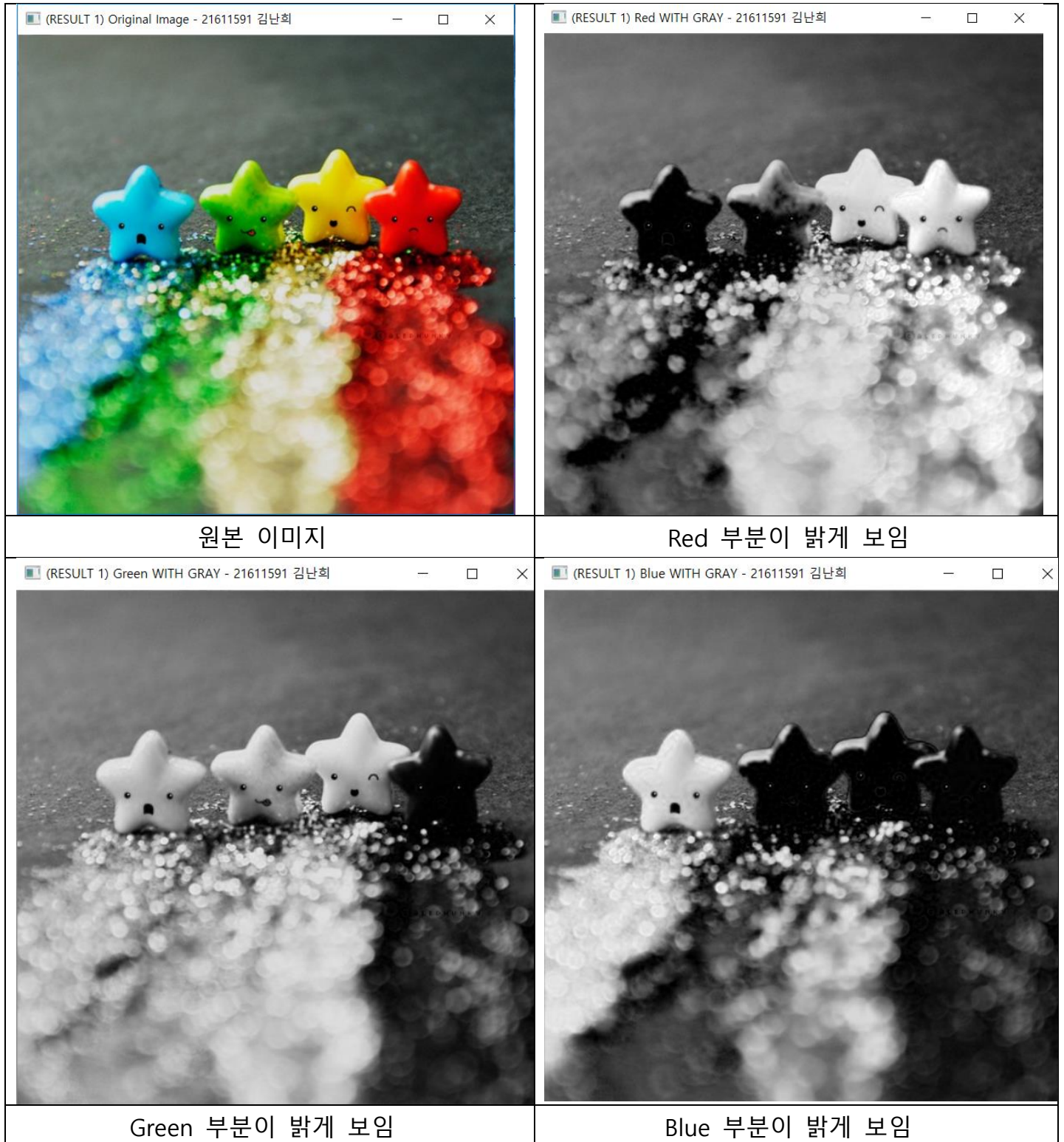


[Computer Vision Programming] Lab3. Color Space

전자공학과 21611591 김난희

1. RGB color space 채널별 도식: 흑백 형태, 컬러형태

- 흑백 형태 결과 사진



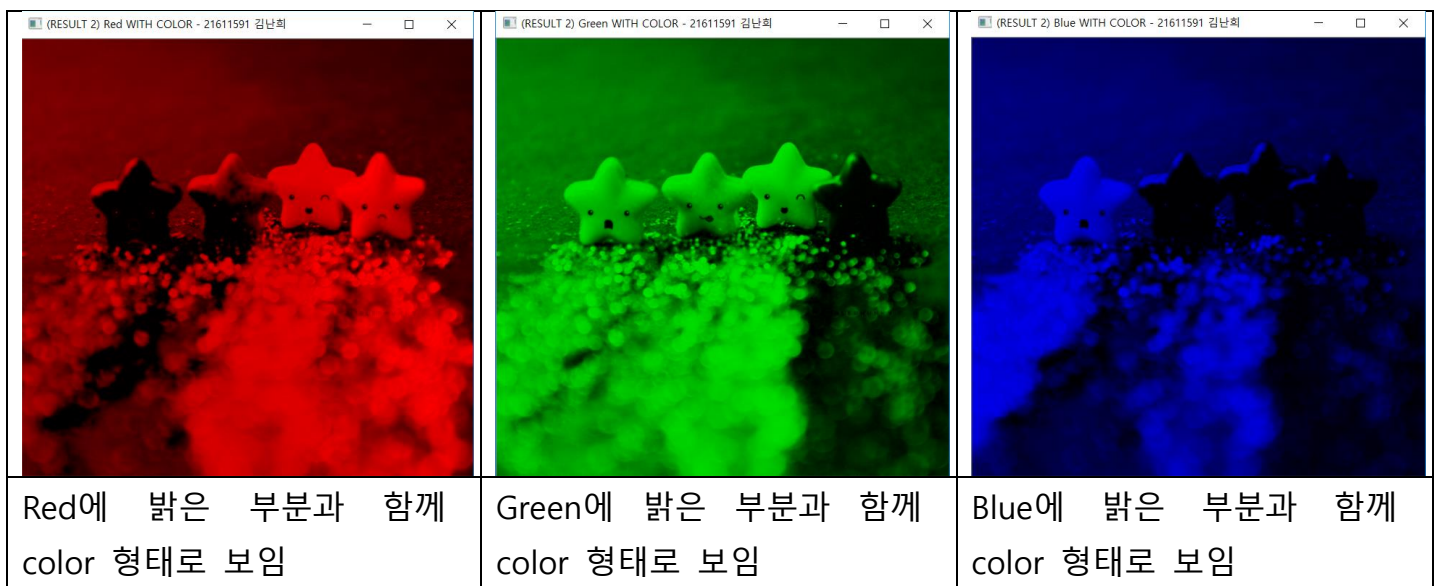
- 흑백 형태 결과 소스 코드

```

12  /*****1.RGB color space 채널별 도시: 흑백형태*****/
13  Mat image = imread("star.jpg");// color load
14
15  //3채널 분리에 사용할 vector
16  vector<Mat> channels;
17
18  //Partition image into three channel planes
19  split(image, channels);
20
21  imshow("(RESULT 1) Original Image - 21611591 김난희", image);
22  imshow("(RESULT 1) Red WITH GRAY - 21611591 김난희", channels[2]);
23  imshow("(RESULT 1) Green WITH GRAY - 21611591 김난희", channels[1]);
24  imshow("(RESULT 1) Blue WITH GRAY - 21611591 김난희", channels[0]);
25

```

- 컬러 형태 결과 사진



- 컬러 형태 결과 소스 코드

```

26  /*****1.RGB color space 채널별 도시: 컬러형태*****/
27
28  //원본 이미지의 크기만큼 1채널 unsigned char type으로 0을 만들어줌
29  Mat zero = Mat::zeros(image.rows, image.cols, CV_8UC1);
30
31  //BGR 순서로 zero와 조합하여 3채널을 만들어줄 배열을 만들어줌
32  Mat B[] = { channels[0],zero,zero };
33  Mat G[] = { zero,channels[1],zero };
34  Mat R[] = { zero,zero,channels[2] };
35
36  //RGB 중 한 가지 색상(3채널로 된)을 나타내줄 OUTPUT 선언
37  vector<Mat> output(3);
38
39  //위에 선언한 RGB에 대한 각각의 색상을 3채널로 합성하여 output에 넣음
40  //merge함수는 배열에 따라 채널 수를 정하여 하나의 컬러 이미지로 합성하여 보여줌
41  merge(R, 3, output[2]);
42  merge(G, 3, output[1]);
43  merge(B, 3, output[0]);
44
45  imshow("(RESULT 2) Red WITH COLOR - 21611591 김난희", output[2]);
46  imshow("(RESULT 2) Green WITH COLOR - 21611591 김난희", output[1]);
47  imshow("(RESULT 2) Blue WITH COLOR - 21611591 김난희", output[0]);

```

2. RGB to gray 변환: opencv 내장 함수, 구현 코드 비교

- 내장 함수와 구현 코드 결과 사진

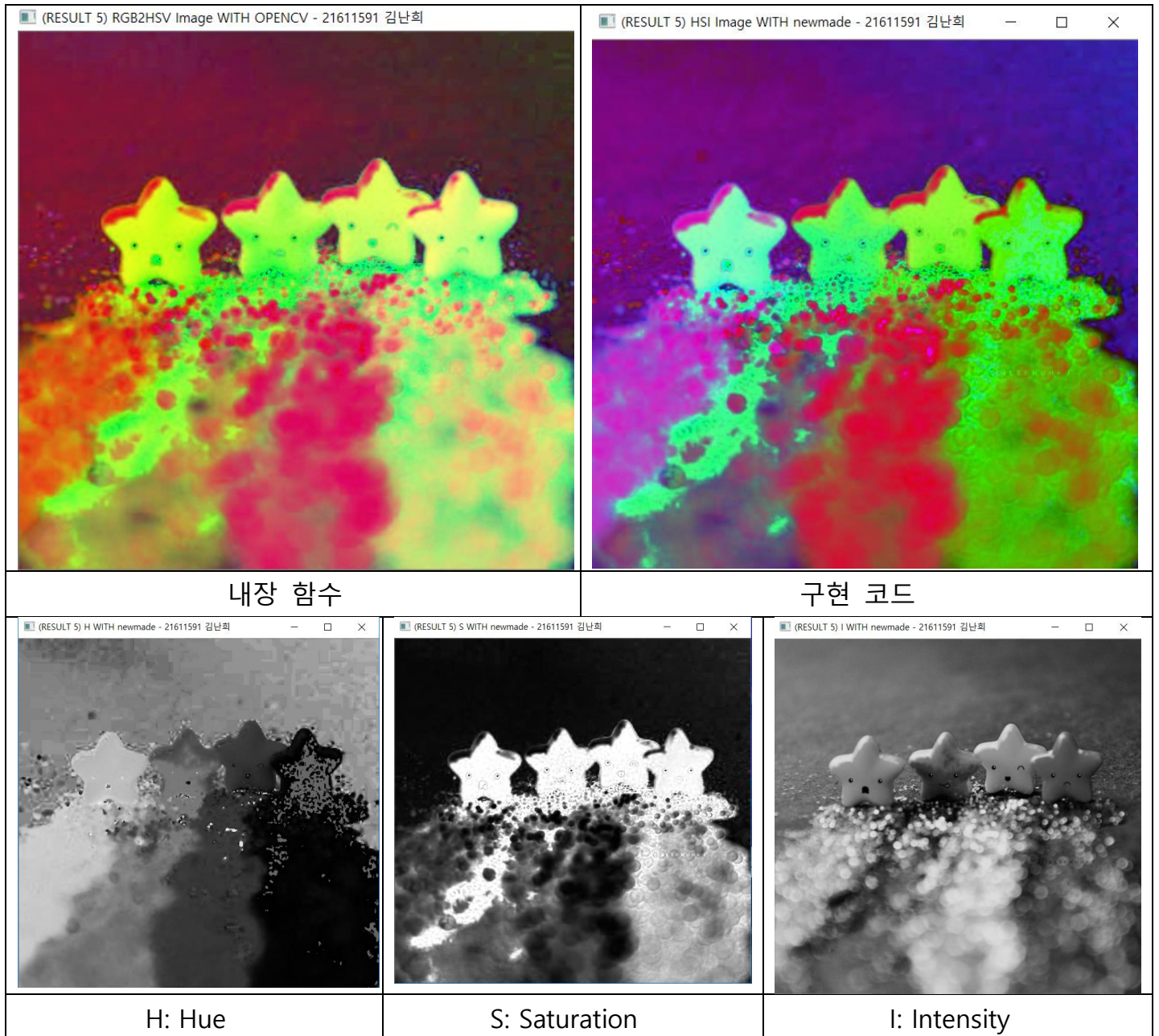


- 결과 소스 코드

```
49 /*****2. RGB to gray 변환: opencv 내장 함수*****/
50 Mat gray;
51 cvtColor(image, gray, CV_RGB2GRAY);
52 imshow("(RESULT 3) RGB2GRAY Image WITH OPENCV - 21611591 김난희", gray);
53
54 /*****2. RGB to gray 변환: 구현 코드*****/
55
56 //원본 이미지의 크기만큼 1채널 unsigned char type으로 0을 만들어줌
57 Mat NEW_gray = Mat(image.rows, image.cols, CV_8UC1);
58
59 for (int i = 0; i < image.rows; i++)
60 {
61     for (int j = 0; j < image.cols; j++)
62     {
63         //input image로부터 현재 위치 (i,j) 픽셀의 blue, green, red 값을 읽어옴
64         uchar blue = image.at<Vec3b>(i, j)[0];
65         uchar green = image.at<Vec3b>(i, j)[1];
66         uchar red = image.at<Vec3b>(i, j)[2];
67
68         //blue, green, red를 더한 후, 3으로 나누면 gray scale이 됨
69         uchar gray = (blue + green + red) / 3.0;
70
71         //Mat타입 변수 NEW_gray에 저장함
72         NEW_gray.at<uchar>(i, j) = gray;
73     }
74 }
75
76 imshow("(RESULT 4) RGB2GRAY Image WITH new made - 21611591 김난희", NEW_gray);
```


3. RGB to HSI 변환: opencv 내장 함수, 구현 코드 비교

- 내장 함수와 구현 코드 결과 사진



- 결과 소스 코드

```

78 /*****3. RGB to HSI 변환: opencv 내장 함수*****/
79 Mat HSI;
80 cvtColor(image, HSI, CV_RGB2HSV);
81 imshow("(RESULT 5) RGB2HSV Image WITH OPENCV - 21611591 김난희", HSI);

```

$$I = \frac{1}{3}(R + G + B)$$

$$S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)]$$

$$\text{if } B \leq G$$

$$H = \cos^{-1} \left[\frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right]$$

$$\text{else, } H = 360 - H$$

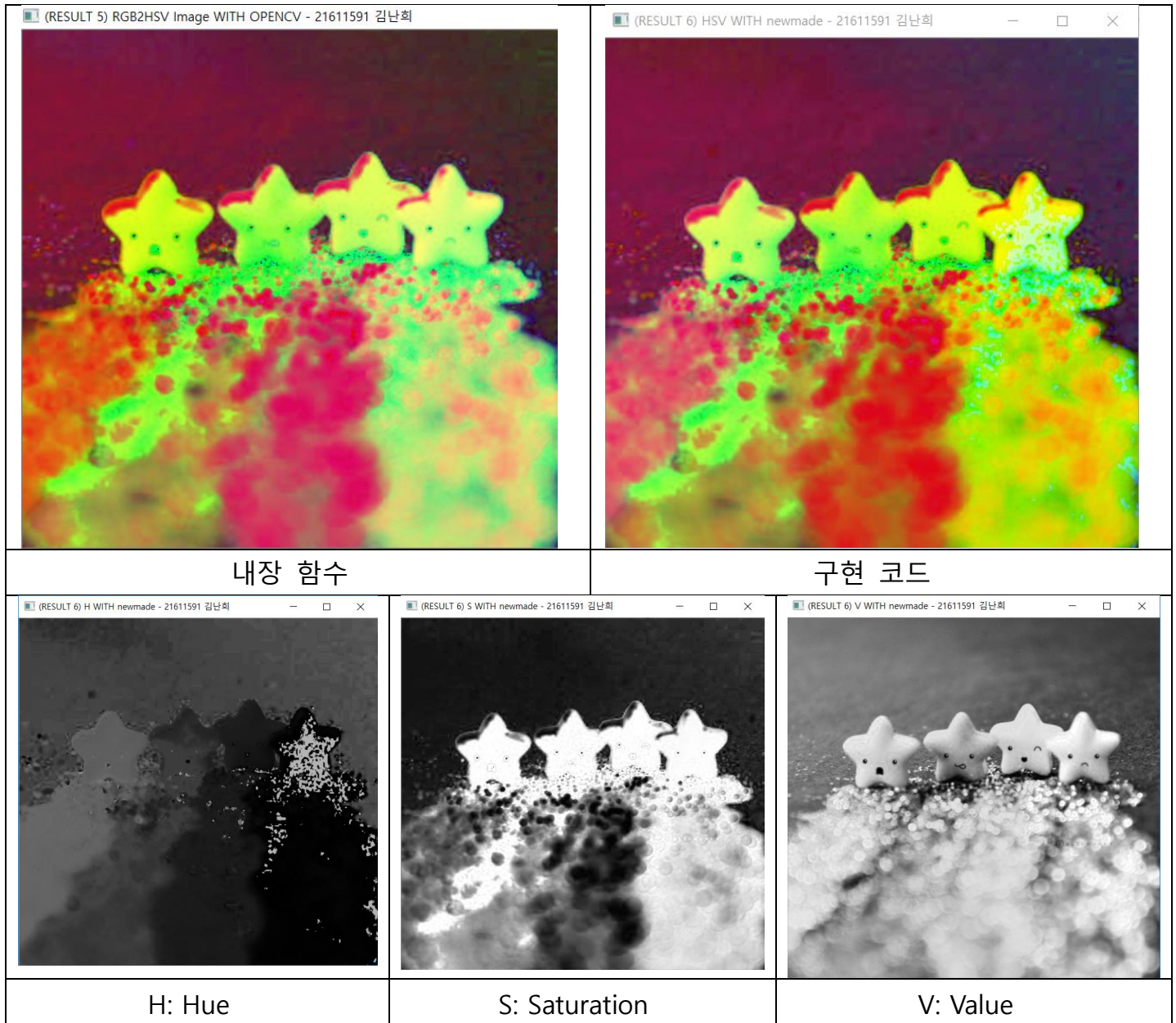
```

83 /*****3. RGB to HSI 변환: 구현 코드*****/
84 Mat New_hsi(image.rows, image.cols, image.type());
85
86 float r, g, b, h, s, in; //HSI와 RGB float형
87
88 for (int i = 0; i < image.rows; i++)
89 {
90     for (int j = 0; j < image.cols; j++)
91     {
92         //input image로부터 현재 위치 (i,j) 픽셀의 blue, green, red 값을 읽어옴
93         b = image.at<Vec3b>(i, j)[0];
94         g = image.at<Vec3b>(i, j)[1];
95         r = image.at<Vec3b>(i, j)[2];
96
97         // blue, green, red를 더한 후, 3으로 나누면 gray scale이 됨, I: intensity
98         in = (b + g + r) / 3;
99
100        // RGB의 최소값을 구해줌
101        int min_val = 0;
102        min_val = std::min(r, std::min(b, g));
103
104        //S: saturation
105        s = 1 - 3 * (min_val / (b + g + r));
106
107        if (s < 0.00001)
108        {
109            s = 0;
110        }
111        else if (s > 0.99999) {
112            s = 1;
113        }
114
115        if (s != 0)
116        {
117            //H: hue
118            h = 0.5 * ((r - g) + (r - b)) / sqrt(((r - g)*(r - g)) + ((r - b)*(g - b)));
119            h = acos(h);
120
121            if (b <= g)
122            {
123                h = h;
124            }
125            else {
126                h = ((360 * 3.14159265) / 180.0) - h; //degree<->PI 변환
127            }
128        }
129
130        New_hsi.at<Vec3b>(i, j)[0] = (h * 180) / 3.14159265; //degree<->PI 변환
131        New_hsi.at<Vec3b>(i, j)[1] = s * 255;
132        New_hsi.at<Vec3b>(i, j)[2] = in;
133    }
134 }
135 split(New_hsi, channels);
136 imshow("(RESULT 5) H WITH newmade - 21611591 김난희", channels[0]);
137 imshow("(RESULT 5) S WITH newmade - 21611591 김난희", channels[1]);
138 imshow("(RESULT 5) I WITH newmade - 21611591 김난희", channels[2]);
139 imshow("(RESULT 5) HSI Image WITH newmade - 21611591 김난희", New_hsi);

```

3. RGB to HSV 변환: 보충 구현 코드

- OpenCV는 HSV를 지원하기 때문에 구현한 HSI와는 차이가 수식에서 발생하여 결과로도 나타난다. HSV를 직접 구현해보면 어떻게 될까?



RGB ↔ HSV

In case of 8-bit and 16-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range.

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases}$$

If $H < 0$ then $H \leftarrow H + 360$. On output $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$.

The values are then converted to the destination data type:

- 8-bit images: $V \leftarrow 255V, S \leftarrow 255S, H \leftarrow H/2$ (to fit to 0 to 255)
- 16-bit images: (currently not supported) $V \leftarrow 65535V, S \leftarrow 65535S, H \leftarrow H$
- 32-bit images: H, S, and V are left as is

- 결과 소스 코드

```
141 /*****3-보충. RGB to HSV 변환: 구현 코드*****/
142 Mat NEW_hsv( image.rows, image.cols, image.type());
143
144 float h, s, v, b, g, r; //HSV와 RGB float형
145
146 for (int i = 0; i < image.rows; i++)
147 {
148     for (int j = 0; j < image.cols; j++)
149     {
150         // 원본 이미지의 픽셀 성분을 받아옴
151         b = image.at<Vec3b>(i, j)[0]; //Blue
152         g = image.at<Vec3b>(i, j)[1]; //Green
153         r = image.at<Vec3b>(i, j)[2]; //Red
154
155         //V: value
156         v = max(r, max(b, g));
157
158         int min_val = 0;
159         min_val = std::min(min(r, g), b);
160
161         //S: saturation
162         if (v != 0)
163         {
164             s = (v - min_val) / v;
165
166         }
167         else {
168             s = 0;
169         }
170
171         //H: Hue
172         if (v == r)
173         {
174             h = 60 * (g - b) / (v - min_val);
175         }
176         else if (v == g)
177         {
178             h = 120 + 60 * (b - r) / (v - min_val);
179         }
180         else if (v == b) {
181
182             h = 240 + 60 * (r - g) / (v - min_val);
183         }
184         if (h < 0) h = h + 360;
```

```

186 //cout << "Wnv : " << v << "Wns: " << s << "Wnh:" << h; //값의 범위가 잘 나오는지 확인
187
188 NEW_hsv.at<Vec3b>(i, j)[0] = h / 2;
189 NEW_hsv.at<Vec3b>(i, j)[1] = s * 255;
190 NEW_hsv.at<Vec3b>(i, j)[2] = v;
191
192 }
193 }
194 split(NEW_hsv, channels);
195 imshow("(RESULT 6) H WITH newmade - 21611591 김난희", channels[0]);
196 imshow("(RESULT 6) S WITH newmade - 21611591 김난희", channels[1]);
197 imshow("(RESULT 6) V WITH newmade - 21611591 김난희", channels[2]);
198 imshow("(RESULT 6) HSV WITH newmade - 21611591 김난희", NEW_hsv);
199
200 waitKey(); //Wait for keystroke
201
202 return 0;
203 }

```

3. Discussion

과제를 하면서 기본적인 C++ 문법과 OpenCV 함수를 함께 공부하였다. Mat, vector으로 하나의 image로 생성하는 것은 생각만큼이나 쉽지 않았다. 구글이나 다양한 사이트를 찾아보지 않아서는 혼자 힘으로 이해하기에 어려움이 많았다. OpenCV에는 다양한 함수들이 있지만, 어느 목적을 달성하기 위해 하나의 함수를 찾고 사용법을 익히는 것은 여러 과정을 거쳐야 했다. C++을 다루면서도 앞으로 자주 이용할 문법에 대해서 익숙해지는 것 같아 좋았다. 이전에 영상처리 시간에서는 MATLAB에서 많이 영상처리를 다뤘었는데, OpenCV로 하는 것도 재미가 있는 것 같다. MATLAB과 C++, OpenCV로 다룰 때의 차이점이 될까, 생각해볼 수 있었다. 물론 아직도 영상은 어렵고 이해할 것이 많은 것 같다.

친구와 선배와 과제에 대해 많은 이야기를 나눴었다. RGB Band Extraction을 할 때, 각자마다 다양한 생각과 방법으로 해결할 수 있을 것 같다는 것이 너무 신기했다. 어느 사람은 merge를 이용하고, 어느 사람은 흩어진 벡터를 모아 흑백을 RGB로 변환하면 될 것 같다는 방법도 있었다. Mat 벡터 객체를 Mat 하나로 만드는 것에 실패해서 배열을 만들어 결국 merge 함수를 썼지만, 더 공부해보면 좋을 것 같았다. HSI 변환도 수식에 대해 정확하게 이해한 것인지는 모르겠지만, 이러한 색상 공간들을 잘 비교하고 이해한다면 많은 도움이 될 것 같았다. HSI와 HSV에 대해 강의자료 여기 저기 적혀 있어 차이점을 몰랐는데, 풀이를 하는 수식의 차이가 있다고 한다. I는 intensity로, V는 value로 둘 다 밝기(명도)를 의미하는 것은 같다. HSL도 있으니 참고하면 좋을 것 같다. OpenCV에서는 HSV만 내장함수로 있었다. 결과 사진을 비교해보면, HSV가 많이 밝게 나왔다. HSV와 HSI가 수식의 차이가 있어서 결과도 차이가 조금 있을 것이라 생각이 든다. HSV를 직접 구현해보는 것은 사이트를 찾아보아도 참고할 만한 자료가 없어 직접 수식에 근거하여 작성하였다. 다른 사이트에서는 RGB값을 8bit로 사용하기 위해 normalize를 해주는 과정이 있었다. 하지만, V(Value) 값을 구하여 출력해보았을 때 255가 나왔다. 범위에 딱 맞는 것이다. 따로 V를 Normalize를 해주지 않아도 되었다. (I(HSI에서)를 구할 때, R+G+B/3를 기본으로 사용하였으므로, V에만 연관이 있다.)

전체 소스 코드

```
//21611591 김난희
#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp> //image proc lib
#include <opencv2/highgui/highgui.hpp>
//#include <iostream>

using namespace cv;
//using namespace std;

int main()
{
    /*****1.RGB color space 채널별 도시: 흑백형태*****/
    Mat image = imread("star.jpg");// color load

    //3채널 분리에 사용할 vector
    vector<Mat> channels;

    //Partition image into three channel planes
    split(image, channels);

    //imshow("(RESULT 1) Original Image - 21611591 김난희", image);
    //imshow("(RESULT 1) Red WITH GRAY - 21611591 김난희", channels[2]);
    //imshow("(RESULT 1) Green WITH GRAY - 21611591 김난희", channels[1]);
    //imshow("(RESULT 1) Blue WITH GRAY - 21611591 김난희", channels[0]);

    /*****1.RGB color space 채널별 도시: 컬러형태*****/

    //원본 이미지의 크기만큼 1채널 unsigned char type으로 0을 만들어줌
    Mat zero = Mat::zeros(image.rows, image.cols, CV_8UC1);

    //BGR 순서로 zero와 조합하여 3채널을 만들어줄 배열을 만들어줌
    Mat B[] = { channels[0],zero,zero };
    Mat G[] = { zero,channels[1],zero };
    Mat R[] = { zero,zero,channels[2] };

    //RGB 중 한 가지 색상(3채널로 된)을 나타내줄 OUTPUT 선언
    vector<Mat> output(3);

    //위에 선언한 RGB에 대한 각각의 색상을 3채널로 합성하여 output에 넣음
    //merge함수는 배열에 따라 채널 수를 정하여 하나의 컬러 이미지로 합성하여 보여줌
    merge(R, 3, output[2]);
    merge(G, 3, output[1]);
    merge(B, 3, output[0]);

    //imshow("(RESULT 2) Red WITH COLOR - 21611591 김난희", output[2]);
    //imshow("(RESULT 2) Green WITH COLOR - 21611591 김난희", output[1]);
    //imshow("(RESULT 2) Blue WITH COLOR - 21611591 김난희", output[0]);

    /*****2. RGB to gray 변환: opencv 내장 함수*****/
    Mat gray;
    cvtColor(image, gray, CV_RGB2GRAY);
    //imshow("(RESULT 3) RGB2GRAY Image WITH OPENCV - 21611591 김난희", gray);

    /*****2. RGB to gray 변환: 구현 코드*****/

    //원본 이미지의 크기만큼 1채널 unsigned char type으로 0을 만들어줌
    Mat NEW_gray = Mat(image.rows, image.cols, CV_8UC1);
```

```

for (int i = 0; i < image.rows; i++)
{
    for (int j = 0; j < image.cols; j++)
    {
        //input image로부터 현재 위치 (i,j) 픽셀의 blue, green, red 값을 읽어옴
        uchar blue = image.at<Vec3b>(i, j)[0];
        uchar green = image.at<Vec3b>(i, j)[1];
        uchar red = image.at<Vec3b>(i, j)[2];

        //blue, green, red를 더한 후, 3으로 나누면 gray scale이 됨
        uchar gray = (blue + green + red) / 3.0;

        //Mat타입 변수 NEW_gray에 저장함
        NEW_gray.at<uchar>(i, j) = gray;
    }
}

```

```

//imshow("(RESULT 4) RGB2GRAY Image WITH new made - 21611591 김난희", NEW_gray);

```

```

/*****3. RGB to HSI 변환: opencv 내장 함수*****/

```

```

Mat HSI;
cvtColor(image, HSI, CV_RGB2HSV);
//imshow("(RESULT 5) RGB2HSV Image WITH OPENCV - 21611591 김난희", HSI);

```

```

/*****3. RGB to HSI 변환: 구현 코드*****/

```

```

Mat New_hsi(image.rows, image.cols, image.type());

```

```

float r, g, b, h, s, in; //HSI와 RGB float형

```

```

for (int i = 0; i < image.rows; i++)
{
    for (int j = 0; j < image.cols; j++)
    {
        //input image로부터 현재 위치 (i,j) 픽셀의 blue, green, red 값을 읽어옴
        b = image.at<Vec3b>(i, j)[0];
        g = image.at<Vec3b>(i, j)[1];
        r = image.at<Vec3b>(i, j)[2];

```

```

        // blue, green, red를 더한 후, 3으로 나누면 gray scale이 됨, I: intensity
        in = (b + g + r) / 3;

```

```

        // RGB의 최소값을 구해줌

```

```

        int min_val = 0;
        min_val = std::min(r, std::min(b, g));

```

```

        //S: saturation

```

```

        s = 1 - 3 * (min_val / (b + g + r));

```

```

        if (s < 0.00001) //이 부분은 블로그를 보고 하였지만, 왜하는 부분인지 이해가 되지

```

않음,

```

        {
            //S가 0과 1사이 값인 것은 확실함 ->

```

0<=S<=255 8BIT로 바꿔줘야함

```

            s = 0;

```

```

        }

```

```

        else if (s > 0.99999) {

```

```

            s = 1;

```

```

        }

```

```

if (s != 0)
{
    //H: hue
    h = 0.5 * ((r - g) + (r - b)) / sqrt(((r - g)*(r - g)) + ((r - b)*(g - b)));
    h = acos(h);

    if (b <= g)
    {
        h = h;
    }
    else {
        h = ((360 * 3.14159265) / 180.0) - h; //degree<->PI 변환
    }
}

```

```

New_hsi.at<Vec3b>(i, j)[0] = (h * 180) / 3.14159265; //degree<->PI 변환
New_hsi.at<Vec3b>(i, j)[1] = s * 255; //블로그는 100을 곱했지만, 이유를 몰라 가장

```

답인 것 같은 255로 바꿈

```

New_hsi.at<Vec3b>(i, j)[2] = in;

```

```

}
}
split(New_hsi, channels);
//imshow("(RESULT 5) H WITH newmade - 21611591 김난희", channels[0]);
//imshow("(RESULT 5) S WITH newmade - 21611591 김난희", channels[1]);
//imshow("(RESULT 5) I WITH newmade - 21611591 김난희", channels[2]);
//imshow("(RESULT 5) HSI Image WITH newmade - 21611591 김난희", New_hsi);

```

```

/*****3-보충. RGB to HSV 변환: 구현 코드*****/
Mat NEW_hsv(image.rows, image.cols, image.type());

```

```

float h, s, v, b, g, r; //HSV와 RGB float형

```

```

for (int i = 0; i < image.rows; i++)
{
    for (int j = 0; j < image.cols; j++)
    {
        // 원본 이미지의 픽셀 성분을 받아옴
        b = image.at<Vec3b>(i, j)[0]; //Blue
        g = image.at<Vec3b>(i, j)[1]; //Green
        r = image.at<Vec3b>(i, j)[2]; //Red

        //V: value
        v = max(r, max(b, g));

        int min_val = 0;
        min_val = std::min(min(r, g), b);

        //S: saturation
        if (v != 0)
        {
            s = (v - min_val) / v;
        }
        else {
            s = 0;
        }

        //H: Hue
        if (v == r)
        {

```


확인

```
        h = 60 * (g - b) / (v - min_val);
    }
    else if (v == g)
    {
        h = 120 + 60 * (b - r) / (v - min_val);
    }
    else if (v == b) {
        h = 240 + 60 * (r - g) / (v - min_val);
    }
    if (h < 0) h = h + 360;

    //cout << "Wnv : " << v << "Wns: " << s << "Wnh:" << h; //값의 범위가 잘 나오는지

    NEW_hsv.at<Vec3b>(i, j)[0] = h / 2;
    NEW_hsv.at<Vec3b>(i, j)[1] = s * 255;
    NEW_hsv.at<Vec3b>(i, j)[2] = v;

    }
}
split(NEW_hsv, channels);
//imshow("(RESULT 6) H WITH newmade - 21611591 김난희", channels[0]);
//imshow("(RESULT 6) S WITH newmade - 21611591 김난희", channels[1]);
//imshow("(RESULT 6) V WITH newmade - 21611591 김난희", channels[2]);
//imshow("(RESULT 6) HSV WITH newmade - 21611591 김난희", NEW_hsv);

waitKey(); //Wait for keystroke

return 0;
}
```

참고 사이트

Mat 생성 방법

<https://nextus.tistory.com/14>

RGB color space(merge)

<http://blog.comart.io/posts/opencv-rgb-1>

RGB to GRAY

<https://webnautes.tistory.com/1042>

<https://cinema4dr12.tistory.com/942>

<https://cinema4dr12.tistory.com/748>

RGB to HSI

<http://answers.opencv.org/question/62446/conversion-from-rgb-to-hsi/>

HSV, HSV, HSL

<https://stackoverflow.com/questions/20853527/hsi-and-hsv-color-space>

<http://egloos.zum.com/kjs1981/v/500548>

RGB to HSV

https://docs.opencv.org/3.3.0/de/d25/imgproc_color_conversions.html

<https://babytiger.tistory.com/entry/opencv%EC%97%90%EC%84%9C-HSV%EC%9D%98-%EA%B0%81-%EC%B1%84%EB%84%90-%EB%B2%94%EC%9C%84>

<https://www.rapidtables.com/convert/color/rgb-to-hsv.html>