

연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV #2
Color Model Conversion, Color Filter

Amylose

DATE : 2019-07-31 WED

● Today's OpenCV Functions

- `void cv::cvtColor(InputArray src, OutputArray dst, int code, int dstCn=0);`
 - Converts an image from one color space to another.
 - Parameters
 - `src` : input image : `uint_8(CV_8U)`, `uint_16(CV_16U)`, `float(CV_32F)`.
 - `dst` : output image of the same size and depth as `src`.
 - `code` : color space conversion code (ref: <https://docs.opencv.org/>)
 - `dstCn` : number of channels in the destination image; if the parameter is 0, the number of the channels is derived automatically from `src` and `code`.
 - Example (pseudo code)

```
1 cv::Mat imgMat_original, imgMat_grayScale, imgMat_rgbModel, imgMat_hsvModel;
2
3 imgMat_original = cv::imread("sample.jpg", IMREAD_COLOR);
4
5 cv::cvtColor(imgMat_original, imgMat_grayScale, cv::COLOR_BGR2GRAY);
6 // convert to "grayscale" model
7 cv::cvtColor(imgMat_original, imgMat_rgbModel, cv::COLOR_BGR2RGB);
8 // convert to "RGB" model
9 cv::cvtColor(imgMat_original, imgMat_hsvModel, cv::COLOR_BGR2HSV);
10 // convert to "HSV" model
```

● Today's OpenCV Functions

➤ `void cv::inRange(InputArray src, InputArray lowerb, InputArray upperb, OutputArray dst);`

- Checks if array elements lie between the elements of two another arrays.
- The function checks the range as follows:
 - For every element of a single-channel input array:

$$dst(I) = lowerb(I)_0 \leq src(I)_0 \leq upperb(I)_0$$

- For two-channel arrays:

$$dst(I) = lowerb(I)_0 \leq src(I)_0 \leq upperb(I)_0 \wedge lowerb(I)_1 \leq src(I)_1 \leq upperb(I)_1$$

- And so forth.
- When the lower and/or upper boundary parameters are scalars, the indexes (I) at lowerb and upperb in the above formulas should be omitted.
- Parameters
 - src : first input array.
 - lowerb : inclusive lower boundary array or a scalar.
 - upperb : inclusive upper boundary array or a scalar.
 - dst : output array of the same size as src and CV_8U type.

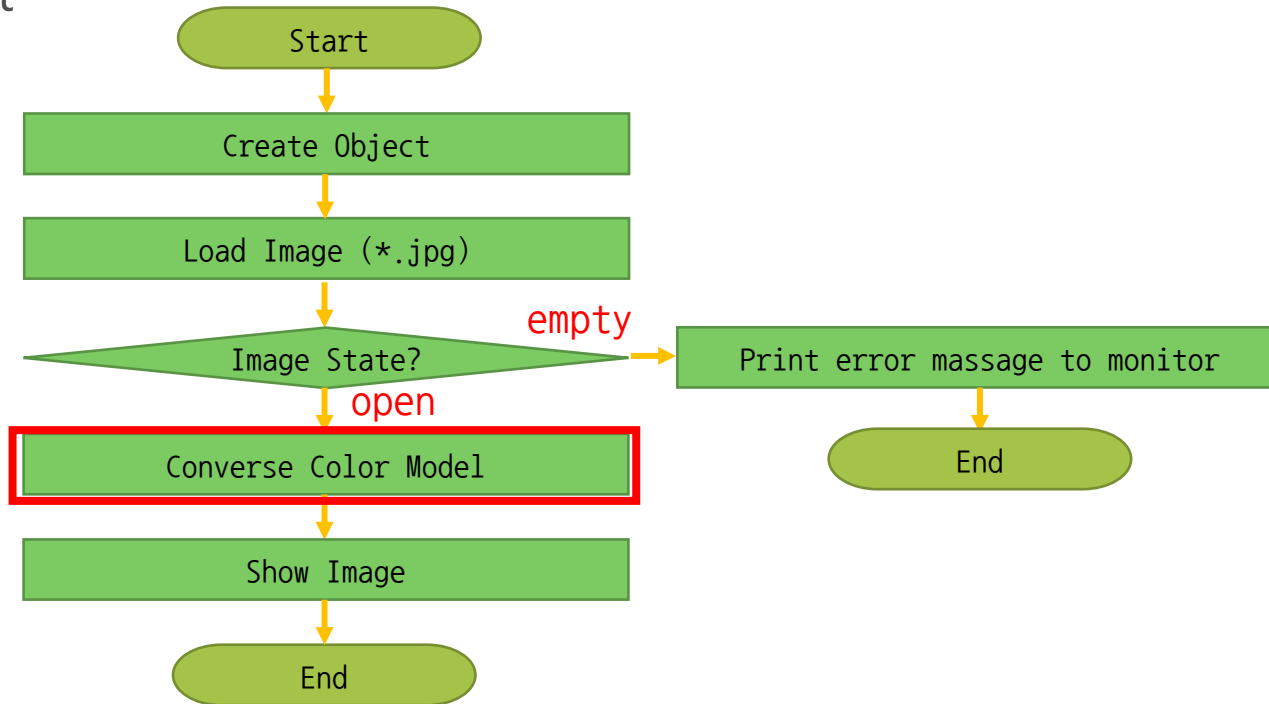
● Today's OpenCV Functions

- `void cv::inRange(InputArray src, InputArray lowerb, InputArray upperb, OutputArray dst);`
 - Example (pseudo code)

```
cv::Mat imgMat_original, imgMat_grayScale, imgMat_mask;  
// create image storage array  
imgMat_original = cv::imread("sample.jpg", IMREAD_COLOR);  
// load sample image from *.jpg file  
cv::cvtColor(imgMat_original, imgMat_grayScale, cv::COLOR_BGR2GRAY);  
// convert to "grayscale" model  
cv::inRange(imgMat_grayScale, cv::Scalar(50), cv::Scalar(120), imgMat_mask);  
// mask color range from 50 to 120
```

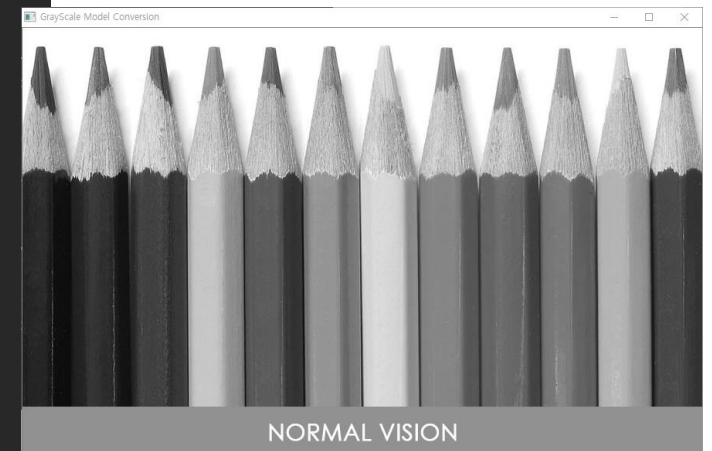
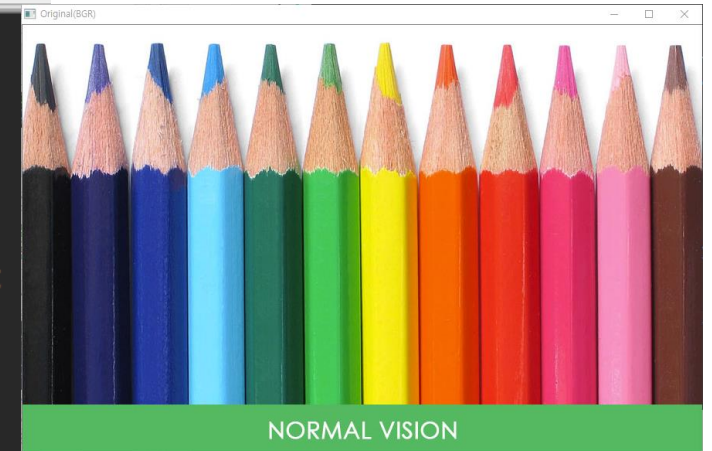
- Examples of use : `cv::cvtColor(InputArray src, OutputArray dst, int code, int dstCn=0);`
 - Programming Environment
 - Used GCC C++, and OpenCV only. (in Windows)
 - Load image file and converse color model.

● Flow Chart



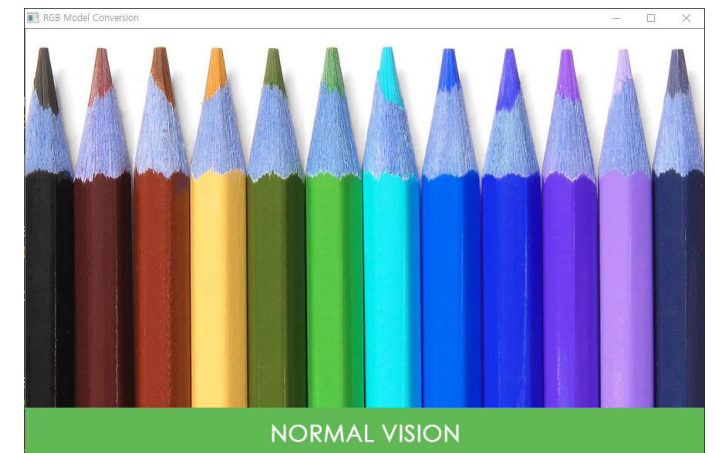
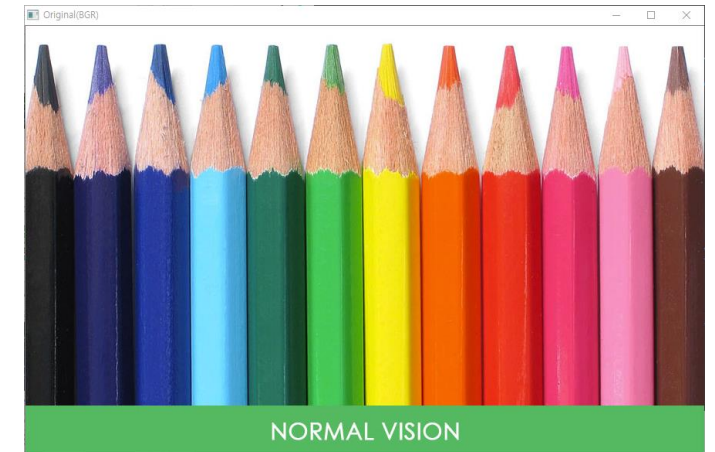
● Result of color model conversion (BGR -> GrayScale)

```
main.cpp x
1  #include <iostream>
2  #include <opencv2/opencv.hpp>
3
4  int main() {
5      cv::Mat imgMat_original, imgMat_model_gray; // Create Object
6      cv::namedWindow("Original(BGR)", cv::WINDOW_AUTOSIZE); // Create Object
7      cv::namedWindow("GrayScale Model Conversion", cv::WINDOW_AUTOSIZE); // Create Object
8
9      imgMat_original = cv::imread("example.jpg", cv::IMREAD_COLOR); // Image Load
10     if ( imgMat_original.empty() ) { // Image State?
11         // Print error message to monitor
12         std::cout << "[Error] cannot open image file" << std::endl;
13         return -1; // End
14     }
15
16     cv::cvtColor(imgMat_original, imgMat_model_gray, CV_BGR2GRAY);
17     // Converse color model : BGR->GRAY
18
19     cv::imshow("Original(BGR)", imgMat_original); // Image Show
20     cv::imshow("GrayScale Model Conversion", imgMat_model_gray); // Image Show
21
22     cv::waitKey(0); //Wait
23 }
24
```



● Result of color model conversion (BGR -> RGB)

```
main.cpp x
1  #include <iostream>
2  #include <opencv2/opencv.hpp>
3
4  int main() {
5      cv::Mat imgMat_original, imgMat_model_rgb; // Create Object
6      cv::namedWindow("Original(BGR)", cv::WINDOW_AUTOSIZE); // Create Object
7      cv::namedWindow("RGB Model Conversion", cv::WINDOW_AUTOSIZE); // Create Object
8
9      imgMat_original = cv::imread("example.jpg", cv::IMREAD_COLOR); // Image Load
10     if ( imgMat_original.empty() ) { // Image State?
11         // Print error message to monitor
12         std::cout << "[Error] cannot open image file" << std::endl;
13         return -1; // End
14     }
15
16     cv::cvtColor(imgMat_original, imgMat_model_rgb, CV_BGR2RGB);
17     // Converse color model : BGR->RGB
18
19     cv::imshow("Original(BGR)", imgMat_original); // Image Show
20     cv::imshow("RGB Model Conversion", imgMat_model_rgb); // Image Show
21
22     cv::waitKey(0); //Wait
23 }
24
```

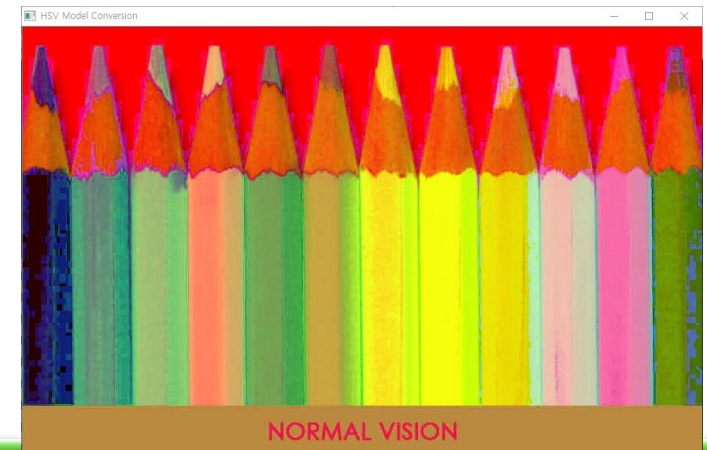
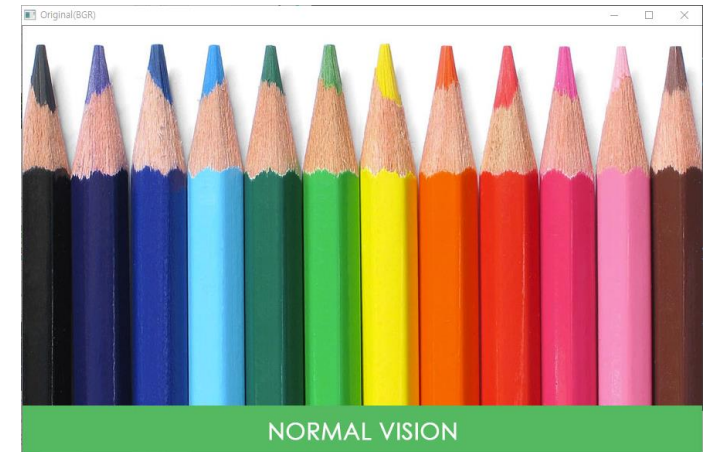


연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV #2 : Color Model Conversion & Color Filter

● Result of color model conversion (BGR -> HSV)

```
main.cpp x
1  #include <iostream>
2  #include <opencv2/opencv.hpp>
3
4  int main() {
5      cv::Mat imgMat_original, imgMat_model_hsv; // Create Object
6      cv::namedWindow("Original(BGR)", cv::WINDOW_AUTOSIZE); // Create Object
7      cv::namedWindow("HSV Model Conversion", cv::WINDOW_AUTOSIZE); // Create Object
8
9      imgMat_original = cv::imread("example.jpg", cv::IMREAD_COLOR); // Image Load
10     if ( imgMat_original.empty() ) { // Image State?
11         // Print error message to monitor
12         std::cout << "[Error] cannot open image file" << std::endl;
13         return -1; // End
14     }
15
16     cv::cvtColor(imgMat_original, imgMat_model_hsv, CV_BGR2HSV);
17     // Converse color model : BGR->HSV
18
19     cv::imshow("Original(BGR)", imgMat_original); // Image Show
20     cv::imshow("HSV Model Conversion", imgMat_model_hsv); // Image Show
21
22     cv::waitKey(0); //Wait
23 }
24
```



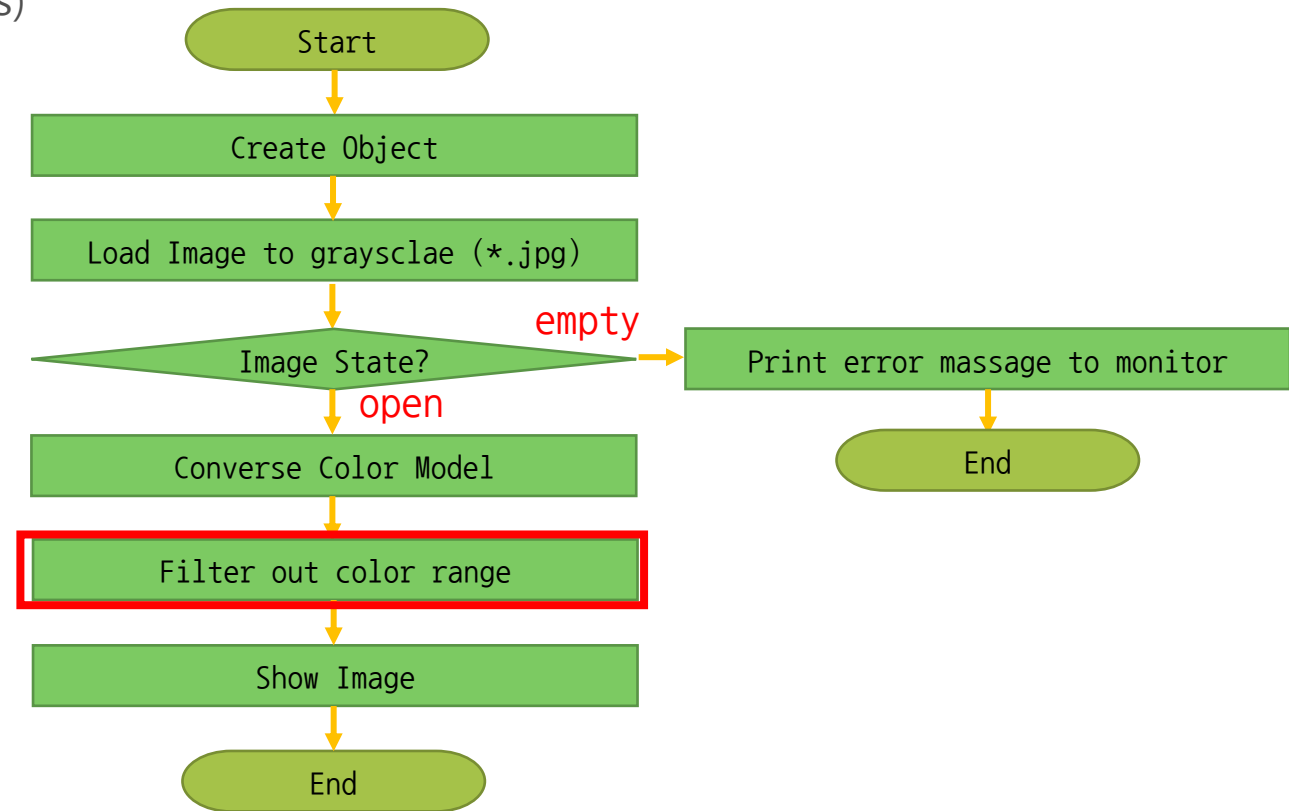
- Examples of use :

`cv::inRange(InputArray src, InputArray lowerb, InputArray upperb, OutputArray dst);`

- Programming Environment

- Used GCC C++, and OpenCV only. (in Windows)
- Load image file and converse color model.

- Flow Chart

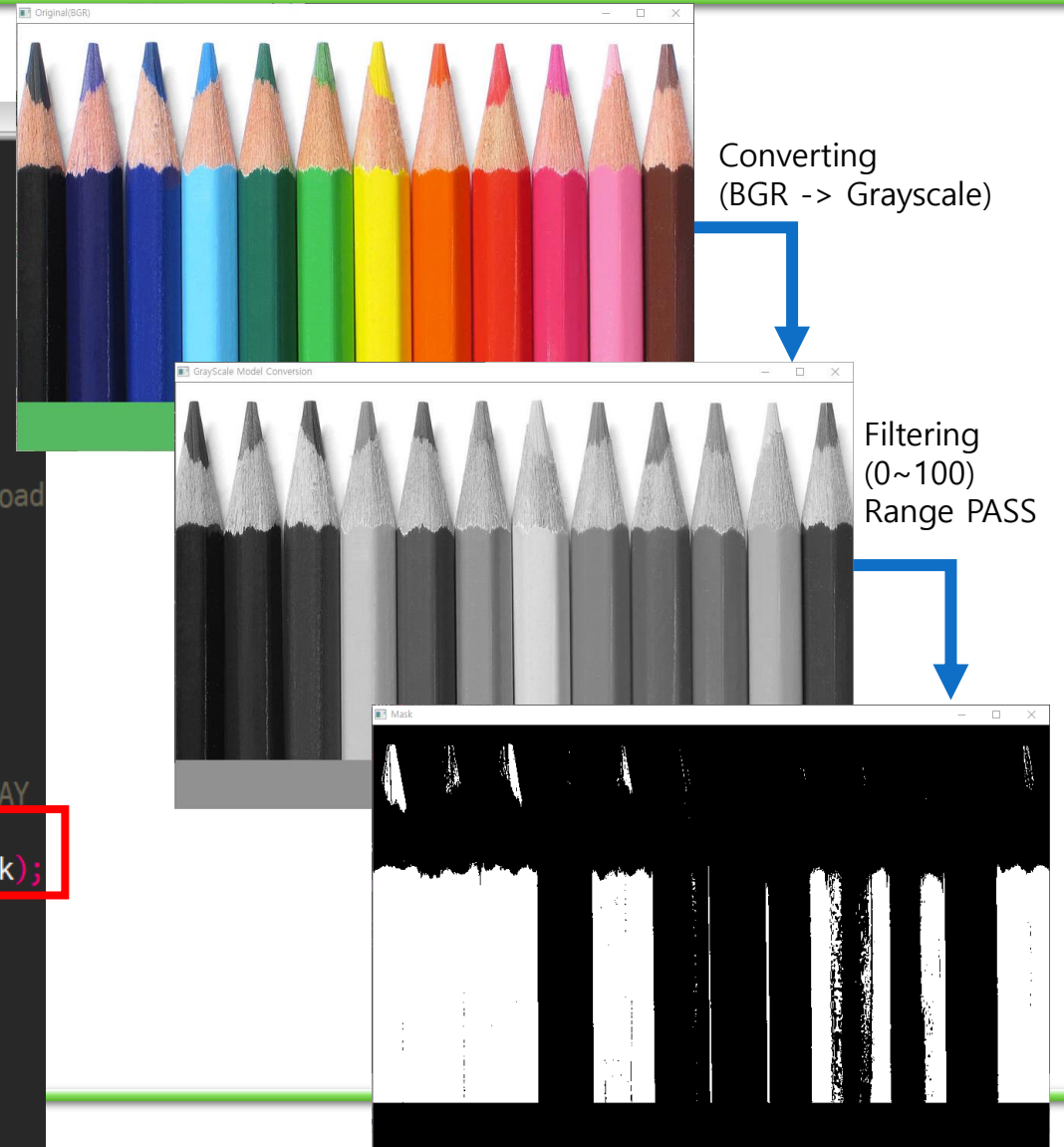


연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV #2 : Color Model Conversion & Color Filter

● Result of color filter example (0~100 Range in Gray)

```
main.cpp x
1  #include <iostream>
2  #include <opencv2/opencv.hpp>
3
4  int main() {
5      cv::Mat imgMat_original, imgMat_model_gray, imgMat_mask;
6      cv::namedWindow("Original(BGR)", cv::WINDOW_AUTOSIZE); // Create Object
7      cv::namedWindow("GrayScale Model", cv::WINDOW_AUTOSIZE); // Create Object
8      cv::namedWindow("Mask", cv::WINDOW_AUTOSIZE);
9
10     imgMat_original = cv::imread("example.jpg", cv::IMREAD_COLOR); // Image Load
11     if ( imgMat_original.empty() ) { // Image State?
12         // Print error message to monitor
13         std::cout << "[Error] cannot open image file" << std::endl;
14         return -1; // End
15     }
16
17     // Converse color model
18     cv::cvtColor(imgMat_original, imgMat_model_gray, CV_BGR2GRAY); // BGR->GRAY
19     // Filter color range in grayscale
20     cv::inRange(imgMat_model_gray, cv::Scalar(0), cv::Scalar(100), imgMat_mask);
21
22     cv::imshow("Original(BGR)", imgMat_original); // Image Show
23     cv::imshow("GrayScale Model", imgMat_model_gray); // Image Show
24     cv::imshow("Mask", imgMat_mask);
25
26     cv::waitKey(0); //Wait
27 }
28
```

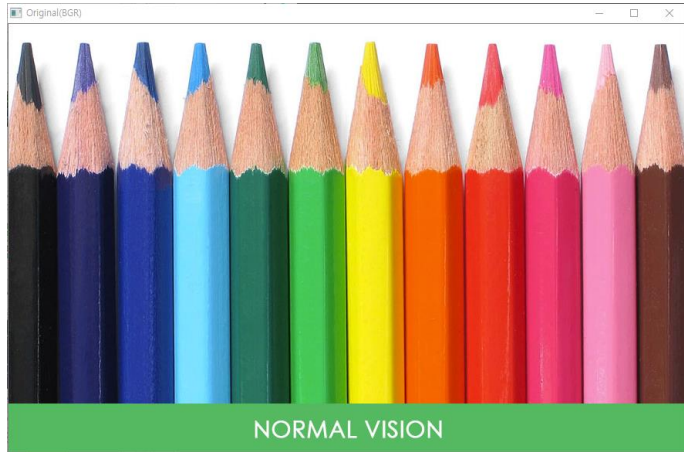


- Result of color filter example

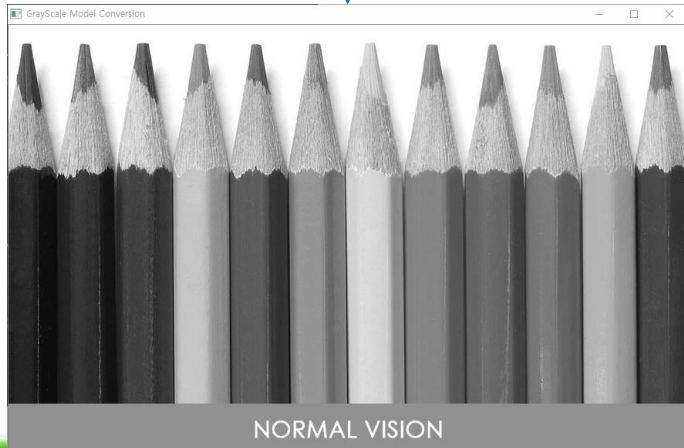
- Lower range : (0~32)
- Upper range : (223 ~255)

```
main.cpp x
1  #include <iostream>
2  #include <opencv2/opencv.hpp>
3
4  int main() {
5      cv::Mat imgMat_original, imgMat_model_gray, imgMat_mask_lower, imgMat_mask_upper;
6      cv::Mat imgMat_maskResult;
7      cv::namedWindow("Original(BGR)", cv::WINDOW_AUTOSIZE); // Create Object
8      cv::namedWindow("GrayScale Model", cv::WINDOW_AUTOSIZE); // Create Object
9      cv::namedWindow("Mask Lower", cv::WINDOW_AUTOSIZE);
10     cv::namedWindow("Mask Upper", cv::WINDOW_AUTOSIZE);
11     cv::namedWindow("Mask Result", cv::WINDOW_AUTOSIZE);
12
13     imgMat_original = cv::imread("example.jpg", cv::IMREAD_COLOR); // Image Load
14     if ( imgMat_original.empty() ) { // Image State?
15         // Print error message to monitor
16         std::cout << "[Error] cannot open image file" << std::endl;
17         return -1; // End
18     }
19
20     // Converse color model
21     cv::cvtColor(imgMat_original, imgMat_model_gray, CV_BGR2GRAY); // BGR->GRAY
22     // Filter color range in grayscale
23     cv::inRange(imgMat_model_gray, cv::Scalar(0), cv::Scalar(32), imgMat_mask_lower);
24     cv::inRange(imgMat_model_gray, cv::Scalar(223), cv::Scalar(255), imgMat_mask_upper);
25
26     imgMat_maskResult = imgMat_mask_lower | imgMat_mask_upper;
27
28     cv::imshow("Original(BGR)", imgMat_original); // Image Show
29     cv::imshow("GrayScale Model", imgMat_model_gray); // Image Show
30     cv::imshow("Mask Lower", imgMat_mask_lower);
31     cv::imshow("Mask Upper", imgMat_mask_upper);
32     cv::imshow("Mask Result", imgMat_maskResult);
33
34     cv::waitKey(0); //Wait
35 }
```

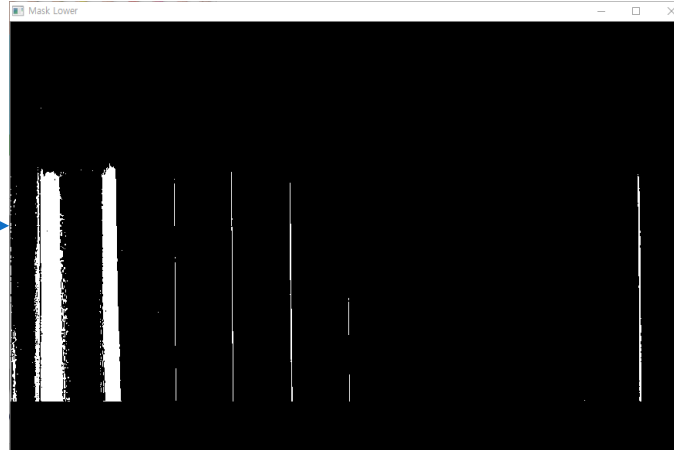
● Result of color filter example



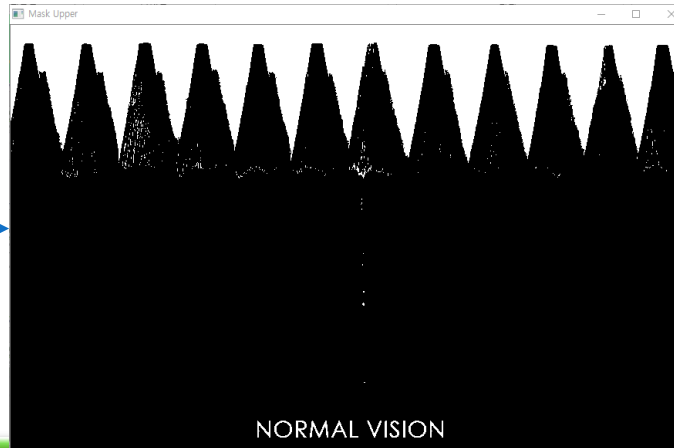
Converting
(BGR -> Grayscale)



Filtering
(0~32) Range Pass



Filtering
(223~255) Range Pass



Bit Operation : "OR"

