Project 4: Face recognition

As usual, we will mount our Google Driver first and change the current folder to be the one you are working with.

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remou
```

```
import os
os.chdir('/content/drive/MyDrive/DL/Homework11')
```

1. Install pacakges

Google Colab has installed most of the frequently used packages. For the packages that's not installed, we can use `!pip install ...` to do it.

This project will use the model called 'keras-vggface' that has been trained and shared through GitHub.com. The VGGFace is a face recoginition NN model trained by Visual Geometry Group (VGG) at the University of Oxford. The model was trained using 3.31 million images of 8631 subjects. We will test how this model behaves in different group of people.

Because the 'keras-VGGFace' is only tested on TensorFlow version 1.14, we will need to downgrade both the TensorFlow and the H5PY to accommadate the package. After running the cell below, please restart the runtime (ctrl+M) to make the installed version effective.

```
%tensorflow_version 1.x
```

```
TensorFlow 1.x selected.
```

```
import tensorflow as tf
import h5py
print('H5PY version:', h5py.__version__)

#print(keras.__version__)
print('TensorFlow version:', tf.__version__)
```

```
H5PY version: 2.10.0
TensorFlow version: 1.15.2
```

```
!pip uninstall h5py
!pip install h5py==2.10.0
```

```
Found existing installation: h5py 2.10.0
Uninstalling h5py-2.10.0:
  Would remove:
    /usr/local/lib/python3.7/dist-packages/h5py-2.10.0.dist-info/*
    /usr/local/lib/python3.7/dist-packages/h5py/*
Proceed (y/n)? y
  Successfully uninstalled h5py-2.10.0
Collecting h5py==2.10.0
  Using cached h5py-2.10.0-cp37-cp37m-manylinux1_x86_64.whl (2.9 MB)
Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.7/dist-packages (from h5py==2.10.0) (1.21.6)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from h5py==2.10.0) (1.15.0)
Installing collected packages: h5py
Successfully installed h5py-2.10.0
WARNING: The following packages were previously imported in this runtime:
  [h5py]
You must restart the runtime in order to use newly installed versions.
```

RESTART RUNTIME

```
!pip install git+https://github.com/rcmalli/keras-vggface.git
!pip install keras_applications
!pip install keras_preprocessing
!pip install mtcnn
```

```
Collecting git+https://github.com/rcmalli/keras-vggface.git
```

```
  Cloning https://github.com/rcmalli/keras-vggface.git to /tmp/pip-req-build-2606oz00
  Running command git clone -q https://github.com/rcmalli/keras-vggface.git /tmp/pip-req-build-2606oz00
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-packages (from keras-vggface==0.6) (1.21.6
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.7/dist-packages (from keras-vggface==0.6) (1.4.1)
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages (from keras-vggface==0.6) (2.10.0)
Requirement already satisfied: pillow in /usr/local/lib/python3.7/dist-packages (from keras-vggface==0.6) (7.1.2)
Requirement already satisfied: keras in /tensorflow-1.15.2/python3.7 (from keras-vggface==0.6) (2.3.1)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from keras-vggface==0.6) (1.15.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from keras-vggface==0.6) (3.13)
Requirement already satisfied: keras-applications>=1.0.6 in /tensorflow-1.15.2/python3.7 (from keras->keras-vggface==0.
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.7/dist-packages (from keras->keras-
Requirement already satisfied: keras_applications in /tensorflow-1.15.2/python3.7 (1.0.8)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-packages (from keras_applications) (1.21.6
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages (from keras_applications) (2.10.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from h5py->keras_applications) (1.15.0)
Requirement already satisfied: keras_preprocessing in /usr/local/lib/python3.7/dist-packages (1.1.2)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-packages (from keras_preprocessing) (1.21.
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from keras_preprocessing) (1.15.0)
Collecting mtcnn
  Downloading mtcnn-0.1.1-py3-none-any.whl (2.3 MB)
     |████████████████████████████████| 2.3 MB 5.4 MB/s
Requirement already satisfied: opencv-python>=4.1.0 in /usr/local/lib/python3.7/dist-packages (from mtcnn) (4.1.2.30)
Requirement already satisfied: keras>=2.0.0 in /tensorflow-1.15.2/python3.7 (from mtcnn) (2.3.1)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (1.21.
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (3.13)
Requirement already satisfied: keras-applications>=1.0.6 in /tensorflow-1.15.2/python3.7 (from keras>=2.0.0->mtcnn) (1.
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (1.4.1)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0-
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (1.15.0)
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (2.10.0)
Installing collected packages: mtcnn
Successfully installed mtcnn-0.1.1
```

We will import the packages here. Please make sure that your package versions are as follows.

H5PY version: 2.10.0

TensorFlow version: 1.x

keras_VGGFace version: 0.6

MTCNN version: 0.1.0

```
import keras
import numpy as np


# check version of keras_vggface
import keras_vggface
# print version
print('keras_VGGFace version:', keras_vggface.__version__)

# confirm mtcnn was installed correctly
import mtcnn
# print version
print('MTCNN version:', mtcnn.__version__)

from matplotlib import pyplot
import matplotlib.image as mpimg
from PIL import Image
from numpy import asarray
from mtcnn.mtcnn import MTCNN
from scipy.spatial.distance import cosine
from keras_vggface.vggface import VGGFace
from keras_vggface.utils import preprocess_input
from keras_vggface.utils import decode_predictions
```

```
Using TensorFlow backend.
keras_VGGFace version: 0.6
MTCNN version: 0.1.0
```

2. Detect faces

Before we can perform face recognition, we need to detect faces.

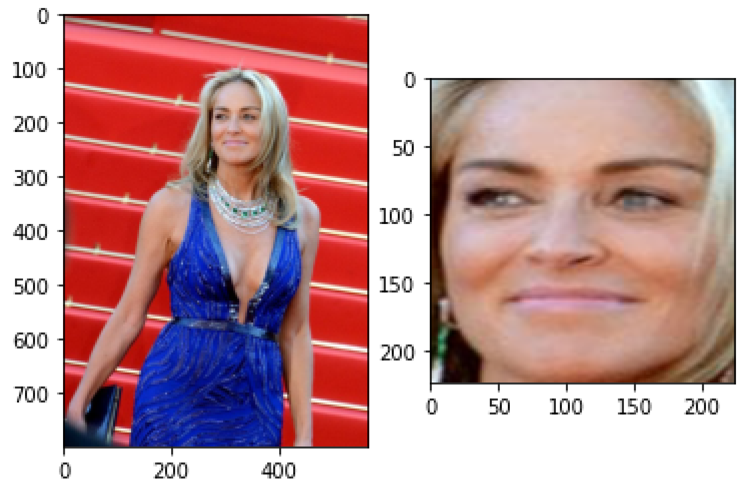
Face detection is the process of automatically locating faces in a photograph and localizing them by drawing a bounding box around their extent.

In this tutorial, we will also use the Multi-Task Cascaded Convolutional Neural Network, or MTCNN, for face detection, e.g. finding and extracting faces from photos. This is a state-of-the-art deep learning model for face detection, described in the 2016 paper titled

```
# extract a single face from a given photograph
def extract_face(filename, required_size=(224, 224)):
    # load image from file
    pixels = pyplot.imread(filename)
    # create the detector, using default weights
    detector = MTCNN()
    # detect faces in the image
    results = detector.detect_faces(pixels)
    # extract the bounding box from the first face
    x1, y1, width, height = results[0]['box']
    x2, y2 = x1 + width, y1 + height
    # extract the face
    face = pixels[y1:y2, x1:x2]
    # resize pixels to the model size
    image = Image.fromarray(face)
    image = image.resize(required_size)
    face_array = asarray(image)
    return face_array

## start your code here
image_path='/content/drive/MyDrive/DL/Homework11/face verification/images/sharon_stone1.jpg' # you can change the path to the
## end your code here

orig_img = mpimg.imread(image_path)

# load the photo and extract the face
pixels = extract_face(image_path)

fig = pyplot.figure()
# plot the original image
ax = fig.add_subplot(1, 2, 1)
imgplot = pyplot.imshow(orig_img)
```

```
# plot the extracted face
ax = fig.add_subplot(1, 2, 2)
imgplot = pyplot.imshow(pixels)

# show the plot
pyplot.show()
```

```
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/tensorflow_core/python/ops/resource_variable_ops.py:1630: calling
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
```



3. Face identification

A VGGFace model can be created using the VGGFace() constructor and specifying the type of model to create via the 'model' argument

`model = VGGFace(model='...')`

The keras-vggface library provides three pre-trained VGGModels, a VGGFace1 model via model='vgg16' (the default), and two VGGFace2 models 'resnet50' and 'senet50'.

The example below creates a 'resnet50' VGGFace2 model.

The first time that a model is created, the library will download the model weights and save them in the ./keras/models/vggface/ directory in your home directory. The size of the weights for the resnet50 model is about 158 megabytes, so the download may take a few minutes depending on the speed of your internet connection.

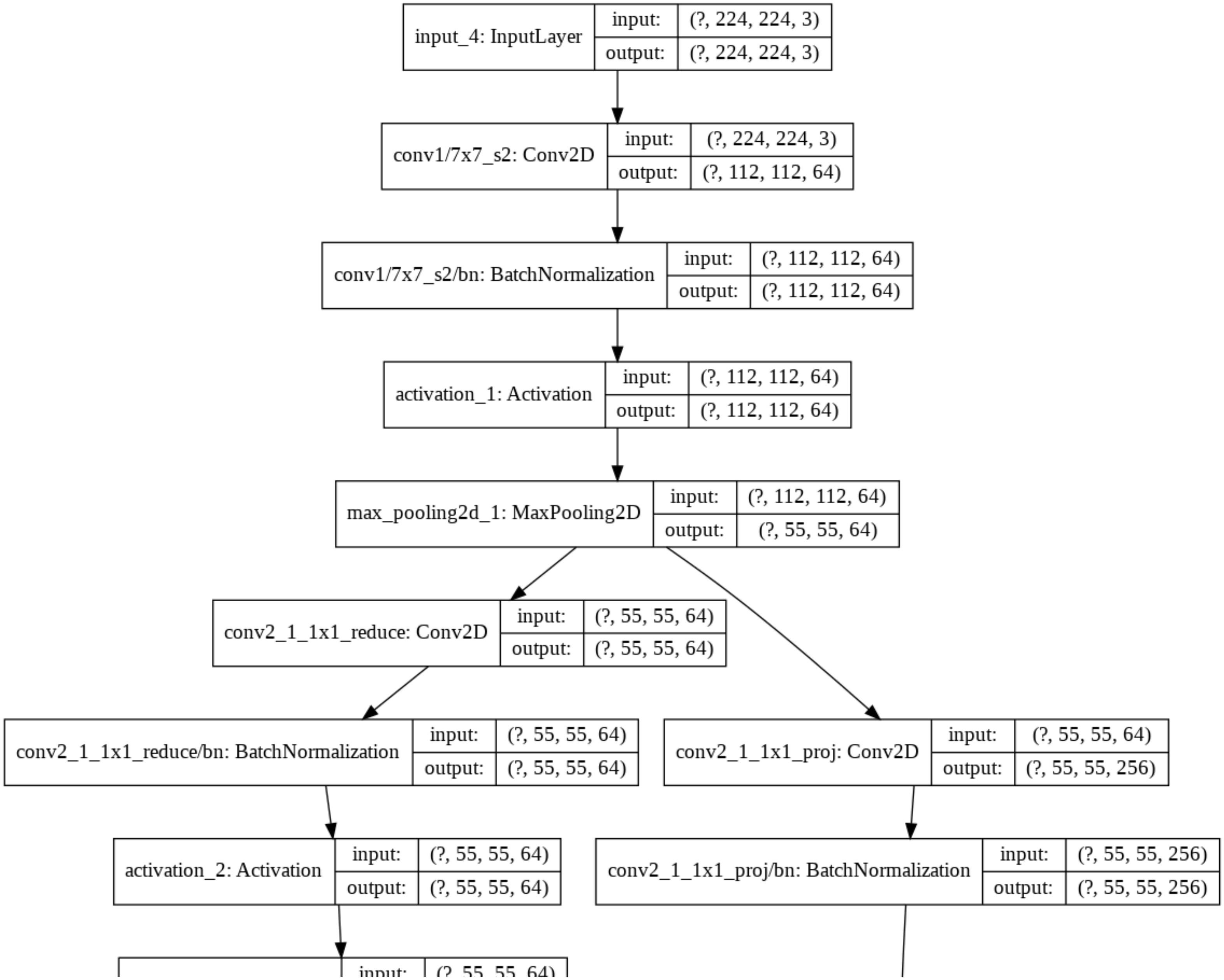
If printing out the NN structure, we can observe the apparant 'resnet' structure.

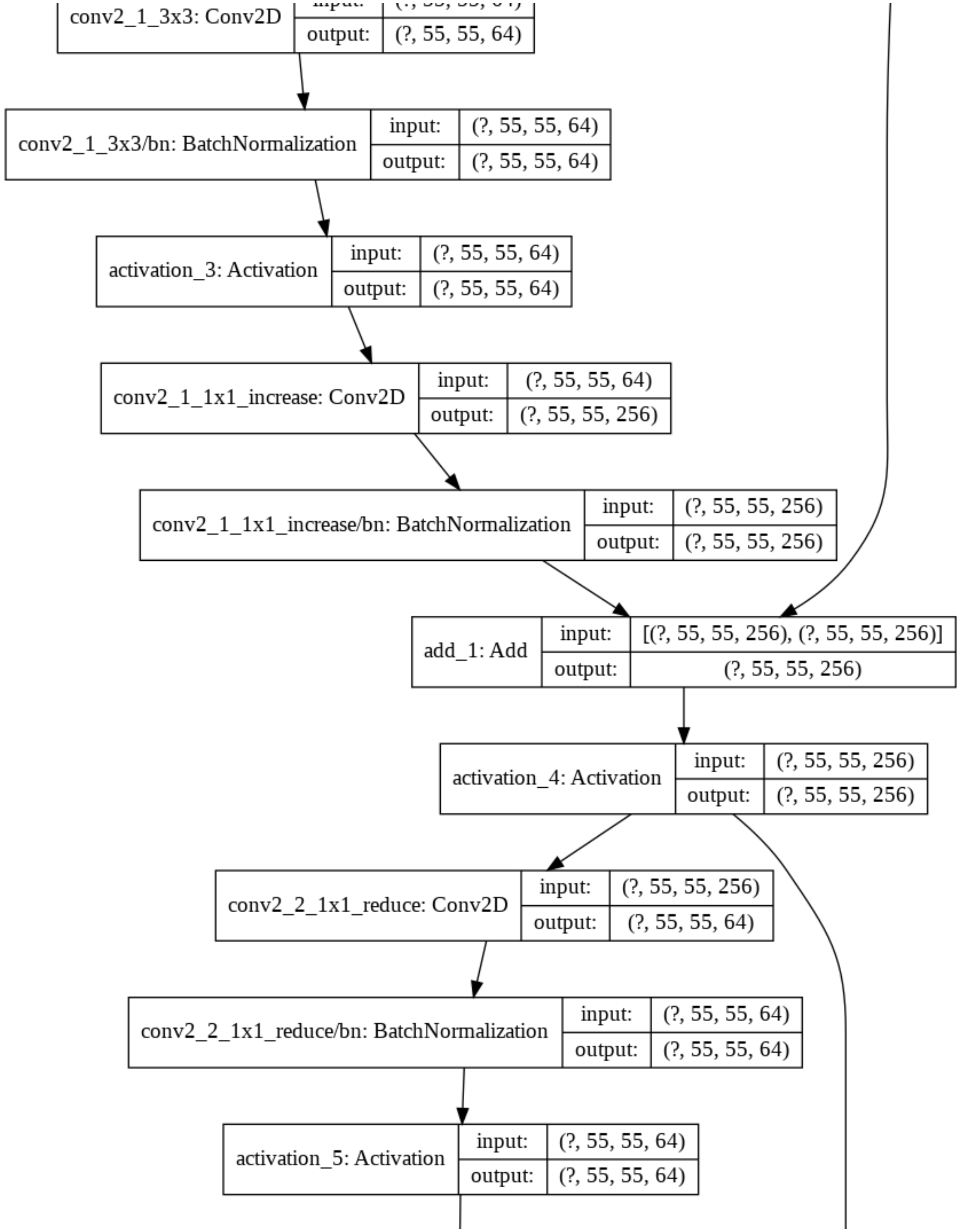
Exercise: Can you find the flatten layer and the dense layer(s) on top of the conv layers? For every image, how many elements are there in the flatten layer? How many elements are there in the output layer?

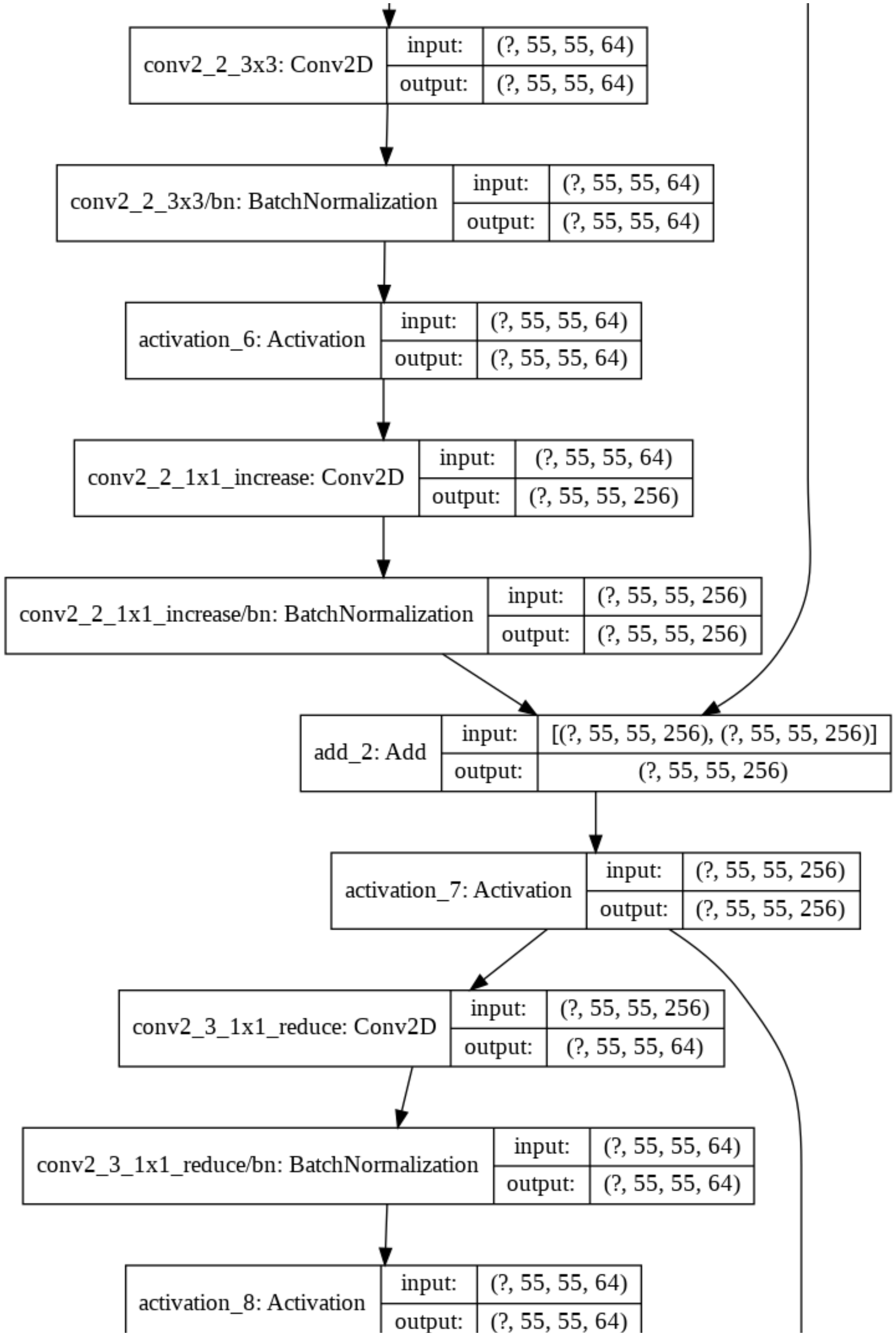
The output of the flatten layer is the features extracted from the face.

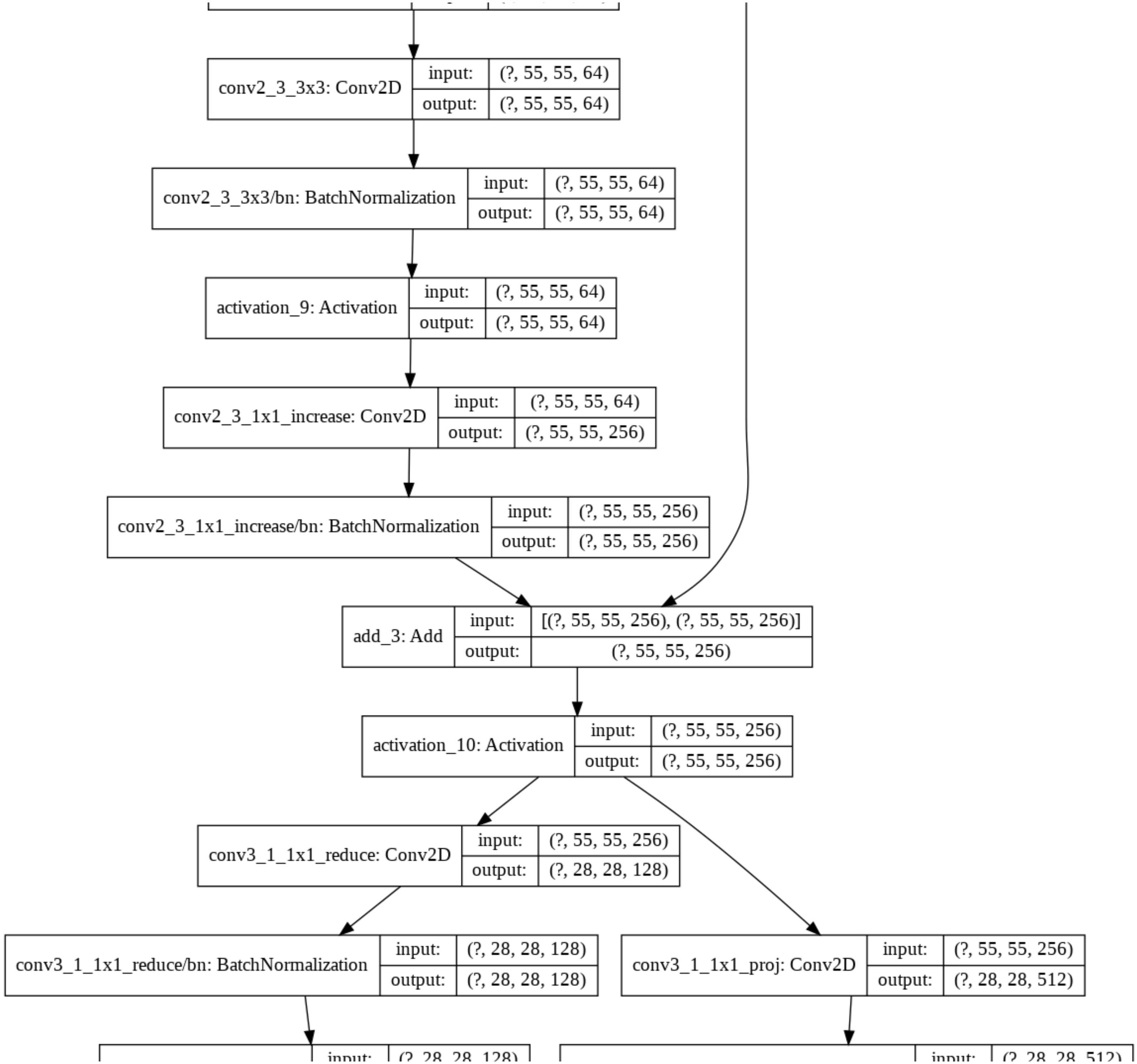
```
# create a vggface model
model = VGGFace(model='resnet50')
tf.keras.utils.plot_model(model, show_shapes=True)
```

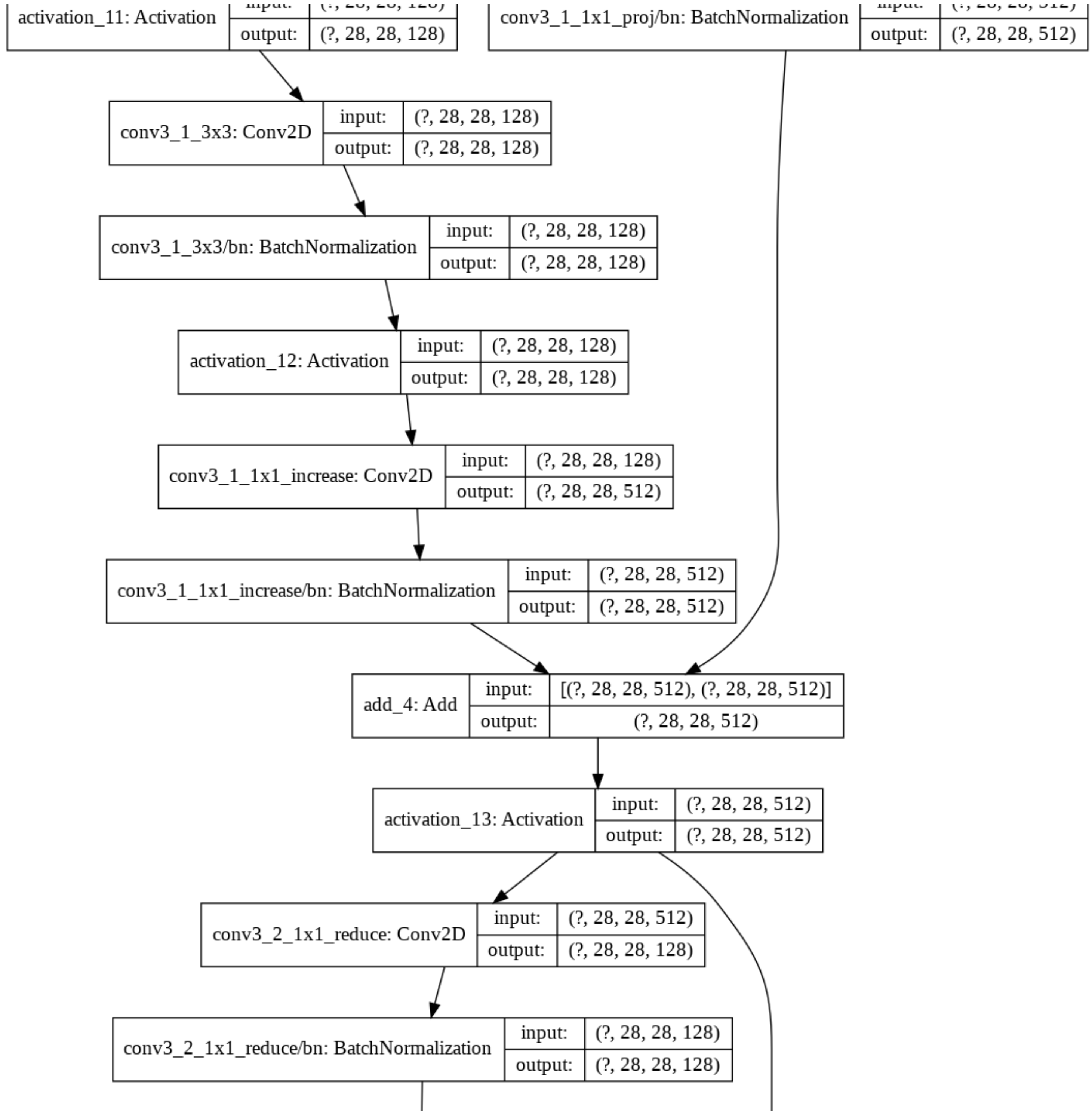
```
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:4070: The name tf.nn.max_pool

WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:4074: The name tf.nn.avg_pool

Downloading data from https://github.com/rcmalli/keras-vggface/releases/download/v2.0/rcmalli_vggface_tf_resnet50.h5
165445632/165439116 [==============================] - 2s 0us/step
```

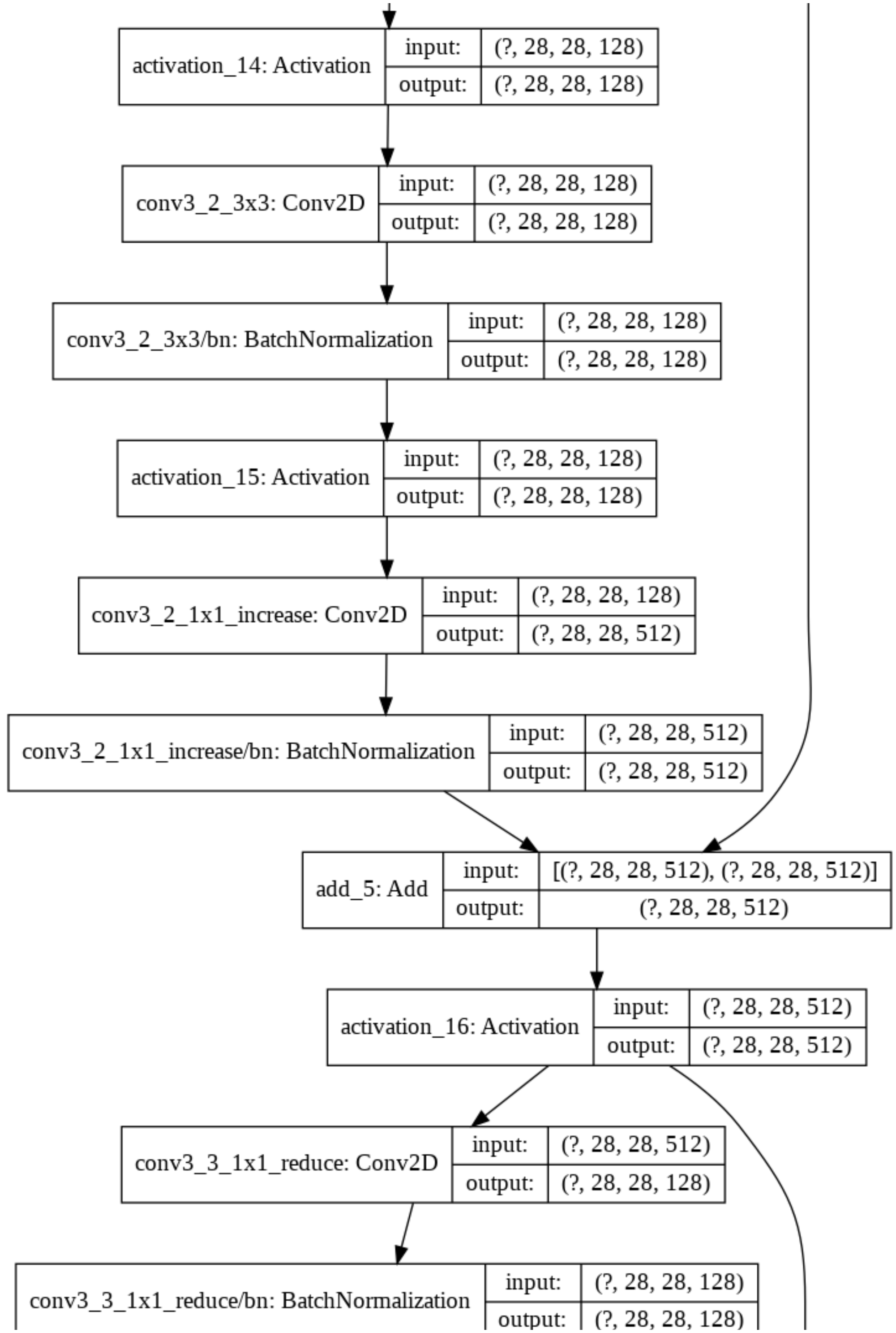


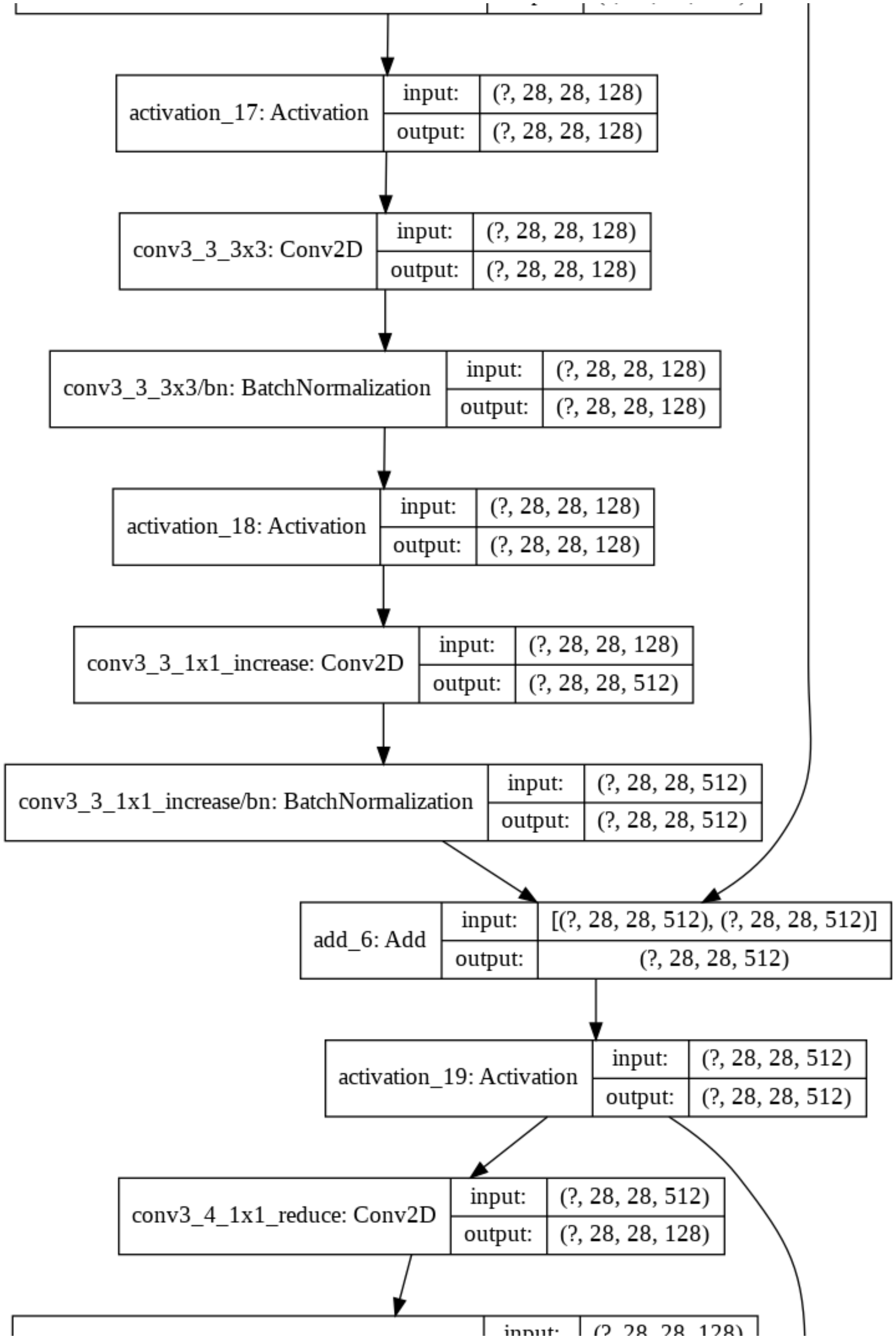

conv2_1_3x3: Conv2D
output: (?, 55, 55, 64)
conv2_1_3x3/bn: BatchNormalization
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
activation_3: Activation
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
conv2_1_1x1_increase: Conv2D
input: (?, 55, 55, 64)
output: (?, 55, 55, 256)
conv2_1_1x1_increase/bn: BatchNormalization
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
add_1: Add
input: [(?, 55, 55, 256), (?, 55, 55, 256)]
output: (?, 55, 55, 256)
activation_4: Activation
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
conv2_2_1x1_reduce: Conv2D
input: (?, 55, 55, 256)
output: (?, 55, 55, 64)
conv2_2_1x1_reduce/bn: BatchNormalization
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
activation_5: Activation
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)


conv2_2_3x3: Conv2D
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
conv2_2_3x3/bn: BatchNormalization
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
activation_6: Activation
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
conv2_2_1x1_increase: Conv2D
input: (?, 55, 55, 64)
output: (?, 55, 55, 256)
conv2_2_1x1_increase/bn: BatchNormalization
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
add_2: Add
input: [(?, 55, 55, 256), (?, 55, 55, 256)]
output: (?, 55, 55, 256)
activation_7: Activation
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
conv2_3_1x1_reduce: Conv2D
input: (?, 55, 55, 256)
output: (?, 55, 55, 64)
conv2_3_1x1_reduce/bn: BatchNormalization
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
activation_8: Activation
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)

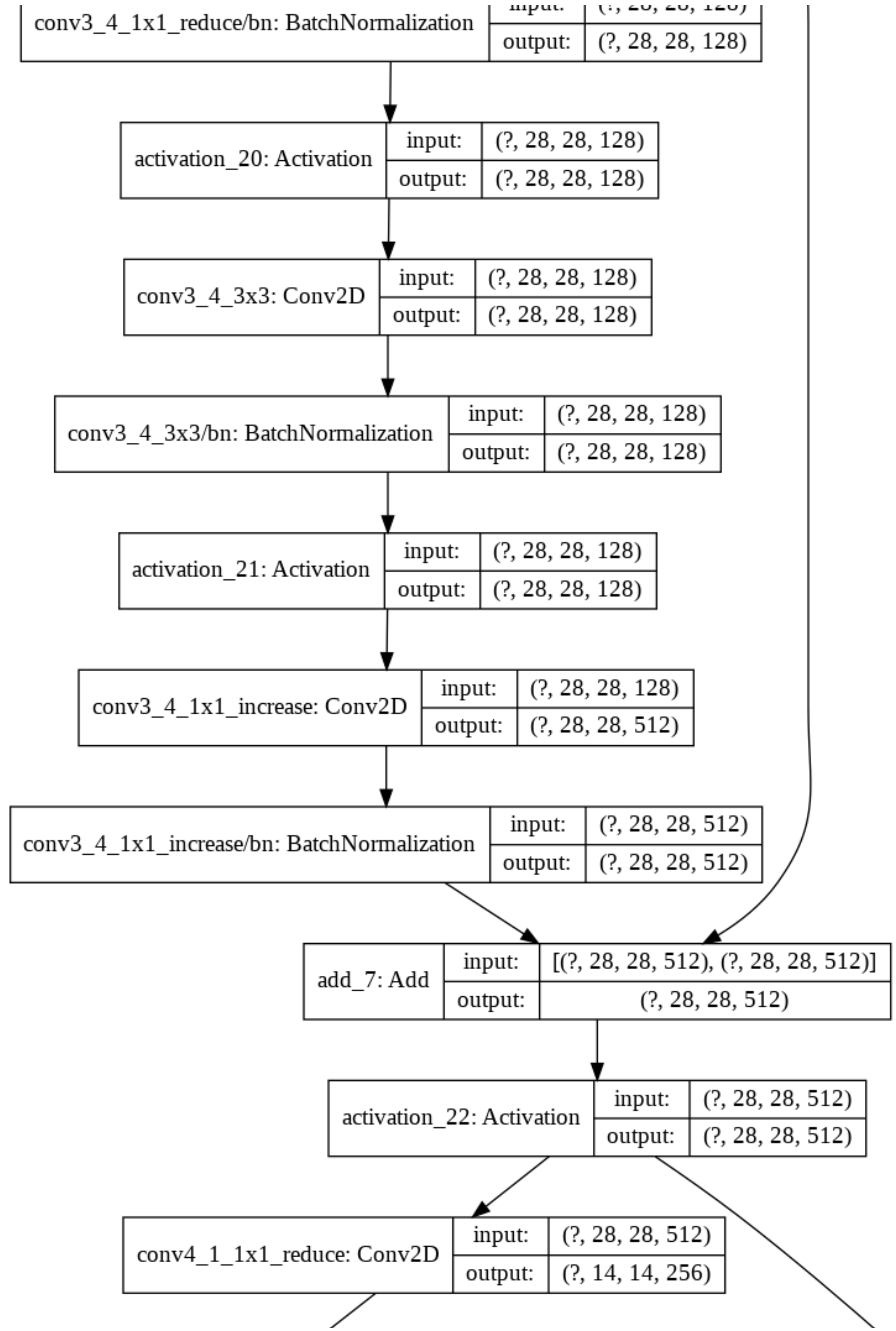

conv2_3_3x3: Conv2D
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
conv2_3_3x3/bn: BatchNormalization
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
activation_9: Activation
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
conv2_3_1x1_increase: Conv2D
input: (?, 55, 55, 64)
output: (?, 55, 55, 256)
conv2_3_1x1_increase/bn: BatchNormalization
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
add_3: Add
input: [(?, 55, 55, 256), (?, 55, 55, 256)]
output: (?, 55, 55, 256)
activation_10: Activation
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
conv3_1_1x1_reduce: Conv2D
input: (?, 55, 55, 256)
output: (?, 28, 28, 128)
conv3_1_1x1_reduce/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_1_1x1_proj: Conv2D
input: (?, 55, 55, 256)
output: (?, 28, 28, 512)


activation_11: Activation
output: (?, 28, 28, 128)
conv3_1_1x1_proj/bn: BatchNormalization
output: (?, 28, 28, 512)
conv3_1_3x3: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_1_3x3/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
activation_12: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_1_1x1_increase: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 512)
conv3_1_1x1_increase/bn: BatchNormalization
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
add_4: Add
input: [(?, 28, 28, 512), (?, 28, 28, 512)]
output: (?, 28, 28, 512)
activation_13: Activation
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
conv3_2_1x1_reduce: Conv2D
input: (?, 28, 28, 512)
output: (?, 28, 28, 128)
conv3_2_1x1_reduce/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)

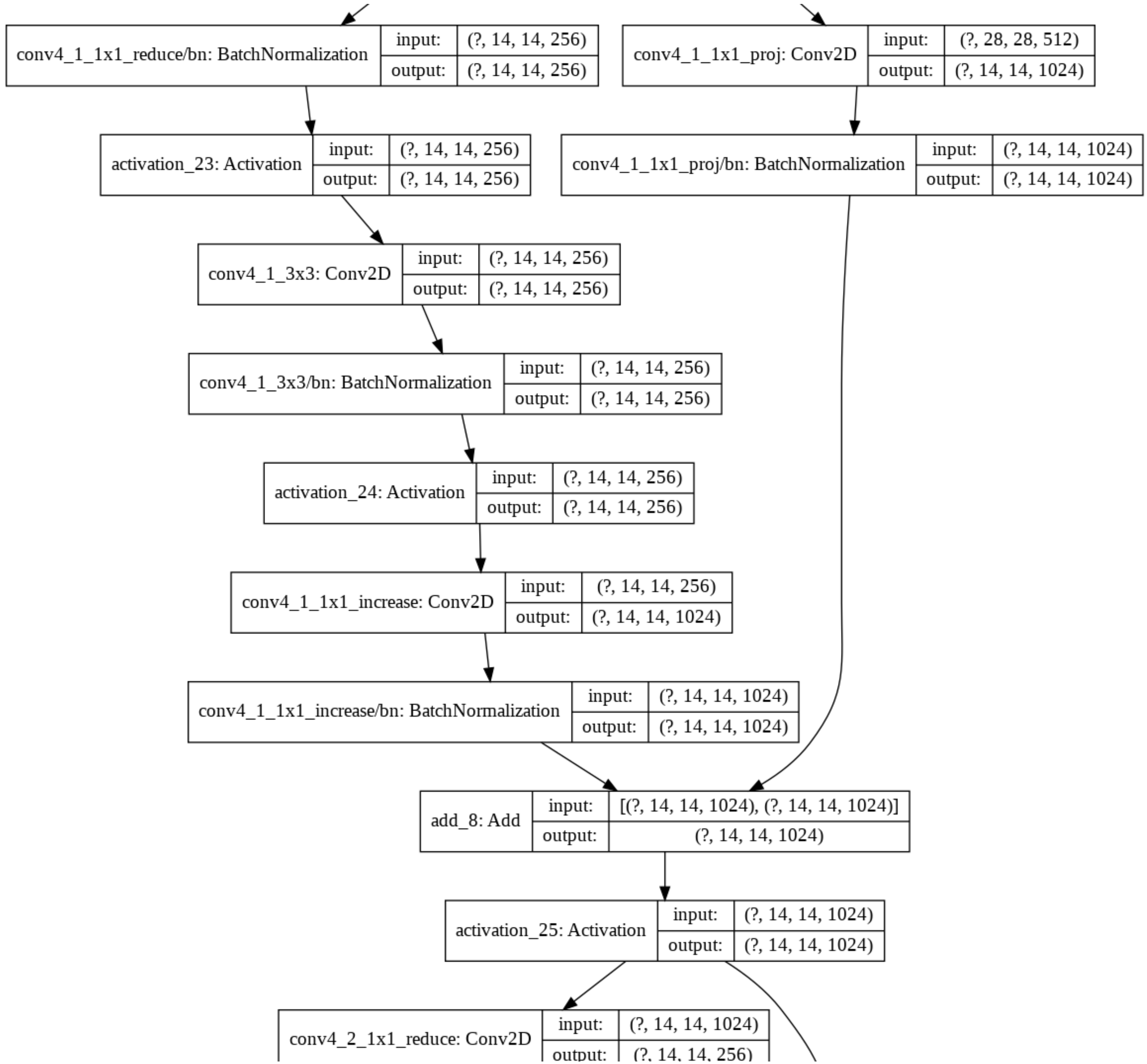

activation_14: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_2_3x3: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_2_3x3/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
activation_15: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_2_1x1_increase: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 512)
conv3_2_1x1_increase/bn: BatchNormalization
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
add_5: Add
input: [(?, 28, 28, 512), (?, 28, 28, 512)]
output: (?, 28, 28, 512)
activation_16: Activation
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
conv3_3_1x1_reduce: Conv2D
input: (?, 28, 28, 512)
output: (?, 28, 28, 128)
conv3_3_1x1_reduce/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)

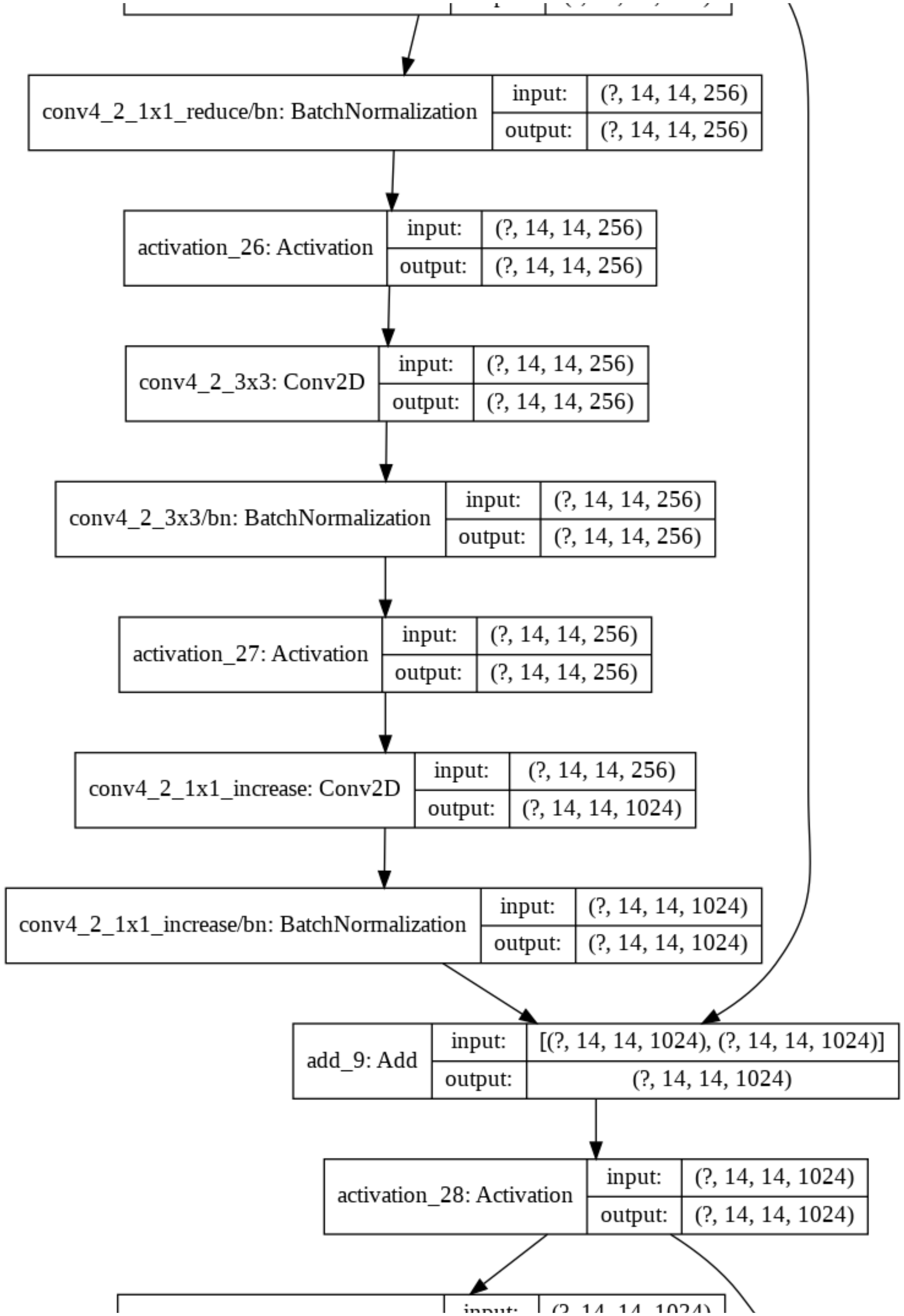


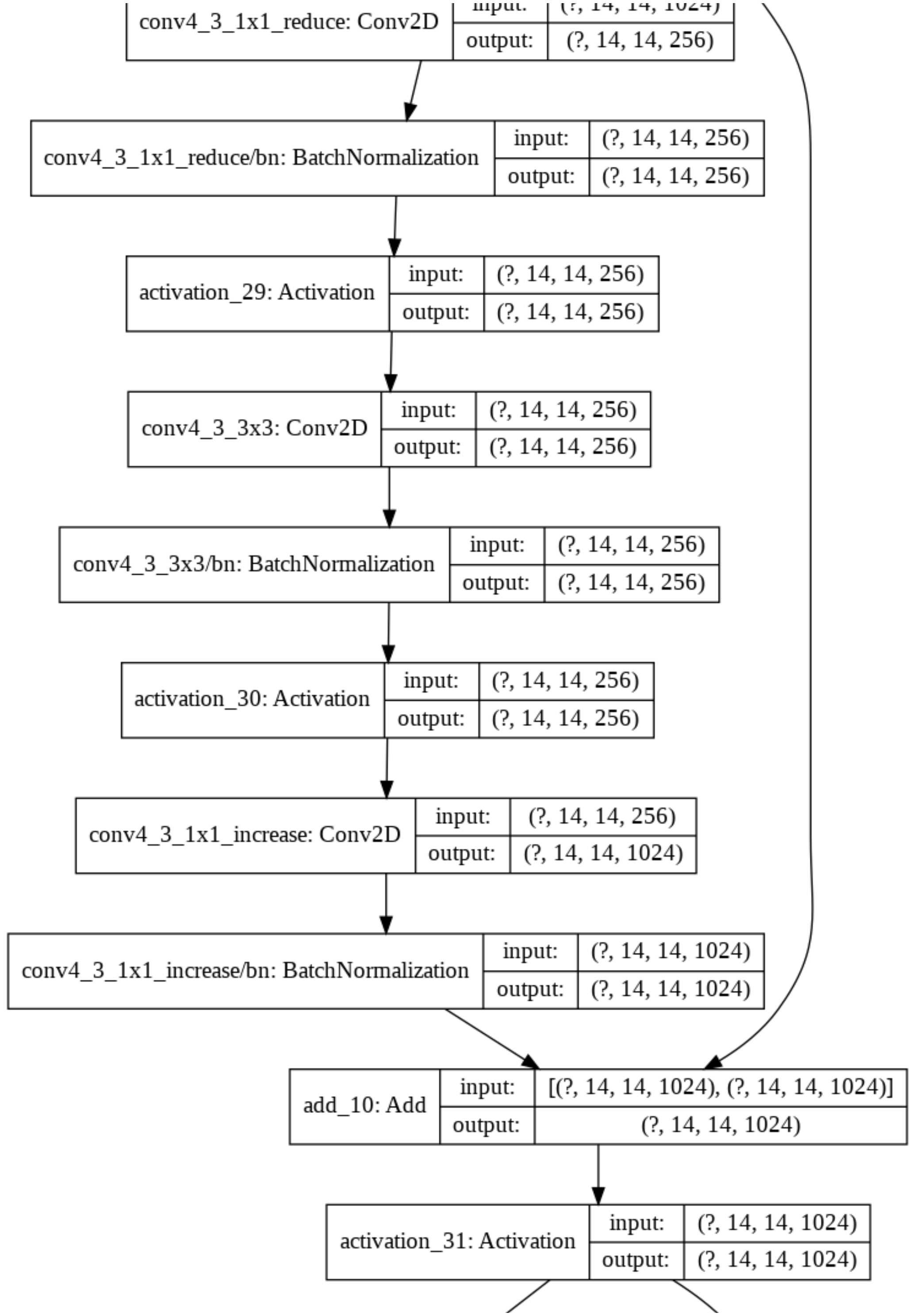
activation_17: Activation
input:
(?, 28, 28, 128)
output:
(?, 28, 28, 128)
conv3_3_3x3: Conv2D
input:
(?, 28, 28, 128)
output:
(?, 28, 28, 128)
conv3_3_3x3/bn: BatchNormalization
input:
(?, 28, 28, 128)
output:
(?, 28, 28, 128)
activation_18: Activation
input:
(?, 28, 28, 128)
output:
(?, 28, 28, 128)
conv3_3_1x1_increase: Conv2D
input:
(?, 28, 28, 128)
output:
(?, 28, 28, 512)
conv3_3_1x1_increase/bn: BatchNormalization
input:
(?, 28, 28, 512)
output:
(?, 28, 28, 512)
add_6: Add
input:
[(?, 28, 28, 512), (?, 28, 28, 512)]
output:
(?, 28, 28, 512)
activation_19: Activation
input:
(?, 28, 28, 512)
output:
(?, 28, 28, 512)
conv3_4_1x1_reduce: Conv2D
input:
(?, 28, 28, 512)
output:
(?, 28, 28, 128)



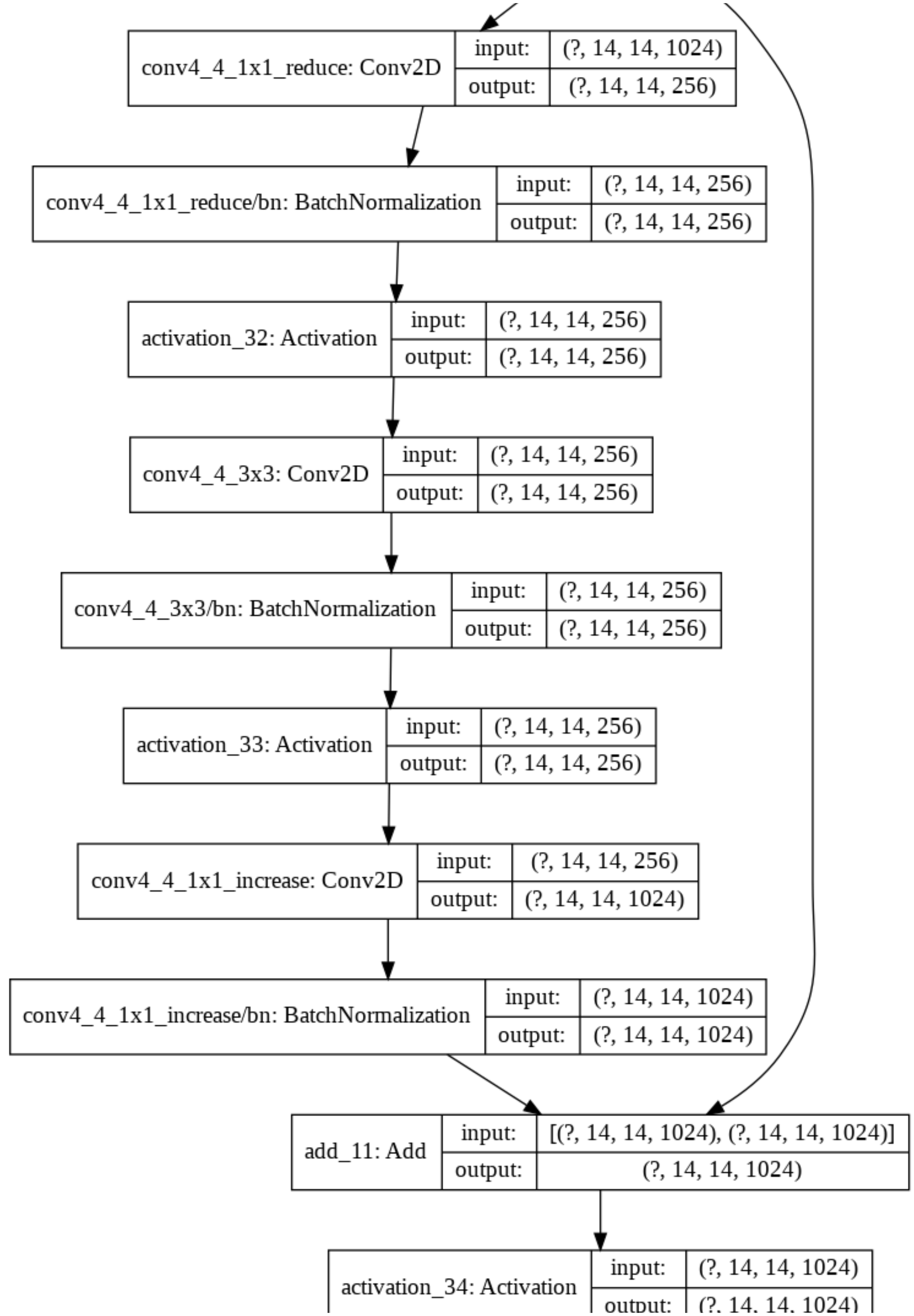
conv3_4_1x1_reduce/bn: BatchNormalization
output: (?, 28, 28, 128)
activation_20: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_4_3x3: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_4_3x3/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
activation_21: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_4_1x1_increase: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 512)
conv3_4_1x1_increase/bn: BatchNormalization
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
add_7: Add
input: [(?, 28, 28, 512), (?, 28, 28, 512)]
output: (?, 28, 28, 512)
activation_22: Activation
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
conv4_1_1x1_reduce: Conv2D
input: (?, 28, 28, 512)
output: (?, 14, 14, 256)


conv4_1_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_1_1x1_proj: Conv2D
input: (?, 28, 28, 512)
output: (?, 14, 14, 1024)
activation_23: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_1_1x1_proj/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
conv4_1_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_1_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_24: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_1_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)
conv4_1_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_8: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)
activation_25: Activation
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
conv4_2_1x1_reduce: Conv2D
input: (?, 14, 14, 1024)
output: (?, 14, 14, 256)

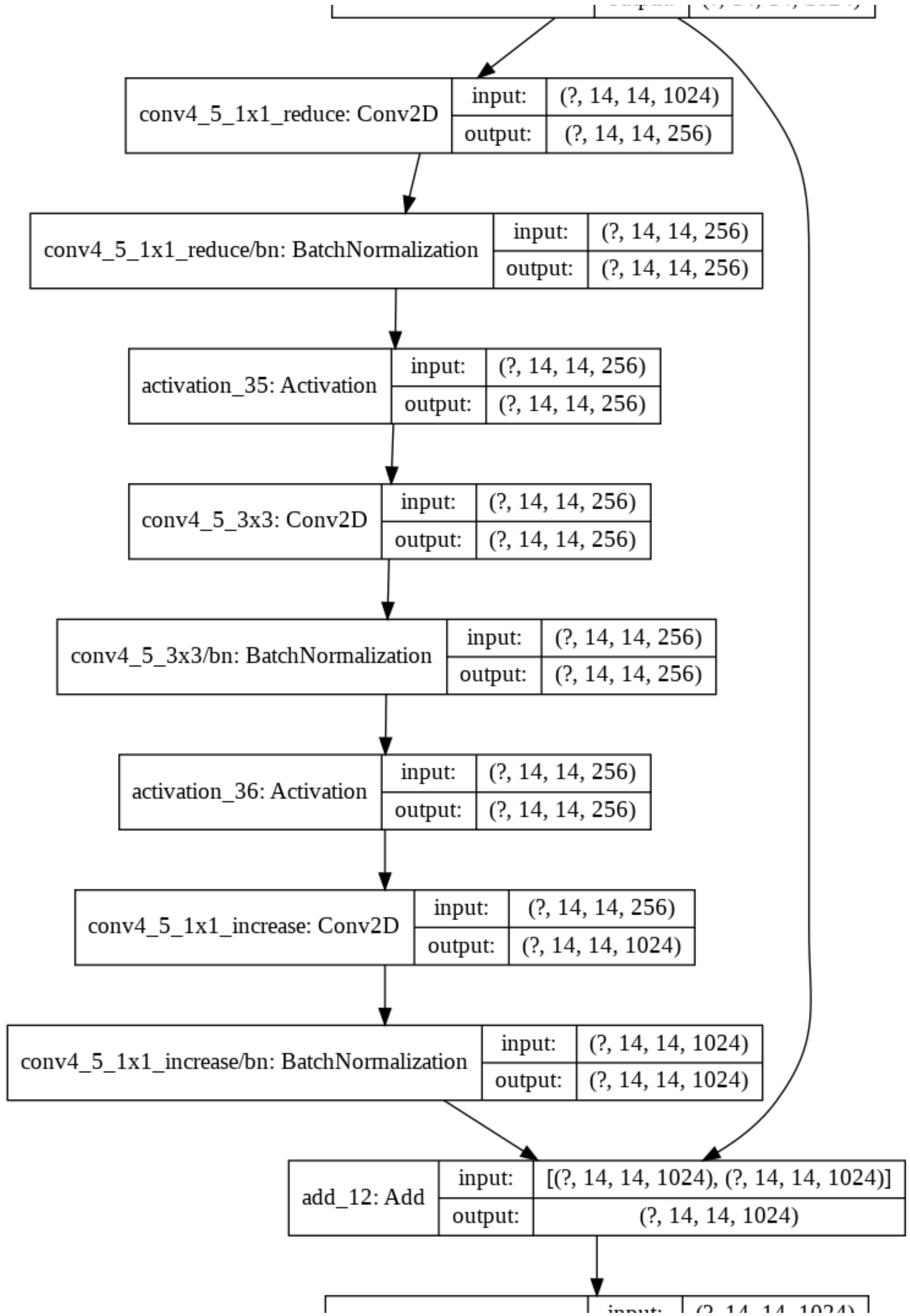

conv4_2_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_26: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_2_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_2_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_27: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_2_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)
conv4_2_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_9: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)
activation_28: Activation
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)



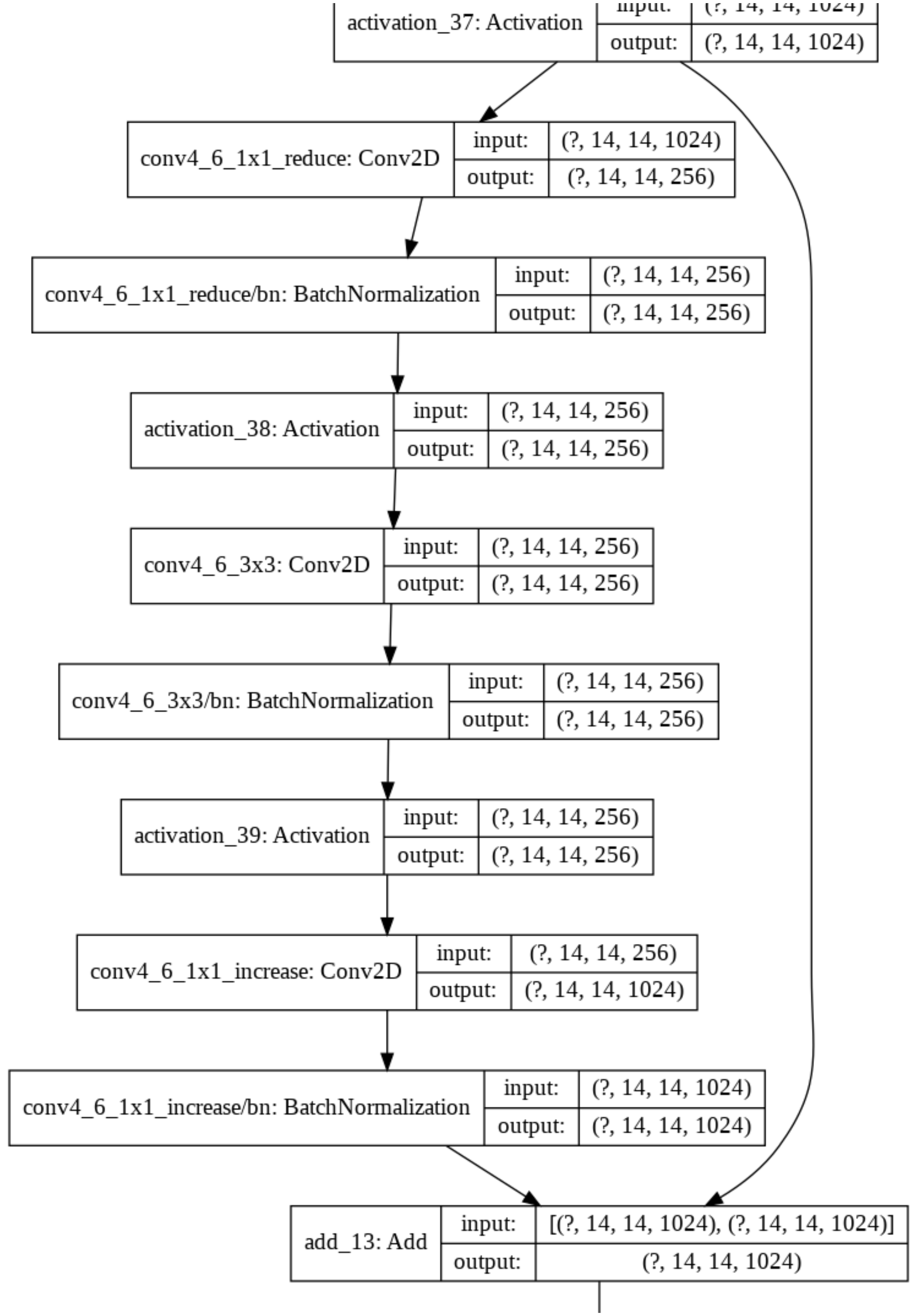
conv4_3_1x1_reduce: Conv2D
output: (?, 14, 14, 256)
conv4_3_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_29: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_3_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_3_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_30: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_3_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)
conv4_3_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_10: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)
activation_31: Activation
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)

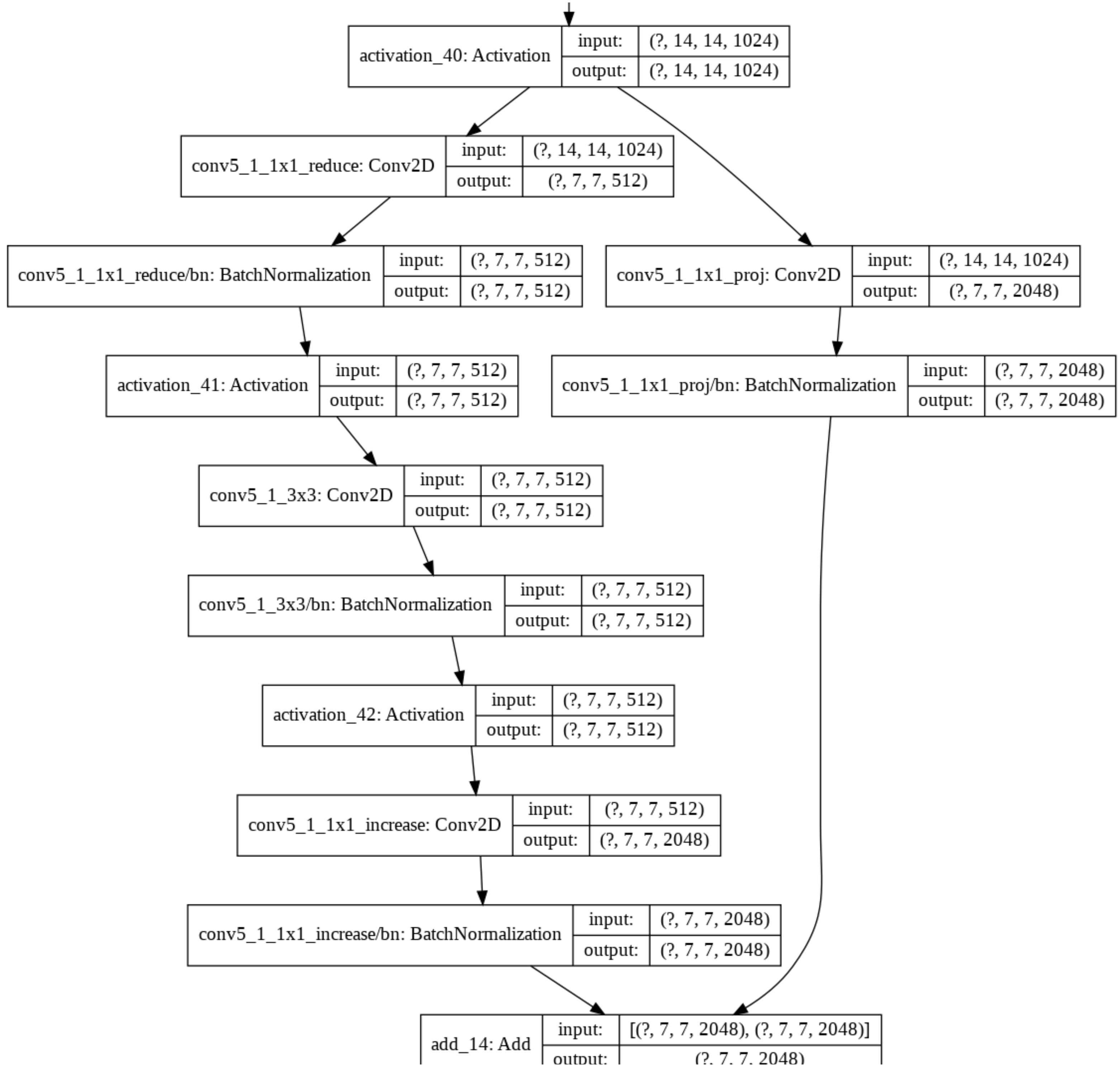


conv4_4_1x1_reduce: Conv2D
input: (?, 14, 14, 1024)
output: (?, 14, 14, 256)
conv4_4_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_32: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_4_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_4_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_33: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_4_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)
conv4_4_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_11: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)
activation_34: Activation
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)

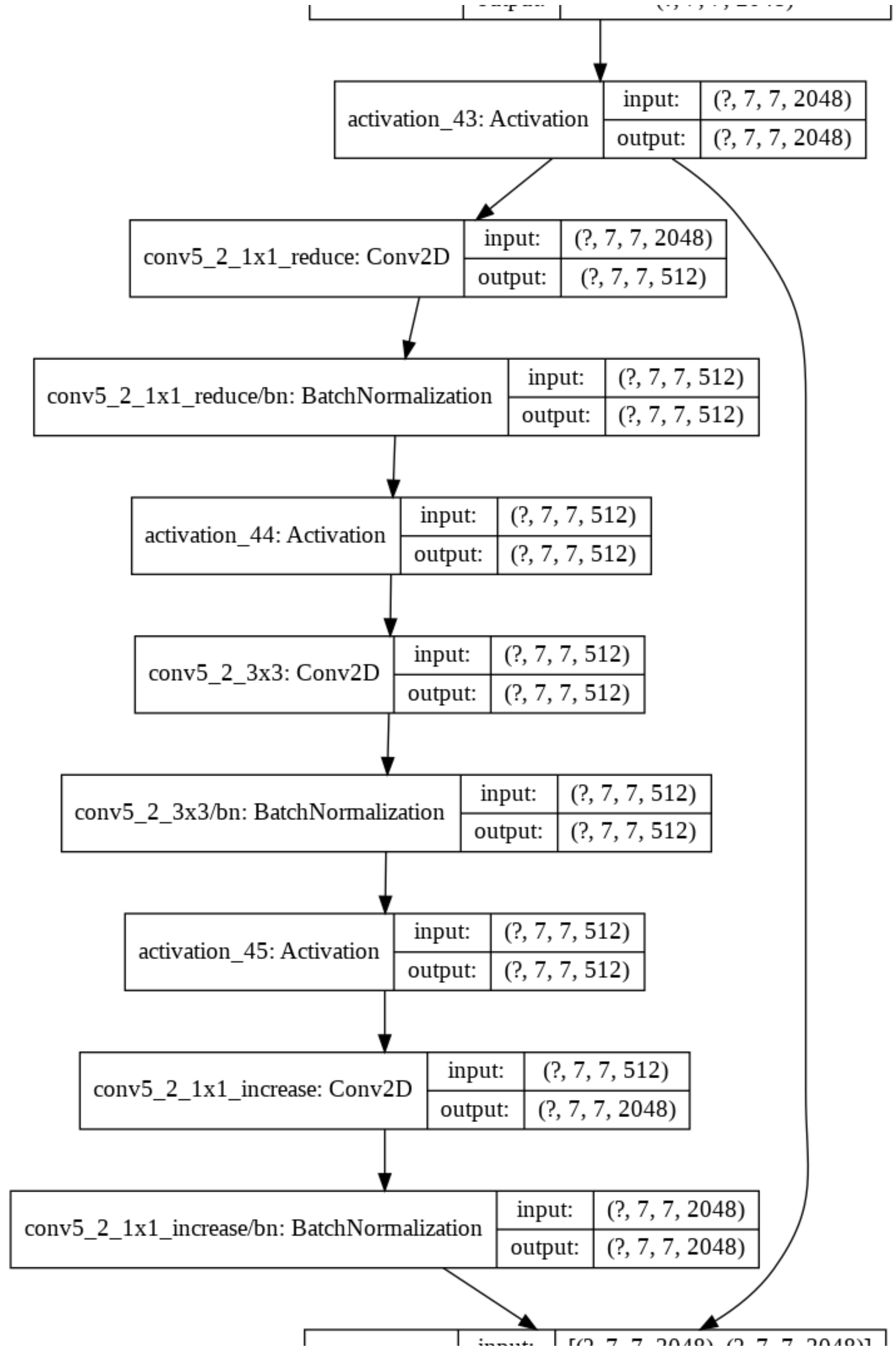


conv4_5_1x1_reduce: Conv2D
input: (?, 14, 14, 1024)
output: (?, 14, 14, 256)
conv4_5_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_35: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_5_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_5_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_36: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_5_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)
conv4_5_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_12: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)

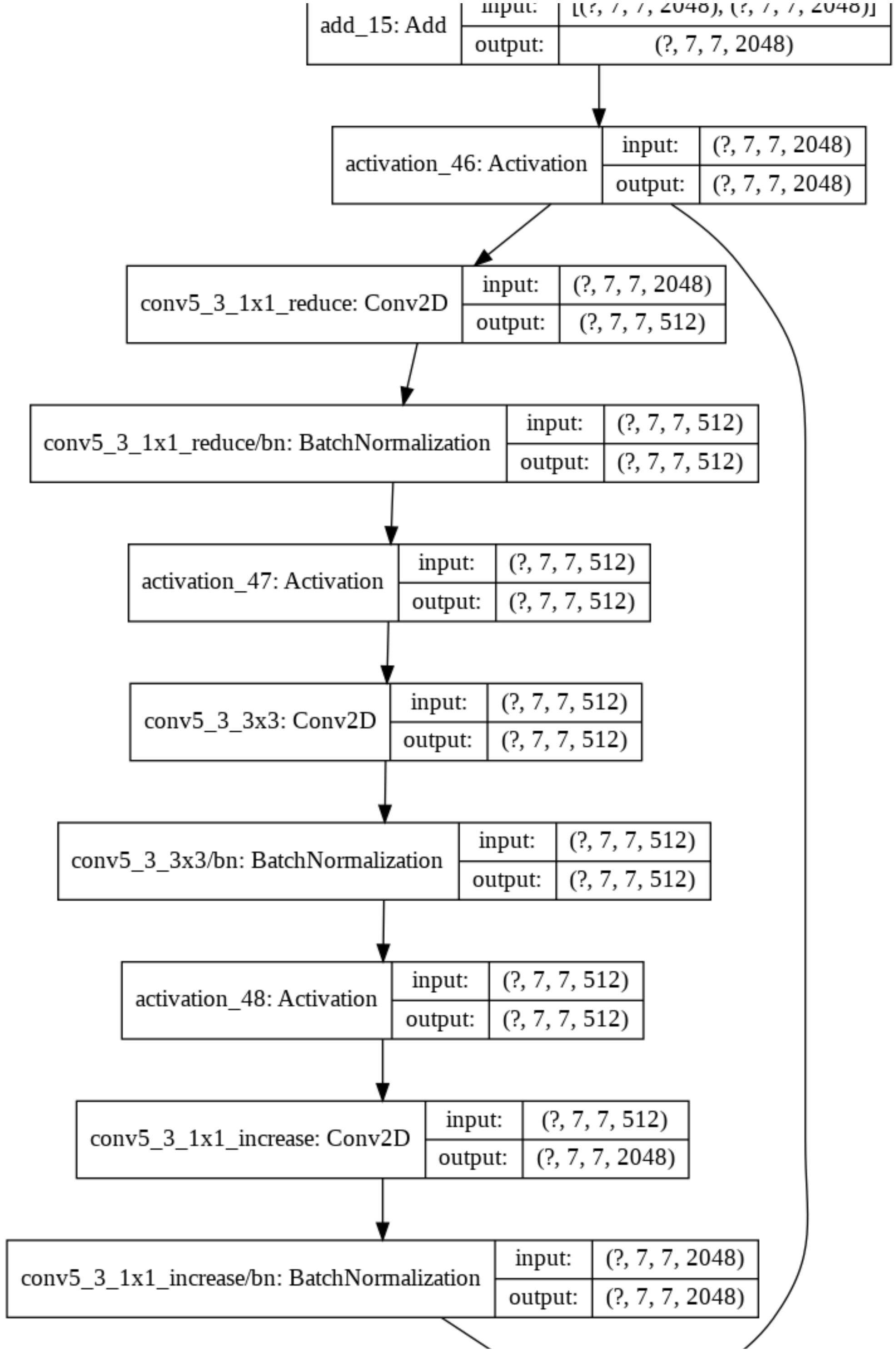

activation_37: Activation
output: (?, 14, 14, 1024)
conv4_6_1x1_reduce: Conv2D
input: (?, 14, 14, 1024)
output: (?, 14, 14, 256)
conv4_6_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_38: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_6_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_6_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_39: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_6_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)
conv4_6_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_13: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)

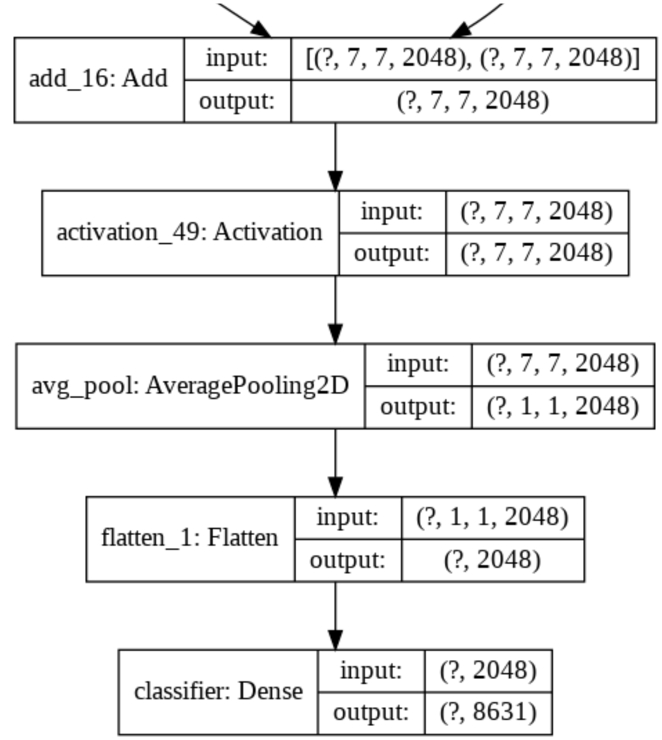


activation_40: Activation
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
conv5_1_1x1_reduce: Conv2D
input: (?, 14, 14, 1024)
output: (?, 7, 7, 512)
conv5_1_1x1_reduce/bn: BatchNormalization
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_1_1x1_proj: Conv2D
input: (?, 14, 14, 1024)
output: (?, 7, 7, 2048)
activation_41: Activation
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_1_1x1_proj/bn: BatchNormalization
input: (?, 7, 7, 2048)
output: (?, 7, 7, 2048)
conv5_1_3x3: Conv2D
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_1_3x3/bn: BatchNormalization
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
activation_42: Activation
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_1_1x1_increase: Conv2D
input: (?, 7, 7, 512)
output: (?, 7, 7, 2048)
conv5_1_1x1_increase/bn: BatchNormalization
input: (?, 7, 7, 2048)
output: (?, 7, 7, 2048)
add_14: Add
input: [(?, 7, 7, 2048), (?, 7, 7, 2048)]
output: (?, 7, 7, 2048)



activation_43: Activation
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)
conv5_2_1x1_reduce: Conv2D
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 512)
conv5_2_1x1_reduce/bn: BatchNormalization
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
activation_44: Activation
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_2_3x3: Conv2D
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_2_3x3/bn: BatchNormalization
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
activation_45: Activation
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_2_1x1_increase: Conv2D
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 2048)
conv5_2_1x1_increase/bn: BatchNormalization
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)


add_15: Add
output:
(?, 7, 7, 2048)
activation_46: Activation
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)
conv5_3_1x1_reduce: Conv2D
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 512)
conv5_3_1x1_reduce/bn: BatchNormalization
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
activation_47: Activation
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_3_3x3: Conv2D
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_3_3x3/bn: BatchNormalization
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
activation_48: Activation
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_3_1x1_increase: Conv2D
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 2048)
conv5_3_1x1_increase/bn: BatchNormalization
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)


add_16: Add
input:
[(?, 7, 7, 2048), (?, 7, 7, 2048)]
output:
(?, 7, 7, 2048)
activation_49: Activation
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)
avg_pool: AveragePooling2D
input:
(?, 7, 7, 2048)
output:
(?, 1, 1, 2048)
flatten_1: Flatten
input:
(?, 1, 1, 2048)
output:
(?, 2048)
classifier: Dense
input:
(?, 2048)
output:
(?, 8631)

Before we can make a prediction with a face, the pixel values must be scaled in the same way that data was prepared when the VGGFace model was trained. Specifically, the pixel values must be centered on each channel using the mean from the training dataset.

```
# convert one face into samples
pixels = pixels.astype('float32')
samples = np.expand_dims(pixels, axis=0)
# prepare the face for the model, e.g. center pixels
samples = preprocess_input(samples, version=2)

# perform prediction
yhat = model.predict(samples)
# convert prediction into names
results = decode_predictions(yhat)
# display most likely results
for result in results[0]:
    print('%s: %.3f%%' % (result[0], result[1]*100))
```

```
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:422: The name tf.global_variat

Downloading data from https://github.com/rcmalli/keras-vggface/releases/download/v2.0/rcmalli_vggface_labels_v2.npy
1351680/1346516 [==============================] - 0s 0us/step
b' Sharon_Stone': 99.574%
b' Noelle_Reno': 0.080%
b' Anita_Lipnicka': 0.027%
b' Elisabeth_R\xc3\xb6hm': 0.027%
b' Emma_Atkins': 0.019%
```

Do you want to find out which super star(s) you look like the most? Upload your images to your Google Drive, and change the image_path to the path of your own image in Section 2. The place between 'start your code here' and 'end your code here'.

4. Face verification

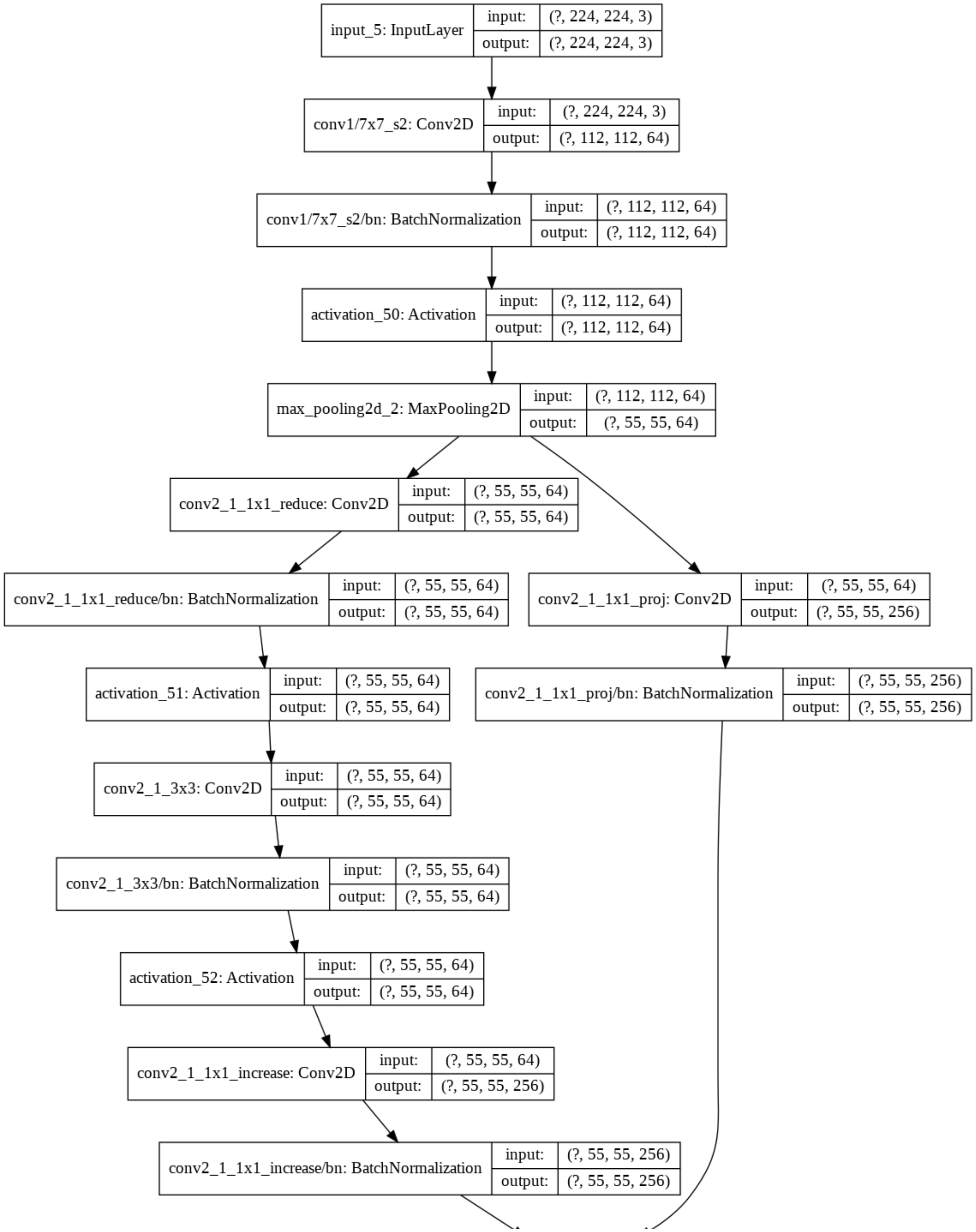
Have enough fun? Now, let us see whether the VGGFace model can tell I am who I am.

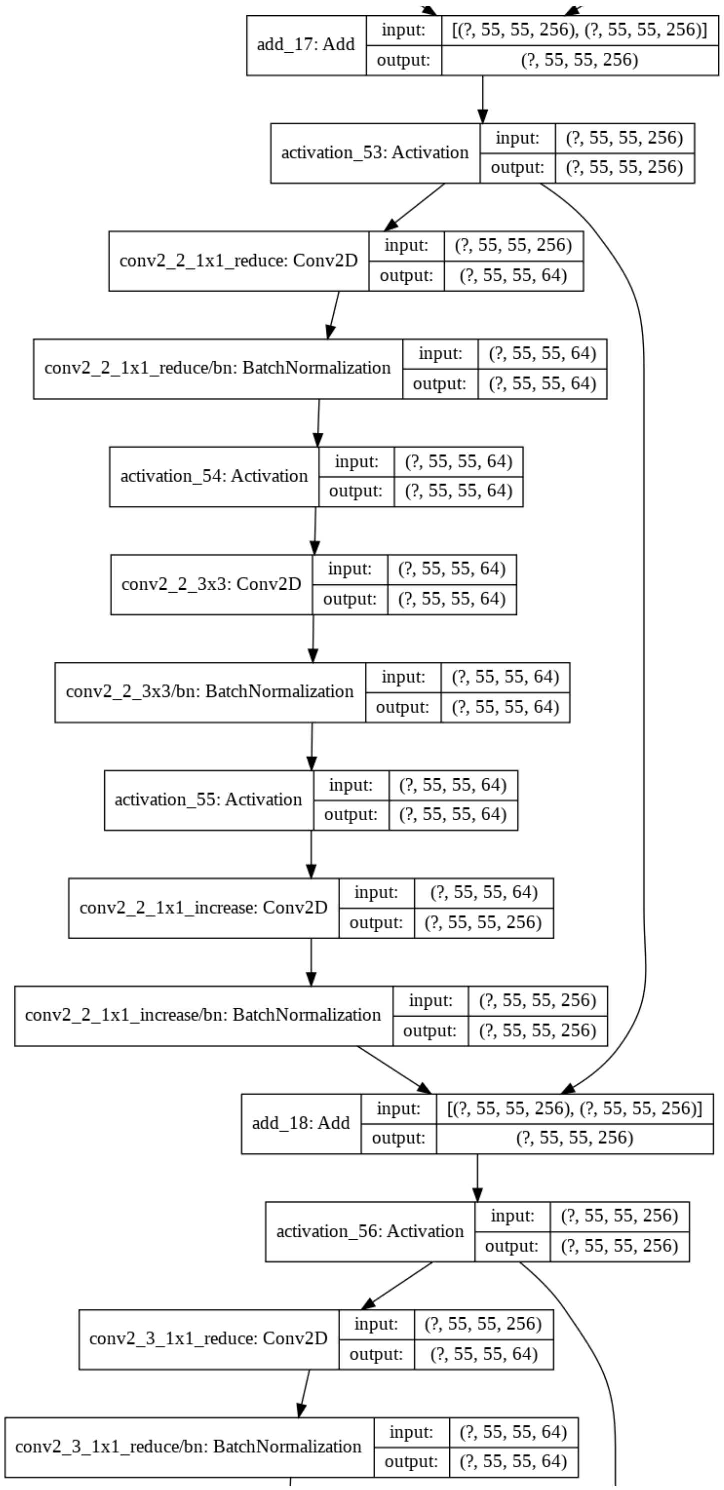
First of all, we need to extract the features from the face. Compared with the model we used before, we would like the output from the flatten layer. Therefore, this time when loading the model, we will ignore the top dense layer.

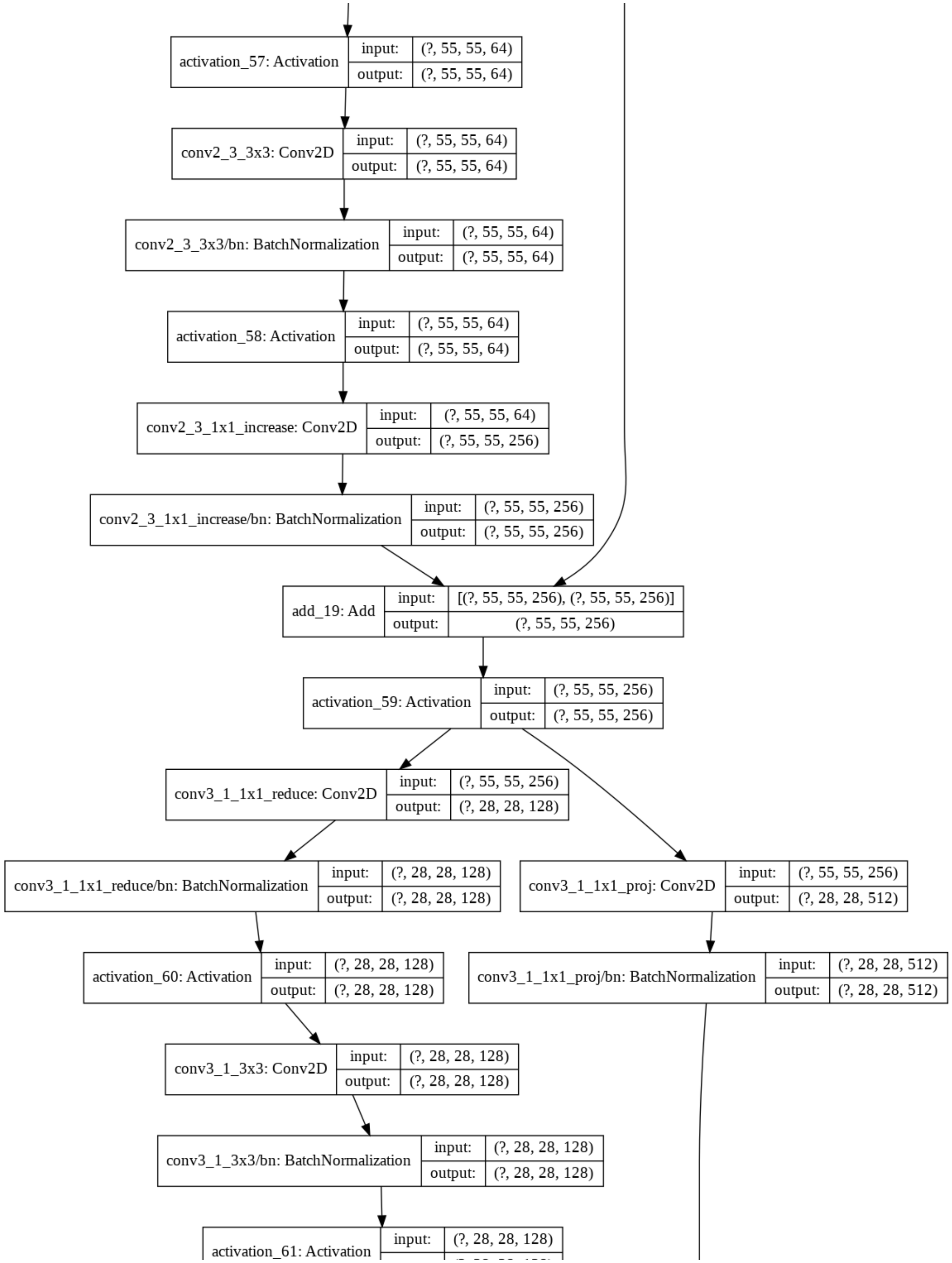
Exercise: please compare 'model' with 'model_notop'. Do you see any differences?

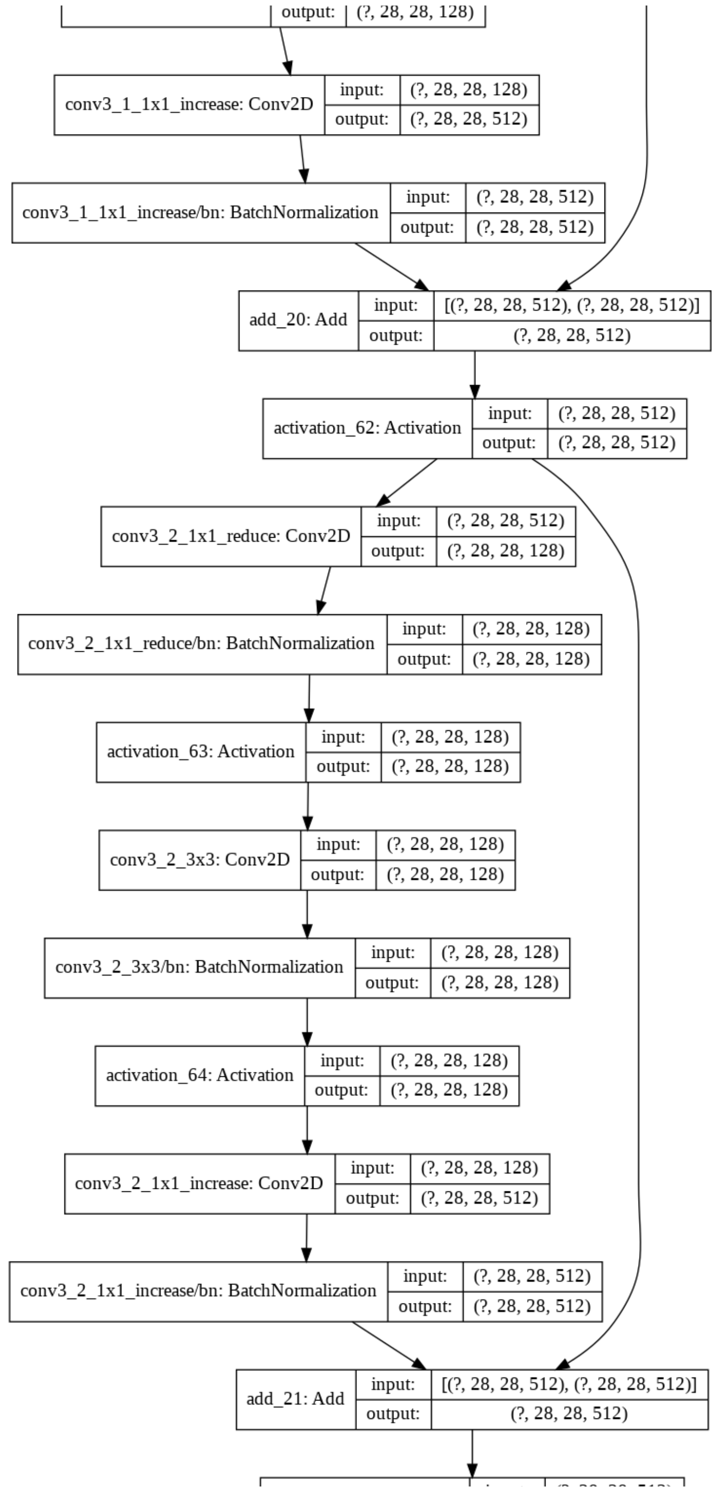
```
# create a vggface model without top levels
model_notop = VGGFace(model='resnet50', include_top=False, input_shape=(224, 224, 3), pooling='avg')
tf.keras.utils.plot_model(model_notop, show_shapes=True)
```

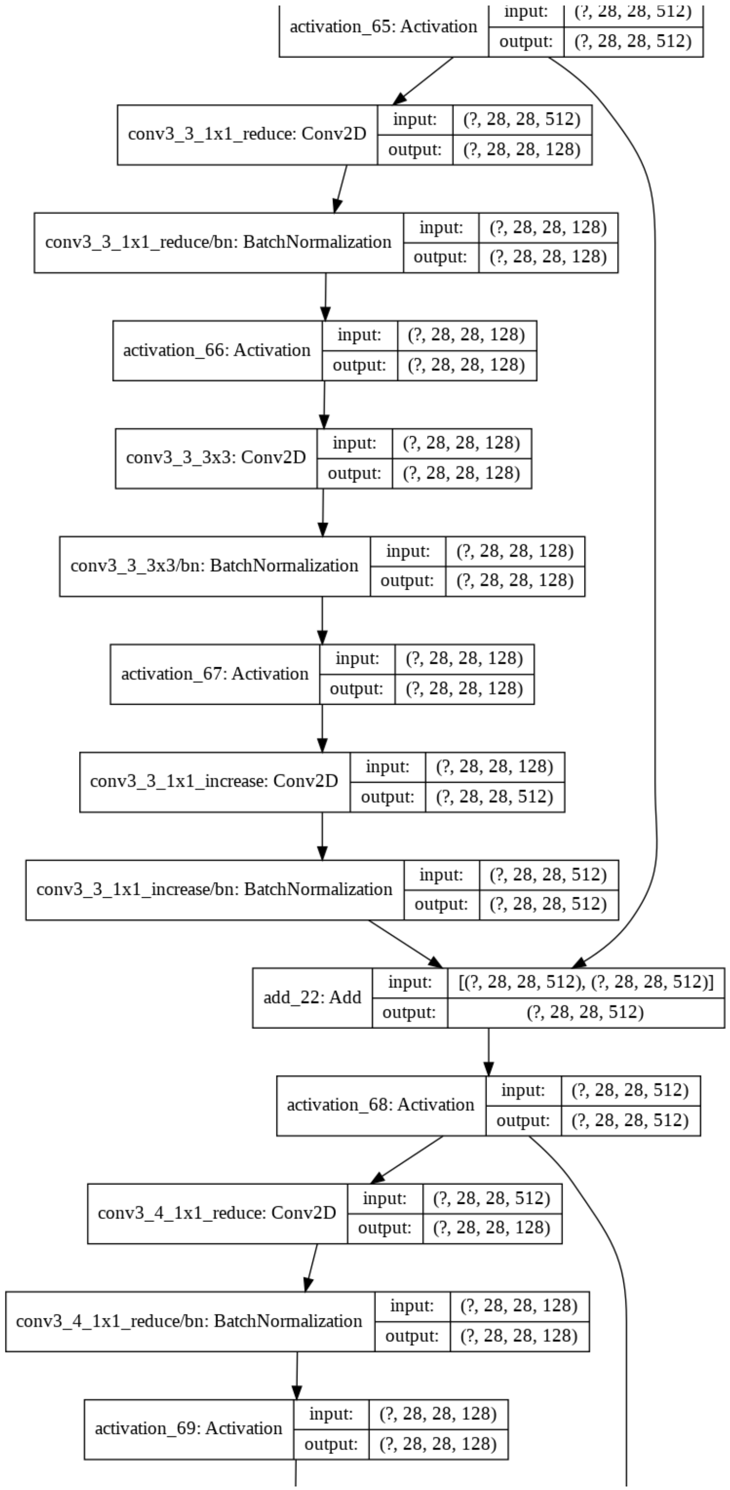
```
Downloading data from https://github.com/rcmalli/keras-vggface/releases/download/v
94699520/94694792 [==============================] - 2s 0us/step
```

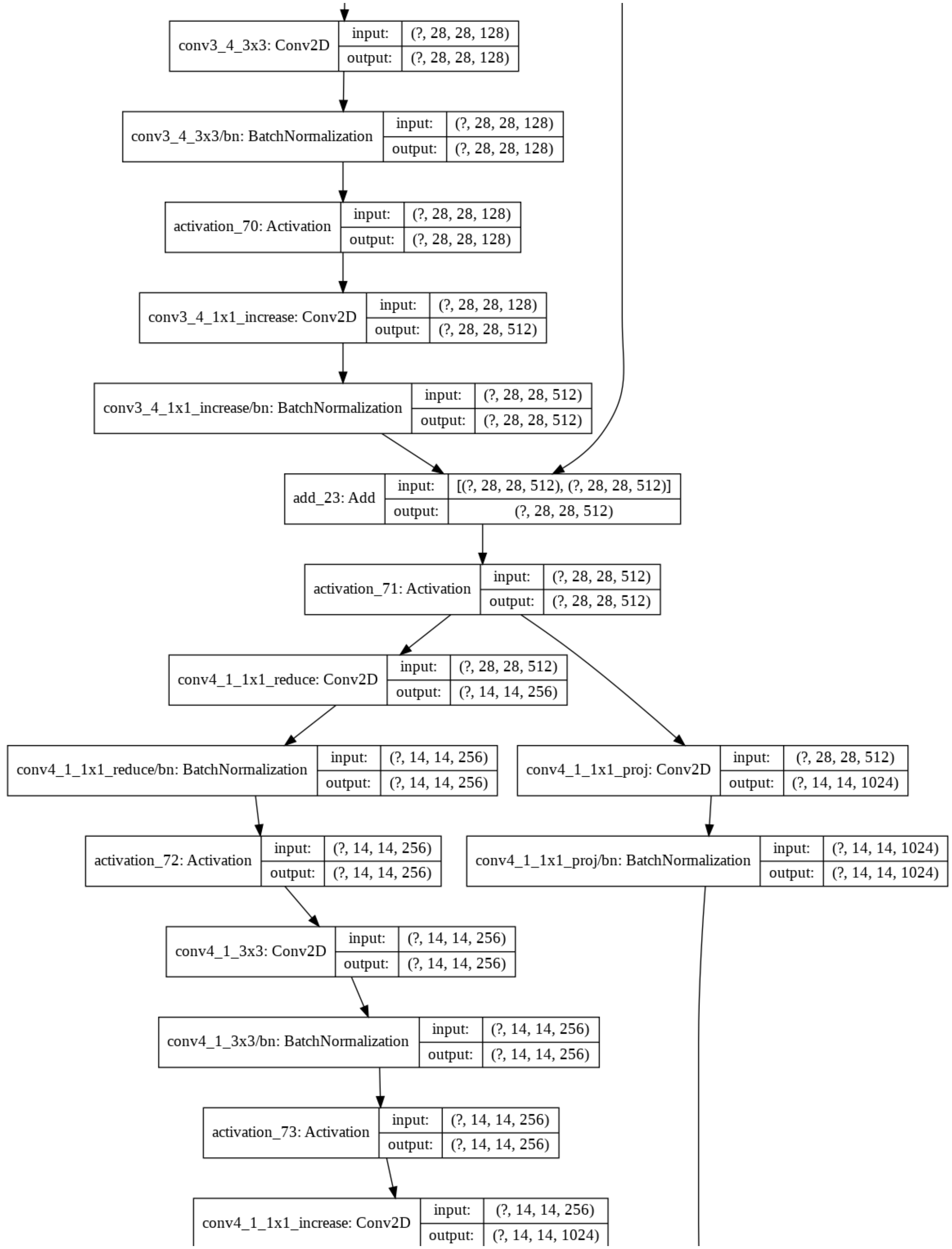



add_17: Add
input: [(?, 55, 55, 256), (?, 55, 55, 256)]
output: (?, 55, 55, 256)
activation_53: Activation
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
conv2_2_1x1_reduce: Conv2D
input: (?, 55, 55, 256)
output: (?, 55, 55, 64)
conv2_2_1x1_reduce/bn: BatchNormalization
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
activation_54: Activation
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
conv2_2_3x3: Conv2D
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
conv2_2_3x3/bn: BatchNormalization
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
activation_55: Activation
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)
conv2_2_1x1_increase: Conv2D
input: (?, 55, 55, 64)
output: (?, 55, 55, 256)
conv2_2_1x1_increase/bn: BatchNormalization
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
add_18: Add
input: [(?, 55, 55, 256), (?, 55, 55, 256)]
output: (?, 55, 55, 256)
activation_56: Activation
input: (?, 55, 55, 256)
output: (?, 55, 55, 256)
conv2_3_1x1_reduce: Conv2D
input: (?, 55, 55, 256)
output: (?, 55, 55, 64)
conv2_3_1x1_reduce/bn: BatchNormalization
input: (?, 55, 55, 64)
output: (?, 55, 55, 64)

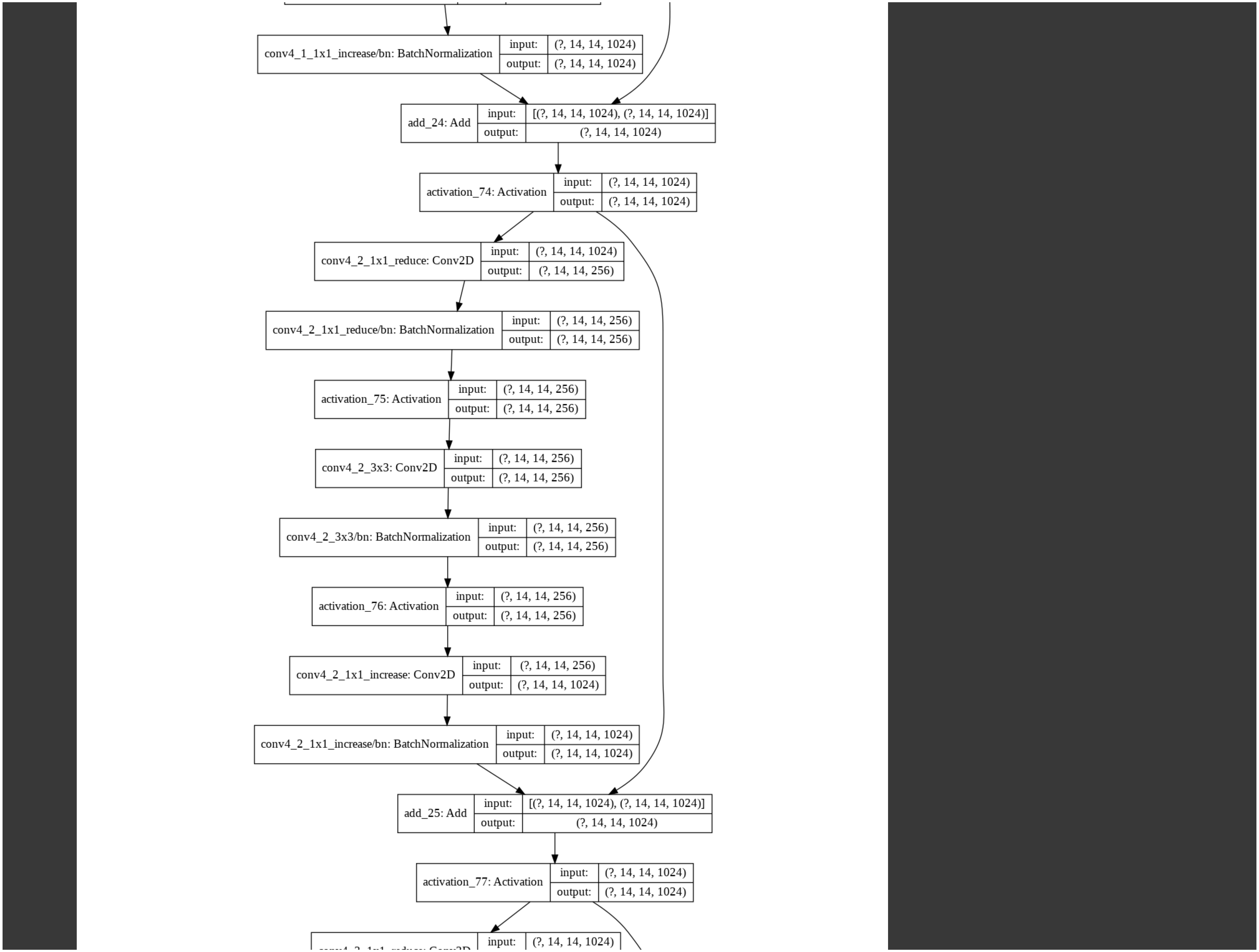

activation_57: Activation | input: | (?, 55, 55, 64) | output: | (?, 55, 55, 64)
conv2_3_3x3: Conv2D | input: | (?, 55, 55, 64) | output: | (?, 55, 55, 64)
conv2_3_3x3/bn: BatchNormalization | input: | (?, 55, 55, 64) | output: | (?, 55, 55, 64)
activation_58: Activation | input: | (?, 55, 55, 64) | output: | (?, 55, 55, 64)
conv2_3_1x1_increase: Conv2D | input: | (?, 55, 55, 64) | output: | (?, 55, 55, 256)
conv2_3_1x1_increase/bn: BatchNormalization | input: | (?, 55, 55, 256) | output: | (?, 55, 55, 256)
add_19: Add | input: | [(?, 55, 55, 256), (?, 55, 55, 256)] | output: | (?, 55, 55, 256)
activation_59: Activation | input: | (?, 55, 55, 256) | output: | (?, 55, 55, 256)
conv3_1_1x1_reduce: Conv2D | input: | (?, 55, 55, 256) | output: | (?, 28, 28, 128)
conv3_1_1x1_reduce/bn: BatchNormalization | input: | (?, 28, 28, 128) | output: | (?, 28, 28, 128)
conv3_1_1x1_proj: Conv2D | input: | (?, 55, 55, 256) | output: | (?, 28, 28, 512)
activation_60: Activation | input: | (?, 28, 28, 128) | output: | (?, 28, 28, 128)
conv3_1_1x1_proj/bn: BatchNormalization | input: | (?, 28, 28, 512) | output: | (?, 28, 28, 512)
conv3_1_3x3: Conv2D | input: | (?, 28, 28, 128) | output: | (?, 28, 28, 128)
conv3_1_3x3/bn: BatchNormalization | input: | (?, 28, 28, 128) | output: | (?, 28, 28, 128)
activation_61: Activation | input: | (?, 28, 28, 128)

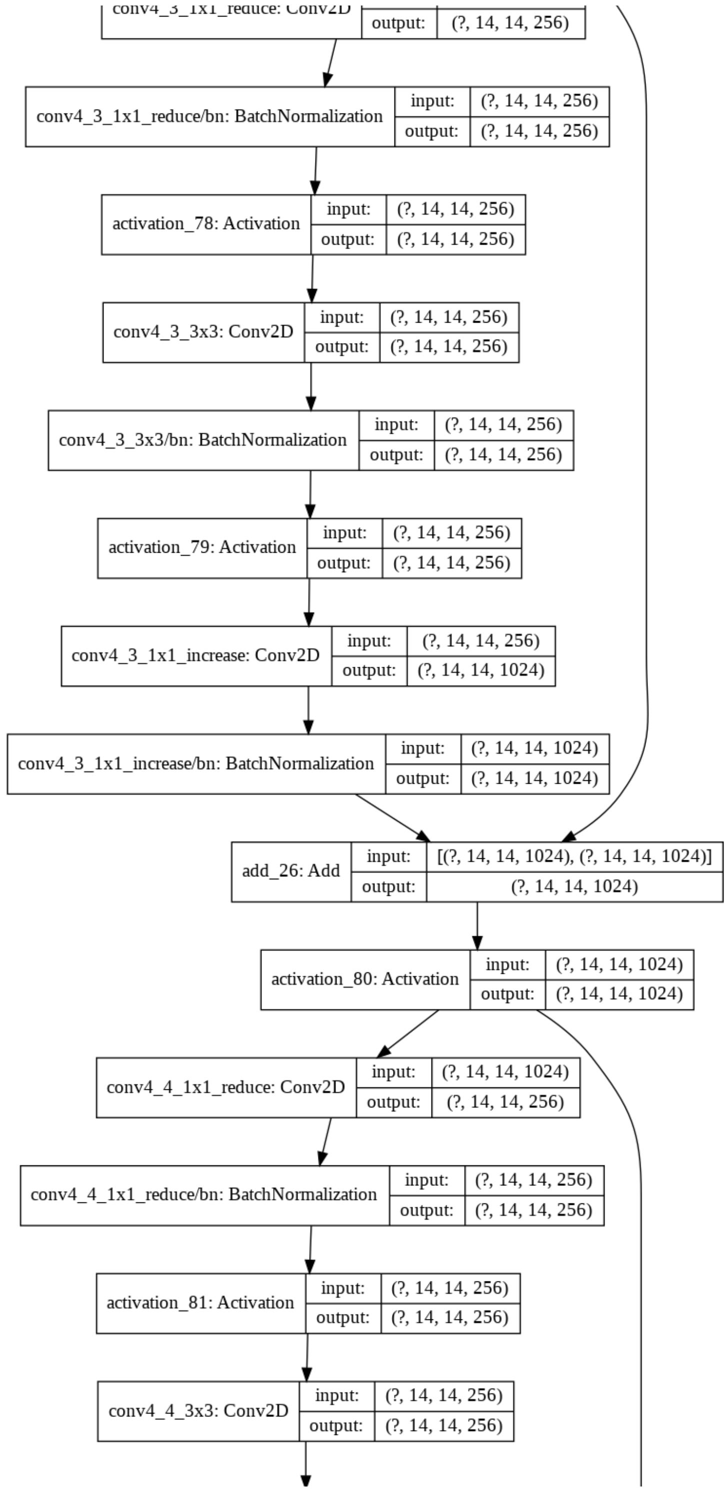

output: (?, 28, 28, 128)
conv3_1_1x1_increase: Conv2D input: (?, 28, 28, 128) output: (?, 28, 28, 512)
conv3_1_1x1_increase/bn: BatchNormalization input: (?, 28, 28, 512) output: (?, 28, 28, 512)
add_20: Add input: [(?, 28, 28, 512), (?, 28, 28, 512)] output: (?, 28, 28, 512)
activation_62: Activation input: (?, 28, 28, 512) output: (?, 28, 28, 512)
conv3_2_1x1_reduce: Conv2D input: (?, 28, 28, 512) output: (?, 28, 28, 128)
conv3_2_1x1_reduce/bn: BatchNormalization input: (?, 28, 28, 128) output: (?, 28, 28, 128)
activation_63: Activation input: (?, 28, 28, 128) output: (?, 28, 28, 128)
conv3_2_3x3: Conv2D input: (?, 28, 28, 128) output: (?, 28, 28, 128)
conv3_2_3x3/bn: BatchNormalization input: (?, 28, 28, 128) output: (?, 28, 28, 128)
activation_64: Activation input: (?, 28, 28, 128) output: (?, 28, 28, 128)
conv3_2_1x1_increase: Conv2D input: (?, 28, 28, 128) output: (?, 28, 28, 512)
conv3_2_1x1_increase/bn: BatchNormalization input: (?, 28, 28, 512) output: (?, 28, 28, 512)
add_21: Add input: [(?, 28, 28, 512), (?, 28, 28, 512)] output: (?, 28, 28, 512)


activation_65: Activation
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
conv3_3_1x1_reduce: Conv2D
input: (?, 28, 28, 512)
output: (?, 28, 28, 128)
conv3_3_1x1_reduce/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
activation_66: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_3_3x3: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_3_3x3/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
activation_67: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_3_1x1_increase: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 512)
conv3_3_1x1_increase/bn: BatchNormalization
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
add_22: Add
input: [(?, 28, 28, 512), (?, 28, 28, 512)]
output: (?, 28, 28, 512)
activation_68: Activation
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
conv3_4_1x1_reduce: Conv2D
input: (?, 28, 28, 512)
output: (?, 28, 28, 128)
conv3_4_1x1_reduce/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
activation_69: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)

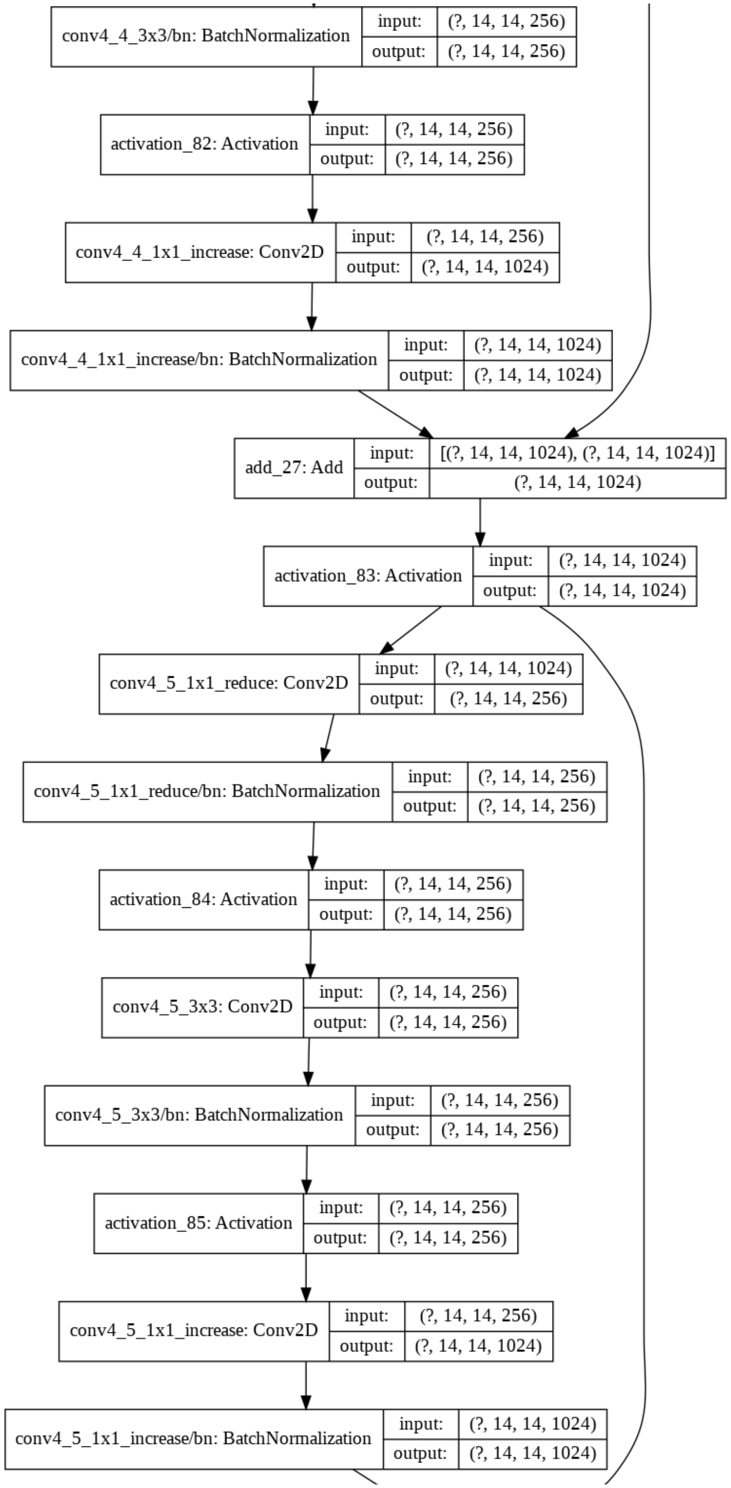


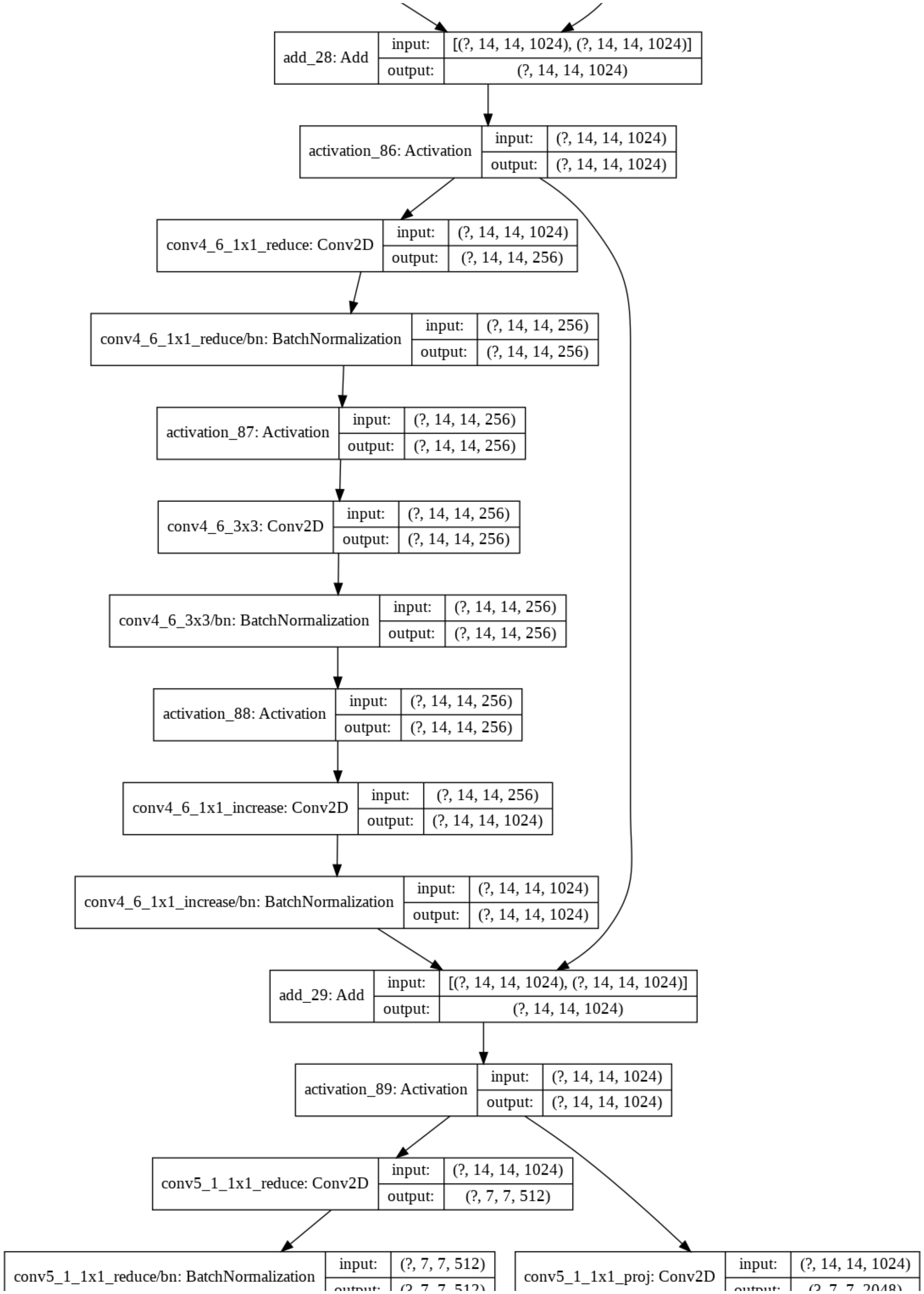
conv3_4_3x3: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_4_3x3/bn: BatchNormalization
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
activation_70: Activation
input: (?, 28, 28, 128)
output: (?, 28, 28, 128)
conv3_4_1x1_increase: Conv2D
input: (?, 28, 28, 128)
output: (?, 28, 28, 512)
conv3_4_1x1_increase/bn: BatchNormalization
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
add_23: Add
input: [(?, 28, 28, 512), (?, 28, 28, 512)]
output: (?, 28, 28, 512)
activation_71: Activation
input: (?, 28, 28, 512)
output: (?, 28, 28, 512)
conv4_1_1x1_reduce: Conv2D
input: (?, 28, 28, 512)
output: (?, 14, 14, 256)
conv4_1_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_1_1x1_proj: Conv2D
input: (?, 28, 28, 512)
output: (?, 14, 14, 1024)
activation_72: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_1_1x1_proj/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
conv4_1_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_1_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_73: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_1_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)

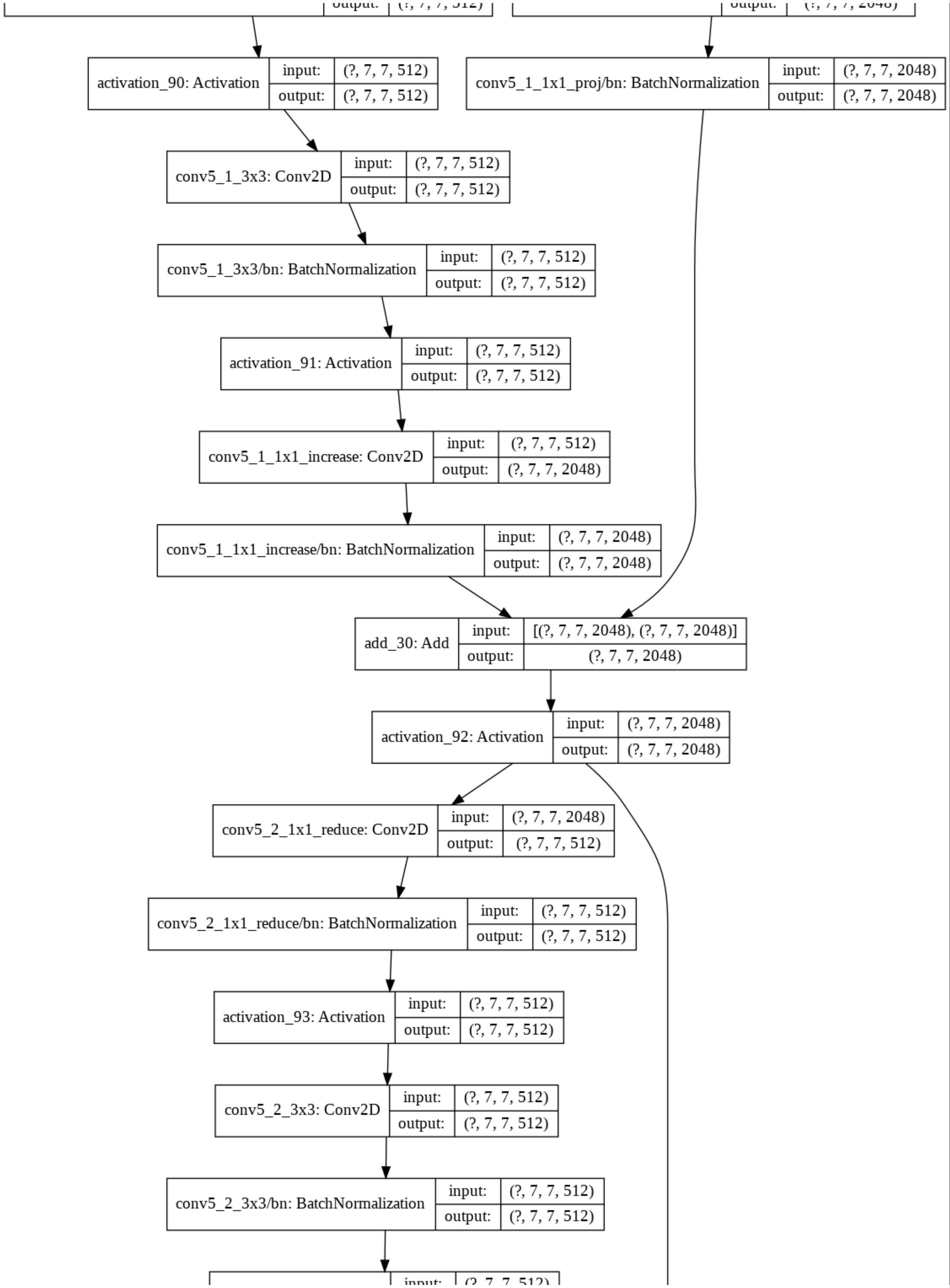

conv4_1_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_24: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)
activation_74: Activation
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
conv4_2_1x1_reduce: Conv2D
input: (?, 14, 14, 1024)
output: (?, 14, 14, 256)
conv4_2_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_75: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_2_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_2_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_76: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_2_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)
conv4_2_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_25: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)
activation_77: Activation
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
input: (?, 14, 14, 1024)



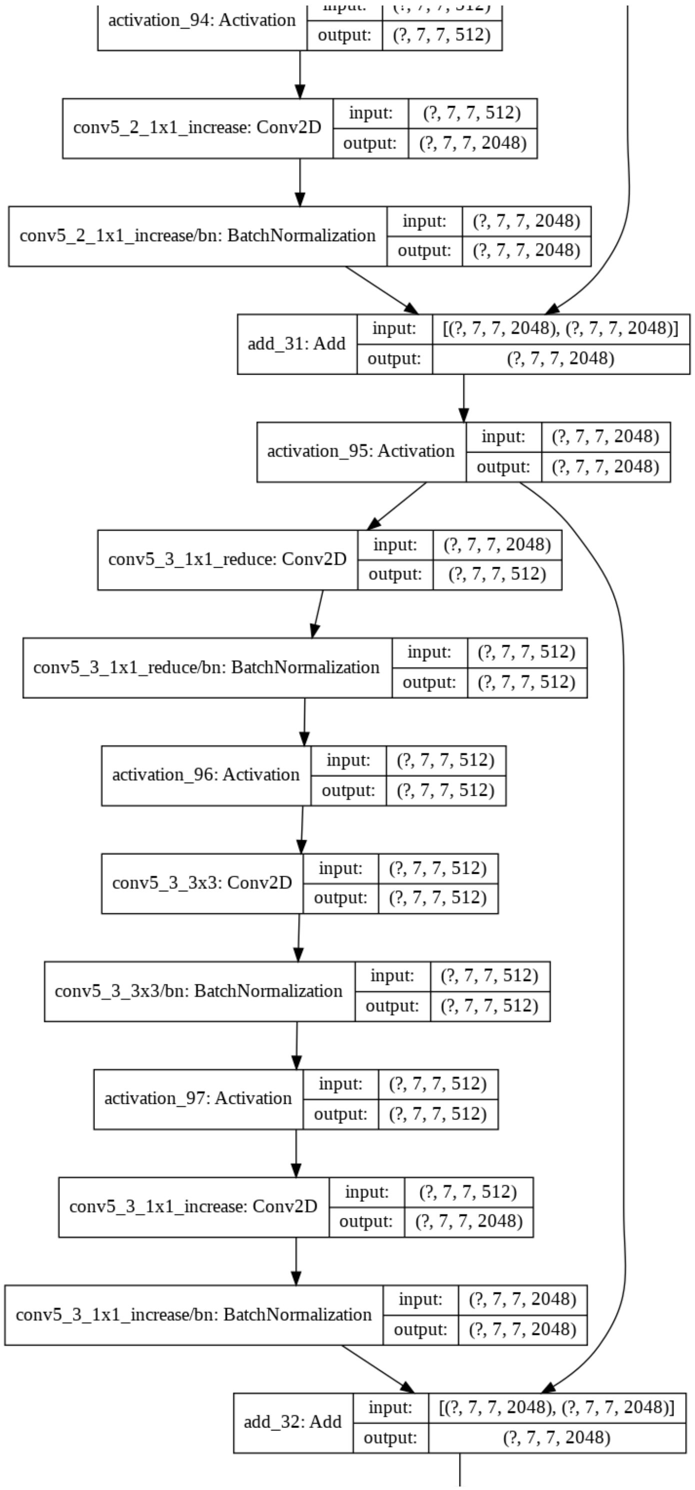
conv4_3_1x1_reduce: Conv2D
output: (?, 14, 14, 256)
conv4_3_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_78: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_3_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_3_3x3/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_79: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_3_1x1_increase: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 1024)
conv4_3_1x1_increase/bn: BatchNormalization
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
add_26: Add
input: [(?, 14, 14, 1024), (?, 14, 14, 1024)]
output: (?, 14, 14, 1024)
activation_80: Activation
input: (?, 14, 14, 1024)
output: (?, 14, 14, 1024)
conv4_4_1x1_reduce: Conv2D
input: (?, 14, 14, 1024)
output: (?, 14, 14, 256)
conv4_4_1x1_reduce/bn: BatchNormalization
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
activation_81: Activation
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)
conv4_4_3x3: Conv2D
input: (?, 14, 14, 256)
output: (?, 14, 14, 256)


conv4_4_3x3/bn: BatchNormalization | input: (?, 14, 14, 256) | output: (?, 14, 14, 256)
activation_82: Activation | input: (?, 14, 14, 256) | output: (?, 14, 14, 256)
conv4_4_1x1_increase: Conv2D | input: (?, 14, 14, 256) | output: (?, 14, 14, 1024)
conv4_4_1x1_increase/bn: BatchNormalization | input: (?, 14, 14, 1024) | output: (?, 14, 14, 1024)
add_27: Add | input: [(?, 14, 14, 1024), (?, 14, 14, 1024)] | output: (?, 14, 14, 1024)
activation_83: Activation | input: (?, 14, 14, 1024) | output: (?, 14, 14, 1024)
conv4_5_1x1_reduce: Conv2D | input: (?, 14, 14, 1024) | output: (?, 14, 14, 256)
conv4_5_1x1_reduce/bn: BatchNormalization | input: (?, 14, 14, 256) | output: (?, 14, 14, 256)
activation_84: Activation | input: (?, 14, 14, 256) | output: (?, 14, 14, 256)
conv4_5_3x3: Conv2D | input: (?, 14, 14, 256) | output: (?, 14, 14, 256)
conv4_5_3x3/bn: BatchNormalization | input: (?, 14, 14, 256) | output: (?, 14, 14, 256)
activation_85: Activation | input: (?, 14, 14, 256) | output: (?, 14, 14, 256)
conv4_5_1x1_increase: Conv2D | input: (?, 14, 14, 256) | output: (?, 14, 14, 1024)
conv4_5_1x1_increase/bn: BatchNormalization | input: (?, 14, 14, 1024) | output: (?, 14, 14, 1024)

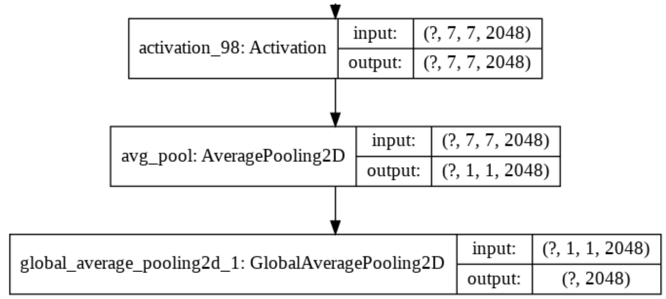

add_28: Add
input:
[(?, 14, 14, 1024), (?, 14, 14, 1024)]
output:
(?, 14, 14, 1024)
activation_86: Activation
input:
(?, 14, 14, 1024)
output:
(?, 14, 14, 1024)
conv4_6_1x1_reduce: Conv2D
input:
(?, 14, 14, 1024)
output:
(?, 14, 14, 256)
conv4_6_1x1_reduce/bn: BatchNormalization
input:
(?, 14, 14, 256)
output:
(?, 14, 14, 256)
activation_87: Activation
input:
(?, 14, 14, 256)
output:
(?, 14, 14, 256)
conv4_6_3x3: Conv2D
input:
(?, 14, 14, 256)
output:
(?, 14, 14, 256)
conv4_6_3x3/bn: BatchNormalization
input:
(?, 14, 14, 256)
output:
(?, 14, 14, 256)
activation_88: Activation
input:
(?, 14, 14, 256)
output:
(?, 14, 14, 256)
conv4_6_1x1_increase: Conv2D
input:
(?, 14, 14, 256)
output:
(?, 14, 14, 1024)
conv4_6_1x1_increase/bn: BatchNormalization
input:
(?, 14, 14, 1024)
output:
(?, 14, 14, 1024)
add_29: Add
input:
[(?, 14, 14, 1024), (?, 14, 14, 1024)]
output:
(?, 14, 14, 1024)
activation_89: Activation
input:
(?, 14, 14, 1024)
output:
(?, 14, 14, 1024)
conv5_1_1x1_reduce: Conv2D
input:
(?, 14, 14, 1024)
output:
(?, 7, 7, 512)
conv5_1_1x1_reduce/bn: BatchNormalization
input:
(?, 7, 7, 512)
conv5_1_1x1_proj: Conv2D
input:
(?, 14, 14, 1024)



output: (?, 7, 7, 512)
output: (?, 7, 7, 2048)
activation_90: Activation
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_1_1x1_proj/bn: BatchNormalization
input: (?, 7, 7, 2048)
output: (?, 7, 7, 2048)
conv5_1_3x3: Conv2D
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_1_3x3/bn: BatchNormalization
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
activation_91: Activation
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_1_1x1_increase: Conv2D
input: (?, 7, 7, 512)
output: (?, 7, 7, 2048)
conv5_1_1x1_increase/bn: BatchNormalization
input: (?, 7, 7, 2048)
output: (?, 7, 7, 2048)
add_30: Add
input: [(?, 7, 7, 2048), (?, 7, 7, 2048)]
output: (?, 7, 7, 2048)
activation_92: Activation
input: (?, 7, 7, 2048)
output: (?, 7, 7, 2048)
conv5_2_1x1_reduce: Conv2D
input: (?, 7, 7, 2048)
output: (?, 7, 7, 512)
conv5_2_1x1_reduce/bn: BatchNormalization
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
activation_93: Activation
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_2_3x3: Conv2D
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
conv5_2_3x3/bn: BatchNormalization
input: (?, 7, 7, 512)
output: (?, 7, 7, 512)
input: (?, 7, 7, 512)



activation_94: Activation
input:
output:
(?, 7, 7, 512)
conv5_2_1x1_increase: Conv2D
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 2048)
conv5_2_1x1_increase/bn: BatchNormalization
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)
add_31: Add
input:
[(?, 7, 7, 2048), (?, 7, 7, 2048)]
output:
(?, 7, 7, 2048)
activation_95: Activation
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)
conv5_3_1x1_reduce: Conv2D
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 512)
conv5_3_1x1_reduce/bn: BatchNormalization
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
activation_96: Activation
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_3_3x3: Conv2D
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_3_3x3/bn: BatchNormalization
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
activation_97: Activation
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 512)
conv5_3_1x1_increase: Conv2D
input:
(?, 7, 7, 512)
output:
(?, 7, 7, 2048)
conv5_3_1x1_increase/bn: BatchNormalization
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)
add_32: Add
input:
[(?, 7, 7, 2048), (?, 7, 7, 2048)]
output:
(?, 7, 7, 2048)


activation_98: Activation
input:
(?, 7, 7, 2048)
output:
(?, 7, 7, 2048)
avg_pool: AveragePooling2D
input:
(?, 7, 7, 2048)
output:
(?, 1, 1, 2048)
global_average_pooling2d_1: GlobalAveragePooling2D
input:
(?, 1, 1, 2048)
output:
(?, 2048)

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

A VGGFace2 model calculates a face embedding, a vector that represents the features extracted from the face. This can then be compared with the vectors generated for other faces. For example, another vector that is close (by some measure) may be the same person, whereas another vector that is far (by some measure) may be a different person.

Typical measures such as Euclidean distance and Cosine distance are calculated between two embeddings and faces are said to match or verify if the distance is below a predefined threshold, often tuned for a specific dataset or application.

The face embedding can be calculated by calling `model_name.predict(pre_processed_image)`. We integrate image processing and face embedding in one function 'get_embeddings(filenames, model)'. In this function, the name of the NN model is 'model'.

Exercise: finish the function and call the function with correct NN model.

```
# extract faces and calculate face embeddings for a list of photo files
def get_embeddings(filenames,model):
    # extract faces
    faces = [extract_face(f) for f in filenames]
    # convert into an array of samples
    samples = asarray(faces, 'float32')
    # prepare the face for the model, e.g. center pixels
    samples = preprocess_input(samples, version=2)
    # perform prediction

  ## start your code here
    yhat = model_notop.predict(samples)
  ## end your code here

    return yhat

# start your code here
embedding_test = get_embeddings(['/content/drive/MyDrive/DL/Homework11/face verification/images/sharon_stone3.jpg'], model_n
# end your code here
print(embedding_test.shape)
```

```
(1, 2048)
```

If two embeddings extracted from two faces are similar to each other, we say the two faces are for the same person. Here, we use cosine similarity.

```
# determine if a candidate face is a match for a known face
def is_match(known_embedding, candidate_embedding, thresh=0.5):
    # calculate distance between embeddings
    score = cosine(known_embedding, candidate_embedding)
    if score <= thresh:
        print('>face is a Match (%.3f <= %.3f)' % (score, thresh))
    else:
        print('>face is NOT a Match (%.3f > %.3f)' % (score, thresh))

# Get the face embedding of sharon
sharon_id=get_embeddings(['/content/drive/MyDrive/DL/Homework11/face verification/images/sharon_stone3.jpg'],model_notop)
# Get the face embedding of another image
# start your code here
sharon_cand=get_embeddings(['/content/drive/MyDrive/DL/Homework11/face verification/images/sharon_stone1.jpg'], model_notop)
is_match(sharon_id,sharon_cand) # call the is_match function to see whether the image you choose is Sharon Stone.
# end your code here
```

```
>face is a Match (0.300 <= 0.500)
```

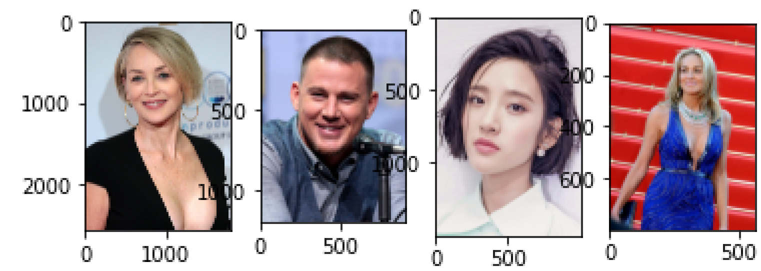
We can do this in bulk. Get the embeddings of a bunch of pictures and to see whether the first image matches the persons in the other images.

Exercise: you can try your own pictures to test how well the VGGFace model is.

I find the VGGFace model is not quite effective on asian faces. Do you have any idea to improve it?

```
# define filenames, check whether the person in the first image is the same person that appears in the other images.

# start your code here
filenames = ['/content/drive/MyDrive/DL/Homework11/face verification/images/sharon_stone3.jpg',
```

```
             '/content/drive/MyDrive/DL/Homework11/face verification/images/channing_tatum.jpg',
             '/content/drive/MyDrive/DL/Homework11/face verification/images/yixin_tang.jpg',
             '/content/drive/MyDrive/DL/Homework11/face verification/images/sharon_stone1.jpg'] # you can download the images
# end your code here

#filenames =['images/sharon_stone1.jpg','images/sharon_stone2.jpg','images/sharon_stone3.jpg','images/li_gong1.jpg']

fig = pyplot.figure()
# plot the original image
ax = fig.add_subplot(1, 4, 1)
imgplot = pyplot.imshow(mpimg.imread(filenames[0]))

ax = fig.add_subplot(1, 4, 2)
imgplot = pyplot.imshow(mpimg.imread(filenames[1]))

ax = fig.add_subplot(1, 4, 3)
imgplot = pyplot.imshow(mpimg.imread(filenames[2]))

ax = fig.add_subplot(1, 4, 4)
imgplot = pyplot.imshow(mpimg.imread(filenames[3]))

# show the plot
pyplot.show()

# get embeddings file filenames
embeddings = get_embeddings(filenames,model_notop)
# define sharon stone
sharon_id = embeddings[0]
# verify known photos of Gong-Li
print('image 1 vs image 2:')
is_match(embeddings[0], embeddings[1])
print('image 1 vs image 3:')
is_match(embeddings[0], embeddings[2])
# verify known photos of other people
print('image 1 vs image 4:')
is_match(embeddings[0], embeddings[3])
```




```
image 1 vs image 2:
>face is NOT a Match (0.768 > 0.500)
image 1 vs image 3:
>face is NOT a Match (0.745 > 0.500)
image 1 vs image 4:
>face is a Match (0.300 <= 0.500)
```

40s completed at 10:58 PM