Data Science and Economics

Algorithms for massive data, cloud and distributed computing

Eye disease recognition with NN

Andrea Joseph Froiio

922907

Index

Chapter 1

Introduction

The aim of this paper is to build a classifier able to detect between 8 classes of eye diseases using as input images of left and right eyes.
The classes are:

- **N**ormal
- **D**iabetic
- **G**laucoma
- **C**ataract
- **A**ge-related Macular Degeneration
- **H**ypertension
- **M**yopia
- **O**ther diseases/abnormalities

The model must be able to scale with data and keep working also when a huge dataset is used.

A ResNet50 will be used in combination with TensorFlow datasets to find out that some diseases are hard to detect, while for others there is the possibility of automated detection.

Chapter 2

Experimental results

2.1 Dataset

The data used for the experiment is a subset of the *Ocular Disease Intelligent Recognition* (**ODIR**) dataset, which is a structured ophthalmic database of 5000 patients with age, colour, fundus photographs from left and right eyes and doctors' diagnostic keywords.
The dataset is meant to represent "real-life" set of patient information collected by Shanggong Medical Technology Co., Ltd. from different hospitals/medical centers in China.
In these institutions, fundus images are captured by various cameras in the market, such as Canon, Zeiss and Kowa, resulting into varied image resolutions.
Annotations were labeled by trained human readers with quality control management.
Once downloaded and unzipped, the dataset presents itself like this:

- *ODIR-5K*
  - *ODIR-5K*
    - *Testing Images*, a folder containing 1000 images, of different sizes and with different aspect ratios.
    - *Training Images*, a folder containing 7000 images, of different sizes and with different aspect ratios.
    - *data.xlsx*, contains 3500 rows, one for each patient:
      1. **ID**, the ID of the patient.
      2. **Patient Age**, the age of the patient.
      3. **Patient Sex**, the sex of the patient.
      4. **Left-Fundus**, the name of the file for the left eye.
      5. **Right-Fundus**, the name of the file for the right eye.
      6. **Left-Diagnostic Keywords**, the doctors' diagnosis for the left eye.
      7. **Right-Diagnostic Keywords**, the doctors' diagnosis for the right eye.
      8. **N**, whether the patient has a normal fundus.
      9. **D**, whether the patient has diabetes.
      10. **G**, whether the patient has glaucoma.
      11. **C**, whether the patient has cataract.
      12. **A**, whether the patient has age-related macular degeneration.
      13. **H**, whether the patient has hypertension.
      14. **M**, whether the patient has myopia.
      15. **O**, whether the patient has some other disease/abnormality.

      Unfortunately we are missing testing labels as all the paths point to the files in the *Training Images* folder.
      Also, all "letter" columns for the diseases are by patient, so for both eyes.

- *preprocessed_images*, a folder containing 6392 images that have been cropped and resized to be all 512 × 512.
- *full_df.csv*, contains 6392 rows (one for each eye); having the same columns as *data.xlsx* but adding the following:

  16. **filepath**, the full path of the image (right eye in the first half of the data and left in the other).
      It is not corresponding to the path once downloaded on Colab.
  17. **labels**, a letter corresponding to an eye disease.
      They are labels for the patient, so for both eyes.
  18. **filename**, the name of the image file.
      Equal to the column *Right-Fundus* in the first half of the data, and to the *Left-Fundus* column in the other.

The images that will be used are the ones in the *preprocessed_images* folder, indicated by the *full_df.csv* file.

2.2 Evaluation strategy

The evaluation strategy will be based on 2 metrics:

- **Accuracy**:
$$\frac{TP + TN}{TP + FP + FN + TN}$$
  The classic measure of evaluation for classifiers.
- **F1 Score**:
$$2 \times \frac{precision \times recall}{precision + recall} = 2 \times \frac{\frac{TP}{TP+FP} \times \frac{TP}{TP+FN}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}}$$
  It is introduced when the classes in the data are unbalanced and/or skewed as in our case.
  The F1 Score is the harmonic mean of precision and recall.
  Where:
  - **Precision**:
$$\frac{TP}{TP + FP}$$
    What proportion of predicted positives is truly positive?
    It is a valid choice of evaluation metric when we want to be very sure of our prediction.
  - **Recall**:
$$\frac{TP}{TP + FN}$$
    What proportion of truly positives is correctly classified?
    It is a valid choice of evaluation metric when we want to capture as many positives as possible.

2.3 Experimental methodology

2.3.1 Data preparation

First of all, all columns but *target* and *filename* are dropped, since the task implies only the use of the images themselves:

target	**filename**
[1, 0, 0, 0, 0, 0, 0, 0]	0_right.jpg
...	...

Then the *target* column is one-hot encoded:

filename	**O**	**M**	**H**	**A**	**C**	**G**	**D**	**N**
0_right.jpg	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...

Since the classes are very unbalanced:

	N	**D**	**O**	**C**	**G**	**A**	**M**	**H**	Total
#images	2873	1608	708	293	284	266	232	128	**6392**

Data augmentation is put up to create new images starting from the ones we have.
The new images will be generated by rotating randomly the fundus in a 360°space and random brightness will be applied.
Augmentation is applied to each class (but **N**) the number of times equal to their (rounded) proportion with respect to the cardinality of the class with more images, minus 1, to keep and use the original images too:

	N	**D**	**O**	**C**	**G**	**A**	**M**	**H**	Total
factor	0	1	3	9	9	10	11	21	
new #images	2873	3216	2832	2930	2840	2926	2784	2816	**23217**

Augmentation computational times:

	CPU times	Wall time
user	00:14:57	-
sys	00:00:14	-
Total	**00:15:11**	**00:15:09**

2.3.2 Scalability

Data are split into Training set and Test set in a 90-10 proportion and then passed to TensorFlow to create two TensorFlow Dataset which work by training models in batches, and not overflow the RAM by loading all the images in main memory.

2.3.3 Model

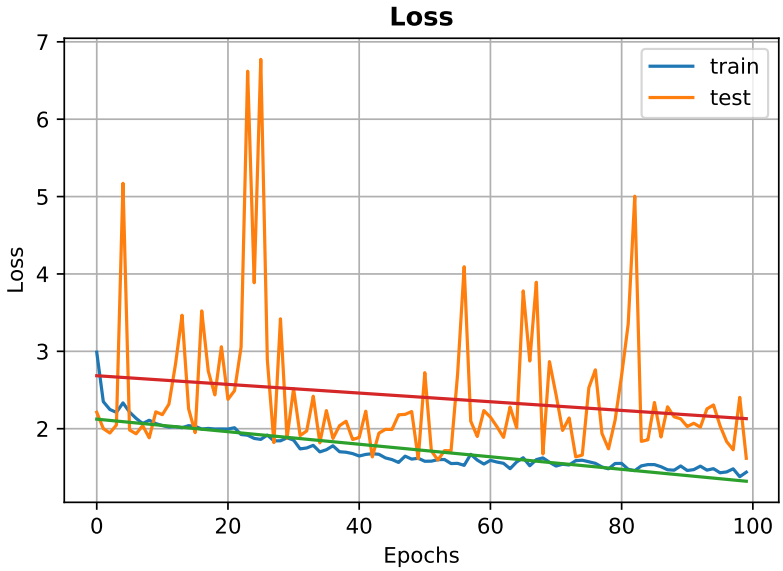
The model used is the ResNet50, which structure can be found in the Appendix at chapter 5 of this report.
*Res*idual *Net*works are a kind of Neural Network which solves in part the problem of vanishing gradients by skipping some connections between convolutional layers and re-adding inputs further down the convolution blocks.

Model training and testing times:

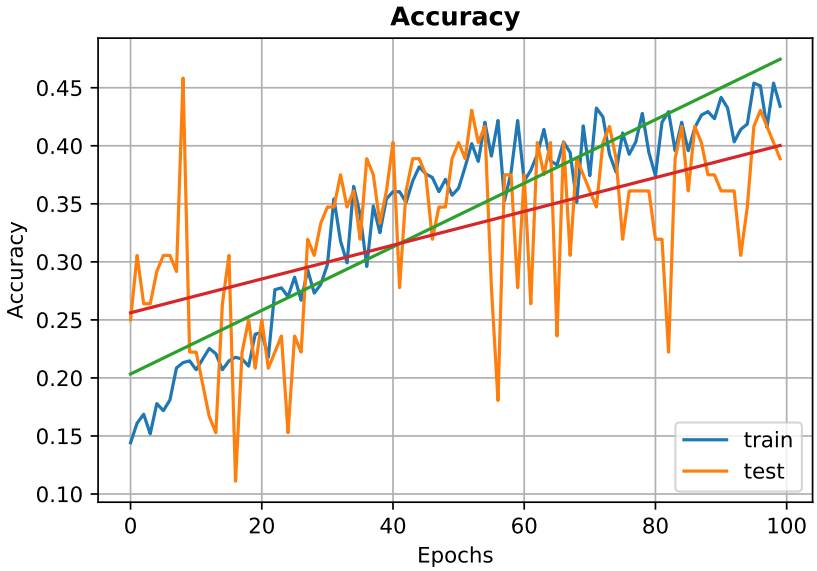
	CPU times	Wall time
user	01:27:57	-
sys	00:30:32	-
Total	**01:58:30**	**02:41:11**

Chapter 3

Concluding remarks



Train slope: -0.00810485 / Test slope: -0.00560649



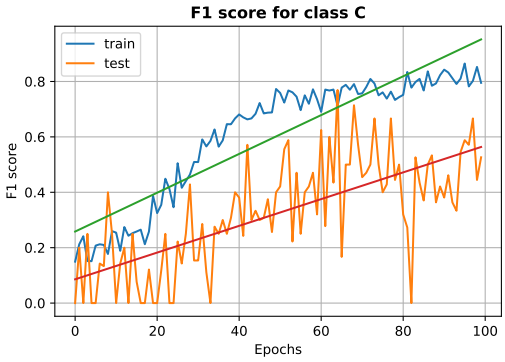
Train slope: 0.00273933 (0.27 %) / Test slope: 0.00145590 (0.15 %)

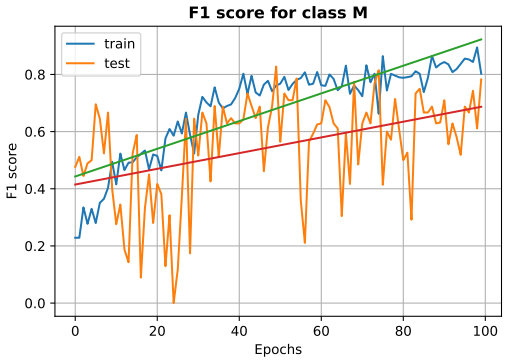
At the 100th epoch we have:

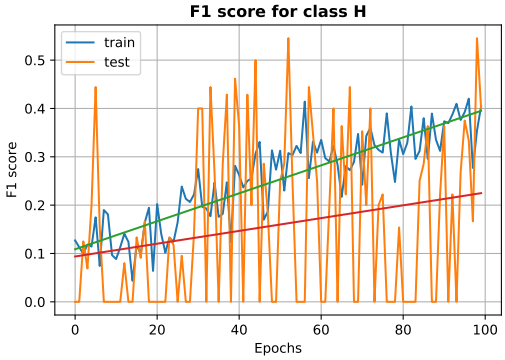
	Accuracy	**F1 score**
Train	0.4340	0.4128
Test	0.3889	0.3412

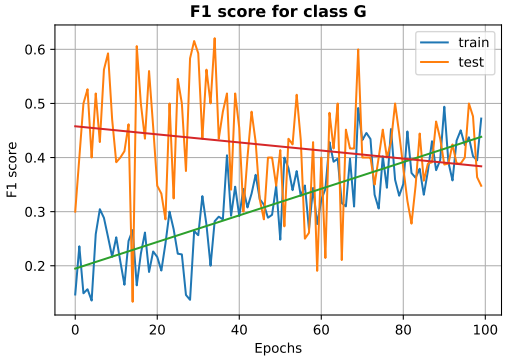
Which is not much but distinguishing between eye diseases is not an easy task, such that many of them look very similar even "to the human eye".

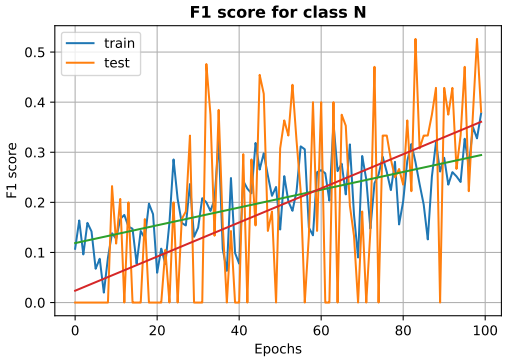
As said, classes have different difficulties because of their peculiar tolls on the fundus and it can be seen by looking at the F1 scores by class:

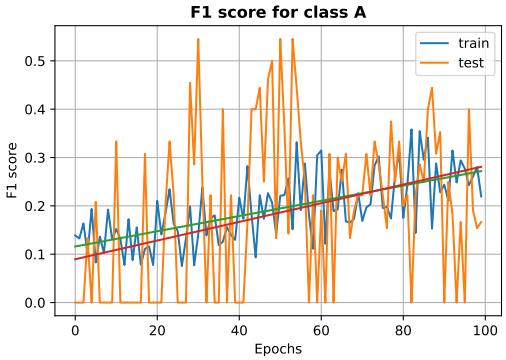


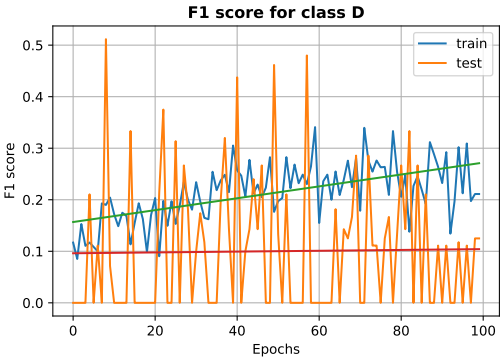


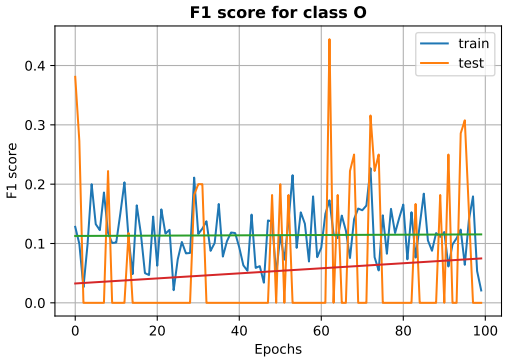












Class	*Average train % gain each epoch*	*Average test % gain each epoch*
Cataract	0.70057 %	0.48257 %
Myopia	0.48444 %	0.27480 %
Hypertension	0.28924 %	0.13236 %
Glaucoma	0.24597 %	-0.07472 %
Normal	0.17750 %	0.34056 %
AMD	0.15738 %	0.19316 %
Diabetes	0.11525 %	0.08048 %
Others	0.00264 %	0.04263 %

Multiplying the slope by 100 we can interpret it as the average % gained at each epoch

3.1 Comment

As we can clearly see **C**ataract is the easiest eye disease to detect, due to its peculiar fading of the eye lens, **M**yopia because of some darker spots around the eye and **H**ypertension and **G**laucoma for the presence of much red and red veins in the fundus. The other classes are difficult to detect because the fundus does not show any evident detail characteristic of that disease.

Another problem is the fact that photos come from different sources and may not have had the same original quality, and finally, the class **O** includes many other diseases not included in the other 7 classes that may vary too much from each other to be able to capture what really falls in it, taking a big toll on the overall accuracy.

The classifier could perform better with more images, with better and uniform quality, and without classes that contain many diseases.

3.2 Scalability

A bigger dataset can be easily handled with the TensorFlow object used, as it iterates in a streaming fashion (with batches), so that it does not need to fit all into the main memory, but if the images provided are still unbalanced, in different sizes or not enough, data preprocessing needs to be put in place on bigger scale, meaning longer computational times.

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

Chapter 4

References

1. Ocular Disease Recognition — Kaggle
2. Understand Deep Residual Networks
3. Keras applications — ResNet and ResNetV2
4. TensorFlow DataSets
5. Common eye disorders and diseases

Chapter 5

Appendix

```
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 512, 512, 3) 0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 518, 518, 3)  0           input_1[0][0]
__________________________________________________________________________________________________
conv1_conv (Conv2D)             (None, 256, 256, 64) 9472        conv1_pad[0][0]
__________________________________________________________________________________________________
conv1_bn (BatchNormalization)   (None, 256, 256, 64) 256         conv1_conv[0][0]
__________________________________________________________________________________________________
conv1_relu (Activation)         (None, 256, 256, 64) 0           conv1_bn[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 258, 258, 64) 0           conv1_relu[0][0]
__________________________________________________________________________________________________
pool1_pool (MaxPooling2D)       (None, 128, 128, 64) 0           pool1_pad[0][0]
__________________________________________________________________________________________________
conv2_block1_1_conv (Conv2D)    (None, 128, 128, 64) 4160        pool1_pool[0][0]
__________________________________________________________________________________________________
conv2_block1_1_bn (BatchNormali (None, 128, 128, 64) 256         conv2_block1_1_conv[0][0]
__________________________________________________________________________________________________
conv2_block1_1_relu (Activation (None, 128, 128, 64) 0           conv2_block1_1_bn[0][0]
__________________________________________________________________________________________________
conv2_block1_2_conv (Conv2D)    (None, 128, 128, 64) 36928       conv2_block1_1_relu[0][0]
__________________________________________________________________________________________________
conv2_block1_2_bn (BatchNormali (None, 128, 128, 64) 256         conv2_block1_2_conv[0][0]
__________________________________________________________________________________________________
conv2_block1_2_relu (Activation (None, 128, 128, 64) 0           conv2_block1_2_bn[0][0]
__________________________________________________________________________________________________
conv2_block1_0_conv (Conv2D)    (None, 128, 128, 256 16640       pool1_pool[0][0]
__________________________________________________________________________________________________
conv2_block1_3_conv (Conv2D)    (None, 128, 128, 256 16640       conv2_block1_2_relu[0][0]
__________________________________________________________________________________________________
conv2_block1_0_bn (BatchNormali (None, 128, 128, 256 1024        conv2_block1_0_conv[0][0]
__________________________________________________________________________________________________
conv2_block1_3_bn (BatchNormali (None, 128, 128, 256 1024        conv2_block1_3_conv[0][0]
__________________________________________________________________________________________________
conv2_block1_add (Add)          (None, 128, 128, 256 0           conv2_block1_0_bn[0][0]
                                                                 conv2_block1_3_bn[0][0]
__________________________________________________________________________________________________
```

```
conv2_block1_out (Activation)   (None, 128, 128, 256 0           conv2_block1_add[0][0]
__________________________________________________________________________________________________
conv2_block2_1_conv (Conv2D)    (None, 128, 128, 64) 16448       conv2_block1_out[0][0]
__________________________________________________________________________________________________
conv2_block2_1_bn (BatchNormali (None, 128, 128, 64) 256         conv2_block2_1_conv[0][0]
__________________________________________________________________________________________________
conv2_block2_1_relu (Activation (None, 128, 128, 64) 0           conv2_block2_1_bn[0][0]
__________________________________________________________________________________________________
conv2_block2_2_conv (Conv2D)    (None, 128, 128, 64) 36928       conv2_block2_1_relu[0][0]
__________________________________________________________________________________________________
conv2_block2_2_bn (BatchNormali (None, 128, 128, 64) 256         conv2_block2_2_conv[0][0]
__________________________________________________________________________________________________
conv2_block2_2_relu (Activation (None, 128, 128, 64) 0           conv2_block2_2_bn[0][0]
__________________________________________________________________________________________________
conv2_block2_3_conv (Conv2D)    (None, 128, 128, 256 16640       conv2_block2_2_relu[0][0]
__________________________________________________________________________________________________
conv2_block2_3_bn (BatchNormali (None, 128, 128, 256 1024        conv2_block2_3_conv[0][0]
__________________________________________________________________________________________________
conv2_block2_add (Add)          (None, 128, 128, 256 0           conv2_block1_out[0][0]
                                                                 conv2_block2_3_bn[0][0]
__________________________________________________________________________________________________
conv2_block2_out (Activation)   (None, 128, 128, 256 0           conv2_block2_add[0][0]
__________________________________________________________________________________________________
conv2_block3_1_conv (Conv2D)    (None, 128, 128, 64) 16448       conv2_block2_out[0][0]
__________________________________________________________________________________________________
conv2_block3_1_bn (BatchNormali (None, 128, 128, 64) 256         conv2_block3_1_conv[0][0]
__________________________________________________________________________________________________
conv2_block3_1_relu (Activation (None, 128, 128, 64) 0           conv2_block3_1_bn[0][0]
__________________________________________________________________________________________________
conv2_block3_2_conv (Conv2D)    (None, 128, 128, 64) 36928       conv2_block3_1_relu[0][0]
__________________________________________________________________________________________________
conv2_block3_2_bn (BatchNormali (None, 128, 128, 64) 256         conv2_block3_2_conv[0][0]
__________________________________________________________________________________________________
conv2_block3_2_relu (Activation (None, 128, 128, 64) 0           conv2_block3_2_bn[0][0]
__________________________________________________________________________________________________
conv2_block3_3_conv (Conv2D)    (None, 128, 128, 256 16640       conv2_block3_2_relu[0][0]
__________________________________________________________________________________________________
conv2_block3_3_bn (BatchNormali (None, 128, 128, 256 1024        conv2_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv2_block3_add (Add)          (None, 128, 128, 256 0           conv2_block2_out[0][0]
                                                                 conv2_block3_3_bn[0][0]
__________________________________________________________________________________________________
conv2_block3_out (Activation)   (None, 128, 128, 256 0           conv2_block3_add[0][0]
__________________________________________________________________________________________________
conv3_block1_1_conv (Conv2D)    (None, 64, 64, 128)  32896       conv2_block3_out[0][0]
__________________________________________________________________________________________________
conv3_block1_1_bn (BatchNormali (None, 64, 64, 128)  512         conv3_block1_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block1_1_relu (Activation (None, 64, 64, 128)  0           conv3_block1_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block1_2_conv (Conv2D)    (None, 64, 64, 128)  147584      conv3_block1_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block1_2_bn (BatchNormali (None, 64, 64, 128)  512         conv3_block1_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block1_2_relu (Activation (None, 64, 64, 128)  0           conv3_block1_2_bn[0][0]
__________________________________________________________________________________________________
```

```
conv3_block1_0_conv (Conv2D)    (None, 64, 64, 512)  131584      conv2_block3_out[0][0]
__________________________________________________________________________________________________
conv3_block1_3_conv (Conv2D)    (None, 64, 64, 512)  66048       conv3_block1_2_relu[0][0]
__________________________________________________________________________________________________
conv3_block1_0_bn (BatchNormali (None, 64, 64, 512)  2048        conv3_block1_0_conv[0][0]
__________________________________________________________________________________________________
conv3_block1_3_bn (BatchNormali (None, 64, 64, 512)  2048        conv3_block1_3_conv[0][0]
__________________________________________________________________________________________________
conv3_block1_add (Add)          (None, 64, 64, 512)  0           conv3_block1_0_bn[0][0]
                                                                 conv3_block1_3_bn[0][0]
__________________________________________________________________________________________________
conv3_block1_out (Activation)   (None, 64, 64, 512)  0           conv3_block1_add[0][0]
__________________________________________________________________________________________________
conv3_block2_1_conv (Conv2D)    (None, 64, 64, 128)  65664       conv3_block1_out[0][0]
__________________________________________________________________________________________________
conv3_block2_1_bn (BatchNormali (None, 64, 64, 128)  512         conv3_block2_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block2_1_relu (Activation (None, 64, 64, 128)  0           conv3_block2_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block2_2_conv (Conv2D)    (None, 64, 64, 128)  147584      conv3_block2_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block2_2_bn (BatchNormali (None, 64, 64, 128)  512         conv3_block2_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block2_2_relu (Activation (None, 64, 64, 128)  0           conv3_block2_2_bn[0][0]
__________________________________________________________________________________________________
conv3_block2_3_conv (Conv2D)    (None, 64, 64, 512)  66048       conv3_block2_2_relu[0][0]
__________________________________________________________________________________________________
conv3_block2_3_bn (BatchNormali (None, 64, 64, 512)  2048        conv3_block2_3_conv[0][0]
__________________________________________________________________________________________________
conv3_block2_add (Add)          (None, 64, 64, 512)  0           conv3_block1_out[0][0]
                                                                 conv3_block2_3_bn[0][0]
__________________________________________________________________________________________________
conv3_block2_out (Activation)   (None, 64, 64, 512)  0           conv3_block2_add[0][0]
__________________________________________________________________________________________________
conv3_block3_1_conv (Conv2D)    (None, 64, 64, 128)  65664       conv3_block2_out[0][0]
__________________________________________________________________________________________________
conv3_block3_1_bn (BatchNormali (None, 64, 64, 128)  512         conv3_block3_1_conv[0][0]
__________________________________________________________________________________________________
conv3_block3_1_relu (Activation (None, 64, 64, 128)  0           conv3_block3_1_bn[0][0]
__________________________________________________________________________________________________
conv3_block3_2_conv (Conv2D)    (None, 64, 64, 128)  147584      conv3_block3_1_relu[0][0]
__________________________________________________________________________________________________
conv3_block3_2_bn (BatchNormali (None, 64, 64, 128)  512         conv3_block3_2_conv[0][0]
__________________________________________________________________________________________________
conv3_block3_2_relu (Activation (None, 64, 64, 128)  0           conv3_block3_2_bn[0][0]
__________________________________________________________________________________________________
conv3_block3_3_conv (Conv2D)    (None, 64, 64, 512)  66048       conv3_block3_2_relu[0][0]
__________________________________________________________________________________________________
conv3_block3_3_bn (BatchNormali (None, 64, 64, 512)  2048        conv3_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv3_block3_add (Add)          (None, 64, 64, 512)  0           conv3_block2_out[0][0]
                                                                 conv3_block3_3_bn[0][0]
__________________________________________________________________________________________________
conv3_block3_out (Activation)   (None, 64, 64, 512)  0           conv3_block3_add[0][0]
__________________________________________________________________________________________________
conv3_block4_1_conv (Conv2D)    (None, 64, 64, 128)  65664       conv3_block3_out[0][0]
```

```
conv3_block4_1_bn (BatchNormali (None, 64, 64, 128)  512         conv3_block4_1_conv[0][0]
conv3_block4_1_relu (Activation (None, 64, 64, 128)  0           conv3_block4_1_bn[0][0]
conv3_block4_2_conv (Conv2D)    (None, 64, 64, 128)  147584      conv3_block4_1_relu[0][0]
conv3_block4_2_bn (BatchNormali (None, 64, 64, 128)  512         conv3_block4_2_conv[0][0]
conv3_block4_2_relu (Activation (None, 64, 64, 128)  0           conv3_block4_2_bn[0][0]
conv3_block4_3_conv (Conv2D)    (None, 64, 64, 512)  66048       conv3_block4_2_relu[0][0]
conv3_block4_3_bn (BatchNormali (None, 64, 64, 512)  2048        conv3_block4_3_conv[0][0]
conv3_block4_add (Add)          (None, 64, 64, 512)  0           conv3_block3_out[0][0]
                                                                 conv3_block4_3_bn[0][0]
conv3_block4_out (Activation)   (None, 64, 64, 512)  0           conv3_block4_add[0][0]
conv4_block1_1_conv (Conv2D)    (None, 32, 32, 256)  131328      conv3_block4_out[0][0]
conv4_block1_1_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block1_1_conv[0][0]
conv4_block1_1_relu (Activation (None, 32, 32, 256)  0           conv4_block1_1_bn[0][0]
conv4_block1_2_conv (Conv2D)    (None, 32, 32, 256)  590080      conv4_block1_1_relu[0][0]
conv4_block1_2_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block1_2_conv[0][0]
conv4_block1_2_relu (Activation (None, 32, 32, 256)  0           conv4_block1_2_bn[0][0]
conv4_block1_0_conv (Conv2D)    (None, 32, 32, 1024) 525312      conv3_block4_out[0][0]
conv4_block1_3_conv (Conv2D)    (None, 32, 32, 1024) 263168      conv4_block1_2_relu[0][0]
conv4_block1_0_bn (BatchNormali (None, 32, 32, 1024) 4096        conv4_block1_0_conv[0][0]
conv4_block1_3_bn (BatchNormali (None, 32, 32, 1024) 4096        conv4_block1_3_conv[0][0]
conv4_block1_add (Add)          (None, 32, 32, 1024) 0           conv4_block1_0_bn[0][0]
                                                                 conv4_block1_3_bn[0][0]
conv4_block1_out (Activation)   (None, 32, 32, 1024) 0           conv4_block1_add[0][0]
conv4_block2_1_conv (Conv2D)    (None, 32, 32, 256)  262400      conv4_block1_out[0][0]
conv4_block2_1_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block2_1_conv[0][0]
conv4_block2_1_relu (Activation (None, 32, 32, 256)  0           conv4_block2_1_bn[0][0]
conv4_block2_2_conv (Conv2D)    (None, 32, 32, 256)  590080      conv4_block2_1_relu[0][0]
conv4_block2_2_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block2_2_conv[0][0]
conv4_block2_2_relu (Activation (None, 32, 32, 256)  0           conv4_block2_2_bn[0][0]
```

```
conv4_block2_3_conv (Conv2D)    (None, 32, 32, 1024) 263168      conv4_block2_2_relu[0][0]
conv4_block2_3_bn (BatchNormali (None, 32, 32, 1024) 4096        conv4_block2_3_conv[0][0]
conv4_block2_add (Add)          (None, 32, 32, 1024) 0           conv4_block1_out[0][0]
                                                                 conv4_block2_3_bn[0][0]
conv4_block2_out (Activation)   (None, 32, 32, 1024) 0           conv4_block2_add[0][0]
conv4_block3_1_conv (Conv2D)    (None, 32, 32, 256)  262400      conv4_block2_out[0][0]
conv4_block3_1_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block3_1_conv[0][0]
conv4_block3_1_relu (Activation (None, 32, 32, 256)  0           conv4_block3_1_bn[0][0]
conv4_block3_2_conv (Conv2D)    (None, 32, 32, 256)  590080      conv4_block3_1_relu[0][0]
conv4_block3_2_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block3_2_conv[0][0]
conv4_block3_2_relu (Activation (None, 32, 32, 256)  0           conv4_block3_2_bn[0][0]
conv4_block3_3_conv (Conv2D)    (None, 32, 32, 1024) 263168      conv4_block3_2_relu[0][0]
conv4_block3_3_bn (BatchNormali (None, 32, 32, 1024) 4096        conv4_block3_3_conv[0][0]
conv4_block3_add (Add)          (None, 32, 32, 1024) 0           conv4_block2_out[0][0]
                                                                 conv4_block3_3_bn[0][0]
conv4_block3_out (Activation)   (None, 32, 32, 1024) 0           conv4_block3_add[0][0]
conv4_block4_1_conv (Conv2D)    (None, 32, 32, 256)  262400      conv4_block3_out[0][0]
conv4_block4_1_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block4_1_conv[0][0]
conv4_block4_1_relu (Activation (None, 32, 32, 256)  0           conv4_block4_1_bn[0][0]
conv4_block4_2_conv (Conv2D)    (None, 32, 32, 256)  590080      conv4_block4_1_relu[0][0]
conv4_block4_2_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block4_2_conv[0][0]
conv4_block4_2_relu (Activation (None, 32, 32, 256)  0           conv4_block4_2_bn[0][0]
conv4_block4_3_conv (Conv2D)    (None, 32, 32, 1024) 263168      conv4_block4_2_relu[0][0]
conv4_block4_3_bn (BatchNormali (None, 32, 32, 1024) 4096        conv4_block4_3_conv[0][0]
conv4_block4_add (Add)          (None, 32, 32, 1024) 0           conv4_block3_out[0][0]
                                                                 conv4_block4_3_bn[0][0]
conv4_block4_out (Activation)   (None, 32, 32, 1024) 0           conv4_block4_add[0][0]
conv4_block5_1_conv (Conv2D)    (None, 32, 32, 256)  262400      conv4_block4_out[0][0]
conv4_block5_1_bn (BatchNormali (None, 32, 32, 256)  1024        conv4_block5_1_conv[0][0]
```

```
conv4_block5_1_relu (Activation (None, 32, 32, 256)  0          conv4_block5_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block5_2_conv (Conv2D)    (None, 32, 32, 256)  590080     conv4_block5_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block5_2_bn (BatchNormali (None, 32, 32, 256)  1024       conv4_block5_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block5_2_relu (Activation (None, 32, 32, 256)  0          conv4_block5_2_bn[0][0]
__________________________________________________________________________________________________
conv4_block5_3_conv (Conv2D)    (None, 32, 32, 1024) 263168     conv4_block5_2_relu[0][0]
__________________________________________________________________________________________________
conv4_block5_3_bn (BatchNormali (None, 32, 32, 1024) 4096       conv4_block5_3_conv[0][0]
__________________________________________________________________________________________________
conv4_block5_add (Add)          (None, 32, 32, 1024) 0          conv4_block4_out[0][0]
                                                                conv4_block5_3_bn[0][0]
__________________________________________________________________________________________________
conv4_block5_out (Activation)   (None, 32, 32, 1024) 0          conv4_block5_add[0][0]
__________________________________________________________________________________________________
conv4_block6_1_conv (Conv2D)    (None, 32, 32, 256)  262400     conv4_block5_out[0][0]
__________________________________________________________________________________________________
conv4_block6_1_bn (BatchNormali (None, 32, 32, 256)  1024       conv4_block6_1_conv[0][0]
__________________________________________________________________________________________________
conv4_block6_1_relu (Activation (None, 32, 32, 256)  0          conv4_block6_1_bn[0][0]
__________________________________________________________________________________________________
conv4_block6_2_conv (Conv2D)    (None, 32, 32, 256)  590080     conv4_block6_1_relu[0][0]
__________________________________________________________________________________________________
conv4_block6_2_bn (BatchNormali (None, 32, 32, 256)  1024       conv4_block6_2_conv[0][0]
__________________________________________________________________________________________________
conv4_block6_2_relu (Activation (None, 32, 32, 256)  0          conv4_block6_2_bn[0][0]
__________________________________________________________________________________________________
conv4_block6_3_conv (Conv2D)    (None, 32, 32, 1024) 263168     conv4_block6_2_relu[0][0]
__________________________________________________________________________________________________
conv4_block6_3_bn (BatchNormali (None, 32, 32, 1024) 4096       conv4_block6_3_conv[0][0]
__________________________________________________________________________________________________
conv4_block6_add (Add)          (None, 32, 32, 1024) 0          conv4_block5_out[0][0]
                                                                conv4_block6_3_bn[0][0]
__________________________________________________________________________________________________
conv4_block6_out (Activation)   (None, 32, 32, 1024) 0          conv4_block6_add[0][0]
__________________________________________________________________________________________________
conv5_block1_1_conv (Conv2D)    (None, 16, 16, 512)  524800     conv4_block6_out[0][0]
__________________________________________________________________________________________________
conv5_block1_1_bn (BatchNormali (None, 16, 16, 512)  2048       conv5_block1_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block1_1_relu (Activation (None, 16, 16, 512)  0          conv5_block1_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block1_2_conv (Conv2D)    (None, 16, 16, 512)  2359808    conv5_block1_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block1_2_bn (BatchNormali (None, 16, 16, 512)  2048       conv5_block1_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block1_2_relu (Activation (None, 16, 16, 512)  0          conv5_block1_2_bn[0][0]
__________________________________________________________________________________________________
conv5_block1_0_conv (Conv2D)    (None, 16, 16, 2048) 2099200    conv4_block6_out[0][0]
__________________________________________________________________________________________________
conv5_block1_3_conv (Conv2D)    (None, 16, 16, 2048) 1050624    conv5_block1_2_relu[0][0]
__________________________________________________________________________________________________
conv5_block1_0_bn (BatchNormali (None, 16, 16, 2048) 8192       conv5_block1_0_conv[0][0]
__________________________________________________________________________________________________
```

```
conv5_block1_3_bn (BatchNormali (None, 16, 16, 2048) 8192        conv5_block1_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block1_add (Add)          (None, 16, 16, 2048) 0           conv5_block1_0_bn[0][0]
                                                                 conv5_block1_3_bn[0][0]
__________________________________________________________________________________________________
conv5_block1_out (Activation)   (None, 16, 16, 2048) 0           conv5_block1_add[0][0]
__________________________________________________________________________________________________
conv5_block2_1_conv (Conv2D)    (None, 16, 16, 512)  1049088     conv5_block1_out[0][0]
__________________________________________________________________________________________________
conv5_block2_1_bn (BatchNormali (None, 16, 16, 512)  2048        conv5_block2_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block2_1_relu (Activation (None, 16, 16, 512)  0           conv5_block2_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block2_2_conv (Conv2D)    (None, 16, 16, 512)  2359808     conv5_block2_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block2_2_bn (BatchNormali (None, 16, 16, 512)  2048        conv5_block2_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block2_2_relu (Activation (None, 16, 16, 512)  0           conv5_block2_2_bn[0][0]
__________________________________________________________________________________________________
conv5_block2_3_conv (Conv2D)    (None, 16, 16, 2048) 1050624     conv5_block2_2_relu[0][0]
__________________________________________________________________________________________________
conv5_block2_3_bn (BatchNormali (None, 16, 16, 2048) 8192        conv5_block2_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block2_add (Add)          (None, 16, 16, 2048) 0           conv5_block1_out[0][0]
                                                                 conv5_block2_3_bn[0][0]
__________________________________________________________________________________________________
conv5_block2_out (Activation)   (None, 16, 16, 2048) 0           conv5_block2_add[0][0]
__________________________________________________________________________________________________
conv5_block3_1_conv (Conv2D)    (None, 16, 16, 512)  1049088     conv5_block2_out[0][0]
__________________________________________________________________________________________________
conv5_block3_1_bn (BatchNormali (None, 16, 16, 512)  2048        conv5_block3_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_1_relu (Activation (None, 16, 16, 512)  0           conv5_block3_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_2_conv (Conv2D)    (None, 16, 16, 512)  2359808     conv5_block3_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block3_2_bn (BatchNormali (None, 16, 16, 512)  2048        conv5_block3_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_2_relu (Activation (None, 16, 16, 512)  0           conv5_block3_2_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_3_conv (Conv2D)    (None, 16, 16, 2048) 1050624     conv5_block3_2_relu[0][0]
__________________________________________________________________________________________________
conv5_block3_3_bn (BatchNormali (None, 16, 16, 2048) 8192        conv5_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_add (Add)          (None, 16, 16, 2048) 0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_out (Activation)   (None, 16, 16, 2048) 0           conv5_block3_add[0][0]
__________________________________________________________________________________________________
global_average_pooling2d (Globa (None, 2048)         0           conv5_block3_out[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 2048)         0           global_average_pooling2d[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 8)            16392       dropout[0][0]
==================================================================================================
Total params: 23,604,104         Trainable params: 23,550,984    Non-trainable params: 53,120
```