# Visual Kinship Recognition

B Pravena
Deepa Shree C V
Keerthi Joshi K

# Contents

# PROBLEM STATEMENT

Given a pair of images of faces, recognise whether the people in the images are related are not.

The output is a label which says whether the people are related are not.

# DATASET DESCRIPTION

The dataset being used is Faces In the Wild. The folder ‘train’ consists of subfolders of families with names(F0123), then these family folder contains subfolders for individuals (MIDx). Images in the same MIDx folder belong to the same person. Images in the same folder F0123 belong to the same family.

# DATASET DESCRIPTION

## Train folder is shown below:


## Structure of each subfolder:

# DATASET DESCRIPTION

Each folder contains facial images of a person:

# DATASET DESCRIPTION

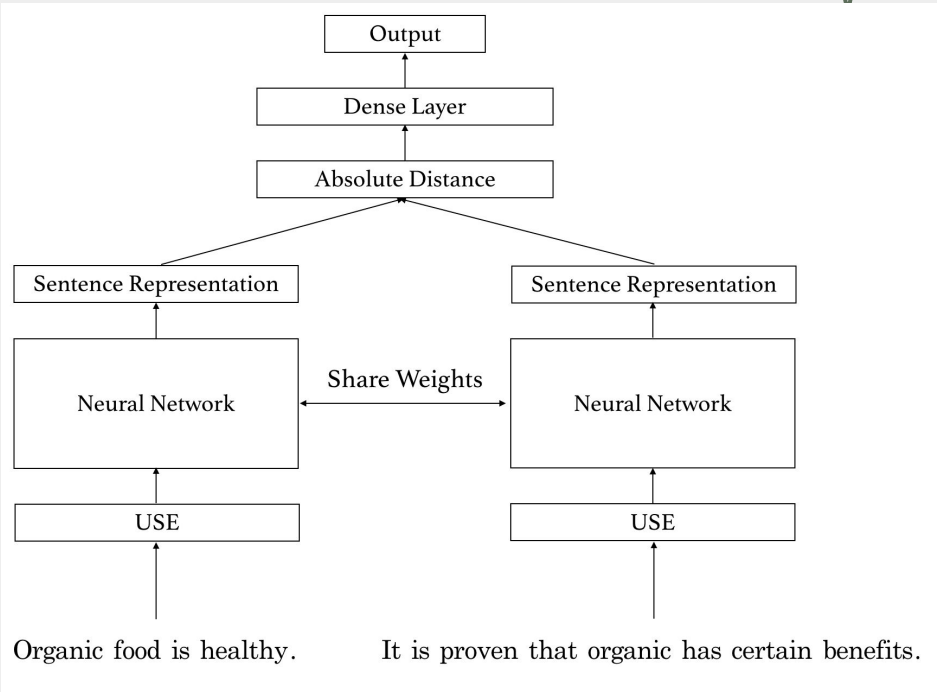
The folder ‘test’ contains images of faces that need to be tested with some other random image to be kin related or not.
The file ‘train_relationships.csv’ contains training labels:

| | | |
|---|---|---|
| 1 | F0002/MID1 | F0002/MID3 |
| 2 | F0002/MID2 | F0002/MID3 |
| 3 | F0005/MID1 | F0005/MID2 |
| 4 | F0005/MID3 | F0005/MID2 |
| 5 | F0009/MID1 | F0009/MID4 |
| 6 | F0009/MID1 | F0009/MID3 |
| 7 | F0009/MID1 | F0009/MID2 |
| 8 | F0009/MID1 | F0009/MID6 |
| 9 | F0009/MID2 | F0009/MID4 |
| 10 | F0009/MID2 | F0009/MID6 |

train_relationships.csv

# MODEL DESCRIPTION:

## Siamese Networks:



# STRUCTURE OF THE MODEL

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 input_2 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 vggface_resnet50 (Functional)  (None, None, None,   23561152    ['input_1[0][0]',
                                2048)                             'input_2[0][0]']

 global_max_pooling2d (GlobalMa  (None, 2048)        0           ['vggface_resnet50[0][0]']
 xPooling2D)

 global_average_pooling2d (Glob  (None, 2048)        0           ['vggface_resnet50[0][0]']
 alAveragePooling2D)

 global_max_pooling2d_1 (Global  (None, 2048)        0           ['vggface_resnet50[1][0]']
 MaxPooling2D)

 global_average_pooling2d_1 (Gl  (None, 2048)        0           ['vggface_resnet50[1][0]']
 obalAveragePooling2D)

 concatenate (Concatenate)      (None, 4096)         0           ['global_max_pooling2d[0][0]',
                                                                  'global_average_pooling2d[0][0]'
                                                                 ]

 concatenate_1 (Concatenate)    (None, 4096)         0           ['global_max_pooling2d_1[0][0]',
                                                                  'global_average_pooling2d_1[0][0
                                                                 ']
```

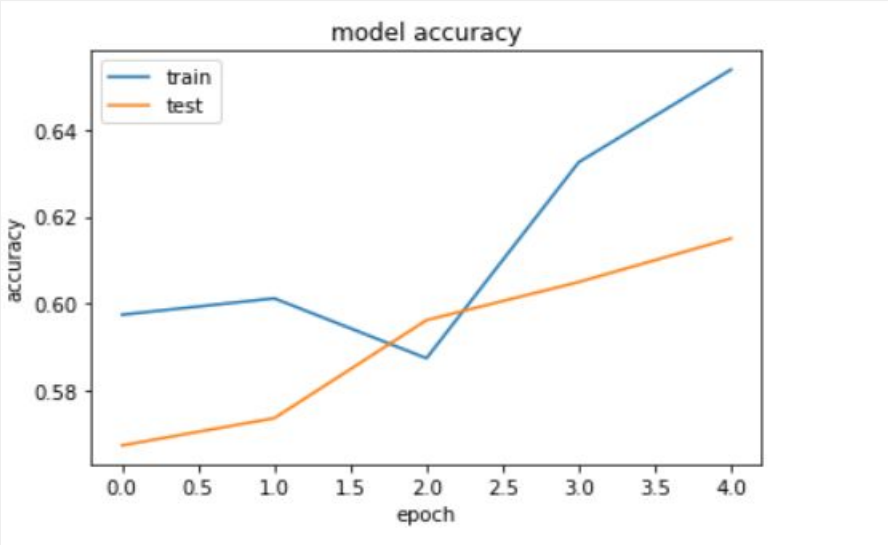
```
                                                                  ]

concatenate_1 (Concatenate)    (None, 4096)         0            ['global_max_pooling2d_1[0][0]',
                                                                  'global_average_pooling2d_1[0][0
                                                                  ]']

subtract (Subtract)            (None, 4096)         0            ['concatenate[0][0]',
                                                                  'concatenate_1[0][0]']

multiply_1 (Multiply)          (None, 4096)         0            ['concatenate[0][0]',
                                                                  'concatenate[0][0]']

multiply_2 (Multiply)          (None, 4096)         0            ['concatenate_1[0][0]',
                                                                  'concatenate_1[0][0]']

multiply (Multiply)            (None, 4096)         0            ['subtract[0][0]',
                                                                  'subtract[0][0]']

subtract_1 (Subtract)          (None, 4096)         0            ['multiply_1[0][0]',
                                                                  'multiply_2[0][0]']

multiply_3 (Multiply)          (None, 4096)         0            ['concatenate[0][0]',
                                                                  'concatenate_1[0][0]']

concatenate_2 (Concatenate)    (None, 12288)        0            ['multiply[0][0]',
                                                                  'subtract_1[0][0]',
                                                                  'multiply_3[0][0]']

dense (Dense)                  (None, 100)          1228900      ['concatenate_2[0][0]']

dropout (Dropout)              (None, 100)          0            ['dense[0][0]']
```

# Training the model:

```
[ ]  history=model.fit_generator(data_generator(train,train_person_to_images_map,batch_size=16),\
                        use_multiprocessing=True,\
                        validation_data=data_generator(val, val_person_to_images_map, batch_size=16),\
                        epochs=5,verbose=1,workers=4,callbacks=callbacks_list,steps_per_epoch=50,validation_steps=50)
```

```
Epoch 1/5
50/50 [==============================] - ETA: 0s - loss: 4.4602 - acc: 0.5975
Epoch 1: val_acc improved from -inf to 0.56750, saving model to /content/drive/MyDrive/Dataset/vgg_face_1.h5
50/50 [==============================] - 57s 728ms/step - loss: 4.4602 - acc: 0.5975 - val_loss: 7.8688 - val_acc: 0.5675 - lr: 1.0000e-05
Epoch 2/5
50/50 [==============================] - ETA: 0s - loss: 3.8433 - acc: 0.6012
Epoch 2: val_acc improved from 0.56750 to 0.57375, saving model to /content/drive/MyDrive/Dataset/vgg_face_1.h5
50/50 [==============================] - 34s 692ms/step - loss: 3.8433 - acc: 0.6012 - val_loss: 6.3762 - val_acc: 0.5738 - lr: 1.0000e-05
Epoch 3/5
50/50 [==============================] - ETA: 0s - loss: 3.6852 - acc: 0.5875
Epoch 3: val_acc improved from 0.57375 to 0.59625, saving model to /content/drive/MyDrive/Dataset/vgg_face_1.h5
50/50 [==============================] - 36s 726ms/step - loss: 3.6852 - acc: 0.5875 - val_loss: 4.9484 - val_acc: 0.5962 - lr: 1.0000e-05
Epoch 4/5
50/50 [==============================] - ETA: 0s - loss: 2.5923 - acc: 0.6325
Epoch 4: val_acc improved from 0.59625 to 0.60500, saving model to /content/drive/MyDrive/Dataset/vgg_face_1.h5
50/50 [==============================] - 37s 738ms/step - loss: 2.5923 - acc: 0.6325 - val_loss: 4.1381 - val_acc: 0.6050 - lr: 1.0000e-05
Epoch 5/5
50/50 [==============================] - ETA: 0s - loss: 2.3952 - acc: 0.6538
Epoch 5: val_acc improved from 0.60500 to 0.61500, saving model to /content/drive/MyDrive/Dataset/vgg_face_1.h5
50/50 [==============================] - 39s 797ms/step - loss: 2.3952 - acc: 0.6538 - val_loss: 3.4258 - val_acc: 0.6150 - lr: 1.0000e-05
```

# Results:


model accuracy
train
test
accuracy
0.64
0.62
0.60
0.58
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
epoch

# THANK YOU