```
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_remo
```

Which super star do you look like the most?

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 super stars from all over the world.

To earn the credits of this project, you will need to

1. highlight the resnet structures in the VGGFace model after you print the project out.
2. get a final output showing the possibility that you might be the superstar(s).

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.

```
%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
```

If the version says

```
H5PY version: 2.10.0

TensorFlow version: 1.15.2
```

skip the next cell.

Otherwise,

1. run the cell below,
2. restart the runtime (ctrl+M)
3. Rerun all the cells above
4. check the version requirement again. If satsified, skip the cell below.

```
!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)? n
     Requirement already satisfied: h5py==2.10.0 in /usr/local/lib/python3.7/dist-packages (2.10.0)
     Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from h5py==2.10.0) (1.15.0)
     Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.7/dist-packages (from h5py==2.10.0) (1.21.5)
```

Next, we will install face recognition pacakages for this project.

```
!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-0639eu0f
  Running command git clone -q https://github.com/rcmalli/keras-vggface.git /tmp/pip-req-build-0639eu0f
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-packages (from keras-vggface==0.6) (1.21.
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.
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: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from keras_preprocessing) (1.15.0
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/dist-packages (from keras_preprocessing) (1.21
Requirement already satisfied: mtcnn in /usr/local/lib/python3.7/dist-packages (0.1.1)
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: scipy>=0.14 in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (1.4.1
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: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0
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: h5py in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (2.10.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (3.13)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from keras>=2.0.0->mtcnn) (1.15.0
```

We then import the packages here.

```
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
```

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

Please make sure that your package versions are as follows.

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 “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks.”

```
# 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/Homework9/thumbnail_rbh17b_2.jpg' # you need to change the path to the path of your ow
## 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()
```

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.

```
# create a vggface model
model = VGGFace(model='resnet50')
```

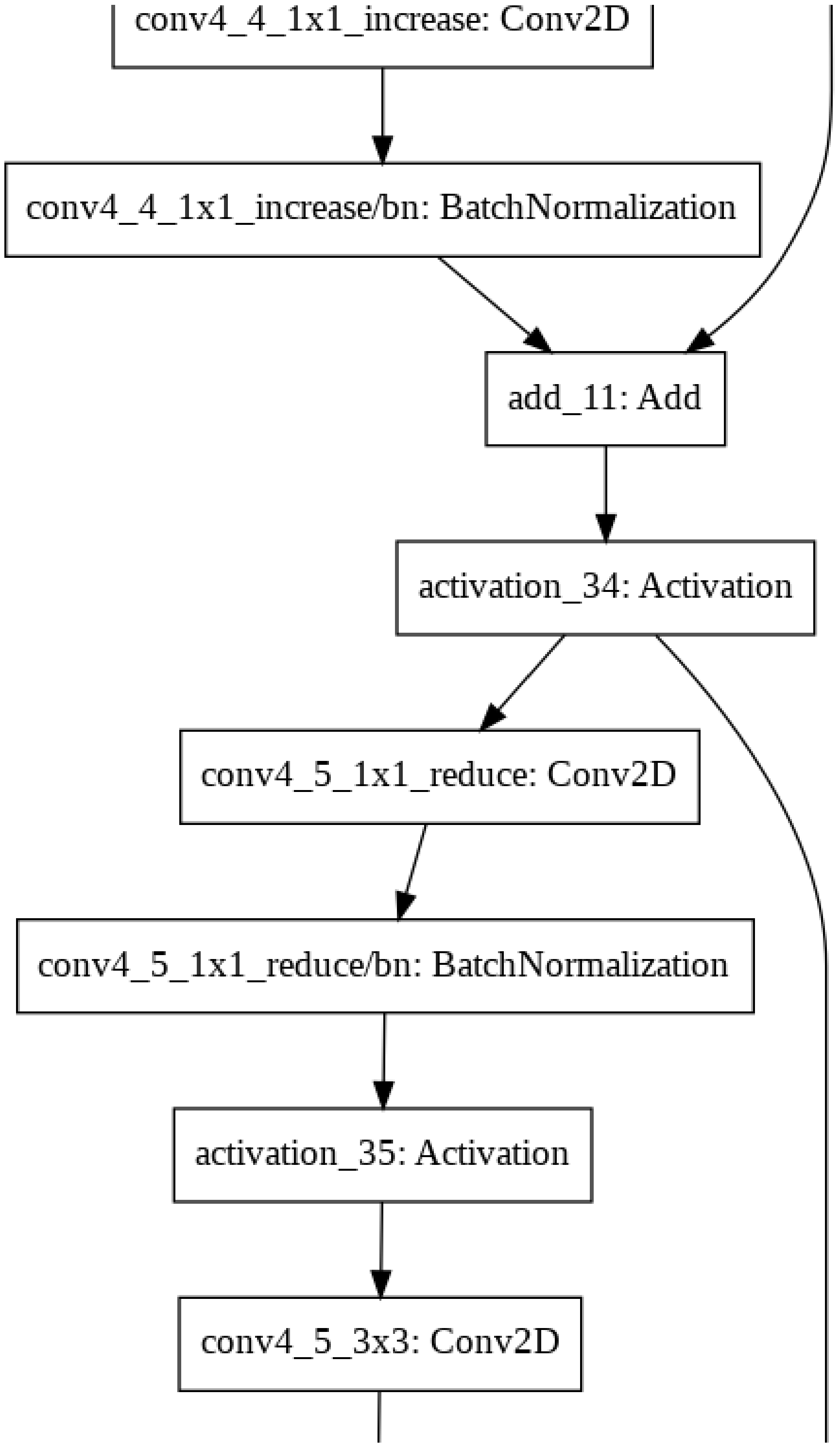
```
WARNING:tensorflow:From /tensorflow-1.15.2/python3.7/keras/backend/tensorflow_backend.py:4070: The name tf.nn.max_pool

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

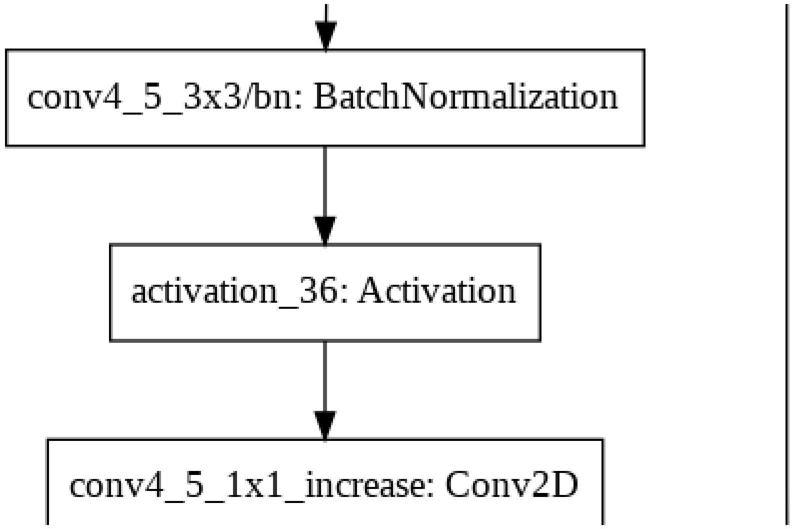
If printing out the NN structure, we can observe the apparant 'resnet' structure (looks like a jump from bottom layers to several layers above).

Exercise: Can you find the 'resnet' structure (jumps) in the VGGFace model?

***After you print out your output, please highlight the 'resnet' structure in the model to earn points. ***

```
tf.keras.utils.plot_model(model, show_shapes=False)
```


conv4_4_1x1_increase: Conv2D
conv4_4_1x1_increase/bn: BatchNormalization
add_11: Add
activation_34: Activation
conv4_5_1x1_reduce: Conv2D
conv4_5_1x1_reduce/bn: BatchNormalization
activation_35: Activation
conv4_5_3x3: Conv2D


conv4_5_3x3/bn: BatchNormalization
activation_36: Activation
conv4_5_1x1_increase: Conv2D