

## OpenCV 7차시

### ○ Trackbar (트랙바)

슬라이드에 의해서 값을 조절할 수 있는 컨트롤.

아래 그림에서 화살표 표시된 슬라이더를 좌우로 움직일 수 있다.



### ○ 소스코드

- . trackbarname: 문자열로, trackbar 이름을 표시한다.
- . winname: 문자열로 trackbar가 표시될 창 이름을 의미
- . value: 초기값
- . count: 최대값 (최소값은 0)
- . onChange: callback 함수로 trackbar 값이 변경될 때 호출됨

```
int cv::createTrackbar ( const String & trackbarname, //  
                        const String & winname, // ( ? )  
                        int * value, // ( )  
                        int count, // =>  
                        onChange =  
                        TrackbarCallback 0, // call back  
                        void * userdata = 0 //  
                        )
```

```

1  #include <opencv2/opencv.hpp>
2
3  using namespace cv;
4  using namespace std;
5
6  int main(int argc, char* argv[])
7  {
8
9      if (argc < 2)
10     {
11         cout << "No image name" << endl;
12         return -1;
13     }
14
15     Mat image = imread(argv[1], IMREAD_UNCHANGED);
16
17     if (image.empty() == true)
18     {
19         cout << "unable to read image" << endl;
20         return -1;
21     }
22
23     int initialValue = 100;
24
25     imshow("MyImage", image); //
26     createTrackbar("MyTrackbar", "MyImage", &initialValue, 200, 0);
27     waitKey(0);
28     return 0;
29 }

```

○ Trackbar의 callback 함수

- . callback함수의 원형은 void callback\_function (int value, void \*other)
- . value는 현재 trackbar의 값

○ Trackbar callback함수의 구현

- . trackbar의 값이 변경될 때마다, 현재 값을 console에 출력하도록 해보자.
- . Callback함수의 이름은 trackBarChanged로 설정
- . 함수 createTrackbar ( )에서 callback함수의 이름을 지정

```

6 void trackBarChanged(int value, void *)
7 {
8     cout << "trackbar value is " << value << endl;
9 }
10
11 int main(int argc, char* argv[])
12 {
13
14     if (argc < 2)
15     {
16         cout << "No image name" << endl;
17         return -1;
18     }
19
20     Mat image = imread(argv[1], IMREAD_UNCHANGED);
21
22     if (image.empty() == true)
23     {
24         cout << "unable to read image" << endl;
25         return -1;
26     }
27
28     int initialValue = 100;
29
30     imshow("MyImage", image);
31     createTrackbar("MyTrackbar", "MyImage", &initialValue, 200, trackBarChanged);
32     waitKey(0);
33     return 0;
34 }

```

#### ○ Trackbar의 응용

- . Trackbar value에 따라서 이진화를 위한 threshold 결과값이 다르게 해보자.
- . Value가 커지면, 검정 픽셀이 많아진다.
- . 슬라이더를 움직이면서, threshold의 결과를 살펴보자.



```

14  int main(int argc, char* argv[])
15  {
16
17      if (argc < 2)
18      {
19          cout << "No image name" << endl;
20          return -1;
21      }
22
23      Mat image = imread(argv[1], IMREAD_UNCHANGED);
24
25      if (image.empty() == true)
26      {
27          cout << "unable to read image" << endl;
28          return -1;
29      }
30
31      Mat grayImage;
32      cvtColor(image, grayImage, COLOR_BGR2GRAY);
33
34      int initialValue = 100;
35
36      Mat binImage;
37      threshold(grayImage, binImage, initialValue, 255, CV_THRESH_BINARY);
38
39      imshow("MyImage", binImage);
40      createTrackbar("MyTrackbar", "MyImage", &initialValue, 200, trackBarChanged, &grayImage);
41      waitKey(0);
42      return 0;
43  }

```

```

6  void trackBarChanged(int value, void *param)
7  {
8      Mat* _grayImage = (Mat *)param;
9      Mat resImage;
10     threshold(*_grayImage, resImage, value, 255, CV_THRESH_BINARY);
11     imshow("MyImage", resImage);
12 }

```

○ Trackbar를 이용한 영상의 밝기 (brightness) 조절

- . 픽셀값에 일률적으로 같은 양수를 더하면 전체적으로 이미지가 밝아진다.
- . 컬러이미지에 대해서, trackbar를 이용하여 선택한 값을 더해서 이미지의 brightness를 조절해 보자.
- . 이미지의 픽셀들에 값을 곱하거나, 더하는 것은 함수 `Mat::convertTo( )`를 이용한다.

```
void cv::Mat::convertTo ( OutputArray m,  
                        int rtype,  
                        alpha =  
                        double 1,  
                        double beta = 0  
                        ) const
```

- . m은 결과 이미지
- . rtype은 결과 이미지의 type으로 입력이미지 type과 동일하게 하면 된다.
- . alpha는 각 픽셀에 곱할 값
- . beta는 각 픽셀에 더할 값
- . 따라서 각 픽셀값 p는 위 함수에 의해  $\alpha * p + \beta$ 로 갱신된다.

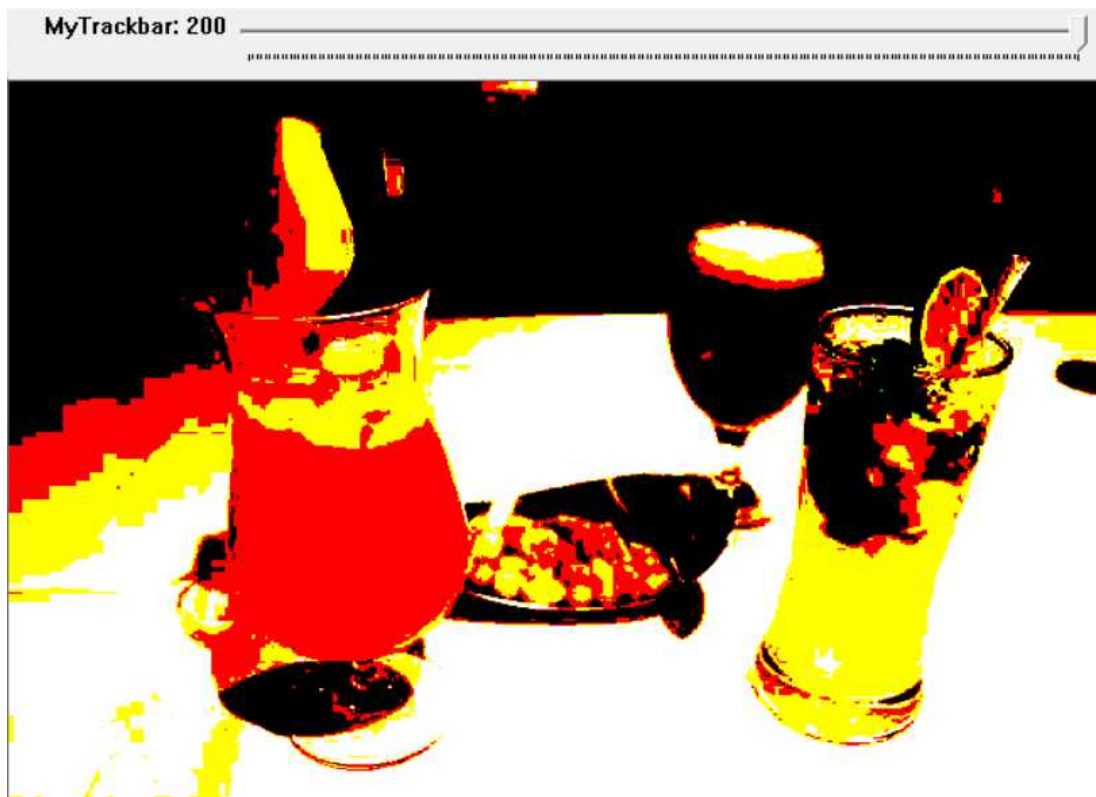
```
7 void trackBarChanged(int value, void *param)  
8 {  
9     Mat* _image = (Mat *)param;  
10    Mat resImage;  
11    _image->convertTo(resImage, _image->type(), 1.0, (double)value);  
12    imshow("MyImage", resImage);  
13 }  
14  
15 int main(int argc, char* argv[])  
16 {  
17  
18     if (argc < 2)  
19     {  
20         cout << "No image name" << endl;  
21         return -1;  
22     }  
23  
24     Mat image = imread(argv[1], IMREAD_UNCHANGED);  
25  
26     if (image.empty() == true)  
27     {  
28         cout << "unable to read image" << endl;  
29         return -1;  
30     }  
31  
32     int initialValue = 50;  
33  
34     Mat resImage;  
35     image.convertTo(resImage, image.type(), 1.0, (double)initialValue);  
36  
37     imshow("MyImage", resImage);  
38     createTrackbar("MyTrackbar", "MyImage", &initialValue, 255, trackBarChanged, &image);  
39     waitKey(0);  
40     return 0;  
41 }
```

- Trackbar를 이용한 대비 (contrast)의 조절
- . brightness는 이미지의 전반적인 밝기를 조절한다.
- . contrast는 이미지의 어두운 곳과 밝은 곳의 대조를 부각시키는 것임.
- . contrast의 원리는 ‘어두운 곳은 더 어둡게, 밝은 곳은 더 밝게’
- . contrast를 조절하는 공식은 다음과 같다.

```
var_brightness = init_brightness - 100;
var_contrast = init_contrast - 100;

if var_contrast > 0
    delta(idx) = 127.0*var_contrast / 100;
    a(idx) = 255.0 / (255.0 - delta(idx) * 2);
    b(idx) = a(idx)*(var_brightness - delta(idx));
else
    delta(idx) = -128.0*var_contrast / 100;
    a(idx) = (255.0 - delta(idx) * 2) / 255.0;
    b(idx) = a(idx)*var_brightness + delta(idx);
end
```

- . init\_contrast와 init\_brightness는 주어진 contrast와 brightness의 값으로, 범위는 0 ~ 200까지이고, 100이면 original image이다.
- . 주어진 공식에 의해 a (alpha)와 b (beta)값이 계산되면, 이를 함수 Mat::convertTo( )에 활용하면 된다.



- . contrast를 최대치인 200으로 했을 때의 결과.



o 소스코드

. brightness = 100으로 가정.

```
7 void trackBarChanged(int value, void *param)
8 {
9     Mat* _image = (Mat *)param;
10    Mat resImage;
11
12    int var_contrast = value - 100;
13    double a, b;
14
15    if (var_contrast > 0)
16    {
17        double delta = 127.0*var_contrast / 100.0;
18        a = 255.0 / (255 - delta * 2);
19        b = a * (- delta);
20    }
21    else
22    {
23        double delta = -128.0*var_contrast / 100.0;
24        a = (256.0 - delta * 2.0) / 255.0;
25        b = delta;
26    }
27
28    _image->convertTo(resImage, _image->type(), a, b);
29    imshow("MyImage", resImage);
30 }
31
32 int main(int argc, char* argv[])
33 {
34
35     if (argc < 2)
36     {
37         cout << "No image name" << endl;
38         return -1;
39     }
40
41     Mat image = imread(argv[1], IMREAD_UNCHANGED);
42
43     if (image.empty() == true)
44     {
45         cout << "unable to read image" << endl;
46         return -1;
47     }
48
49     int initialValue = 100;
50
51     imshow("MyImage", image);
52     createTrackbar("MyTrackbar", "MyImage", &initialValue, 200, trackBarC
53     waitKey(0);
54     return 0;
55 }
```

○ 실습

- . 위에서 실습한 contrast 조절을 클래스로 구현하고자 한다.
- . 아래 코드가 동작하도록 'MyContrastClass'를 구현하시오.

```
7 void trackBarChanged(int value, void *param)
8 {
9     MyContrastClass mcc = new MyContrastClass((Mat *)param);
10    imshow("MyImage", mcc->getContrastedImage(value));
11    delete(mcc);
12 }
13
14 int main(int argc, char* argv[])
15 {
16
17     if (argc < 2)
18     {
19         cout << "No image name" << endl;
20         return -1;
21     }
22
23     Mat image = imread(argv[1], IMREAD_UNCHANGED);
24
25     if (image.empty() == true)
26     {
27         cout << "unable to read image" << endl;
28         return -1;
29     }
30
31     int initialValue = 100;
32
33     imshow("MyImage", image);
34     createTrackbar("MyTrackbar", "MyImage", &initialValue, 200, trackBarChanged, &image);
35     waitKey(0);
36     return 0;
37 }
```