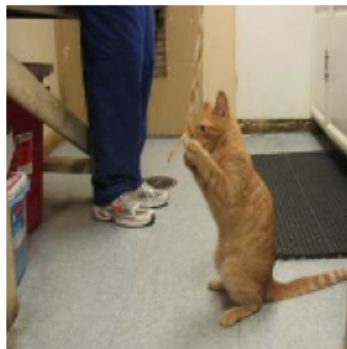
2023-04-22 16:02:32.926328: I tensorflow/core/common\_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID\_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/\_4' with dtype int32 and shape [2000]

[[{{node Placeholder/\_4}}]]

dogs



cats



dogs



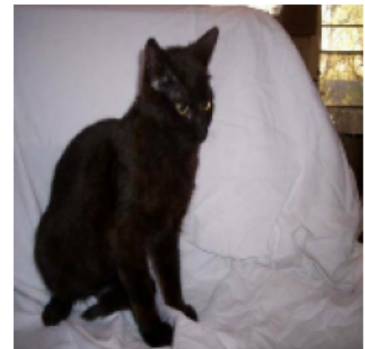
cats



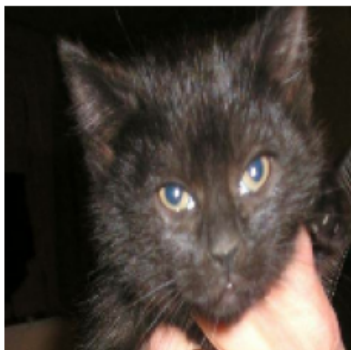
cats



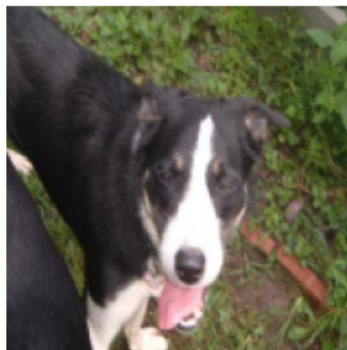
cats



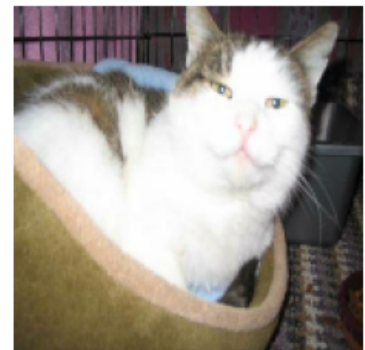
cats



dogs



cats



```
In [ ]: val_batches = tf.data.experimental.cardinality(validation_dataset)
        test_dataset = validation_dataset.take(val_batches // 5)
```

```
validation_dataset = validation_dataset.skip(val_batches // 5)
```

```
In [ ]: print('Number of validation batches: %d' % tf.data.experimental.cardinality(
print('Number of test batches: %d' % tf.data.experimental.cardinality(test_c
```

```
Number of validation batches: 26
Number of test batches: 6
```

```
In [ ]: AUTOTUNE = tf.data.AUTOTUNE

train_dataset = train_dataset.prefetch(buffer_size=AUTOTUNE)
validation_dataset = validation_dataset.prefetch(buffer_size=AUTOTUNE)
test_dataset = test_dataset.prefetch(buffer_size=AUTOTUNE)
```

```
In [ ]: # Create the base model from the pre-trained model MobileNet V2
IMG_SHAPE = IMG_SIZE + (3,)
base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SHAPE,
                                                include_top=False,
                                                weights='imagenet')
```

```
In [ ]: image_batch, label_batch = next(iter(train_dataset))
feature_batch = base_model(image_batch)
print(feature_batch.shape)
```

```
(32, 5, 5, 1280)
```

```
2023-04-22 16:02:34.253232: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indica
te an error and you can ignore this message): INVALID_ARGUMENT: You must feed
a value for placeholder tensor 'Placeholder/_0' with dtype string and shape
[2000]
```

```
[[{{node Placeholder/_0}}]]
```

```
2023-04-22 16:02:34.253502: I tensorflow/core/common_runtime/executor.cc:119
7] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indica
te an error and you can ignore this message): INVALID_ARGUMENT: You must feed
a value for placeholder tensor 'Placeholder/_0' with dtype string and shape
[2000]
```

```
[[{{node Placeholder/_0}}]]
```

```
In [ ]: base_model.trainable = False
```

```
In [ ]: # Let's take a look at the base model architecture
base_model.summary()
```

Model: "mobilenetv2\_1.00\_160"

Layer (type)	Output Shape	Param #	Connected to
input_7 (InputLayer)	[(None, 160, 160, 3)]	0	[]
Conv1 (Conv2D)	(None, 80, 80, 32)	864	['input_7[0]']
bn_Conv1 (BatchNormalization)	(None, 80, 80, 32)	128	['Conv1[0]']
Conv1_relu (ReLU)	(None, 80, 80, 32)	0	['bn_Conv1[0][0]']
expanded_conv_depthwise (DepthwiseConv2D)	(None, 80, 80, 32)	288	['Conv1_relu[0][0]']
expanded_conv_depthwise_BN (BatchNormalization)	(None, 80, 80, 32)	128	['expanded_conv_depthwise[0][0]']
expanded_conv_depthwise_relu (ReLU)	(None, 80, 80, 32)	0	['expanded_conv_depthwise_BN[0][0]']
expanded_conv_project (Conv2D)	(None, 80, 80, 16)	512	['expanded_conv_depthwise_relu[0]']
expanded_conv_project_BN (BatchNormalization)	(None, 80, 80, 16)	64	['expanded_conv_project[0][0]']
block_1_expand (Conv2D)	(None, 80, 80, 96)	1536	['expanded_conv_project_BN[0][0]']
block_1_expand_BN (BatchNormalization)	(None, 80, 80, 96)	384	['block_1_expand[0][0]']
block_1_expand_relu (ReLU)	(None, 80, 80, 96)	0	['block_1_expand_BN[0][0]']
input_7 (InputLayer)	[(None, 160, 160, 3)]	0	[]



Conv1 (Conv2D)	(None, 80, 80, 32)	864	['input_7[0][0]']
bn_Conv1 (BatchNormalization)	(None, 80, 80, 32)	128	['Conv1[0][0]']
Conv1_relu (ReLU)	(None, 80, 80, 32)	0	['bn_Conv1[0][0]']
expanded_conv_depthwise (DepthwiseConv2D)	(None, 80, 80, 32)	288	['Conv1_relu[0][0]']
expanded_conv_depthwise_BN (BatchNormalization)	(None, 80, 80, 32)	128	['expanded_conv_depthwise[0][0]']
expanded_conv_depthwise_relu (ReLU)	(None, 80, 80, 32)	0	['expanded_conv_depthwise_BN[0][0]']
expanded_conv_project (Conv2D)	(None, 80, 80, 16)	512	['expanded_conv_depthwise_relu[0][0]']
expanded_conv_project_BN (BatchNormalization)	(None, 80, 80, 16)	64	['expanded_conv_project[0][0]']
block_1_expand (Conv2D)	(None, 80, 80, 96)	1536	['expanded_conv_project_BN[0][0]']
block_1_expand_BN (BatchNormalization)	(None, 80, 80, 96)	384	['block_1_expand[0][0]']
block_1_expand_relu (ReLU)	(None, 80, 80, 96)	0	['block_1_expand_BN[0][0]']
block_1_pad (ZeroPadding2D)	(None, 81, 81, 96)	0	['block_1_expand_relu[0][0]']
block_1_depthwise (DepthwiseConv2D)	(None, 40, 40, 96)	864	['block_1_pad[0][0]']
block_1_depthwise_BN (BatchNormalization)	(None, 40, 40, 96)	384	['block_1_depthwise[0][0]']
block_1_depthwise_relu (ReLU)	(None, 40, 40, 96)	0	['block_1_depthwise_BN[0][0]']
block_1_project (Conv2D)	(None, 40, 40, 24)	2304	['block_1_depthwise_relu[0][0]']

block_1_project_BN (BatchNormalization)	(None, 40, 40, 24)	96	['block_1_pr ject[0][0]']
block_2_expand (Conv2D)	(None, 40, 40, 144)	3456	['block_1_pr ject_BN[0][0]']
block_2_expand_BN (BatchNormalization)	(None, 40, 40, 144)	576	['block_2_ex pand[0][0]']
block_2_expand_relu (ReLU)	(None, 40, 40, 144)	0	['block_2_ex pand_BN[0][0]']
block_2_depthwise (DepthwiseConv2D)	(None, 40, 40, 144)	1296	['block_2_ex pand_relu[0][0]']
block_2_depthwise_BN (BatchNormalization)	(None, 40, 40, 144)	576	['block_2_de pthwise[0][0]']
block_2_depthwise_relu (ReLU)	(None, 40, 40, 144)	0	['block_2_de pthwise_BN[0][0]']
block_2_project (Conv2D)	(None, 40, 40, 24)	3456	['block_2_de pthwise_relu[0][0]']
block_2_project_BN (BatchNormalization)	(None, 40, 40, 24)	96	['block_2_pr ject[0][0]']
block_2_add (Add)	(None, 40, 40, 24)	0	['block_1_pr ject_BN[0][0]',  'block_2_pr ject_BN[0][0]']
block_3_expand (Conv2D)	(None, 40, 40, 144)	3456	['block_2_ad d[0][0]']
block_3_expand_BN (BatchNormalization)	(None, 40, 40, 144)	576	['block_3_ex pand[0][0]']
block_3_expand_relu (ReLU)	(None, 40, 40, 144)	0	['block_3_ex pand_BN[0][0]']
block_3_pad (ZeroPadding2D)	(None, 41, 41, 144)	0	['block_3_ex pand_relu[0][0]']
block_3_depthwise (DepthwiseConv2D)	(None, 20, 20, 144)	1296	['block_3_pa d[0][0]']
block_3_depthwise_BN (BatchNormalization)	(None, 20, 20, 144)	576	['block_3_de

```

pthwise[0][0]']
malization)

block_3_depthwise_relu (ReLU) (None, 20, 20, 144) 0 ['block_3_de
pthwise_BN[0][0]']

block_3_project (Conv2D) (None, 20, 20, 32) 4608 ['block_3_de
pthwise_relu[0][0]']

block_3_project_BN (BatchNorma (None, 20, 20, 32) 128 ['block_3_pr
object[0][0]']
alization)

block_4_expand (Conv2D) (None, 20, 20, 192) 6144 ['block_3_pr
object_BN[0][0]']

block_4_expand_BN (BatchNormal (None, 20, 20, 192) 768 ['block_4_ex
pand[0][0]']
alization)

block_4_expand_relu (ReLU) (None, 20, 20, 192) 0 ['block_4_ex
pand_BN[0][0]']

block_4_depthwise (DepthwiseCo (None, 20, 20, 192) 1728 ['block_4_ex
pand_relu[0][0]']
nv2D)

block_4_depthwise_BN (BatchNor (None, 20, 20, 192) 768 ['block_4_de
pthwise[0][0]']
malization)

block_4_depthwise_relu (ReLU) (None, 20, 20, 192) 0 ['block_4_de
pthwise_BN[0][0]']

block_4_project (Conv2D) (None, 20, 20, 32) 6144 ['block_4_de
pthwise_relu[0][0]']

block_4_project_BN (BatchNorma (None, 20, 20, 32) 128 ['block_4_pr
object[0][0]']
alization)

block_4_add (Add) (None, 20, 20, 32) 0 ['block_3_pr
object_BN[0][0]','
'block_4_pr
object_BN[0][0]']

block_5_expand (Conv2D) (None, 20, 20, 192) 6144 ['block_4_ad
d[0][0]']

block_5_expand_BN (BatchNormal (None, 20, 20, 192) 768 ['block_5_ex
pand[0][0]']
alization)

block_5_expand_relu (ReLU) (None, 20, 20, 192) 0 ['block_5_ex
pand_BN[0][0]']

```

block_5_depthwise (DepthwiseConv2D) expand_relu[0][0]'	(None, 20, 20, 192)	1728	['block_5_ex
block_5_depthwise_BN (BatchNormalization)	(None, 20, 20, 192)	768	['block_5_de
block_5_depthwise_relu (ReLU)	(None, 20, 20, 192)	0	['block_5_de
block_5_project (Conv2D)	(None, 20, 20, 32)	6144	['block_5_de
block_5_project_BN (BatchNormalization)	(None, 20, 20, 32)	128	['block_5_pr
block_5_add (Add)	(None, 20, 20, 32)	0	['block_4_ad
d[0][0]',			'block_5_pr
object_BN[0][0]'			
block_6_expand (Conv2D)	(None, 20, 20, 192)	6144	['block_5_ad
d[0][0]'			
block_6_expand_BN (BatchNormalization)	(None, 20, 20, 192)	768	['block_6_ex
block_6_expand_relu (ReLU)	(None, 20, 20, 192)	0	['block_6_ex
pand_BN[0][0]'			
block_6_pad (ZeroPadding2D)	(None, 21, 21, 192)	0	['block_6_ex
pand_relu[0][0]'			
block_6_depthwise (DepthwiseConv2D) nv2D)	(None, 10, 10, 192)	1728	['block_6_pa
block_6_depthwise_BN (BatchNormalization)	(None, 10, 10, 192)	768	['block_6_de
block_6_depthwise_relu (ReLU)	(None, 10, 10, 192)	0	['block_6_de
pand_BN[0][0]'			
block_6_project (Conv2D)	(None, 10, 10, 64)	12288	['block_6_de
pand_relu[0][0]'			
block_6_project_BN (BatchNormalization)	(None, 10, 10, 64)	256	['block_6_pr
object[0][0]'			
block_7_expand (Conv2D)	(None, 10, 10, 384)	24576	['block_6_pr
object_BN[0][0]'			

block_7_expand_BN (BatchNormal pand[0][0]') ization)	(None, 10, 10, 384)	1536	['block_7_ex
block_7_expand_relu (ReLU) pand_BN[0][0]')	(None, 10, 10, 384)	0	['block_7_ex
block_7_depthwise (DepthwiseCo pand_relu[0][0]') nv2D)	(None, 10, 10, 384)	3456	['block_7_ex
block_7_depthwise_BN (BatchNor pthwise[0][0]') malization)	(None, 10, 10, 384)	1536	['block_7_de
block_7_depthwise_relu (ReLU) pthwise_BN[0][0]')	(None, 10, 10, 384)	0	['block_7_de
block_7_project (Conv2D) pthwise_relu[0][0]')	(None, 10, 10, 64)	24576	['block_7_de
block_7_project_BN (BatchNorma object[0][0]') lization)	(None, 10, 10, 64)	256	['block_7_pr
block_7_add (Add) object_BN[0][0]',' object_BN[0][0]')	(None, 10, 10, 64)	0	['block_6_pr  'block_7_pr
block_8_expand (Conv2D) d[0][0]')	(None, 10, 10, 384)	24576	['block_7_ad
block_8_expand_BN (BatchNormal pand[0][0]') ization)	(None, 10, 10, 384)	1536	['block_8_ex
block_8_expand_relu (ReLU) pand_BN[0][0]')	(None, 10, 10, 384)	0	['block_8_ex
block_8_depthwise (DepthwiseCo pand_relu[0][0]') nv2D)	(None, 10, 10, 384)	3456	['block_8_ex
block_8_depthwise_BN (BatchNor pthwise[0][0]') malization)	(None, 10, 10, 384)	1536	['block_8_de
block_8_depthwise_relu (ReLU) pthwise_BN[0][0]')	(None, 10, 10, 384)	0	['block_8_de
block_8_project (Conv2D) pthwise_relu[0][0]')	(None, 10, 10, 64)	24576	['block_8_de
block_8_project_BN (BatchNorma	(None, 10, 10, 64)	256	['block_8_pr

```

object[0][0]']
    lization)

    block_8_add (Add)                (None, 10, 10, 64)    0                ['block_7_ad
d[0][0]'],
                                'block_8_pr
object_BN[0][0]']

    block_9_expand (Conv2D)          (None, 10, 10, 384)  24576            ['block_8_ad
d[0][0]']

    block_9_expand_BN (BatchNormal   (None, 10, 10, 384)  1536            ['block_9_ex
pand[0][0]']
    ization)

    block_9_expand_relu (ReLU)       (None, 10, 10, 384)  0                ['block_9_ex
pand_BN[0][0]']

    block_9_depthwise (DepthwiseCo   (None, 10, 10, 384)  3456            ['block_9_ex
pand_relu[0][0]']
    nv2D)

    block_9_depthwise_BN (BatchNor   (None, 10, 10, 384)  1536            ['block_9_de
pthwise[0][0]']
    malization)

    block_9_depthwise_relu (ReLU)    (None, 10, 10, 384)  0                ['block_9_de
pthwise_BN[0][0]']

    block_9_project (Conv2D)         (None, 10, 10, 64)   24576            ['block_9_de
pthwise_relu[0][0]']

    block_9_project_BN (BatchNorma   (None, 10, 10, 64)   256             ['block_9_pr
object[0][0]']
    lization)

    block_9_add (Add)                (None, 10, 10, 64)   0                ['block_8_ad
d[0][0]'],
                                'block_9_pr
object_BN[0][0]']

    block_10_expand (Conv2D)         (None, 10, 10, 384)  24576            ['block_9_ad
d[0][0]']

    block_10_expand_BN (BatchNorma   (None, 10, 10, 384)  1536            ['block_10_e
xpand[0][0]']
    lization)

    block_10_expand_relu (ReLU)      (None, 10, 10, 384)  0                ['block_10_e
xpand_BN[0][0]']

    block_10_depthwise (DepthwiseC   (None, 10, 10, 384)  3456            ['block_10_e
xpand_relu[0][0]']
    onv2D)

    block_10_depthwise_BN (BatchNo   (None, 10, 10, 384)  1536            ['block_10_d

```

```

epthwise[0][0]']
    rmalization)

    block_10_depthwise_relu (ReLU) (None, 10, 10, 384) 0 ['block_10_d
epthwise_BN[0][0]']

    block_10_project (Conv2D) (None, 10, 10, 96) 36864 ['block_10_d
epthwise_relu[0][0]']

    block_10_project_BN (BatchNorm (None, 10, 10, 96) 384 ['block_10_p
roject[0][0]']
    alization)

    block_11_expand (Conv2D) (None, 10, 10, 576) 55296 ['block_10_p
roject_BN[0][0]']

    block_11_expand_BN (BatchNorma (None, 10, 10, 576) 2304 ['block_11_e
xpand[0][0]']
    lization)

    block_11_expand_relu (ReLU) (None, 10, 10, 576) 0 ['block_11_e
xpand_BN[0][0]']

    block_11_depthwise (DepthwiseC (None, 10, 10, 576) 5184 ['block_11_e
xpand_relu[0][0]']
    onv2D)

    block_11_depthwise_BN (BatchNo (None, 10, 10, 576) 2304 ['block_11_d
epthwise[0][0]']
    rmalization)

    block_11_depthwise_relu (ReLU) (None, 10, 10, 576) 0 ['block_11_d
epthwise_BN[0][0]']

    block_11_project (Conv2D) (None, 10, 10, 96) 55296 ['block_11_d
epthwise_relu[0][0]']

    block_11_project_BN (BatchNorm (None, 10, 10, 96) 384 ['block_11_p
roject[0][0]']
    alization)

    block_11_add (Add) (None, 10, 10, 96) 0 ['block_10_p
roject_BN[0][0]'],
    'block_11_p
roject_BN[0][0]']

    block_12_expand (Conv2D) (None, 10, 10, 576) 55296 ['block_11_a
dd[0][0]']

    block_12_expand_BN (BatchNorma (None, 10, 10, 576) 2304 ['block_12_e
xpand[0][0]']
    lization)

    block_12_expand_relu (ReLU) (None, 10, 10, 576) 0 ['block_12_e
xpand_BN[0][0]']

```

block_12_depthwise (DepthwiseConv2D) expand_relu[0][0]'	(None, 10, 10, 576)	5184	['block_12_e
block_12_depthwise_BN (BatchNormalization) epthwise[0][0]'	(None, 10, 10, 576)	2304	['block_12_d
block_12_depthwise_relu (ReLU) epthwise_BN[0][0]'	(None, 10, 10, 576)	0	['block_12_d
block_12_project (Conv2D) epthwise_relu[0][0]'	(None, 10, 10, 96)	55296	['block_12_d
block_12_project_BN (BatchNormalization) roject[0][0]'	(None, 10, 10, 96)	384	['block_12_p
block_12_add (Add) dd[0][0]',	(None, 10, 10, 96)	0	['block_11_a
block_12_add (Add) roject_BN[0][0]'	(None, 10, 10, 96)	0	'block_12_p
block_13_expand (Conv2D) dd[0][0]'	(None, 10, 10, 576)	55296	['block_12_a
block_13_expand_BN (BatchNormalization) expand[0][0]'	(None, 10, 10, 576)	2304	['block_13_e
block_13_expand_relu (ReLU) expand_BN[0][0]'	(None, 10, 10, 576)	0	['block_13_e
block_13_pad (ZeroPadding2D) expand_relu[0][0]'	(None, 11, 11, 576)	0	['block_13_e
block_13_depthwise (DepthwiseConv2D) ad[0][0]'	(None, 5, 5, 576)	5184	['block_13_p
block_13_depthwise_BN (BatchNormalization) epthwise[0][0]'	(None, 5, 5, 576)	2304	['block_13_d
block_13_depthwise_relu (ReLU) epthwise_BN[0][0]'	(None, 5, 5, 576)	0	['block_13_d
block_13_project (Conv2D) epthwise_relu[0][0]'	(None, 5, 5, 160)	92160	['block_13_d
block_13_project_BN (BatchNormalization) roject[0][0]'	(None, 5, 5, 160)	640	['block_13_p
block_14_expand (Conv2D) roject_BN[0][0]'	(None, 5, 5, 960)	153600	['block_13_p



block_14_expand_BN (BatchNormalization)	(None, 5, 5, 960)	3840	['block_14_expand[0][0]']
block_14_expand_relu (ReLU)	(None, 5, 5, 960)	0	['block_14_expand_BN[0][0]']
block_14_depthwise (DepthwiseConv2D)	(None, 5, 5, 960)	8640	['block_14_expand_relu[0][0]']
block_14_depthwise_BN (BatchNormalization)	(None, 5, 5, 960)	3840	['block_14_depthwise[0][0]']
block_14_depthwise_relu (ReLU)	(None, 5, 5, 960)	0	['block_14_depthwise_BN[0][0]']
block_14_project (Conv2D)	(None, 5, 5, 160)	153600	['block_14_depthwise_relu[0][0]']
block_14_project_BN (BatchNormalization)	(None, 5, 5, 160)	640	['block_14_project[0][0]']
block_14_add (Add)	(None, 5, 5, 160)	0	['block_13_project_BN[0][0]', 'block_14_project_BN[0][0]']
block_15_expand (Conv2D)	(None, 5, 5, 960)	153600	['block_14_add[0][0]']
block_15_expand_BN (BatchNormalization)	(None, 5, 5, 960)	3840	['block_15_expand[0][0]']
block_15_expand_relu (ReLU)	(None, 5, 5, 960)	0	['block_15_expand_BN[0][0]']
block_15_depthwise (DepthwiseConv2D)	(None, 5, 5, 960)	8640	['block_15_expand_relu[0][0]']
block_15_depthwise_BN (BatchNormalization)	(None, 5, 5, 960)	3840	['block_15_depthwise[0][0]']
block_15_depthwise_relu (ReLU)	(None, 5, 5, 960)	0	['block_15_depthwise_BN[0][0]']
block_15_project (Conv2D)	(None, 5, 5, 160)	153600	['block_15_depthwise_relu[0][0]']
block_15_project_BN (BatchNormalization)	(None, 5, 5, 160)	640	['block_15_project[0][0]']

```

project[0][0]']
alization)

block_15_add (Add)          (None, 5, 5, 160)    0      ['block_14_a
dd[0][0]'],
                                'block_15_p
project_BN[0][0]']

block_16_expand (Conv2D)    (None, 5, 5, 960)   153600  ['block_15_a
dd[0][0]']

block_16_expand_BN (BatchNorma (None, 5, 5, 960)   3840    ['block_16_e
xpand[0][0]']
alization)

block_16_expand_relu (ReLU)  (None, 5, 5, 960)    0      ['block_16_e
xpand_BN[0][0]']

block_16_depthwise (DepthwiseC (None, 5, 5, 960)   8640    ['block_16_e
xpand_relu[0][0]']
onv2D)

block_16_depthwise_BN (BatchNo (None, 5, 5, 960)   3840    ['block_16_d
epthwise[0][0]']
rmalization)

block_16_depthwise_relu (ReLU) (None, 5, 5, 960)    0      ['block_16_d
epthwise_BN[0][0]']

block_16_project (Conv2D)    (None, 5, 5, 320)   307200  ['block_16_d
epthwise_relu[0][0]']

block_16_project_BN (BatchNorm (None, 5, 5, 320)   1280    ['block_16_p
roject[0][0]']
alization)

Conv_1 (Conv2D)             (None, 5, 5, 1280)  409600  ['block_16_p
project_BN[0][0]']

Conv_1_bn (BatchNormalization) (None, 5, 5, 1280)  5120    ['Conv_1[0]
[0]']

out_relu (ReLU)             (None, 5, 5, 1280)    0      ['Conv_1_bn
[0][0]']

```

```

=====
=====

```

```

Total params: 2,257,984
Trainable params: 0
Non-trainable params: 2,257,984

```

```

In [ ]: global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
feature_batch_average = global_average_layer(feature_batch)
print(feature_batch_average.shape)

```

```
prediction_layer = tf.keras.layers.Dense(1)
prediction_batch = prediction_layer(feature_batch_average)
print(prediction_batch.shape)
```

```
(32, 1280)
```

```
(32, 1)
```

```
In [ ]: # Not sure why the data augmentation cannot be defined, the library is in the
inputs = tf.keras.Input(shape=(160, 160, 3))
x = data_augmentation(inputs)
x = preprocess_input(x)
x = base_model(x, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.2)(x)
outputs = prediction_layer(x)
t_model = tf.keras.Model(inputs, outputs)
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[159], line 3
      1 # Not sure why the data augmentation cannot be defined, the library is in the file
      2 inputs = tf.keras.Input(shape=(160, 160, 3))
----> 3 x = data_augmentation(inputs)
      4 x = preprocess_input(x)
      5 x = base_model(x, training=False)

NameError: name 'data_augmentation' is not defined
```

```
In [ ]: base_learning_rate = 0.0001
t_model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),
                loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
                metrics=['accuracy'])

t_model.summary()
```

```
In [ ]: initial_epochs = 10

loss0, accuracy0 = t_model.evaluate(validation_dataset)

print("initial loss: {:.2f}".format(loss0))
print("initial accuracy: {:.2f}".format(accuracy0))

history = t_model.fit(train_dataset,
                      epochs=initial_epochs,
                      validation_data=validation_dataset)

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
```

```
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.ylabel('Accuracy')
plt.ylim([min(plt.ylim()),1])
plt.title('Training and Validation Accuracy')

plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.ylabel('Cross Entropy')
plt.ylim([0,1.0])
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.show()
```

### Analysis

This was a very interesting project because the results are something that are very relevant to what is happening in the computer science world right now. I picked a dataset of different balls and made models that would be able to classify them. My first model, the Sequential model, was my most effective model compared to the the CNN model. This was interesting since CNN are known for being efficient on Image Classification cases. I need to continue investigating if I added my layers correctly or not. Even so, my Sequential model reached 97.7% accuracy which is still very efficient. I also tested the model on 3 images and it classified them correctly with all around 98% accuracy. I am curious to see how a better trained CNN model will perform on this dataset.