영상처리 실제 3주차 실습(2)

2023254015 장욱진

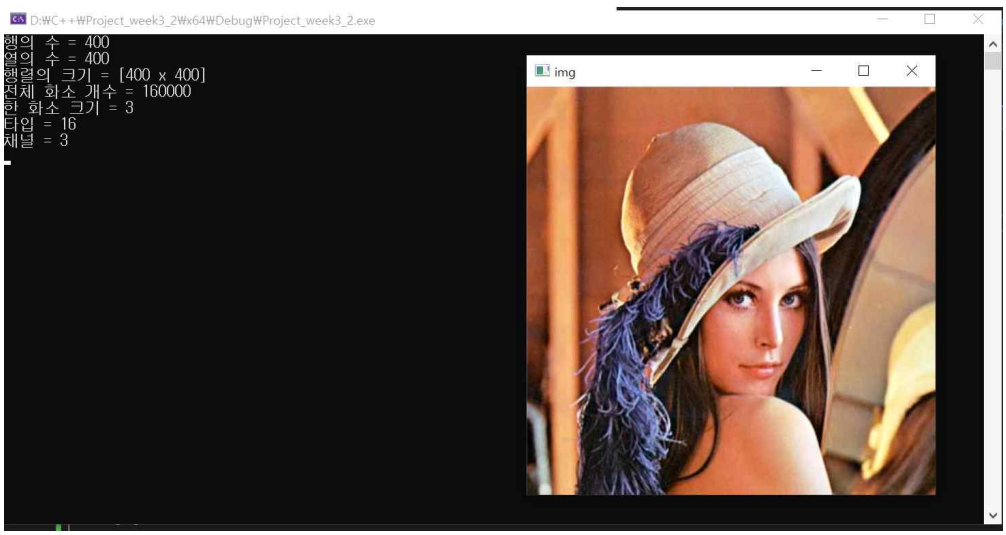
```
#include <opencv2/opencv.hpp>

using namespace cv;
using namespace std;

Mat img, roi;
int mx1, my1, mx2, my2;
bool cropping = false;

void onMouse(int event, int x, int y, int flags, void* param)
{
	if (event == EVENT_LBUTTONDOWN)
	{
		mx1 = x;
		my1 = y;
		cropping = true;
	}
	else if (event == EVENT_LBUTTONUP)
	{
		mx2 = x;
		my2 = y;
		cropping = false;
		rectangle(img, Rect(mx1, my1, mx2 - mx1, my2 - my1),
			Scalar(0, 255, 0), 2);
		imshow("image", img);
	}
}

void page6()
{
	float data[] = { 1.2f, 2.3f, 3.2f, 4.5f, 5.0f, 6.5f };

	Mat m1(2, 3, CV_8U);
	Mat m2(2, 3, CV_8U, Scalar(300));
	Mat m3(2, 3, CV_16S, Scalar(300));
	Mat m4(2, 3, CV_32F, data);

	Size sz(2, 3);
	Mat m5(Size(2, 3), CV_64F);
	Mat m6(sz, CV_32F, data);

	cout << "[m1] = " << endl << m1 << endl;
	cout << "[m2] = " << endl << m2 << endl;
	cout << "[m3] = " << endl << m3 << endl;
	cout << "[m4] = " << endl << m4 << endl << endl;
```

```
	cout << "[m5] = " << endl << m5 << endl;
	cout << "[m6] = " << endl << m6 << endl;

}

int page17()
{
	Mat img = imread("d:/lenna.jpg");

	if (img.empty()) { cout << "영상을 읽을 수 없음" << endl; return -1; }
	imshow("img", img);

	cout << "행의 수 = " << img.rows << endl;
	cout << "열의 수 = " << img.cols << endl;
	cout << "행렬의 크기 = " << img.size() << endl;
	cout << "전체 화소 개수 = " << img.total() << endl;
	cout << "한 화소 크기 = " << img.elemSize() << endl;
	cout << "타입 = " << img.type() << endl;
	cout << "채널 = " << img.channels() << endl;

	waitKey(0);
	return 0;
}

void page21()
{
	Mat m1(2, 3, CV_8U, 2);
	Mat m2(2, 3, CV_8U, Scalar(10));

	Mat m3 = m1 + m2;
	Mat m4 = m2 - 6;
	Mat m5 = m1;

	cout << "[m2] = " << endl << m2 << endl;
	cout << "[m3] = " << endl << m3 << endl;
	cout << "[m4] = " << endl << m4 << endl << endl;

	cout << "[m1]" << endl << m1 << endl;
	cout << "[m5]" << endl << m5 << endl << endl;

	m5 = 100;
	cout << "[m1] = " << endl << m1 << endl;
	cout << "[m5] = " << endl << m5 << endl;
}

void page33()
{
	img = imread("./lenna.jpg");
	imshow("image", img);
```

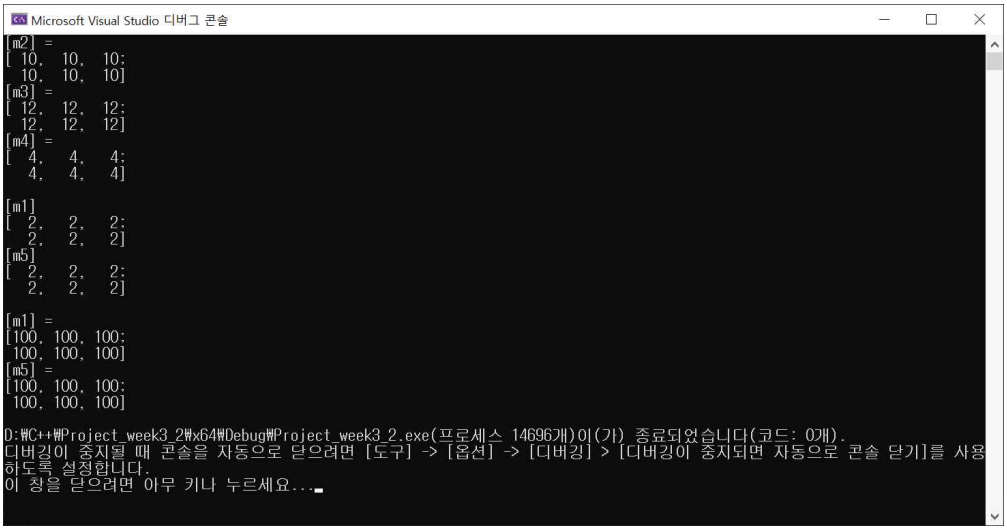
```
	Mat clone = img.clone();

	setMouseCallback("image", onMouse);

	while (1)
	{
		int key = waitKey(100);
		if (key == 'q')break;
		else if (key == 'c')
		{
			roi = clone(Rect(mx1, my1, mx2 - mx1, my2 - my1));
			imwrite("./result.jpg", roi);
		}
	}
}

int main()
{
	page6();
	page17();
	page21();
	page33();

	return 0;
}
```

결과화면

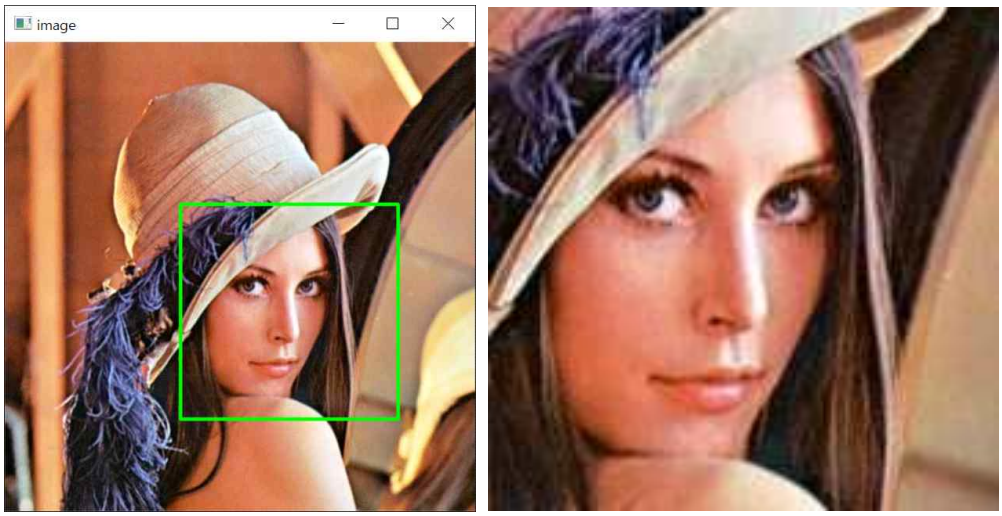
<page6 결과화면>



<page17 결과화면>



<page 21 결과화면>



<page33 결과화면>