

OPENCV를 이용한 실시간 얼굴 인식 및 모자이크 처리 시스템



프로젝트 조원 명단

지능정보통신공학과 김웅빈 - 19100108

멀티미디어학과 나랑체체 -

이광호

장다훈

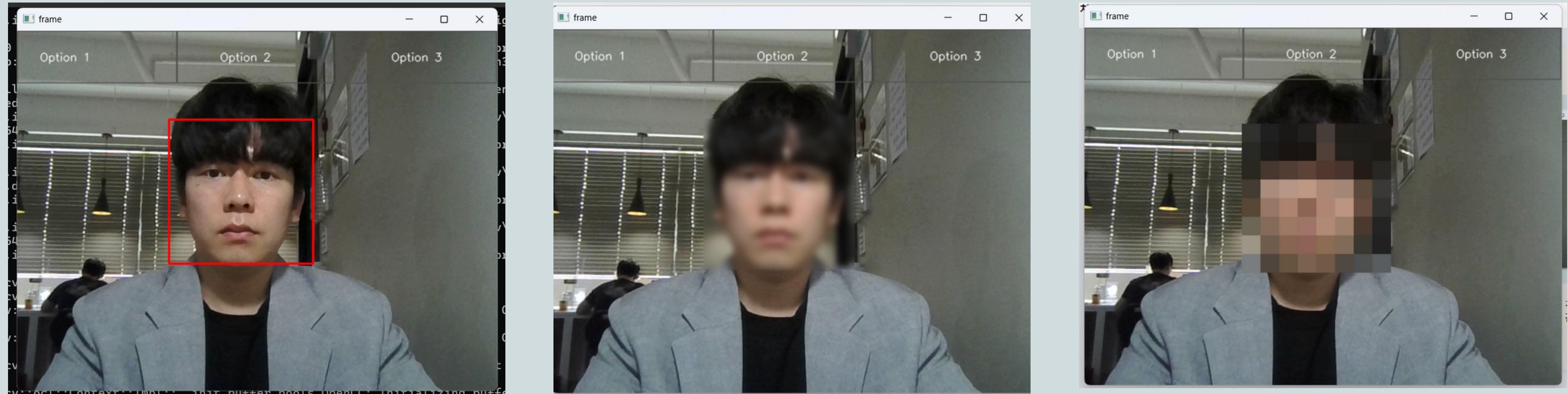
최재훈

CONTENT

- 1 개요
- 2 목적 및 필요성
- 3 구현 내용
- 4 주요 코드
- 5 시연 영상



개요



opencv 파일 안에 있는 haarcascade_frontalface_default.xml 파일을 사용하여 얼굴 객체를 탐지할 수 있도록 하였다.
인터페이스 상단에 3개의 옵션을 클릭 받을 수 있게 하여 사용자가 원하는 옵션을 실시간 감지된 얼굴 객체에 적용한다.

목적 및 필요성



일상에서 방송을 통해 얼굴이 모자이크 처리되어 방송에 녹화되는 모습들을 자주 볼 수 있다.

대표적인 인터넷 방송 플랫폼 ‘아프리카 티비’를 보면 스트리머가 야외 실시간 녹화 방송을 한 경우, 수많은 길거리 사람들이 자신의 얼굴을 모자이크 처리 안되고 그대로 영상에 송출되는 모습들을 많이 볼 수 있었다.

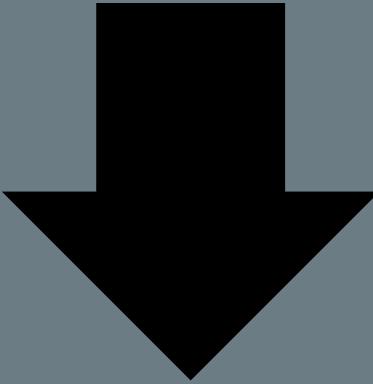
목적 및 필요성



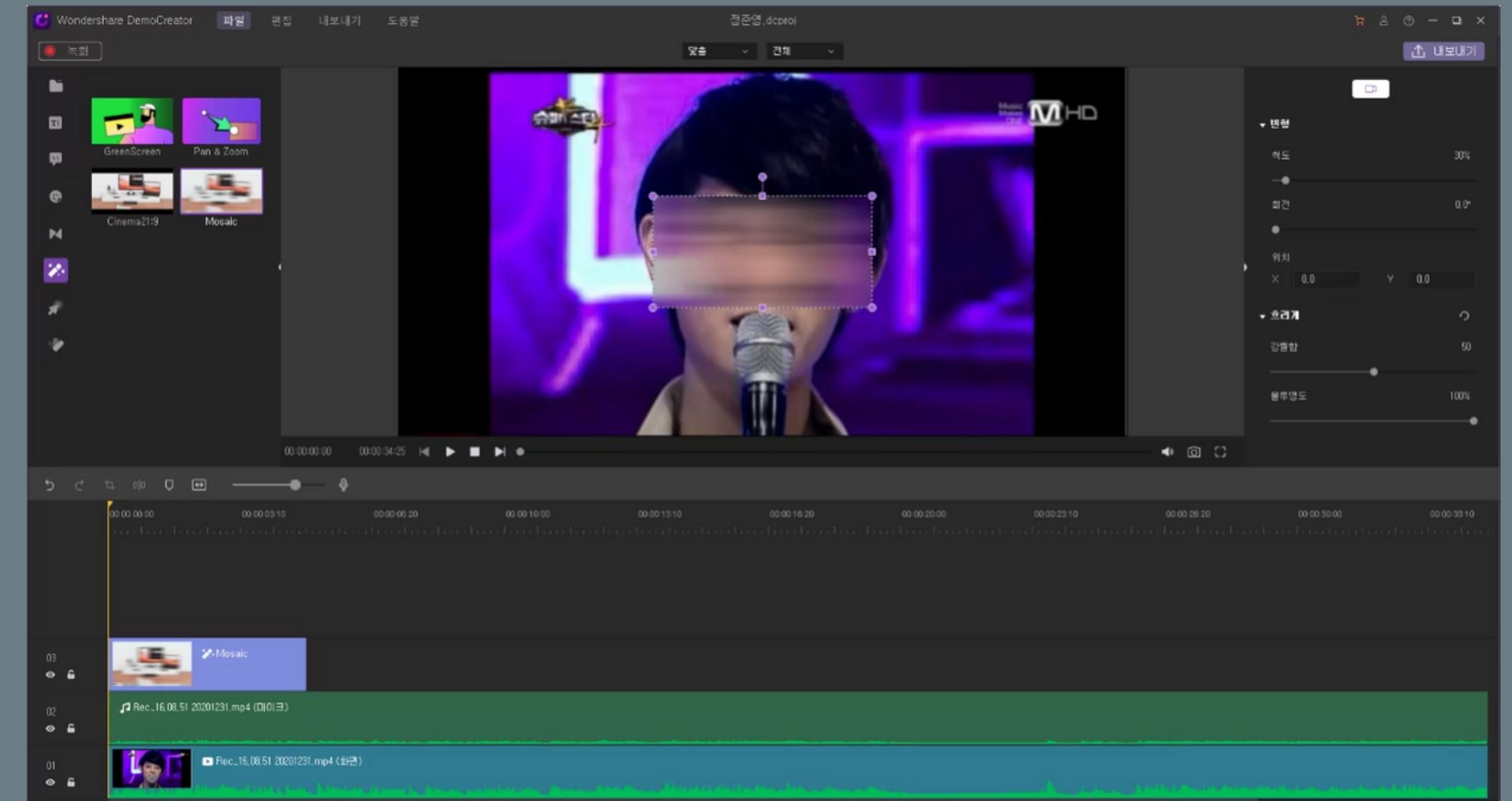
OTT 서비스 시장의 규모가 매년 증가하고 있음.

목적 및 필요성

모자이크를 사용하는 이유



초상권 등의 여러가지 이유로 사람들
의 얼굴을 모자이크 처리한다.



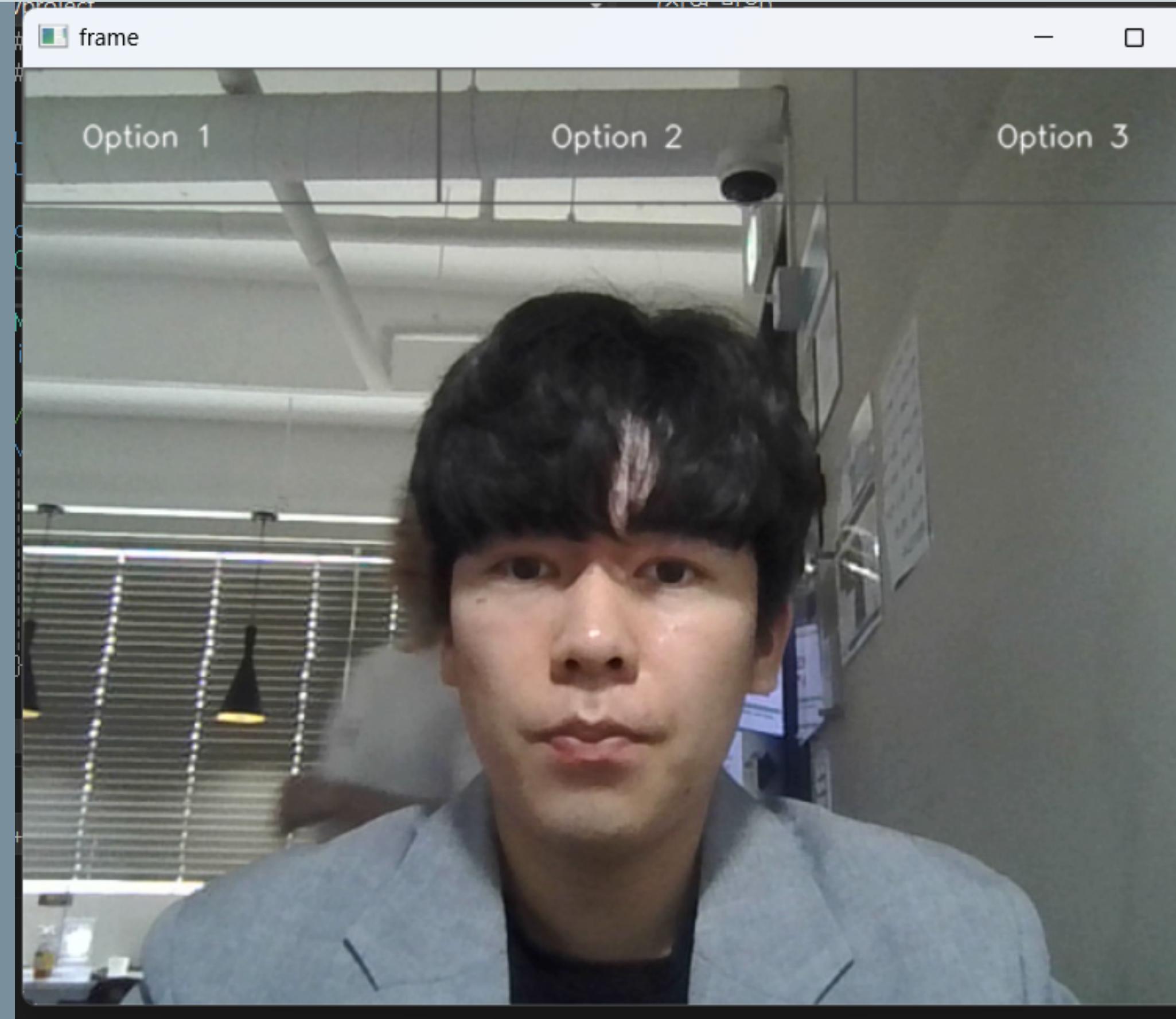
모자이크 처리 프로그램

모자이크 처리는 대부분 수작업으로 처리하지만 본 프로그램은 실시간 탐지된 객체를 옵션을 통하여 모자이크 처리를 한다.

구현 내용

프로그램 실행시 첫 화면 모습

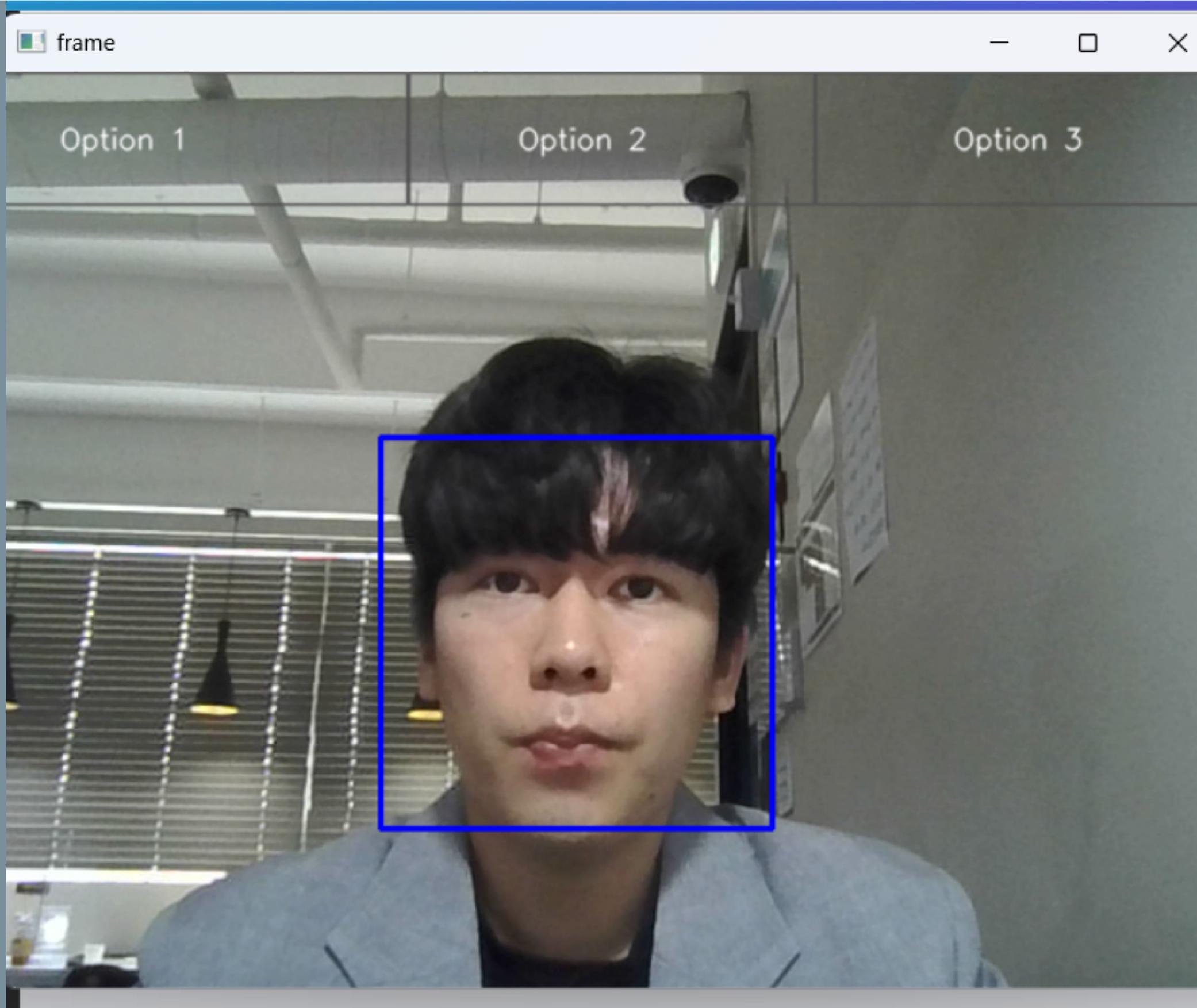
프로그램 실행 했을 경우 실행된 실시간 화면 모습이다.
화면의 상단 바에는 옵션을 클릭할 수 있는 3개의 바가 구
성되어있으면 3의 옵션 바는 각각의 기능을 구현한다.



구현 내용

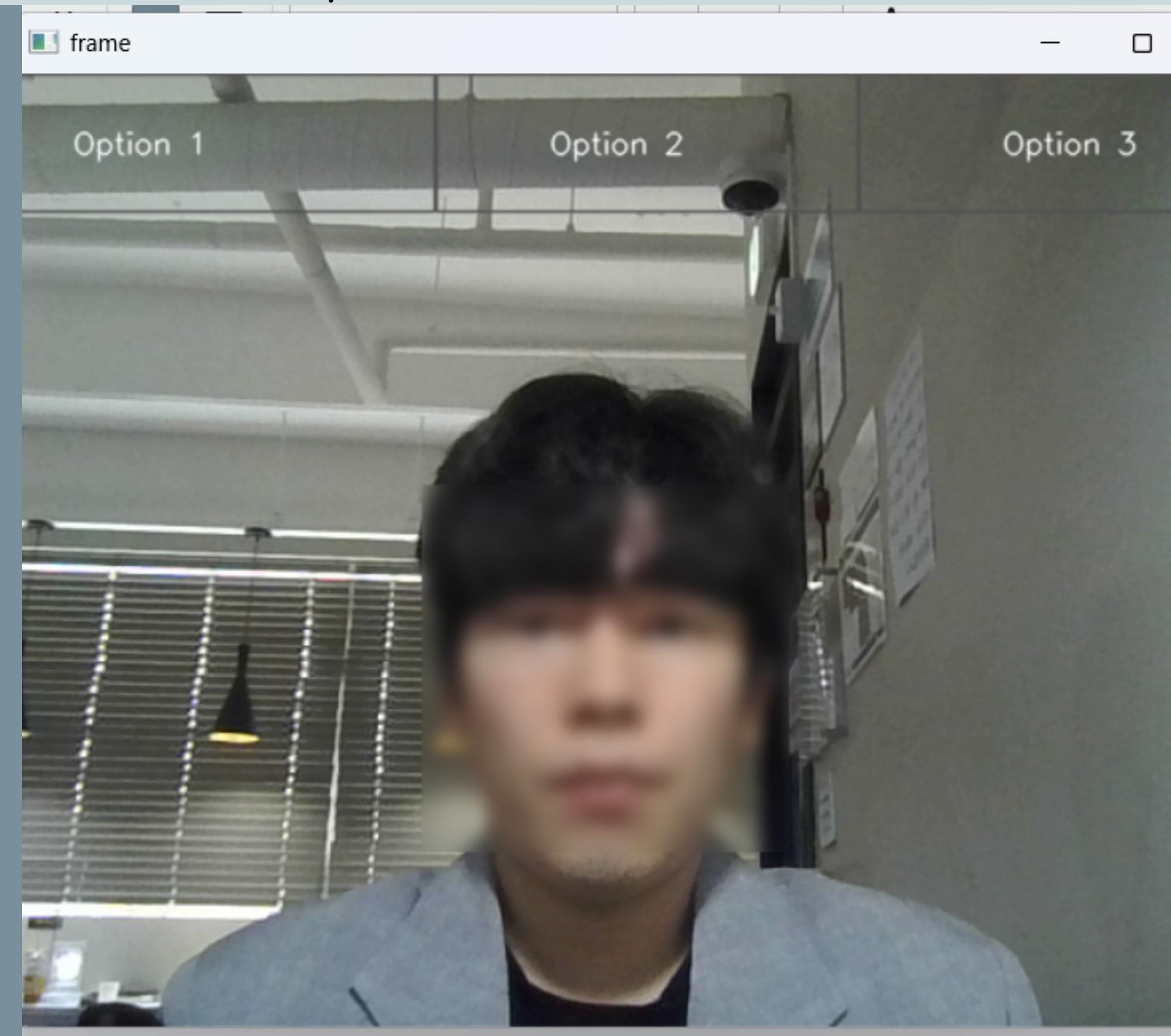
Option1 클릭한 경우의 기능 모습

상단에 있는 옵션1을 마우스로 클릭했을 경우, 감지된 객체의 영역을, 즉 얼굴의 영역을 파란색 사각형 테두리로 표시한다.



구현 내용

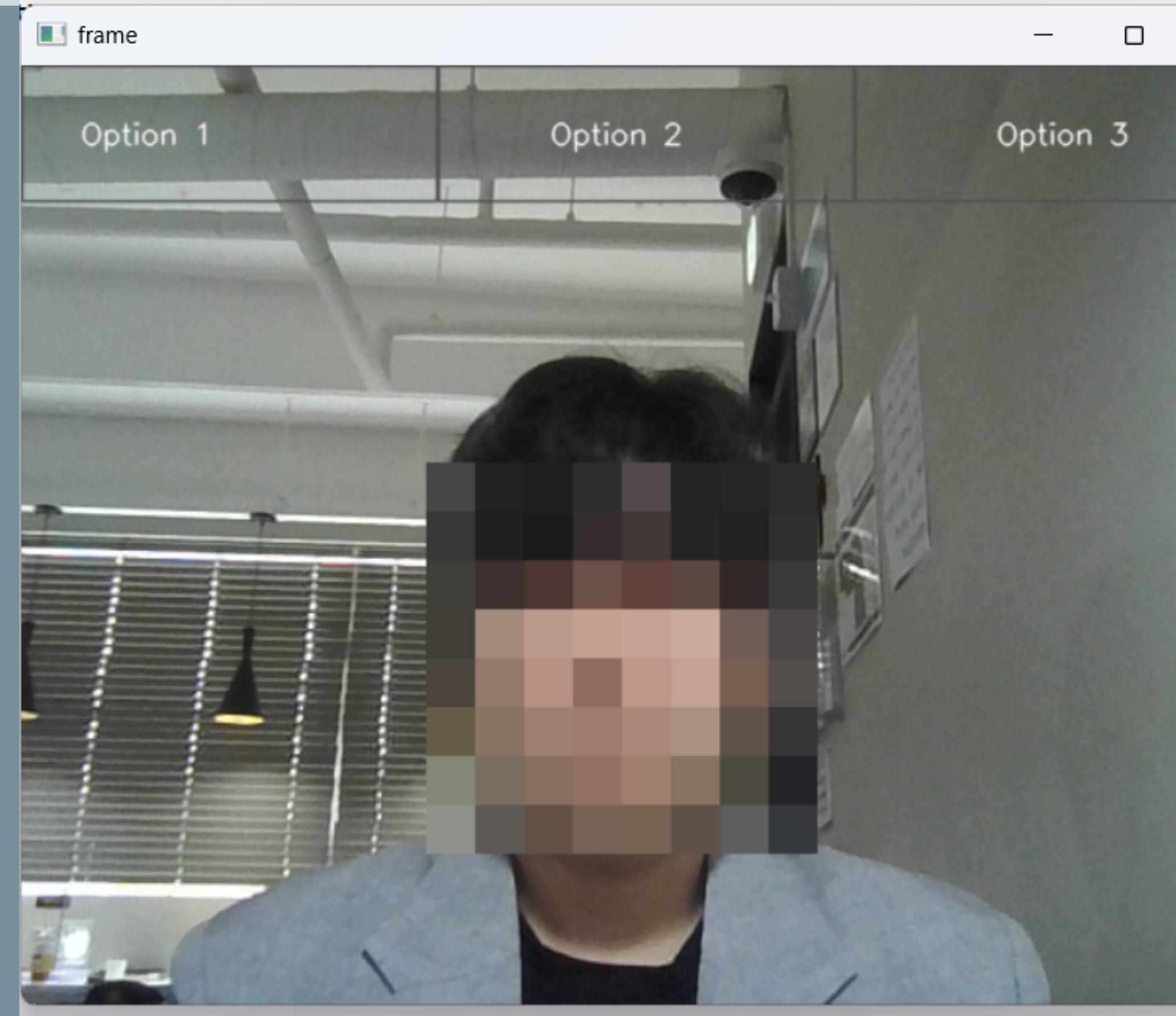
Option2 클릭한 경우의 기능 모습



상단에 있는 옵션2를 마우스로 클릭했을 경우, 감지된 객체의 영역을 가우시안 블러를 적용시킨기능이다.

구현 내용

Option3 클릭한 경우의 기능 모습



상단에 있는 옵션3을 마우스로 클릭했을 경우, 감지된 객체의 영역을 모자이크 처리를 하는 기능이다.

주요 코드

```
namedWindow("frame"); // 윈도우 생성  
setMouseCallback("frame", onMouse); // 마우스 콜백 함수 설정  
  
Scalar white = Scalar(255, 255, 255); // 텍스트 색상 (하양)  
Scalar gray = Scalar(80, 80, 80); // 사각형 테두리 색상 (회색)  
Scalar clicked = Scalar(0, 0, 0); // 사용되지 않음
```

```
Rect option1 = Rect(Point(0, 0), Size(213, 70)); // 첫 번째 옵션 영역  
Rect option2 = Rect(Point(213, 0), Size(213, 70)); // 두 번째 옵션 영역  
Rect option3 = Rect(Point(426, 0), Size(213, 70)); // 세 번째 옵션 영역
```

프로그램 실행시, 실행된 실시간 화면에서 상단에 위치한
옵션 형태 구현하기 위한 코드

```
// 옵션 텍스트와 사각형 테두리 그리기  
putText(frame, "Option 1", Point(30, 40), FONT_HERSHEY_SIMPLEX, 0.5, white, 1, LINE_AA);  
putText(frame, "Option 2", Point(270, 40), FONT_HERSHEY_SIMPLEX, 0.5, white, 1, LINE_AA);  
putText(frame, "Option 3", Point(498, 40), FONT_HERSHEY_SIMPLEX, 0.5, white, 1, LINE_AA);  
  
rectangle(frame, option1, gray, 1, LINE_AA);  
rectangle(frame, option2, gray, 1, LINE_AA);  
rectangle(frame, option3, gray, 1, LINE_AA);
```

주요 코드

```
// 마우스 이벤트 처리 함수
void onMouse(int event, int x, int y, int flags, void* userdata) {
    if (event == EVENT_LBUTTONDOWN) { // 마우스 왼쪽 버튼이 눌렸을 때
        if (x >= 0 && x < 213 && y >= 0 && y < 70) optionNum = 1; // 첫 번째 영역 선택
        else if (x >= 213 && x < 426 && y >= 0 && y < 70) optionNum = 2; // 두 번째 영역 선택
        else if (x >= 426 && x < 640 && y >= 0 && y < 70) optionNum = 3; // 세 번째 영역 선택
        else optionNum = 0; // 그 외 영역
    }
}
```

정해 놓은 옵션 영역에 마우스로 클릭시 이벤트 처리하는 기능의 코드

주요 코드

```
for (const Rect& face : faces) { // 각 얼굴에 대해
    switch (optionNum) { // 선택된 옵션에 따라 다른 효과 적용
        case 1:
            rectangle(frame, face, Scalar(255, 0, 0), 2); // 얼굴 주위에 빨간 사각형 그리기
            break;
        case 2:
            GaussianBlur(frame(face), frame(face), Size(25, 25), 0); // 얼굴 부위에 가우시안 블러 적용
            break;
        case 3:
            mosaic(face); // 얼굴 부위에 모자이크 적용
            break;
    }
}
```

```
// 모자이크 효과를 적용하는 함수
void mosaic(Rect roi) {
    int mosaicSize = 25; // 모자이크 크기
    for (int i = roi.y; i < roi.y + roi.height; i += mosaicSize) {
        for (int j = roi.x; j < roi.x + roi.width; j += mosaicSize) {
            Rect rect = Rect(j, i, mosaicSize, mosaicSize) & Rect(0, 0, frame.cols, frame.rows);
            Scalar color = mean(frame(rect)); // 해당 영역의 평균 색상 계산
            rectangle(frame, rect, color, FILLED); // 계산된 색상으로 사각형 채우기
        }
    }
}
```

각 옵션별, 기능을 구현하기 위한 코드

(switch - case 문 사용)

case1: 객체 감지 영역 빨간 사각형 테두리 그리기

case2: 객체 감지 영역에 가우시안 블러 적용

case3: 객체 감지 영역에 모자이크 처리 적용

참고 문헌 및 사진 출처

<참고 문헌>

[https://velog.io/@mouse0429/openCVVisual-Studio-OpenCV-%EC%84%A4%EC%B9%98_\(Visual%20Studio%20에%20opencv%20설치%20문제\)](https://velog.io/@mouse0429/openCVVisual-Studio-OpenCV-%EC%84%A4%EC%B9%98_(Visual%20Studio%20에%20opencv%20설치%20문제))

<a href="https://m.blog.naver.com/PostView.naver?blogId=sjangos1234&logNo=222375856975&proxyReferer= (객체 인식
영역 테두리 코드 문헌)

<사진 출처>

<https://dc.wondershare.kr/screen-recorder/how-to-add-blur-in-video.html>

<PPT 템플릿 출처>

https://www.canva.com/ko_kr/create/