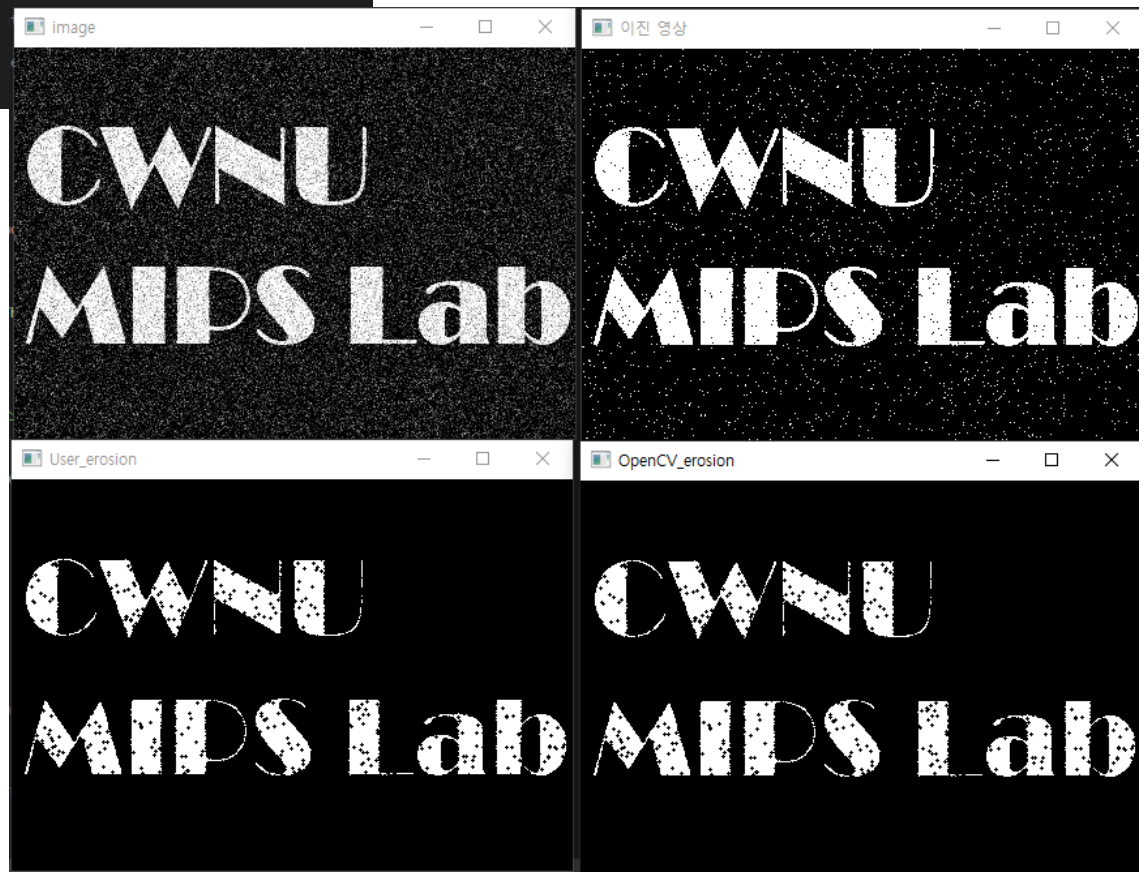
영상처리 실제 - 7주차 실습

2023254009 최현동

: 12.형태학적 연산 – p.6

```cpp
//마스크 원소와 마스크 범위 입력화소 간의 일치 여부 체크
bool check_match(Mat img, Point start, Mat mask, int mode = 0)
{
    for (int u = 0; u < mask.rows; u++)
    {
        for (int v = 0; v < mask.cols; v++)
        {
            Point pt(v, u);                     //순회좌표
            int m = mask.at<uchar>(pt);         //마스크 계수
            int p = img.at<uchar>(start + pt);  //해당 위치 입력화소

            bool ch = (p == 255);               //일치 여부 비교
            if (m == 1 && ch == mode)           //mode 0이면 침식, 1이면 팽창
            {
                return  false;
            }
        }
    }
    return true;
}

//침식 연산
void erosion(Mat img, Mat& dst, Mat mask)
{
    dst = Mat(img.size(), CV_8U, Scalar(0));
    if (mask.empty())   mask = Mat(3, 3, CV_8UC1, Scalar(0));

    Point h_m = mask.size() / 2; //마스크 절반 크기

    for (int i = h_m.y; i < img.rows - h_m.y; i++)
    {
        for (int j = h_m.x; j < img.cols - h_m.x; j++)
        {
            Point start = Point(j, i) - h_m;
            bool  check = check_match(img, start, mask, 0); //원소 일치여부 비교
            dst.at<uchar>(i, j) = (check) ? 255 : 0;        // 출력화소 저장
        }
    }
}
```

```cpp
//12 - 형태학적 - 침식연산 - p.6
#if 1
    Mat image = imread("D:\\999.Image\\morph_test1.jpg", 0);
    CV_Assert(image.data);

    Mat th_img, dst1, dst2;
    threshold(image, th_img, 128, 255, THRESH_BINARY); //영상이진화

    uchar data[] = { 0,1,0,
                     1,1,1,
                     0,1,0 };
    Mat mask(3, 3, CV_8UC1, data); //마스크 선언 및 초기화

    erosion(th_img, dst1, (Mat)mask);   //사용자정의 침식 함수
    morphologyEx(th_img, dst2, MORPH_ERODE, mask); //OpenCV 침식 함수

    imshow("image", image), imshow("이진 영상", th_img);
    imshow("User_erosion", dst1);
    imshow("OpenCV_erosion", dst2);

    waitKey();
#endif
```

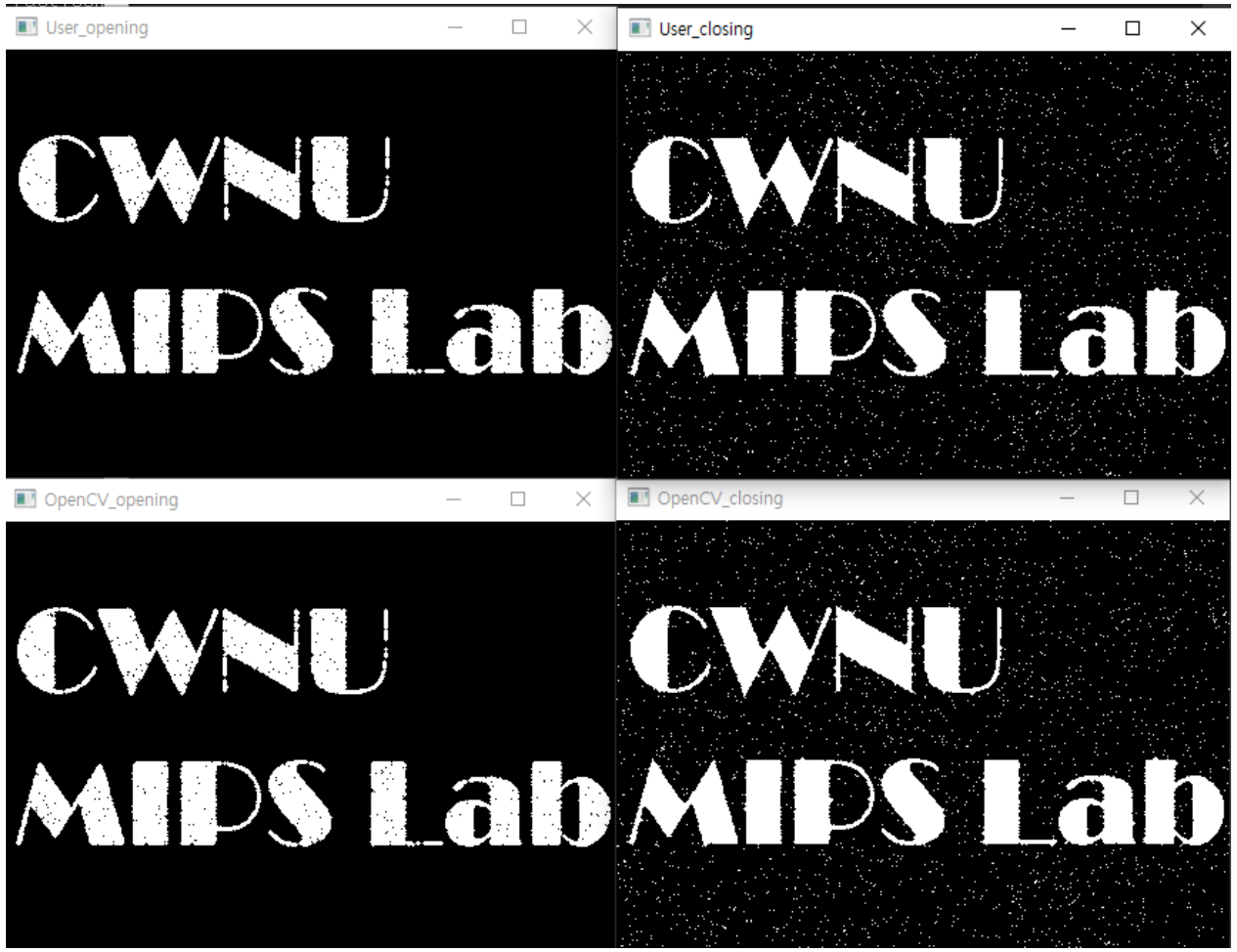


: 12.형태학적 연산 – p.10

```cpp
//팽창연산
void dilation(Mat img, Mat& dst, Mat mask)
{
    dst = Mat(img.size(), CV_8U, Scalar(0));
    if (mask.empty())   mask = Mat(3, 3, CV_8UC1, Scalar(0));

    Point h_m = mask.size() / 2;
    for (int i = h_m.y; i < img.rows - h_m.y; i++)
    {
        for (int j = h_m.x; j < img.cols - h_m.x; j++)
        {
            Point start = Point(j, i) - h_m;
            bool  check = check_match(img, start, mask, 1); // 원소 일치여부 비교
            dst.at<uchar>(i, j) = (check) ? 0 : 255;           // 침식연산과 반대
        }
    }
}

    //12 - 형태학적 - 팽창연산 - p.10
#if 1
    Mat image = imread("D:\\999.Image\\morph_test1.jpg", 0);
    CV_Assert(image.data);

    Mat th_img, dst1, dst2;
    threshold(image, th_img, 128, 255, THRESH_BINARY);

    Matx < uchar, 3, 3> mask;
    mask << 0, 1, 0,
        1, 1, 1,
        0, 1, 1;

    dilation(th_img, dst1, (Mat)mask);
    morphologyEx(th_img, dst2, MORPH_DILATE, mask);

    imshow("image", image);
    imshow("User_dilation", dst1);
    imshow("OpenCV_dilation", dst2);
    waitKey();
#endif
```

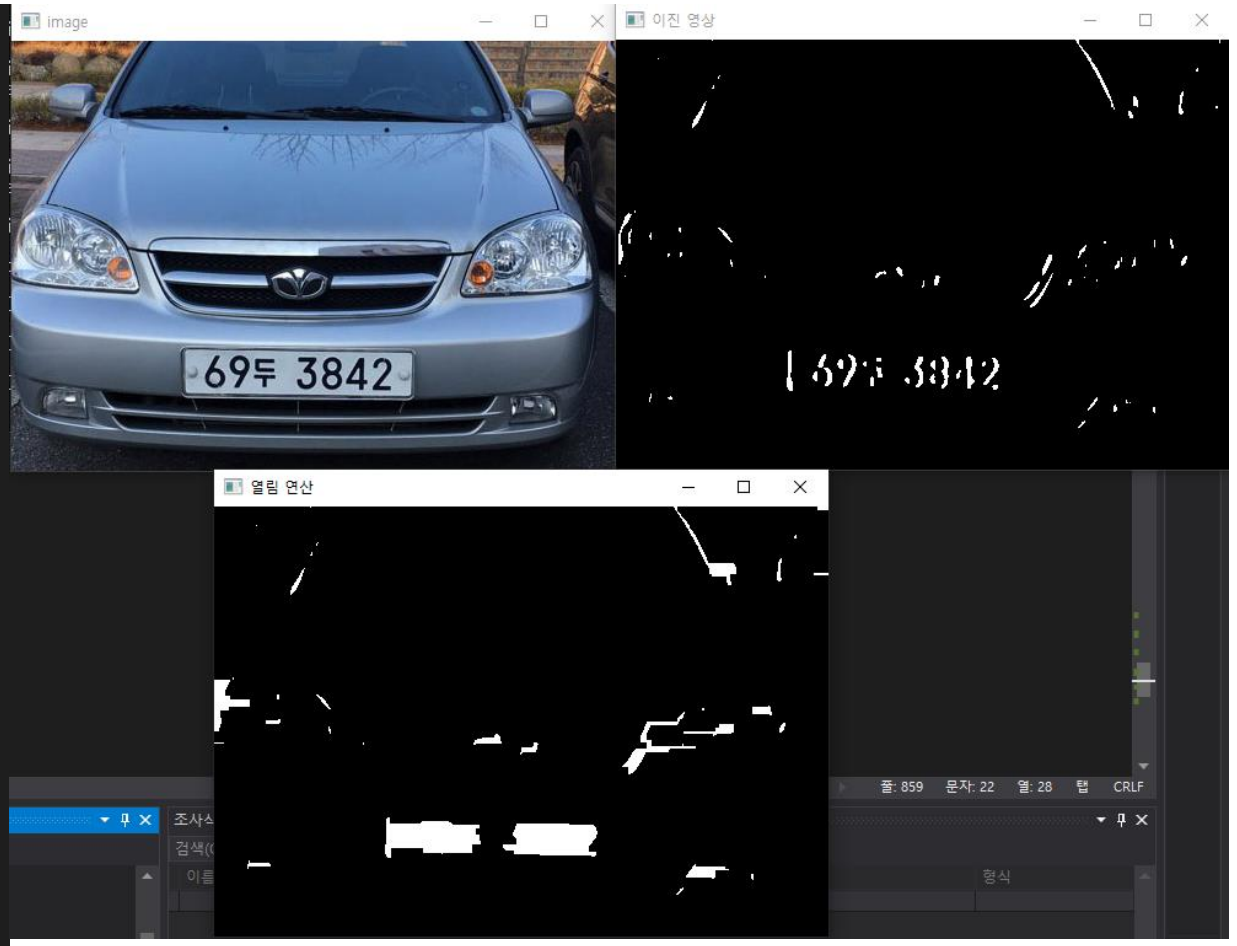
: 12.형태학적 연산 – p.18

```cpp
//열림연산
void opening(Mat img, Mat& dst, Mat mask)
{
    Mat tmp;
    erosion(img, tmp, mask);
    dilation(tmp, dst, mask);
}

//닫힘연산
void closing(Mat img, Mat& dst, Mat mask)
{
    Mat tmp;
    dilation(img, tmp, mask);
    erosion(tmp, dst, mask);
}

    //12 - 형태학적 - 열림/닫힘연산 - p.18
#if 1
    Mat image = imread("D:\\999.Image\\morph_test1.jpg", 0);
    CV_Assert(image.data);
    Mat th_img, dst1, dst2, dst3, dst4;
    threshold(image, th_img, 128, 255, THRESH_BINARY);

    Matx < uchar, 3, 3> mask;
    mask << 0, 1, 0,
        1, 1, 1,
        0, 1, 0;

    opening(th_img, dst1, (Mat)mask);  //사용자 정의 함수 열림함수 호출
    closing(th_img, dst2, (Mat)mask);
    morphologyEx(th_img, dst3, MORPH_OPEN, mask); //OpenCV 열림함수
    morphologyEx(th_img, dst4, MORPH_CLOSE, mask);//OpenCV 닫힘함수

    imshow("User_opening", dst1);
    imshow("User_closing", dst2);
    imshow("OpenCV_opening", dst3);
    imshow("OpenCV_closing", dst4);
    waitKey();
#endif
```



: 12.형태학적 연산 – p.23

```cpp
    //12 - 형태학적 - 차량번호판 검출 - p.23
#if 1
    while (1)
    {
        int no;
        cout << "차량 영상 번호( 0:종료 ) : ";
        cin >> no;
        if (no == 0) break;

        string fname = format("D:\\999.Image\\test_car\\%02d.jpg", no);
        Mat image = imread(fname, 1);
        if (image.empty())
        {
            cout << to_string(no) + "번 영상 파일이 없습니다. " << endl;
            continue;
        }

        Mat gray, sobel, th_img, morph;
        Mat kernel(5, 25, CV_8UC1, Scalar(1));      // 닫힘 연산 마스크
        cvtColor(image, gray, COLOR_BGR2GRAY);      // 명암도 영상 변환

        blur(gray, gray, Size(5, 5));               // 블러링
        Sobel(gray, gray, CV_8U, 1, 0, 3);          // 소벨 에지 검출

        threshold(gray, th_img, 120, 255, THRESH_BINARY);   // 이진화 수행
        morphologyEx(th_img, morph, MORPH_CLOSE, kernel);   // 닫힘 연산 수행

        imshow("image", image);
        imshow("이진 영상", th_img);
        imshow("열림 연산", morph);
        waitKey();
    }
#endif
```



: 12.형태학적 연산 – p.27

```cpp
    //12 - 형태학적 - 골격화 - p.27
#if 1
    Mat img = imread("D:\\999.Image\\letterb.png", IMREAD_GRAYSCALE);
    threshold(img, img, 127, 255, THRESH_BINARY);

    imshow("src", img);
    Mat skel(img.size(), CV_8UC1, Scalar(0));
    Mat element = getStructuringElement(MORPH_CROSS, Size(3, 3));
    Mat temp, eroded;
    do
    {
        erode(img, eroded, element);
        dilate(eroded, temp, element);
        subtract(img, temp, temp);
        bitwise_or(skel, temp, skel);
        eroded.copyTo(img);

    } while ((countNonZero(img) != 0));

    imshow("result", skel);
    waitKey();
#endif
```