

Task1:

Epoch: [1/10]; Time: 14.96; Loss: 0.53782

Epoch: [2/10]; Time: 22.58; Loss: 0.25015

Epoch: [3/10]; Time: 49.07; Loss: 0.20976

Epoch: [4/10]; Time: 49.29; Loss: 0.19032

Epoch: [5/10]; Time: 48.69; Loss: 0.17217

Epoch: [6/10]; Time: 29.68; Loss: 0.16702

Epoch: [7/10]; Time: 54.25; Loss: 0.15679

Epoch: [8/10]; Time: 45.85; Loss: 0.14697

Epoch: [9/10]; Time: 48.68; Loss: 0.14987

Epoch: [10/10]; Time: 35.98; Loss: 0.14171

Training completed, saving model to ./save/

Evaluation result: 98.64

Inference result: 1 Epoch: [1/10]; Time: 14.96; Loss: 0.53782

Epoch: [2/10]; Time: 22.58; Loss: 0.25015

Epoch: [3/10]; Time: 49.07; Loss: 0.20976

Epoch: [4/10]; Time: 49.29; Loss: 0.19032

Epoch: [5/10]; Time: 48.69; Loss: 0.17217

Epoch: [6/10]; Time: 29.68; Loss: 0.16702

Epoch: [7/10]; Time: 54.25; Loss: 0.15679

Epoch: [8/10]; Time: 45.85; Loss: 0.14697

Epoch: [9/10]; Time: 48.68; Loss: 0.14987

Epoch: [10/10]; Time: 35.98; Loss: 0.14171

Training completed, saving model to ./save/

Evaluation result: 98.64

Inference result: 1

The evaluation result is 98.64, so the performance is good. The loss is decreasing as epoch number increases.

Task2:

1. The paper proposed multiple ResNet with different numbers of layers residual nets. There are ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, and ResNet 20, ResNet-32, ResNet-44, ResNet-56, ResNet-110, ResNet-1202. There are 11 types of them.
In the code, besides different models with different layers of residual nets, there are ResNet, wide ResNet, and ResNet_weights. The wide ResNet used a wider layer, and the ResNet_weights used an existed parameter set to initialize the model. The implementation code includes 19 different ResNet models.
2. ResNet uses deeper residual nets with lower complexity compared to the VGG.
3. The input size of the images for ResNet-50 is 224x224.

Task3:

1. Verification



The verification works well on distinguishing images from different people.

For the two images above, it returns:

```
{'verified': False, 'distance': 0.984907538653792, 'threshold': 0.4, 'model': 'Facenet',  
'detector_backend': 'opencv', 'similarity_metric': 'cosine', 'facial_areas': {'img1': {'x': 244, 'y': 54,  
'w': 142, 'h': 142}, 'img2': {'x': 463, 'y': 332, 'w': 744, 'h': 744}}, 'time': 3.52}
```



The verification also works on recognizing images from the same person.

For the two images above, it returns:

```
{'verified': True, 'distance': 0.2686079555034586, 'threshold': 0.4, 'model': 'Facenet',  
'detector_backend': 'opencv', 'similarity_metric': 'cosine', 'facial_areas': {'img1': {'x': 426, 'y':  
115, 'w': 180, 'h': 180}, 'img2': {'x': 369, 'y': 72, 'w': 166, 'h': 166}}, 'time': 2.77}
```



However, it doesn't work when there's an age difference. For the two figures above, both are photos of Leonardo DiCaprio, but DeepFace failed to recognize that they belong to the same person.

```
{'verified': False, 'distance': 0.4312148677646823, 'threshold': 0.4, 'model': 'Facenet',  
'detector_backend': 'opencv', 'similarity_metric': 'cosine', 'facial_areas': {'img1': {'x': 270, 'y':  
262, 'w': 645, 'h': 645}, 'img2': {'x': 123, 'y': 115, 'w': 275, 'h': 275}}, 'time': 4.48}
```



DeepFace can also handle many faces in a picture. It can select the most similar face for comparison. In this case, DeepFace compares two Kim Kardashians faces and returns “True”, and it is correct.

```
{'verified': True, 'distance': 0.3291297550385953, 'threshold': 0.4, 'model': 'Facenet',  
'detector_backend': 'opencv', 'similarity_metric': 'cosine', 'facial_areas': {'img1': {'x': 1120, 'y':  
143, 'w': 859, 'h': 859}, 'img2': {'x': 94, 'y': 180, 'w': 365, 'h': 365}}, 'time': 3.31}
```

2. Facial Attribute Analysis



```
[{'age': 30, 'region': {'x': 388, 'y': 212, 'w': 612, 'h': 612},  
'gender': {'Woman': 99.9091386795044, 'Man': 0.09086516220122576}, 'dominant_gender':  
'Woman',  
'race': {'asian': 4.907157631350856e-05, 'indian': 2.5392490887732322e-08, 'black':  
4.041953837602419e-10, 'white': 99.99734163284302, 'middle eastern':  
0.0005765687546954723, 'latino hispanic': 0.002037527519860305}, 'dominant_race': 'white',  
'emotion': {'angry': 5.53036704659462, 'disgust': 0.0004688793978857575, 'fear':  
20.279434323310852, 'happy': 0.8676848374307156, 'sad': 16.988113522529602, 'surprise':  
0.1581718330271542, 'neutral': 56.17575645446777}, 'dominant_emotion': 'neutral'}]
```



[{'age': 25, 'region': {'x': 270, 'y': 262, 'w': 645, 'h': 645},
'gender': {'Woman': 8.725249358576548e-05, 'Man': 99.99991655349731}, 'dominant_gender':
'Man',
'race': {'asian': 0.22067935205996037, 'indian': 0.7121640257537365, 'black':
0.2358671510592103, 'white': 61.129653453826904, 'middle eastern': 12.578311562538147,
'latino hispanic': 25.123324990272522}, 'dominant_race': 'white',
'emotion': {'angry': 3.182036246778741, 'disgust': 0.005694551121630959, 'fear':
0.7382215440703359, 'happy': 0.13116541867571585, 'sad': 11.304753947541354, 'surprise':
0.030510996657147803, 'neutral': 84.60761812970746}, 'dominant_emotion': 'neutral'}]



[{'age': 26, 'region': {'x': 123, 'y': 115, 'w': 275, 'h': 275}, 'gender': {'Woman':
0.04401826881803572, 'Man': 99.95598196983337}, 'dominant_gender': 'Man',
'race': {'asian': 6.734340597999331e-07, 'indian': 1.2240962307321256e-07, 'black':
4.2368671862192675e-10, 'white': 99.98872876100116, 'middle eastern':
0.006960886508921848, 'latino hispanic': 0.004309201433359545}, 'dominant_race': 'white',
'emotion': {'angry': 3.6320807714510424, 'disgust': 0.0004468566924380868, 'fear':
81.92163598070358, 'happy': 0.08077209831160893, 'sad': 3.4232316716451447, 'surprise':
2.494813836801877, 'neutral': 8.447021004380048}, 'dominant_emotion': 'fear'}]



```
[{'age': 31, 'region': {'x': 830, 'y': 321, 'w': 1604, 'h': 1604}, 'gender': {'Woman': 0.00044139792407804634, 'Man': 99.99955892562866}, 'dominant_gender': 'Man', 'race': {'asian': 1.2718439279524318e-05, 'indian': 0.0004766209258377785, 'black': 99.99946355819702, 'white': 9.828313857207505e-09, 'middle eastern': 2.5131858399918272e-09, 'latino hispanic': 5.345376621335163e-05}, 'dominant_race': 'black', 'emotion': {'angry': 6.589976345895311, 'disgust': 0.011652283832194438, 'fear': 6.491379826747169, 'happy': 73.83990131854422, 'sad': 7.610711163920829, 'surprise': 2.558293346064886, 'neutral': 2.898093485718577}, 'dominant_emotion': 'happy'}}
```

Observation: DeepFace works well on figuring out age, gender, race, and emotion. However, for the Leonardo DiCaprio image, it assumes the first image to be 25 years old, and this picture is younger than the second one. This is not true.

3. Face Detection

Face Detection can crop out the face area of each image.

The image:



The output:

