```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!unzip "/content/drive/MyDrive/Colab Notebooks/Deeplearning1/archive (1).zip"
```

```
  inflating: myData/8/00018_00001.jpg
  inflating: myData/8/00018_00002.jpg
  inflating: myData/8/00018_00003.jpg
  inflating: myData/8/00018_00004.jpg
  inflating: myData/8/00018_00005.jpg
  inflating: myData/8/00018_00006.jpg
  inflating: myData/8/00018_00007.jpg
  inflating: myData/8/00018_00008.jpg
  inflating: myData/8/00018_00009.jpg
  inflating: myData/8/00018_00010.jpg
  inflating: myData/8/00018_00011.jpg
  inflating: myData/8/00018_00012.jpg
  inflating: myData/8/00018_00013.jpg
  inflating: myData/8/00018_00014.jpg
  inflating: myData/8/00018_00015.jpg
  inflating: myData/8/00018_00016.jpg
  inflating: myData/8/00018_00017.jpg
  inflating: myData/8/00018_00018.jpg
  inflating: myData/8/00018_00019.jpg
  inflating: myData/8/00018_00020.jpg
  inflating: myData/8/00018_00021.jpg
  inflating: myData/8/00018_00022.jpg
  inflating: myData/8/00018_00023.jpg
  inflating: myData/8/00018_00024.jpg
  inflating: myData/8/00018_00025.jpg
  inflating: myData/8/00018_00026.jpg
  inflating: myData/8/00018_00027.jpg
  inflating: myData/8/00018_00028.jpg
  inflating: myData/8/00018_00029.jpg
  inflating: myData/8/00019_00000.jpg
  inflating: myData/8/00019_00001.jpg
  inflating: myData/8/00019_00002.jpg
  inflating: myData/8/00019_00003.jpg
  inflating: myData/8/00019_00004.jpg
  inflating: myData/8/00019_00005.jpg
  inflating: myData/8/00019_00006.jpg
  inflating: myData/8/00019_00007.jpg
  inflating: myData/8/00019_00008.jpg
  inflating: myData/8/00019_00009.jpg
  inflating: myData/8/00019_00010.jpg
  inflating: myData/8/00019_00011.jpg
  inflating: myData/8/00019_00012.jpg
  inflating: myData/8/00019_00013.jpg
  inflating: myData/8/00019_00014.jpg
  inflating: myData/8/00019_00015.jpg
  inflating: myData/8/00019_00016.jpg
```

```
  inflating: myData/8/00019_00017.jpg
  inflating: myData/8/00019_00018.jpg
  inflating: myData/8/00019_00019.jpg
  inflating: myData/8/00019_00020.jpg
  inflating: myData/8/00019_00021.jpg
  inflating: myData/8/00019_00022.jpg
  inflating: myData/8/00019_00023.jpg
  inflating: myData/8/00019_00024.jpg
  inflating: myData/8/00019_00025.jpg
  inflating: myData/8/00019_00026.jpg
  inflating: myData/8/00019_00027.jpg
  inflating: myData/8/00019_00028.jpg
```

```
%cd "/content/drive/MyDrive/Colab Notebooks/Deeplearning1"
```

```
/content/drive/MyDrive/Colab Notebooks/Deeplearning1
```

Taffic Sign Classification

Traffic Sign Classification Using **Convolutional Neural Networks(CNNs)**, Which is widely used in various applications in the field of Artificial Intelligence. This notebook focuses on developing a deep learning model in order to classify the traffic signs. 🛑❌🚫🚷🚳


```
import numpy as np
import pandas as pd
import os
import keras
```

Importing Libraries

```
# Visualization
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('whitegrid')
%matplotlib inline
```

```
from tensorflow.keras.utils import plot_model

# Splitting data
from sklearn.model_selection import train_test_split

# Metrics
from sklearn.metrics import confusion_matrix, classification_report

# Deep Learning
import tensorflow as tf
print('TensoFlow Version: ', tf.__version__)
from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Flatten, GlobalAveragePooling2D, BatchNormalizatio
from tensorflow.keras.applications.resnet import ResNet50

from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau, CSV
```

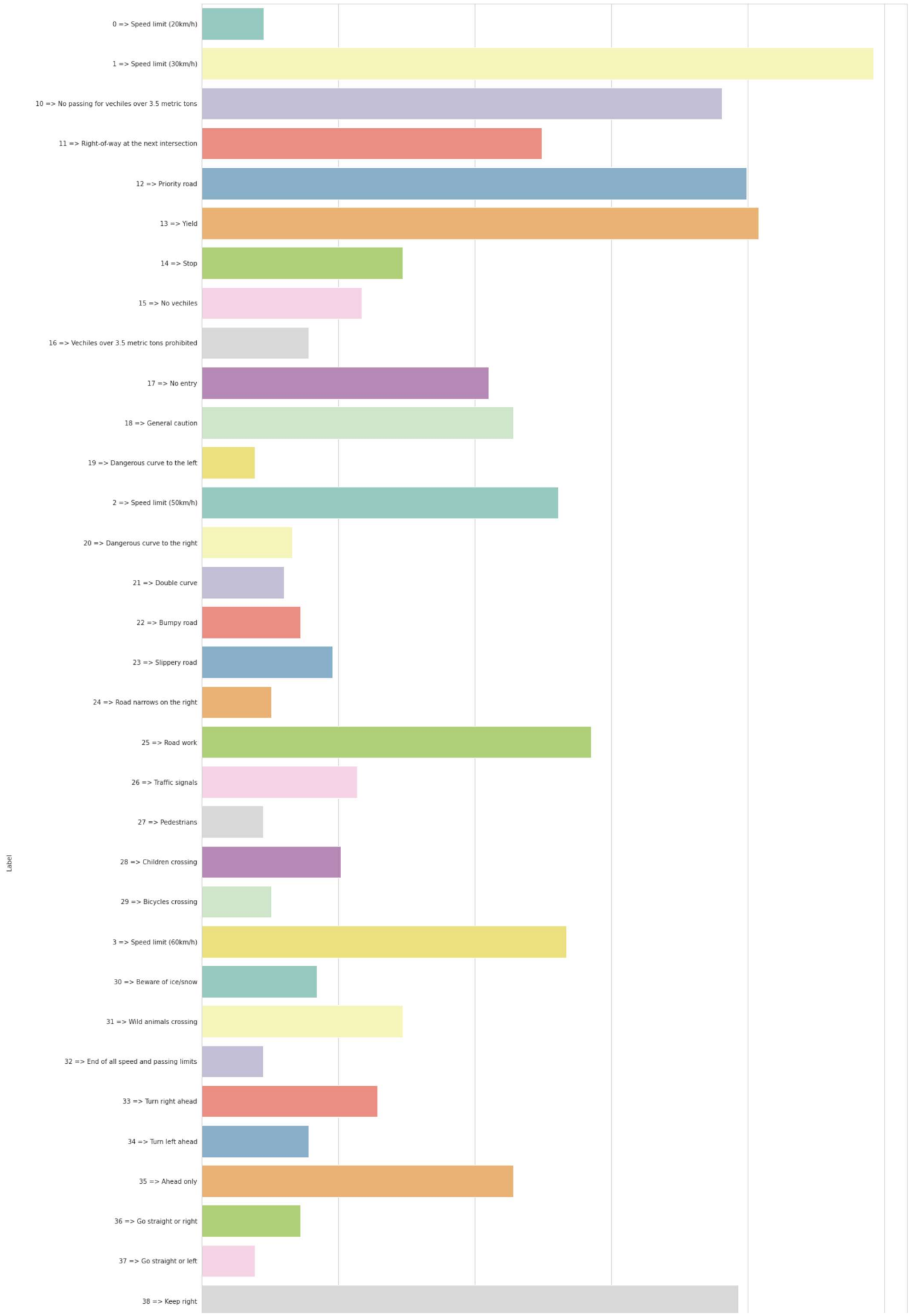
```
TensoFlow Version:  2.8.0
```

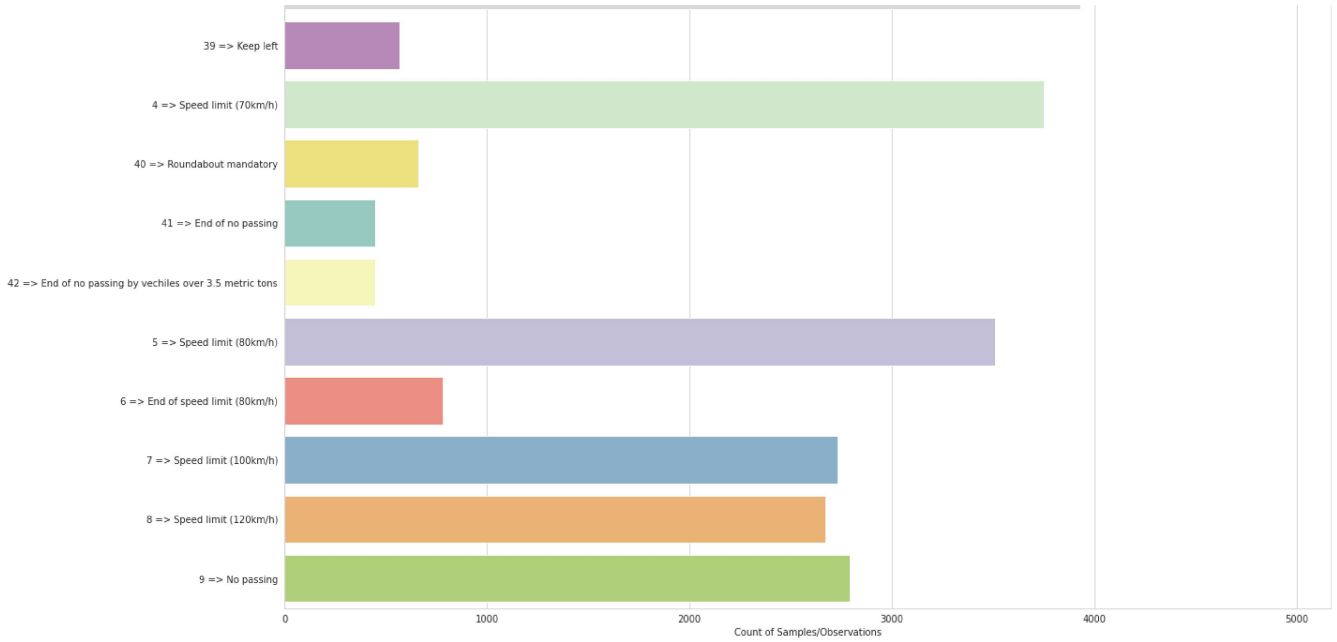
Reading Data of Class Labels

```
path = '/content/drive/MyDrive/Colab Notebooks/Deeplearning1/myData'
lab = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Deeplearning1/labels.csv')
```

Visualizing countplot of the classes

```
# Count PLot of the samples/observations w.r.t the classes
d = dict()
class_labels = dict()
for dirs in os.listdir(path):
    count = len(os.listdir(path +"/" +dirs))
    d[dirs+' => '+lab[lab.ClassId == int(dirs)].values[0][1]] = count
    class_labels[int(dirs)] = lab[lab.ClassId == int(dirs)].values[0][1]

plt.figure(figsize = (20, 50))
sns.barplot(y = list(d.keys()), x = list(d.values()), palette = 'Set3')
plt.ylabel('Label')
plt.xlabel('Count of Samples/Observations');
```


0 => Speed limit (20km/h)
1 => Speed limit (30km/h)
10 => No passing for vechiles over 3.5 metric tons
11 => Right-of-way at the next intersection
12 => Priority road
13 => Yield
14 => Stop
15 => No vechiles
16 => Vechiles over 3.5 metric tons prohibited
17 => No entry
18 => General caution
19 => Dangerous curve to the left
2 => Speed limit (50km/h)
20 => Dangerous curve to the right
21 => Double curve
22 => Bumpy road
23 => Slippery road
24 => Road narrows on the right
25 => Road work
26 => Traffic signals
27 => Pedestrians
28 => Children crossing
29 => Bicycles crossing
3 => Speed limit (60km/h)
30 => Beware of ice/snow
31 => Wild animals crossing
32 => End of all speed and passing limits
33 => Turn right ahead
34 => Turn left ahead
35 => Ahead only
36 => Go straight or right
37 => Go straight or left
38 => Keep right
Label


39 => Keep left
4 => Speed limit (70km/h)
40 => Roundabout mandatory
41 => End of no passing
42 => End of no passing by vechiles over 3.5 metric tons
5 => Speed limit (80km/h)
6 => End of speed limit (80km/h)
7 => Speed limit (100km/h)
8 => Speed limit (120km/h)
9 => No passing
0
1000
2000
3000
4000
5000
Count of Samples/Observations

Reading Image Data

```
# input image dimensions
img_rows, img_cols = 32, 32
# The images are RGB.
img_channels = 3
nb_classes = len(class_labels.keys())

datagen = ImageDataGenerator()
data = datagen.flow_from_directory(path,
                                   target_size=(32, 32),
                                   batch_size=73139,
                                   class_mode='categorical',
                                   shuffle=True )
```

```
Found 73139 images belonging to 43 classes.
```

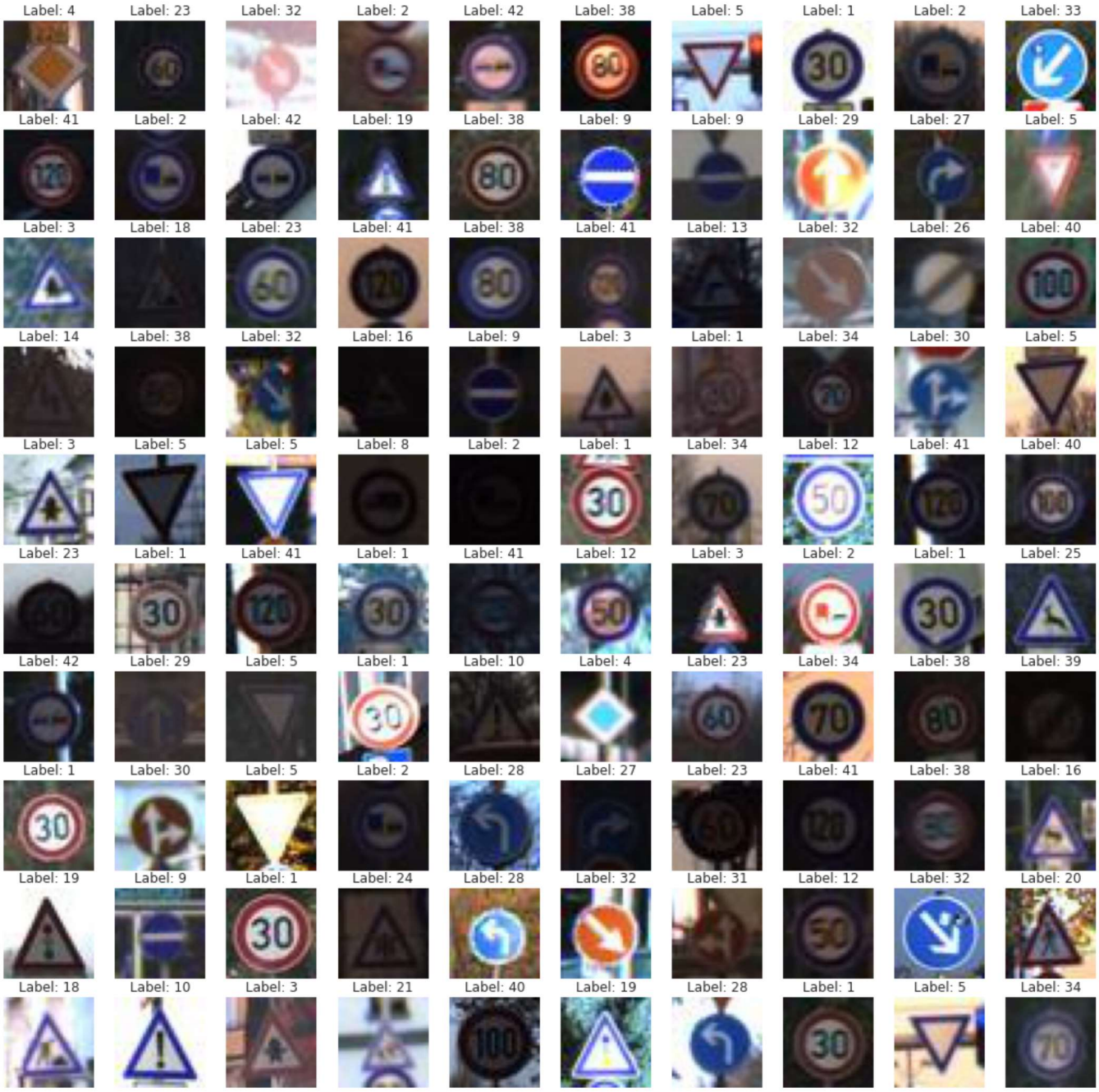
```
X , y = data.next()
```

```
# Labels are one hot encoded
print(f"Data Shape   :{X.shape}\nLabels shape :{y.shape}")
```

```
Data Shape   :(73139, 32, 32, 3)
Labels shape :(73139, 43)
```

Sample Images of Dataset

```
fig, axes = plt.subplots(10,10, figsize=(18,18))
for i,ax in enumerate(axes.flat):
    r = np.random.randint(X.shape[0])
    ax.imshow(X[r].astype('uint8'))
    ax.grid(False)
    ax.axis('off')
    ax.set_title('Label: '+str(np.argmax(y[r])))
```


Label: 4 Label: 23 Label: 32 Label: 2 Label: 42 Label: 38 Label: 5 Label: 1 Label: 2 Label: 33
Label: 41 Label: 2 Label: 42 Label: 19 Label: 38 Label: 9 Label: 9 Label: 29 Label: 27 Label: 5
Label: 3 Label: 18 Label: 23 Label: 41 Label: 38 Label: 41 Label: 13 Label: 32 Label: 26 Label: 40
Label: 14 Label: 38 Label: 32 Label: 16 Label: 9 Label: 3 Label: 1 Label: 34 Label: 30 Label: 5
Label: 3 Label: 5 Label: 5 Label: 8 Label: 2 Label: 1 Label: 34 Label: 12 Label: 41 Label: 40
Label: 23 Label: 1 Label: 41 Label: 1 Label: 41 Label: 12 Label: 3 Label: 2 Label: 1 Label: 25
Label: 42 Label: 29 Label: 5 Label: 1 Label: 10 Label: 4 Label: 23 Label: 34 Label: 38 Label: 39
Label: 1 Label: 30 Label: 5 Label: 2 Label: 28 Label: 27 Label: 23 Label: 41 Label: 38 Label: 16
Label: 19 Label: 9 Label: 1 Label: 24 Label: 28 Label: 32 Label: 31 Label: 12 Label: 32 Label: 20
Label: 18 Label: 10 Label: 3 Label: 21 Label: 40 Label: 19 Label: 28 Label: 1 Label: 5 Label: 34

Dividing data into **train** and **test** in the split percentage of 80:20

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=11)
```

```
print("Train Shape: {}\nTest Shape : {}".format(X_train.shape, X_test.shape))
```

```
Train Shape: (58511, 32, 32, 3)
Test Shape : (14628, 32, 32, 3)
```

Customising ResNet50 model

```
resnet = ResNet50(weights= None, include_top=False, input_shape= (img_rows,img_cols,img_chann
```

```
x = resnet.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.5)(x)
predictions = Dense(nb_classes, activation= 'softmax')(x)
model = Model(inputs = resnet.input, outputs = predictions)
```

```
model.load_weights('/content/drive/MyDrive/Colab Notebooks/Deeplearning1/TSC_model.h5')
#model
```

```
model.summary()
```

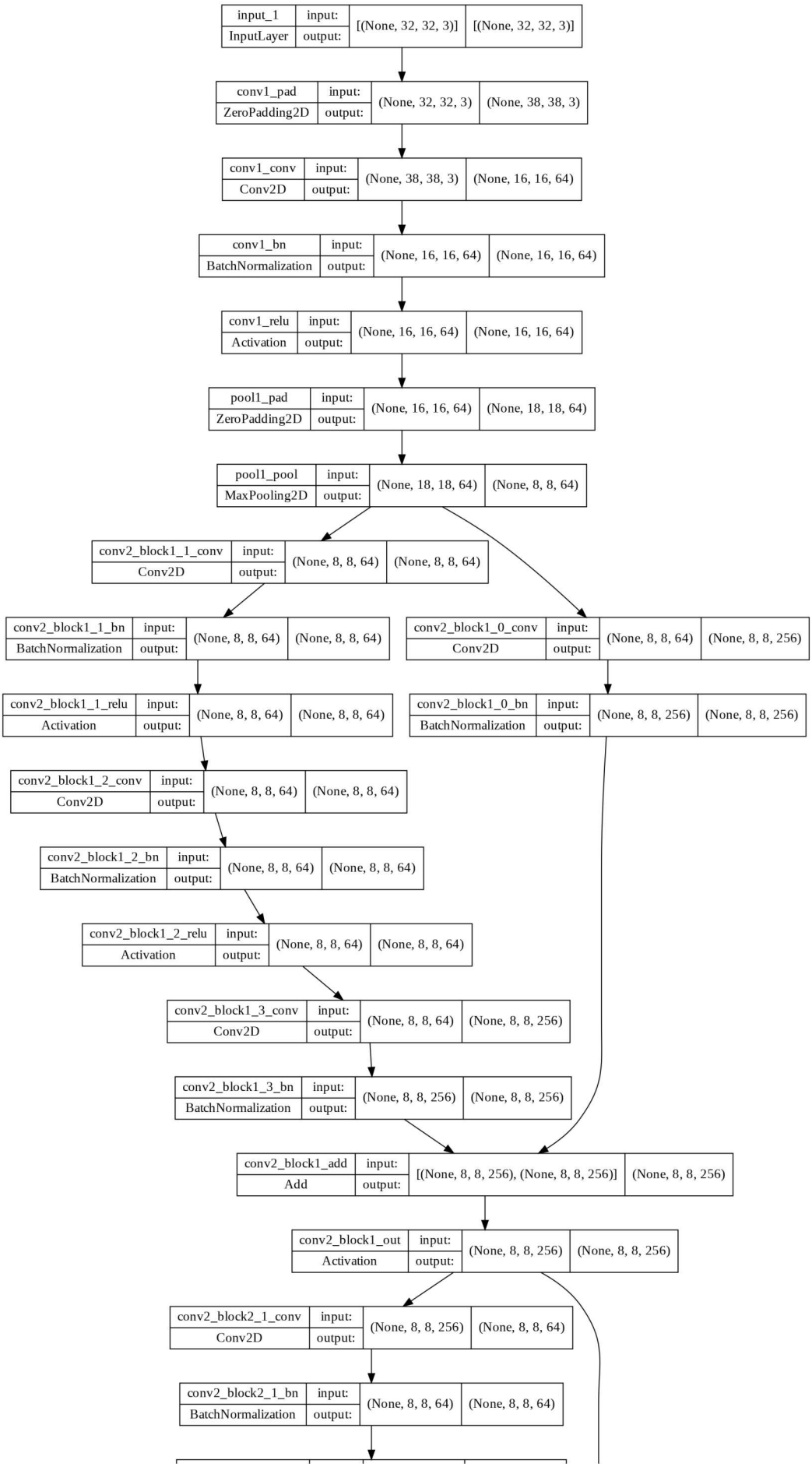
```
conv5_block2_2_conv (Conv2D)   (None, 1, 1, 512)    2359808     ['conv5_block2_1_rel
                                                                 u[0][0]']

conv5_block2_2_bn (BatchNormal  (None, 1, 1, 512)   2048        ['conv5_block2_2_con
ization)                                                         v[0][0]']

conv5_block2_2_relu (Activatio  (None, 1, 1, 512)   0           ['conv5_block2_2_bn[
n)                                                               0][0]']

conv5_block2_3_conv (Conv2D)   (None, 1, 1, 2048)   1050624     ['conv5_block2_2_rel
                                                                 u[0][0]']

conv5_block2_3_bn (BatchNormal  (None, 1, 1, 2048)  8192        ['conv5_block2_3_con
ization)                                                         v[0][0]']

conv5_block2_add (Add)         (None, 1, 1, 2048)   0           ['conv5_block1_out[0
                                                                 'conv5_block2_3_bn[

conv5_block2_out (Activation)  (None, 1, 1, 2048)   0           ['conv5_block2_add[0

conv5_block3_1_conv (Conv2D)   (None, 1, 1, 512)    1049088     ['conv5_block2_out[0

conv5_block3_1_bn (BatchNormal  (None, 1, 1, 512)   2048        ['conv5_block3_1_con
ization)                                                         v[0][0]']
```

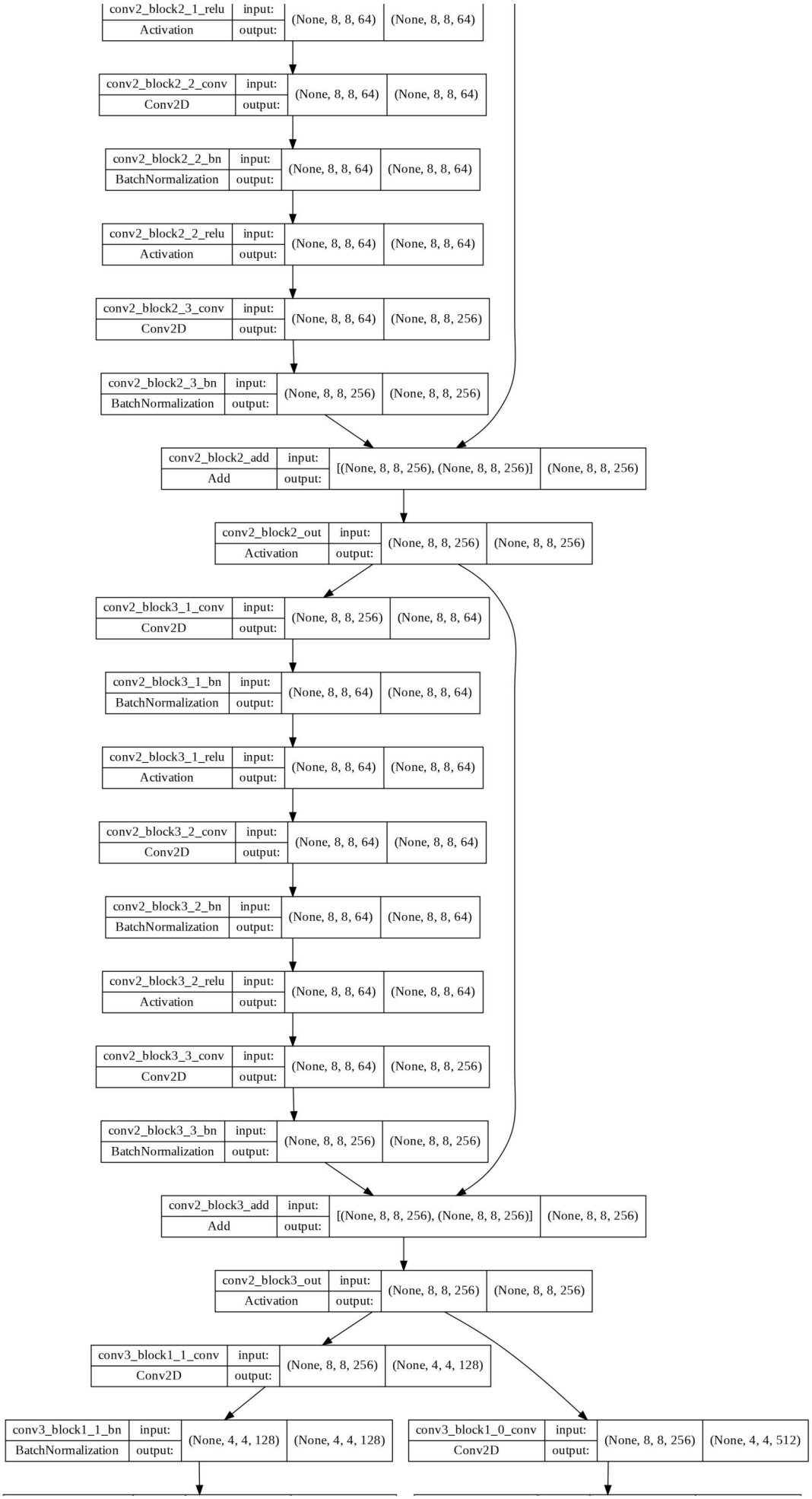
```
conv5_block3_1_relu (Activatio  (None, 1, 1, 512)   0           ['conv5_block3_1_bn[
n)

conv5_block3_2_conv (Conv2D)   (None, 1, 1, 512)    2359808     ['conv5_block3_1_rel

conv5_block3_2_bn (BatchNormal  (None, 1, 1, 512)   2048        ['conv5_block3_2_con
ization)

conv5_block3_2_relu (Activatio  (None, 1, 1, 512)   0           ['conv5_block3_2_bn[
n)

conv5_block3_3_conv (Conv2D)   (None, 1, 1, 2048)   1050624     ['conv5_block3_2_rel

conv5_block3_3_bn (BatchNormal  (None, 1, 1, 2048)  8192        ['conv5_block3_3_con
ization)

conv5_block3_add (Add)         (None, 1, 1, 2048)   0           ['conv5_block2_out[0
                                                                 'conv5_block3_3_bn[

conv5_block3_out (Activation)  (None, 1, 1, 2048)   0           ['conv5_block3_add[0

global_average_pooling2d (Glob  (None, 2048)        0           ['conv5_block3_out[0
alAveragePooling2D)

dropout (Dropout)              (None, 2048)         0           ['global_average_poo
                                                                ]

dense (Dense)                  (None, 43)           88107       ['dropout[0][0]']

==================================================================================
Total params: 23,675,819
Trainable params: 23,622,699
Non-trainable params: 53,120
__________________________________________________________________________________
```

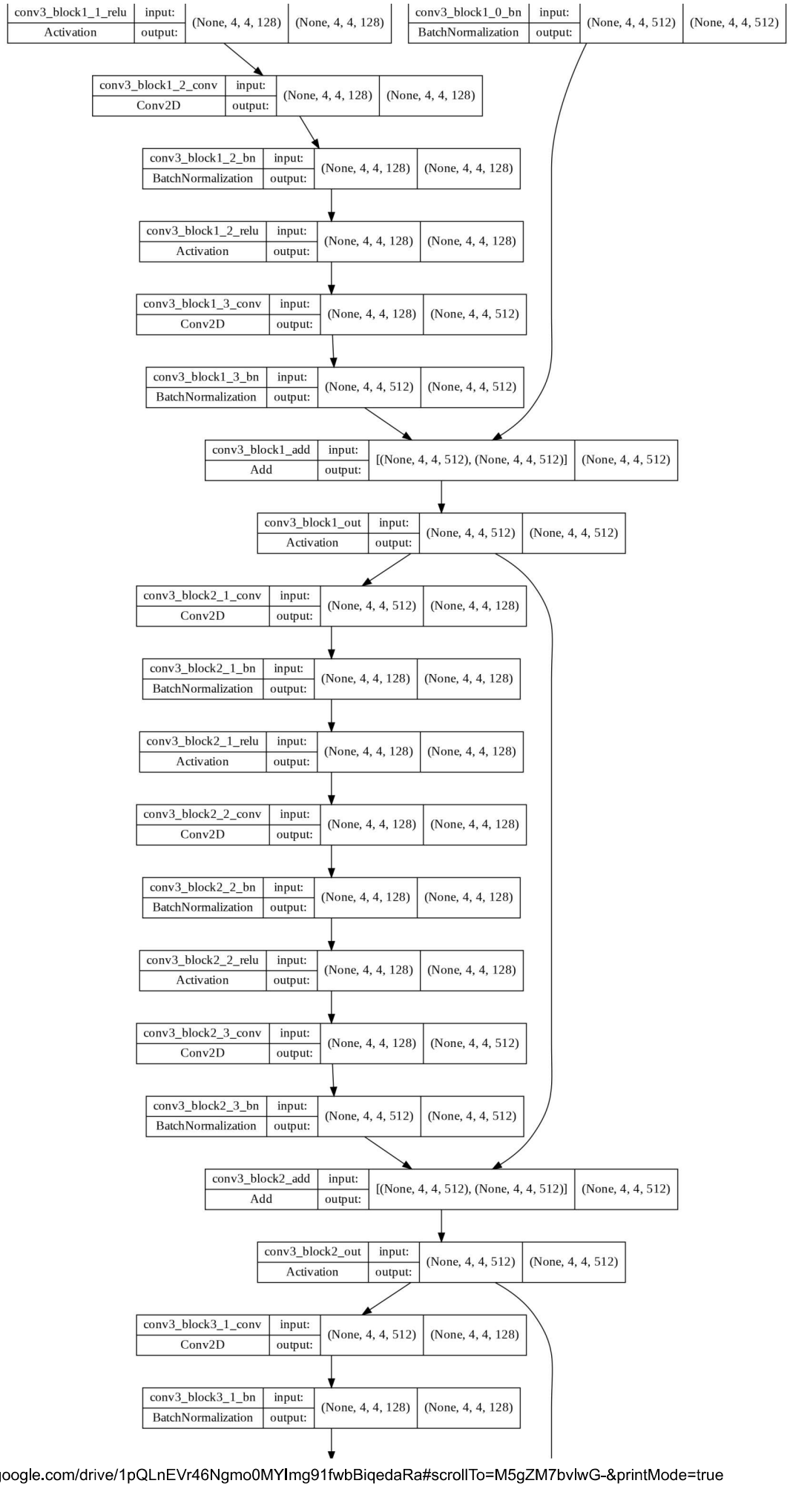
Visualising Model Architecture

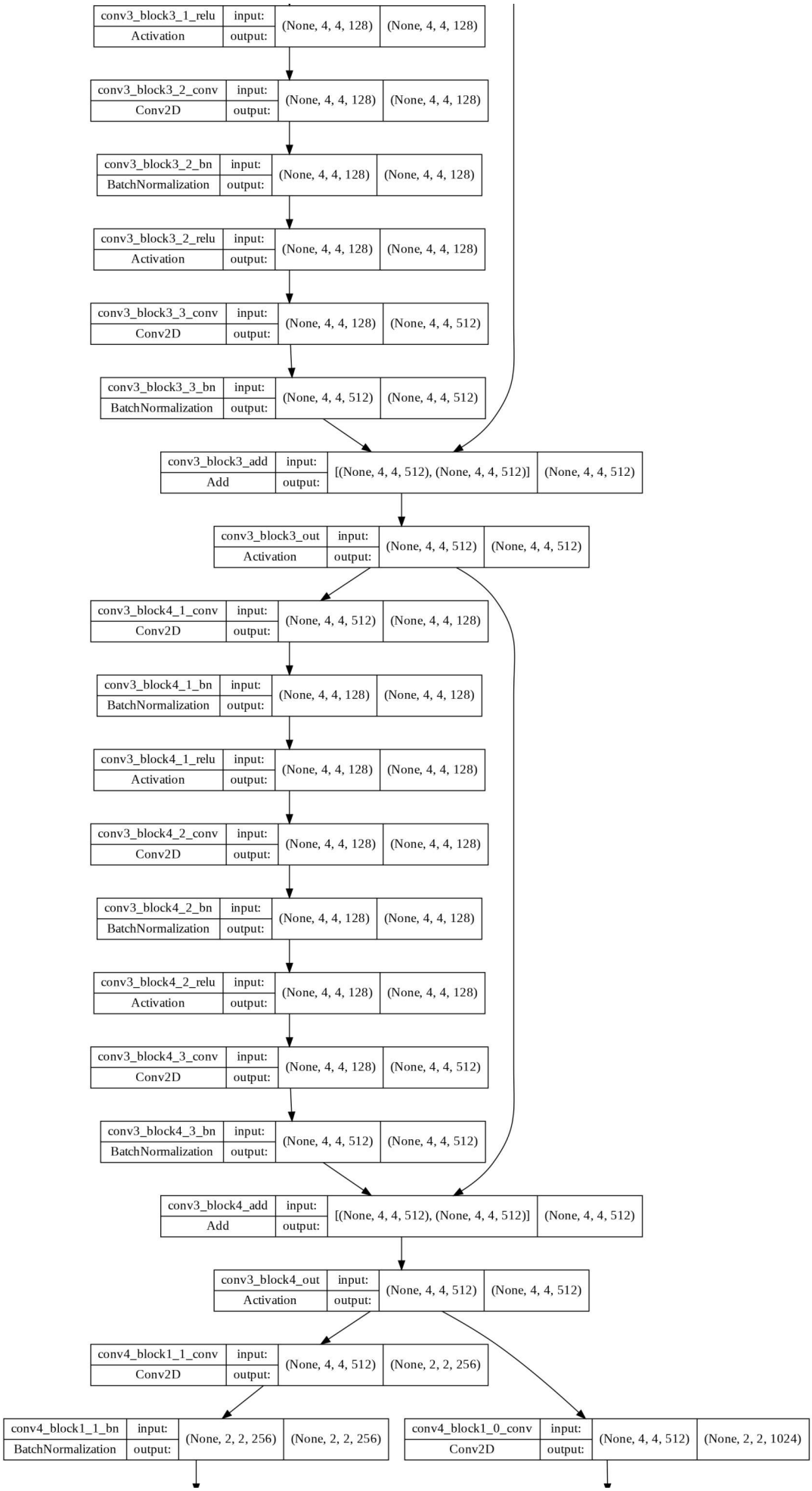
```
plot_model(model, show_layer_names=True, show_shapes =True, to_file='model.png', dpi=350)
```

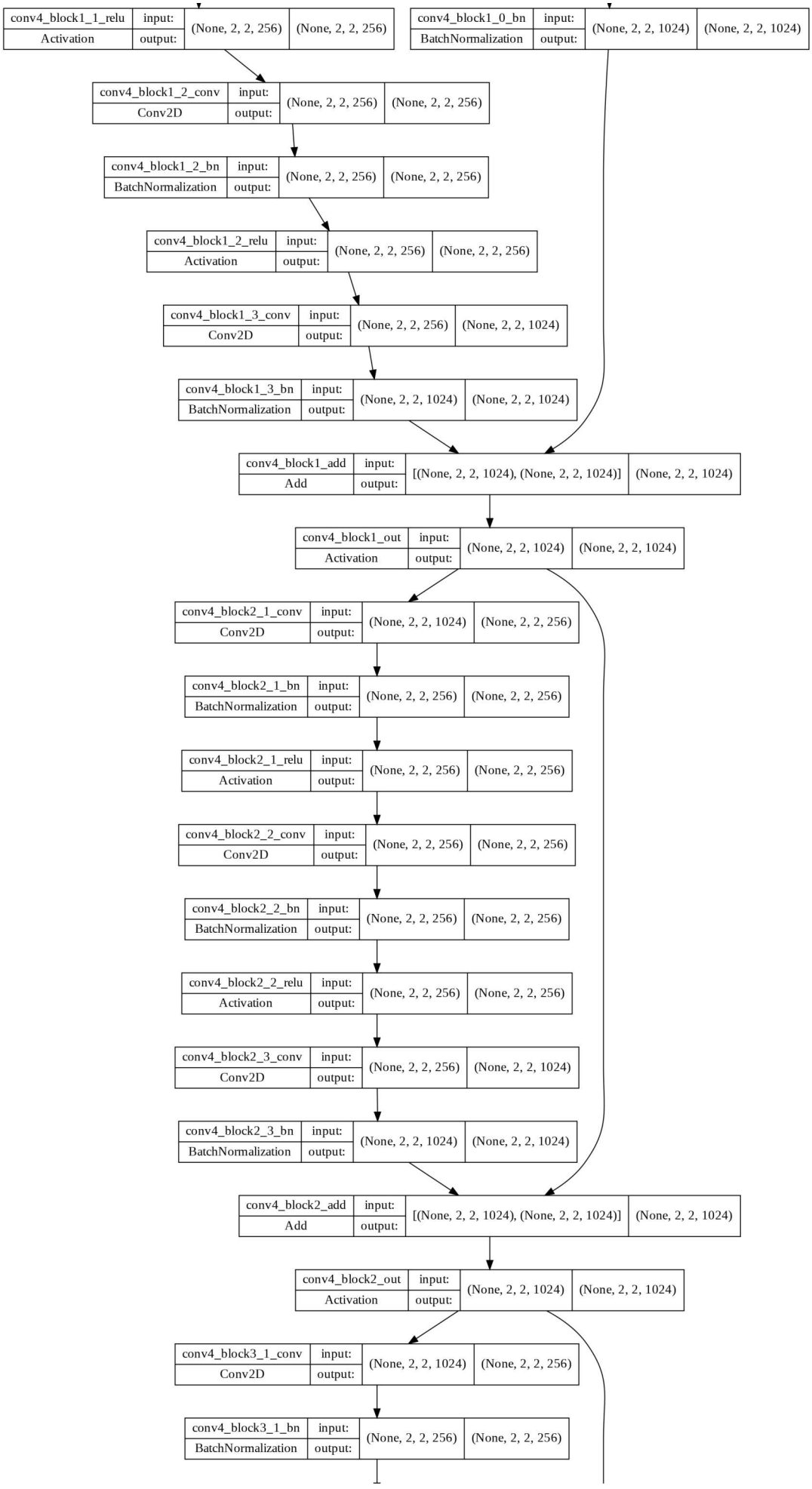
```
dot: graph is too large for cairo-renderer bitmaps. Scaling by 0.478672 to fit
```

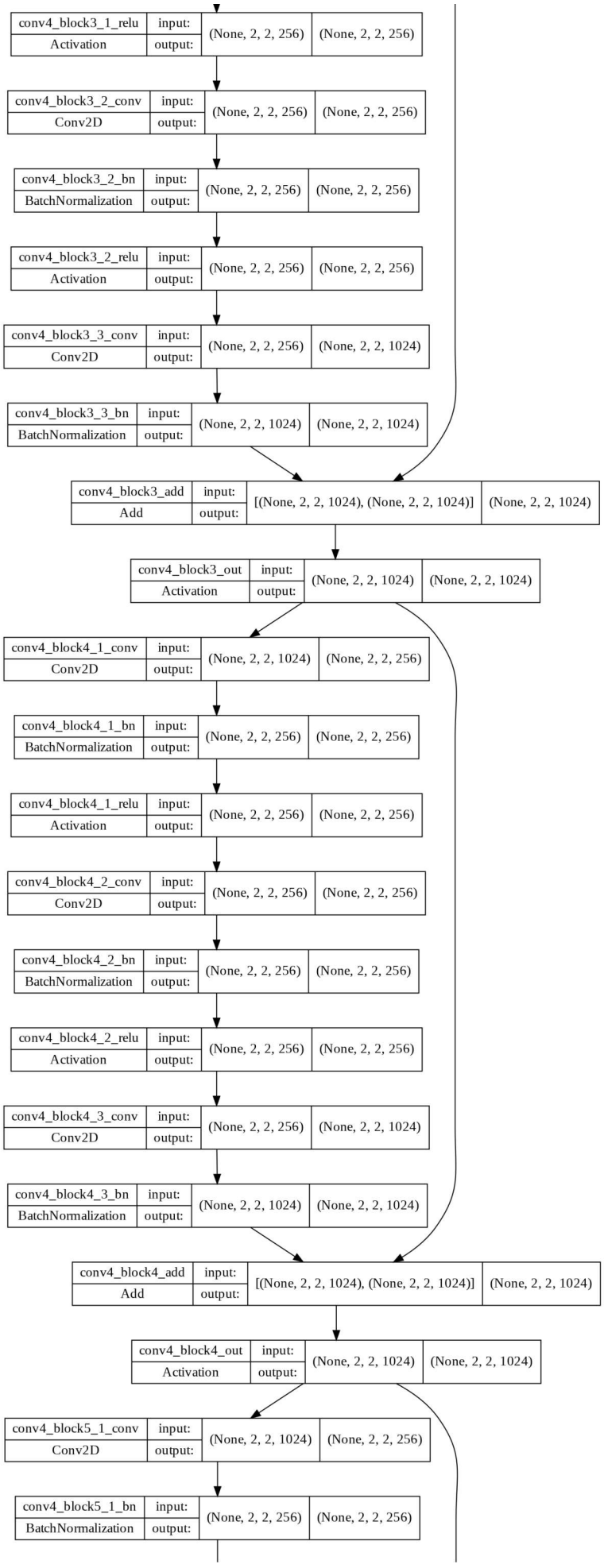


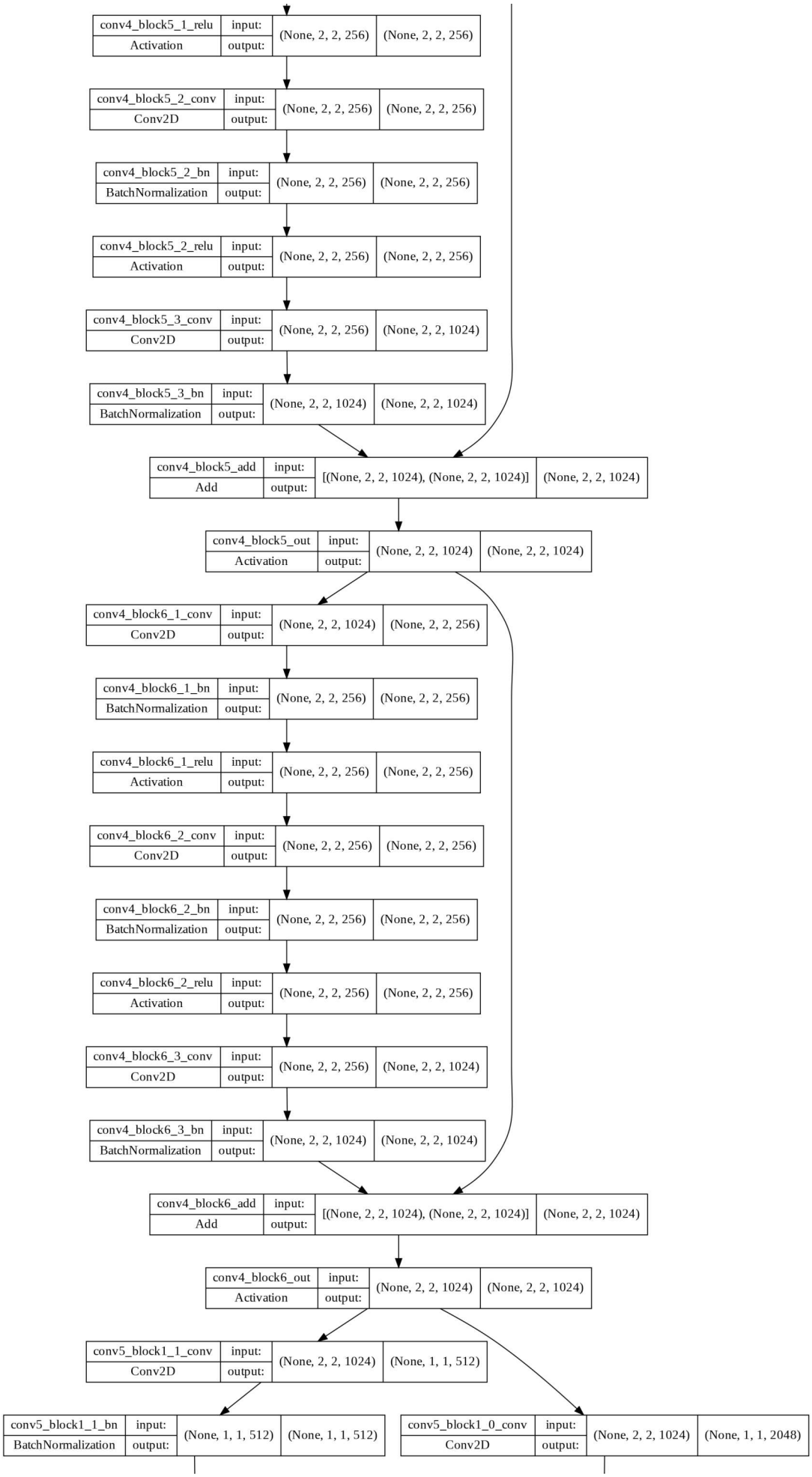

conv2_block2_1_relu | Activation | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block2_2_conv | Conv2D | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block2_2_bn | BatchNormalization | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block2_2_relu | Activation | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block2_3_conv | Conv2D | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 256)
conv2_block2_3_bn | BatchNormalization | input: | output: | (None, 8, 8, 256) | (None, 8, 8, 256)
conv2_block2_add | Add | input: | output: | [(None, 8, 8, 256), (None, 8, 8, 256)] | (None, 8, 8, 256)
conv2_block2_out | Activation | input: | output: | (None, 8, 8, 256) | (None, 8, 8, 256)
conv2_block3_1_conv | Conv2D | input: | output: | (None, 8, 8, 256) | (None, 8, 8, 64)
conv2_block3_1_bn | BatchNormalization | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block3_1_relu | Activation | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block3_2_conv | Conv2D | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block3_2_bn | BatchNormalization | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block3_2_relu | Activation | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 64)
conv2_block3_3_conv | Conv2D | input: | output: | (None, 8, 8, 64) | (None, 8, 8, 256)
conv2_block3_3_bn | BatchNormalization | input: | output: | (None, 8, 8, 256) | (None, 8, 8, 256)
conv2_block3_add | Add | input: | output: | [(None, 8, 8, 256), (None, 8, 8, 256)] | (None, 8, 8, 256)
conv2_block3_out | Activation | input: | output: | (None, 8, 8, 256) | (None, 8, 8, 256)
conv3_block1_1_conv | Conv2D | input: | output: | (None, 8, 8, 256) | (None, 4, 4, 128)
conv3_block1_1_bn | BatchNormalization | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block1_0_conv | Conv2D | input: | output: | (None, 8, 8, 256) | (None, 4, 4, 512)

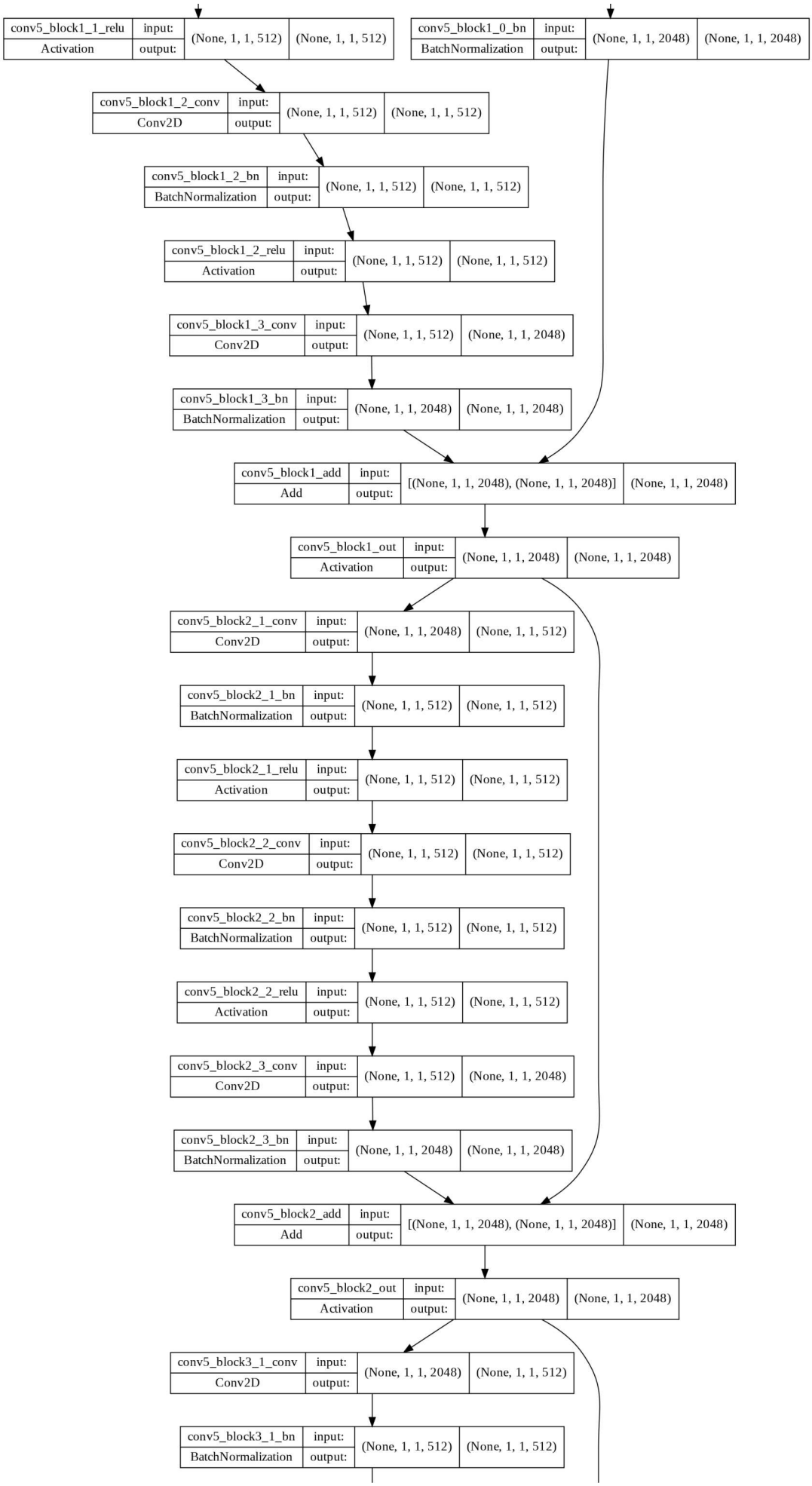

conv3_block1_1_relu | Activation | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block1_0_bn | BatchNormalization | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv3_block1_2_conv | Conv2D | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block1_2_bn | BatchNormalization | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block1_2_relu | Activation | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block1_3_conv | Conv2D | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 512)
conv3_block1_3_bn | BatchNormalization | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv3_block1_add | Add | input: | output: | [(None, 4, 4, 512), (None, 4, 4, 512)] | (None, 4, 4, 512)
conv3_block1_out | Activation | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv3_block2_1_conv | Conv2D | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 128)
conv3_block2_1_bn | BatchNormalization | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block2_1_relu | Activation | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block2_2_conv | Conv2D | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block2_2_bn | BatchNormalization | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block2_2_relu | Activation | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block2_3_conv | Conv2D | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 512)
conv3_block2_3_bn | BatchNormalization | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv3_block2_add | Add | input: | output: | [(None, 4, 4, 512), (None, 4, 4, 512)] | (None, 4, 4, 512)
conv3_block2_out | Activation | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv3_block3_1_conv | Conv2D | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 128)
conv3_block3_1_bn | BatchNormalization | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)

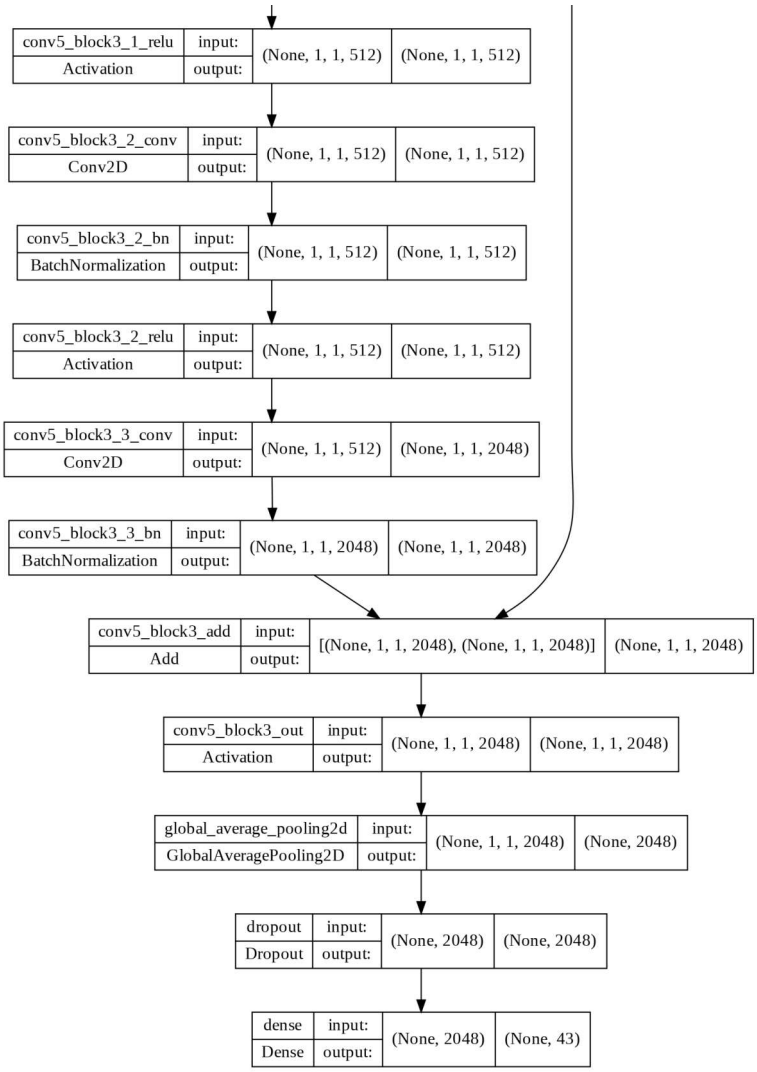

conv3_block3_1_relu | Activation | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block3_2_conv | Conv2D | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block3_2_bn | BatchNormalization | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block3_2_relu | Activation | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block3_3_conv | Conv2D | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 512)
conv3_block3_3_bn | BatchNormalization | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv3_block3_add | Add | input: | output: | [(None, 4, 4, 512), (None, 4, 4, 512)] | (None, 4, 4, 512)
conv3_block3_out | Activation | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv3_block4_1_conv | Conv2D | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 128)
conv3_block4_1_bn | BatchNormalization | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block4_1_relu | Activation | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block4_2_conv | Conv2D | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block4_2_bn | BatchNormalization | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block4_2_relu | Activation | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
conv3_block4_3_conv | Conv2D | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 512)
conv3_block4_3_bn | BatchNormalization | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv3_block4_add | Add | input: | output: | [(None, 4, 4, 512), (None, 4, 4, 512)] | (None, 4, 4, 512)
conv3_block4_out | Activation | input: | output: | (None, 4, 4, 512) | (None, 4, 4, 512)
conv4_block1_1_conv | Conv2D | input: | output: | (None, 4, 4, 512) | (None, 2, 2, 256)
conv4_block1_1_bn | BatchNormalization | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block1_0_conv | Conv2D | input: | output: | (None, 4, 4, 512) | (None, 2, 2, 1024)


conv4_block1_1_relu | Activation | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block1_0_bn | BatchNormalization | input: output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block1_2_conv | Conv2D | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block1_2_bn | BatchNormalization | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block1_2_relu | Activation | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block1_3_conv | Conv2D | input: output: | (None, 2, 2, 256) | (None, 2, 2, 1024)
conv4_block1_3_bn | BatchNormalization | input: output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block1_add | Add | input: output: | [(None, 2, 2, 1024), (None, 2, 2, 1024)] | (None, 2, 2, 1024)
conv4_block1_out | Activation | input: output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block2_1_conv | Conv2D | input: output: | (None, 2, 2, 1024) | (None, 2, 2, 256)
conv4_block2_1_bn | BatchNormalization | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block2_1_relu | Activation | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block2_2_conv | Conv2D | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block2_2_bn | BatchNormalization | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block2_2_relu | Activation | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block2_3_conv | Conv2D | input: output: | (None, 2, 2, 256) | (None, 2, 2, 1024)
conv4_block2_3_bn | BatchNormalization | input: output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block2_add | Add | input: output: | [(None, 2, 2, 1024), (None, 2, 2, 1024)] | (None, 2, 2, 1024)
conv4_block2_out | Activation | input: output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block3_1_conv | Conv2D | input: output: | (None, 2, 2, 1024) | (None, 2, 2, 256)
conv4_block3_1_bn | BatchNormalization | input: output: | (None, 2, 2, 256) | (None, 2, 2, 256)


conv4_block3_1_relu | Activation | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block3_2_conv | Conv2D | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block3_2_bn | BatchNormalization | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block3_2_relu | Activation | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block3_3_conv | Conv2D | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 1024)
conv4_block3_3_bn | BatchNormalization | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block3_add | Add | input: | output: | [(None, 2, 2, 1024), (None, 2, 2, 1024)] | (None, 2, 2, 1024)
conv4_block3_out | Activation | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block4_1_conv | Conv2D | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 256)
conv4_block4_1_bn | BatchNormalization | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block4_1_relu | Activation | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block4_2_conv | Conv2D | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block4_2_bn | BatchNormalization | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block4_2_relu | Activation | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block4_3_conv | Conv2D | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 1024)
conv4_block4_3_bn | BatchNormalization | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block4_add | Add | input: | output: | [(None, 2, 2, 1024), (None, 2, 2, 1024)] | (None, 2, 2, 1024)
conv4_block4_out | Activation | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block5_1_conv | Conv2D | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 256)
conv4_block5_1_bn | BatchNormalization | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)


conv4_block5_1_relu | Activation | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block5_2_conv | Conv2D | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block5_2_bn | BatchNormalization | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block5_2_relu | Activation | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block5_3_conv | Conv2D | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 1024)
conv4_block5_3_bn | BatchNormalization | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block5_add | Add | input: | output: | [(None, 2, 2, 1024), (None, 2, 2, 1024)] | (None, 2, 2, 1024)
conv4_block5_out | Activation | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block6_1_conv | Conv2D | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 256)
conv4_block6_1_bn | BatchNormalization | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block6_1_relu | Activation | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block6_2_conv | Conv2D | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block6_2_bn | BatchNormalization | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block6_2_relu | Activation | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 256)
conv4_block6_3_conv | Conv2D | input: | output: | (None, 2, 2, 256) | (None, 2, 2, 1024)
conv4_block6_3_bn | BatchNormalization | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv4_block6_add | Add | input: | output: | [(None, 2, 2, 1024), (None, 2, 2, 1024)] | (None, 2, 2, 1024)
conv4_block6_out | Activation | input: | output: | (None, 2, 2, 1024) | (None, 2, 2, 1024)
conv5_block1_1_conv | Conv2D | input: | output: | (None, 2, 2, 1024) | (None, 1, 1, 512)
conv5_block1_1_bn | BatchNormalization | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block1_0_conv | Conv2D | input: | output: | (None, 2, 2, 1024) | (None, 1, 1, 2048)


conv5_block1_1_relu | Activation | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block1_0_bn | BatchNormalization | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 2048)
conv5_block1_2_conv | Conv2D | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block1_2_bn | BatchNormalization | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block1_2_relu | Activation | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block1_3_conv | Conv2D | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 2048)
conv5_block1_3_bn | BatchNormalization | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 2048)
conv5_block1_add | Add | input: | output: | [(None, 1, 1, 2048), (None, 1, 1, 2048)] | (None, 1, 1, 2048)
conv5_block1_out | Activation | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 2048)
conv5_block2_1_conv | Conv2D | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 512)
conv5_block2_1_bn | BatchNormalization | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block2_1_relu | Activation | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block2_2_conv | Conv2D | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block2_2_bn | BatchNormalization | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block2_2_relu | Activation | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block2_3_conv | Conv2D | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 2048)
conv5_block2_3_bn | BatchNormalization | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 2048)
conv5_block2_add | Add | input: | output: | [(None, 1, 1, 2048), (None, 1, 1, 2048)] | (None, 1, 1, 2048)
conv5_block2_out | Activation | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 2048)
conv5_block3_1_conv | Conv2D | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 512)
conv5_block3_1_bn | BatchNormalization | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)


conv5_block3_1_relu | Activation | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block3_2_conv | Conv2D | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block3_2_bn | BatchNormalization | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block3_2_relu | Activation | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 512)
conv5_block3_3_conv | Conv2D | input: | output: | (None, 1, 1, 512) | (None, 1, 1, 2048)
conv5_block3_3_bn | BatchNormalization | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 2048)
conv5_block3_add | Add | input: | output: | [(None, 1, 1, 2048), (None, 1, 1, 2048)] | (None, 1, 1, 2048)
conv5_block3_out | Activation | input: | output: | (None, 1, 1, 2048) | (None, 1, 1, 2048)
global_average_pooling2d | GlobalAveragePooling2D | input: | output: | (None, 1, 1, 2048) | (None, 2048)
dropout | Dropout | input: | output: | (None, 2048) | (None, 2048)
dense | Dense | input: | output: | (None, 2048) | (None, 43)