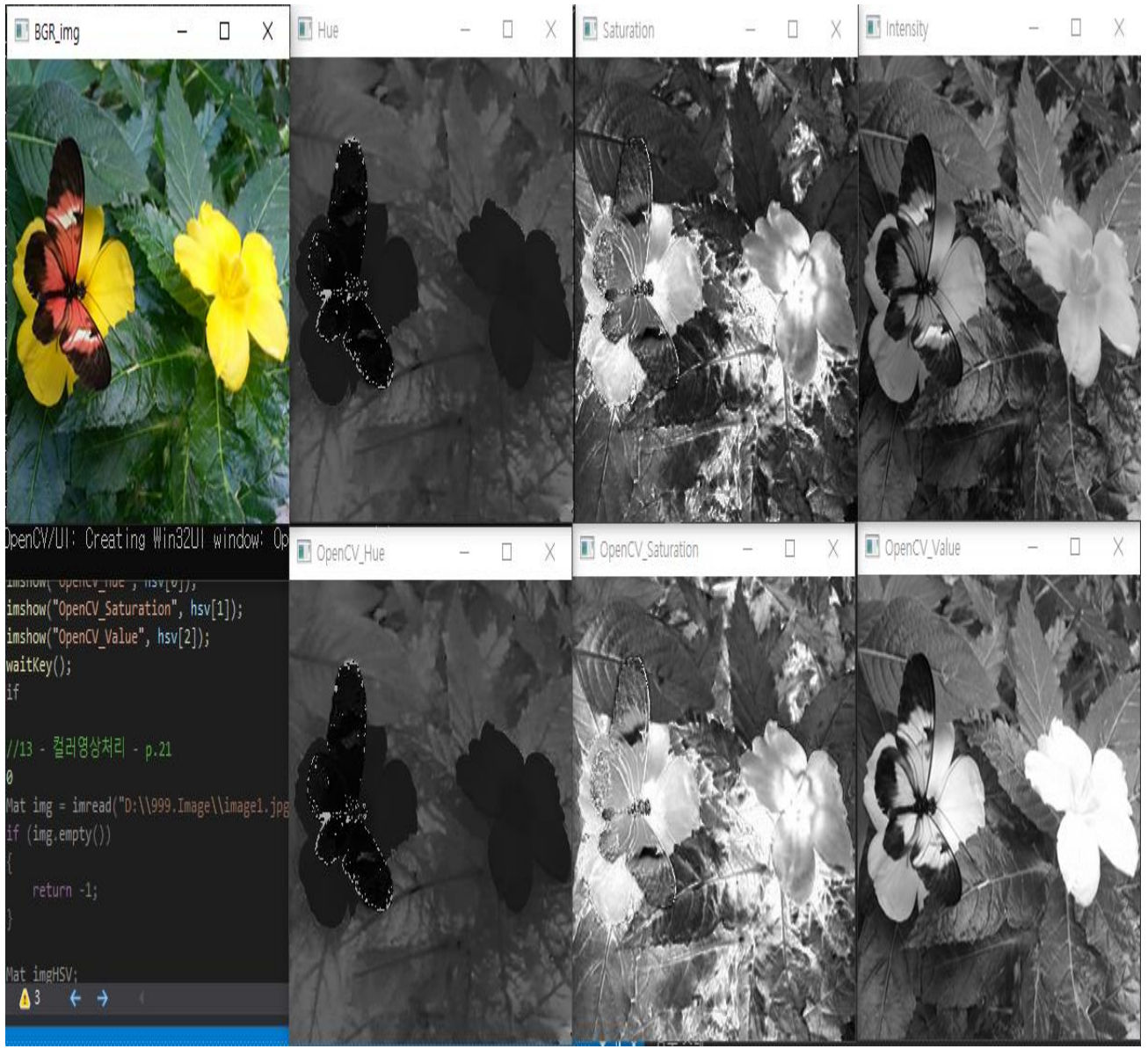
# 영상처리 실제 - 9주차 실습

2023254009 최현동

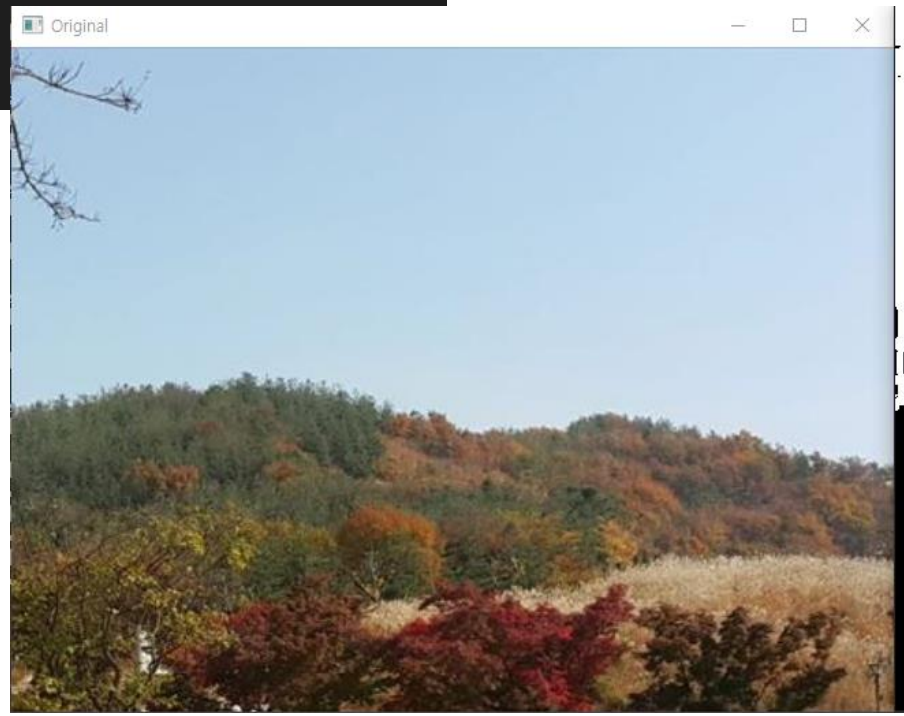
: 13.컬러영상처리 – p.16 ~ 17

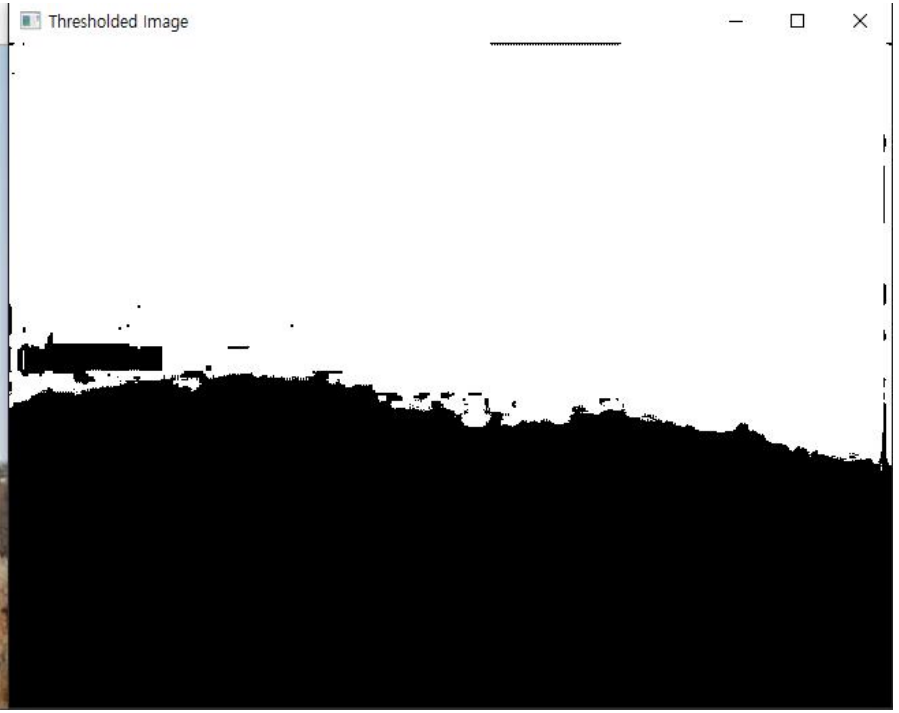
```
void bgr2hsi(Mat img, Mat& hsv)
{
    hsv = Mat(img.size(), CV_32FC3);
    for (int i = 0; i < img.rows; i++)
    {
        for (int j = 0; j < img.cols; j++)
        {
            float B = img.at<Vec3b>(i, j)[0];
            float G = img.at<Vec3b>(i, j)[1];
            float R = img.at<Vec3b>(i, j)[2];

            float s = 1 - 3 * min(R, min(G, B)) / (R + B + G);
            float v = (R + B + G) / 3.0f;

            float tmp1 = ((R - G) + (R - B)) * 0.5f;
            float tmp2 = sqrt((R - G) * (R - B) + (G - B) * (G - B));
            float angle = acos(tmp1 / tmp2) * (180.f / CV_PI);
            float h = (B <= G) ? angle : 360 - angle;
            hsv.at<Vec3f>(i, j) = Vec3f(h / 2, s * 255, v);
        }
    }
    hsv.convertTo(hsv, CV_8U);
}
    //13 - 컬러영상처리 - p.16
#if 1
    Mat BGR_img = imread("D:\\999.Image\\color_space.jpg", IMREAD_COLOR);
    CV_Assert(BGR_img.data);
    Mat HSI_img, HSV_img, hsi[3], hsv[3];

    bgr2hsi(BGR_img, HSI_img);
    cvtColor(BGR_img, HSV_img, COLOR_BGR2HSV);
    split(HSI_img, hsi);
    split(HSV_img, hsv);

    imshow("BGR_img", BGR_img);
    imshow("Hue", hsi[0]);
    imshow("Saturation", hsi[1]);
    imshow("Intensity", hsi[2]);
    imshow("OpenCV_Hue", hsv[0]);
    imshow("OpenCV_Saturation", hsv[1]);
    imshow("OpenCV_Value", hsv[2]);
    waitKey();
#endif
```



: 13.컬러영상처리 – p.21

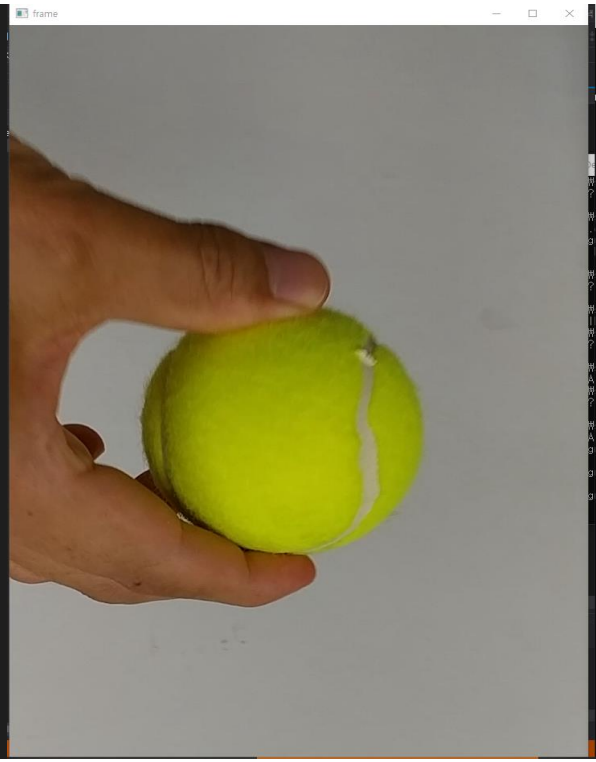
```
//13 - 컬러영상처리 - p.21
#if 1
Mat img = imread("D:\\999.Image\\image1.jpg", IMREAD_COLOR);
if (img.empty())
{
    return -1;
}

Mat imgHSV;
cvtColor(img, imgHSV, COLOR_BGR2HSV);

Mat imgThresholded;
inRange(imgHSV, Scalar(100, 0, 0), Scalar(120, 255, 255), imgThresholded);

imshow("Thresholded Image", imgThresholded);
imshow("Original", img);

waitKey(0);
#endif
```



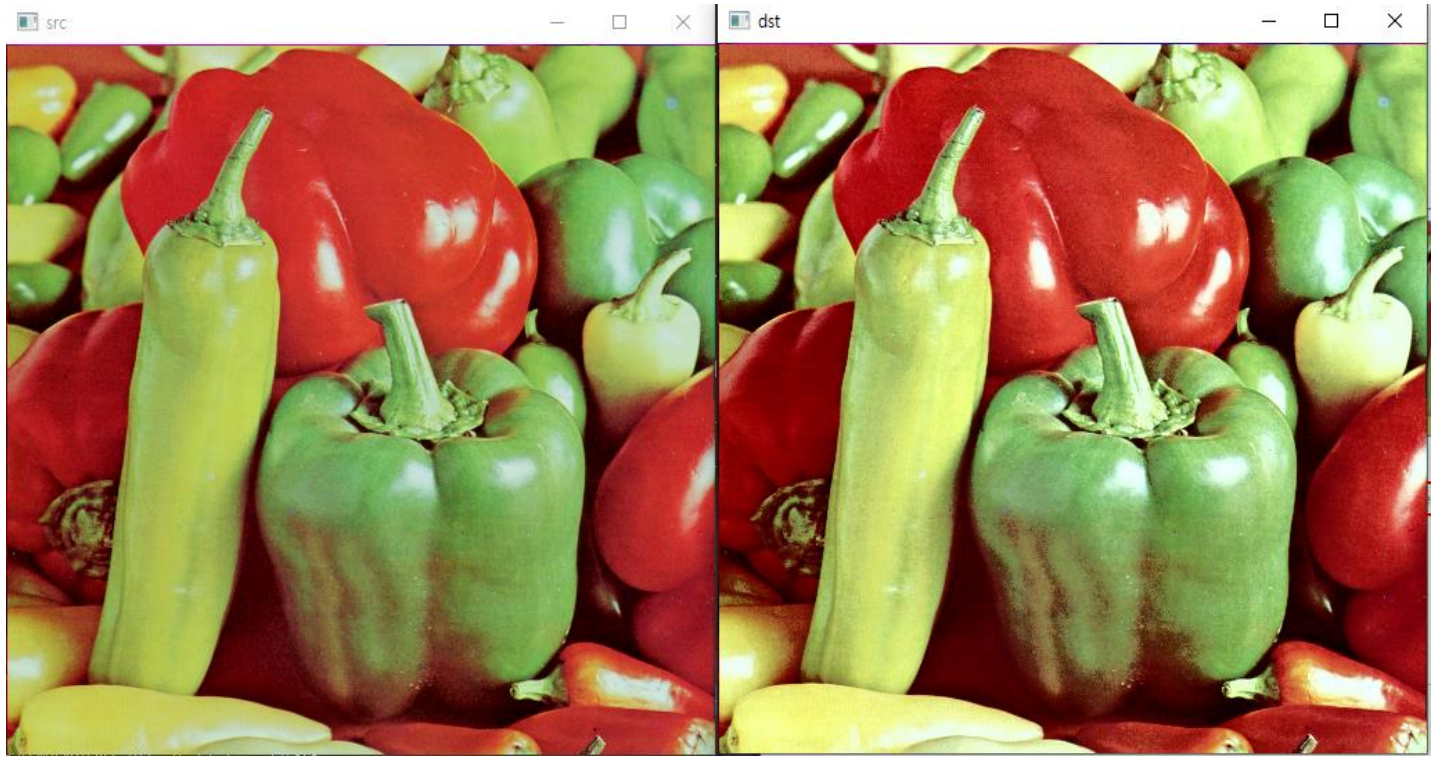
: 13.컬러영상처리 – p.22

```cpp
    //13 - 컬러영상처리 - p.22
#if 1
    VideoCapture cap("D:\\999.Image\\tennis_ball.mp4");
    if (!cap.isOpened())
    {
        return -1;
    }
    for (;;)
    {
        Mat imgHSV;
        Mat frame;
        cap >> frame;
        cvtColor(frame, imgHSV, COLOR_BGR2HSV);

        Mat imgThresholded;
        inRange(imgHSV, Scalar(30, 10, 10), Scalar(38, 255, 255), imgThresholded);

        imshow("frame", frame);
        imshow("dst", imgThresholded);
        if (waitKey(30) >= 0) break;
    }
    waitKey();
#endif
```




: 13.컬러영상처리 – p.27

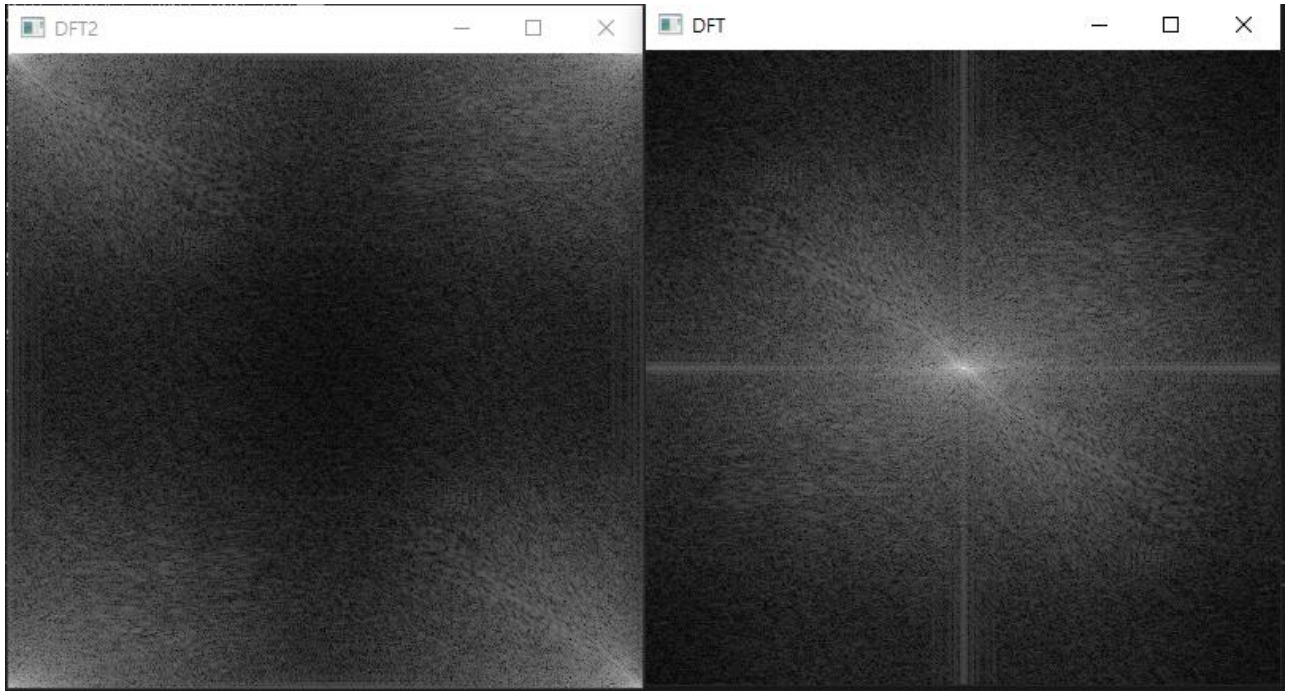
```
    //13 - 컬러영상처리 - p.27
#if 1
    Mat src = imread("D:\\999.Image\\pepper.bmp", IMREAD_COLOR);
    if (src.empty())
    {
        cerr << "Image Load Failed!" << endl;
    }
    Mat src_ycrcb;
    cvtColor(src, src_ycrcb, COLOR_BGR2YCrCb);

    vector<Mat> ycrcb_planes;
    split(src_ycrcb, ycrcb_planes);

    equalizeHist(ycrcb_planes[0], ycrcb_planes[0]);

    Mat dst_ycrcb;
    merge(ycrcb_planes, dst_ycrcb);

    Mat dst;
    cvtColor(dst_ycrcb, dst, COLOR_YCrCb2BGR);

    imshow("src", src);
    imshow("dst", dst);
    waitKey();
#endif
```



# : 14.주파수영역처리 – p.10

```
//14 - 주파수영역처리 - p.10
void displayDFT(Mat& src)
{
    Mat image_array[2] = { Mat::zeros(src.size(), CV_32F), Mat::zeros(src.size(), CV_32F) };
    // ① DFT 결과 영상을 2개의 영상으로 분리한다.
    split(src, image_array);

    Mat mag_image;
    // ② 푸리에 변환 계수들의 절대값을 계산한다.
    magnitude(image_array[0], image_array[1], mag_image);

    // ③ 푸리에 변환 계수들은 상당히 크기 때문에 로그 스케일로 변환한다.
    // 0값이 나오지 않도록 1을 더해준다.
    mag_image += Scalar::all(1);
    log(mag_image, mag_image);

    // ④ 0에서 255로 범위로 정규화한다.
    normalize(mag_image, mag_image, 0, 1, NORM_MINMAX);
    imshow("DFT", mag_image);
}

void shuffleDFT(Mat& src)
{
    int cX = src.cols / 2;
    int cY = src.rows / 2;

    Mat q1(src, Rect(0, 0, cX, cY));
    Mat q2(src, Rect(cX, 0, cX, cY));
    Mat q3(src, Rect(0, cY, cX, cY));
    Mat q4(src, Rect(cX, cY, cX, cY));

    Mat tmp;
    q1.copyTo(tmp);
    q4.copyTo(q1);
    tmp.copyTo(q4);
    q2.copyTo(tmp);
    q3.copyTo(q2);
    tmp.copyTo(q3);
}
```

```
//14 - 주파수영역처리 - p.10
if 1
    Mat src = imread("D:\\999.Image\\lenna.jpg", IMREAD_GRAYSCALE);
    Mat src_float;

    // 그레이스케일 영상을 실수 영상으로 변환한다.
    src.convertTo(src_float, CV_32FC1, 1.0 / 255.0);
    Mat dft_image;
    dft(src_float, dft_image, DFT_COMPLEX_OUTPUT);
    shuffleDFT(dft_image);
    displayDFT(dft_image);
    waitKey();
endif
```



: 14.주파수영역처리 – p.12

```
    //14 - 주파수영역처리 - p.12
#if 1
    Mat img = imread("D:\\999.Image\\lenna.jpg", IMREAD_GRAYSCALE);

    Mat img_float, dft1, inversedft, inversedft1;
    img.convertTo(img_float, CV_32F);
    dft(img_float, dft1, DFT_COMPLEX_OUTPUT);

    // 역변환을 수행한다.
    idft(dft1, inversedft, DFT_SCALE | DFT_REAL_OUTPUT);

    inversedft.convertTo(inversedft1, CV_8U);
    imshow("invertedfft", inversedft1);

    imshow("original", img);
    waitKey();
#endif
```