

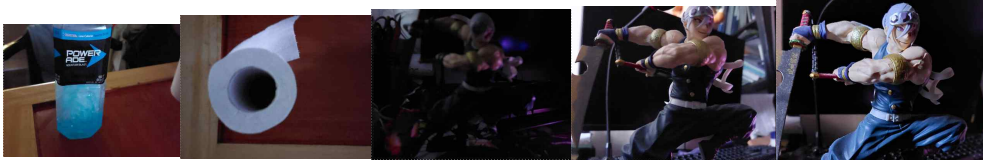
Histogram Equalization을 이용한 contrast 개선

2021204045 이성민

본 리포트는 사전에 촬영한 contrast가 작은 사진을 이용하여 Color Space 변환(필자는 YUV 변환을 이용하였음) 및 RGB 변환을 이용하여 사진을 변환한 뒤, Global Histogram Equalization(GHE), Adaptive Histogram Equalization(AHE), 그리고 CLAHE(Contrast Limited Adaptive Histogram Equalization)을 이용하여 최종적으로 히스토그램을 비교해 이미지의 contrast를 개선하는 과정을 적은 리포트이다. 추가적으로, 개인적인 궁금증에 의해 흑백 사진으로도 시도해 본 것을 미리 전한다.

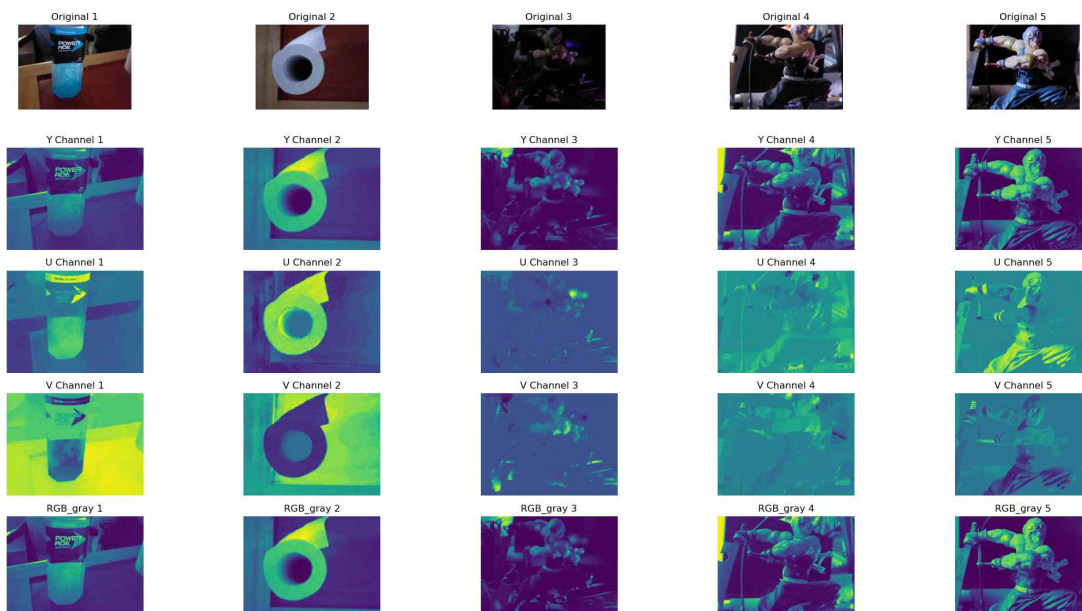
1. 이미지 촬영

먼저, 사전에 촬영한 이미지들은 집안의 전등을 모두 끈 뒤에 사물을 촬영하여 일부러 contrast가 작은 환경을 유도하였다.



3~5번째 사진은 결과 비교를 위해 같은 사물을 contrast가 다른 환경에서 촬영하였다. 3번째 사진은 커튼을 거의 완전히 닫은 채로, 4번째는 커튼을 반만, 5번째는 모두 열어 촬영하였다. 그 뒤에, Adobe Photoshop을 이용해 사진의 크기를 640 X 480으로 모두 맞추는 뒤에 작업을 진행하였다.

2. 색상 처리 (YUV, RGB)



위쪽부터 원본, YUV의 Y, U, V, 그리고 RGB 흑백 변환이다. Y의 경우, Gray색으로 표시하면 원본의 흑백 전환과 똑같다.

```

img_yuv_cl = img_yuv_list[4]
img_yuv_eq = img_yuv_list[4]

y_channel105 = img_yuv4[:, :, 0]
y_channel_cl_05 = img_yuv_cl[:, :, 0]
y_channel_eq_05 = img_yuv_eq[:, :, 0]

clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
clahe_y = clahe.apply(y_channel_cl_05)
y_channel_eq = cv2.equalizeHist(y_channel_eq_05)

y_channel_thresh = cv2.adaptiveThreshold(y_channel105, 255,
                                         cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                         cv2.THRESH_BINARY, 11, 2)

u_channel105 = img_yuv5[:, :, 1]
v_channel105 = img_yuv5[:, :, 2]

img_yuv_cl[:, :, 0] = clahe_y
img_yuv_eq[:, :, 0] = y_channel_eq
img_yuv_thresh = np.zeros_like(img_yuv4)
img_yuv_thresh[:, :, 0] = y_channel_thresh
img_yuv_thresh[:, :, 1] = u_channel105
img_yuv_thresh[:, :, 2] = v_channel105

img_clahe_yuv = cv2.cvtColor(img_yuv_cl, cv2.COLOR_YUV2BGR)
img_eq_yuv = cv2.cvtColor(img_yuv_eq, cv2.COLOR_YUV2BGR)
img_thresh_yuv = cv2.cvtColor(img_yuv_thresh, cv2.COLOR_YUV2BGR)

```

color space 변환이란, 이미지를 RGB에서 다른 색상 공간으로 변환하여 특정 채널(명도/Y)만 조작하는 과정이다. YUV의 경우, Y는 명도, U,V는 색상을 뜻한다.

장점: 명도를 따로 조작할 수 있어 contrast 개선과 같은 밝기 작업에 적합하다. Y만 조작하고, U와 V는 건드리지 않는 식으로 작업할 수 있다.

단점: 화면에 표시하는 용이 아니기에 이를 화면에 그대로 표시하면 RGB로 변환되지 않기에 색상 왜곡이 발생한다.

YUV를 이용해 Y를 조정하는 과정

```

b_channel, g_channel, r_channel = cv2.split(img_or15)

clahe_r = clahe.apply(r_channel)
clahe_g = clahe.apply(g_channel)
clahe_b = clahe.apply(b_channel)

r_channel_eq = cv2.equalizeHist(r_channel)
g_channel_eq = cv2.equalizeHist(g_channel)
b_channel_eq = cv2.equalizeHist(b_channel)

b_channel_thresh = cv2.adaptiveThreshold(b_channel, 255,
                                         cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                         cv2.THRESH_BINARY, 11, 2)
g_channel_thresh = cv2.adaptiveThreshold(g_channel, 255,
                                         cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                         cv2.THRESH_BINARY, 11, 2)
r_channel_thresh = cv2.adaptiveThreshold(r_channel, 255,
                                         cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                         cv2.THRESH_BINARY, 11, 2)

img_clahe_rgb = cv2.merge((clahe_b, clahe_g, clahe_r))
img_eq_rgb = cv2.merge((b_channel_eq, g_channel_eq, r_channel_eq))
img_thresh_rgb = cv2.merge((b_channel_thresh, g_channel_thresh, r_channel_thresh))

```

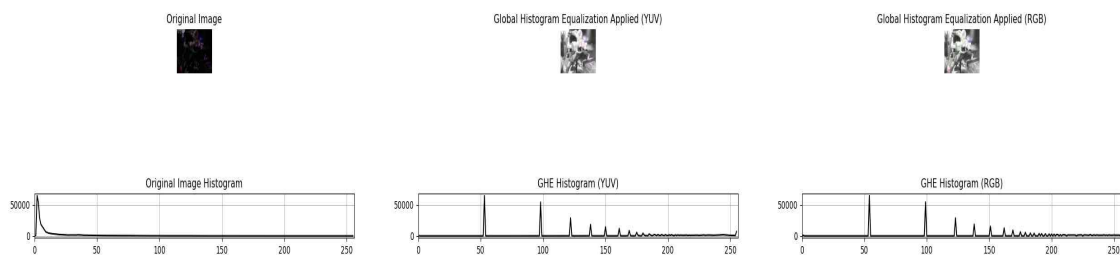
RGB 개별 채널 변환이란, 말 그대로 R, G, B를 개별적으로 분리하여 처리하는 방법이다.

장점: 구현이 간단하고 직관적이며, 화면에 바로 표시할 수 있다.

단점: 밝기 항목이 독립적으로 존재하지 않아 contrast 개선과 같은 밝기 작업엔 값 계산이 복잡해질 수 있다.

RGB 채널을 분리해 작업 후 다시 결합하는 과정

3. Histogram Equalization (global, adaptive, CLAHE)



Global Histogram Equalization (GHE)

GHE란, 이미지의 전체적인 명암을 조정하여 대비를 증가시키는 방법이다. OpenCV의 `cv2.equalizeHist()`로 사용할 수 있으며, 컬러 이미지에서 사용하기 위해선, Grayscale로 변환한 후, 해당 이미지에 GHE를 적용한 다음 다시 컬러 이미지에 매핑하는 것이 일반적이다. 그러나 이 경우 색상 왜곡이 발생할 수 있다.

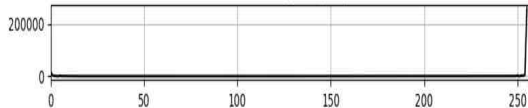
Adaptive Histogram Equalization Applied (YUV)



Adaptive Histogram Equalization Applied (RGB)



AHE Histogram (YUV)



AHE Histogram (RGB)



Adaptive Histogram Equalization (AHE)

AHE는 이미지를 작은 블록으로 나누어 각 블록에 대해 Equalization를 적용한다. 이 방법은 세부 정보와 대비를 개선하는 데 유용하다. skimage의 `exposure.equalize_adapthist()`로 사용할 수 있으며, 컬러 이미지에서 사용하기 위해선, 각 채널에 대해 AHE를 개별적으로 적용하는 식으로 사용할 수 있다. 그러나 사진처럼 대비가 과도하게 증가할 수 있다(자세히 보면 밝기 250 이상의 픽셀 수가 기하급수적으로 증가하는 것을 볼 수 있다.).

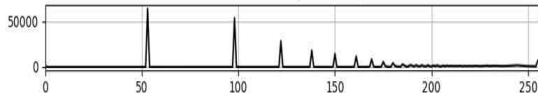
CLAHE Applied (YUV)



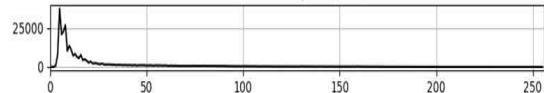
CLAHE Applied (RGB)



CLAHE Histogram (YUV)



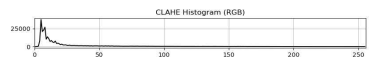
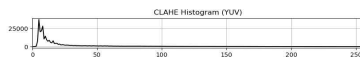
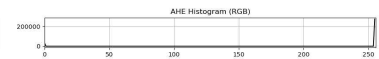
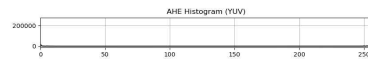
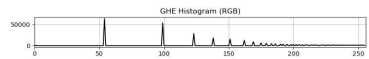
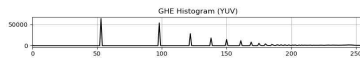
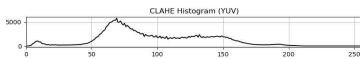
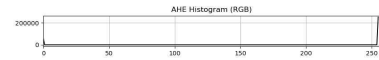
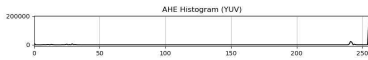
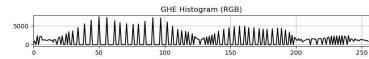
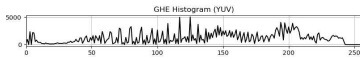
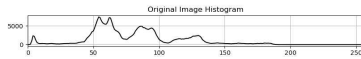
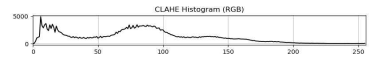
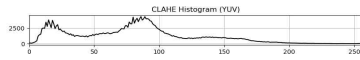
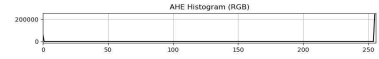
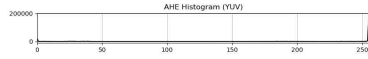
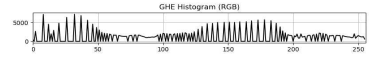
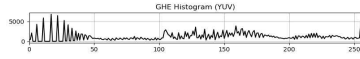
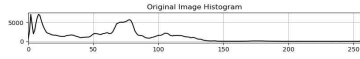
CLAHE Histogram (RGB)

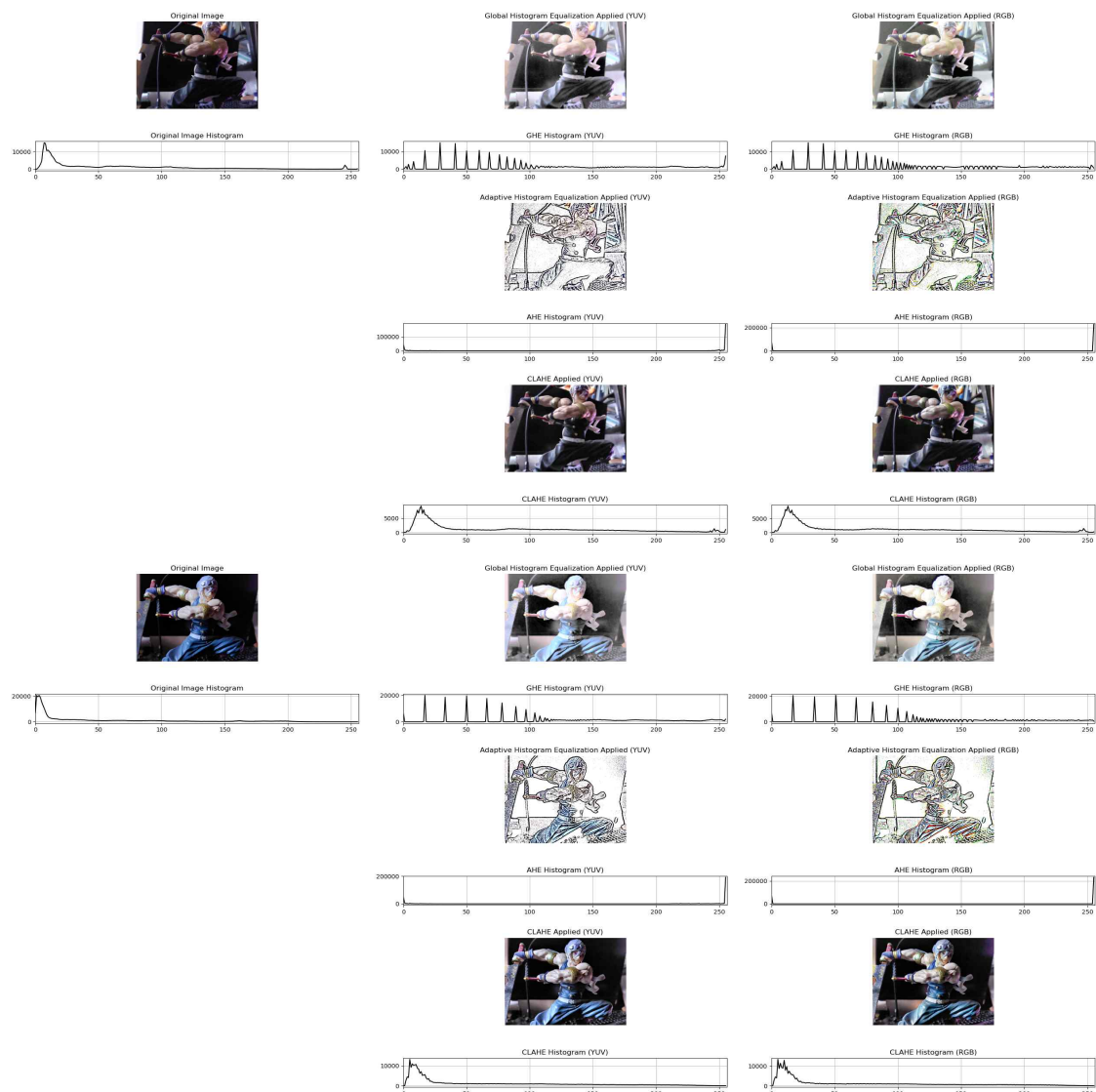


Contrast Limited Adaptive Histogram Equalization (CLAHE)

CLAHE는 AHE의 개선 버전으로, 대비의 과도한 증가를 방지하기 위해 대비의 한계를 설정한다. 특히 노이즈가 많은 이미지에서 유용하다. OpenCV의 `cv2.createCLAHE()`로 사용할 수 있으며, CLAHE 또한 각 채널에 대해 개별적으로 적용할 수 있다.

4. 결과와 히스토그램





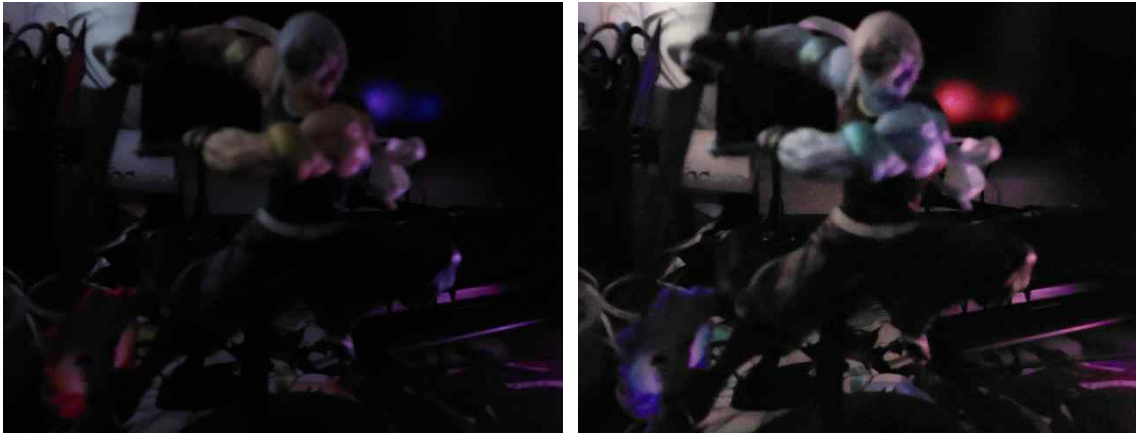
5개의 사진에 대한 결과들이다. 사진이 잘 보이지 않는다면 같이 첨부된 사진을 확인하길 바란다.

먼저 Original Image를 따로 YUV와 RGB시킨 다음, 이들을 각각 equalization 하였다.

GHE의 경우, 전체적으로 경향성이 잘 유지되었으나, 원본처럼 색 당 픽셀의 수가 자연스럽게 못하고 뚝뚝 끊긴 모습을 볼 수 있다.

AHE는 대비가 너무 심하여 사진의 경우 테두리밖에 볼 수 없으며, 히스토그램 또한 특정 밝기(250~)에만 몰려 있는 모습을 볼 수 있다.

이에 반해 **CLAHE**는 원본의 경향성을 잘 유지하면서도, RGB쪽은 GHE처럼 색 당 픽셀의 수가 뚝뚝 끊기지 않고 자연스럽게 이어지는 모습을 볼 수 있다.



좌- 원본. 우-CLAHE로 contrast를 개선한 진.(RGB)

원본 사진이 어두운 환경에서 촬영되어 초점이 잘 맞지 않는다는 것을 생각하면 다른 방식에 비해 contrast 개선이 잘 이루어졌다 볼 수 있다.



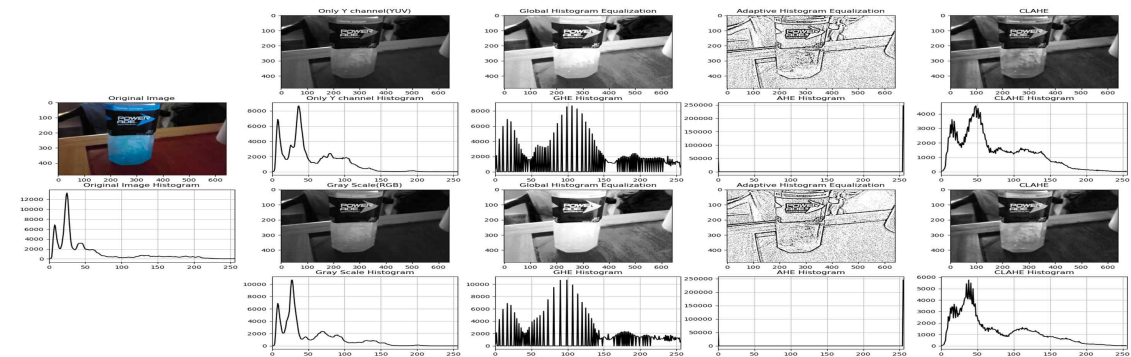
좌-GHE로 개선. 우-AHE로 개선. (RGB)

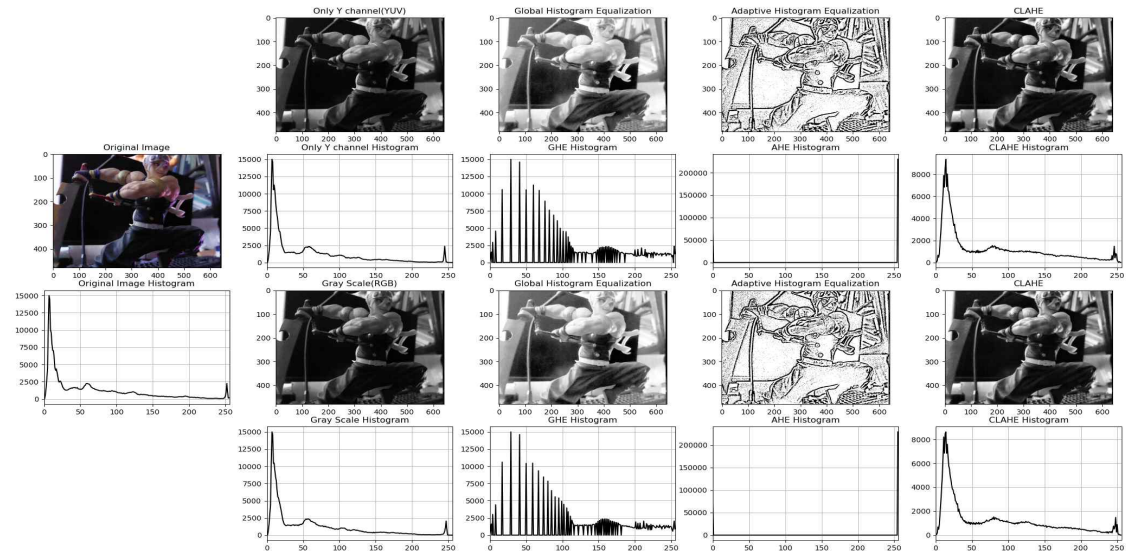
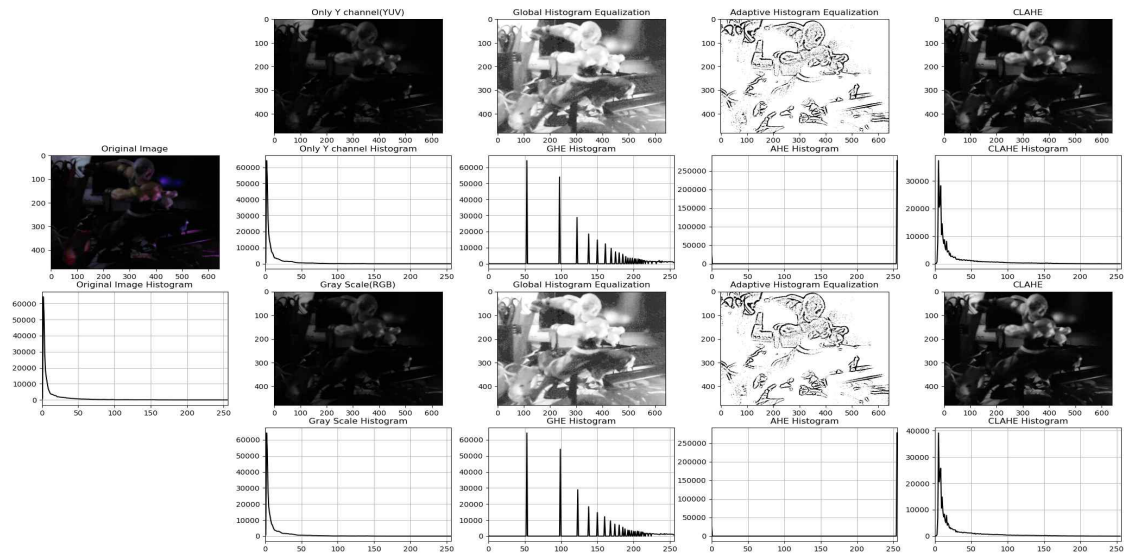
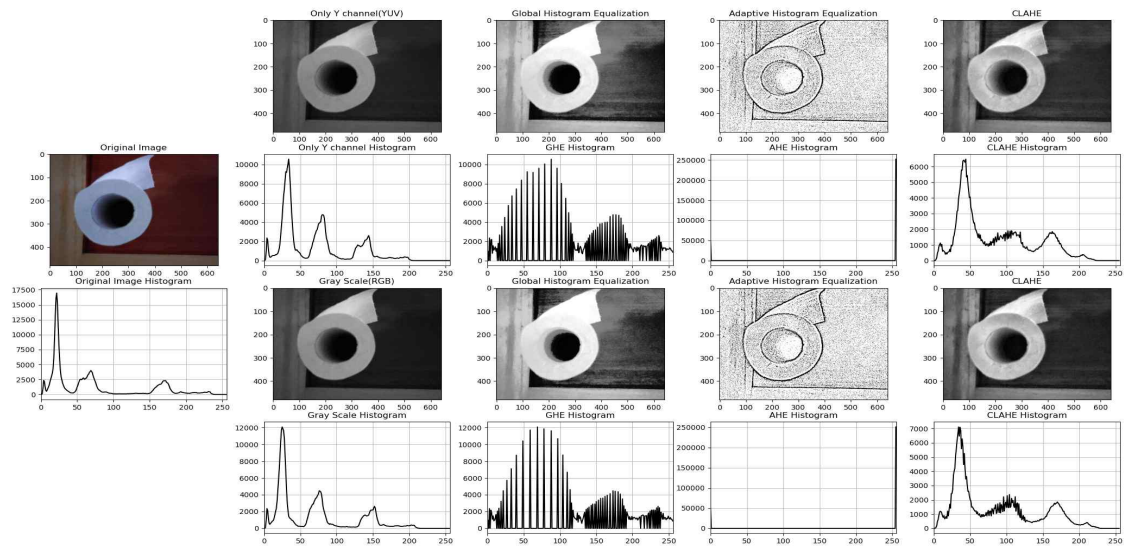
상대적으로 GHE가 나은 모습을 볼 수 있다. AHE는 형태밖에 남지 않았다...

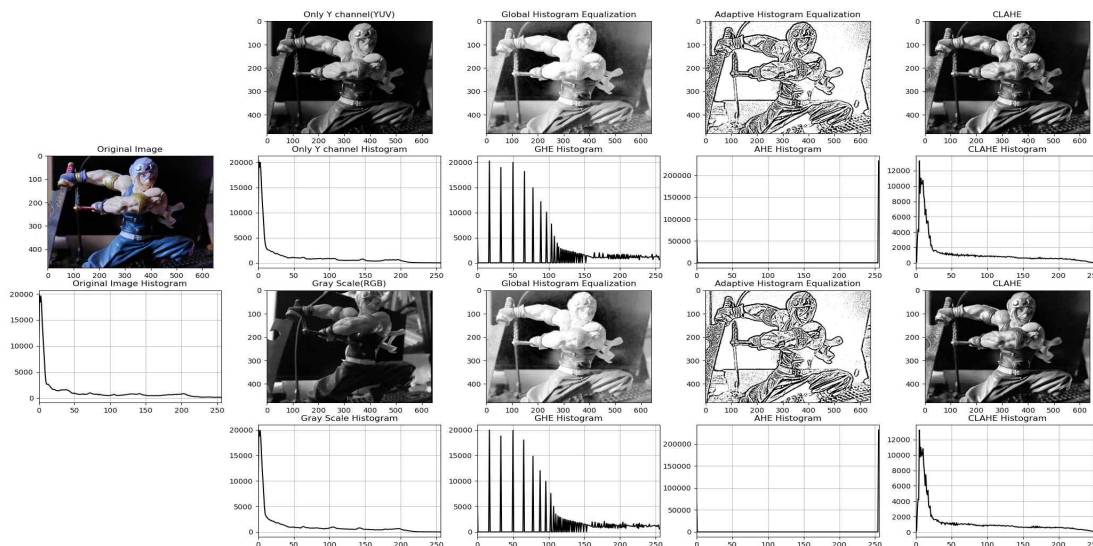
히스토그램을 보았을 때나 결과 사진을 보았을 때나, 원본 사진이 어두울수록(contrast가 낮을수록) 개선된 사진의 퀄리티가 낮아진다는 결론을 얻을 수 있다.

5. 흑백

이전에 Histogram Equalization은 일반적으로 Grayscale에서 적용된다고 하였다. 지금까지는 컬러 이미지를 대상으로 equalization을 시행하였지만, 기존 목적대로 흑백에서 사용한다면 효과가 더 극대화될까? 라는 물음으로부터 이 단계가 시작되었다.







색상과 밝기를 따로 작업할 수 있는 YUV의 경우 밝기인 Y만 따로 떼어다가 작업했다. RGB는 밝기를 따로 나눌 수 없어 Grayscale하여 작업했다. 흑백처리만 한 사진의 경우에는, 아주 미세한 차이를 제외하면 서로 같음을 알 수 있다. 이에 따라 컬러와 마찬가지로 YUV와 RGB가 서로 큰 차이를 보이지 않는다. 애초에 3개의 Histogram Equalization 모두 컬러 이미지도 가능하게 설계되어 있는 만큼 흑백이든 컬러든 큰 차이가 없는 듯 하다.

6. 결론

본 리포트에서는 contrast가 낮은 이미지를 대상으로 YUV, RGB 변환과 여러 Histogram Equalization 기법(GHE, AHE, CLAHE)을 이용하여 이미지를 개선하는 과정을 살펴보았다. 촬영한 이미지들은 일부러 낮은 Contrast 환경에서 촬영하였으며, 이로 인해 Contrast 개선의 필요성이 더욱 두드러졌다.

Color Space 변환: YUV 변환을 사용하여 명도(Y)만을 조정하는 방식은 Contrast 개선에 효과적임을 확인하였다. YUV의 경우 색상(U, V)을 유지하면서 밝기 조정이 가능하며, 명도를 독립적으로 조작할 수 있다는 장점이 있다. 반면, **RGB 개별 채널 변환**은 구현이 간단하고 직관적이지만, 명도를 독립적으로 조정하기 어려운 점이 아쉬웠다.

Histogram Equalization: GHE는 전체적인 명암 조정에 효과적이지만 색상 왜곡이 발생할 수 있으며, AHE는 과도한 대비 증가로 인해 이미지가 희미해질 수 있다는 단점을 보였다. 반면, CLAHE는 이러한 단점을 개선하여 세부 정보는 유지하면서도 과도한 대비 증가를 방지하였다. 결과적으로 CLAHE는 원본의 경향성을 잘 유지하며, 자연스러운 색 분포를 보여주어 최상의 성능을 발휘하였다.

흑백 이미지에서 Histogram Equalization을 시행한 결과, 컬러 이미지와 유사한 결과를 나타냈다. 이는 해당 Histogram Equalization들이 흑백 이미지와 컬러 이미지를 가리지 않고 효과적으로 적용될 수 있음을 나타낸다. 또한 원본 사진의 조도와 Contrast 수준에 따라 개선된 이미지의 품질이 달라질 수 있다. 향후에는 다양한 이미지 환경에서 이 기법들을 비교하고 분석하여, 최적의 Contrast 개선 방법을 탐구하는 것이 필요할 것으로 보인다.