# Final Project

Statistical Learning with Deep Artificial Neural Networks

Alexander J Ohrt

24. mai. 2022

## Contents

# Introduction

In this final project we will build an image recognition application using R. The work is heavily influenced by the two blog posts (Blog post one and Blog post two). The user of the web-based application will be able to upload a photo and receive predictions on what is shown in the photo, based on a machine learning model. In this case, the model is trained on 50 different bird species, which means that this app can be used to predict birds among these species. If any other file is uploaded, the application will give strange results. Notice that if the model makes predictions with a certainty probability of under 45%, it outputs a warning to the user.

The machine learning model used in this case is a convolutional deep neural network. Models of this sort are leading in the areas of image recognition and image classification. We use a pre-trained model, with a demonstrated performance on ImageNet. We load the Xception network, with weights pre-trained on the ImageNet dataset. This network is used as a baseline model and we add a final hidden layer, which will be fine-tuned on our dataset. In this way we can use **transfer learning** to make a classification model - we use the pre-trained weights of the Xception network, which, even though they are not trained our dataset, they can probably be used to extract relevant features from our data. Then, the model is fine-tuned to our data, in order to increase the performance of the model.

# Part 1 - Implementing a CNN

## Setting Parameters and Importing Images

```
#> [1] 50
```

```
#> [1] TRUE
```

```
#> Loaded Tensorflow version 2.7.2
```

```
#>
#>   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
#> 133 144 144 110 117 110 104 107 129 132 133 100 144 136 127 107 104 144 112 105
#>  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39
#> 108 116 118 118 120 112 120 112 124 123 124 120 126 132 109 110 100 120 128 118
#>  40  41  42  43  44  45  46  47  48  49
#> 106 110 133  99 156 133 104  96 106 109
```

```
#>
#>  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
#> 33 36 35 27 29 27 26 26 32 33 33 25 35 34 31 26 26 35 27 26 26 28 29 29 30 27
#> 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
#> 30 27 30 30 31 29 31 32 27 27 24 30 32 29 26 27 33 24 38 33 26 24 26 27
```

```
#>
#>  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
#>  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5
#> 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
#>  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5
```

I downloaded the Kaggle dataset. Via the zip file I manually extracted the first 50 folders of train and test images (different types of birds), and saved the folders in a directory called "train" and "test" respectively. I checked that all the bird types were equal in both the directories, which seemed correct. Thus, we have our two datasets necessary to continue with the project.

After having the data ready, we read all bird names into R by listing the subfolder names of the train directory. Moreover, we check that the subfolder names are the same when reading from train and test directories, just to double check that we have selected the same folders from the Kaggle download. This is true. We save the label names to disk for convenience.

Next we define the image width and height, defining the target size of the images. We also define the number of color channels, which is 3 in this case, since the images are RGB. Moreover, we define the batch size which will be used during the fitting and the epochs.

In order to work efficiently with the images, we use image data generators from Keras. In these we also define the preprocessing of the images and make a validation split from the test data. We simply rescale the images to have pixel values within [0,1] and set the validation split to 0.2. We do not apply any further image augmentation techniques. Then we use the flow_images_from_directory() function to automatically import batches of images. We do this for all three data sets - training, validation and testing.

## Define and Train First Model

```
#> Model: "xception"
#> __________________________________________________________________________________
#>  Layer (type)              Output Shape      Param #  Connected to
#> ==================================================================================
#>  input_1 (InputLayer)      [(None, 224, 224  0        []
#>                            , 3)]
#>
#>  block1_conv1 (Conv2D)     (None, 111, 111,  864      ['input_1[0][0]']
#>                             32)
#>
#>  block1_conv1_bn (BatchNo  (None, 111, 111,  128     ['block1_conv1[0][0]']
#>  rmalization)               32)
#>
#>  block1_conv1_act (Activa  (None, 111, 111,  0       ['block1_conv1_bn[0][0]']
#>  tion)                      32)
#>
#>  block1_conv2 (Conv2D)     (None, 109, 109,  18432    ['block1_conv1_act[0][0]']
#>                             64)
#>
```

```
#>  block1_conv2_bn (BatchNo  (None, 109, 109,  256     ['block1_conv2[0][0]']
#>  rmalization)              64)
#>
#>  block1_conv2_act (Activa  (None, 109, 109,  0       ['block1_conv2_bn[0][0]']
#>  tion)                     64)
#>
#>  block2_sepconv1 (Separab  (None, 109, 109,  8768    ['block1_conv2_act[0][0]']
#>  leConv2D)                 128)
#>
#>  block2_sepconv1_bn (Batc  (None, 109, 109,  512     ['block2_sepconv1[0][0]']
#>  hNormalization)           128)
#>
#>  block2_sepconv2_act (Act  (None, 109, 109,  0       ['block2_sepconv1_bn[0][0]'
#>  ivation)                  128)                      ]
#>
#>  block2_sepconv2 (Separab  (None, 109, 109,  17536   ['block2_sepconv2_act[0][0]
#>  leConv2D)                 128)                      ']
#>
#>  block2_sepconv2_bn (Batc  (None, 109, 109,  512     ['block2_sepconv2[0][0]']
#>  hNormalization)           128)
#>
#>  conv2d (Conv2D)          (None, 55, 55, 1  8192     ['block1_conv2_act[0][0]']
#>                           28)
#>
#>  block2_pool (MaxPooling2  (None, 55, 55, 1  0       ['block2_sepconv2_bn[0][0]'
#>  D)                       28)                       ]
#>
#>  batch_normalization (Bat  (None, 55, 55, 1  512     ['conv2d[0][0]']
#>  chNormalization)         28)
#>
#>  add (Add)                (None, 55, 55, 1  0        ['block2_pool[0][0]',
#>                           28)                        'batch_normalization[0][0]
#>                                                     ']
#>
#>  block3_sepconv1_act (Act  (None, 55, 55, 1  0       ['add[0][0]']
#>  ivation)                  28)
#>
#>  block3_sepconv1 (Separab  (None, 55, 55, 2  33920   ['block3_sepconv1_act[0][0]
#>  leConv2D)                 56)                       ']
#>
#>  block3_sepconv1_bn (Batc  (None, 55, 55, 2  1024    ['block3_sepconv1[0][0]']
#>  hNormalization)           56)
#>
#>  block3_sepconv2_act (Act  (None, 55, 55, 2  0       ['block3_sepconv1_bn[0][0]'
#>  ivation)                  56)                       ]
#>
#>  block3_sepconv2 (Separab  (None, 55, 55, 2  67840   ['block3_sepconv2_act[0][0]
#>  leConv2D)                 56)                       ']
#>
#>  block3_sepconv2_bn (Batc  (None, 55, 55, 2  1024    ['block3_sepconv2[0][0]']
#>  hNormalization)           56)
#>
#>  conv2d_1 (Conv2D)        (None, 28, 28, 2  32768    ['add[0][0]']
#>                           56)
#>
#>  block3_pool (MaxPooling2  (None, 28, 28, 2  0       ['block3_sepconv2_bn[0][0]'
```

```
#>  D)                       56)                         ]
#>
#>  batch_normalization_1 (B  (None, 28, 28, 2  1024    ['conv2d_1[0][0]']
#>  atchNormalization)       56)
#>
#>  add_1 (Add)              (None, 28, 28, 2  0         ['block3_pool[0][0]',
#>                           56)                          'batch_normalization_1[0][[
#>                                                       0]']
#>
#>  block4_sepconv1_act (Act  (None, 28, 28, 2  0       ['add_1[0][0]']
#>  ivation)                 56)
#>
#>  block4_sepconv1 (Separab  (None, 28, 28, 7  188672  ['block4_sepconv1_act[0][0]
#>  leConv2D)                28)                         ']
#>
#>  block4_sepconv1_bn (Batc  (None, 28, 28, 7  2912    ['block4_sepconv1[0][0]']
#>  hNormalization)          28)
#>
#>  block4_sepconv2_act (Act  (None, 28, 28, 7  0       ['block4_sepconv1_bn[0][0]'
#>  ivation)                 28)                         ]
#>
#>  block4_sepconv2 (Separab  (None, 28, 28, 7  536536  ['block4_sepconv2_act[0][0]
#>  leConv2D)                28)                         ']
#>
#>  block4_sepconv2_bn (Batc  (None, 28, 28, 7  2912    ['block4_sepconv2[0][0]']
#>  hNormalization)          28)
#>
#>  conv2d_2 (Conv2D)        (None, 14, 14, 7  186368   ['add_1[0][0]']
#>                           28)
#>
#>  block4_pool (MaxPooling2  (None, 14, 14, 7  0       ['block4_sepconv2_bn[0][0]'
#>  D)                       28)                         ]
#>
#>  batch_normalization_2 (B  (None, 14, 14, 7  2912    ['conv2d_2[0][0]']
#>  atchNormalization)       28)
#>
#>  add_2 (Add)              (None, 14, 14, 7  0         ['block4_pool[0][0]',
#>                           28)                          'batch_normalization_2[0][[
#>                                                       0]']
#>
#>  block5_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_2[0][0]']
#>  ivation)                 28)
#>
#>  block5_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block5_sepconv1_act[0][0]
#>  leConv2D)                28)                         ']
#>
#>  block5_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block5_sepconv1[0][0]']
#>  hNormalization)          28)
#>
#>  block5_sepconv2_act (Act  (None, 14, 14, 7  0       ['block5_sepconv1_bn[0][0]'
#>  ivation)                 28)                         ]
#>
#>  block5_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block5_sepconv2_act[0][0]
#>  leConv2D)                28)                         ']
#>
#>  block5_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block5_sepconv2[0][0]']
```

```
#>  hNormalization)          28)
#>
#>  block5_sepconv3_act (Act  (None, 14, 14, 7  0       ['block5_sepconv2_bn[0][0]'
#>  ivation)                 28)                        ]
#>
#>  block5_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block5_sepconv3_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block5_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block5_sepconv3[0][0]']
#>  hNormalization)          28)
#>
#>  add_3 (Add)              (None, 14, 14, 7  0         ['block5_sepconv3_bn[0][0]'
#>                           28)                        , 'add_2[0][0]']
#>
#>  block6_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_3[0][0]']
#>  ivation)                 28)
#>
#>  block6_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block6_sepconv1_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block6_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block6_sepconv1[0][0]']
#>  hNormalization)          28)
#>
#>  block6_sepconv2_act (Act  (None, 14, 14, 7  0       ['block6_sepconv1_bn[0][0]'
#>  ivation)                 28)                        ]
#>
#>  block6_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block6_sepconv2_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block6_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block6_sepconv2[0][0]']
#>  hNormalization)          28)
#>
#>  block6_sepconv3_act (Act  (None, 14, 14, 7  0       ['block6_sepconv2_bn[0][0]'
#>  ivation)                 28)                        ]
#>
#>  block6_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block6_sepconv3_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block6_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block6_sepconv3[0][0]']
#>  hNormalization)          28)
#>
#>  add_4 (Add)              (None, 14, 14, 7  0         ['block6_sepconv3_bn[0][0]'
#>                           28)                        , 'add_3[0][0]']
#>
#>  block7_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_4[0][0]']
#>  ivation)                 28)
#>
#>  block7_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block7_sepconv1_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block7_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block7_sepconv1[0][0]']
#>  hNormalization)          28)
#>
#>  block7_sepconv2_act (Act  (None, 14, 14, 7  0       ['block7_sepconv1_bn[0][0]'
#>  ivation)                 28)                        ]
#>
```

```
#>  block7_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block7_sepconv2_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block7_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block7_sepconv2[0][0]']
#>  hNormalization)           28)
#>
#>  block7_sepconv3_act (Act  (None, 14, 14, 7  0       ['block7_sepconv2_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block7_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block7_sepconv3_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block7_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block7_sepconv3[0][0]']
#>  hNormalization)           28)
#>
#>  add_5 (Add)               (None, 14, 14, 7  0        ['block7_sepconv3_bn[0][0]'
#>                            28)                       , 'add_4[0][0]']
#>
#>  block8_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_5[0][0]']
#>  ivation)                  28)
#>
#>  block8_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block8_sepconv1_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block8_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block8_sepconv1[0][0]']
#>  hNormalization)           28)
#>
#>  block8_sepconv2_act (Act  (None, 14, 14, 7  0       ['block8_sepconv1_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block8_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block8_sepconv2_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block8_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block8_sepconv2[0][0]']
#>  hNormalization)           28)
#>
#>  block8_sepconv3_act (Act  (None, 14, 14, 7  0       ['block8_sepconv2_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block8_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block8_sepconv3_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block8_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block8_sepconv3[0][0]']
#>  hNormalization)           28)
#>
#>  add_6 (Add)               (None, 14, 14, 7  0        ['block8_sepconv3_bn[0][0]'
#>                            28)                       , 'add_5[0][0]']
#>
#>  block9_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_6[0][0]']
#>  ivation)                  28)
#>
#>  block9_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block9_sepconv1_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block9_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block9_sepconv1[0][0]']
#>  hNormalization)           28)
```

```
#>
#>  block9_sepconv2_act (Act  (None, 14, 14, 7  0       ['block9_sepconv1_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block9_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block9_sepconv2_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block9_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block9_sepconv2[0][0]']
#>  hNormalization)           28)
#>
#>  block9_sepconv3_act (Act  (None, 14, 14, 7  0       ['block9_sepconv2_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block9_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block9_sepconv3_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block9_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block9_sepconv3[0][0]']
#>  hNormalization)           28)
#>
#>  add_7 (Add)               (None, 14, 14, 7  0        ['block9_sepconv3_bn[0][0]'
#>                            28)                       , 'add_6[0][0]']
#>
#>  block10_sepconv1_act (Ac  (None, 14, 14, 7  0       ['add_7[0][0]']
#>  tivation)                 28)
#>
#>  block10_sepconv1 (Separa  (None, 14, 14, 7  536536  ['block10_sepconv1_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block10_sepconv1_bn (Bat  (None, 14, 14, 7  2912    ['block10_sepconv1[0][0]']
#>  chNormalization)          28)
#>
#>  block10_sepconv2_act (Ac  (None, 14, 14, 7  0       ['block10_sepconv1_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block10_sepconv2 (Separa  (None, 14, 14, 7  536536  ['block10_sepconv2_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block10_sepconv2_bn (Bat  (None, 14, 14, 7  2912    ['block10_sepconv2[0][0]']
#>  chNormalization)          28)
#>
#>  block10_sepconv3_act (Ac  (None, 14, 14, 7  0       ['block10_sepconv2_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block10_sepconv3 (Separa  (None, 14, 14, 7  536536  ['block10_sepconv3_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block10_sepconv3_bn (Bat  (None, 14, 14, 7  2912    ['block10_sepconv3[0][0]']
#>  chNormalization)          28)
#>
#>  add_8 (Add)               (None, 14, 14, 7  0        ['block10_sepconv3_bn[0][0]
#>                            28)                       ',
#>                                                       'add_7[0][0]']
#>
#>  block11_sepconv1_act (Ac  (None, 14, 14, 7  0       ['add_8[0][0]']
#>  tivation)                 28)
#>
```

```
#>  block11_sepconv1 (Separa  (None, 14, 14, 7  536536  ['block11_sepconv1_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block11_sepconv1_bn (Bat  (None, 14, 14, 7  2912    ['block11_sepconv1[0][0]']
#>  chNormalization)          28)
#>
#>  block11_sepconv2_act (Ac  (None, 14, 14, 7  0       ['block11_sepconv1_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block11_sepconv2 (Separa  (None, 14, 14, 7  536536  ['block11_sepconv2_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block11_sepconv2_bn (Bat  (None, 14, 14, 7  2912    ['block11_sepconv2[0][0]']
#>  chNormalization)          28)
#>
#>  block11_sepconv3_act (Ac  (None, 14, 14, 7  0       ['block11_sepconv2_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block11_sepconv3 (Separa  (None, 14, 14, 7  536536  ['block11_sepconv3_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block11_sepconv3_bn (Bat  (None, 14, 14, 7  2912    ['block11_sepconv3[0][0]']
#>  chNormalization)          28)
#>
#>  add_9 (Add)               (None, 14, 14, 7  0        ['block11_sepconv3_bn[0][0]
#>                            28)                       ',
#>                                                       'add_8[0][0]']
#>
#>  block12_sepconv1_act (Ac  (None, 14, 14, 7  0       ['add_9[0][0]']
#>  tivation)                 28)
#>
#>  block12_sepconv1 (Separa  (None, 14, 14, 7  536536  ['block12_sepconv1_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block12_sepconv1_bn (Bat  (None, 14, 14, 7  2912    ['block12_sepconv1[0][0]']
#>  chNormalization)          28)
#>
#>  block12_sepconv2_act (Ac  (None, 14, 14, 7  0       ['block12_sepconv1_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block12_sepconv2 (Separa  (None, 14, 14, 7  536536  ['block12_sepconv2_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block12_sepconv2_bn (Bat  (None, 14, 14, 7  2912    ['block12_sepconv2[0][0]']
#>  chNormalization)          28)
#>
#>  block12_sepconv3_act (Ac  (None, 14, 14, 7  0       ['block12_sepconv2_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block12_sepconv3 (Separa  (None, 14, 14, 7  536536  ['block12_sepconv3_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block12_sepconv3_bn (Bat  (None, 14, 14, 7  2912    ['block12_sepconv3[0][0]']
#>  chNormalization)          28)
#>
#>  add_10 (Add)              (None, 14, 14, 7  0        ['block12_sepconv3_bn[0][0]
```

```
#>                           28)                          ',
#>                                                         'add_9[0][0]']
#>
#>  block13_sepconv1_act (Ac  (None, 14, 14, 7  0       ['add_10[0][0]']
#>  tivation)                 28)
#>
#>  block13_sepconv1 (Separa  (None, 14, 14, 7  536536  ['block13_sepconv1_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block13_sepconv1_bn (Bat  (None, 14, 14, 7  2912    ['block13_sepconv1[0][0]']
#>  chNormalization)          28)
#>
#>  block13_sepconv2_act (Ac  (None, 14, 14, 7  0       ['block13_sepconv1_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block13_sepconv2 (Separa  (None, 14, 14, 1  752024  ['block13_sepconv2_act[0][0
#>  bleConv2D)                024)                      ]']
#>
#>  block13_sepconv2_bn (Bat  (None, 14, 14, 1  4096    ['block13_sepconv2[0][0]']
#>  chNormalization)          024)
#>
#>  conv2d_3 (Conv2D)         (None, 7, 7, 102  745472   ['add_10[0][0]']
#>                            4)
#>
#>  block13_pool (MaxPooling  (None, 7, 7, 102  0       ['block13_sepconv2_bn[0][0]
#>  2D)                       4)                        ']
#>
#>  batch_normalization_3 (B  (None, 7, 7, 102  4096    ['conv2d_3[0][0]']
#>  atchNormalization)        4)
#>
#>  add_11 (Add)              (None, 7, 7, 102  0        ['block13_pool[0][0]',
#>                            4)                         'batch_normalization_3[0][[
#>                                                      0]']
#>
#>  block14_sepconv1 (Separa  (None, 7, 7, 153  1582080  ['add_11[0][0]']
#>  bleConv2D)                6)
#>
#>  block14_sepconv1_bn (Bat  (None, 7, 7, 153  6144    ['block14_sepconv1[0][0]']
#>  chNormalization)          6)
#>
#>  block14_sepconv1_act (Ac  (None, 7, 7, 153  0       ['block14_sepconv1_bn[0][0]
#>  tivation)                 6)                        ']
#>
#>  block14_sepconv2 (Separa  (None, 7, 7, 204  3159552  ['block14_sepconv1_act[0][0
#>  bleConv2D)                8)                        ]']
#>
#>  block14_sepconv2_bn (Bat  (None, 7, 7, 204  8192    ['block14_sepconv2[0][0]']
#>  chNormalization)          8)
#>
#>  block14_sepconv2_act (Ac  (None, 7, 7, 204  0       ['block14_sepconv2_bn[0][0]
#>  tivation)                 8)                        ']
#>
#> ==================================================================================
#> Total params: 20,861,480
#> Trainable params: 20,806,952
#> Non-trainable params: 54,528
```

```
#> ____________________________________________________________________________
#> Model: "xception"
#> ____________________________________________________________________________
#>  Layer (type)              Output Shape      Param #  Connected to
#> ============================================================================
#>  input_1 (InputLayer)      [(None, 224, 224  0        []
#>                            , 3)]
#>
#>  block1_conv1 (Conv2D)     (None, 111, 111,  864      ['input_1[0][0]']
#>                             32)
#>
#>  block1_conv1_bn (BatchNo  (None, 111, 111,  128      ['block1_conv1[0][0]']
#>  rmalization)               32)
#>
#>  block1_conv1_act (Activa  (None, 111, 111,  0        ['block1_conv1_bn[0][0]']
#>  tion)                      32)
#>
#>  block1_conv2 (Conv2D)     (None, 109, 109,  18432    ['block1_conv1_act[0][0]']
#>                             64)
#>
#>  block1_conv2_bn (BatchNo  (None, 109, 109,  256      ['block1_conv2[0][0]']
#>  rmalization)               64)
#>
#>  block1_conv2_act (Activa  (None, 109, 109,  0        ['block1_conv2_bn[0][0]']
#>  tion)                      64)
#>
#>  block2_sepconv1 (Separab  (None, 109, 109,  8768     ['block1_conv2_act[0][0]']
#>  leConv2D)                  128)
#>
#>  block2_sepconv1_bn (Batc  (None, 109, 109,  512      ['block2_sepconv1[0][0]']
#>  hNormalization)            128)
#>
#>  block2_sepconv2_act (Act  (None, 109, 109,  0        ['block2_sepconv1_bn[0][0]'
#>  ivation)                   128)                      ]
#>
#>  block2_sepconv2 (Separab  (None, 109, 109,  17536    ['block2_sepconv2_act[0][0]
#>  leConv2D)                  128)                      ']
#>
#>  block2_sepconv2_bn (Batc  (None, 109, 109,  512      ['block2_sepconv2[0][0]']
#>  hNormalization)            128)
#>
#>  conv2d (Conv2D)           (None, 55, 55, 1  8192     ['block1_conv2_act[0][0]']
#>                            28)
#>
#>  block2_pool (MaxPooling2  (None, 55, 55, 1  0        ['block2_sepconv2_bn[0][0]'
#>  D)                        28)                       ]
#>
#>  batch_normalization (Bat  (None, 55, 55, 1  512      ['conv2d[0][0]']
#>  chNormalization)          28)
#>
#>  add (Add)                 (None, 55, 55, 1  0        ['block2_pool[0][0]',
#>                            28)                        'batch_normalization[0][0]
#>                                                       ']
#>
#>  block3_sepconv1_act (Act  (None, 55, 55, 1  0        ['add[0][0]']
#>  ivation)                  28)
```

```
#>
#>  block3_sepconv1 (Separab  (None, 55, 55, 2  33920   ['block3_sepconv1_act[0][0]
#>  leConv2D)                 56)                       ']
#>
#>  block3_sepconv1_bn (Batc  (None, 55, 55, 2  1024    ['block3_sepconv1[0][0]']
#>  hNormalization)           56)
#>
#>  block3_sepconv2_act (Act  (None, 55, 55, 2  0       ['block3_sepconv1_bn[0][0]'
#>  ivation)                  56)                       ]
#>
#>  block3_sepconv2 (Separab  (None, 55, 55, 2  67840   ['block3_sepconv2_act[0][0]
#>  leConv2D)                 56)                       ']
#>
#>  block3_sepconv2_bn (Batc  (None, 55, 55, 2  1024    ['block3_sepconv2[0][0]']
#>  hNormalization)           56)
#>
#>  conv2d_1 (Conv2D)         (None, 28, 28, 2  32768   ['add[0][0]']
#>                            56)
#>
#>  block3_pool (MaxPooling2  (None, 28, 28, 2  0       ['block3_sepconv2_bn[0][0]'
#>  D)                        56)                       ]
#>
#>  batch_normalization_1 (B  (None, 28, 28, 2  1024    ['conv2d_1[0][0]']
#>  atchNormalization)        56)
#>
#>  add_1 (Add)               (None, 28, 28, 2  0        ['block3_pool[0][0]',
#>                            56)                        'batch_normalization_1[0][[
#>                                                      0]']
#>
#>  block4_sepconv1_act (Act  (None, 28, 28, 2  0       ['add_1[0][0]']
#>  ivation)                  56)
#>
#>  block4_sepconv1 (Separab  (None, 28, 28, 7  188672  ['block4_sepconv1_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block4_sepconv1_bn (Batc  (None, 28, 28, 7  2912    ['block4_sepconv1[0][0]']
#>  hNormalization)           28)
#>
#>  block4_sepconv2_act (Act  (None, 28, 28, 7  0       ['block4_sepconv1_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block4_sepconv2 (Separab  (None, 28, 28, 7  536536  ['block4_sepconv2_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block4_sepconv2_bn (Batc  (None, 28, 28, 7  2912    ['block4_sepconv2[0][0]']
#>  hNormalization)           28)
#>
#>  conv2d_2 (Conv2D)         (None, 14, 14, 7  186368  ['add_1[0][0]']
#>                            28)
#>
#>  block4_pool (MaxPooling2  (None, 14, 14, 7  0       ['block4_sepconv2_bn[0][0]'
#>  D)                        28)                       ]
#>
#>  batch_normalization_2 (B  (None, 14, 14, 7  2912    ['conv2d_2[0][0]']
#>  atchNormalization)        28)
#>
```

```
#>  add_2 (Add)              (None, 14, 14, 7  0        ['block4_pool[0][0]',
#>                           28)                         'batch_normalization_2[0][[
#>                                                      0]']
#>
#>  block5_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_2[0][0]']
#>  ivation)                 28)
#>
#>  block5_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block5_sepconv1_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block5_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block5_sepconv1[0][0]']
#>  hNormalization)          28)
#>
#>  block5_sepconv2_act (Act  (None, 14, 14, 7  0       ['block5_sepconv1_bn[0][0]'
#>  ivation)                 28)                        ]
#>
#>  block5_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block5_sepconv2_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block5_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block5_sepconv2[0][0]']
#>  hNormalization)          28)
#>
#>  block5_sepconv3_act (Act  (None, 14, 14, 7  0       ['block5_sepconv2_bn[0][0]'
#>  ivation)                 28)                        ]
#>
#>  block5_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block5_sepconv3_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block5_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block5_sepconv3[0][0]']
#>  hNormalization)          28)
#>
#>  add_3 (Add)              (None, 14, 14, 7  0        ['block5_sepconv3_bn[0][0]'
#>                           28)                        , 'add_2[0][0]']
#>
#>  block6_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_3[0][0]']
#>  ivation)                 28)
#>
#>  block6_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block6_sepconv1_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block6_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block6_sepconv1[0][0]']
#>  hNormalization)          28)
#>
#>  block6_sepconv2_act (Act  (None, 14, 14, 7  0       ['block6_sepconv1_bn[0][0]'
#>  ivation)                 28)                        ]
#>
#>  block6_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block6_sepconv2_act[0][0]
#>  leConv2D)                28)                        ']
#>
#>  block6_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block6_sepconv2[0][0]']
#>  hNormalization)          28)
#>
#>  block6_sepconv3_act (Act  (None, 14, 14, 7  0       ['block6_sepconv2_bn[0][0]'
#>  ivation)                 28)                        ]
#>
#>  block6_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block6_sepconv3_act[0][0]
```

```
#>  leConv2D)               28)                         ']
#>
#>  block6_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block6_sepconv3[0][0]']
#>  hNormalization)         28)
#>
#>  add_4 (Add)             (None, 14, 14, 7  0         ['block6_sepconv3_bn[0][0]'
#>                          28)                         , 'add_3[0][0]']
#>
#>  block7_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_4[0][0]']
#>  ivation)                28)
#>
#>  block7_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block7_sepconv1_act[0][0]
#>  leConv2D)               28)                         ']
#>
#>  block7_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block7_sepconv1[0][0]']
#>  hNormalization)         28)
#>
#>  block7_sepconv2_act (Act  (None, 14, 14, 7  0       ['block7_sepconv1_bn[0][0]'
#>  ivation)                28)                         ]
#>
#>  block7_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block7_sepconv2_act[0][0]
#>  leConv2D)               28)                         ']
#>
#>  block7_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block7_sepconv2[0][0]']
#>  hNormalization)         28)
#>
#>  block7_sepconv3_act (Act  (None, 14, 14, 7  0       ['block7_sepconv2_bn[0][0]'
#>  ivation)                28)                         ]
#>
#>  block7_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block7_sepconv3_act[0][0]
#>  leConv2D)               28)                         ']
#>
#>  block7_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block7_sepconv3[0][0]']
#>  hNormalization)         28)
#>
#>  add_5 (Add)             (None, 14, 14, 7  0         ['block7_sepconv3_bn[0][0]'
#>                          28)                         , 'add_4[0][0]']
#>
#>  block8_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_5[0][0]']
#>  ivation)                28)
#>
#>  block8_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block8_sepconv1_act[0][0]
#>  leConv2D)               28)                         ']
#>
#>  block8_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block8_sepconv1[0][0]']
#>  hNormalization)         28)
#>
#>  block8_sepconv2_act (Act  (None, 14, 14, 7  0       ['block8_sepconv1_bn[0][0]'
#>  ivation)                28)                         ]
#>
#>  block8_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block8_sepconv2_act[0][0]
#>  leConv2D)               28)                         ']
#>
#>  block8_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block8_sepconv2[0][0]']
#>  hNormalization)         28)
#>
```

```
#>  block8_sepconv3_act (Act  (None, 14, 14, 7  0       ['block8_sepconv2_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block8_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block8_sepconv3_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block8_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block8_sepconv3[0][0]']
#>  hNormalization)           28)
#>
#>  add_6 (Add)               (None, 14, 14, 7  0       ['block8_sepconv3_bn[0][0]'
#>                            28)                       , 'add_5[0][0]']
#>
#>  block9_sepconv1_act (Act  (None, 14, 14, 7  0       ['add_6[0][0]']
#>  ivation)                  28)
#>
#>  block9_sepconv1 (Separab  (None, 14, 14, 7  536536  ['block9_sepconv1_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block9_sepconv1_bn (Batc  (None, 14, 14, 7  2912    ['block9_sepconv1[0][0]']
#>  hNormalization)           28)
#>
#>  block9_sepconv2_act (Act  (None, 14, 14, 7  0       ['block9_sepconv1_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block9_sepconv2 (Separab  (None, 14, 14, 7  536536  ['block9_sepconv2_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block9_sepconv2_bn (Batc  (None, 14, 14, 7  2912    ['block9_sepconv2[0][0]']
#>  hNormalization)           28)
#>
#>  block9_sepconv3_act (Act  (None, 14, 14, 7  0       ['block9_sepconv2_bn[0][0]'
#>  ivation)                  28)                       ]
#>
#>  block9_sepconv3 (Separab  (None, 14, 14, 7  536536  ['block9_sepconv3_act[0][0]
#>  leConv2D)                 28)                       ']
#>
#>  block9_sepconv3_bn (Batc  (None, 14, 14, 7  2912    ['block9_sepconv3[0][0]']
#>  hNormalization)           28)
#>
#>  add_7 (Add)               (None, 14, 14, 7  0       ['block9_sepconv3_bn[0][0]'
#>                            28)                       , 'add_6[0][0]']
#>
#>  block10_sepconv1_act (Ac  (None, 14, 14, 7  0       ['add_7[0][0]']
#>  tivation)                 28)
#>
#>  block10_sepconv1 (Separa  (None, 14, 14, 7  536536  ['block10_sepconv1_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block10_sepconv1_bn (Bat  (None, 14, 14, 7  2912    ['block10_sepconv1[0][0]']
#>  chNormalization)          28)
#>
#>  block10_sepconv2_act (Ac  (None, 14, 14, 7  0       ['block10_sepconv1_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block10_sepconv2 (Separa  (None, 14, 14, 7  536536  ['block10_sepconv2_act[0][0
#>  bleConv2D)                28)                       ]']
```

```
#>
#>  block10_sepconv2_bn (Bat  (None, 14, 14, 7  2912    ['block10_sepconv2[0][0]']
#>  chNormalization)         28)
#>
#>  block10_sepconv3_act (Ac  (None, 14, 14, 7  0       ['block10_sepconv2_bn[0][0]
#>  tivation)                28)                        ']
#>
#>  block10_sepconv3 (Separa  (None, 14, 14, 7  536536  ['block10_sepconv3_act[0][0
#>  bleConv2D)               28)                        ]']
#>
#>  block10_sepconv3_bn (Bat  (None, 14, 14, 7  2912    ['block10_sepconv3[0][0]']
#>  chNormalization)         28)
#>
#>  add_8 (Add)              (None, 14, 14, 7  0        ['block10_sepconv3_bn[0][0]
#>                           28)                        ',
#>                                                       'add_7[0][0]']
#>
#>  block11_sepconv1_act (Ac  (None, 14, 14, 7  0       ['add_8[0][0]']
#>  tivation)                28)
#>
#>  block11_sepconv1 (Separa  (None, 14, 14, 7  536536  ['block11_sepconv1_act[0][0
#>  bleConv2D)               28)                        ]']
#>
#>  block11_sepconv1_bn (Bat  (None, 14, 14, 7  2912    ['block11_sepconv1[0][0]']
#>  chNormalization)         28)
#>
#>  block11_sepconv2_act (Ac  (None, 14, 14, 7  0       ['block11_sepconv1_bn[0][0]
#>  tivation)                28)                        ']
#>
#>  block11_sepconv2 (Separa  (None, 14, 14, 7  536536  ['block11_sepconv2_act[0][0
#>  bleConv2D)               28)                        ]']
#>
#>  block11_sepconv2_bn (Bat  (None, 14, 14, 7  2912    ['block11_sepconv2[0][0]']
#>  chNormalization)         28)
#>
#>  block11_sepconv3_act (Ac  (None, 14, 14, 7  0       ['block11_sepconv2_bn[0][0]
#>  tivation)                28)                        ']
#>
#>  block11_sepconv3 (Separa  (None, 14, 14, 7  536536  ['block11_sepconv3_act[0][0
#>  bleConv2D)               28)                        ]']
#>
#>  block11_sepconv3_bn (Bat  (None, 14, 14, 7  2912    ['block11_sepconv3[0][0]']
#>  chNormalization)         28)
#>
#>  add_9 (Add)              (None, 14, 14, 7  0        ['block11_sepconv3_bn[0][0]
#>                           28)                        ',
#>                                                       'add_8[0][0]']
#>
#>  block12_sepconv1_act (Ac  (None, 14, 14, 7  0       ['add_9[0][0]']
#>  tivation)                28)
#>
#>  block12_sepconv1 (Separa  (None, 14, 14, 7  536536  ['block12_sepconv1_act[0][0
#>  bleConv2D)               28)                        ]']
#>
#>  block12_sepconv1_bn (Bat  (None, 14, 14, 7  2912    ['block12_sepconv1[0][0]']
#>  chNormalization)         28)
```

```
#>
#>  block12_sepconv2_act (Ac  (None, 14, 14, 7  0       ['block12_sepconv1_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block12_sepconv2 (Separa  (None, 14, 14, 7  536536  ['block12_sepconv2_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block12_sepconv2_bn (Bat  (None, 14, 14, 7  2912    ['block12_sepconv2[0][0]']
#>  chNormalization)          28)
#>
#>  block12_sepconv3_act (Ac  (None, 14, 14, 7  0       ['block12_sepconv2_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block12_sepconv3 (Separa  (None, 14, 14, 7  536536  ['block12_sepconv3_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block12_sepconv3_bn (Bat  (None, 14, 14, 7  2912    ['block12_sepconv3[0][0]']
#>  chNormalization)          28)
#>
#>  add_10 (Add)              (None, 14, 14, 7  0        ['block12_sepconv3_bn[0][0]
#>                            28)                       ',
#>                                                       'add_9[0][0]']
#>
#>  block13_sepconv1_act (Ac  (None, 14, 14, 7  0       ['add_10[0][0]']
#>  tivation)                 28)
#>
#>  block13_sepconv1 (Separa  (None, 14, 14, 7  536536  ['block13_sepconv1_act[0][0
#>  bleConv2D)                28)                       ]']
#>
#>  block13_sepconv1_bn (Bat  (None, 14, 14, 7  2912    ['block13_sepconv1[0][0]']
#>  chNormalization)          28)
#>
#>  block13_sepconv2_act (Ac  (None, 14, 14, 7  0       ['block13_sepconv1_bn[0][0]
#>  tivation)                 28)                       ']
#>
#>  block13_sepconv2 (Separa  (None, 14, 14, 1  752024  ['block13_sepconv2_act[0][0
#>  bleConv2D)                024)                      ]']
#>
#>  block13_sepconv2_bn (Bat  (None, 14, 14, 1  4096    ['block13_sepconv2[0][0]']
#>  chNormalization)          024)
#>
#>  conv2d_3 (Conv2D)         (None, 7, 7, 102  745472   ['add_10[0][0]']
#>                            4)
#>
#>  block13_pool (MaxPooling  (None, 7, 7, 102  0       ['block13_sepconv2_bn[0][0]
#>  2D)                       4)                        ']
#>
#>  batch_normalization_3 (B  (None, 7, 7, 102  4096    ['conv2d_3[0][0]']
#>  atchNormalization)        4)
#>
#>  add_11 (Add)              (None, 7, 7, 102  0        ['block13_pool[0][0]',
#>                            4)                        'batch_normalization_3[0][
#>                                                      0]']
#>
#>  block14_sepconv1 (Separa  (None, 7, 7, 153  1582080  ['add_11[0][0]']
#>  bleConv2D)                6)
```

```
#>
#>  block14_sepconv1_bn (Bat  (None, 7, 7, 153  6144    ['block14_sepconv1[0][0]']
#>  chNormalization)         6)
#>
#>  block14_sepconv1_act (Ac  (None, 7, 7, 153  0       ['block14_sepconv1_bn[0][0]
#>  tivation)                6)                         ']
#>
#>  block14_sepconv2 (Separa  (None, 7, 7, 204  3159552  ['block14_sepconv1_act[0][0
#>  bleConv2D)               8)                        ]']
#>
#>  block14_sepconv2_bn (Bat  (None, 7, 7, 204  8192    ['block14_sepconv2[0][0]']
#>  chNormalization)         8)
#>
#>  block14_sepconv2_act (Ac  (None, 7, 7, 204  0       ['block14_sepconv2_bn[0][0]
#>  tivation)                8)                         ']
#>
#> ==================================================================================
#> Total params: 20,861,480
#> Trainable params: 0
#> Non-trainable params: 20,861,480
#> _________________________________________________________________________________
#> Model: "sequential"
#> _________________________________________________________________________________
#>  Layer (type)                        Output Shape                      Param #
#> =================================================================================
#>  xception (Functional)               (None, 7, 7, 2048)                20861480
#>
#>  global_average_pooling2d (GlobalAv  (None, 2048)                      0
#>  eragePooling2D)
#>
#>  dense_1 (Dense)                     (None, 1024)                      2098176
#>
#>  activation (Activation)             (None, 1024)                      0
#>
#>  dropout (Dropout)                   (None, 1024)                      0
#>
#>  dense (Dense)                       (None, 50)                        51250
#>
#> ==================================================================================
#> Total params: 23,010,906
#> Trainable params: 2,149,426
#> Non-trainable params: 20,861,480
#> _________________________________________________________________________________
```

We use a CNN to classify the images. Instead of defining our own model, we load a pretrained model to quickly get acceptable baseline results. In this case we load the xception network with the weights pre-trained on the ImageNet dataset. We freeze all weights in this pre-trained model, but couple it with another convolutional layer, which has weights that will be trained by us on our specific problem.

We train the model for the first time and save the entire model image. Notice that I was not able to install the tensorflow backend onto my NVIDIA GPU, which meant that the training process was very slow. Thus, instead of training it on my computer, effectively disabilitating it for a few hours, I trained the models on a large computer located at my home university, via ssh. On this computer the process is parallelized, meaning that it is trained faster. Moreover, since I was using tmux, which is a terminal multiplexer, I was able to detach the session and log out of the computer without the computations stopping. The reason I saved the entire model image was so that I could copy it to my computer (via scp), load it into my file and make predictions.

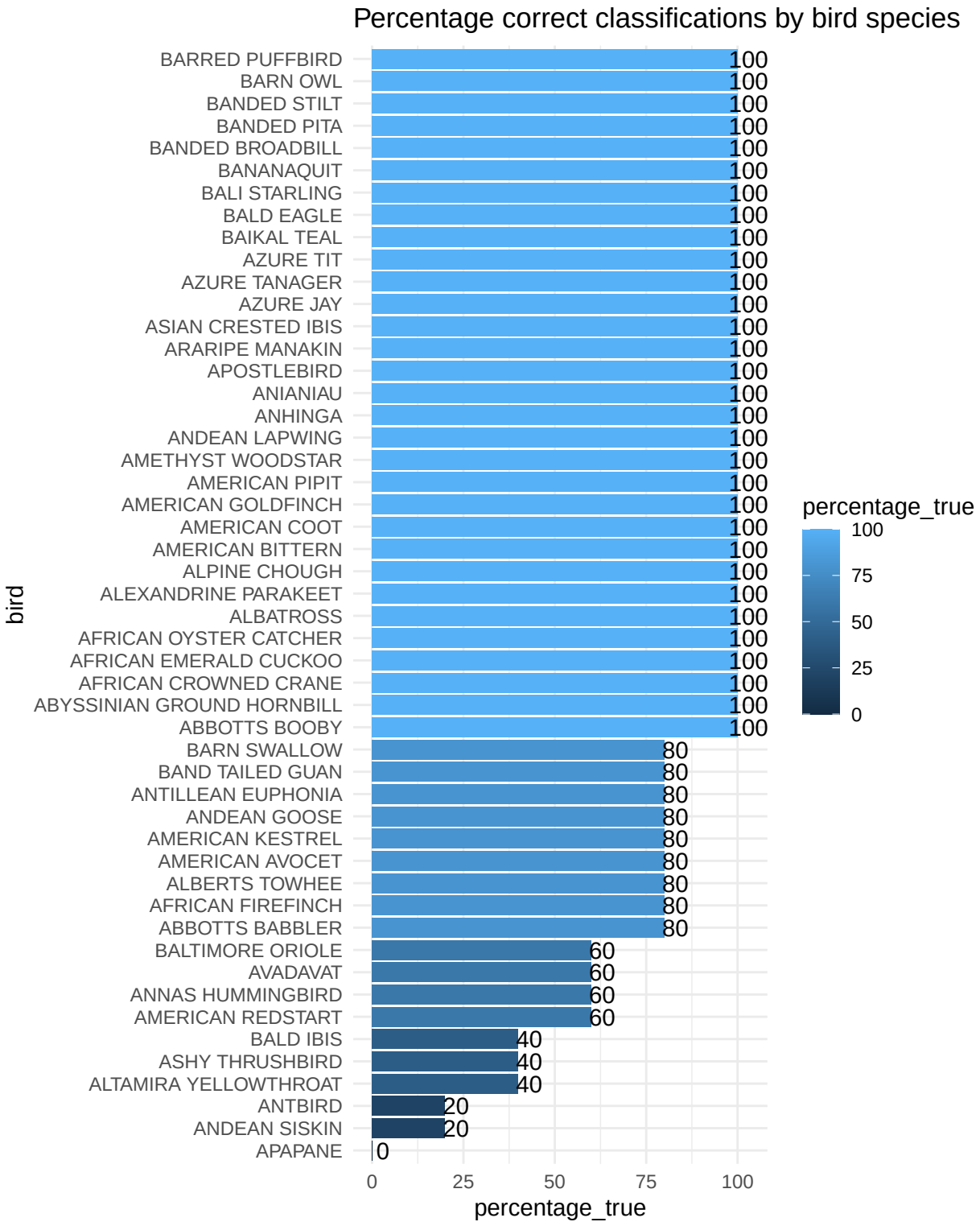
### Evaluate First Model

```
#>     loss accuracy
#> 0.506085 0.844000
```

```
#>                      Bird Probability
#> 1        ABBOTTS BABBLER 71.933359 %
#> 27               ANTBIRD  7.443158 %
#> 43            BANANAQUIT  6.662891 %
#> 19     AMERICAN REDSTART  5.563317 %
#> 12 ALTAMIRA YELLOWTHROAT  2.402629 %
```

After loading the fitted model, we evaluate the model on the test data, in order to see how well it has performed. 84% for the first model is the accuracy given, which is not bad for the first model, before tuning the hyperparameters. Then we predict on another image, which has not been used in training, taken from the Wikipedia page of Abbot's babbler, which is one of the bird species we have trained our model on. An overview of the model's predictions is made, showing the top 5 most probable bird species according to the model. The model gives the majority of the probability density (approximately 72%) to Abbot's babbler, meaning that it classifies the bird correctly, with some certainty. So far, not bad for a first model.

After this we have a look at which birds are well identified up against the birds that are not well identified. We can see that over half of the species are 100% correctly classified by our model in our test data. However, there are a few species that are mostly misclassified or not classified correctly at all, for instance "Apapane", which is never classified correctly in our test set.




## Model Tuning

We want to improve the performance of the model by tuning some of the hyperparameters. In the guide he used a brute force, self-made approach, using simple for-loops over the parameters he wanted to test. I will use tfruns, which is very similar, but seems slightly more sofisticated compared to his approach. This was also run on the computer located at my home university, naturally taking a longer time than the original computations, since a lot more models will be trained. We explore the same grid of hyperparameters as the author of the blog posts.

The **runs** directory as well as the saved **performance_table** are copied from the remote computer to mine via scp

after the tfruns hyperparameter tuning is done. The results from this tuning is shown.

Table 1: Validation Accuracy and Hyperparameter Values of all Trained Models

| Validation accuracy | Dropput rate | Learning rate | Dense nodes |
|---|---|---|---|
| 0.8556 | 0.2 | 1e-04 | 1024 |
| 0.8417 | 0.3 | 1e-04 | 1024 |
| 0.8194 | 0.3 | 1e-03 | 1024 |
| 0.8160 | 0.2 | 1e-04 | 256 |
| 0.8083 | 0.2 | 1e-03 | 1024 |
| 0.7944 | 0.3 | 1e-03 | 256 |
| 0.7889 | 0.2 | 1e-03 | 256 |
| 0.7847 | 0.3 | 1e-04 | 256 |

We can see that the best model according to the runs is the model fitted with hyperparameters

- Learning rate: 0.0001
- Dropout rate: 0.2
- Dense nodes: 1024

We train the model with these hyperparameters on the remote computer and save it as the final model that will be used in the app.

# Part 2 - Implementing a Shiny App

## Implementing the App Locally

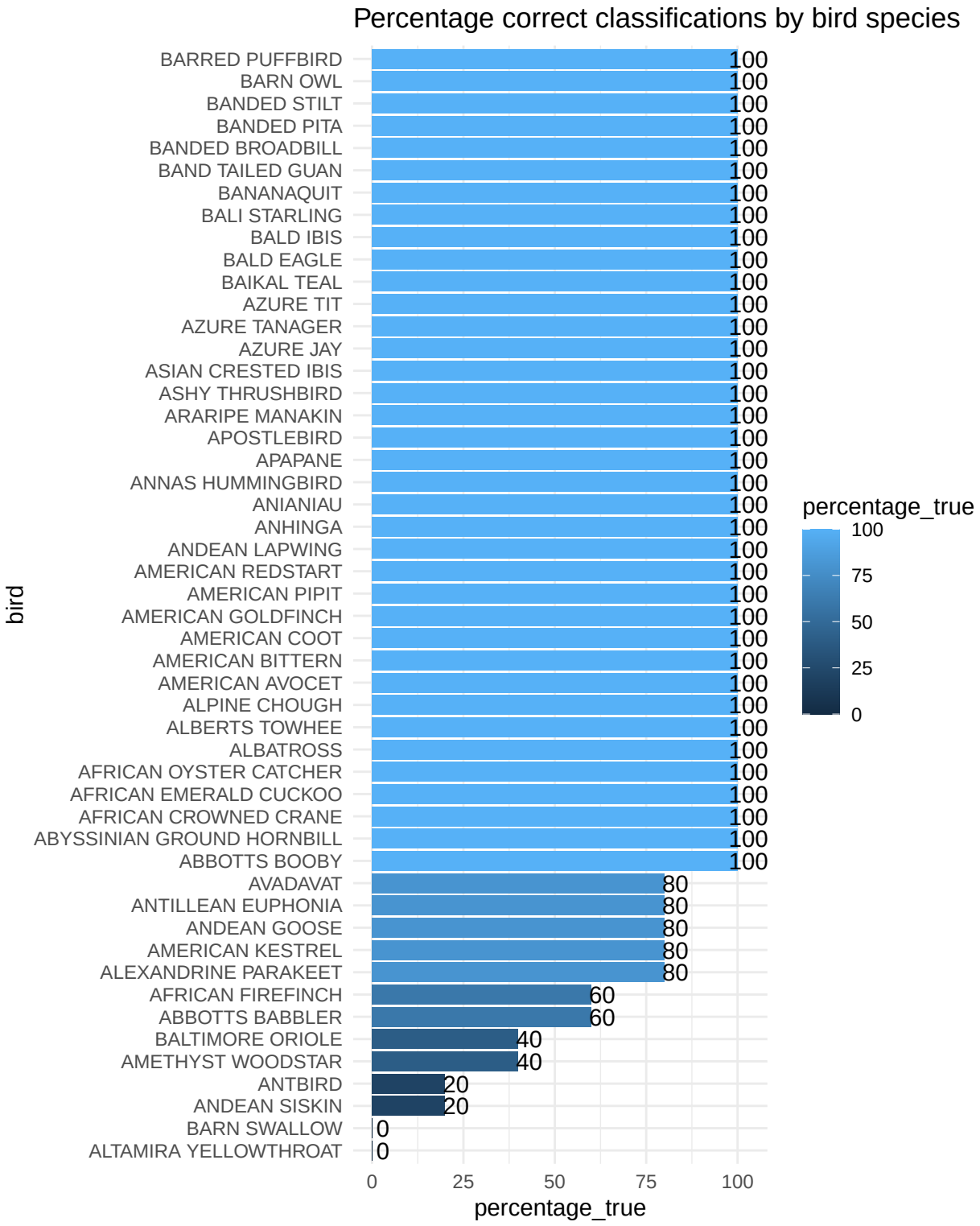
Copy the final model we trained into the "www" subdirectory of the "birdapp" directory. We also copy the label list, i.e. the list of birds, into the subdirectory.

First we define the ui object (User Interface). We use the dashboardPage function to create a dashboard page for the Shiny app. Inside the dashboardPage we define a dashboardHeader, a dashboardSidebar and a dashboardBody. Next we create the server object, which contains the interactive elements of the app. Inside the server function we load the image that is uploaded to the webapp by the user. Then we use the model we trained earlier to predict on the newly uploaded image. We create a dataframe with the top five predicted bird species after prediction, similar to what we did when testing the model earlier. Then we render it to the ui as a table, using the renderTable function. A warning text is defined, which display a warning to the visitor if the model has highest predicted probability below 45%. Finally we display the image that was loaded from the user and delete the file from memory.

## Evalution of New Model

```
#>      loss  accuracy
#> 0.5479541 0.8680000
```

```
#>                  Bird Probability
#> 1    ABBOTTS BABBLER 65.470839 %
#> 43        BANANAQUIT 10.877670 %
#> 19 AMERICAN REDSTART  7.057343 %
#> 9     ALBERTS TOWHEE  6.188409 %
#> 27           ANTBIRD  4.056965 %
```




## Deployment

We make a user on shinyapps.io and deploy the app there. DEPLOY AGAIN AFTER I AM DONE CHANING THE APP A BIT! Visit the app here.