U-Net (modified)

In [316]:
```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import pandas as pd
from tensorflow import keras
```

Data input pipeline ¶

In [26]:
```
# loading in the cleaned and normalized data
train_data_og = np.load("../../gcp/train-processed/train_x_l2a_clea
ned_norm.npy")
train_labels_og= np.load('../../gcp/train-processed/data_y_l2a_proc
essed.npy')
train_length = np.load('../../gcp/train-processed/length_l2a_proces
sed.npy')
test_data_og = np.load("../../gcp/test-processed/test_x_l2a_cleaned
_norm.npy")
test_labels_og = np.load('../../gcp/test-processed/test_y_l2a_proce
ssed.npy')
test_length = np.load('../../gcp/test-processed/test_length_l2a_pro
cessed.npy')
```

[https://www.tensorflow.org/tutorials/load_data/numpy (https://www.tensorflow.org/tutorials/load_data/numpy)](https://www.tensorflow.org/tutorials/load_data/numpy)

In [30]:
```
BATCH_SIZE = 64
SHUFFLE_BUFFER_SIZE = 100
TRAIN_LENGTH = len(train_length)
```

In [36]:
```
# selecting a single time-slice of the data
train_data = train_data_og[:,18,:,:,:]
test_data = test_data_og[:,18,:,:,:]

# create a Dataset using the tf.data API
train_dataset = tf.data.Dataset.from_tensor_slices((train_data, tra
in_labels_og))
test_dataset = tf.data.Dataset.from_tensor_slices((test_data, test_
labels_og))

# creating the random batches
train_dataset = train_dataset.shuffle(SHUFFLE_BUFFER_SIZE).batch(BA
TCH_SIZE)
test_dataset = test_dataset.batch(BATCH_SIZE)
```

In [37]:
```
train_dataset
```

Out[37]:
```
<BatchDataset shapes: ((None, 16, 16, 16), (None, 14, 14)), types:
(tf.float32, tf.float64)>
```

BUFFER_SIZE: This dataset fills a buffer with buffer_size elements, then randomly samples elements from this buffer, replacing the selected elements with new elements. For perfect shuffling, a buffer size greater than or equal to the full size of the dataset is required.

For instance, if your dataset contains 10,000 elements but buffer_size is set to 1,000, then shuffle will initially select a random element from only the first 1,000 elements in the buffer. Once an element is selected, its space in the buffer is replaced by the next (i.e. 1,001-st) element, maintaining the 1,000 element buffer. https://www.tensorflow.org/api_docs/python/tf/data/Dataset (https://www.tensorflow.org/api_docs/python/tf/data/Dataset)

Building the U-Net

In [33]:
```
# setting the number of output classes
# ? I could try with 1 and then classify by ><0.5 -> will try later
OUTPUT_CHANNELS = 2
```

below from: https://www.tensorflow.org/tutorials/generative/pix2pix (https://www.tensorflow.org/tutorials/generative/pix2pix)

In [52]:
```
def downsample(filters, size, stride=2, apply_batchnorm=True):
  initializer = tf.random_normal_initializer(0., 0.02)

  result = tf.keras.Sequential()
  result.add(
      tf.keras.layers.Conv2D(filters, size, strides=stride, padding
='same',
                             kernel_initializer=initializer, use_bi
as=False))

  if apply_batchnorm:
    result.add(tf.keras.layers.BatchNormalization())

  result.add(tf.keras.layers.LeakyReLU())

  return result
```

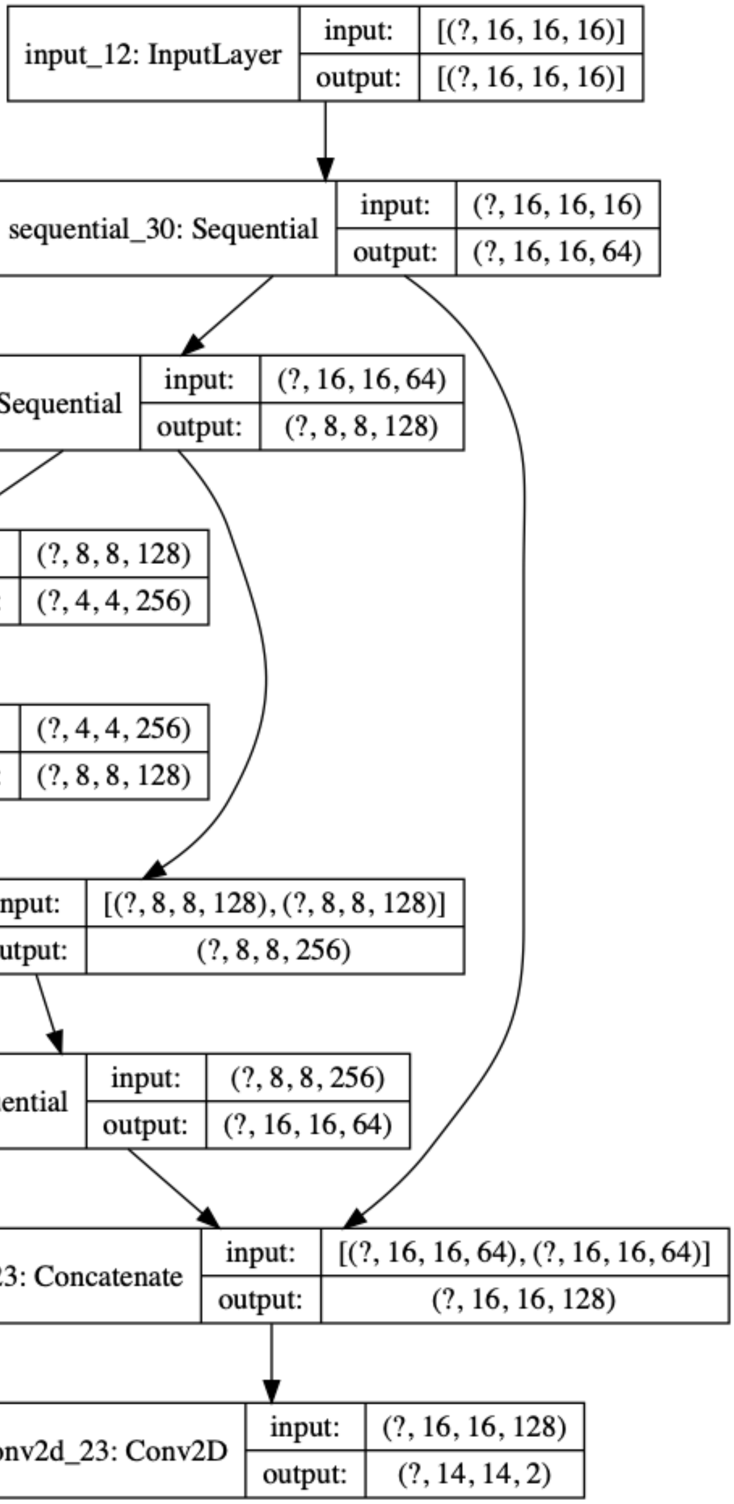
In [53]:
```
def upsample(filters, size, apply_dropout=False):
  initializer = tf.random_normal_initializer(0., 0.02)

  result = tf.keras.Sequential()
  result.add(
    tf.keras.layers.Conv2DTranspose(filters, size, strides=2,
                                    padding='same',
                                    kernel_initializer=initializer,
                                    use_bias=False))

  result.add(tf.keras.layers.BatchNormalization())

  if apply_dropout:
      result.add(tf.keras.layers.Dropout(0.5))

  result.add(tf.keras.layers.ReLU())

  return result
```

In [66]:
```
def unet_model(output_channels):
  inputs = tf.keras.layers.Input(shape=[16, 16, 16])

  down_stack = [
    downsample(64, 4, 1, apply_batchnorm=False), # (bs, 16,16, 64)
    downsample(128, 4, apply_batchnorm=False), # (bs, 8, 8, 128)
    downsample(256, 4), # (bs, 4, 4, 256)
  ]

  up_stack = [
    upsample(128, 4, apply_dropout=False), # (bs, 8, 8, 256)
    upsample(64, 4, apply_dropout=False), # (bs, 16, 16, 128)
  ]


  x = inputs

  # Downsampling through the model
  skips = []
  for down in down_stack:
    x = down(x)
    skips.append(x)

  skips = reversed(skips[:-1])

  # Upsampling and establishing the skip connections
  for up, skip in zip(up_stack, skips):
    x = up(x)
    concat = tf.keras.layers.Concatenate()
    x = concat([x, skip])

  # This is the last layer of the model
  last = tf.keras.layers.Conv2D(output_channels, 3, strides=1, padd
ing='valid')  #8x8 -> 14x14

  x = last(x)

  return tf.keras.Model(inputs=inputs, outputs=x)
```

In [67]:
```
unet = unet_model(2)
```

In [68]:
```
tf.keras.utils.plot_model(unet, show_shapes=True)
```

Out[68]:



In [ ]:

In [71]:

```
model = unet_model(OUTPUT_CHANNELS)
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(fr
om_logits=True),
              metrics=['accuracy', ])
```

In [77]:
```
EPOCHS = 300
VAL_SUBSPLITS = 5
VALIDATION_STEPS = len(test_length)//BATCH_SIZE//VAL_SUBSPLITS

model_history = model.fit(train_dataset, epochs=EPOCHS,
                          validation_steps=VALIDATION_STEPS,
                          validation_data=test_dataset)
```

```
Train for 137 steps, validate for 4 steps
Epoch 1/300
137/137 [==============================] - 17s 126ms/step - loss:
0.4470 - accuracy: 0.8090 - val_loss: 0.4699 - val_accuracy: 0.751
4
Epoch 2/300
137/137 [==============================] - 16s 120ms/step - loss:
0.4468 - accuracy: 0.8052 - val_loss: 0.6900 - val_accuracy: 0.605
6
Epoch 3/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4493 - accuracy: 0.8060 - val_loss: 0.8598 - val_accuracy: 0.563
5
Epoch 4/300
137/137 [==============================] - 16s 114ms/step - loss:
0.4434 - accuracy: 0.8079 - val_loss: 0.9856 - val_accuracy: 0.488
6
Epoch 5/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4449 - accuracy: 0.8108 - val_loss: 0.8051 - val_accuracy: 0.571
1
Epoch 6/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4419 - accuracy: 0.8085 - val_loss: 0.6651 - val_accuracy: 0.630
5
Epoch 7/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4366 - accuracy: 0.8114 - val_loss: 0.6133 - val_accuracy: 0.685
0
Epoch 8/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4355 - accuracy: 0.8114 - val_loss: 0.4953 - val_accuracy: 0.734
1
Epoch 9/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4302 - accuracy: 0.8132 - val_loss: 1.0764 - val_accuracy: 0.473
1
Epoch 10/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4357 - accuracy: 0.8097 - val_loss: 1.0689 - val_accuracy: 0.521
1
Epoch 11/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4307 - accuracy: 0.8136 - val_loss: 0.8537 - val_accuracy: 0.561
```

```
1
Epoch 12/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4342 - accuracy: 0.8121 - val_loss: 0.5873 - val_accuracy: 0.694
8
Epoch 13/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4303 - accuracy: 0.8166 - val_loss: 0.5790 - val_accuracy: 0.705
8
Epoch 14/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4237 - accuracy: 0.8185 - val_loss: 0.7493 - val_accuracy: 0.603
6
Epoch 15/300
137/137 [==============================] - 15s 111ms/step - loss:
0.4276 - accuracy: 0.8140 - val_loss: 0.6134 - val_accuracy: 0.674
6
Epoch 16/300
137/137 [==============================] - 16s 113ms/step - loss:
0.4290 - accuracy: 0.8153 - val_loss: 0.7993 - val_accuracy: 0.599
2
Epoch 17/300
137/137 [==============================] - 16s 114ms/step - loss:
0.4229 - accuracy: 0.8137 - val_loss: 0.7220 - val_accuracy: 0.625
5
Epoch 18/300
137/137 [==============================] - 15s 112ms/step - loss:
0.4278 - accuracy: 0.8154 - val_loss: 0.6697 - val_accuracy: 0.654
5
Epoch 19/300
137/137 [==============================] - 15s 113ms/step - loss:
0.4360 - accuracy: 0.8117 - val_loss: 0.5511 - val_accuracy: 0.714
0
Epoch 20/300
137/137 [==============================] - 16s 120ms/step - loss:
0.4203 - accuracy: 0.8203 - val_loss: 0.8454 - val_accuracy: 0.605
5
Epoch 21/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4284 - accuracy: 0.8155 - val_loss: 0.6784 - val_accuracy: 0.648
0
Epoch 22/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4260 - accuracy: 0.8167 - val_loss: 0.7652 - val_accuracy: 0.616
4
Epoch 23/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4173 - accuracy: 0.8199 - val_loss: 0.8706 - val_accuracy: 0.590
8
Epoch 24/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4213 - accuracy: 0.8177 - val_loss: 0.5219 - val_accuracy: 0.714
8
```

```
Epoch 25/300
137/137 [==============================] - 16s 117ms/step - loss:
0.4153 - accuracy: 0.8207 - val_loss: 0.6332 - val_accuracy: 0.644
6
Epoch 26/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4222 - accuracy: 0.8166 - val_loss: 0.7188 - val_accuracy: 0.640
8
Epoch 27/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4257 - accuracy: 0.8146 - val_loss: 0.9179 - val_accuracy: 0.574
1
Epoch 28/300
137/137 [==============================] - 16s 117ms/step - loss:
0.4218 - accuracy: 0.8183 - val_loss: 0.8792 - val_accuracy: 0.596
7
Epoch 29/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4228 - accuracy: 0.8167 - val_loss: 0.6654 - val_accuracy: 0.646
4
Epoch 30/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4158 - accuracy: 0.8228 - val_loss: 0.6252 - val_accuracy: 0.652
3
Epoch 31/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4173 - accuracy: 0.8218 - val_loss: 0.6302 - val_accuracy: 0.656
2
Epoch 32/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4154 - accuracy: 0.8207 - val_loss: 0.6603 - val_accuracy: 0.658
7
Epoch 33/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4200 - accuracy: 0.8178 - val_loss: 0.6154 - val_accuracy: 0.665
8
Epoch 34/300
137/137 [==============================] - 16s 117ms/step - loss:
0.4151 - accuracy: 0.8195 - val_loss: 0.7064 - val_accuracy: 0.614
9
Epoch 35/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4083 - accuracy: 0.8250 - val_loss: 0.6958 - val_accuracy: 0.615
3
Epoch 36/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4130 - accuracy: 0.8255 - val_loss: 0.6526 - val_accuracy: 0.647
7
Epoch 37/300
137/137 [==============================] - 16s 114ms/step - loss:
0.4134 - accuracy: 0.8219 - val_loss: 0.9147 - val_accuracy: 0.589
5
Epoch 38/300
```

```
137/137 [==============================] - 16s 114ms/step - loss:
0.4115 - accuracy: 0.8238 - val_loss: 0.6179 - val_accuracy: 0.663
7
Epoch 39/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4052 - accuracy: 0.8235 - val_loss: 0.6722 - val_accuracy: 0.651
4
Epoch 40/300
137/137 [==============================] - 16s 114ms/step - loss:
0.4115 - accuracy: 0.8223 - val_loss: 0.7652 - val_accuracy: 0.618
5
Epoch 41/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4038 - accuracy: 0.8259 - val_loss: 0.5680 - val_accuracy: 0.688
7
Epoch 42/300
137/137 [==============================] - 16s 114ms/step - loss:
0.4082 - accuracy: 0.8243 - val_loss: 0.9011 - val_accuracy: 0.552
5
Epoch 43/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3986 - accuracy: 0.8292 - val_loss: 0.9240 - val_accuracy: 0.554
8
Epoch 44/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4014 - accuracy: 0.8276 - val_loss: 0.8837 - val_accuracy: 0.543
4
Epoch 45/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4091 - accuracy: 0.8243 - val_loss: 1.2102 - val_accuracy: 0.472
6
Epoch 46/300
137/137 [==============================] - 16s 117ms/step - loss:
0.4019 - accuracy: 0.8278 - val_loss: 0.7077 - val_accuracy: 0.621
8
Epoch 47/300
137/137 [==============================] - 16s 117ms/step - loss:
0.4075 - accuracy: 0.8237 - val_loss: 1.3254 - val_accuracy: 0.477
8
Epoch 48/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4076 - accuracy: 0.8262 - val_loss: 1.3283 - val_accuracy: 0.477
8
Epoch 49/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3972 - accuracy: 0.8311 - val_loss: 1.3671 - val_accuracy: 0.474
6
Epoch 50/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3973 - accuracy: 0.8283 - val_loss: 1.4092 - val_accuracy: 0.493
6
Epoch 51/300
137/137 [==============================] - 16s 115ms/step - loss:
```

```
0.3985 - accuracy: 0.8309 - val_loss: 2.2403 - val_accuracy: 0.432
4
Epoch 52/300
137/137 [==============================] - 16s 116ms/step - loss:
0.4039 - accuracy: 0.8276 - val_loss: 1.2755 - val_accuracy: 0.536
4
Epoch 53/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3989 - accuracy: 0.8293 - val_loss: 1.5812 - val_accuracy: 0.484
1
Epoch 54/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4006 - accuracy: 0.8279 - val_loss: 1.6874 - val_accuracy: 0.479
2
Epoch 55/300
137/137 [==============================] - 16s 115ms/step - loss:
0.4003 - accuracy: 0.8276 - val_loss: 1.1945 - val_accuracy: 0.535
8
Epoch 56/300
137/137 [==============================] - 16s 114ms/step - loss:
0.4051 - accuracy: 0.8255 - val_loss: 1.6273 - val_accuracy: 0.474
9
Epoch 57/300
137/137 [==============================] - 16s 114ms/step - loss:
0.4011 - accuracy: 0.8271 - val_loss: 1.2659 - val_accuracy: 0.506
4
Epoch 58/300
137/137 [==============================] - 16s 114ms/step - loss:
0.3965 - accuracy: 0.8305 - val_loss: 1.8539 - val_accuracy: 0.416
5
Epoch 59/300
137/137 [==============================] - 16s 114ms/step - loss:
0.3944 - accuracy: 0.8315 - val_loss: 1.7858 - val_accuracy: 0.465
0
Epoch 60/300
137/137 [==============================] - 16s 114ms/step - loss:
0.3940 - accuracy: 0.8320 - val_loss: 1.6105 - val_accuracy: 0.420
0
Epoch 61/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3951 - accuracy: 0.8318 - val_loss: 1.7747 - val_accuracy: 0.370
5
Epoch 62/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3966 - accuracy: 0.8314 - val_loss: 1.8340 - val_accuracy: 0.416
2
Epoch 63/300
137/137 [==============================] - 16s 114ms/step - loss:
0.3935 - accuracy: 0.8319 - val_loss: 1.6210 - val_accuracy: 0.475
7
Epoch 64/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3925 - accuracy: 0.8326 - val_loss: 1.9863 - val_accuracy: 0.412
```

```
4
Epoch 65/300
137/137 [==============================] - 16s 114ms/step - loss:
0.4036 - accuracy: 0.8265 - val_loss: 1.3228 - val_accuracy: 0.460
5
Epoch 66/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3899 - accuracy: 0.8333 - val_loss: 1.5717 - val_accuracy: 0.443
6
Epoch 67/300
137/137 [==============================] - 16s 114ms/step - loss:
0.3905 - accuracy: 0.8317 - val_loss: 1.6719 - val_accuracy: 0.409
2
Epoch 68/300
137/137 [==============================] - 16s 114ms/step - loss:
0.3881 - accuracy: 0.8329 - val_loss: 1.7704 - val_accuracy: 0.439
0
Epoch 69/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3913 - accuracy: 0.8331 - val_loss: 2.1727 - val_accuracy: 0.397
6
Epoch 70/300
137/137 [==============================] - 16s 114ms/step - loss:
0.3939 - accuracy: 0.8323 - val_loss: 1.4417 - val_accuracy: 0.474
2
Epoch 71/300
137/137 [==============================] - 16s 118ms/step - loss:
0.3971 - accuracy: 0.8292 - val_loss: 1.6989 - val_accuracy: 0.478
4
Epoch 72/300
137/137 [==============================] - 16s 116ms/step - loss:
0.3941 - accuracy: 0.8322 - val_loss: 0.9800 - val_accuracy: 0.551
8
Epoch 73/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3927 - accuracy: 0.8320 - val_loss: 1.2685 - val_accuracy: 0.517
1
Epoch 74/300
137/137 [==============================] - 16s 116ms/step - loss:
0.3879 - accuracy: 0.8357 - val_loss: 0.7228 - val_accuracy: 0.615
5
Epoch 75/300
137/137 [==============================] - 16s 116ms/step - loss:
0.3956 - accuracy: 0.8314 - val_loss: 0.4406 - val_accuracy: 0.784
6
Epoch 76/300
137/137 [==============================] - 16s 115ms/step - loss:
0.3872 - accuracy: 0.8345 - val_loss: 0.4668 - val_accuracy: 0.761
4
Epoch 77/300
137/137 [==============================] - 16s 114ms/step - loss:
0.3908 - accuracy: 0.8337 - val_loss: 0.4494 - val_accuracy: 0.779
1
```

```
Epoch 78/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3849 - accuracy: 0.8362 - val_loss: 0.4685 - val_accuracy: 0.761
7
Epoch 79/300
137/137 [==============================] - 14s 105ms/step - loss:
0.3835 - accuracy: 0.8384 - val_loss: 0.5298 - val_accuracy: 0.717
1
Epoch 80/300
137/137 [==============================] - 15s 106ms/step - loss:
0.3873 - accuracy: 0.8343 - val_loss: 0.6263 - val_accuracy: 0.666
0
Epoch 81/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3862 - accuracy: 0.8356 - val_loss: 0.5808 - val_accuracy: 0.700
7
Epoch 82/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3745 - accuracy: 0.8422 - val_loss: 0.5133 - val_accuracy: 0.748
1
Epoch 83/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3758 - accuracy: 0.8396 - val_loss: 0.5615 - val_accuracy: 0.729
5
Epoch 84/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3878 - accuracy: 0.8333 - val_loss: 0.5458 - val_accuracy: 0.732
9
Epoch 85/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3735 - accuracy: 0.8411 - val_loss: 0.5850 - val_accuracy: 0.700
1
Epoch 86/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3800 - accuracy: 0.8396 - val_loss: 0.9263 - val_accuracy: 0.593
8
Epoch 87/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3732 - accuracy: 0.8417 - val_loss: 0.6322 - val_accuracy: 0.691
8
Epoch 88/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3758 - accuracy: 0.8406 - val_loss: 1.2348 - val_accuracy: 0.545
0
Epoch 89/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3782 - accuracy: 0.8387 - val_loss: 0.8686 - val_accuracy: 0.593
6
Epoch 90/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3751 - accuracy: 0.8407 - val_loss: 0.8953 - val_accuracy: 0.597
4
Epoch 91/300
```

```
137/137 [==============================] - 15s 113ms/step - loss:
0.3737 - accuracy: 0.8415 - val_loss: 0.8281 - val_accuracy: 0.624
3
Epoch 92/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3739 - accuracy: 0.8402 - val_loss: 0.9418 - val_accuracy: 0.608
1
Epoch 93/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3718 - accuracy: 0.8425 - val_loss: 0.9403 - val_accuracy: 0.590
4
Epoch 94/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3752 - accuracy: 0.8385 - val_loss: 1.0778 - val_accuracy: 0.590
0
Epoch 95/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3703 - accuracy: 0.8424 - val_loss: 0.8689 - val_accuracy: 0.613
0
Epoch 96/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3711 - accuracy: 0.8427 - val_loss: 0.9023 - val_accuracy: 0.598
3
Epoch 97/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3680 - accuracy: 0.8435 - val_loss: 1.0670 - val_accuracy: 0.574
9
Epoch 98/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3650 - accuracy: 0.8450 - val_loss: 0.8251 - val_accuracy: 0.628
8
Epoch 99/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3686 - accuracy: 0.8425 - val_loss: 1.3384 - val_accuracy: 0.521
1
Epoch 100/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3603 - accuracy: 0.8482 - val_loss: 0.8655 - val_accuracy: 0.594
4
Epoch 101/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3554 - accuracy: 0.8501 - val_loss: 1.6335 - val_accuracy: 0.505
4
Epoch 102/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3598 - accuracy: 0.8475 - val_loss: 2.1484 - val_accuracy: 0.464
0
Epoch 103/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3569 - accuracy: 0.8493 - val_loss: 1.7869 - val_accuracy: 0.484
9
Epoch 104/300
137/137 [==============================] - 15s 112ms/step - loss:
```

```
0.3586 - accuracy: 0.8496 - val_loss: 2.2067 - val_accuracy: 0.431
1
Epoch 105/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3498 - accuracy: 0.8534 - val_loss: 3.0760 - val_accuracy: 0.404
9
Epoch 106/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3555 - accuracy: 0.8497 - val_loss: 2.3710 - val_accuracy: 0.432
8
Epoch 107/300
137/137 [==============================] - 15s 113ms/step - loss:
0.3529 - accuracy: 0.8496 - val_loss: 1.4800 - val_accuracy: 0.510
1
Epoch 108/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3563 - accuracy: 0.8497 - val_loss: 2.2842 - val_accuracy: 0.366
0
Epoch 109/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3556 - accuracy: 0.8474 - val_loss: 2.0611 - val_accuracy: 0.432
8
Epoch 110/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3493 - accuracy: 0.8527 - val_loss: 2.4002 - val_accuracy: 0.389
0
Epoch 111/300
137/137 [==============================] - 15s 113ms/step - loss:
0.3575 - accuracy: 0.8498 - val_loss: 2.6304 - val_accuracy: 0.421
5
Epoch 112/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3491 - accuracy: 0.8545 - val_loss: 2.4909 - val_accuracy: 0.391
6
Epoch 113/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3500 - accuracy: 0.8525 - val_loss: 1.7791 - val_accuracy: 0.454
6
Epoch 114/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3496 - accuracy: 0.8534 - val_loss: 1.4518 - val_accuracy: 0.529
9
Epoch 115/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3520 - accuracy: 0.8517 - val_loss: 1.3466 - val_accuracy: 0.509
1
Epoch 116/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3518 - accuracy: 0.8528 - val_loss: 0.8743 - val_accuracy: 0.589
3
Epoch 117/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3519 - accuracy: 0.8518 - val_loss: 0.9872 - val_accuracy: 0.625
```

```
9
Epoch 118/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3456 - accuracy: 0.8545 - val_loss: 0.7644 - val_accuracy: 0.616
4
Epoch 119/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3463 - accuracy: 0.8542 - val_loss: 0.6235 - val_accuracy: 0.693
3
Epoch 120/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3483 - accuracy: 0.8532 - val_loss: 0.4817 - val_accuracy: 0.756
7
Epoch 121/300
137/137 [==============================] - 16s 113ms/step - loss:
0.3457 - accuracy: 0.8543 - val_loss: 0.6864 - val_accuracy: 0.655
0
Epoch 122/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3492 - accuracy: 0.8533 - val_loss: 0.5091 - val_accuracy: 0.744
6
Epoch 123/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3444 - accuracy: 0.8556 - val_loss: 0.4798 - val_accuracy: 0.762
1
Epoch 124/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3455 - accuracy: 0.8558 - val_loss: 0.6617 - val_accuracy: 0.691
6
Epoch 125/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3452 - accuracy: 0.8546 - val_loss: 0.7851 - val_accuracy: 0.647
6
Epoch 126/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3415 - accuracy: 0.8582 - val_loss: 0.9178 - val_accuracy: 0.620
1
Epoch 127/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3417 - accuracy: 0.8574 - val_loss: 0.7553 - val_accuracy: 0.663
1
Epoch 128/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3369 - accuracy: 0.8578 - val_loss: 0.8093 - val_accuracy: 0.659
6
Epoch 129/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3321 - accuracy: 0.8609 - val_loss: 0.7163 - val_accuracy: 0.691
9
Epoch 130/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3288 - accuracy: 0.8623 - val_loss: 0.8041 - val_accuracy: 0.686
3
```

```
Epoch 131/300
137/137 [==============================] - 15s 108ms/step - loss:
0.3419 - accuracy: 0.8577 - val_loss: 0.9208 - val_accuracy: 0.627
5
Epoch 132/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3299 - accuracy: 0.8629 - val_loss: 0.6392 - val_accuracy: 0.720
7
Epoch 133/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3394 - accuracy: 0.8574 - val_loss: 0.9589 - val_accuracy: 0.659
6
Epoch 134/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3306 - accuracy: 0.8624 - val_loss: 0.6947 - val_accuracy: 0.693
4
Epoch 135/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3304 - accuracy: 0.8623 - val_loss: 0.8899 - val_accuracy: 0.624
6
Epoch 136/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3256 - accuracy: 0.8638 - val_loss: 0.9278 - val_accuracy: 0.611
0
Epoch 137/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3221 - accuracy: 0.8662 - val_loss: 0.7707 - val_accuracy: 0.644
0
Epoch 138/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3214 - accuracy: 0.8659 - val_loss: 1.1069 - val_accuracy: 0.579
4
Epoch 139/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3212 - accuracy: 0.8663 - val_loss: 1.2196 - val_accuracy: 0.560
2
Epoch 140/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3166 - accuracy: 0.8693 - val_loss: 1.7738 - val_accuracy: 0.512
7
Epoch 141/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3201 - accuracy: 0.8673 - val_loss: 1.8007 - val_accuracy: 0.517
2
Epoch 142/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3209 - accuracy: 0.8664 - val_loss: 1.2329 - val_accuracy: 0.546
6
Epoch 143/300
137/137 [==============================] - 15s 109ms/step - loss:
0.3230 - accuracy: 0.8661 - val_loss: 1.3904 - val_accuracy: 0.515
9
Epoch 144/300
```

```
137/137 [==============================] - 15s 110ms/step - loss:
0.3217 - accuracy: 0.8678 - val_loss: 1.0921 - val_accuracy: 0.585
3
Epoch 145/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3138 - accuracy: 0.8709 - val_loss: 1.6019 - val_accuracy: 0.519
1
Epoch 146/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3179 - accuracy: 0.8687 - val_loss: 2.3049 - val_accuracy: 0.471
7
Epoch 147/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3217 - accuracy: 0.8667 - val_loss: 1.7994 - val_accuracy: 0.505
4
Epoch 148/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3157 - accuracy: 0.8695 - val_loss: 2.9591 - val_accuracy: 0.457
7
Epoch 149/300
137/137 [==============================] - 15s 110ms/step - loss:
0.3094 - accuracy: 0.8725 - val_loss: 1.6297 - val_accuracy: 0.511
7
Epoch 150/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3055 - accuracy: 0.8749 - val_loss: 1.4636 - val_accuracy: 0.523
6
Epoch 151/300
137/137 [==============================] - 16s 117ms/step - loss:
0.3035 - accuracy: 0.8750 - val_loss: 1.8095 - val_accuracy: 0.491
3
Epoch 152/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3020 - accuracy: 0.8767 - val_loss: 1.7303 - val_accuracy: 0.493
4
Epoch 153/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3013 - accuracy: 0.8765 - val_loss: 2.0792 - val_accuracy: 0.457
6
Epoch 154/300
137/137 [==============================] - 15s 112ms/step - loss:
0.3035 - accuracy: 0.8753 - val_loss: 1.4682 - val_accuracy: 0.547
5
Epoch 155/300
137/137 [==============================] - 15s 111ms/step - loss:
0.3042 - accuracy: 0.8749 - val_loss: 1.2810 - val_accuracy: 0.528
1
Epoch 156/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2892 - accuracy: 0.8828 - val_loss: 1.3085 - val_accuracy: 0.531
2
Epoch 157/300
137/137 [==============================] - 15s 111ms/step - loss:
```

```
0.2916 - accuracy: 0.8805 - val_loss: 1.0647 - val_accuracy: 0.578
8
Epoch 158/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2892 - accuracy: 0.8824 - val_loss: 0.7003 - val_accuracy: 0.679
1
Epoch 159/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2919 - accuracy: 0.8806 - val_loss: 0.6618 - val_accuracy: 0.694
8
Epoch 160/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2911 - accuracy: 0.8825 - val_loss: 0.5639 - val_accuracy: 0.744
1
Epoch 161/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2860 - accuracy: 0.8845 - val_loss: 0.6755 - val_accuracy: 0.700
1
Epoch 162/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2887 - accuracy: 0.8834 - val_loss: 0.6215 - val_accuracy: 0.736
8
Epoch 163/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2873 - accuracy: 0.8840 - val_loss: 0.5790 - val_accuracy: 0.750
2
Epoch 164/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2856 - accuracy: 0.8849 - val_loss: 0.7539 - val_accuracy: 0.665
7
Epoch 165/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2781 - accuracy: 0.8880 - val_loss: 0.6307 - val_accuracy: 0.718
4
Epoch 166/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2706 - accuracy: 0.8919 - val_loss: 0.7113 - val_accuracy: 0.712
0
Epoch 167/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2746 - accuracy: 0.8899 - val_loss: 0.5433 - val_accuracy: 0.764
6
Epoch 168/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2643 - accuracy: 0.8953 - val_loss: 0.5872 - val_accuracy: 0.736
4
Epoch 169/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2681 - accuracy: 0.8930 - val_loss: 0.9620 - val_accuracy: 0.635
8
Epoch 170/300
137/137 [==============================] - 16s 114ms/step - loss:
0.2730 - accuracy: 0.8904 - val_loss: 0.7589 - val_accuracy: 0.672
```

```
8
Epoch 171/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2694 - accuracy: 0.8924 - val_loss: 0.5618 - val_accuracy: 0.751
7
Epoch 172/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2719 - accuracy: 0.8915 - val_loss: 0.5900 - val_accuracy: 0.741
2
Epoch 173/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2700 - accuracy: 0.8917 - val_loss: 0.5663 - val_accuracy: 0.741
0
Epoch 174/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2751 - accuracy: 0.8902 - val_loss: 0.6294 - val_accuracy: 0.748
8
Epoch 175/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2655 - accuracy: 0.8945 - val_loss: 0.7180 - val_accuracy: 0.710
6
Epoch 176/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2618 - accuracy: 0.8952 - val_loss: 0.8088 - val_accuracy: 0.690
9
Epoch 177/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2579 - accuracy: 0.8976 - val_loss: 0.8490 - val_accuracy: 0.681
4
Epoch 178/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2603 - accuracy: 0.8956 - val_loss: 0.7304 - val_accuracy: 0.696
3
Epoch 179/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2561 - accuracy: 0.8985 - val_loss: 0.9434 - val_accuracy: 0.660
7
Epoch 180/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2560 - accuracy: 0.8982 - val_loss: 1.6727 - val_accuracy: 0.575
1
Epoch 181/300
137/137 [==============================] - 16s 113ms/step - loss:
0.2531 - accuracy: 0.8992 - val_loss: 0.9589 - val_accuracy: 0.642
5
Epoch 182/300
137/137 [==============================] - 15s 113ms/step - loss:
0.2521 - accuracy: 0.8999 - val_loss: 0.6305 - val_accuracy: 0.741
6
Epoch 183/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2505 - accuracy: 0.9000 - val_loss: 0.6726 - val_accuracy: 0.722
7
```

```
Epoch 184/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2461 - accuracy: 0.9022 - val_loss: 1.1795 - val_accuracy: 0.606
7
Epoch 185/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2456 - accuracy: 0.9028 - val_loss: 0.8395 - val_accuracy: 0.664
5
Epoch 186/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2426 - accuracy: 0.9045 - val_loss: 0.9202 - val_accuracy: 0.664
9
Epoch 187/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2387 - accuracy: 0.9052 - val_loss: 0.5681 - val_accuracy: 0.762
0
Epoch 188/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2432 - accuracy: 0.9049 - val_loss: 0.5718 - val_accuracy: 0.748
2
Epoch 189/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2368 - accuracy: 0.9064 - val_loss: 0.5676 - val_accuracy: 0.759
8
Epoch 190/300
137/137 [==============================] - 15s 113ms/step - loss:
0.2434 - accuracy: 0.9033 - val_loss: 0.5444 - val_accuracy: 0.752
9
Epoch 191/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2302 - accuracy: 0.9098 - val_loss: 0.5895 - val_accuracy: 0.760
6
Epoch 192/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2328 - accuracy: 0.9086 - val_loss: 0.5991 - val_accuracy: 0.735
3
Epoch 193/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2317 - accuracy: 0.9089 - val_loss: 0.5547 - val_accuracy: 0.761
5
Epoch 194/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2306 - accuracy: 0.9094 - val_loss: 0.6153 - val_accuracy: 0.727
9
Epoch 195/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2291 - accuracy: 0.9103 - val_loss: 0.5398 - val_accuracy: 0.757
6
Epoch 196/300
137/137 [==============================] - 15s 111ms/step - loss:
0.2243 - accuracy: 0.9120 - val_loss: 0.7471 - val_accuracy: 0.699
9
Epoch 197/300
```

```
137/137 [==============================] - 15s 112ms/step - loss: 
0.2240 - accuracy: 0.9125 - val_loss: 0.6282 - val_accuracy: 0.742
0
Epoch 198/300
137/137 [==============================] - 15s 111ms/step - loss: 
0.2320 - accuracy: 0.9084 - val_loss: 0.6401 - val_accuracy: 0.733
4
Epoch 199/300
137/137 [==============================] - 15s 111ms/step - loss: 
0.2239 - accuracy: 0.9125 - val_loss: 0.7800 - val_accuracy: 0.709
6
Epoch 200/300
137/137 [==============================] - 15s 110ms/step - loss: 
0.2195 - accuracy: 0.9148 - val_loss: 0.6670 - val_accuracy: 0.730
4
Epoch 201/300
137/137 [==============================] - 15s 111ms/step - loss: 
0.2190 - accuracy: 0.9143 - val_loss: 0.6218 - val_accuracy: 0.743
6
Epoch 202/300
137/137 [==============================] - 15s 111ms/step - loss: 
0.2137 - accuracy: 0.9172 - val_loss: 0.5512 - val_accuracy: 0.761
6
Epoch 203/300
137/137 [==============================] - 15s 111ms/step - loss: 
0.2215 - accuracy: 0.9128 - val_loss: 0.6609 - val_accuracy: 0.733
7
Epoch 204/300
137/137 [==============================] - 15s 111ms/step - loss: 
0.2169 - accuracy: 0.9150 - val_loss: 0.5787 - val_accuracy: 0.776
1
Epoch 205/300
137/137 [==============================] - 15s 113ms/step - loss: 
0.2108 - accuracy: 0.9179 - val_loss: 0.7512 - val_accuracy: 0.705
7
Epoch 206/300
137/137 [==============================] - 15s 113ms/step - loss: 
0.2079 - accuracy: 0.9191 - val_loss: 0.5485 - val_accuracy: 0.780
6
Epoch 207/300
137/137 [==============================] - 15s 113ms/step - loss: 
0.2087 - accuracy: 0.9185 - val_loss: 0.6248 - val_accuracy: 0.747
3
Epoch 208/300
137/137 [==============================] - 15s 112ms/step - loss: 
0.2156 - accuracy: 0.9155 - val_loss: 0.8191 - val_accuracy: 0.700
2
Epoch 209/300
137/137 [==============================] - 16s 113ms/step - loss: 
0.2096 - accuracy: 0.9175 - val_loss: 0.8379 - val_accuracy: 0.710
9
Epoch 210/300
137/137 [==============================] - 16s 115ms/step - loss: 
```

```
0.2061 - accuracy: 0.9196 - val_loss: 0.7280 - val_accuracy: 0.737
3
Epoch 211/300
137/137 [==============================] - 16s 113ms/step - loss:
0.2071 - accuracy: 0.9191 - val_loss: 0.6208 - val_accuracy: 0.742
9
Epoch 212/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2016 - accuracy: 0.9217 - val_loss: 0.6077 - val_accuracy: 0.765
0
Epoch 213/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2036 - accuracy: 0.9203 - val_loss: 0.6447 - val_accuracy: 0.748
6
Epoch 214/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1991 - accuracy: 0.9226 - val_loss: 0.8233 - val_accuracy: 0.707
9
Epoch 215/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2049 - accuracy: 0.9197 - val_loss: 0.7134 - val_accuracy: 0.741
8
Epoch 216/300
137/137 [==============================] - 15s 112ms/step - loss:
0.2065 - accuracy: 0.9186 - val_loss: 0.8713 - val_accuracy: 0.709
6
Epoch 217/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1999 - accuracy: 0.9221 - val_loss: 0.6925 - val_accuracy: 0.743
2
Epoch 218/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1962 - accuracy: 0.9237 - val_loss: 0.6351 - val_accuracy: 0.757
3
Epoch 219/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1986 - accuracy: 0.9229 - val_loss: 1.0523 - val_accuracy: 0.663
3
Epoch 220/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1944 - accuracy: 0.9246 - val_loss: 0.6929 - val_accuracy: 0.752
2
Epoch 221/300
137/137 [==============================] - 16s 114ms/step - loss:
0.1984 - accuracy: 0.9222 - val_loss: 0.7519 - val_accuracy: 0.745
3
Epoch 222/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1916 - accuracy: 0.9259 - val_loss: 0.8212 - val_accuracy: 0.721
7
Epoch 223/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1912 - accuracy: 0.9256 - val_loss: 0.8819 - val_accuracy: 0.714
```

```
8
Epoch 224/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1871 - accuracy: 0.9276 - val_loss: 0.8911 - val_accuracy: 0.713
8
Epoch 225/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1813 - accuracy: 0.9297 - val_loss: 0.9076 - val_accuracy: 0.707
5
Epoch 226/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1816 - accuracy: 0.9297 - val_loss: 0.8607 - val_accuracy: 0.721
0
Epoch 227/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1861 - accuracy: 0.9281 - val_loss: 0.7651 - val_accuracy: 0.750
4
Epoch 228/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1791 - accuracy: 0.9306 - val_loss: 0.7574 - val_accuracy: 0.745
1
Epoch 229/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1860 - accuracy: 0.9275 - val_loss: 1.2579 - val_accuracy: 0.660
6
Epoch 230/300
137/137 [==============================] - 16s 114ms/step - loss:
0.1781 - accuracy: 0.9309 - val_loss: 0.8312 - val_accuracy: 0.738
2
Epoch 231/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1758 - accuracy: 0.9319 - val_loss: 0.7733 - val_accuracy: 0.732
8
Epoch 232/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1749 - accuracy: 0.9320 - val_loss: 1.4052 - val_accuracy: 0.651
0
Epoch 233/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1822 - accuracy: 0.9292 - val_loss: 1.1135 - val_accuracy: 0.667
3
Epoch 234/300
137/137 [==============================] - 15s 113ms/step - loss:
0.1735 - accuracy: 0.9331 - val_loss: 0.8291 - val_accuracy: 0.730
3
Epoch 235/300
137/137 [==============================] - 16s 113ms/step - loss:
0.1710 - accuracy: 0.9341 - val_loss: 0.8926 - val_accuracy: 0.723
3
Epoch 236/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1705 - accuracy: 0.9340 - val_loss: 1.0075 - val_accuracy: 0.694
2
```

```
Epoch 237/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1646 - accuracy: 0.9363 - val_loss: 1.0468 - val_accuracy: 0.687
3
Epoch 238/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1637 - accuracy: 0.9370 - val_loss: 1.0851 - val_accuracy: 0.680
6
Epoch 239/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1662 - accuracy: 0.9351 - val_loss: 1.0715 - val_accuracy: 0.687
0
Epoch 240/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1621 - accuracy: 0.9371 - val_loss: 1.1957 - val_accuracy: 0.675
5
Epoch 241/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1679 - accuracy: 0.9347 - val_loss: 1.2349 - val_accuracy: 0.648
3
Epoch 242/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1615 - accuracy: 0.9372 - val_loss: 1.1350 - val_accuracy: 0.673
5
Epoch 243/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1629 - accuracy: 0.9367 - val_loss: 1.1904 - val_accuracy: 0.665
8
Epoch 244/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1641 - accuracy: 0.9361 - val_loss: 0.8247 - val_accuracy: 0.725
4
Epoch 245/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1629 - accuracy: 0.9367 - val_loss: 1.0698 - val_accuracy: 0.680
9
Epoch 246/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1585 - accuracy: 0.9383 - val_loss: 0.9630 - val_accuracy: 0.705
5
Epoch 247/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1562 - accuracy: 0.9394 - val_loss: 0.8343 - val_accuracy: 0.730
5
Epoch 248/300
137/137 [==============================] - 16s 114ms/step - loss:
0.1563 - accuracy: 0.9397 - val_loss: 1.0092 - val_accuracy: 0.703
0
Epoch 249/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1490 - accuracy: 0.9422 - val_loss: 1.0013 - val_accuracy: 0.723
1
Epoch 250/300
```

```
137/137 [==============================] - 15s 111ms/step - loss:
0.1554 - accuracy: 0.9392 - val_loss: 1.0575 - val_accuracy: 0.695
2
Epoch 251/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1489 - accuracy: 0.9422 - val_loss: 0.8292 - val_accuracy: 0.747
9
Epoch 252/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1536 - accuracy: 0.9399 - val_loss: 1.0061 - val_accuracy: 0.707
3
Epoch 253/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1602 - accuracy: 0.9374 - val_loss: 0.7325 - val_accuracy: 0.765
4
Epoch 254/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1513 - accuracy: 0.9412 - val_loss: 0.9843 - val_accuracy: 0.707
0
Epoch 255/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1583 - accuracy: 0.9384 - val_loss: 0.8504 - val_accuracy: 0.732
8
Epoch 256/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1488 - accuracy: 0.9421 - val_loss: 0.6070 - val_accuracy: 0.805
8
Epoch 257/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1459 - accuracy: 0.9433 - val_loss: 0.7649 - val_accuracy: 0.763
3
Epoch 258/300
137/137 [==============================] - 15s 110ms/step - loss:
0.1410 - accuracy: 0.9454 - val_loss: 0.9418 - val_accuracy: 0.730
6
Epoch 259/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1376 - accuracy: 0.9466 - val_loss: 0.8728 - val_accuracy: 0.729
2
Epoch 260/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1391 - accuracy: 0.9461 - val_loss: 0.8750 - val_accuracy: 0.744
2
Epoch 261/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1400 - accuracy: 0.9454 - val_loss: 0.7723 - val_accuracy: 0.753
7
Epoch 262/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1367 - accuracy: 0.9470 - val_loss: 0.9563 - val_accuracy: 0.725
9
Epoch 263/300
137/137 [==============================] - 15s 111ms/step - loss:
```

```
0.1364 - accuracy: 0.9469 - val_loss: 0.8495 - val_accuracy: 0.747
4
Epoch 264/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1360 - accuracy: 0.9468 - val_loss: 1.0365 - val_accuracy: 0.718
9
Epoch 265/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1348 - accuracy: 0.9476 - val_loss: 1.1359 - val_accuracy: 0.692
8
Epoch 266/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1376 - accuracy: 0.9461 - val_loss: 1.3252 - val_accuracy: 0.671
3
Epoch 267/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1382 - accuracy: 0.9459 - val_loss: 1.4250 - val_accuracy: 0.646
7
Epoch 268/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1447 - accuracy: 0.9442 - val_loss: 1.0422 - val_accuracy: 0.721
0
Epoch 269/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1313 - accuracy: 0.9489 - val_loss: 0.9316 - val_accuracy: 0.736
3
Epoch 270/300
137/137 [==============================] - 15s 110ms/step - loss:
0.1284 - accuracy: 0.9497 - val_loss: 0.9225 - val_accuracy: 0.747
3
Epoch 271/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1324 - accuracy: 0.9483 - val_loss: 0.8321 - val_accuracy: 0.763
5
Epoch 272/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1290 - accuracy: 0.9498 - val_loss: 0.7156 - val_accuracy: 0.783
6
Epoch 273/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1282 - accuracy: 0.9498 - val_loss: 0.8484 - val_accuracy: 0.747
8
Epoch 274/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1314 - accuracy: 0.9482 - val_loss: 1.1707 - val_accuracy: 0.688
9
Epoch 275/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1309 - accuracy: 0.9484 - val_loss: 0.7647 - val_accuracy: 0.770
5
Epoch 276/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1242 - accuracy: 0.9516 - val_loss: 0.8234 - val_accuracy: 0.755
```

```
5
Epoch 277/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1273 - accuracy: 0.9503 - val_loss: 0.8462 - val_accuracy: 0.760
2
Epoch 278/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1221 - accuracy: 0.9522 - val_loss: 0.9351 - val_accuracy: 0.738
2
Epoch 279/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1234 - accuracy: 0.9518 - val_loss: 0.7983 - val_accuracy: 0.757
9
Epoch 280/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1306 - accuracy: 0.9487 - val_loss: 0.8439 - val_accuracy: 0.749
9
Epoch 281/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1262 - accuracy: 0.9507 - val_loss: 1.0490 - val_accuracy: 0.717
7
Epoch 282/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1238 - accuracy: 0.9518 - val_loss: 0.9083 - val_accuracy: 0.734
5
Epoch 283/300
137/137 [==============================] - 15s 111ms/step - loss:
0.1212 - accuracy: 0.9527 - val_loss: 0.9010 - val_accuracy: 0.724
4
Epoch 284/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1168 - accuracy: 0.9548 - val_loss: 1.1194 - val_accuracy: 0.700
9
Epoch 285/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1158 - accuracy: 0.9549 - val_loss: 0.8436 - val_accuracy: 0.759
7
Epoch 286/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1206 - accuracy: 0.9528 - val_loss: 0.8466 - val_accuracy: 0.752
5
Epoch 287/300
137/137 [==============================] - 16s 113ms/step - loss:
0.1192 - accuracy: 0.9535 - val_loss: 0.9038 - val_accuracy: 0.737
6
Epoch 288/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1169 - accuracy: 0.9542 - val_loss: 0.9550 - val_accuracy: 0.738
4
Epoch 289/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1153 - accuracy: 0.9548 - val_loss: 1.1860 - val_accuracy: 0.696
9
```

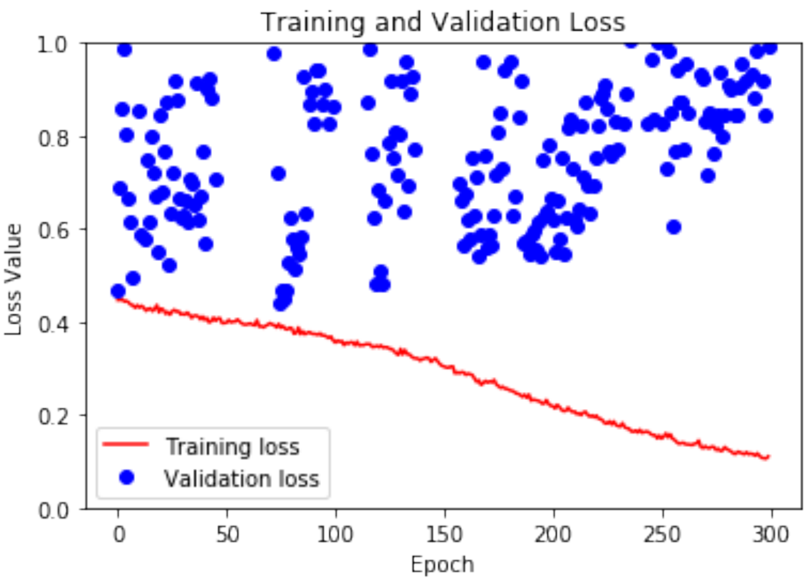
```
Epoch 290/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1196 - accuracy: 0.9530 - val_loss: 0.9185 - val_accuracy: 0.744
2
Epoch 291/300
137/137 [==============================] - 15s 113ms/step - loss:
0.1151 - accuracy: 0.9549 - val_loss: 1.1149 - val_accuracy: 0.702
4
Epoch 292/300
137/137 [==============================] - 16s 117ms/step - loss:
0.1167 - accuracy: 0.9549 - val_loss: 0.9325 - val_accuracy: 0.749
4
Epoch 293/300
137/137 [==============================] - 16s 114ms/step - loss:
0.1154 - accuracy: 0.9547 - val_loss: 0.8832 - val_accuracy: 0.753
0
Epoch 294/300
137/137 [==============================] - 15s 113ms/step - loss:
0.1121 - accuracy: 0.9563 - val_loss: 0.9845 - val_accuracy: 0.734
9
Epoch 295/300
137/137 [==============================] - 15s 112ms/step - loss:
0.1172 - accuracy: 0.9539 - val_loss: 1.3958 - val_accuracy: 0.672
0
Epoch 296/300
137/137 [==============================] - 16s 113ms/step - loss:
0.1115 - accuracy: 0.9560 - val_loss: 1.1089 - val_accuracy: 0.720
8
Epoch 297/300
137/137 [==============================] - 15s 113ms/step - loss:
0.1076 - accuracy: 0.9580 - val_loss: 0.9205 - val_accuracy: 0.752
0
Epoch 298/300
137/137 [==============================] - 15s 113ms/step - loss:
0.1067 - accuracy: 0.9582 - val_loss: 0.8460 - val_accuracy: 0.765
2
Epoch 299/300
137/137 [==============================] - 15s 113ms/step - loss:
0.1057 - accuracy: 0.9587 - val_loss: 1.0604 - val_accuracy: 0.717
4
Epoch 300/300
137/137 [==============================] - 15s 113ms/step - loss:
0.1104 - accuracy: 0.9571 - val_loss: 0.9930 - val_accuracy: 0.732
7
```

In [78]:
```
len(test_length)//BATCH_SIZE
```

Out[78]:
```
21
```

validation_steps: Only relevant if validation_data is provided and is a tf.data dataset. Total number of steps (batches of samples) to draw before stopping when performing validation at the end of every epoch. If 'validation_steps' is None, validation will run until the validation_data dataset is exhausted. In the case of a infinite dataset, it will run into a infinite loop. If 'validation_steps' is specified and only part of the dataset will be consumed, the evaluation will start from the beginning of the dataset at each epoch. This ensures that the same validation samples are used every time.

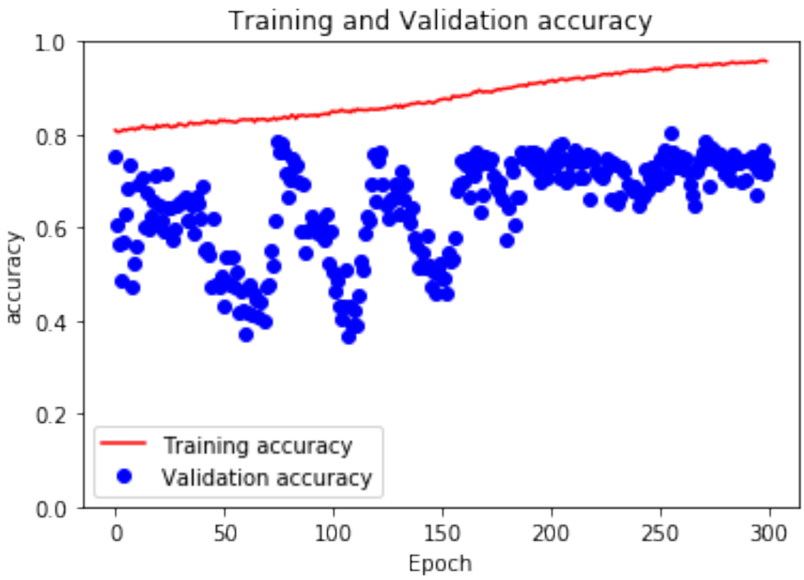
In [79]:
```
loss = model_history.history['loss']
val_loss = model_history.history['val_loss']

epochs = range(EPOCHS)

plt.figure()
plt.plot(epochs, loss, 'r', label='Training loss')
plt.plot(epochs, val_loss, 'bo', label='Validation loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss Value')
plt.ylim([0, 1])
plt.legend()
plt.show()
```



In [279]:
```
dict.keys(model_history.history)
```

Out[279]:
```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

In [280]:
```
accuracy = model_history.history['accuracy']
val_accuracy = model_history.history['val_accuracy']

epochs = range(EPOCHS)

plt.figure()
plt.plot(epochs, accuracy, 'r', label='Training accuracy')
plt.plot(epochs, val_accuracy, 'bo', label='Validation accuracy')
plt.title('Training and Validation accuracy')
plt.xlabel('Epoch')
plt.ylabel('accuracy')
plt.ylim([0, 1])
plt.legend()
plt.show()
```



Tomorrow:

- Check out Adam optimizer

Saving the model

including architecture and weights so training can continue from here and you can load it in

In [319]:
```
model.save('u_net_v1')
```

```
INFO:tensorflow:Assets written to: u_net_v1/assets

INFO:tensorflow:Assets written to: u_net_v1/assets
```

In [320]:
```
model.save('u_net_v1.h5')
```

In [322]:
```
model2 = keras.models.load_model('u_net_v1.h5')
```

In [323]:
```
model2
```

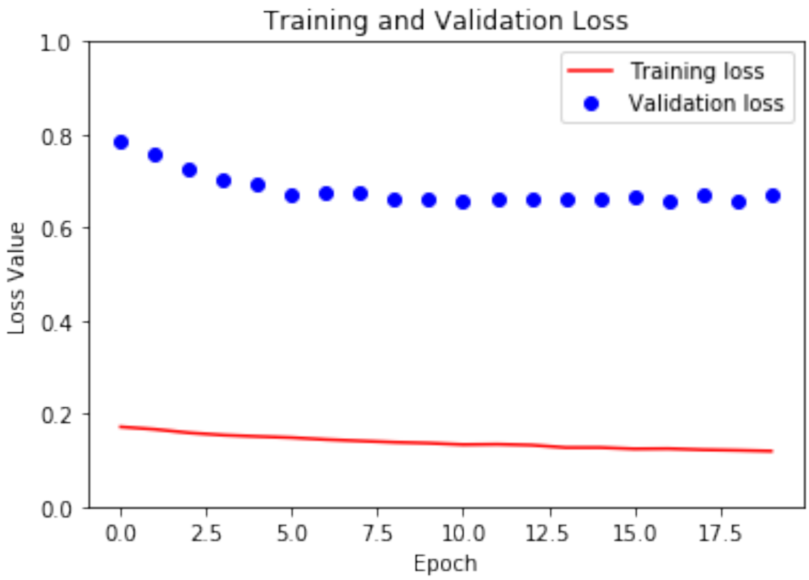
Out[323]: <tensorflow.python.keras.engine.training.Model at 0x155969f10>

In [330]:
```
EPOCHS2 = 20
VAL_SUBSPLITS = 5
VALIDATION_STEPS = len(test_length)//BATCH_SIZE//VAL_SUBSPLITS

model2.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-07,
amsgrad=False,
    name='Adam'
),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(fr
om_logits=True),
              metrics=['accuracy'])

model2_history = model2.fit(train_dataset, epochs=EPOCHS2,
                          validation_steps=VALIDATION_STEPS,
                          validation_data=test_dataset)
```

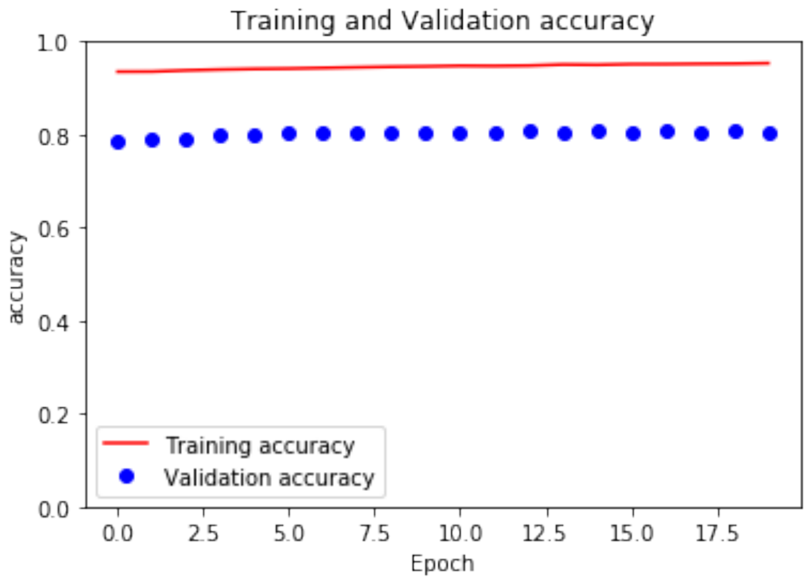
```
Train for 137 steps, validate for 4 steps
Epoch 1/20
137/137 [==============================] - 17s 123ms/step - loss:
0.1716 - accuracy: 0.9349 - val_loss: 0.7868 - val_accuracy: 0.784
1
Epoch 2/20
137/137 [==============================] - 16s 118ms/step - loss:
0.1663 - accuracy: 0.9351 - val_loss: 0.7558 - val_accuracy: 0.788
5
Epoch 3/20
137/137 [==============================] - 16s 120ms/step - loss:
0.1588 - accuracy: 0.9377 - val_loss: 0.7263 - val_accuracy: 0.789
4
Epoch 4/20
137/137 [==============================] - 16s 120ms/step - loss:
0.1539 - accuracy: 0.9395 - val_loss: 0.7033 - val_accuracy: 0.799
0
Epoch 5/20
137/137 [==============================] - 18s 129ms/step - loss:
0.1510 - accuracy: 0.9407 - val_loss: 0.6938 - val_accuracy: 0.798
9
Epoch 6/20
137/137 [==============================] - 18s 134ms/step - loss:
0.1485 - accuracy: 0.9415 - val_loss: 0.6726 - val_accuracy: 0.805
5
Epoch 7/20
137/137 [==============================] - 17s 123ms/step - loss:
0.1443 - accuracy: 0.9428 - val_loss: 0.6744 - val_accuracy: 0.803
3
```

```
Epoch 8/20
137/137 [==============================] - 17s 127ms/step - loss:
0.1414 - accuracy: 0.9442 - val_loss: 0.6738 - val_accuracy: 0.802
7
Epoch 9/20
137/137 [==============================] - 17s 120ms/step - loss:
0.1385 - accuracy: 0.9454 - val_loss: 0.6628 - val_accuracy: 0.803
2
Epoch 10/20
137/137 [==============================] - 16s 120ms/step - loss:
0.1366 - accuracy: 0.9462 - val_loss: 0.6602 - val_accuracy: 0.802
7
Epoch 11/20
137/137 [==============================] - 16s 119ms/step - loss:
0.1335 - accuracy: 0.9474 - val_loss: 0.6589 - val_accuracy: 0.805
8
Epoch 12/20
137/137 [==============================] - 16s 118ms/step - loss:
0.1342 - accuracy: 0.9471 - val_loss: 0.6618 - val_accuracy: 0.804
5
Epoch 13/20
137/137 [==============================] - 16s 115ms/step - loss:
0.1324 - accuracy: 0.9479 - val_loss: 0.6597 - val_accuracy: 0.807
2
Epoch 14/20
137/137 [==============================] - 16s 115ms/step - loss:
0.1271 - accuracy: 0.9502 - val_loss: 0.6613 - val_accuracy: 0.805
4
Epoch 15/20
137/137 [==============================] - 16s 115ms/step - loss:
0.1272 - accuracy: 0.9496 - val_loss: 0.6634 - val_accuracy: 0.806
3
Epoch 16/20
137/137 [==============================] - 15s 112ms/step - loss:
0.1243 - accuracy: 0.9510 - val_loss: 0.6642 - val_accuracy: 0.804
8
Epoch 17/20
137/137 [==============================] - 15s 112ms/step - loss:
0.1248 - accuracy: 0.9511 - val_loss: 0.6582 - val_accuracy: 0.806
8
Epoch 18/20
137/137 [==============================] - 15s 113ms/step - loss:
0.1224 - accuracy: 0.9517 - val_loss: 0.6691 - val_accuracy: 0.803
5
Epoch 19/20
137/137 [==============================] - 16s 115ms/step - loss:
0.1213 - accuracy: 0.9520 - val_loss: 0.6554 - val_accuracy: 0.810
3
Epoch 20/20
137/137 [==============================] - 15s 113ms/step - loss:
0.1195 - accuracy: 0.9531 - val_loss: 0.6704 - val_accuracy: 0.802
9
```

In [331]:
```
loss2 = model2_history.history['loss']
val_loss2 = model2_history.history['val_loss']

epochs = range(EPOCHS2)

plt.figure()
plt.plot(epochs, loss2, 'r', label='Training loss')
plt.plot(epochs, val_loss2, 'bo', label='Validation loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss Value')
plt.ylim([0, 1])
plt.legend()
plt.show()
```



In [332]:
```
accuracy2 = model2_history.history['accuracy']
val_accuracy2 = model2_history.history['val_accuracy']

epochs = range(EPOCHS2)

plt.figure()
plt.plot(epochs, accuracy2, 'r', label='Training accuracy')
plt.plot(epochs, val_accuracy2, 'bo', label='Validation accuracy')
plt.title('Training and Validation accuracy')
plt.xlabel('Epoch')
plt.ylabel('accuracy')
plt.ylim([0, 1])
plt.legend()
plt.show()
```



In [ ]:

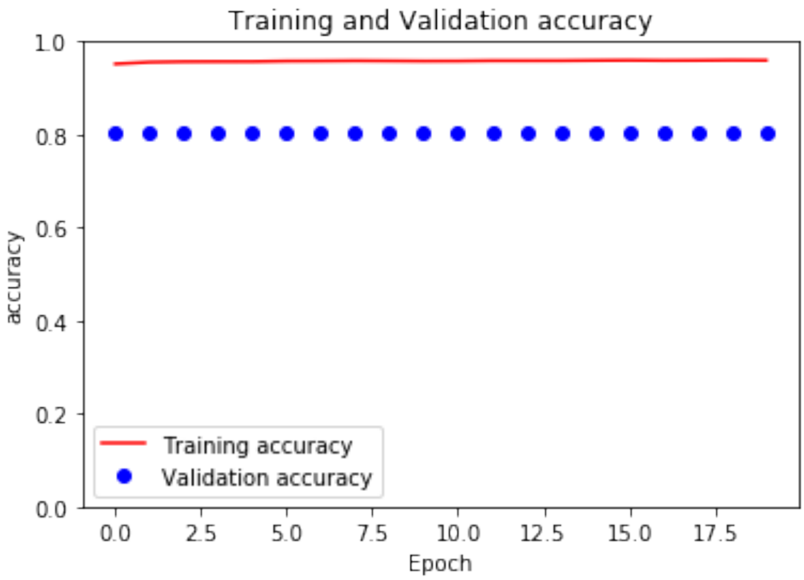
In [335]:
```
EPOCHS2 = 20
VAL_SUBSPLITS = 1
VALIDATION_STEPS = len(test_length)//BATCH_SIZE//VAL_SUBSPLITS

model2.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=0.00001, beta_1=0.9, beta_2=0.999, epsilon=1e-07,
amsgrad=False,
    name='Adam'
),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(fr
om_logits=True),
              metrics=['accuracy'])

model2_history = model2.fit(train_dataset, epochs=EPOCHS2,
                          validation_steps=VALIDATION_STEPS,
                          validation_data=test_dataset)
```

```
Train for 137 steps, validate for 21 steps
```

```
Epoch 1/20
137/137 [==============================] - 15s 109ms/step - loss:
0.1226 - accuracy: 0.9516 - val_loss: 0.7634 - val_accuracy: 0.802
0
Epoch 2/20
137/137 [==============================] - 17s 128ms/step - loss:
0.1124 - accuracy: 0.9553 - val_loss: 0.7569 - val_accuracy: 0.804
9
Epoch 3/20
137/137 [==============================] - 17s 127ms/step - loss:
0.1106 - accuracy: 0.9563 - val_loss: 0.7542 - val_accuracy: 0.804
8
Epoch 4/20
137/137 [==============================] - 18s 129ms/step - loss:
0.1095 - accuracy: 0.9565 - val_loss: 0.7548 - val_accuracy: 0.804
9
Epoch 5/20
137/137 [==============================] - 17s 122ms/step - loss:
0.1096 - accuracy: 0.9566 - val_loss: 0.7531 - val_accuracy: 0.804
6
Epoch 6/20
137/137 [==============================] - 17s 121ms/step - loss:
0.1073 - accuracy: 0.9576 - val_loss: 0.7541 - val_accuracy: 0.804
5
Epoch 7/20
137/137 [==============================] - 17s 125ms/step - loss:
0.1072 - accuracy: 0.9578 - val_loss: 0.7559 - val_accuracy: 0.804
5
Epoch 8/20
137/137 [==============================] - 17s 122ms/step - loss:
0.1064 - accuracy: 0.9583 - val_loss: 0.7559 - val_accuracy: 0.804
6
Epoch 9/20
137/137 [==============================] - 16s 118ms/step - loss:
0.1061 - accuracy: 0.9580 - val_loss: 0.7554 - val_accuracy: 0.804
6
Epoch 10/20
137/137 [==============================] - 16s 117ms/step - loss:
0.1072 - accuracy: 0.9576 - val_loss: 0.7559 - val_accuracy: 0.804
2
Epoch 11/20
137/137 [==============================] - 16s 117ms/step - loss:
0.1069 - accuracy: 0.9577 - val_loss: 0.7560 - val_accuracy: 0.804
0
Epoch 12/20
137/137 [==============================] - 16s 116ms/step - loss:
0.1062 - accuracy: 0.9584 - val_loss: 0.7568 - val_accuracy: 0.804
8
Epoch 13/20
137/137 [==============================] - 16s 117ms/step - loss:
0.1057 - accuracy: 0.9584 - val_loss: 0.7570 - val_accuracy: 0.804
0
Epoch 14/20
```

```
137/137 [==============================] - 16s 118ms/step - loss:
0.1051 - accuracy: 0.9585 - val_loss: 0.7551 - val_accuracy: 0.804
1
Epoch 15/20
137/137 [==============================] - 16s 117ms/step - loss:
0.1041 - accuracy: 0.9590 - val_loss: 0.7561 - val_accuracy: 0.804
0
Epoch 16/20
137/137 [==============================] - 16s 118ms/step - loss:
0.1033 - accuracy: 0.9593 - val_loss: 0.7561 - val_accuracy: 0.804
4
Epoch 17/20
137/137 [==============================] - 16s 118ms/step - loss:
0.1043 - accuracy: 0.9589 - val_loss: 0.7568 - val_accuracy: 0.804
1
Epoch 18/20
137/137 [==============================] - 16s 119ms/step - loss:
0.1041 - accuracy: 0.9590 - val_loss: 0.7568 - val_accuracy: 0.804
0
Epoch 19/20
137/137 [==============================] - 16s 119ms/step - loss:
0.1031 - accuracy: 0.9595 - val_loss: 0.7577 - val_accuracy: 0.804
3
Epoch 20/20
137/137 [==============================] - 16s 118ms/step - loss:
0.1040 - accuracy: 0.9592 - val_loss: 0.7565 - val_accuracy: 0.804
0
```

In [336]:
```
accuracy2 = model2_history.history['accuracy']
val_accuracy2 = model2_history.history['val_accuracy']

epochs = range(EPOCHS2)

plt.figure()
plt.plot(epochs, accuracy2, 'r', label='Training accuracy')
plt.plot(epochs, val_accuracy2, 'bo', label='Validation accuracy')
plt.title('Training and Validation accuracy')
plt.xlabel('Epoch')
plt.ylabel('accuracy')
plt.ylim([0, 1])
plt.legend()
plt.show()
```



In [340]:
```
model2.save('u_net_v2')
model2.save('u_net_v2.h5')
```

```
INFO:tensorflow:Assets written to: u_net_v2/assets

INFO:tensorflow:Assets written to: u_net_v2/assets
```

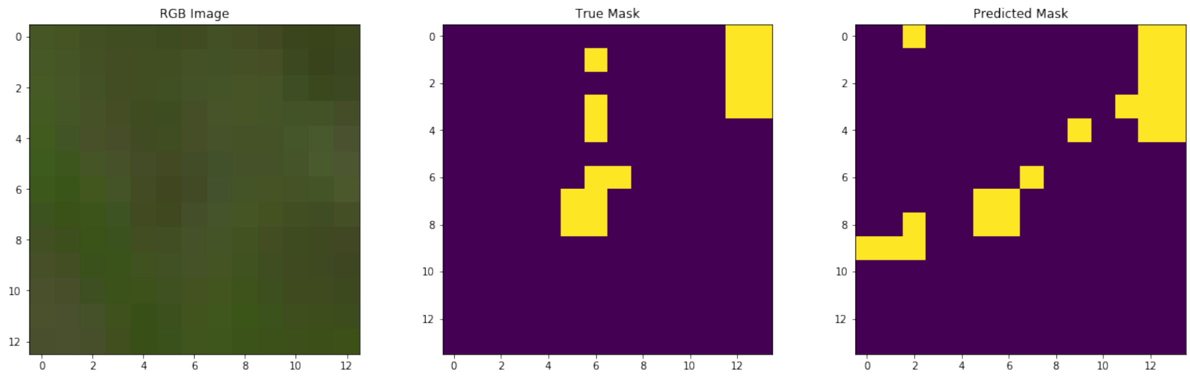
Plotting some examples

Train data

In [237]:
```
import matplotlib.gridspec as gridspec
```

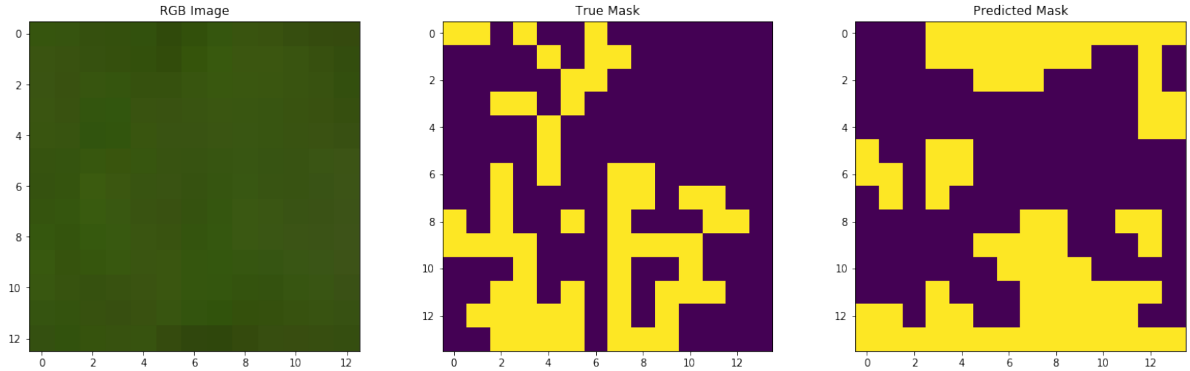
In [306]:
```
def create_mask(pred_mask, n):
  pred_mask = tf.argmax(pred_mask, axis=-1)
  return pred_mask[n]
```

In [308]:
```
n = 4

plt.figure(figsize=(20,20))
gs1 = gridspec.GridSpec(1, 3)
gs1.update(wspace=0.25, hspace=0.05) # set the spacing between axes
.

for image, mask in train_dataset.take(1):
    plt.subplot(gs1[0])
    plt.imshow(np.asarray(image[n,2:15,2:15,2:5])[:,:,::-1])
    plt.title('RGB Image')
    plt.subplot(gs1[1])
    plt.imshow(mask[n,:,:])
    plt.title('True Mask')
    plt.subplot(gs1[2])
    plt.imshow(create_mask(model.predict(image[:,:,:,:]), n))
    plt.title('Predicted Mask')
```



Test data

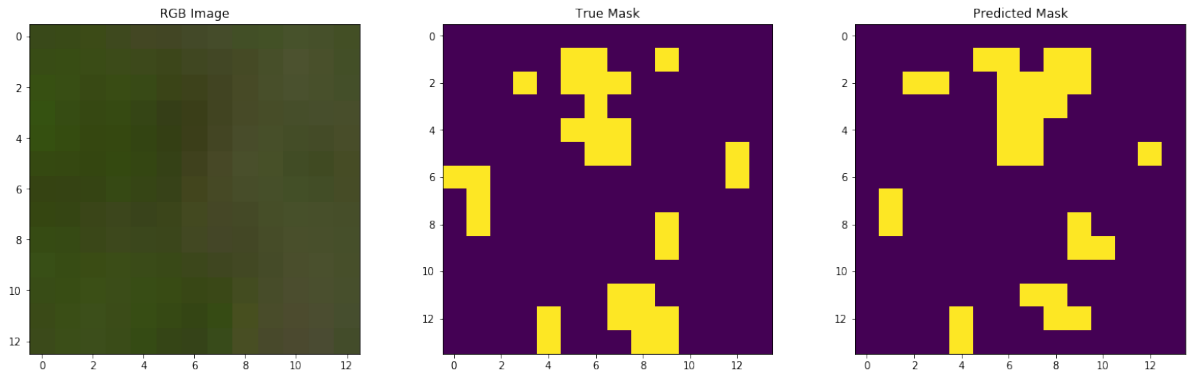
In [305]:
```
n = 10

plt.figure(figsize=(20,20))
gs1 = gridspec.GridSpec(1, 3)
gs1.update(wspace=0.25, hspace=0.05) # set the spacing between axes
.

for image, mask in test_dataset.take(1):
    plt.subplot(gs1[0])
    plt.imshow(np.asarray(image[n,2:15,2:15,2:5])[:,:,::-1])
    plt.title('RGB Image')
    plt.subplot(gs1[1])
    plt.imshow(mask[n,:,:])
    plt.title('True Mask')
    plt.subplot(gs1[2])
    plt.imshow(create_mask(model.predict(image[:,:,:,:]), n))
    plt.title('Predicted Mask')
```



In [163]:
```
def show_predictions(dataset=None, num=1):
  if dataset:
    for image, mask in dataset.take(num):
      pred_mask = model.predict(image)
      plt.imshow([image[0], create_mask(pred_mask)])
  else:
    display([sample_image, sample_mask,
             create_mask(model.predict(sample_image[tf.newaxis, ...
]))])
```

Model 2

In [339]:
```
n = 3

plt.figure(figsize=(20,20))
gs1 = gridspec.GridSpec(1, 3)
gs1.update(wspace=0.25, hspace=0.05) # set the spacing between axes
.

for image, mask in train_dataset.take(1):
    plt.subplot(gs1[0])
    plt.imshow(np.asarray(image[n,2:15,2:15,2:5])[:,:,::-1])
    plt.title('RGB Image')
    plt.subplot(gs1[1])
    plt.imshow(mask[n,:,:])
    plt.title('True Mask')
    plt.subplot(gs1[2])
    plt.imshow(create_mask(model2.predict(image[:,:,:,:]), n))
    plt.title('Predicted Mask')
```



In [ ]:

In [ ]:

Next I could try something like:

In [ ]:

In [ ]:

In [2]:
```
import tensorflow as tf
```

In [3]:
```
from tensorflow_examples.models.pix2pix import pix2pix

import tensorflow_datasets as tfds
tfds.disable_progress_bar()

from IPython.display import clear_output
import matplotlib.pyplot as plt
```

In [4]:
```
dataset, info = tfds.load('oxford_iiit_pet:3.*.*', with_info=True)
```

In [5]:
```
dataset
```

Out[5]:
```
{'test': <DatasetV1Adapter shapes: {file_name: (), image: (None, N
one, 3), label: (), segmentation_mask: (None, None, 1), species: (
)}, types: {file_name: tf.string, image: tf.uint8, label: tf.int64
, segmentation_mask: tf.uint8, species: tf.int64}>,
 'train': <DatasetV1Adapter shapes: {file_name: (), image: (None,
None, 3), label: (), segmentation_mask: (None, None, 1), species:
()}, types: {file_name: tf.string, image: tf.uint8, label: tf.int6
4, segmentation_mask: tf.uint8, species: tf.int64}>}
```

In [6]:
```
def normalize(input_image, input_mask):
  input_image = tf.cast(input_image, tf.float32) / 255.0
  input_mask -= 1
  return input_image, input_mask
```

In [7]:
```
@tf.function
def load_image_train(datapoint):
  input_image = tf.image.resize(datapoint['image'], (128, 128))
  input_mask = tf.image.resize(datapoint['segmentation_mask'], (128
, 128))

  if tf.random.uniform(()) > 0.5:
    input_image = tf.image.flip_left_right(input_image)
    input_mask = tf.image.flip_left_right(input_mask)

  input_image, input_mask = normalize(input_image, input_mask)

  return input_image, input_mask
```

In [8]:
```
def load_image_test(datapoint):
  input_image = tf.image.resize(datapoint['image'], (128, 128))
  input_mask = tf.image.resize(datapoint['segmentation_mask'], (128
, 128))

  input_image, input_mask = normalize(input_image, input_mask)

  return input_image, input_mask
```

In [9]:
```
TRAIN_LENGTH = info.splits['train'].num_examples
BATCH_SIZE = 64
BUFFER_SIZE = 1000
STEPS_PER_EPOCH = TRAIN_LENGTH // BATCH_SIZE
```

In [10]:
```
train = dataset['train'].map(load_image_train, num_parallel_calls=t
f.data.experimental.AUTOTUNE)
test = dataset['test'].map(load_image_test)
```

In [11]:
```
train_dataset = train.cache().shuffle(BUFFER_SIZE).batch(BATCH_SIZE
).repeat()
train_dataset = train_dataset.prefetch(buffer_size=tf.data.experime
ntal.AUTOTUNE)
test_dataset = test.batch(BATCH_SIZE)
```

In [12]:
```
def display(display_list):
  plt.figure(figsize=(15, 15))

  title = ['Input Image', 'True Mask', 'Predicted Mask']

  for i in range(len(display_list)):
    plt.subplot(1, len(display_list), i+1)
    plt.title(title[i])
    plt.imshow(tf.keras.preprocessing.image.array_to_img(display_li
st[i]))
    plt.axis('off')
  plt.show()
```

In [13]:
```
OUTPUT_CHANNELS = 3
```

In [14]:
```
base_model = tf.keras.applications.MobileNetV2(input_shape=[128, 12
8, 3], include_top=False)

# Use the activations of these layers
layer_names = [
    'block_1_expand_relu',   # 64x64
    'block_3_expand_relu',   # 32x32
    'block_6_expand_relu',   # 16x16
    'block_13_expand_relu',  # 8x8
    'block_16_project',      # 4x4
]
layers = [base_model.get_layer(name).output for name in layer_names
]

# Create the feature extraction model
down_stack = tf.keras.Model(inputs=base_model.input, outputs=layers
)

down_stack.trainable = False
```

In [15]:
```
base_model.input
```

Out[15]:
```
<tf.Tensor 'input_1:0' shape=(None, 128, 128, 3) dtype=float32>
```

In [16]:
```
down_stack
```

Out[16]:
```
<tensorflow.python.keras.engine.training.Model at 0x14bc9afd0>
```

In [17]:
```
up_stack = [
    pix2pix.upsample(512, 3),  # 4x4 -> 8x8
    pix2pix.upsample(256, 3),  # 8x8 -> 16x16
    pix2pix.upsample(128, 3),  # 16x16 -> 32x32
    pix2pix.upsample(64, 3),   # 32x32 -> 64x64
]
```

In [ ]:

In [18]:
```
def unet_model(output_channels):
  inputs = tf.keras.layers.Input(shape=[128, 128, 3])
  x = inputs

  # Downsampling through the model
  skips = down_stack(x)
  x = skips[-1]
  skips = reversed(skips[:-1])

  # Upsampling and establishing the skip connections
  for up, skip in zip(up_stack, skips):
    x = up(x)
    concat = tf.keras.layers.Concatenate()
    x = concat([x, skip])

  # This is the last layer of the model
  last = tf.keras.layers.Conv2DTranspose(
      output_channels, 3, strides=2,
      padding='same')  #64x64 -> 128x128

  x = last(x)

  return tf.keras.Model(inputs=inputs, outputs=x)
```

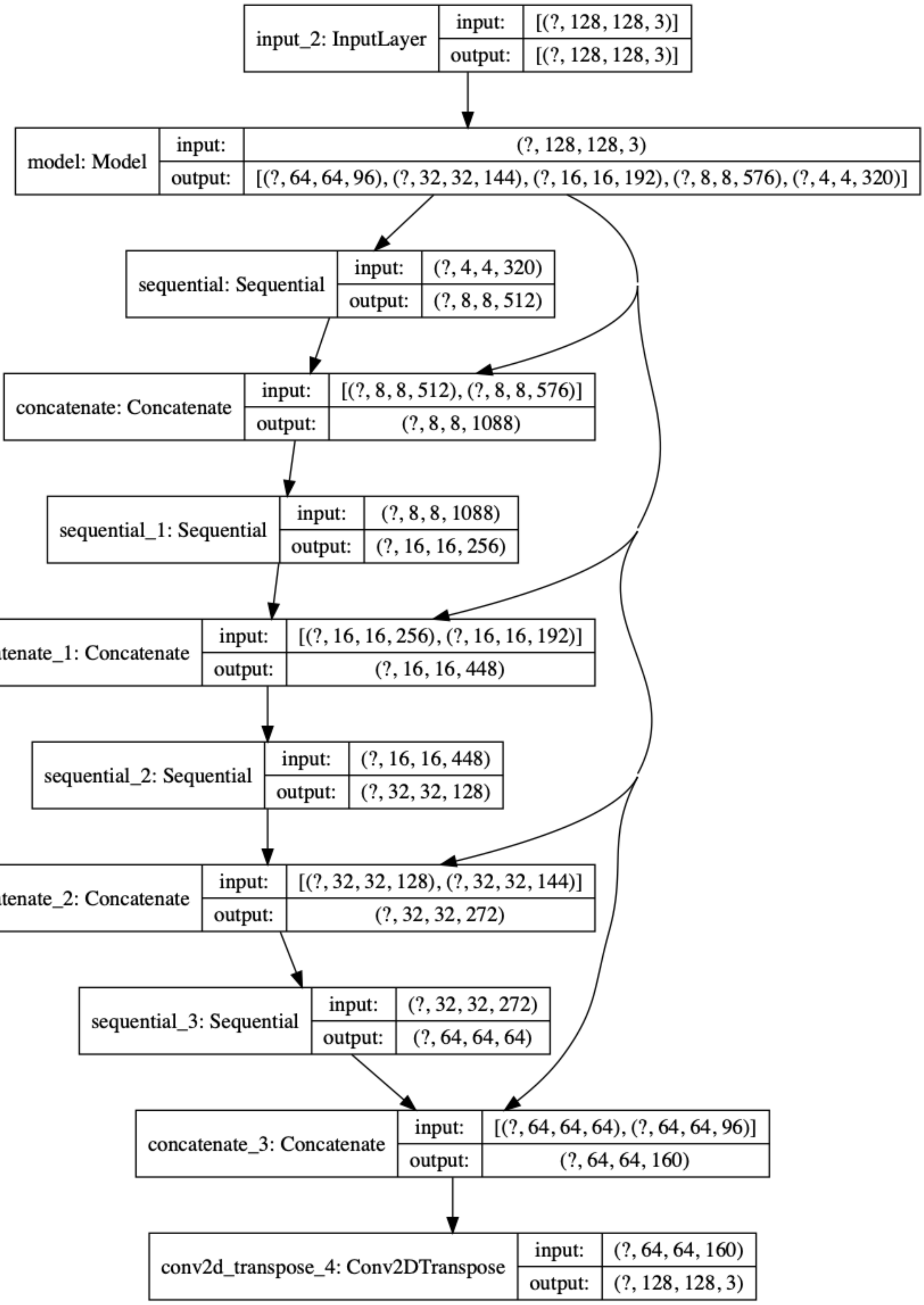
?

- why does the last layer have to be separate?
- how does Conv2DTranspose work?
  - what is the 3 in its arguments? why is stride =2

In [19]:
```
model = unet_model(OUTPUT_CHANNELS)
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(fr
om_logits=True),
              metrics=['accuracy'])
```

In [20]:
```
tf.keras.utils.plot_model(model, show_shapes=True)
```

Out[20]:



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: