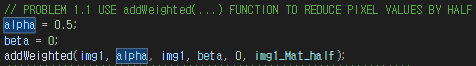
[고소실\_1주차과제]2반\_20161595\_배성현

1**. 모든 컬러 값들을 grayscale로 변환해주는 공식에 대입하였다. 이는 특정 컬러 값이 많이 나타나는 이미지에서 불필요하게 동일한 연산을 반복하는 결과를 불러온다. 이러한 문제를 해결할 수 있는 효율적인 컬러 변환 방법에 대하여 기술한다.**

특정 컬러 값이 많이 나타나는 이미지에서 불필요하게 동일한 연산을 반복하게 되는 문제를 해결하기 위한 하나의 방법으로는 RGB로 나타낼 수 있는 모든 컬러 경우의 수에 대하여 grayscale값을 먼저 계산하여 미리 저장해 놓는 방법이 있다. 이렇게 되면 이미지에서 특정 컬러 값이 많이 나타나더라도 이에 대해 계속 동일한 연산을 반복할 필요가 없이, 해당 컬러 값에 대한 grayscale값을 미리 저장해 놓은 곳에서 가져와 사용하기만 하면 된다. 또 이렇게 구현하게 되면 연산에서 상당히 시간을 많이 차지하게 되는 floating point 연산만을 고려하였을 때, 각 컬러마다 floating point 연산이 3번씩 들어가게 되므로 컬러가 24비트로 표현되는 이미지의 경우 floating point연산이 총 3\*256\*256\*256번만 수행되게 된다.

**2. Open source 라이브러리인 OpenCV에 컬러 변환을 수행하는 함수들이 지원된다. 이러한 함수를 사용하여 위에서 실습한 컬러 변환을 해보고, 어떤 경우에 OpenCV와 같은 툴의 사용이 제한될 수 있는지에 관하여 기술한다. 제공되는 OpenCVcolorConversion 프로젝트를 이용한다.**

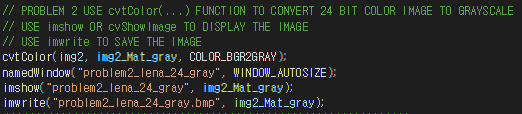
Problem 1.1. alpha에 0.5값을 주고 beta에 0값을 주어, 이를 addWeighted함수에 이용하여 img1(lena\_8.bmp)의 픽셀값을 반으로 줄일 수 있도록 하였다.



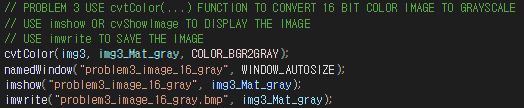
Problem 1.2 . alpha에 2값을 주고 beta에 0값을 주어, 이를 addWeighted함수에 이용하여 img1(lena\_8.bmp)의 픽셀 값을 두 배로 늘릴 수 있도록 하였다.



Problem 2. img2(lena\_24.bmp)를 cvtColor함수를 이용하여 grayscale로 전환하여 img2\_Mat\_gray에 저장될 수 있도록 하였고 또 이를 출력하고 파일로 저장할 수 있도록 하였다.



Problem 3. img3(lena\_16.bmp)를 cvtColor함수를 이용하여 grayscale로 전환하여 img3\_Mat\_gray에 저장될 수 있도록 하였고 또 이를 출력하고 파일로 저장할 수 있도록 하였다.



OpenCV와 같은 툴을 이용하게 되면 위의 cvtColor 함수의 ‘COLOR\_BGR2GRAY’와 같이 color변환에 대한 값들이 이미 지정 되어있다. 따라서 전체 이미지가 아닌 이미지의 특정 부분에만 색깔의 가중치를 준다던가, 이미지 소스의 각각의 R, G, B 값에 대하여 서로 다른 가중치를 주는 것과 같은 색깔 변화가 필요한 경우, 즉 컬러에 있어 사용자의 세세한 control이 필요한 경우에 OpenCV와 같은 툴의 사용이 제한되게 된다.