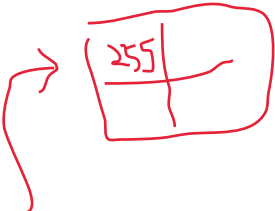


## OpenCV 4차시

### ○ Color plane ON/OFF

BGR 컬러모델의 이미지에 대해, 컬러 요소를 ON (=255) 하거나 OFF(=0) 할 때 이미지의 컬러에 어떠한 변화가 생기는지 확인해 보자. 아래 예제에서는 'R(ed)' 컬러 요소를 255로 강화함으로써, 이미지 전체가 붉게 보인다.

```
7 int main(int argc, char** argv)
8 {
9
10 if (argc != 2)
11 {
12     cout << " Provide image name to read" << endl;
13     return -1;
14 }
15
16 Mat inputImg;
17
18 inputImg = imread(argv[1], IMREAD_UNCHANGED);
19
20 if (inputImg.empty())
21 {
22     cout << "Unable to open file " << endl;
23     return 0;
24 }
25
26 vector<Mat> bgr_images(3);
27 split(inputImg, bgr_images); // bgr
28                                     [2] = R
29 bgr_images[2] = Mat::ones(bgr_images[2].size(), bgr_images[2].type())*255.0;
30                                     // 가 1 . ( , type)
31
32 Mat onOffImg;
33 merge(bgr_images, onOffImg); // bgr -> R 255가 .
34
35 imshow("Original", inputImg);
36 imshow("onOffImg", onOffImg);
37 waitKey(0);
38 return 0;
39 }
```



### ○ 실험

Mat::zeros( ) 함수로 컬러 요소를 off할 때 어떠한 변화가 생기는지 확인해 보자.

#### o Color 바꾸기

BGR 컬러모델의 이미지에 대해, Blue 요소의 값과 Green 요소의 값을 바꾼 이미지를 만들어 보자. 이를 위해서 함수 `split()`을 이용하여 각 color 요소들을 독립된 Mat로 추출하고, Mat를 switch한다.

```
7  int main(int argc, char** argv)
8  {
9
10     if (argc != 2)
11     {
12         cout << " Provide image name to read" << endl;
13         return -1;
14     }
15
16     Mat inputImg;
17
18     inputImg = imread(argv[1], IMREAD_UNCHANGED);
19
20     if (inputImg.empty())
21     {
22         cout << "Unable to open file " << endl;
23         return 0;
24     }
25
26     vector<Mat> bgr_images(3);
27     split(inputImg, bgr_images);
28
29     #define FROM 0
30     #define TO 1
31
32     Mat temp;
33     temp = bgr_images[FROM];
34     bgr_images[FROM] = bgr_images[TO];
35     bgr_images[TO] = temp;
36
37     Mat switchedImg;
38     merge(bgr_images, switchedImg);
39
40     imshow("Original", inputImg);
41     imshow("Switched", switchedImg);
42     waitKey(0);
43     return 0;
44 }
```

#### o 실험

- 다른 조합으로 컬러들을 바꾸어 보자.

```
*           *
circle(img,Point(           ),           , Scalar(           (=0,0,255)),           )
```

o MP4 포맷 비디오 파일 표시

mp4 포맷의 비디오 파일 (cyber 사이트에서 다운로드)을 열어서, frame 단위로 표시

```
7 int main(int argc, char* argv[]) {
8
9     // Create a VideoCapture object and open the input file
10    // If the input is the web camera, pass 0 instead of the video file name
11    VideoCapture cap(argv[1]);
12
13    // Check if camera opened successfully
14    if (!cap.isOpened()) {
15        cout << "Error opening video stream or file" << endl;
16        return -1;
17    }
18
19    while (1) {
20
21        Mat frame; // video
22        // Capture frame-by-frame
23        cap >> frame; capture frame
24
25        // If the frame is empty, break immediately
26        if (frame.empty())
27            break;
28
29        // Display the resulting frame
30        imshow("Frame", frame); // frame
31
32        // Press ESC on keyboard to exit
33        char c = (char)waitKey(1); // 1ms
34        if (c == 27) // ASCII 27 = ESC KEY
35            break;
36    }
37
38    // When everything done, release the video capture object
39    cap.release();
40
41    // Closes all the frames
42    destroyAllWindows();
43
44    return 0;
45 }
```

· 라인 33 함수 waitKey(1)의 '1' 대신 다른 양의 정수를 써보자 어떤 일이 벌어지는가?

· '1' 대신 0을 써보자 어떤 일이 벌어지는가?

· 컬러 frame 대신에 함수 cvtColor( )를 이용하여 gray로 변환된 이미지를 출력

○ 흑백 binary image로 비디오 표시하기

thresholding 방법을 이용하여 비디오 frame들을 binary image로 표시해 보자

```
31 // Display the gray converted frame.  
32 cvtColor(frame, frame, COLOR_BGR2GRAY);  
33 threshold(frame, binImg, 127, 255, THRESH_OTSU);  
34 imshow("Frame", binImg);
```

○ 실험

함수 adaptiveThreshold( )를 이용하여 색다른 느낌의 binary image를 만들어 보자.

예를 들어,

```
adaptiveThreshold(frame, binImg, 255, ADAPTIVE_THRESH_GAUSSIAN_C,  
THRESH_BINARY, 5, 3.0);
```

○ Histogram Equalized된 컬러 이미지로 비디오를 출력해 보자.

힌트.

함수 equlizeHist( )를 사용한다.

○ 절대값 threshold에 의해서, binary화된 이미지에 대해 morphological open operation을 수행해서 출력해 보자.

○ 절대값 threshold에 의해서, binary화된 이미지에 대해 morphological close operation을 수행해서 출력해 보자.

○ Canny edge detection

Canny edge detection에 의해 비디오 frame들을 처리한 결과를 표시해 보자

```
29 // The Canny edge detection requires the input images are gray  
30 cvtColor(frame, frame, COLOR_BGR2GRAY);  
31 Canny(frame, frame, 100, 200);
```

• Canny edge detection 방법의 동작원리에 대해서 학습해 보자.

• 라인 31의 함수 Canny( )의 3번째와 4번째 인수 100과 200의 의미는 무엇인가?

Threshold 200을 넘는 gradient를 갖는 pixel은 edge임.

Threshold 100을 넘는 gradient를 갖는 pixel의 경우, edge pixel과 인접해 있다면 edge, 아니면 non-edge