**[고소실\_1주차과제]3반\_20151591\_이지현**

**숙제: 효율적인 컬러 변환 및 OpenCV를 이용한 컬러 변환**

**1)** 위 컬러 변환에서 우리는 읽혀지는 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 기술해보자.

이미지의 크기가 512\*512인 파일의 총 픽셀 수는 262144개이며, 한 픽셀 당 R, G, B 3개를 계산해야 하프로 총 786432번 곱하고 더하고 대입해야 한다. 과제에서 사용한 이미지는 작은 편이라 금방 계산이 되지만 크기가 커지거나 많은 이미지를 처리해야 할 경우 속도 문제가 생길 수 있다. 이와 같이 여러 번 계산하는 문제를 해결하기 위해 Lookup Table을 사용한다.

Lookup Table은 일대일(다대일) 함수로, 픽셀 값을 새로운 픽셀 값으로 변환하는 방법을 정의한다. Lookup Table의 특징은 다음과 같다.

* 1차원 배열로 그레이레벨 영상 수준인 경우 256개 항목이 있음.
* 테이블의 i 항목은 대응하는 그레이레벨 영상의 새로운 명암도 값.
* newIntensity = lookup[oldIntensity];

**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 기술해보자. 제공되는 OpenCVcolorConversion 프로젝트를 이용한다.

1. 메모리

과제에서 사용한 addWeighted와 cvtColor는 모두 src와 dst를 입력 값으로 받아야 하는데 이는 원본 이미지와 변환 이미지의 Mat 변수를 각각 따로 설정해주어야 한다. 이 불가피한 선언으로 인해 프로그램이 더 많은 메모리를 차지하게 되면 크기가 큰 데이터나 많은 양의 데이터를 처리할 때 문제가 될 수 있다.

1. 정확도

cvtColor 함수에서 수행할 수 있는 컬러변환은 GRAY, HSV, YCrCb, Luv 가 있다. 이들은 모두 고유한 방법으로 색을 표현하기 때문에 사용자는 이들 중 자신의 목적에 맞는 표현 방법을 선택해야 한다.

그러나 영상 인식 목적에 있어서 이러한 정해진 방식은 수 많은 오차를 불러올 수 있다. 이 오차는 원본 이미지의 밝기나 채도에 따라 다양한 범위로 나타난다. 오차를 줄이기 위해서는 사용자가 원하는 정도의 입력 값을 받아 처리해야 하는데 cvtColor 처럼 변환의 정도가 정해져 있는 함수에서는 그 과정이 어렵다.