Delhi Technological University
Assignment-3
EC-412:Machine Learning

Shaurya Singh Rawat - 2K17/ME/214

10 May 2021

0.1 Problem Statement

A University wants to use Face Verification for validating student IDs in facilities (Library, dining halls, gym, indoor sports complex . . .)

Data: Picture of every student labelled with their name

Input: Image to be verified

Output: y=1 (It's true) ;y=0 (It's not true)

Model an independent face verification task with a machine learning algorithm.

0.2 Introduction

Face Recognition is a computer vision task of identifying and verifying a person based on a photograph of their face.

Recently, deep learning convolutional neural networks have surpassed classical methods and are achieving state-of-the-art results on standard face recognition datasets. One example of a state-of-the-art model is the VGGFace and VGGFace2 model developed by researchers at the Visual Geometry Group at Oxford.

Although the model can be challenging to implement and resource intensive to train, it can be easily used in standard deep learning libraries such as Keras through the use of freely available pre-trained models and third-party open source libraries.

0.3 Methodology

For the purpose of face verification, the pre-trained VGGFace 2 model developed by the Visual Geometry Group (VGG) at University of Oxford is used. A pre-traiend model is deployed for the purposes of verification since the developemnt and training of models from scratch can be computationally very expensive and offer less accuracy as compared to existing pre-trained models.

To perform the task of face recognition, the model has to detect the faces first. The Multi-Task Cascaded Convolutional Neural Network is used to detect and extract faces from images. 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."

To perform the face verification task, a face embedding for a new given face is calculated and compared to the embedding for the single example of the face known to the system. A face embedding is 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. In this particular case, we use the cosine distance to compare the embeddings and generate results.

The VGGFace2 dataset is made of around 3.31 million images divided into 9131 classes, each representing a different person identity. The dataset is divided into two splits, one for the training and one for test. The latter contains around 170000 images divided into 500 identities while all the other images belong to the remaining 8631 classes available for training. While constructing the datasets, the authors focused their efforts on reaching a very low label noise and a high pose and age diversity thus, making the VGGFace2 dataset a suitable choice to train state-of-the-art deep learning models on face-related tasks. The images of the training set have an average resolution of 137x180 pixels, with less than 1% at a resolution below 32 pixels.

For the purposes of testing and validating the model, a dataset of images of people collected from their wikipedia page was collected and stored. Certain test images were collected of the people in the dataset and then parsed to the model to validate the models perfomance.

0.4 Python Code

```
from numpy import expand_dims
from matplotlib import pyplot
from PIL import Image
from numpy import asarray
from mtcnn.mtcnn import MTCNN
from keras_vggface.vggface import VGGFace
from keras_vggface.utils import preprocess_input
from keras_vggface.utils import decode_predictions
from scipy.spatial import distance
import glob

# 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

# extract faces and calculate face embeddings for a list of photo files
def get_embeddings(filenames):
    # 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)
    # create a vggface model
    model = VGGFace(model='resnet50', include_top=False, input_shape=(224, 224, 3), pooling=
    'avg')
    # perform prediction
    yhat = model.predict(samples)
    return yhat

# determine if a candidate face is a match for a known face
def is_match(known_embedding, candidate_embedding, filename, thresh=0.5):
    # calculate distance between embeddings
    score = distance.cosine(known_embedding, candidate_embedding)
    if score <= thresh:
        print('>face is a Match (%.3f <= %.3f)' % (score, thresh))
        print(filename)

#define path where database is stored
path = "/home/darthnoobisuke/ml/Datasets/Assignment-3"
all_files = glob.glob(path + "/*.jpg")

#get embeddings for image to be verified
embedding_verification = get_embeddings(image_verification)

#get database embeddings and compare to image and print result
for filename in all_files:
    embeddings = get_embeddings(filename)
    is_match(embedding_verification,embeddings)
```

0.5 Results

For demonstration purposes, a database of images of some people were collected and stored. Some examples from the database are shown.

Figure 1: **Some Samples from the Database**

The following test image was parsed through the model:

Figure 2: **Test Image**

And the following result was obtained:

```
In [39]: #define path where database is stored
path = "/home/darthnoobisuke/ml/EC-412:Machine Learning/Assignment-3/Database"
all_files = glob.glob(path + "/*.jpg")
image_verification = 'dp.jpg'

#get embeddings for image to be verified
embedding_verification = get_embeddings(image_verification)

#get database embeddings and compare to image and print result
for filename in all_files:
    embeddings = get_embeddings(filename)
    is_match(embedding_verification,embeddings)
```

```
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:5 out of the last 14 calls to <function Model.make_predict_function.<locals>.predict_function
at 0x7fdd320219d8> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 15 calls to <function Model.make_predict_function.<locals>.predict_function
at 0x7fdd320219d8> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
>face is a Match (0.239 <= 0.500)
Deepika Padukone.jpg
```

Figure 3: **Result of the Test Image**

To test the robustness of the model, multiple different types of images of the person with varying age, lighting effects, beards,facial features, expressions and hairstyles were parsed through the model. The images and their results are shown below. As we can see, the model is able to consistently verify the faces regardless of age, lighting effects, beards,facial features, expressions and hairstyles and is very robust in performance. Hence, the VGG Face 2 model is excellent for facial verification.

GitHub Link: https://github.com/PoitatoMaster/EC-412-Assignment-3

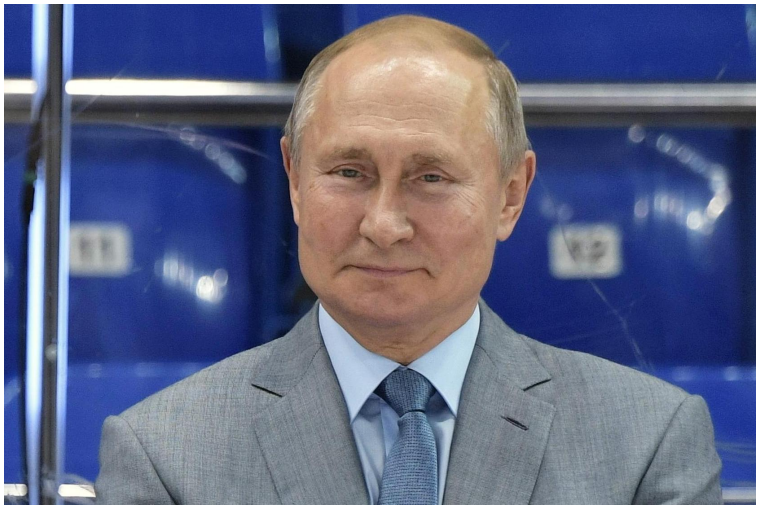

Figure 4: **Test Image**

```
In [42]: #define path where database is stored
path = "/home/darthnoobisuke/ml/EC-412:Machine Learning/Assignment-3/Database"
all_files = glob.glob(path + "/*.jpg")
image_verification = 'vp.jpg'

#get embeddings for image to be verified
embedding_verification = get_embeddings(image_verification)

#get database embeddings and compare to image and print result
for filename in all_files:
    embeddings = get_embeddings(filename)
    is_match(embedding_verification,embeddings)
```

```
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:5 out of the last 14 calls to <function Model.make_predict_function.<locals>.predict_function
at 0x7fdd40195158> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 15 calls to <function Model.make_predict_function.<locals>.predict_function
at 0x7fdd40195158> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
>face is a Match (0.287 <= 0.500)
Vladimir Putin.jpg
```

Figure 5: **Result of the Test Image**

Figure 6: **Test Image**

```
In [40]: #define path where database is stored
path = "/home/darthnoobisuke/ml/EC-412:Machine Learning/Assignment-3/Database"
all_files = glob.glob(path + "/*.jpg")
image_verification = 'dp2.jpg'

#get embeddings for image to be verified
embedding_verification = get_embeddings(image_verification)

#get database embeddings and compare to image and print result
for filename in all_files:
    embeddings = get_embeddings(filename)
    is_match(embedding_verification,embeddings)
```

```
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:5 out of the last 14 calls to <function Model.make_predict_function.<locals>.predict_function
at 0x7fdd68915950> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 15 calls to <function Model.make_predict_function.<locals>.predict_function
at 0x7fdd68915950> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
>face is a Match (0.298 <= 0.500)
Deepika Padukone.jpg
```

Figure 7: **Result of the Test Image**

Figure 8: **Test Image**

```
In [35]: #define path where database is stored
path = "/home/darthnoobisuke/ml/EC-412:Machine Learning/Assignment-3/Database"
all_files = glob.glob(path + "/*.jpg")
image_verification = 'dhl.jpg'

#get embeddings for image to be verified
embedding_verification = get_embeddings(image_verification)

#get database embeddings and compare to image and print result
for filename in all_files:
    embeddings = get_embeddings(filename)
    is_match(embedding_verification,embeddings)
```

```
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:5 out of the last 14 calls to <function Model.make_predict_function.<locals>.predict_function
at 0x7fdd388b1c80> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 15 calls to <function Model.make_predict_function.<locals>.predict_function
at 0x7fdd388b1c80> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
>face is a Match (0.264 <= 0.500)
Dhoni.jpg
```

Figure 9: **Result of the Test Image**

Figure 10: **Test Image**

```
In [41]: #define path where database is stored
         path = "/home/darthnoobisuke/ml/EC-412:Machine Learning/Assignment-3/Database"
         all_files = glob.glob(path + "/*.jpg")
         image_verification = 'kk.jpg'

         #get embeddings for image to be verified
         embedding_verification = get_embeddings(image_verification)

         #get database embeddings and compare to image and print result
         for filename in all_files:
             embeddings = get_embeddings(filename)
             is_match(embedding_verification,embeddings)

         function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
         ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
         s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
         WARNING:tensorflow:5 out of the last 14 calls to <function Model.make_predict_function.<locals>.predict_function
         at 0x7fdd4911de18> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
         ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
         ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
         function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
         ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
         s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
         WARNING:tensorflow:6 out of the last 15 calls to <function Model.make_predict_function.<locals>.predict_function
         at 0x7fdd4911de18> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings cou
         ld be due to (1) creating @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passi
         ng Python objects instead of tensors. For (1), please define your @tf.function outside of the loop. For (2), @tf.
         function has experimental_relax_shapes=True option that relaxes argument shapes that can avoid unnecessary retrac
         ing. For (3), please refer to https://www.tensorflow.org/tutorials/customization/performance#python_or_tensor_arg
         s and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
         >face is a Match (0.233 <= 0.500)
         Katrina Kaif.jpg
```

Figure 11: **Result of the Test Image**