In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries
 installed
# It is defined by the kaggle/python docker image: https://github.com/k
aggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np
import pandas as pd
import gc
import cv2
import glob

from collections import defaultdict


import time
from keras.optimizers import Adam
from keras.layers import Conv2D, ZeroPadding2D, Activation, Input, conc
atenate
from keras.models import Model
import seaborn as sns
from keras.layers.normalization import BatchNormalization
from keras.layers.pooling import MaxPooling2D, AveragePooling2D
from keras.layers.merge import Concatenate
from keras.layers.core import Lambda, Flatten, Dense
from keras.initializers import glorot_uniform
from sklearn.preprocessing import LabelBinarizer
from keras.optimizers import *
from keras.engine.topology import Layer
from keras import backend as K
from keras.regularizers import l2
K.set_image_data_format('channels_last')
import os
from numpy import genfromtxt
import tensorflow as tf
#from fr_utils import *
```

```
#from inception_blocks_v2 import *
import numpy.random as rng
from sklearn.utils import shuffle
from keras.models import Sequential

from sklearn.metrics import roc_auc_score
from keras.callbacks import ReduceLROnPlateau, ModelCheckpoint, EarlySt
opping
import threading
from tqdm import tqdm

print(os.listdir("../input"))
```

```
Using TensorFlow backend.
```

```
['train', 'test', 'train_relationships.csv', 'sample_submission.csv']
```

In [2]:
```
!pip install git+https://github.com/rcmalli/keras-vggface.git
```

```
Collecting git+https://github.com/rcmalli/keras-vggface.git
  Cloning https://github.com/rcmalli/keras-vggface.git to /tmp/pip-req-
build-3584jlbl
  Running command git clone -q https://github.com/rcmalli/keras-vggfac
e.git /tmp/pip-req-build-3584jlbl
Requirement already satisfied: numpy>=1.9.1 in /opt/conda/lib/python3.
6/site-packages (from keras-vggface==0.6) (1.16.4)
Requirement already satisfied: scipy>=0.14 in /opt/conda/lib/python3.6/
site-packages (from keras-vggface==0.6) (1.2.1)
Requirement already satisfied: h5py in /opt/conda/lib/python3.6/site-pa
ckages (from keras-vggface==0.6) (2.9.0)
Requirement already satisfied: pillow in /opt/conda/lib/python3.6/site-
packages (from keras-vggface==0.6) (5.4.1)
Requirement already satisfied: keras in /opt/conda/lib/python3.6/site-p
ackages (from keras-vggface==0.6) (2.2.4)
Requirement already satisfied: six>=1.9.0 in /opt/conda/lib/python3.6/s
ite-packages (from keras-vggface==0.6) (1.12.0)
Requirement already satisfied: pyyaml in /opt/conda/lib/python3.6/site-
packages (from keras-vggface==0.6) (5.1.1)
Requirement already satisfied: keras-applications>=1.0.6 in /opt/conda/
lib/python3.6/site-packages (from keras->keras-vggface==0.6) (1.0.8)
```

```
Requirement already satisfied: keras-preprocessing>=1.0.5 in /opt/cond
a/lib/python3.6/site-packages (from keras->keras-vggface==0.6) (1.1.0)
Building wheels for collected packages: keras-vggface
  Building wheel for keras-vggface (setup.py) ... -\ |done
  Created wheel for keras-vggface: filename=keras_vggface-0.6-cp36-none
-any.whl size=8311 sha256=bdf7bc657b9de2740be759c95a86ab55d5b1b77f0551e
47d31b54828930e074c
  Stored in directory: /tmp/pip-ephem-wheel-cache-o9eda2xx/wheels/36/0
7/46/06c25ce8e9cd396dabe151ea1d8a2bc28dafcb11321c1f3a6d
Successfully built keras-vggface
Installing collected packages: keras-vggface
Successfully installed keras-vggface-0.6
```

In [3]:
```
train_file_path = "../input/train_relationships.csv"
train_folders_path = "../input/train/"
```

In [4]:
```
from keras_vggface.utils import preprocess_input
from keras_vggface.vggface import VGGFace
def read_img(path):
    img = cv2.imread(path)
    img = np.array(img).astype(np.float)
    return preprocess_input(img,version=2)
```

In [5]:
```
#keeps all photos path in a dictionary
allPhotos = defaultdict(list)
for family in glob.glob(train_folders_path+"/*"):
    for mem in glob.glob(family+'/*'):
        for photo in glob.glob(mem+'/*'):
            allPhotos[mem].append(photo)

#list of all members with valid photo
ppl = list(allPhotos.keys())
```

In [6]:
```
#few valid members with valid photo
ppl[-5:]
```

Out[6]:
```
['../input/train/F0670/MID1',
 '../input/train/F0670/MID2',
```

```
 '../input/train/F0470/MID4',
 '../input/train/F0470/MID3',
 '../input/train/F0470/MID2']
```

In [7]:
```
#getting all the photos of this member
allPhotos['../input/train/F0470/MID2']
```

Out[7]:
```
['../input/train/F0470/MID2/P04949_face1.jpg',
 '../input/train/F0470/MID2/P04957_face1.jpg',
 '../input/train/F0470/MID2/P04950_face1.jpg',
 '../input/train/F0470/MID2/P04956_face1.jpg',
 '../input/train/F0470/MID2/P04955_face1.jpg',
 '../input/train/F0470/MID2/P04952_face1.jpg']
```

In [8]:
```
data = pd.read_csv('../input/train_relationships.csv')
data.p1 = data.p1.apply( lambda x: '../input/train/'+x )
data.p2 = data.p2.apply( lambda x: '../input/train/'+x )
data.head()
```

Out[8]:

	p1	p2
0	../input/train/F0002/MID1	../input/train/F0002/MID3
1	../input/train/F0002/MID2	../input/train/F0002/MID3
2	../input/train/F0005/MID1	../input/train/F0005/MID2
3	../input/train/F0005/MID3	../input/train/F0005/MID2
4	../input/train/F0009/MID1	../input/train/F0009/MID4

In [9]:
```
# creating tuples of images which are there in train images
data = data[((data.p1.isin(ppl)) & (data.p2.isin(ppl)))]
data = [( x[0], x[1]) for x in data.values ]
print(len(data))
```

```
3362
```

In [10]:
```
#Splitting data into train and validation
```

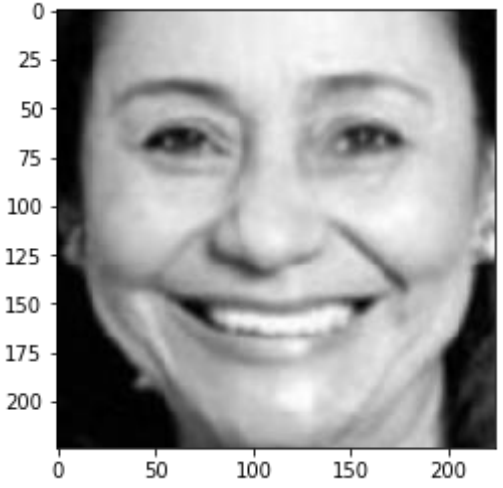
```
train = [ x for x in data if 'F09' not in x[0]]
val = [ x for x in data if 'F09' in x[0]]
print("Images in training data",len(train))
print("Images in test data",len(val))
```

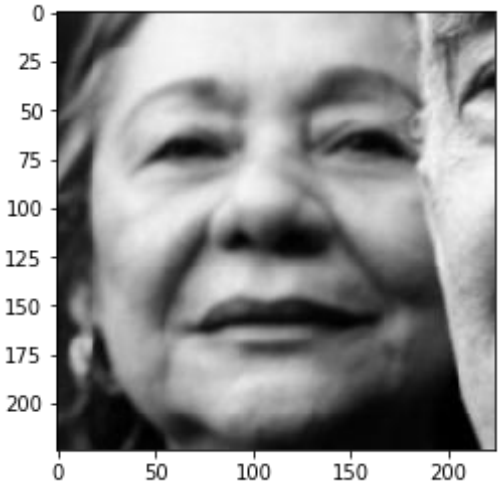
```
Images in training data 3066
Images in test data 296
```

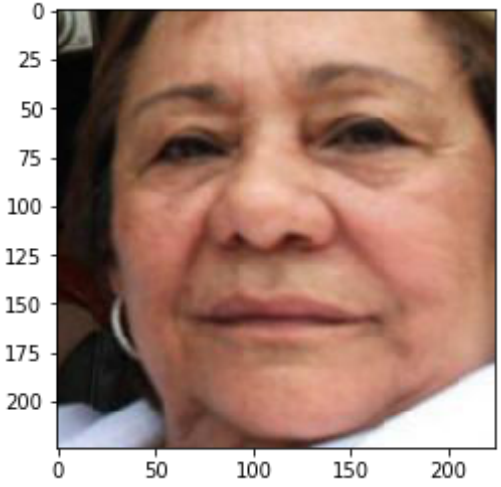
In [11]:
```
# visualizing some photos of the family member
%pylab inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
for i in allPhotos['../input/train/F0470/MID2']:
    img=mpimg.imread(i)
    imgplot = plt.imshow(img)
    plt.show()
```

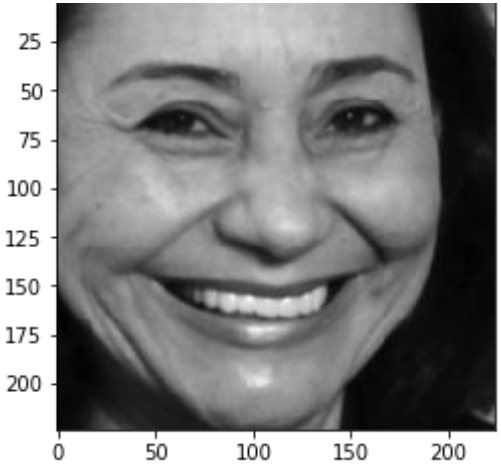
```
Populating the interactive namespace from numpy and matplotlib
```

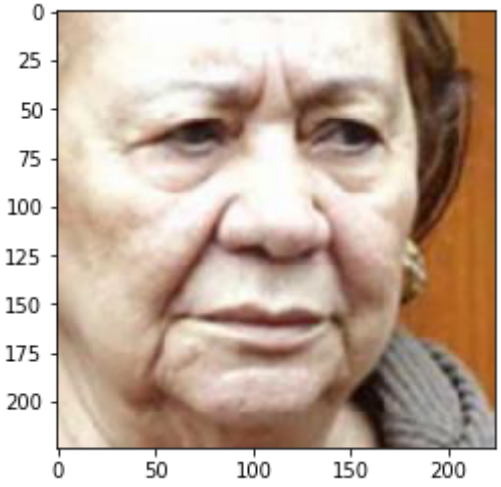
```
/opt/conda/lib/python3.6/site-packages/IPython/core/magics/pylab.py:16
0: UserWarning: pylab import has clobbered these variables: ['concatena
te', 'shuffle', 'copy', 'get']
`%matplotlib` prevents importing * from pylab and numpy
  "\n`%matplotlib` prevents importing * from pylab and numpy"
```

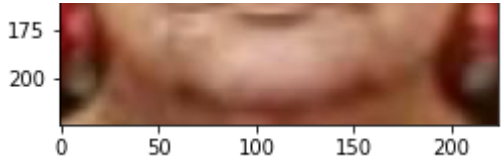












In [12]:
```
#Creating batch of images
from random import choice, sample
def getImages(p1,p2):
    p1 = read_img(choice(allPhotos[p1]))
    p2 = read_img(choice(allPhotos[p2]))
    return p1,p2

def getMiniBatch(batch_size=16, data=train):
    p1 = []; p2 = []; Y = []
    batch = sample(data, batch_size//2)
    for x in batch:
        _p1, _p2 = getImages(*x)
        p1.append(_p1);p2.append(_p2);Y.append(1)
    while len(Y) < batch_size:
        _p1,_p2 = tuple(np.random.choice(ppl,size=2, replace=False))
        if (_p1,_p2) not in train+val and (_p2,_p1) not in train+val:
            _p1,_p2 = getImages(_p1,_p2)
            p1.append(_p1);p2.append(_p2);Y.append(0)
    return [np.array(p1),np.array(p2)], np.array(Y)
```

In [13]:
```
def initialize_bias(shape, name=None):
    """
        The paper, http://www.cs.utoronto.ca/~gkoch/files/msc-thesis.pd
f
        suggests to initialize CNN layer bias with mean as 0.5 and stan
dard deviation of 0.01
    """
    return np.random.normal(loc = 0.5, scale = 1e-2, size = shape)
```

```
def initialize_weights(shape, name=None):
    """
        The paper, http://www.cs.utoronto.ca/~gkoch/files/msc-thesis.pd
f
        suggests to initialize CNN layer weights with mean as 0.0 and s
tandard deviation of 0.01
    """
    return np.random.normal(loc = 0.0, scale = 1e-2, size = shape)
```

In [14]:
```
def auc(y_true, y_pred):
    return tf.py_func(roc_auc_score, (y_true, y_pred), tf.double)
```

In [15]:
```
from keras.layers import Input, Dense, GlobalMaxPool2D, GlobalAvgPool2D
, Concatenate, Multiply, Dropout, Subtract
def baseline_model():
    input_1 = Input(shape=(224, 224, 3))
    input_2 = Input(shape=(224, 224, 3))

    base_model = VGGFace(model='resnet50', include_top=False)

    for x in base_model.layers[:-3]:
        x.trainable = True
    for x in base_model.layers[-3:]:
        x.trainable=False

    x1 = base_model(input_1)
    x2 = base_model(input_2)

#     x1_ = Reshape(target_shape=(7*7, 2048))(x1)
#     x2_ = Reshape(target_shape=(7*7, 2048))(x2)
#     #
#     x_dot = Dot(axes=[2, 2], normalize=True)([x1_, x2_])
#     x_dot = Flatten()(x_dot)

    x1 = Concatenate(axis=-1)([GlobalMaxPool2D()(x1), GlobalAvgPool2D()
(x1)])
    x2 = Concatenate(axis=-1)([GlobalMaxPool2D()(x2), GlobalAvgPool2D()
```

```
(x2)])

    x3 = Subtract()([x1, x2])
    x3 = Multiply()([x3, x3])

    x1_ = Multiply()([x1, x1])
    x2_ = Multiply()([x2, x2])
    x4 = Subtract()([x1_, x2_])
    x = Concatenate(axis=-1)([x4, x3])

    x = Dense(100, activation="relu")(x)
    x = Dropout(0.01)(x)
    out = Dense(1, activation="sigmoid")(x)

    model = Model([input_1, input_2], out)

    # loss="binary_crossentropy"
    model.compile(loss="binary_crossentropy",optimizer=Adam(1e-5),metri
cs=['accuracy',auc])

    model.summary()

    return model
```

In [16]:
```
model = baseline_model()
```

```
Downloading data from https://github.com/rcmalli/keras-vggface/release
s/download/v2.0/rcmalli_vggface_tf_notop_resnet50.h5
94699520/94694792 [==============================] - 1s 0us/step
_________________________________________________________________
Layer (type)                 Output Shape              Param #     Connec
ted to
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)  0

_________________________________________________________________
```

```
input_2 (InputLayer)            (None, 224, 224, 3)  0

__________________________________________________________________________
_____
vggface_resnet50 (Model)        multiple             23561152    input_
1[0][0]
                                                                 input_
2[0][0]
__________________________________________________________________________
_____
global_max_pooling2d_1 (GlobalM (None, 2048)         0           vggfac
e_resnet50[1][0]
__________________________________________________________________________
_____
global_average_pooling2d_1 (Glo (None, 2048)         0           vggfac
e_resnet50[1][0]
__________________________________________________________________________
_____
global_max_pooling2d_2 (GlobalM (None, 2048)         0           vggfac
e_resnet50[2][0]
__________________________________________________________________________
_____
global_average_pooling2d_2 (Glo (None, 2048)         0           vggfac
e_resnet50[2][0]
__________________________________________________________________________
_____
concatenate_1 (Concatenate)     (None, 4096)         0           global
_max_pooling2d_1[0][0]
                                                                 global
_average_pooling2d_1[0][0]
__________________________________________________________________________
_____
concatenate_2 (Concatenate)     (None, 4096)         0           global
_max_pooling2d_2[0][0]
                                                                 global
_average_pooling2d_2[0][0]
__________________________________________________________________________
_____
multiply_2 (Multiply)           (None, 4096)         0           concat
```

```
enate_1[0][0]
                                                                 concat
enate_1[0][0]
__________________________________________________________________________
__________________________________
multiply_3 (Multiply)           (None, 4096)         0           concat
enate_2[0][0]
                                                                 concat
enate_2[0][0]
__________________________________________________________________________
__________________________________
subtract_1 (Subtract)           (None, 4096)         0           concat
enate_1[0][0]
                                                                 concat
enate_2[0][0]
__________________________________________________________________________
__________________________________
subtract_2 (Subtract)           (None, 4096)         0           multip
ly_2[0][0]
                                                                 multip
ly_3[0][0]
__________________________________________________________________________
__________________________________
multiply_1 (Multiply)           (None, 4096)         0           subtra
ct_1[0][0]
                                                                 subtra
ct_1[0][0]
__________________________________________________________________________
__________________________________
concatenate_3 (Concatenate)     (None, 8192)         0           subtra
ct_2[0][0]
                                                                 multip
ly_1[0][0]
__________________________________________________________________________
__________________________________
dense_1 (Dense)                 (None, 100)          819300      concat
enate_3[0][0]
__________________________________________________________________________
__________________________________
```

```
dropout_1 (Dropout)             (None, 100)          0           dense_
1[0][0]
_________________________________________________________________
_________________________
dense_2 (Dense)                 (None, 1)            101         dropou
t_1[0][0]
=================================================================
==========================
Total params: 24,380,553
Trainable params: 24,327,433
Non-trainable params: 53,120
_________________________________________________________________
_________________________
```

In [17]:

```
reducelr = ReduceLROnPlateau(monitor='val_auc', mode='max',patience=6,f
actor=0.1,verbose=1)

model_checkpoint  = ModelCheckpoint('model_best_checkpoint.h5', save_be
st_only=True,
                                    save_weights_only=True, monitor='va
l_auc', mode='max', verbose=1)

early_stopping = EarlyStopping(monitor='val_auc', patience=15, mode='ma
x')

callbacks_list = [reducelr, model_checkpoint, early_stopping]

def Generator(batch_size,data):
    while True:
        yield getMiniBatch(batch_size=batch_size, data=data)

train_gen = Generator(batch_size=16,data=train)
val_gen = Generator(batch_size=16,data=val)
model.fit_generator( train_gen, samples_per_epoch=100, epochs=50,
                            validation_data=val_gen, validation_steps=100
, use_multiprocessing=True,
                            verbose=1, workers=4)
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:18: UserWa
rning: Update your `fit_generator` call to the Keras 2 API: `fit_genera
```

```
tor(<generator..., epochs=50, validation_data=<generator..., validation

_steps=100, use_multiprocessing=True, verbose=1, workers=4, steps_per_e
poch=100)`
```

```
Epoch 1/50
```

```
/opt/conda/lib/python3.6/site-packages/keras/engine/training_generator.
py:47: UserWarning: Using a generator with `use_multiprocessing=True` a
nd multiple workers may duplicate your data. Please consider using the`
keras.utils.Sequence class.
  UserWarning('Using a generator with `use_multiprocessing=True`'
```

```
100/100 [==============================] - 67s 672ms/step - loss: 4.312
8 - acc: 0.5763 - auc: 0.6099 - val_loss: 3.2481 - val_acc: 0.6994 - va
l_auc: 0.7679
Epoch 2/50
100/100 [==============================] - 41s 413ms/step - loss: 3.927
5 - acc: 0.6063 - auc: 0.6445 - val_loss: 3.2864 - val_acc: 0.6869 - va
l_auc: 0.7434
Epoch 3/50
100/100 [==============================] - 42s 421ms/step - loss: 3.242
7 - acc: 0.6581 - auc: 0.6870 - val_loss: 3.4857 - val_acc: 0.6706 - va
l_auc: 0.7164
Epoch 4/50
100/100 [==============================] - 40s 404ms/step - loss: 3.014
6 - acc: 0.6300 - auc: 0.6723 - val_loss: 3.0205 - val_acc: 0.6794 - va
l_auc: 0.7406
Epoch 5/50
100/100 [==============================] - 40s 404ms/step - loss: 2.275
7 - acc: 0.6587 - auc: 0.7064 - val_loss: 2.5122 - val_acc: 0.6450 - va
l_auc: 0.7133
Epoch 6/50
100/100 [==============================] - 40s 402ms/step - loss: 1.714
9 - acc: 0.6544 - auc: 0.7047 - val_loss: 1.7460 - val_acc: 0.6700 - va
l_auc: 0.7568
Epoch 7/50
100/100 [==============================] - 40s 403ms/step - loss: 1.270
7 - acc: 0.6469 - auc: 0.7030 - val_loss: 1.6161 - val_acc: 0.6450 - va
l_auc: 0.7098
```

```
Epoch 8/50
100/100 [==============================] - 40s 405ms/step - loss: 1.040
4 - acc: 0.6437 - auc: 0.7006 - val_loss: 1.2338 - val_acc: 0.6394 - va
l_auc: 0.7029
Epoch 9/50
100/100 [==============================] - 40s 398ms/step - loss: 0.859
3 - acc: 0.6813 - auc: 0.7430 - val_loss: 1.0884 - val_acc: 0.6519 - va
l_auc: 0.7084
Epoch 10/50
100/100 [==============================] - 40s 403ms/step - loss: 0.787
6 - acc: 0.6531 - auc: 0.7103 - val_loss: 1.0221 - val_acc: 0.6331 - va
l_auc: 0.6769
Epoch 11/50
100/100 [==============================] - 41s 409ms/step - loss: 0.681
6 - acc: 0.7000 - auc: 0.7591 - val_loss: 0.7991 - val_acc: 0.6875 - va
l_auc: 0.7495
Epoch 12/50
100/100 [==============================] - 40s 401ms/step - loss: 0.615
9 - acc: 0.7156 - auc: 0.7766 - val_loss: 0.8174 - val_acc: 0.6694 - va
l_auc: 0.7305
Epoch 13/50
100/100 [==============================] - 41s 408ms/step - loss: 0.691
4 - acc: 0.6506 - auc: 0.7203 - val_loss: 0.7854 - val_acc: 0.6519 - va
l_auc: 0.7123
Epoch 14/50
100/100 [==============================] - 40s 402ms/step - loss: 0.571
5 - acc: 0.7094 - auc: 0.7783 - val_loss: 0.7000 - val_acc: 0.6713 - va
l_auc: 0.7461
Epoch 15/50
100/100 [==============================] - 40s 404ms/step - loss: 0.585
1 - acc: 0.7094 - auc: 0.7758 - val_loss: 0.7056 - val_acc: 0.6806 - va
l_auc: 0.7475
Epoch 16/50
100/100 [==============================] - 40s 403ms/step - loss: 0.578
5 - acc: 0.7006 - auc: 0.7761 - val_loss: 0.6939 - val_acc: 0.6787 - va
l_auc: 0.7430
Epoch 17/50
100/100 [==============================] - 40s 401ms/step - loss: 0.603
8 - acc: 0.6900 - auc: 0.7589 - val_loss: 0.6492 - val_acc: 0.6769 - va
l_auc: 0.7567
```

```
l_auc: 0.7567
Epoch 18/50

100/100 [==============================] - 41s 407ms/step - loss: 0.515
8 - acc: 0.7469 - auc: 0.8130 - val_loss: 0.6042 - val_acc: 0.7006 - va
l_auc: 0.7937
Epoch 19/50
100/100 [==============================] - 40s 401ms/step - loss: 0.530
3 - acc: 0.7319 - auc: 0.8094 - val_loss: 0.6429 - val_acc: 0.7006 - va
l_auc: 0.7712
Epoch 20/50
100/100 [==============================] - 40s 403ms/step - loss: 0.535
6 - acc: 0.7313 - auc: 0.8097 - val_loss: 0.5878 - val_acc: 0.7137 - va
l_auc: 0.7934
Epoch 21/50
100/100 [==============================] - 40s 405ms/step - loss: 0.502
5 - acc: 0.7475 - auc: 0.8303 - val_loss: 0.5897 - val_acc: 0.7069 - va
l_auc: 0.7947
Epoch 22/50
100/100 [==============================] - 40s 403ms/step - loss: 0.485
6 - acc: 0.7619 - auc: 0.8403 - val_loss: 0.5852 - val_acc: 0.7281 - va
l_auc: 0.8070
Epoch 23/50
100/100 [==============================] - 40s 402ms/step - loss: 0.524
9 - acc: 0.7438 - auc: 0.8158 - val_loss: 0.6422 - val_acc: 0.6800 - va
l_auc: 0.7608
Epoch 24/50
100/100 [==============================] - 40s 400ms/step - loss: 0.503
5 - acc: 0.7419 - auc: 0.8228 - val_loss: 0.6370 - val_acc: 0.6756 - va
l_auc: 0.7711
Epoch 25/50
100/100 [==============================] - 40s 400ms/step - loss: 0.445
6 - acc: 0.7944 - auc: 0.8727 - val_loss: 0.5522 - val_acc: 0.7438 - va
l_auc: 0.8341
Epoch 26/50
100/100 [==============================] - 40s 403ms/step - loss: 0.435
4 - acc: 0.7919 - auc: 0.8720 - val_loss: 0.5639 - val_acc: 0.7306 - va
l_auc: 0.8273
Epoch 27/50
100/100 [==============================] - 40s 400ms/step - loss: 0.470
8 - acc: 0.7719 - auc: 0.8408 - val_loss: 0.5871 - val_acc: 0.7262 - va
```

```
8 - acc: 0.7719 - auc: 0.8498 - val_loss: 0.5871 - val_acc: 0.7262 - va
l_auc: 0.8159

Epoch 28/50
100/100 [==============================] - 40s 402ms/step - loss: 0.476
6 - acc: 0.7800 - auc: 0.8484 - val_loss: 0.5486 - val_acc: 0.7262 - va
l_auc: 0.8256
Epoch 29/50
100/100 [==============================] - 40s 399ms/step - loss: 0.486
5 - acc: 0.7581 - auc: 0.8377 - val_loss: 0.5813 - val_acc: 0.7125 - va
l_auc: 0.8130
Epoch 30/50
100/100 [==============================] - 40s 399ms/step - loss: 0.445
5 - acc: 0.7850 - auc: 0.8686 - val_loss: 0.5941 - val_acc: 0.7044 - va
l_auc: 0.8041
Epoch 31/50
100/100 [==============================] - 40s 402ms/step - loss: 0.433
4 - acc: 0.7869 - auc: 0.8697 - val_loss: 0.5808 - val_acc: 0.7144 - va
l_auc: 0.8183
Epoch 32/50
100/100 [==============================] - 40s 400ms/step - loss: 0.454
1 - acc: 0.7762 - auc: 0.8652 - val_loss: 0.5470 - val_acc: 0.7444 - va
l_auc: 0.8384
Epoch 33/50
100/100 [==============================] - 40s 404ms/step - loss: 0.467
5 - acc: 0.7694 - auc: 0.8555 - val_loss: 0.5784 - val_acc: 0.7369 - va
l_auc: 0.8355
Epoch 34/50
100/100 [==============================] - 40s 400ms/step - loss: 0.460
8 - acc: 0.7781 - auc: 0.8575 - val_loss: 0.5428 - val_acc: 0.7525 - va
l_auc: 0.8297
Epoch 35/50
100/100 [==============================] - 40s 399ms/step - loss: 0.426
8 - acc: 0.8031 - auc: 0.8816 - val_loss: 0.5597 - val_acc: 0.7369 - va
l_auc: 0.8467
Epoch 36/50
100/100 [==============================] - 40s 405ms/step - loss: 0.443
6 - acc: 0.7994 - auc: 0.8627 - val_loss: 0.5698 - val_acc: 0.7381 - va
l_auc: 0.8466
Epoch 37/50
100/100 [==============================] - 40s 404ms/step - loss: 0.431
```

```
100/100 [==============================] - 40s 404ms/step - loss: 0.431
8 - acc: 0.8025 - auc: 0.8747 - val_loss: 0.5420 - val_acc: 0.7519 - va

l_auc: 0.8644
Epoch 38/50
100/100 [==============================] - 40s 399ms/step - loss: 0.396
9 - acc: 0.8187 - auc: 0.8955 - val_loss: 0.5151 - val_acc: 0.7631 - va
l_auc: 0.8525
Epoch 39/50
100/100 [==============================] - 40s 398ms/step - loss: 0.444
2 - acc: 0.7987 - auc: 0.8752 - val_loss: 0.6489 - val_acc: 0.6787 - va
l_auc: 0.7927
Epoch 40/50
100/100 [==============================] - 40s 399ms/step - loss: 0.426
8 - acc: 0.8088 - auc: 0.8780 - val_loss: 0.5948 - val_acc: 0.7094 - va
l_auc: 0.8316
Epoch 41/50
100/100 [==============================] - 40s 404ms/step - loss: 0.404
9 - acc: 0.8137 - auc: 0.8936 - val_loss: 0.5644 - val_acc: 0.7306 - va
l_auc: 0.8422
Epoch 42/50
100/100 [==============================] - 40s 398ms/step - loss: 0.424
4 - acc: 0.7944 - auc: 0.8836 - val_loss: 0.5833 - val_acc: 0.7250 - va
l_auc: 0.8447
Epoch 43/50
100/100 [==============================] - 40s 399ms/step - loss: 0.435
8 - acc: 0.7944 - auc: 0.8720 - val_loss: 0.6532 - val_acc: 0.7063 - va
l_auc: 0.8231
Epoch 44/50
100/100 [==============================] - 40s 400ms/step - loss: 0.395
7 - acc: 0.8131 - auc: 0.9027 - val_loss: 0.5751 - val_acc: 0.7362 - va
l_auc: 0.8491
Epoch 45/50
100/100 [==============================] - 40s 404ms/step - loss: 0.402
2 - acc: 0.8225 - auc: 0.8914 - val_loss: 0.5538 - val_acc: 0.7362 - va
l_auc: 0.8427
Epoch 46/50
100/100 [==============================] - 41s 406ms/step - loss: 0.407
7 - acc: 0.8250 - auc: 0.8937 - val_loss: 0.5762 - val_acc: 0.7212 - va
l_auc: 0.8413
Epoch 47/50
```

```
Epoch 47/50
100/100 [==============================] - 40s 405ms/step - loss: 0.357

0 - acc: 0.8356 - auc: 0.9172 - val_loss: 0.5424 - val_acc: 0.7494 - va
l_auc: 0.8480
Epoch 48/50
100/100 [==============================] - 41s 411ms/step - loss: 0.370
0 - acc: 0.8319 - auc: 0.9083 - val_loss: 0.6264 - val_acc: 0.7137 - va
l_auc: 0.8275
Epoch 49/50
100/100 [==============================] - 41s 406ms/step - loss: 0.379
9 - acc: 0.8250 - auc: 0.9025 - val_loss: 0.6085 - val_acc: 0.7200 - va
l_auc: 0.8470
Epoch 50/50
100/100 [==============================] - 41s 405ms/step - loss: 0.393
4 - acc: 0.8287 - auc: 0.8989 - val_loss: 0.6172 - val_acc: 0.7256 - va
l_auc: 0.8314
```

Out[17]: `<keras.callbacks.History at 0x7fb4b6064198>`

In [18]:
```
submission = pd.read_csv('../input/sample_submission.csv')
submission['p1'] = submission.img_pair.apply( lambda x: '../input/tes
t/'+x.split('-')[0] )
submission['p2'] = submission.img_pair.apply( lambda x: '../input/tes
t/'+x.split('-')[1] )
print(submission.shape)
submission.head()
```

```
(5310, 4)
```

Out[18]:

	img_pair	is_related	p1	p2
0	face05508.jpg-face01210.jpg	0	../input/test/face05508.jpg	../input/test/face01210.jpg
1	face05750.jpg-face00898.jpg	0	../input/test/face05750.jpg	../input/test/face00898.jpg
2	face05820.jpg-face03938.jpg	0	../input/test/face05820.jpg	../input/test/face03938.jpg
3	face02104.jpg-face01172.jpg	0	../input/test/face02104.jpg	../input/test/face01172.jpg
4	face02428.jpg-face05611.jpg	0	../input/test/face02428.jpg	../input/test/face05611.jpg

In [19]:
```
probs = []
for i,j in tqdm([ (0,500),(500,1000),(1000,1500),(1500,2000),(2000,2500),
                  (2500,3000),(3000,3500),(3500,4000),(4000,4500),(4500,5000),(5000,5310) ]):
    imgs1 = np.array( [ read_img(photo) for photo in submission.p1.values[i:j] ] )
    imgs2 = np.array( [ read_img(photo) for photo in submission.p2.values[i:j] ] )
    prob =  model.predict( [ imgs1, imgs2 ] )
    probs.append(np.squeeze(prob))
    del imgs1,imgs2; gc.collect()
```

```
100%|██████████| 11/11 [01:19<00:00,  6.65s/it]
```

In [20]:
```
submission.is_related = np.concatenate(probs)
submission.drop( ['p1','p2'],axis=1,inplace=True )
submission.head()
```

Out[20]:

	img_pair	is_related
0	face05508.jpg-face01210.jpg	0.001957
1	face05750.jpg-face00898.jpg	0.960254
2	face05820.jpg-face03938.jpg	0.922596
3	face02104.jpg-face01172.jpg	0.919392
4	face02428.jpg-face05611.jpg	0.821485

References

- https://www.kaggle.com/hsinwenchang/vggface-baseline-197x197
- https://www.kaggle.com/vaishvik25/blend-of-smiles

In [ ]: