Image Classification

Yazan Abughazaleh

The purpose of this notebook is to demonstrate image classification with neural networks. The notebook will be utilizing a sequential model, a CNN model, a pre-trained model, and a transfer learning model. The first step we need to perform is to load the image data which we would like to classify. Our data is separated into four groups: Two sets of images for two different classes and two CSV files containing labels and other data associated with each image. The data set is the Animal Crossing or Doom Dataset found at https://www.kaggle.com/datasets/andrewmvd/doom-crossing?resource=download.

In [ ]:
```
from keras.layers import Dense, Activation, Conv2D, Flatten, Dropout, MaxPoo
from keras_preprocessing.image import ImageDataGenerator
from keras import regularizers, optimizers
import os
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf
import PIL
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
```

After importing tensorflow and other libraries above, we can start to load our data.

In [ ]:
```
from sklearn.model_selection import train_test_split
dfA = pd.read_csv("dataset/animal_crossing_dataset.csv",dtype="str")
dfB = pd.read_csv("dataset/doom_crossing_dataset.csv",dtype="str")
df = pd.concat([dfA,dfB])
print(dfA.shape)
print(dfB.shape)
print(df.shape)
train, test = train_test_split(df, test_size=0.2, random_state=1234)
train.head()
```

```
(757, 11)
(839, 11)
(1596, 11)
```

Out[ ]:

	subreddit	id	title	ups	downs	upvote_ratio	total_awards_received
458	AnimalCrossing	g1ddyw	My mid-century modern bathroom~	15971	0	0.99	1
486	AnimalCrossing	gietxl	Okay well one of us is gonna have to change	15476	0	0.99	0
514	Doom	fq21bg	You damn well know he was excited	3249	0	1	1
645	Doom	cvsvgi	I present you the 2016 version	2829	0	0.98	0
33	AnimalCrossing	g3ogs3	This bugs me so much	53800	0	0.93	2

In [ ]:

```
import shutil


def idTrainImage(dfA,dfB):
    if os.path.exists('./dataset/test/'):
        shutil.rmtree('./dataset/test/')
    if os.path.exists('./dataset/train/'):
        shutil.rmtree('./dataset/train/')
    os.mkdir('./dataset/train/')
    os.mkdir('./dataset/test/')
    #dfA['filename'] = dfA['filename'].astype('|S')
    #dfB['filename'] = dfB['filename'].astype('|S')
    #print(dfA['filename'])
    for index, row in dfA.iterrows():
        filename = str(dfA['filename'][index])
        filename = filename.split()
        if len(filename) > 1:
            copyname = './dataset/images/'+filename[1]

        shutil.copy(str(copyname),'./dataset/train')
    for index, row in dfB.iterrows():
        filename = str(dfB['filename'][index])
        filename = filename.split()
        if len(filename) > 1:
            copyname = './dataset/images/'+filename[1]
            shutil.copy(str(copyname),'./dataset/test/')

idTrainImage(train,test)
```

The function above splits the images into a train and test data set based on the labels in the csv files. We can now load our images.

In [ ]:
```
trainingImage = tf.keras.utils.image_dataset_from_directory('./archive/',
    batch_size=32,
    image_size=(100, 100),
    seed=1234,
    validation_split=.2,
    subset="training")

validationImage = tf.keras.utils.image_dataset_from_directory(
    './archive/',
    batch_size=32,
    image_size=(100, 100),
    seed=1234,
    validation_split=.2,
    subset="validation"
)
```

```
Found 1597 files belonging to 2 classes.
Using 1278 files for training.
Found 1597 files belonging to 2 classes.
Using 319 files for validation.
```

In [ ]:
```
print("Cardinality Train Set %d" % tf.data.experimental.cardinality(trainingI
val = tf.data.experimental.cardinality(validationImage) // 5
testImage = validationImage.take(val)
validationImage = validationImage.skip(val)
print("Cardinality Validation Set %d" % tf.data.experimental.cardinality(val
print("Cardinality Test Set %d" % tf.data.experimental.cardinality(testImage
```

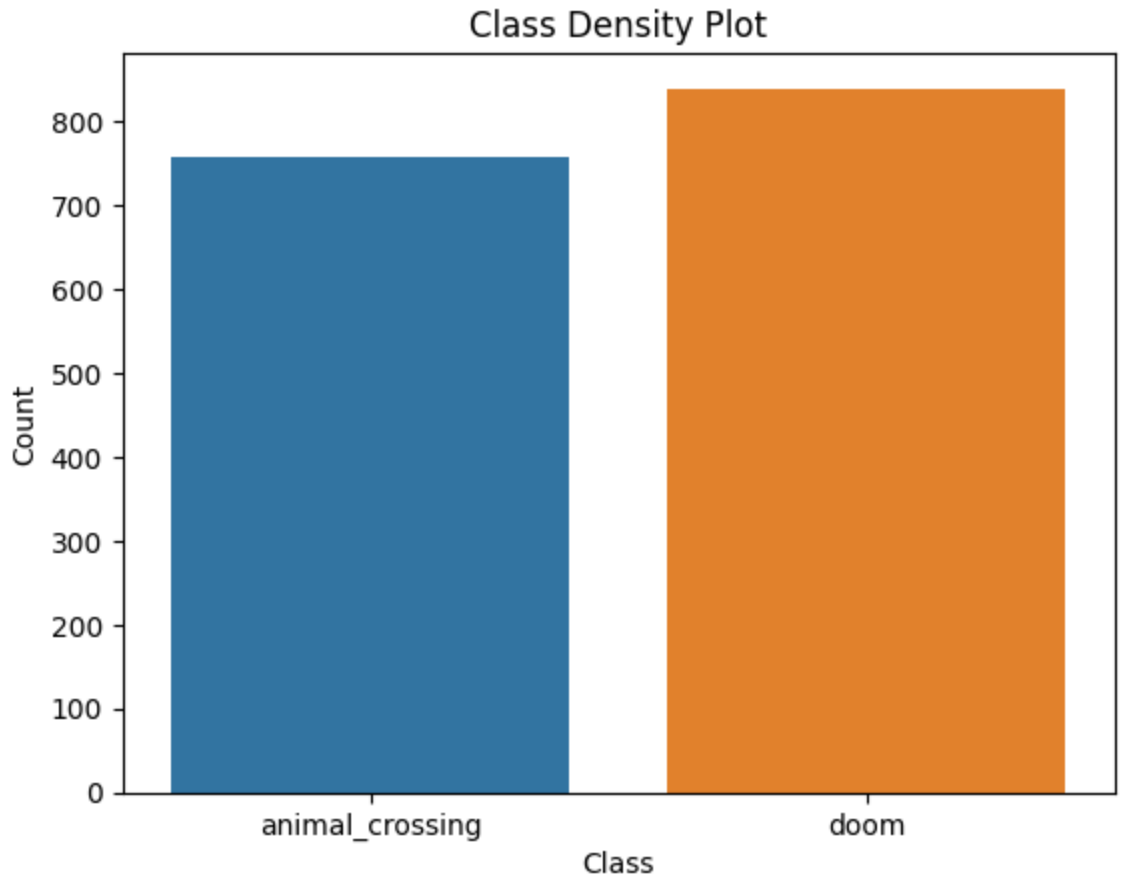
```
Cardinality Train Set 40
Cardinality Validation Set 8
Cardinality Test Set 2
```

Now we would like to create a graph of the distribution of the image classes. To do this, we can take the count of each image class and graph that as a bar plot.

In [ ]:
```
import seaborn as sb
class_dist = []
class_dist.append(len(dfA.index))
class_dist.append(len(dfB.index))

#X = np.array(['animal_crossing','doom'])
X = np.array(trainingImage.class_names)
Y = np.array(class_dist)
bPlot = sb.barplot(x=X,y=Y)
bPlot.set(xlabel="Class",ylabel="Count", title="Class Density Plot")
```

Out[ ]:
```
[Text(0.5, 0, 'Class'),
 Text(0, 0.5, 'Count'),
 Text(0.5, 1.0, 'Class Density Plot')]
```

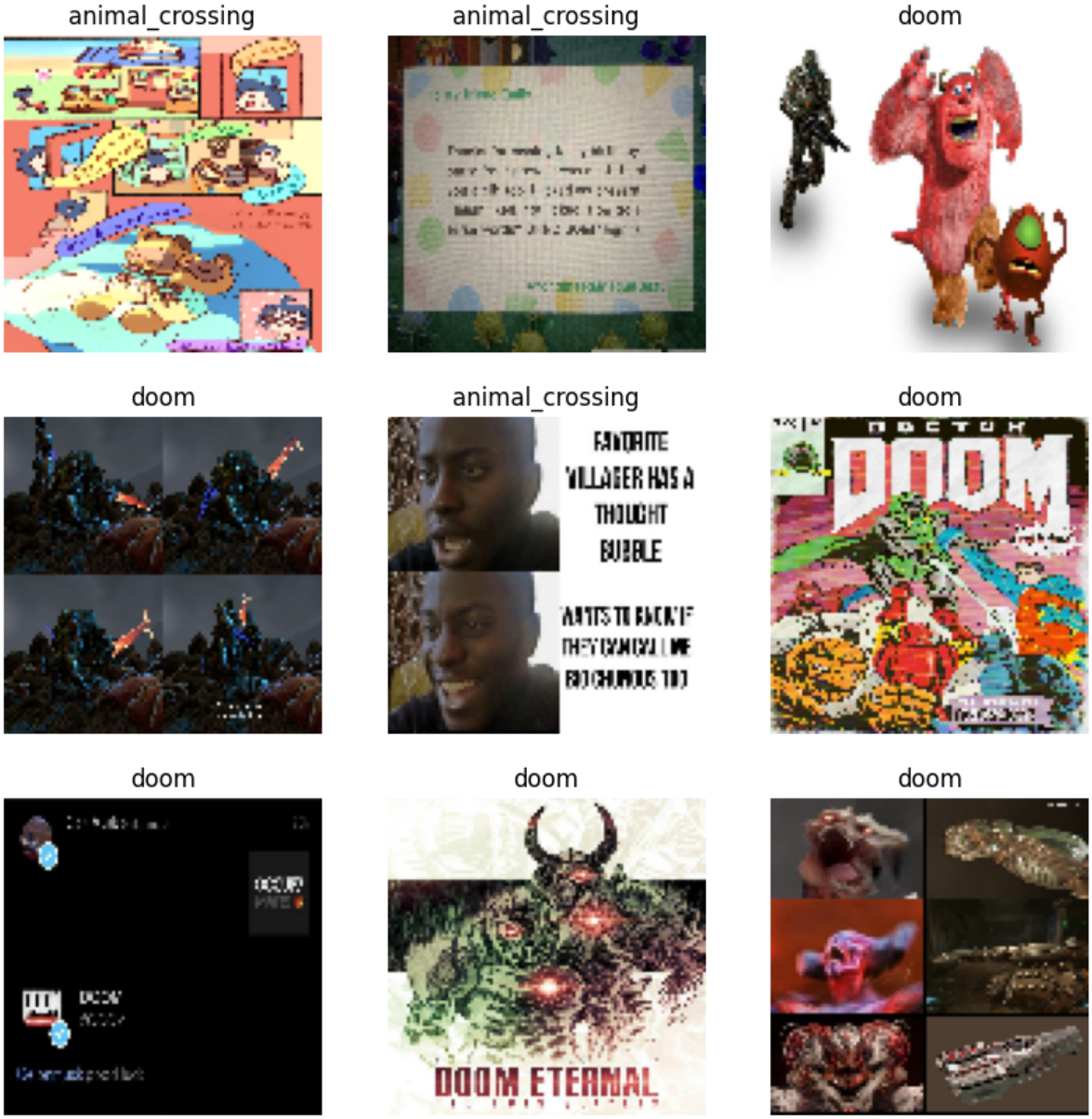


The goal of this classification model is to determine whether an image is associated with the Animal Crossing or Doom Franchise of video games. Below a few examples are provided showing labels with the correct images.

In [ ]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
class_names = trainingImage.class_names
plt.figure(figsize=(10, 10))
for images, labels in trainingImage.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")
```

```
2023-04-16 15:10:35.097536: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate 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 [1278]
         [[{{node Placeholder/_0}}]]
2023-04-16 15:10:35.099485: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate an error and you can ignore this message): INVALID_ARGUMENT: You must
feed a value for placeholder tensor 'Placeholder/_4' with dtype int32 and s
hape [1278]
         [[{{node Placeholder/_4}}]]
```

```
<Figure size 1000x1000 with 0 Axes>
```



Configure Data for Performance

To maximize the performance of the training process, we will use buffered prefetching to stop I/O blocking from happening and speed up the data fetching process.

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

trainingImage = trainingImage.cache().shuffle(1000).prefetch(buffer_size=AUT
validationImage = validationImage.cache().prefetch(buffer_size=AUTOTUNE)
```

Scaling the Data

The range of the input values between 0 and 255 is not ideal for training with a neural network. We want to put it in the range of 0 and 1 because neural networks work best on small data.

In [ ]:
```
normalization_layer = layers.Rescaling(1./255)
normalized_ds = trainingImage.map(lambda x, y: (normalization_layer(x), y))
image_batch, labels_batch = next(iter(normalized_ds))
first_image = image_batch[0]

print(np.min(first_image), np.max(first_image))
```

```
2023-04-16 15:10:51.446038: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate an error and you can ignore this message): INVALID_ARGUMENT: You must
feed a value for placeholder tensor 'Placeholder/_4' with dtype int32 and s
hape [1278]
         [[{{node Placeholder/_4}}]]
2023-04-16 15:10:51.447871: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate an error and you can ignore this message): INVALID_ARGUMENT: You must
feed a value for placeholder tensor 'Placeholder/_4' with dtype int32 and s
hape [1278]
         [[{{node Placeholder/_4}}]]
2023-04-16 15:11:02.748863: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 15:11:03.505834: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 15:11:06.808786: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 15:11:11.507229: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 15:11:20.154900: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 15:11:24.291026: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: extra compressed data
2023-04-16 15:11:31.342900: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
0.0 1.0
```

Data Augmentation

In [ ]:
```
data_augmentation = keras.Sequential(
  [
    layers.RandomFlip("horizontal",
                      input_shape=(100,
                                   100,
                                   3)),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
  ]
)
```

Building the Model

In [ ]:
```
num_classes = 2
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(100, 100,3)),
```

```
  tf.keras.layers.Dense(512, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(512, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(num_classes, activation='softmax'),
])
```

Compile the Model

In [ ]:
```
model.compile(optimizer='rmsprop',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits
              metrics=['accuracy'])
```

Model Summary

In [ ]:
```
model.summary()
```

```
Model: "sequential_10"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_9 (Flatten)         (None, 784)               0

 dense_20 (Dense)            (None, 512)               401920

 dropout_6 (Dropout)         (None, 512)               0

 dense_21 (Dense)            (None, 512)               262656

 dropout_7 (Dropout)         (None, 512)               0

 dense_22 (Dense)            (None, 2)                 1026

=================================================================
Total params: 665,602
Trainable params: 665,602
Non-trainable params: 0
_________________________________________________________________
```

Model Training

In [ ]:
```
epochs=20
history = model.fit(
  trainingImage,
  validation_data=validationImage,
  epochs=epochs
)
```

```
Epoch 1/20
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-pack
ages/keras/backend.py:5612: UserWarning: "`sparse_categorical_crossentropy`
received `from_logits=True`, but the `output` argument was produced by a So
ftmax activation and thus does not represent logits. Was this intended?
  output, from_logits = _get_logits(
```

```
40/40 [==============================] - 10s 182ms/step - loss: 3723.5117 -
accuracy: 0.4969 - val_loss: 652.2740 - val_accuracy: 0.5333
Epoch 2/20
40/40 [==============================] - 6s 161ms/step - loss: 135.5471 - a
ccuracy: 0.5070 - val_loss: 0.7382 - val_accuracy: 0.5333
Epoch 3/20
40/40 [==============================] - 6s 158ms/step - loss: 2.1876 - acc
uracy: 0.4977 - val_loss: 0.6914 - val_accuracy: 0.5294
Epoch 4/20
40/40 [==============================] - 6s 157ms/step - loss: 3.3401 - acc
uracy: 0.5149 - val_loss: 0.7035 - val_accuracy: 0.5294
Epoch 5/20
40/40 [==============================] - 6s 161ms/step - loss: 2.6347 - acc
uracy: 0.5243 - val_loss: 0.7078 - val_accuracy: 0.5294
Epoch 6/20
40/40 [==============================] - 7s 166ms/step - loss: 2.9214 - acc
uracy: 0.5219 - val_loss: 0.7092 - val_accuracy: 0.5294
Epoch 7/20
40/40 [==============================] - 7s 162ms/step - loss: 0.7978 - acc
uracy: 0.5203 - val_loss: 0.7037 - val_accuracy: 0.5294
Epoch 8/20
40/40 [==============================] - 10s 242ms/step - loss: 0.6907 - ac
curacy: 0.5235 - val_loss: 0.7022 - val_accuracy: 0.5294
Epoch 9/20
40/40 [==============================] - 7s 175ms/step - loss: 8.7998 - acc
uracy: 0.5235 - val_loss: 0.7053 - val_accuracy: 0.5294
Epoch 10/20
40/40 [==============================] - 7s 175ms/step - loss: 0.7007 - acc
uracy: 0.5211 - val_loss: 0.7005 - val_accuracy: 0.5294
Epoch 11/20
40/40 [==============================] - 7s 179ms/step - loss: 0.6908 - acc
uracy: 0.5227 - val_loss: 0.7005 - val_accuracy: 0.5294
Epoch 12/20
40/40 [==============================] - 7s 182ms/step - loss: 0.6902 - acc
uracy: 0.5235 - val_loss: 0.7004 - val_accuracy: 0.5294
Epoch 13/20
40/40 [==============================] - 7s 172ms/step - loss: 2.1415 - acc
uracy: 0.5227 - val_loss: 0.7013 - val_accuracy: 0.5294
Epoch 14/20
40/40 [==============================] - 7s 184ms/step - loss: 0.6909 - acc
uracy: 0.5235 - val_loss: 0.7012 - val_accuracy: 0.5294
Epoch 15/20
40/40 [==============================] - 7s 165ms/step - loss: 0.6904 - acc
uracy: 0.5235 - val_loss: 0.7018 - val_accuracy: 0.5294
Epoch 16/20
40/40 [==============================] - 7s 164ms/step - loss: 0.6902 - acc
uracy: 0.5235 - val_loss: 0.7016 - val_accuracy: 0.5294
Epoch 17/20
40/40 [==============================] - 7s 167ms/step - loss: 0.6904 - acc
uracy: 0.5235 - val_loss: 0.7021 - val_accuracy: 0.5294
Epoch 18/20
40/40 [==============================] - 7s 175ms/step - loss: 0.6902 - acc
uracy: 0.5235 - val_loss: 0.7017 - val_accuracy: 0.5294
Epoch 19/20
40/40 [==============================] - 7s 168ms/step - loss: 0.6908 - acc
uracy: 0.5235 - val_loss: 0.7015 - val_accuracy: 0.5294
```

```
Epoch 20/20
40/40 [==============================] - 7s 176ms/step - loss: 0.7126 - acc
uracy: 0.5227 - val_loss: 0.6985 - val_accuracy: 0.5294
```

Visualizing the Training

In [ ]:

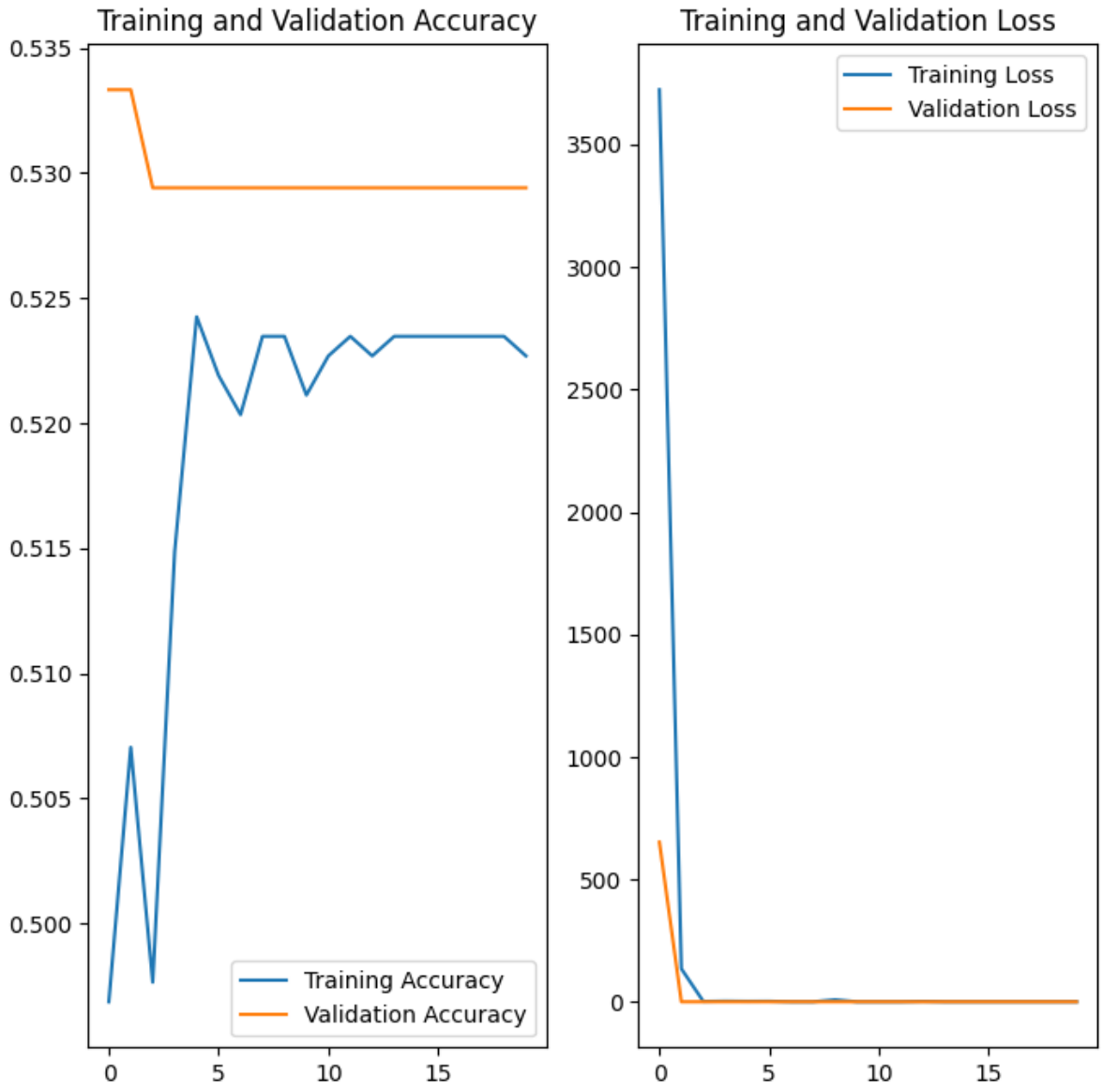
```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

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

epochs_range = range(epochs)

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()
```



The plots above show a validation accuracy that is higher than the training accuracy at around 53%. The model is likely not very good at predicting the image class.

Evaluate the Model on Test Data

In [ ]:
```
score = model.evaluate(testImage)

print("Loss: ",score[0])
print("Accuracy: ",score[1])
```

```
2023-04-16 17:27:20.059850: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: extra compressed data
2023-04-16 17:27:20.744041: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 17:27:22.891015: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 17:27:22.891209: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: cHRM chunk does not match sRGB
```

```
2/2 [==============================] - 10s 891ms/step - loss: 0.7327 - accu
racy: 0.5000
Loss:  0.7327266931533813
Accuracy:  0.5
```

We can see that the model has very mediocre accuracy on the test set at 50%.

Prediction on an Independent Image

Now we would like to classify a new image that has not been previously seen by the model. The first image should be classified under the animal crossing label while the second should belong to the doom class.

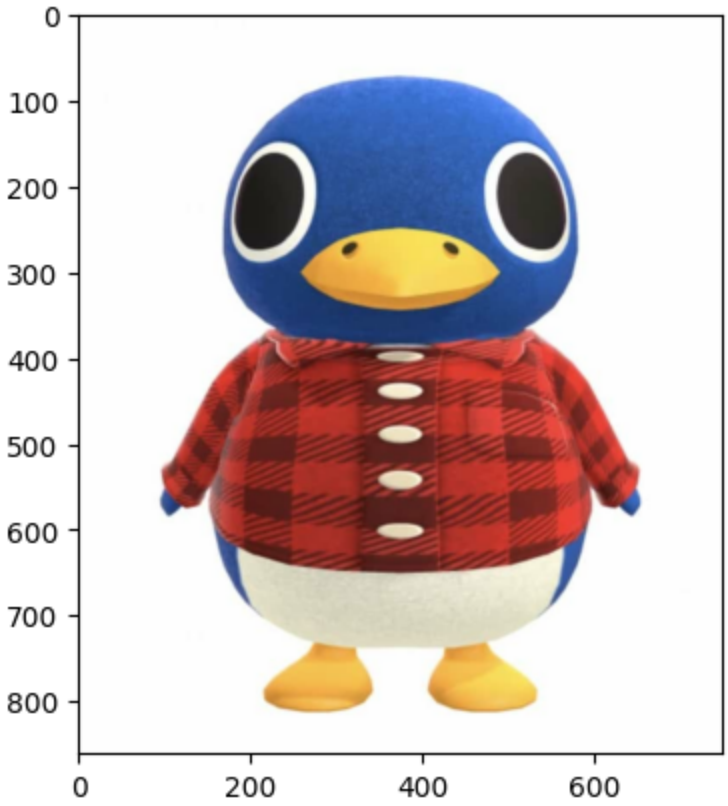
In [ ]:
```
from tensorflow import io
from tensorflow import image
img_path = 'penguino.jpeg'
im = io.read_file(img_path)
im = image.decode_jpeg(im,channels=3)
plt.imshow(im)

img = tf.keras.utils.load_img(
    img_path, target_size=(100, 100)
)
img_array = tf.keras.utils.img_to_array(img)
img_array = tf.expand_dims(img_array, 0) # Create a batch

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))
)
```

```
1/1 [==============================] - 0s 194ms/step
This image most likely belongs to doom with a 51.30 percent confidence.
```



In [ ]:
```
img_path = 'imgA.jpg'
im = io.read_file(img_path)
im = image.decode_jpeg(im,channels=3)
plt.imshow(im)

img = tf.keras.utils.load_img(
    img_path, target_size=(100, 100)
)
img_array = tf.keras.utils.img_to_array(img)
img_array = tf.expand_dims(img_array, 0) # Create a batch

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))
)
```

```
1/1 [==============================] - 0s 62ms/step
This image most likely belongs to doom with a 51.30 percent confidence.
```

Convolutional Network Base

We now would like to create a new model with a CNN architecture to see if we can yield any improvements.

In [ ]:
```
model = Sequential([
  layers.Rescaling(1./255, input_shape=(100, 100, 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)
])

model.summary()
```

```
Model: "sequential_12"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling_10 (Rescaling)    (None, 100, 100, 3)       0

 conv2d_18 (Conv2D)          (None, 100, 100, 16)      448


_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling_10 (Rescaling)    (None, 100, 100, 3)       0

 conv2d_18 (Conv2D)          (None, 100, 100, 16)      448

 max_pooling2d_18 (MaxPoolin  (None, 50, 50, 16)       0
 g2D)

 conv2d_19 (Conv2D)          (None, 50, 50, 32)        4640

 max_pooling2d_19 (MaxPoolin  (None, 25, 25, 32)       0
 g2D)

 conv2d_20 (Conv2D)          (None, 25, 25, 64)        18496

 max_pooling2d_20 (MaxPoolin  (None, 12, 12, 64)       0
 g2D)

 flatten_11 (Flatten)        (None, 9216)              0

 dense_26 (Dense)            (None, 128)               1179776

 dense_27 (Dense)            (None, 2)                 258

=================================================================
Total params: 1,203,618
Trainable params: 1,203,618
Non-trainable params: 0
_________________________________________________________________
```

Compile the CNN model

In [ ]:
```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits
              metrics=['accuracy'])
```

Fit the Model to the Training Data

We now we can proceed to train the model by using fitting it to the training data.

In [ ]:
```
epochs=20
history = model.fit(
  trainingImage,
  validation_data=validationImage,
```

```
  epochs=epochs
)
```

```
Epoch 1/20
40/40 [==============================] - 14s 257ms/step - loss: 0.6833 - ac
curacy: 0.5610 - val_loss: 0.6729 - val_accuracy: 0.5882
Epoch 2/20
40/40 [==============================] - 9s 227ms/step - loss: 0.5511 - acc
uracy: 0.7183 - val_loss: 0.5676 - val_accuracy: 0.7137
Epoch 3/20
40/40 [==============================] - 10s 242ms/step - loss: 0.5065 - ac
curacy: 0.7582 - val_loss: 0.5130 - val_accuracy: 0.7451
Epoch 4/20
40/40 [==============================] - 10s 255ms/step - loss: 0.4379 - ac
curacy: 0.7926 - val_loss: 0.4550 - val_accuracy: 0.7529
Epoch 5/20
40/40 [==============================] - 11s 271ms/step - loss: 0.4180 - ac
curacy: 0.7903 - val_loss: 0.5379 - val_accuracy: 0.7451
Epoch 6/20
40/40 [==============================] - 11s 262ms/step - loss: 0.3830 - ac
curacy: 0.8310 - val_loss: 0.5039 - val_accuracy: 0.7647
Epoch 7/20
40/40 [==============================] - 11s 273ms/step - loss: 0.3195 - ac
curacy: 0.8584 - val_loss: 0.4995 - val_accuracy: 0.7608
Epoch 8/20
40/40 [==============================] - 11s 268ms/step - loss: 0.2606 - ac
curacy: 0.8920 - val_loss: 0.4845 - val_accuracy: 0.7765
Epoch 9/20
40/40 [==============================] - 11s 287ms/step - loss: 0.2480 - ac
curacy: 0.8998 - val_loss: 0.4931 - val_accuracy: 0.7804
Epoch 10/20
40/40 [==============================] - 11s 276ms/step - loss: 0.1683 - ac
curacy: 0.9382 - val_loss: 0.6738 - val_accuracy: 0.7569
Epoch 11/20
40/40 [==============================] - 11s 270ms/step - loss: 0.1466 - ac
curacy: 0.9531 - val_loss: 0.6017 - val_accuracy: 0.7725
Epoch 12/20
40/40 [==============================] - 11s 277ms/step - loss: 0.1142 - ac
curacy: 0.9656 - val_loss: 0.7402 - val_accuracy: 0.7725
Epoch 13/20
40/40 [==============================] - 11s 282ms/step - loss: 0.0612 - ac
curacy: 0.9844 - val_loss: 0.7951 - val_accuracy: 0.7961
Epoch 14/20
40/40 [==============================] - 11s 274ms/step - loss: 0.0405 - ac
curacy: 0.9914 - val_loss: 0.9560 - val_accuracy: 0.7333
Epoch 15/20
40/40 [==============================] - 11s 275ms/step - loss: 0.0265 - ac
curacy: 0.9969 - val_loss: 0.8120 - val_accuracy: 0.7490
Epoch 16/20
40/40 [==============================] - 12s 297ms/step - loss: 0.0200 - ac
curacy: 0.9969 - val_loss: 0.8765 - val_accuracy: 0.7569
Epoch 17/20
40/40 [==============================] - 11s 274ms/step - loss: 0.0197 - ac
curacy: 0.9969 - val_loss: 1.0628 - val_accuracy: 0.7333
Epoch 18/20
40/40 [==============================] - 12s 294ms/step - loss: 0.0142 - ac
curacy: 0.9977 - val_loss: 0.9524 - val_accuracy: 0.7686
Epoch 19/20
40/40 [==============================] - 14s 351ms/step - loss: 0.0132 - ac
```

```
curacy: 0.9977 - val_loss: 1.0085 - val_accuracy: 0.7725
Epoch 20/20
40/40 [==============================] - 16s 402ms/step - loss: 0.0643 - ac
curacy: 0.9804 - val_loss: 0.9362 - val_accuracy: 0.7647
```

The model has now been trained, so we can again visualize its performance.

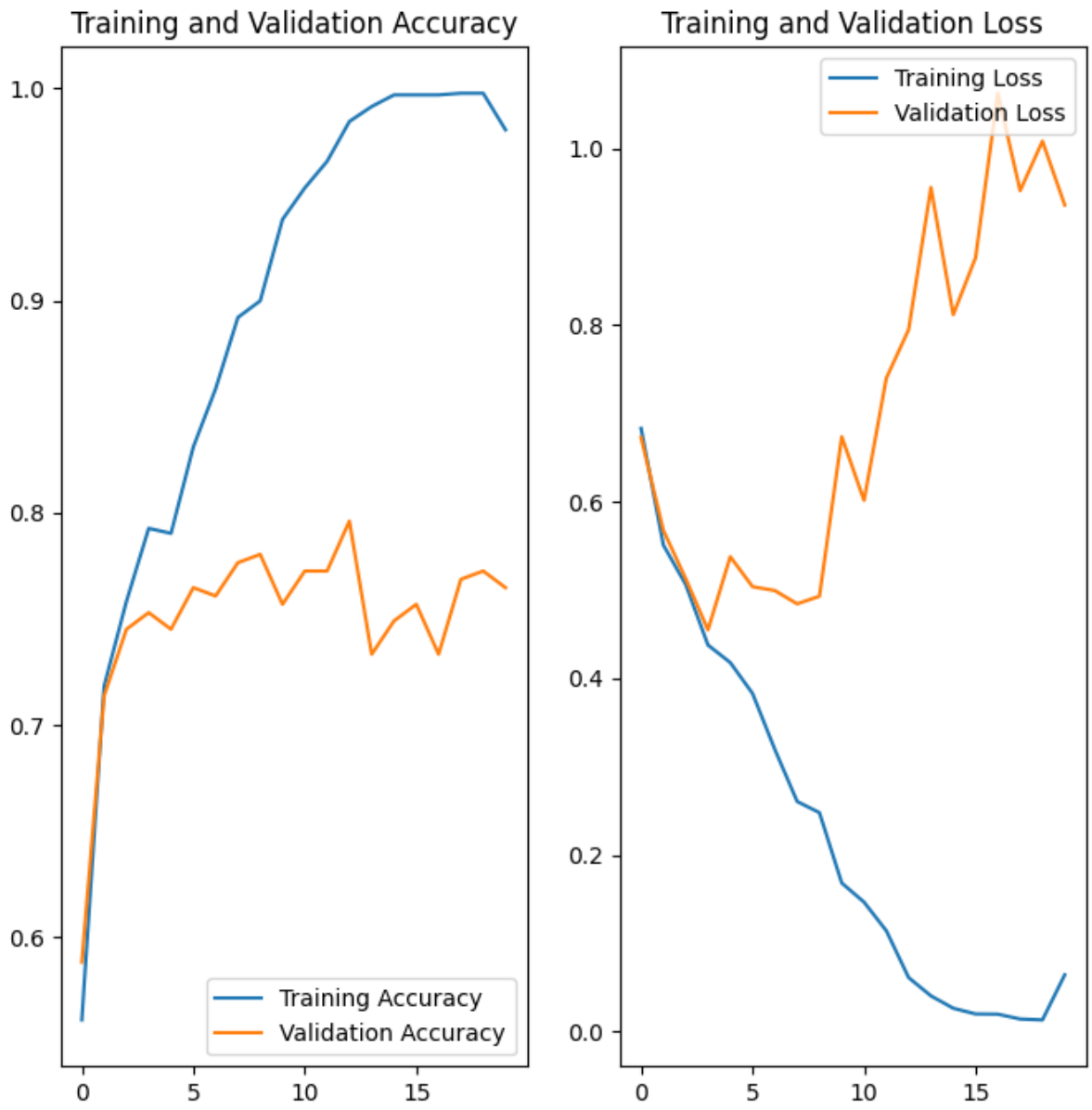
In [ ]:
```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

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

epochs_range = range(epochs)

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()
```



In the plots above we can see that the training anv validation accuracies have a significant difference in their results. The training accuracy is significantly higher than the validation accuracy, indicating that the model is not over fitting and the validation accuracy is relatively high at around 75% indicating good results for predictions. This is significantly better than the Sequential Model.

Evaluate the Model

In [ ]:
```
score = model.evaluate(testImage)

print("Loss: ",score[0])
print("Accuracy: ",score[1])
```

```
2023-04-16 17:33:23.598832: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: extra compressed data
2023-04-16 17:33:24.589550: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 17:33:27.284245: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 17:33:27.284287: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: cHRM chunk does not match sRGB
2/2 [==============================] - 10s 1s/step - loss: 0.7030 - accurac
y: 0.7969
Loss:  0.7030230164527893
Accuracy:  0.796875
```

With the evaluation on the test set, the performance of the CNN model had a significant improvement in accuracy over the sequential model

Predicting on New Data

For this we will repeat the data used earlier.

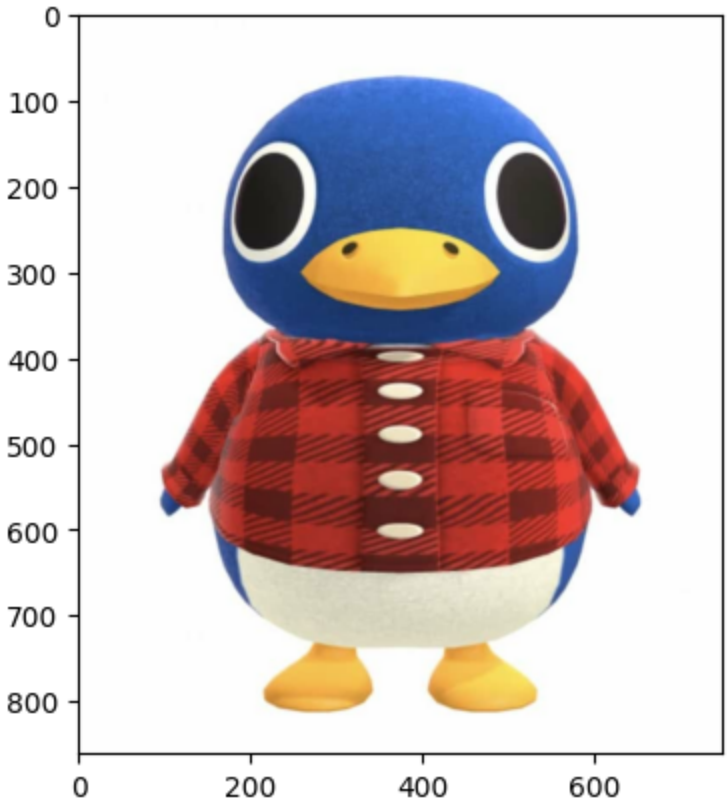
In [ ]:
```
img_path = 'penguino.jpeg'
im = io.read_file(img_path)
im = image.decode_jpeg(im,channels=3)
plt.imshow(im)

img = tf.keras.utils.load_img(
    img_path, target_size=(100, 100)
)
img_array = tf.keras.utils.img_to_array(img)
img_array = tf.expand_dims(img_array, 0) # Create a batch

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 12 calls to <function Model.make_predi
ct_function.<locals>.predict_function at 0x16018f060> 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 tenso
rs with different shapes, (3) passing Python objects instead of tensors. Fo
r (1), please define your @tf.function outside of the loop. For (2), @tf.fu
nction has reduce_retracing=True option that can avoid unnecessary retracin
g. For (3), please refer to https://www.tensorflow.org/guide/function#contr
olling_retracing and https://www.tensorflow.org/api_docs/python/tf/function
for  more details.
1/1 [==============================] - 0s 171ms/step
This image most likely belongs to animal_crossing with a 93.26 percent conf
idence.
```



In [ ]:
```
img_path = 'imgA.jpg'
im = io.read_file(img_path)
im = image.decode_jpeg(im,channels=3)
plt.imshow(im)

img = tf.keras.utils.load_img(
    img_path, target_size=(100, 100)
)
img_array = tf.keras.utils.img_to_array(img)
img_array = tf.expand_dims(img_array, 0) # Create a batch

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))
)
```

```
1/1 [==============================] - 0s 30ms/step
This image most likely belongs to animal_crossing with a 66.05 percent conf
idence.
```

We can see that the model did successfully classify the sets, however the performance was worse than the sequential model.

Transfer Learning

The last type of model we would like to train for the image classifier is a transfer learning model, which uses an existing architecture with a new data set.

In [ ]:
```
for image, _ in trainingImage.take(1):
  plt.figure(figsize=(10, 10))
  first_image = image[0]
  for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    augmented_image = data_augmentation(tf.expand_dims(first_image, 0))
    plt.imshow(augmented_image[0] / 255)
    plt.axis('off')
```

```
2023-04-16 15:16:23.539448: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate 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 [1278]
         [[{{node Placeholder/_0}}]]
2023-04-16 15:16:23.540341: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate 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 [1278]
         [[{{node Placeholder/_0}}]]
```

We need to rescale the pixel values of the images we use to train the new model.

In [ ]:
```
preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input
rescale = tf.keras.layers.Rescaling(1./127.5, offset=-1)
```

Next we will create a base from a pre-trained model.

In [ ]:
```
# Create the base model from the pre-trained model MobileNet V2

IMG_SHAPE = (100,100) + (3,)
base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SHAPE,
                                               alpha=1.0,
                                               include_top=False,
                                               weights='imagenet')
```

```
WARNING:tensorflow:`input_shape` is undefined or non-square, or `rows` is n
ot in [96, 128, 160, 192, 224]. Weights for input shape (224, 224) will be
loaded as the default.
Downloading data from https://storage.googleapis.com/tensorflow/keras-appli
cations/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_22
4_no_top.h5
9406464/9406464 [==============================] - 0s 0us/step
```

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

```
2023-04-16 15:52:14.065846: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate an error and you can ignore this message): INVALID_ARGUMENT: You must
feed a value for placeholder tensor 'Placeholder/_4' with dtype int32 and s
hape [1278]
         [[{{node Placeholder/_4}}]]
2023-04-16 15:52:14.067723: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate 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 [1278]
         [[{{node Placeholder/_0}}]]
(32, 4, 4, 1280)
```

The next step will freeze the convolutional base to prevent weights from being updated.

In [ ]:
```
base_model.trainable = False
base_model.summary()
```

```
Model: "mobilenetv2_1.00_224"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected
to
==================================================================================================
 input_6 (InputLayer)           [(None, 100, 100, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 50, 50, 32)   864         ['input_6
[0][0]']

 bn_Conv1 (BatchNormalization)  (None, 50, 50, 32)   128         ['Conv1[0]
[0]']

 Conv1_relu (ReLU)              (None, 50, 50, 32)   0           ['bn_Conv1
[0][0]']
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected
to
==================================================================================================
 input_6 (InputLayer)           [(None, 100, 100, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 50, 50, 32)   864         ['input_6
[0][0]']

 bn_Conv1 (BatchNormalization)  (None, 50, 50, 32)   128         ['Conv1[0]
[0]']

 Conv1_relu (ReLU)              (None, 50, 50, 32)   0           ['bn_Conv1
[0][0]']

 expanded_conv_depthwise (Depth  (None, 50, 50, 32)  288         ['Conv1_re
lu[0][0]']
 wiseConv2D)

 expanded_conv_depthwise_BN (Ba  (None, 50, 50, 32)  128         ['expanded
_conv_depthwise[0][0]']
 tchNormalization)

 expanded_conv_depthwise_relu (  (None, 50, 50, 32)  0           ['expanded
_conv_depthwise_BN[0][0
 ReLU)                                                          ]']

 expanded_conv_project (Conv2D)  (None, 50, 50, 16)  512         ['expanded
_conv_depthwise_relu[0]
                                                                [0]']

 expanded_conv_project_BN (Batc  (None, 50, 50, 16)  64          ['expanded
_conv_project[0][0]']
 hNormalization)
```

```
 block_1_expand (Conv2D)        (None, 50, 50, 96)   1536        ['expanded
_conv_project_BN[0][0]'
                                                                 ]

 block_1_expand_BN (BatchNormal  (None, 50, 50, 96)  384         ['block_1_
expand[0][0]']
 ization)

 block_1_expand_relu (ReLU)     (None, 50, 50, 96)   0           ['block_1_
expand_BN[0][0]']

 block_1_pad (ZeroPadding2D)    (None, 51, 51, 96)   0           ['block_1_
expand_relu[0][0]']

 block_1_depthwise (DepthwiseCo  (None, 25, 25, 96)  864         ['block_1_
pad[0][0]']
 nv2D)

 block_1_depthwise_BN (BatchNor  (None, 25, 25, 96)  384         ['block_1_
depthwise[0][0]']
 malization)

 block_1_depthwise_relu (ReLU)  (None, 25, 25, 96)   0           ['block_1_
depthwise_BN[0][0]']

 block_1_project (Conv2D)       (None, 25, 25, 24)   2304        ['block_1_
depthwise_relu[0][0]']

 block_1_project_BN (BatchNorma  (None, 25, 25, 24)  96          ['block_1_
project[0][0]']
 lization)

 block_2_expand (Conv2D)        (None, 25, 25, 144)  3456        ['block_1_
project_BN[0][0]']

 block_2_expand_BN (BatchNormal  (None, 25, 25, 144)  576        ['block_2_
expand[0][0]']
 ization)

 block_2_expand_relu (ReLU)     (None, 25, 25, 144)  0           ['block_2_
expand_BN[0][0]']

 block_2_depthwise (DepthwiseCo  (None, 25, 25, 144)  1296       ['block_2_
expand_relu[0][0]']
 nv2D)

 block_2_depthwise_BN (BatchNor  (None, 25, 25, 144)  576        ['block_2_
depthwise[0][0]']
 malization)

 block_2_depthwise_relu (ReLU)  (None, 25, 25, 144)  0           ['block_2_
depthwise_BN[0][0]']

 block_2_project (Conv2D)       (None, 25, 25, 24)   3456        ['block_2_
depthwise_relu[0][0]']
```

```
 block_2_project_BN (BatchNorma  (None, 25, 25, 24)  96          ['block_2_
project[0][0]']
 lization)

 block_2_add (Add)              (None, 25, 25, 24)   0           ['block_1_
project_BN[0][0]',
                                                                 'block_2_
project_BN[0][0]']

 block_3_expand (Conv2D)        (None, 25, 25, 144)  3456        ['block_2_
add[0][0]']

 block_3_expand_BN (BatchNormal  (None, 25, 25, 144)  576        ['block_3_
expand[0][0]']
 ization)

 block_3_expand_relu (ReLU)     (None, 25, 25, 144)  0           ['block_3_
expand_BN[0][0]']

 block_3_pad (ZeroPadding2D)    (None, 27, 27, 144)  0           ['block_3_
expand_relu[0][0]']

 block_3_depthwise (DepthwiseCo  (None, 13, 13, 144)  1296       ['block_3_
pad[0][0]']
 nv2D)

 block_3_depthwise_BN (BatchNor  (None, 13, 13, 144)  576        ['block_3_
depthwise[0][0]']
 malization)

 block_3_depthwise_relu (ReLU)  (None, 13, 13, 144)  0           ['block_3_
depthwise_BN[0][0]']

 block_3_project (Conv2D)       (None, 13, 13, 32)   4608        ['block_3_
depthwise_relu[0][0]']

 block_3_project_BN (BatchNorma  (None, 13, 13, 32)  128         ['block_3_
project[0][0]']
 lization)

 block_4_expand (Conv2D)        (None, 13, 13, 192)  6144        ['block_3_
project_BN[0][0]']

 block_4_expand_BN (BatchNormal  (None, 13, 13, 192)  768        ['block_4_
expand[0][0]']
 ization)

 block_4_expand_relu (ReLU)     (None, 13, 13, 192)  0           ['block_4_
expand_BN[0][0]']

 block_4_depthwise (DepthwiseCo  (None, 13, 13, 192)  1728       ['block_4_
expand_relu[0][0]']
 nv2D)

 block_4_depthwise_BN (BatchNor  (None, 13, 13, 192)  768        ['block_4_
depthwise[0][0]']
```

```
 malization)

 block_4_depthwise_relu (ReLU)  (None, 13, 13, 192)  0           ['block_4_
depthwise_BN[0][0]']

 block_4_project (Conv2D)       (None, 13, 13, 32)   6144        ['block_4_
depthwise_relu[0][0]']

 block_4_project_BN (BatchNorma  (None, 13, 13, 32)  128         ['block_4_
project[0][0]']
 lization)

 block_4_add (Add)              (None, 13, 13, 32)   0           ['block_3_
project_BN[0][0]',
                                                                 'block_4_
project_BN[0][0]']

 block_5_expand (Conv2D)        (None, 13, 13, 192)  6144        ['block_4_
add[0][0]']

 block_5_expand_BN (BatchNormal  (None, 13, 13, 192)  768        ['block_5_
expand[0][0]']
 ization)

 block_5_expand_relu (ReLU)     (None, 13, 13, 192)  0           ['block_5_
expand_BN[0][0]']

 block_5_depthwise (DepthwiseCo  (None, 13, 13, 192)  1728       ['block_5_
expand_relu[0][0]']
 nv2D)

 block_5_depthwise_BN (BatchNor  (None, 13, 13, 192)  768        ['block_5_
depthwise[0][0]']
 malization)

 block_5_depthwise_relu (ReLU)  (None, 13, 13, 192)  0           ['block_5_
depthwise_BN[0][0]']

 block_5_project (Conv2D)       (None, 13, 13, 32)   6144        ['block_5_
depthwise_relu[0][0]']

 block_5_project_BN (BatchNorma  (None, 13, 13, 32)  128         ['block_5_
project[0][0]']
 lization)

 block_5_add (Add)              (None, 13, 13, 32)   0           ['block_4_
add[0][0]',
                                                                 'block_5_
project_BN[0][0]']

 block_6_expand (Conv2D)        (None, 13, 13, 192)  6144        ['block_5_
add[0][0]']

 block_6_expand_BN (BatchNormal  (None, 13, 13, 192)  768        ['block_6_
expand[0][0]']
 ization)
```

```
 block_6_expand_relu (ReLU)     (None, 13, 13, 192)  0           ['block_6_
expand_BN[0][0]']

 block_6_pad (ZeroPadding2D)    (None, 15, 15, 192)  0           ['block_6_
expand_relu[0][0]']

 block_6_depthwise (DepthwiseCo  (None, 7, 7, 192)   1728        ['block_6_
pad[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor  (None, 7, 7, 192)   768         ['block_6_
depthwise[0][0]']
 malization)

 block_6_depthwise_relu (ReLU)  (None, 7, 7, 192)    0           ['block_6_
depthwise_BN[0][0]']

 block_6_project (Conv2D)       (None, 7, 7, 64)     12288       ['block_6_
depthwise_relu[0][0]']

 block_6_project_BN (BatchNorma  (None, 7, 7, 64)    256         ['block_6_
project[0][0]']
 lization)

 block_7_expand (Conv2D)        (None, 7, 7, 384)    24576       ['block_6_
project_BN[0][0]']

 block_7_expand_BN (BatchNormal  (None, 7, 7, 384)   1536        ['block_7_
expand[0][0]']
 ization)

 block_7_expand_relu (ReLU)     (None, 7, 7, 384)    0           ['block_7_
expand_BN[0][0]']

 block_7_depthwise (DepthwiseCo  (None, 7, 7, 384)   3456        ['block_7_
expand_relu[0][0]']
 nv2D)

 block_7_depthwise_BN (BatchNor  (None, 7, 7, 384)   1536        ['block_7_
depthwise[0][0]']
 malization)

 block_7_depthwise_relu (ReLU)  (None, 7, 7, 384)    0           ['block_7_
depthwise_BN[0][0]']

 block_7_project (Conv2D)       (None, 7, 7, 64)     24576       ['block_7_
depthwise_relu[0][0]']

 block_7_project_BN (BatchNorma  (None, 7, 7, 64)    256         ['block_7_
project[0][0]']
 lization)

 block_7_add (Add)              (None, 7, 7, 64)     0           ['block_6_
project_BN[0][0]',
                                                                 'block_7_
```

```
project_BN[0][0]']

 block_8_expand (Conv2D)        (None, 7, 7, 384)    24576       ['block_7_
add[0][0]']

 block_8_expand_BN (BatchNormal  (None, 7, 7, 384)   1536        ['block_8_
expand[0][0]']
 ization)

 block_8_expand_relu (ReLU)     (None, 7, 7, 384)    0           ['block_8_
expand_BN[0][0]']

 block_8_depthwise (DepthwiseCo  (None, 7, 7, 384)   3456        ['block_8_
expand_relu[0][0]']
 nv2D)

 block_8_depthwise_BN (BatchNor  (None, 7, 7, 384)   1536        ['block_8_
depthwise[0][0]']
 malization)

 block_8_depthwise_relu (ReLU)  (None, 7, 7, 384)    0           ['block_8_
depthwise_BN[0][0]']

 block_8_project (Conv2D)       (None, 7, 7, 64)     24576       ['block_8_
depthwise_relu[0][0]']

 block_8_project_BN (BatchNorma  (None, 7, 7, 64)    256         ['block_8_
project[0][0]']
 lization)

 block_8_add (Add)              (None, 7, 7, 64)     0           ['block_7_
add[0][0]',
                                                                 'block_8_
project_BN[0][0]']

 block_9_expand (Conv2D)        (None, 7, 7, 384)    24576       ['block_8_
add[0][0]']

 block_9_expand_BN (BatchNormal  (None, 7, 7, 384)   1536        ['block_9_
expand[0][0]']
 ization)

 block_9_expand_relu (ReLU)     (None, 7, 7, 384)    0           ['block_9_
expand_BN[0][0]']

 block_9_depthwise (DepthwiseCo  (None, 7, 7, 384)   3456        ['block_9_
expand_relu[0][0]']
 nv2D)

 block_9_depthwise_BN (BatchNor  (None, 7, 7, 384)   1536        ['block_9_
depthwise[0][0]']
 malization)

 block_9_depthwise_relu (ReLU)  (None, 7, 7, 384)    0           ['block_9_
depthwise_BN[0][0]']
```

```
 block_9_project (Conv2D)       (None, 7, 7, 64)     24576       ['block_9_
depthwise_relu[0][0]']

 block_9_project_BN (BatchNorma  (None, 7, 7, 64)    256         ['block_9_
project[0][0]']
 lization)

 block_9_add (Add)              (None, 7, 7, 64)     0           ['block_8_
add[0][0]',
                                                                  'block_9_
project_BN[0][0]']

 block_10_expand (Conv2D)       (None, 7, 7, 384)    24576       ['block_9_
add[0][0]']

 block_10_expand_BN (BatchNorma  (None, 7, 7, 384)   1536        ['block_10
_expand[0][0]']
 lization)

 block_10_expand_relu (ReLU)    (None, 7, 7, 384)    0           ['block_10
_expand_BN[0][0]']

 block_10_depthwise (DepthwiseC  (None, 7, 7, 384)   3456        ['block_10
_expand_relu[0][0]']
 onv2D)

 block_10_depthwise_BN (BatchNo  (None, 7, 7, 384)   1536        ['block_10
_depthwise[0][0]']
 rmalization)

 block_10_depthwise_relu (ReLU)  (None, 7, 7, 384)   0           ['block_10
_depthwise_BN[0][0]']

 block_10_project (Conv2D)      (None, 7, 7, 96)     36864       ['block_10
_depthwise_relu[0][0]']

 block_10_project_BN (BatchNorm  (None, 7, 7, 96)    384         ['block_10
_project[0][0]']
 alization)

 block_11_expand (Conv2D)       (None, 7, 7, 576)    55296       ['block_10
_project_BN[0][0]']

 block_11_expand_BN (BatchNorma  (None, 7, 7, 576)   2304        ['block_11
_expand[0][0]']
 lization)

 block_11_expand_relu (ReLU)    (None, 7, 7, 576)    0           ['block_11
_expand_BN[0][0]']

 block_11_depthwise (DepthwiseC  (None, 7, 7, 576)   5184        ['block_11
_expand_relu[0][0]']
 onv2D)

 block_11_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304        ['block_11
_depthwise[0][0]']
```

```
 rmalization)

 block_11_depthwise_relu (ReLU)  (None, 7, 7, 576)   0           ['block_11
_depthwise_BN[0][0]']

 block_11_project (Conv2D)      (None, 7, 7, 96)     55296       ['block_11
_depthwise_relu[0][0]']

 block_11_project_BN (BatchNorm  (None, 7, 7, 96)    384         ['block_11
_project[0][0]']
 alization)

 block_11_add (Add)             (None, 7, 7, 96)     0           ['block_10
_project_BN[0][0]',
                                                                  'block_11
_project_BN[0][0]']

 block_12_expand (Conv2D)       (None, 7, 7, 576)    55296       ['block_11
_add[0][0]']

 block_12_expand_BN (BatchNorma  (None, 7, 7, 576)   2304        ['block_12
_expand[0][0]']
 lization)

 block_12_expand_relu (ReLU)    (None, 7, 7, 576)    0           ['block_12
_expand_BN[0][0]']

 block_12_depthwise (DepthwiseC  (None, 7, 7, 576)   5184        ['block_12
_expand_relu[0][0]']
 onv2D)

 block_12_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304        ['block_12
_depthwise[0][0]']
 rmalization)

 block_12_depthwise_relu (ReLU)  (None, 7, 7, 576)   0           ['block_12
_depthwise_BN[0][0]']

 block_12_project (Conv2D)      (None, 7, 7, 96)     55296       ['block_12
_depthwise_relu[0][0]']

 block_12_project_BN (BatchNorm  (None, 7, 7, 96)    384         ['block_12
_project[0][0]']
 alization)

 block_12_add (Add)             (None, 7, 7, 96)     0           ['block_11
_add[0][0]',
                                                                  'block_12
_project_BN[0][0]']

 block_13_expand (Conv2D)       (None, 7, 7, 576)    55296       ['block_12
_add[0][0]']

 block_13_expand_BN (BatchNorma  (None, 7, 7, 576)   2304        ['block_13
_expand[0][0]']
 lization)
```

```
 block_13_expand_relu (ReLU)    (None, 7, 7, 576)    0           ['block_13
_expand_BN[0][0]']

 block_13_pad (ZeroPadding2D)   (None, 9, 9, 576)    0           ['block_13
_expand_relu[0][0]']

 block_13_depthwise (DepthwiseC  (None, 4, 4, 576)   5184        ['block_13
_pad[0][0]']
 onv2D)

 block_13_depthwise_BN (BatchNo  (None, 4, 4, 576)   2304        ['block_13
_depthwise[0][0]']
 rmalization)

 block_13_depthwise_relu (ReLU)  (None, 4, 4, 576)   0           ['block_13
_depthwise_BN[0][0]']

 block_13_project (Conv2D)      (None, 4, 4, 160)    92160       ['block_13
_depthwise_relu[0][0]']

 block_13_project_BN (BatchNorm  (None, 4, 4, 160)   640         ['block_13
_project[0][0]']
 alization)

 block_14_expand (Conv2D)       (None, 4, 4, 960)    153600      ['block_13
_project_BN[0][0]']

 block_14_expand_BN (BatchNorma  (None, 4, 4, 960)   3840        ['block_14
_expand[0][0]']
 lization)

 block_14_expand_relu (ReLU)    (None, 4, 4, 960)    0           ['block_14
_expand_BN[0][0]']

 block_14_depthwise (DepthwiseC  (None, 4, 4, 960)   8640        ['block_14
_expand_relu[0][0]']
 onv2D)

 block_14_depthwise_BN (BatchNo  (None, 4, 4, 960)   3840        ['block_14
_depthwise[0][0]']
 rmalization)

 block_14_depthwise_relu (ReLU)  (None, 4, 4, 960)   0           ['block_14
_depthwise_BN[0][0]']

 block_14_project (Conv2D)      (None, 4, 4, 160)    153600      ['block_14
_depthwise_relu[0][0]']

 block_14_project_BN (BatchNorm  (None, 4, 4, 160)   640         ['block_14
_project[0][0]']
 alization)

 block_14_add (Add)             (None, 4, 4, 160)    0           ['block_13
_project_BN[0][0]',
                                                                  'block_14
```

```
_project_BN[0][0]']

 block_15_expand (Conv2D)       (None, 4, 4, 960)    153600      ['block_14
_add[0][0]']

 block_15_expand_BN (BatchNorma  (None, 4, 4, 960)   3840        ['block_15
_expand[0][0]']
 lization)

 block_15_expand_relu (ReLU)    (None, 4, 4, 960)    0           ['block_15
_expand_BN[0][0]']

 block_15_depthwise (DepthwiseC  (None, 4, 4, 960)   8640        ['block_15
_expand_relu[0][0]']
 onv2D)

 block_15_depthwise_BN (BatchNo  (None, 4, 4, 960)   3840        ['block_15
_depthwise[0][0]']
 rmalization)

 block_15_depthwise_relu (ReLU)  (None, 4, 4, 960)   0           ['block_15
_depthwise_BN[0][0]']

 block_15_project (Conv2D)      (None, 4, 4, 160)    153600      ['block_15
_depthwise_relu[0][0]']

 block_15_project_BN (BatchNorm  (None, 4, 4, 160)   640         ['block_15
_project[0][0]']
 alization)

 block_15_add (Add)             (None, 4, 4, 160)    0           ['block_14
_add[0][0]',
                                                                  'block_15
_project_BN[0][0]']

 block_16_expand (Conv2D)       (None, 4, 4, 960)    153600      ['block_15
_add[0][0]']

 block_16_expand_BN (BatchNorma  (None, 4, 4, 960)   3840        ['block_16
_expand[0][0]']
 lization)

 block_16_expand_relu (ReLU)    (None, 4, 4, 960)    0           ['block_16
_expand_BN[0][0]']

 block_16_depthwise (DepthwiseC  (None, 4, 4, 960)   8640        ['block_16
_expand_relu[0][0]']
 onv2D)

 block_16_depthwise_BN (BatchNo  (None, 4, 4, 960)   3840        ['block_16
_depthwise[0][0]']
 rmalization)

 block_16_depthwise_relu (ReLU)  (None, 4, 4, 960)   0           ['block_16
_depthwise_BN[0][0]']
```

```
 block_16_project (Conv2D)      (None, 4, 4, 320)    307200      ['block_16
_depthwise_relu[0][0]']

 block_16_project_BN (BatchNorm  (None, 4, 4, 320)   1280        ['block_16
_project[0][0]']
 alization)

 Conv_1 (Conv2D)                (None, 4, 4, 1280)   409600      ['block_16
_project_BN[0][0]']

 Conv_1_bn (BatchNormalization)  (None, 4, 4, 1280)  5120        ['Conv_1
[0][0]']

 out_relu (ReLU)                (None, 4, 4, 1280)   0           ['Conv_1_b
n[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)
```

Now we can build a model by chaining together the inputs from data augmentation and the base model we are using.

In [ ]:
```
inputs = tf.keras.Input(shape=(100, 100, 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)
model = tf.keras.Model(inputs, outputs)
```

The model can now be compiled and we can see an output summary.

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

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_7 (InputLayer)        [(None, 100, 100, 3)]     0


_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_7 (InputLayer)        [(None, 100, 100, 3)]     0

 sequential_9 (Sequential)   (None, 100, 100, 3)       0

 tf.math.truediv (TFOpLambda  (None, 100, 100, 3)      0
 )

 tf.math.subtract (TFOpLambd  (None, 100, 100, 3)      0
 a)

 mobilenetv2_1.00_224 (Funct  (None, 4, 4, 1280)       2257984
 ional)

 global_average_pooling2d (G  (None, 1280)             0
 lobalAveragePooling2D)

 dropout_5 (Dropout)         (None, 1280)              0

 dense_19 (Dense)            (None, 1)                 1281

=================================================================
Total params: 2,259,265
Trainable params: 1,281
Non-trainable params: 2,257,984
_________________________________________________________________
```

Training the Transfer Learning Model

We will now train the model with 20 epochs.

In [ ]:
```
initial_epochs = 20

loss0, accuracy0 = model.evaluate(validationImage)
```

```
2023-04-16 15:59:50.986326: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate 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 [319]
         [[{{node Placeholder/_0}}]]
2023-04-16 15:59:50.987685: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate 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 [319]
         [[{{node Placeholder/_0}}]]
2023-04-16 16:00:00.555708: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: extra compressed data
2023-04-16 16:00:01.204868: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 16:00:03.415221: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 16:00:03.415260: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: cHRM chunk does not match sRGB
8/8 [==============================] - 16s 243ms/step - loss: 1.0755 - accu
racy: 0.4941
```

We can now see the initial loss and accuracy of the model.

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

```
initial loss: 1.08
initial accuracy: 0.49
```

In [ ]:
```
history = model.fit(trainingImage,
                    epochs=initial_epochs,
                    validation_data=validationImage)
```

```
Epoch 1/20
40/40 [==============================] - 20s 349ms/step - loss: 0.8632 - ac
curacy: 0.5454 - val_loss: 0.8713 - val_accuracy: 0.5137
Epoch 2/20
40/40 [==============================] - 14s 340ms/step - loss: 0.8106 - ac
curacy: 0.5775 - val_loss: 0.7927 - val_accuracy: 0.5725
Epoch 3/20
40/40 [==============================] - 14s 348ms/step - loss: 0.7196 - ac
curacy: 0.6189 - val_loss: 0.7423 - val_accuracy: 0.6039
Epoch 4/20
40/40 [==============================] - 14s 345ms/step - loss: 0.6729 - ac
curacy: 0.6565 - val_loss: 0.6969 - val_accuracy: 0.6353
Epoch 5/20
40/40 [==============================] - 15s 373ms/step - loss: 0.6230 - ac
curacy: 0.6808 - val_loss: 0.6678 - val_accuracy: 0.6549
Epoch 6/20
40/40 [==============================] - 15s 364ms/step - loss: 0.6104 - ac
curacy: 0.6823 - val_loss: 0.6379 - val_accuracy: 0.6784
Epoch 7/20
40/40 [==============================] - 15s 364ms/step - loss: 0.5987 - ac
curacy: 0.7003 - val_loss: 0.6207 - val_accuracy: 0.6863
Epoch 8/20
40/40 [==============================] - 15s 373ms/step - loss: 0.5341 - ac
curacy: 0.7285 - val_loss: 0.5961 - val_accuracy: 0.6941
Epoch 9/20
40/40 [==============================] - 15s 365ms/step - loss: 0.5564 - ac
curacy: 0.7207 - val_loss: 0.5751 - val_accuracy: 0.7176
Epoch 10/20
40/40 [==============================] - 15s 371ms/step - loss: 0.5358 - ac
curacy: 0.7238 - val_loss: 0.5553 - val_accuracy: 0.7059
Epoch 11/20
40/40 [==============================] - 14s 358ms/step - loss: 0.5208 - ac
curacy: 0.7394 - val_loss: 0.5495 - val_accuracy: 0.7294
Epoch 12/20
40/40 [==============================] - 15s 367ms/step - loss: 0.5105 - ac
curacy: 0.7457 - val_loss: 0.5342 - val_accuracy: 0.7255
Epoch 13/20
40/40 [==============================] - 15s 387ms/step - loss: 0.5170 - ac
curacy: 0.7402 - val_loss: 0.5285 - val_accuracy: 0.7412
Epoch 14/20
40/40 [==============================] - 15s 366ms/step - loss: 0.4820 - ac
curacy: 0.7653 - val_loss: 0.5171 - val_accuracy: 0.7412
Epoch 15/20
40/40 [==============================] - 15s 372ms/step - loss: 0.4862 - ac
curacy: 0.7559 - val_loss: 0.5150 - val_accuracy: 0.7451
Epoch 16/20
40/40 [==============================] - 14s 358ms/step - loss: 0.4850 - ac
curacy: 0.7637 - val_loss: 0.5070 - val_accuracy: 0.7451
Epoch 17/20
40/40 [==============================] - 14s 356ms/step - loss: 0.4784 - ac
curacy: 0.7754 - val_loss: 0.4977 - val_accuracy: 0.7490
Epoch 18/20
40/40 [==============================] - 14s 358ms/step - loss: 0.4794 - ac
curacy: 0.7551 - val_loss: 0.4935 - val_accuracy: 0.7490
Epoch 19/20
40/40 [==============================] - 14s 354ms/step - loss: 0.4800 - ac
```

```
curacy: 0.7668 - val_loss: 0.4845 - val_accuracy: 0.7529
Epoch 20/20
40/40 [==============================] - 14s 358ms/step - loss: 0.4778 - ac
curacy: 0.7731 - val_loss: 0.4834 - val_accuracy: 0.7608
```

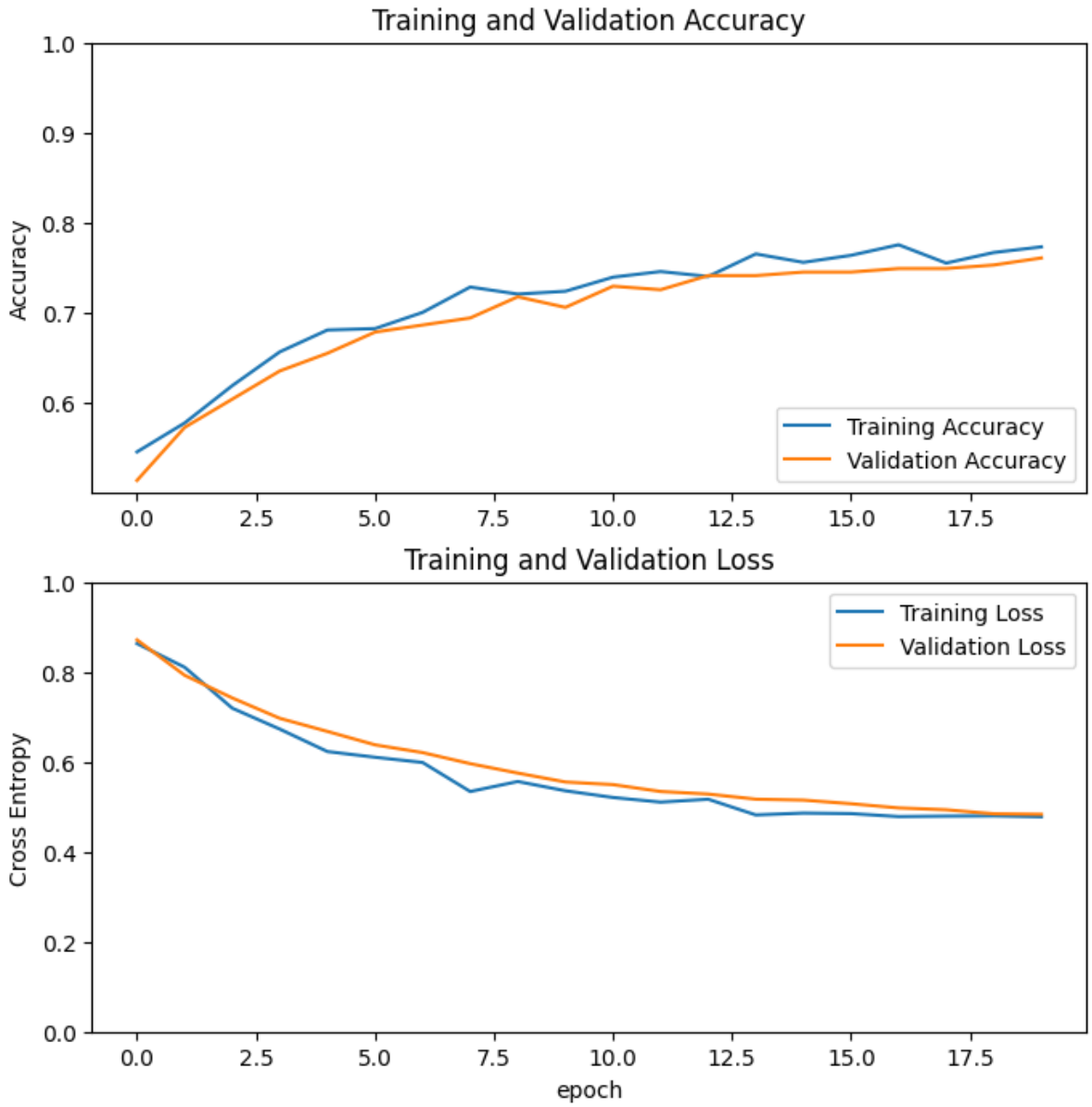
Evaluating the Transfer Learning Model

In [ ]:
```
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()
```



The validation mirrors the accuracy increases with each epoch as seen in the plot above.

Fine Tuning

The next step is to fine tune the model. The first step we will perform is unfreezing the weights of the base model.

In [ ]:
```
base_model.trainable = True
print("Number of layers in the base model: ", len(base_model.layers))
fine_tune_at = 100
for layer in base_model.layers[:fine_tune_at]:
  layer.trainable = False
```

```
Number of layers in the base model:  154
```

Recompile the Model

In [ ]:
```
model.compile(loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
              optimizer = tf.keras.optimizers.RMSprop(learning_rate=base_lea
              metrics=['accuracy'])
```

Continue Training the Model

In [ ]:
```
fine_tune_epochs = 20
total_epochs =  initial_epochs + fine_tune_epochs

history_fine = model.fit(trainingImage,
                         epochs=total_epochs,
                         initial_epoch=history.epoch[-1],
                         validation_data=validationImage)
```

```
Epoch 20/40
40/40 [==============================] - 32s 541ms/step - loss: 0.4723 - ac
curacy: 0.7700 - val_loss: 0.4396 - val_accuracy: 0.8039
Epoch 21/40
40/40 [==============================] - 21s 537ms/step - loss: 0.4156 - ac
curacy: 0.8114 - val_loss: 0.4280 - val_accuracy: 0.8196
Epoch 22/40
40/40 [==============================] - 22s 556ms/step - loss: 0.3641 - ac
curacy: 0.8271 - val_loss: 0.4368 - val_accuracy: 0.8314
Epoch 23/40
40/40 [==============================] - 23s 574ms/step - loss: 0.3736 - ac
curacy: 0.8286 - val_loss: 0.4617 - val_accuracy: 0.8196
Epoch 24/40
40/40 [==============================] - 23s 581ms/step - loss: 0.3477 - ac
curacy: 0.8365 - val_loss: 0.3926 - val_accuracy: 0.8392
Epoch 25/40
40/40 [==============================] - 23s 573ms/step - loss: 0.3184 - ac
curacy: 0.8529 - val_loss: 0.4073 - val_accuracy: 0.8196
Epoch 26/40
40/40 [==============================] - 23s 574ms/step - loss: 0.3055 - ac
curacy: 0.8560 - val_loss: 0.3989 - val_accuracy: 0.8235
Epoch 27/40
40/40 [==============================] - 23s 581ms/step - loss: 0.3126 - ac
curacy: 0.8490 - val_loss: 0.4136 - val_accuracy: 0.8157
Epoch 28/40
40/40 [==============================] - 23s 575ms/step - loss: 0.2858 - ac
curacy: 0.8662 - val_loss: 0.4024 - val_accuracy: 0.8275
Epoch 29/40
40/40 [==============================] - 23s 580ms/step - loss: 0.2709 - ac
curacy: 0.8842 - val_loss: 0.3829 - val_accuracy: 0.8392
Epoch 30/40
40/40 [==============================] - 23s 571ms/step - loss: 0.2878 - ac
curacy: 0.8709 - val_loss: 0.4741 - val_accuracy: 0.8353
Epoch 31/40
40/40 [==============================] - 25s 622ms/step - loss: 0.2598 - ac
curacy: 0.8858 - val_loss: 0.4527 - val_accuracy: 0.8431
Epoch 32/40
40/40 [==============================] - 23s 581ms/step - loss: 0.2477 - ac
curacy: 0.8975 - val_loss: 0.4505 - val_accuracy: 0.8549
Epoch 33/40
40/40 [==============================] - 23s 563ms/step - loss: 0.2431 - ac
curacy: 0.8889 - val_loss: 0.4364 - val_accuracy: 0.8431
Epoch 34/40
40/40 [==============================] - 23s 561ms/step - loss: 0.2390 - ac
curacy: 0.8912 - val_loss: 0.4650 - val_accuracy: 0.8314
Epoch 35/40
40/40 [==============================] - 22s 558ms/step - loss: 0.2290 - ac
curacy: 0.9022 - val_loss: 0.4211 - val_accuracy: 0.8392
Epoch 36/40
40/40 [==============================] - 22s 558ms/step - loss: 0.2215 - ac
curacy: 0.9030 - val_loss: 0.3988 - val_accuracy: 0.8392
Epoch 37/40
40/40 [==============================] - 22s 560ms/step - loss: 0.2023 - ac
curacy: 0.9069 - val_loss: 0.5697 - val_accuracy: 0.7922
Epoch 38/40
40/40 [==============================] - 22s 548ms/step - loss: 0.2148 - ac
```

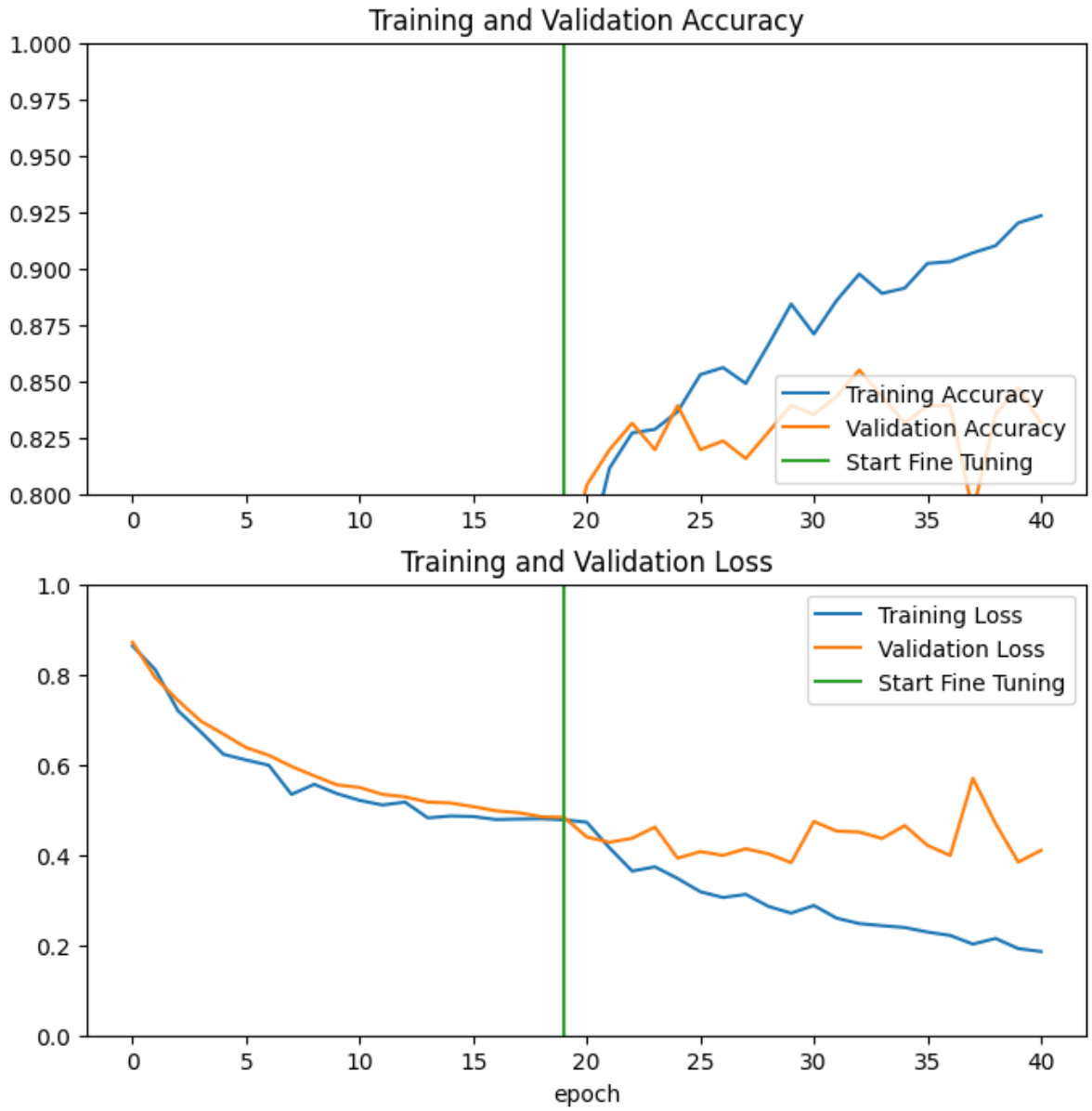
```
curacy: 0.9100 - val_loss: 0.4692 - val_accuracy: 0.8353
Epoch 39/40
40/40 [==============================] - 22s 544ms/step - loss: 0.1922 - ac
curacy: 0.9202 - val_loss: 0.3841 - val_accuracy: 0.8471
Epoch 40/40
40/40 [==============================] - 22s 548ms/step - loss: 0.1859 - ac
curacy: 0.9233 - val_loss: 0.4100 - val_accuracy: 0.8314
```

In [ ]:
```
acc += history_fine.history['accuracy']
val_acc += history_fine.history['val_accuracy']

loss += history_fine.history['loss']
val_loss += history_fine.history['val_loss']
```

In [ ]:
```
plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.ylim([0.8, 1])
plt.plot([initial_epochs-1,initial_epochs-1],
          plt.ylim(), label='Start Fine Tuning')
plt.legend(loc='lower right')
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.ylim([0, 1.0])
plt.plot([initial_epochs-1,initial_epochs-1],
         plt.ylim(), label='Start Fine Tuning')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.show()
```



Evaluate Model

In [ ]:
```
loss, accuracy = model.evaluate(testImage)
print('Test accuracy :', accuracy)
```

```
2023-04-16 16:25:56.379854: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate an error and you can ignore this message): INVALID_ARGUMENT: You must
feed a value for placeholder tensor 'Placeholder/_4' with dtype int32 and s
hape [319]
         [[{{node Placeholder/_4}}]]
2023-04-16 16:25:56.380709: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate an error and you can ignore this message): INVALID_ARGUMENT: You must
feed a value for placeholder tensor 'Placeholder/_4' with dtype int32 and s
hape [319]
         [[{{node Placeholder/_4}}]]
2023-04-16 16:26:03.531106: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: extra compressed data
2023-04-16 16:26:04.340295: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 16:26:06.325219: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 16:26:06.325262: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: cHRM chunk does not match sRGB
2/2 [==============================] - 12s 945ms/step - loss: 0.4267 - accu
racy: 0.7500
Test accuracy : 0.75
```

There is a slight reduction in accuracy when compared to the CNN model.

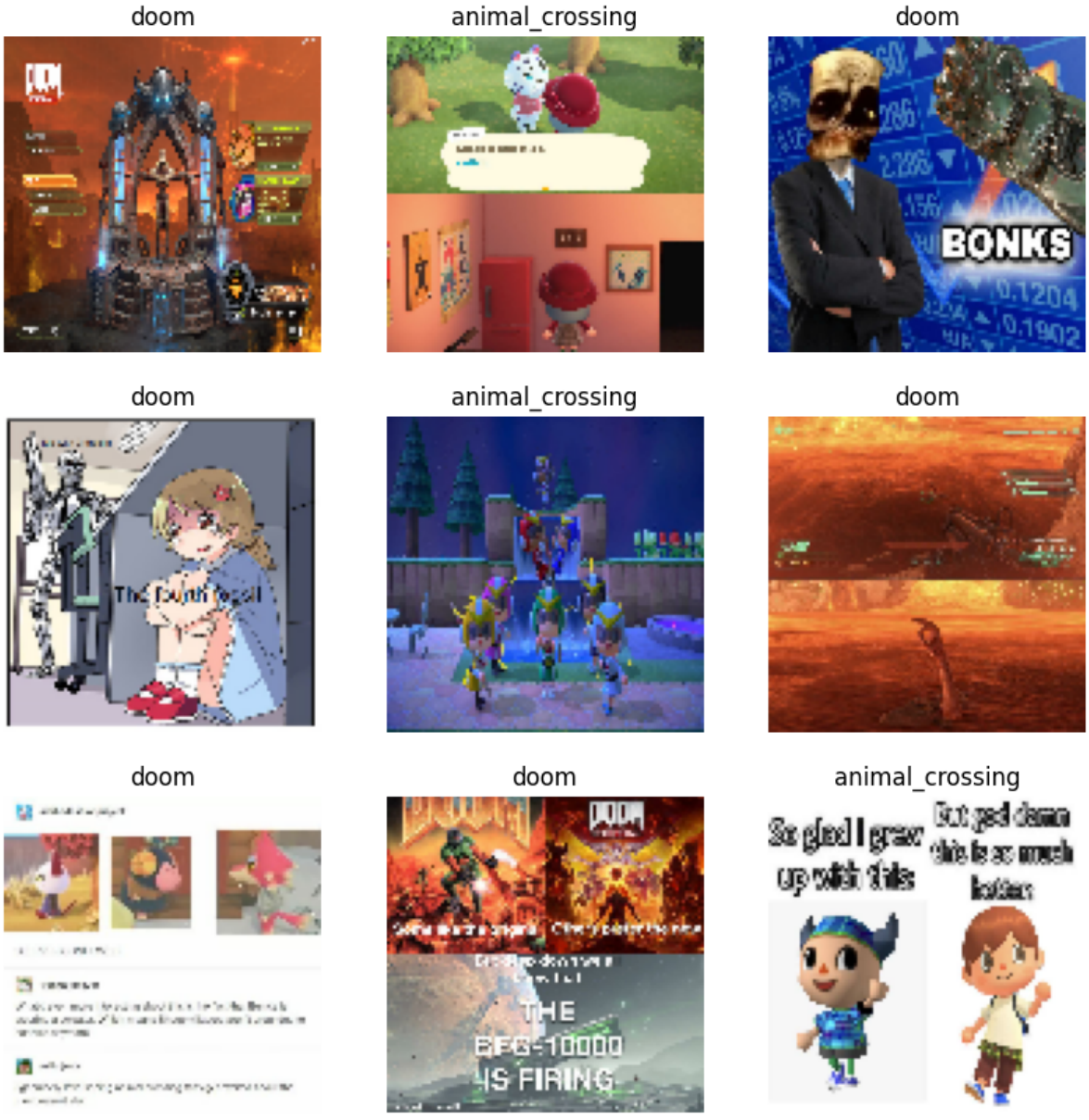
In [ ]:
```python
# Retrieve a batch of images from the test set
image_batch, label_batch = testImage.as_numpy_iterator().next()
predictions = model.predict_on_batch(image_batch).flatten()

# Apply a sigmoid since our model returns logits
predictions = tf.nn.sigmoid(predictions)
predictions = tf.where(predictions < 0.5, 0, 1)

print('Predictions:\n', predictions.numpy())
print('Labels:\n', label_batch)

plt.figure(figsize=(10, 10))
for i in range(9):
  ax = plt.subplot(3, 3, i + 1)
  plt.imshow(image_batch[i].astype("uint8"))
  plt.title(class_names[predictions[i]])
  plt.axis("off")
```

```
2023-04-16 16:28:00.325120: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: extra compressed data
2023-04-16 16:28:01.024892: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 16:28:03.036130: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: known incorrect sRGB profile
2023-04-16 16:28:03.036173: W tensorflow/core/lib/png/png_io.cc:88] PNG war
ning: iCCP: cHRM chunk does not match sRGB
2023-04-16 16:28:03.647205: I tensorflow/core/common_runtime/executor.cc:11
97] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not ind
icate an error and you can ignore this message): INVALID_ARGUMENT: You must
feed a value for placeholder tensor 'Placeholder/_0' with dtype float and s
hape [32,100,100,3]
         [[{{node Placeholder/_0}}]]
Predictions:
 [1 0 1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1 1 1 1 1 1 1 1 0 1]
Labels:
 [1 0 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0]
```



Analysis

The goal of this project was to be able to differentiate images associated with the video game franchises Doom and Animal Crossing. This was done by training three separate models using TensorFlow and Keras. The first model created is a sequential model that uses two dense hidden layers. The second is a convolutional model while the third model was a transfer learning model based on the MobileNetV2 pre-trained model. Looking more in depth at the architectures of each model, the sequential model utilized 2 hidden layers with 512 nodes each and an output layer with 2 nodes for the two classes to be identified. The hidden layers relied on the relu activation function, while the outer layer utilized a softmax activation. Next, the convolutional model contained three inner convolutional layers with, 16, 32, and 64 filters respectively. Another layer with 128 nodes is added in the end, followed by the outer layer with 2 nodes for the output classes. The activation functions used are the same as the. The third model is fine tuned for the dataset and is based on the MobileNetsV2 model. The data was divided using an 80/20 split for train and validation sets. The validation set was further divided to create a test set. Each model was fed the same data, with the top performing model being the convolutional model, followed by the transfer learning model, and in last place the sequential model.