**고급 소프트웨어 실습 1주차 과제**

20171666 이예은

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.

* openCV의 cvtColor 함수를 이용해 간단하게 구현할 수 있다. 또는, 주변 색은 비슷할 확률이 높다는 것을 이용한다. 한 픽셀에서 주변 특정 크기 영역의 픽셀 값들을 평균 등을 내어 그 값을 grayscale로 변환해 해당 영역 픽셀 값들에 넣어주는 것이다. 이 경우에는 정확성은 떨어지겠지만, 동일한 연산이 반복되는 것을 어느 정도 막을 수 있다. 다른 방법으로는 계산한 값을 배열이나 해시맵 등에 저장해 같은 BGR 값이 들어올 경우 계산 없이 이전 값을 참조하는 방법이다. 하지만 이 경우, 같은 컬러 값이 아닌 비슷한 값만 들어온다면, 참조는 거의 하지 못하고 저장하는 메모리만 잡아먹을 가능성이 있다.

1. 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.

* OpenCV의 cvtColor 함수의 경우, 다양한 컬러 변환을 위한 공식이 정해져 있다. 이 때문에 만약 이 공식 이외의 다른 공식을 사용해 컬러 변환을 하고싶을 경우 OpenCV의 함수를 사용할 수 없게 된다. 또한 24비트, 16비트 등이 아닌 다른 방식으로 색을 저장했을 경우에도 openCV 함수를 사용하기가 까다로워진다. 이런 경우들은 OpenCV 함수가 아닌, 이전 실습처럼 모든 컬러 값들을 직접 연산하여 원하는 값이 나오도록 하는 방식을 이용해야 한다.