

지능형 CCTV를 위한 딥러닝 기반 안면 비식별화 연구

Deep learning-based facial de-identification study for intelligent CCTV

유천일

Yu-Tian Yi

yuty@outlook.kr

동국대학교 컴퓨터공학과

Dept. of Computer Science and Engineering, Dongguk Univ., Seoul

요 약

본 논문은 opencv 이용해서 카메라에 찍힌 화면은 실시간으로 안면 모자이크 처리를 하고 모자이크 처리 전 한 프레임의 화면 속 모든 얼굴 이미지를 캡처해 한 장의 큰 그림에 순서대로 배열한 뒤 프레임 순서로 각 얼굴 이미지를 큰 그림 속 좌표 위치를 하나의 텍스트 파일에 저장한다. 그리고 실시간 얼굴 모자이크 처리된 영상을 그대로 원 영상으로 복원하고 모자이크 처리 시 각각의 얼굴을 캡처해 보관하는 큰 그림, 그리고 큰 화면에서 각각의 얼굴이 차지하는 좌표 범위를 통해 모자이크의 범위를 교체하는 방식으로 영상을 복원할 수 있다.

키워드 : 얼굴 인식, 실시간 안면 모자이크 처리, 큰 화면에서 각각의 얼굴이 차지하는 좌표 범위를 모자이크의 범위를 교체를 통해 영상 복원

Abstract

In this paper, using opencv, face mosaic processing is performed on the screen taken by the camera in real time, and before mosaic processing, all face images in the screen are captured and arranged in one large picture, and then each face image is enlarged in frame order. Save the coordinate positions in the picture in one text file. In addition, the real-time face mosaic-processed image is restored to the original image as it is, a large picture that captures and stores each face during mosaic processing, and the image by replacing the range of the mosaic through the coordinate range occupied by each face on the large screen can be restored.

Key Words : Image restoration through face recognition, real-time face mosaic processing, and replacing the mosaic range with the coordinate range occupied by each face on a large screen

1. 서론

CCTV 유출 사고가 빈발하면서 개인의 사생활 침해 사고가 발생하고 있다. 이를 해결하기 위해 얼굴 인식 및 인식 기술을 적용해 영상 속 목표물의 얼굴을 식별하고 추적해 모자이크 처리한다. Pyqt5와 Opencv 기반의 얼굴 인식 및 모자이크 처리 기능, 자체 카메라에 의한 실시간 얼굴 모자이크 처리 및 비디오 저장 기능이다.

본 논문은 먼저 카메라에서 실시간으로 얻은 얼굴 모자이크 처리 및 저장, 카메라에서 실시간으로 모자이크 처리된 얼굴 복원 및 복원된 영상을 저장한다.

2. 얼굴 인식

opencv 인간 안면 분류기 haarcascade_frontalface_default를 이용해서 안면 인식과 좌표 반환한 후에 식별된 안면 범위를 확인한다.

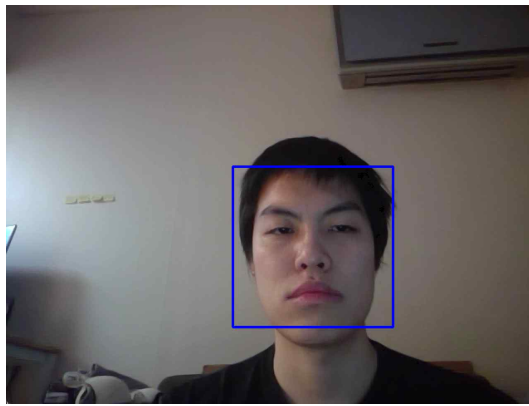


그림 1. opencv 인간 안면 분류기를 이용해서 안면 범위 확인

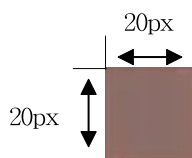
Fig 1. Checking facial range using opencv human face classifier

3. 실시간 모자이크 처리

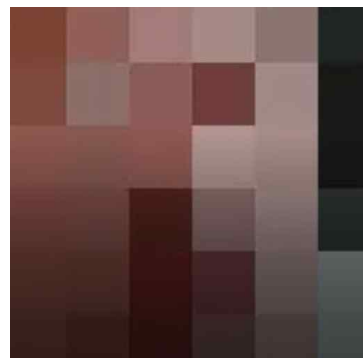
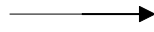
Opencv의 얼굴특징(Haar) 기반의 레벨 커넥터를 이용해 카메라에 찍힌 화면을 실시간으로 얼굴 인식과 모자이크 처리한다. 모자이크를 하기 전에 화면의 모든 얼굴 이미지를 캡처하여 큰 그림 한 장 순으로 배열한 후, 각 얼굴 이미지를 큰 그림 한 장의 좌표 위치에 프레임 순서대로 저장한다.

모자이크 처리방법의 설명:

얼굴 범위를 여러 개의 20x20 행렬로 나누고 각 행렬의 왼쪽 상단 픽셀의 색상 값을 기록한 다음 각 행렬의 모든 픽셀 값을 각 왼쪽 상단 픽셀의 색상 값으로 바꾸어 모자이크 효과를 나타낸다. 모자이크 효과를 얻는 데 사용되는 행렬의 수는 면 범위의 크기에 따라 증가하거나 감소한다. 20x20 행렬로 얼굴 범위를 채울 수 없는 경우 20x20 행렬을 구성할 수 없는 나머지 범위는 모자이크 처리되지 않는다.



(a)



(b)

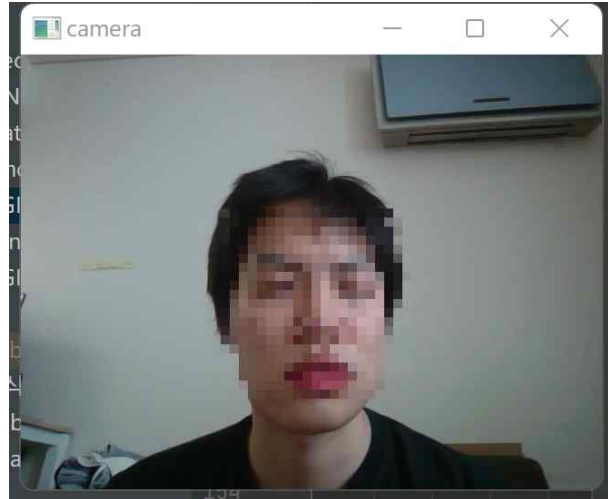
그림 2. 모자이크 처리과정. (a)첫 번째 화소값과 동등하게 처리한 픽셀 블록 (b)얼굴 범위 전체를 모자이크 처리하여 완성한 효과

Fig 2. Process of mosaic treatment. (a)A mosaic image capable of generating an entire image until pixel of the entire picture is processed equally with the first pixel value (b) Effect completed by mosaicizing the entire face area

각 크기의 모자이크 행렬 효과 비교

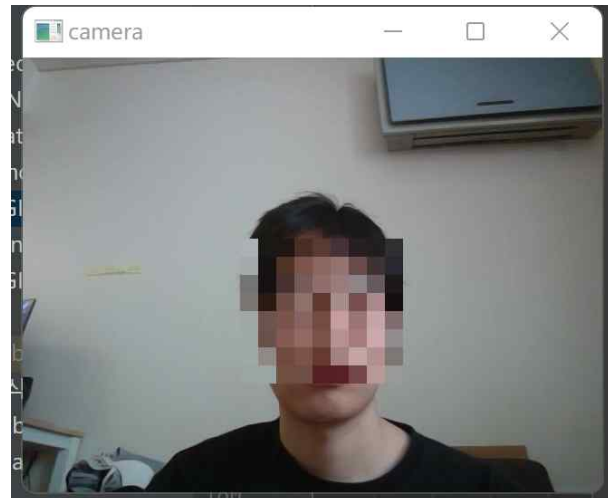
a) 10x10 행렬 모자이크 크기의 모자이크 효과

안면 범위 X%10, y%10 만족하면 만족하는 부분 모자이크 행렬 만든다. 아니면 안 한다.



b) 20x20 행렬 모자이크 크기의 모자이크 효과

안면 범위 X%20, y%20 만족하면 만족하는 부분 모자이크 행렬 만든다. 아니면 안 한다.



c) 55x55 행렬 모자이크 크기의 모자이크 효과

안면 범위 X%55, y%55 만족하면 만족하는 부분 모자이크 행렬 만든다. 아니면 안 한다.

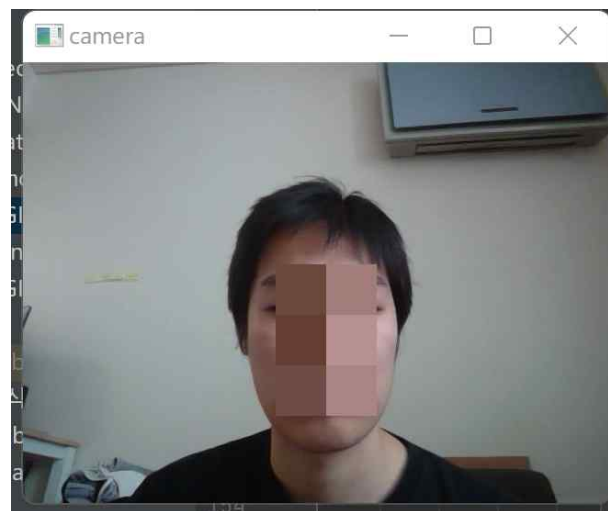


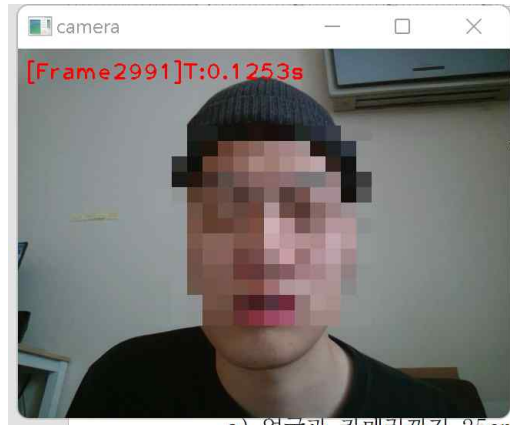
그림 3. 각 크기의 모자이크 행렬 효과 비교. (a)10x10 행렬 크기의 모자이크 효과 (b)20x20 행렬 크기의 모자이크 효과 (c)55x55 행렬 크기의 모자이크 효과

Fig 3. Comparison of the effect of mosaic matrices of each size. (a) Mosaic effect of 10x10 matrix size (b) Mosaic effect of 20x20 matrix size (c) Mosaic effect of 55x55 matrix size

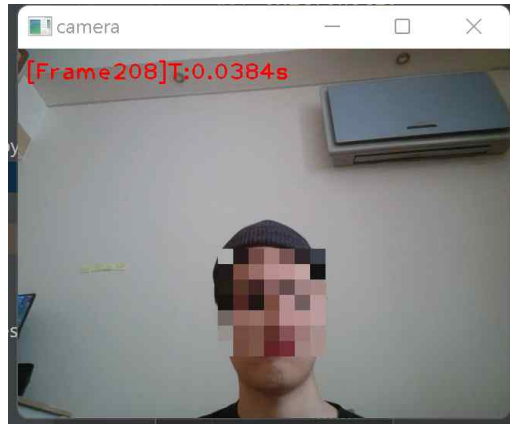
결론: 20x20 행렬의 모자이크 효과가 가장 좋기 때문에 20x20 크기의 행렬로 모자이크 효과를 낸다.

얼굴크기/ 얼굴과 카메라까지의 거리에 대한 설명

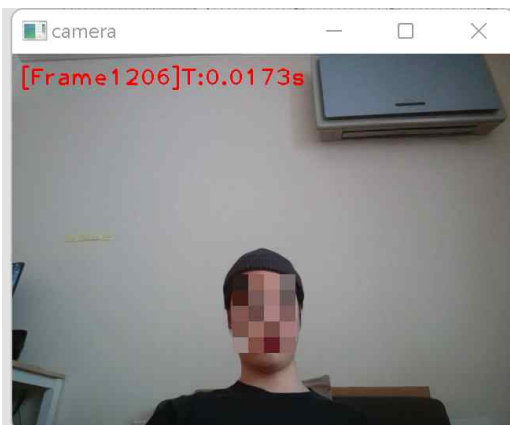
a) 얼굴과 카메라까지 25cm



b) 얼굴과 카메라까지 45cm



c) 얼굴과 카메라까지 75cm



결론: 안면과 카메라 사이의 거리가 멀수록 얼굴 범위가 좁아지고 모자이크 범위도 줄어든다. 얼굴과 카메라 사이의 거리가 가까울수록 얼굴 범위가 넓어지고 모자이크 범위도 넓어진다.

그림 4. 얼굴크기/ 얼굴과 카메라까지의 거리에 대한 설명도 (a)얼굴과 카메라까지 25cm의 모자이크 효과 (b)얼굴과 카메라까지 45cm의 모자이크 효과 (c)얼굴과 카메라까지 75cm의 모자이크 효과

Fig 4. Description of face size/ distance from face to camera (a) Mosaic effect of 25 cm from face to camera (b) Mosaic effect of 45 cm from face to camera (c) Mosaic effect of 75 cm from face to camera

실시간 모자이크 처리과정은 다음과 같다.

realtime_mosaic function(실시간 모자이크 처리과정)

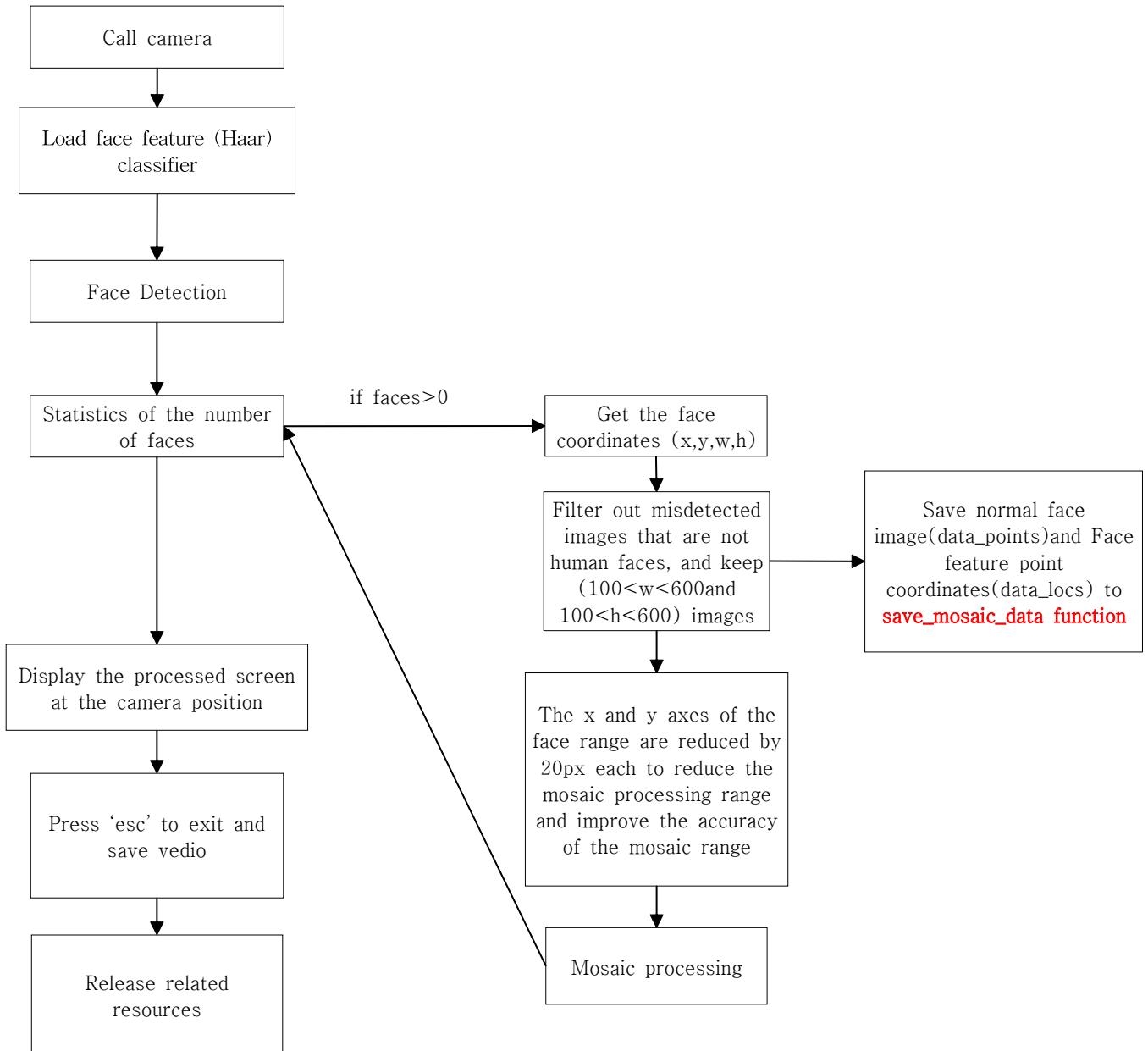


그림 5. 실시간 모자이크 처리 과정도

Fig 5. The process of blurring it out in real-time blurring process.

save_mosaic_data function(실시간 얼굴 모자이크 처리 과정에서 얼굴 관련 정보를 저장)

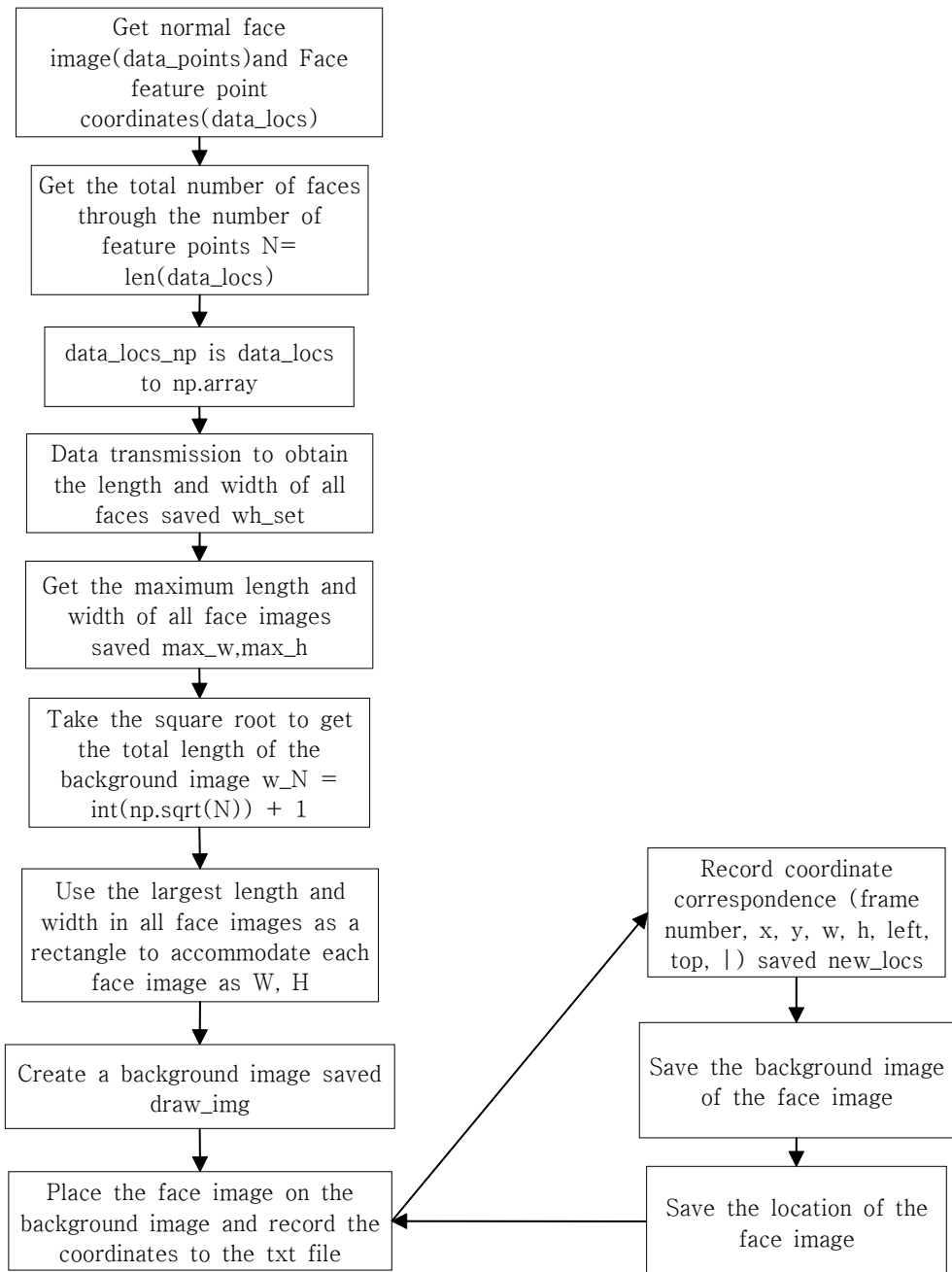


그림 6. 실시간 얼굴 모자이크 처리 과정에서 얼굴 관련 정보를 저장하는 과정이다.

Fig 6. The process of storing face-related information in the real-time face mosaic processing process.

카메라로 실시간 모자이크 처리를 한 이전 단계의 큰 그림과 큰 그림에서 각각의 얼굴 범위 좌표를 기록한 txt 파일을 이용한다. txt 파일에서 좌표점을 받아 큰 그림에서 얼굴을 복사한 뒤에 영상에 모자이크 범위를 한 프레임씩 교체한다. 이런 방법을 통해 동영상상을 복원하는 목적을 달성하다.

a)



큰 사진의 설명

모든 사람의 얼굴 범위 사진 중에서 크기가 가장 큰 한 장의 그림을 찾아 작은 그림 크기로 만들고, 모든 사람의 얼굴 범위 사진을 한 장의 작은 그림에 넣고, 각각의 그림을 직사각형 모양으로 배치한다.

얼굴 이미지 배열 방법(맨 왼쪽 열을 비워두는 것은 그림을 좀 더 세련되게 만들기 위해서다.)

if left > W-max_w-1: #W는배경사각형의 최대 길이

left = 0 #max_w는얼굴이미지의 최대 너비

top += max_h #left는왼쪽에서오른쪽으로 위치 top는 위에서 아래로 위치

b)

```
0,237,241,186,186,0,0|1,238,241,184,184,193,0|2,238,241,183,183,386,0|3,237,240,185,185,579,0|
4,238,240,185,185,772,0|5,239,239,185,185,965,0|6,240,239,185,185,0,193|7,240,240,185,185,193,193|
8,241,239,186,186,386,193|9,241,240,187,187,579,193|10,241,239,188,188,772,193|
11,241,239,189,189,965,193|12,242,239,190,190,0,386|13,243,241,188,188,193,386|
14,240,239,192,192,386,386|15,240,238,193,193,579,386|16,241,238,191,191,772,386|
17,242,240,188,188,965,386|18,241,239,192,192,0,579|19,241,239,191,191,193,579|
20,242,239,191,191,386,579|21,241,238,193,193,579,579|22,242,239,190,190,772,579|
23,241,238,193,193,965,579|24,242,239,191,191,0,772|25,242,240,192,192,193,772|
26,242,239,193,193,386,772|27,241,239,192,192,579,772|28,242,238,192,192,772,772|
29,242,239,192,192,965,772|30,241,238,192,192,0,965|31,241,238,193,193,193,965|
32,242,240,185,185,386,965|33,239,239,187,187,579,965|34,237,240,188,188,772,965|
35,237,243,184,184,965,965|36,235,242,186,186,0,1158|37,234,242,189,189,193,1158|
38,233,244,192,192,386,1158|39,236,245,187,187,579,1158|40,235,245,185,185,772,1158|
41,236,245,182,182,965,1158|
```

그림 7. 실시간 얼굴 모자이크 처리 과정에서 얼굴 관련 정보를 저장된 결과물 a) 캡처된 얼굴을 한 화면에 저장한 사진 b)전체 화면에서 각 얼굴의 데이터 정보(frame number,x,y,w,h,top,left)

Fig 7. The result of saving face-related information in the real-time face mosaic processing process a) A photo of a captured face on one screen b) Data information of each face in the entire screen (frame number,x,y,w,h,top,left)

4. 실시간 동영상 모자이크 처리한 영상 복원

실시간 동영상 모자이크 처리한 영상 복원과정이다.

`read_mosaic_data function`(그림의 얼굴과 txt 파일의 좌표 사이의 대응 관계 읽기)

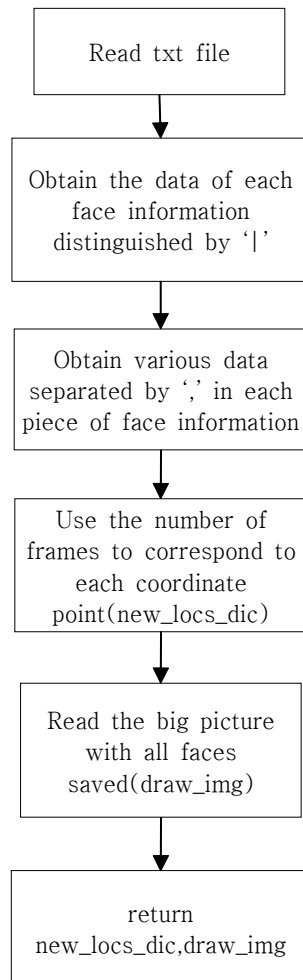


그림 8. 그림의 얼굴과 txt 파일의 좌표 사이의 대응 관계 읽기의 과정

Fig 8. The process of reading the correspondence between the face of the picture and the coordinates of the txt file

realtime_mosaic_reduction funation(모자이크 처리된 동영상 복원 과정)

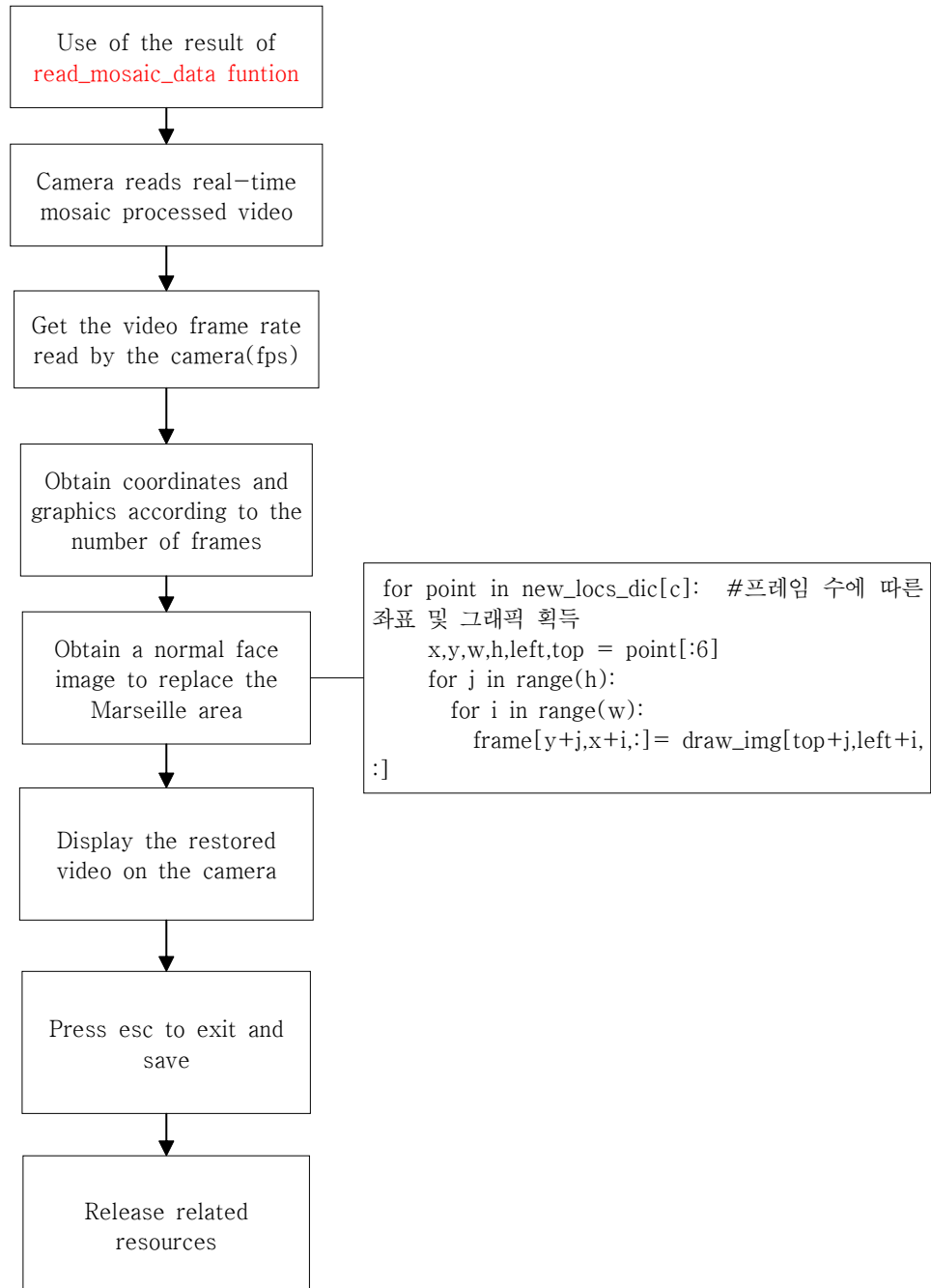


그림 9. 모자이크 처리된 동영상 복원 과정
Fig 9. Mosaic video restoration process

5. 테스트 및 결과

실시간 모자이크 처리의 테스트

카메라에 찍힌 화면을 실시간으로 얼굴 인식과 모자이크 처리한다. 모자이크를 하기 전에 화면의 모든 얼굴 이미지를 캡처하여 큰 그림 순으로 배열한 후, 각 얼굴 이미지를 큰 그림의 좌표 위치에 따라 보존한다. 얼굴 범위를 여러 개의 20x20 행렬로 나누고 각 행렬의 왼쪽 상단 픽셀의 색상 값을 기록한 다음 각 행렬의 모든 픽셀 값을 각 왼쪽 상단 픽셀의 색상 값으로 바꾸어 모자이크 효과를 나타낸다. 모자이크 효과를 얻는 데 사용되는 행렬의 수는 면 범위의 크기에 따라 증가하거나 감소한다. 20x20 행렬로 얼굴 범위를 채울 수 없는 경우 20x20 행렬을 구성할 수 없는 나머지 범위는 모자이크 처리되지 않는다.

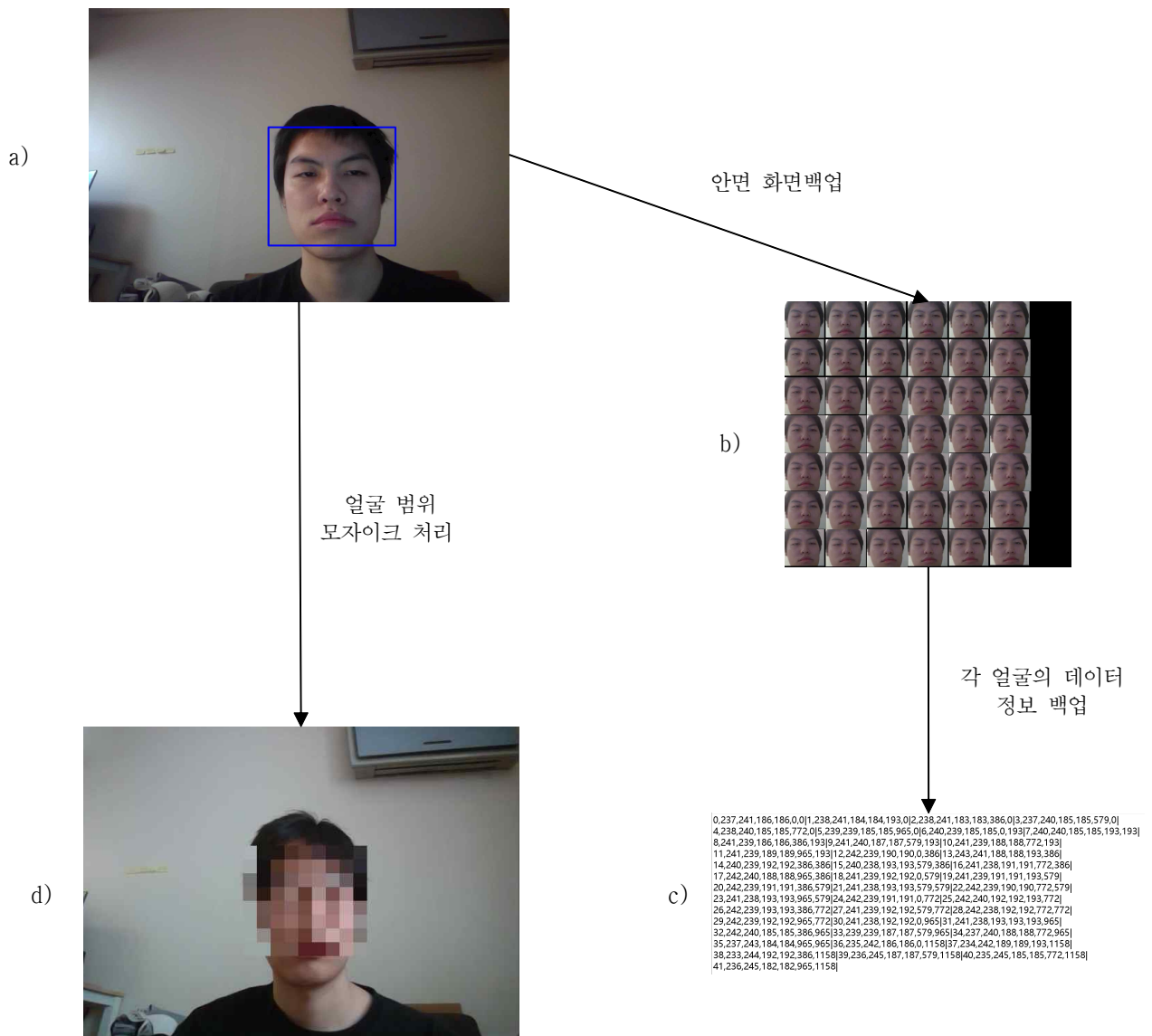
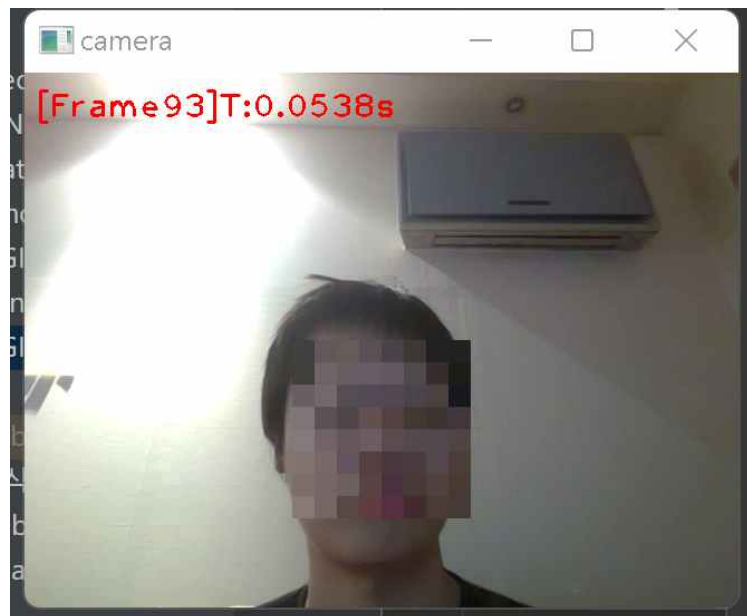


그림 10. 실시간 얼굴 모자이크 처리와 얼굴 데이터백업 과정 a) 안면 범위 확인 b)캡처된 얼굴을 한 화면에 저장한 사진 c)전체 화면에서 각 얼굴의 데이터 정보 d)안면 범위 모자이크 처리

Fig 10. The result of saving face-related information in the real-time face mosaic processing process a) Checking the facial range b) A picture of a captured face on one screen c) Data information of each face in the entire screen d) Mosaicing the facial range

a)얼굴 1개 있는 경우



b)얼굴 2개 있는 경우

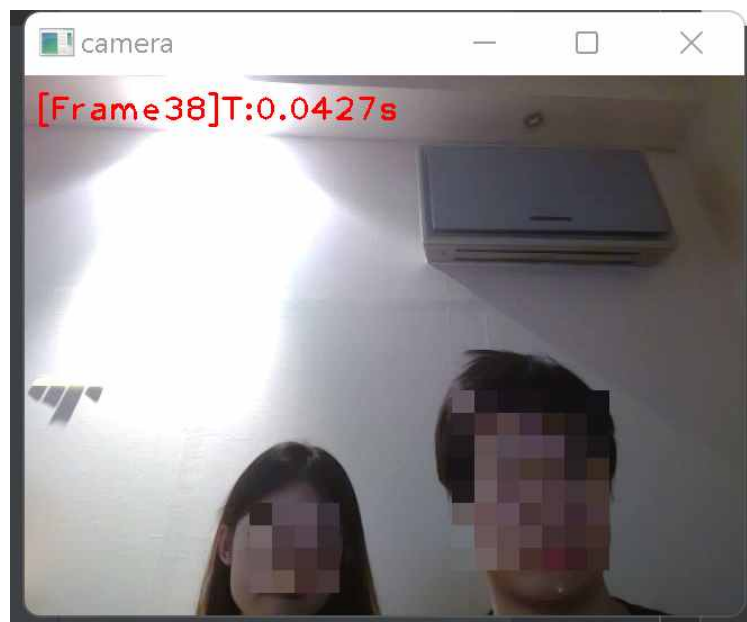


그림 11. 실시간 모자이크 처리 테스트 a)한명 사람 있는 경우 b)여러명 사람 있는 경우

Fig 11. Real-time mosaic processing test a) If there is one person b) If there are multiple people

실시간 모자이크 처리된 영상 복원

모자이크로 실시간 처리된 영상을 사용하며, 프레임 수를 통해 백업한 얼굴 이미지 및 좌표 위치 정보를 통해 동영상 복원 테스트를 한다.

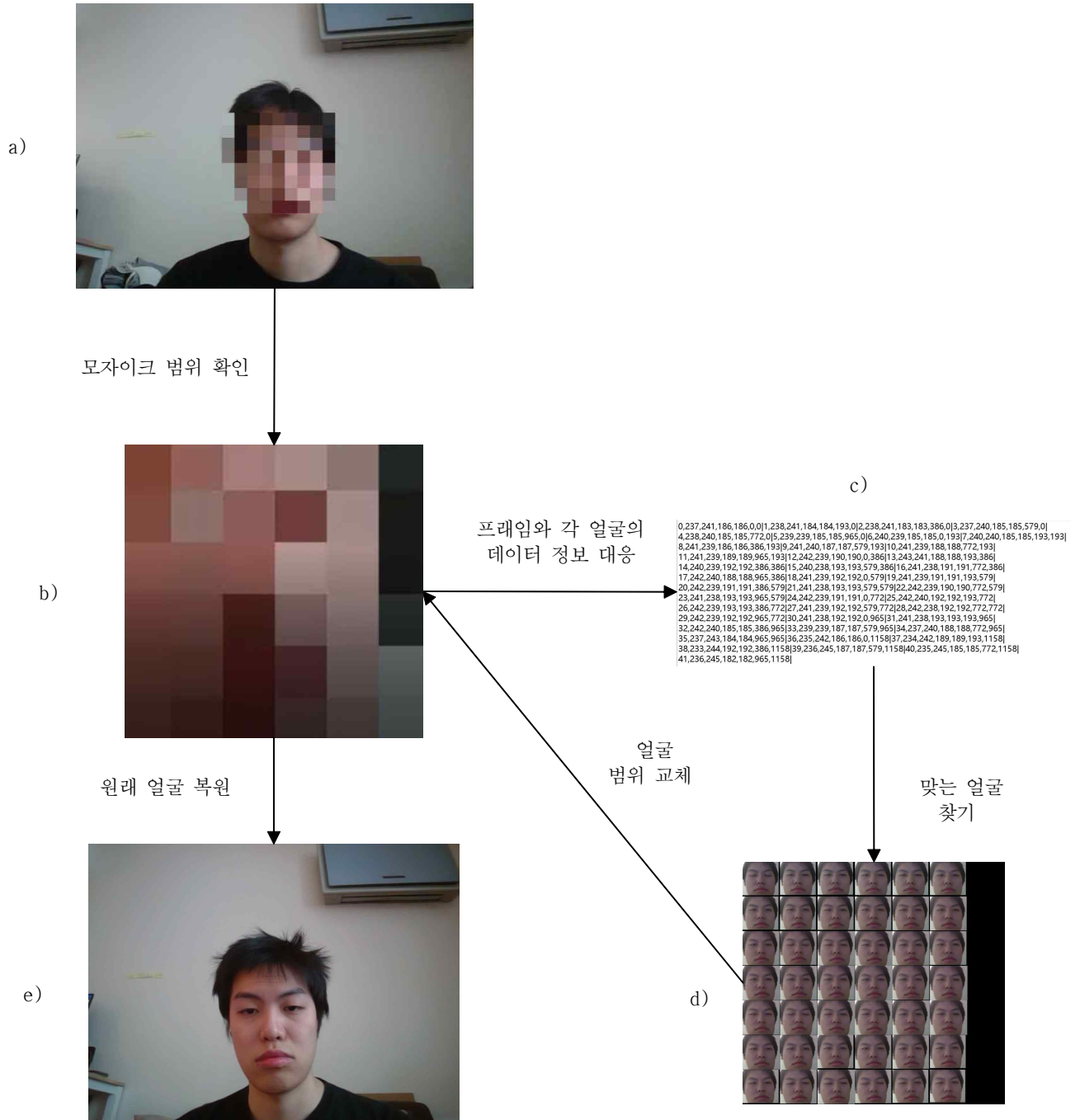
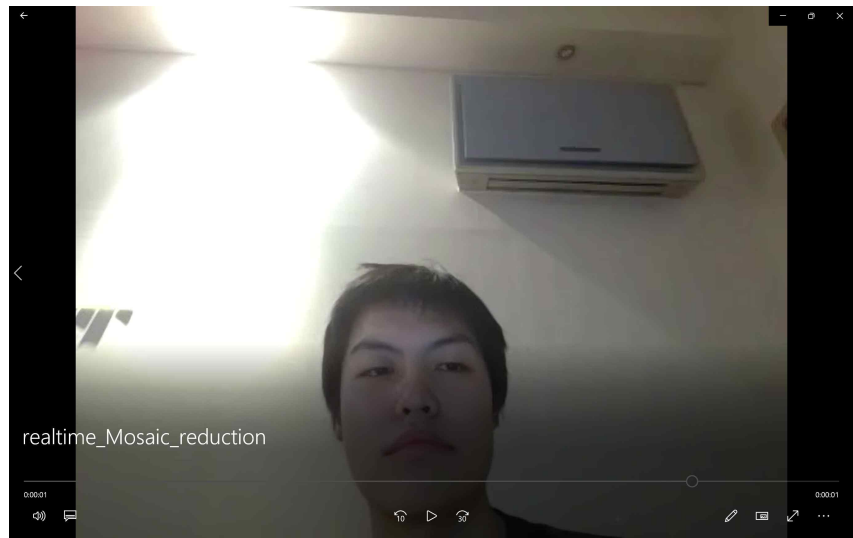


그림 12. 실시간 모자이크 처리된 영상 복원 과정 a,b) 모자이크 범위 확인 c)전체 화면에서 각 얼굴의 데이터 정보 d)안면 범위 모자이크 처리 e)복원한 뒤에 효과도

Fig 12. Real-time mosaiced image restoration process a,b) Mosaic range check c) Data information of each face in full screen d) Facial range mosaic processing e) Effect diagram after restoration

a)얼굴 1개 있는 경우



b)얼굴 2개 있는 경우



그림 13. 실시간 모자이크 처리된 동영상 복원 테스트 a)한명 사람 있는 경우 b)여러명 사람 있는 경우

Fig 13. Real-time mosaic-processed video restoration test a) If there is one person b) If there are multiple people

6. 성능측정결과

실시간 모자이크 성능측정과정(사람의 얼굴이 아닌 이미지를 걸러내고, 메모리 안에 각 사람의 얼굴 특징점 좌표와 정상인의 얼굴 이미지를 저장하고, 얼굴 범위를 모자이크 처리한 후 카메라에 소요시간을 표시한다.)

성능측정 방법(time.time()방법 이용 동작 실행 시간을 얻다)

```
start = time.time()
#long running
#do something other
end = time.time()
```

Cost의 정의: 10프레임마다 평균적으로 각 프레임에 소요되는 시간을 계산한다. (%.4f 형식 표시)

1명사람 있는 경우

10 프레임마다 프레임당 평균 처리 시간 계산 단위(s)

```
Frame[0-9] Cost:0.0627s
Frame[10-19] Cost:0.1306s
Frame[20-29] Cost:0.0626s
Frame[30-39] Cost:0.0665s
Frame[40-49] Cost:0.0496s
Frame[50-59] Cost:0.0525s
Frame[60-69] Cost:0.0626s
Frame[70-79] Cost:0.0513s
Frame[80-89] Cost:0.0553s
Frame[90-99] Cost:0.0546s
```

5초 백업에 필요한 저장 공간의 양
mosaic_data.jpg(안면화면 데이터 저장파일) 860 KB
mosaic_data.txt(안면화면 좌표점을 저장파일) 4.24 KB

30초 백업에 필요한 저장 공간의 양
mosaic_data.jpg(안면화면 데이터 저장파일) 4.77 MB
mosaic_data.txt(안면화면 좌표점을 저장파일) 24.68 KB

2명사람 있는 경우

10 프레임마다 프레임당 평균 처리 시간 계산 단위(s)

```
Frame[0-9] Cost:0.0690s
Frame[10-19] Cost:0.0636s
Frame[20-29] Cost:0.0669s
Frame[30-39] Cost:0.0668s
Frame[40-49] Cost:0.0659s
Frame[50-59] Cost:0.0630s
Frame[60-69] Cost:0.0768s
Frame[70-79] Cost:0.0680s
Frame[80-89] Cost:0.0681s
Frame[90-99] Cost:0.0665s
```

5초 백업에 필요한 저장 공간의 양
mosaic_data.jpg(안면화면 데이터 저장파일) 1.63 MB
mosaic_data.txt(안면화면 좌표점을 저장파일) 7.04 KB

30초 백업에 필요한 저장 공간의 양
mosaic_data.jpg(안면화면 데이터 저장파일) 9.72 MB
mosaic_data.txt(안면화면 좌표점을 저장파일) 42.6 KB

실시간 모자이크처리 된 동영상 복원 성능측정과정(프레임 수에 따라 좌표 및 그래픽을 획득하고 정상 얼굴 이미지를 획득하여 모자이크 영역을 교체한 후 카메라에 소요 시간을 표시한다.)

성능측정 방법(time.time()방법 이용 동작 실행 시간을 얻다)

```
start = time.time()
#long running
#do something other
end = time.time()
```

Cost의 정의: 10프레임마다 평균적으로 각 프레임에 소요되는 시간을 계산한다. (%.4f 형식 표시)

1명사람 있는 경우

10 프레임마다 프레임당 평균 처리 시간 계산 단위(s)

```
Frame[0-9] Cost:0.0344s
Frame[10-19] Cost:0.0737s
Frame[20-29] Cost:0.0374s
Frame[30-39] Cost:0.0369s
Frame[40-49] Cost:0.0313s
Frame[50-59] Cost:0.0316s
Frame[60-69] Cost:0.0329s
Frame[70-79] Cost:0.0258s
Frame[80-89] Cost:0.0325s
Frame[90-99] Cost:0.0328s
```

2명사람 있는 경우

10 프레임마다 프레임당 평균 처리 시간 계산 단위(s)

```
Frame[0-9] Cost:0.0440s
Frame[10-19] Cost:0.0453s
Frame[20-29] Cost:0.0762s
Frame[30-39] Cost:0.0552s
Frame[40-49] Cost:0.0449s
Frame[50-59] Cost:0.0447s
Frame[60-69] Cost:0.0455s
Frame[70-79] Cost:0.0364s
Frame[80-89] Cost:0.0381s
Frame[90-99] Cost:0.0374s
```

7. 결론 및 향후 연구

본 논문은 opencv를 이용하여 카메라가 촬영한 스크린샷에 실시간으로 얼굴 스티칭 처리를 하고, 스티칭하기 전에 화면의 모든 얼굴 이미지를 모아 큰 이미지로 배열한 다음, 각 얼굴 이미지를 프레임 순서대로 확대하고, 그림은 좌표 위치가 텍스트 파일에 저장된다. 또한 실시간 얼굴 접합 영상은 그대로 원본 영상으로 복원되며 접합 과정에서 각 얼굴의 큰 영상을 캡처하여 저장하고 접합 범위의 영상은 점유 좌표 범위로 대체한다. 큰 화면의 각 얼굴을 복원할 수 있다. 나중에 어떻게 백업에서 사용하는 저장 공간이 작아지고 모자이크 처리 속도가 빨라지는지 연구할 계획이 있다.

References

- [1] <https://wikidocs.net/21849>
PyQt5 Tutorial - 파이썬으로 만드는 나만의 GUI 프로그램 검색일:2021.10.10
- [2] <https://github.com/kimjinho1/Real-time-face-recognition-and-mosaic-using-deep-learning>
딥러닝을 활용한 실시간 얼굴 인식과 모자이크 처리 검색일:2021.10.12
-