# Numerical Methods

# HW03(Adaptive interpolation)

담당교수 : 심동규 교수님

학 번 : 2019202050

성 명 : 이강현

## Introduction

이번 과제에서는 Adaptive Interpolation Method를 적용하여 이미지내의 특성을 고려하여 그 특성에 맞는 여러 interpolation filter를 optimal하게 생성하고 interpolation 해본다. interpolation 한 후 생성된 이미지를 원본 이미지와 PSNR를 비교해보며 Adaptive Interpolation Method의 성능을 확인한다.

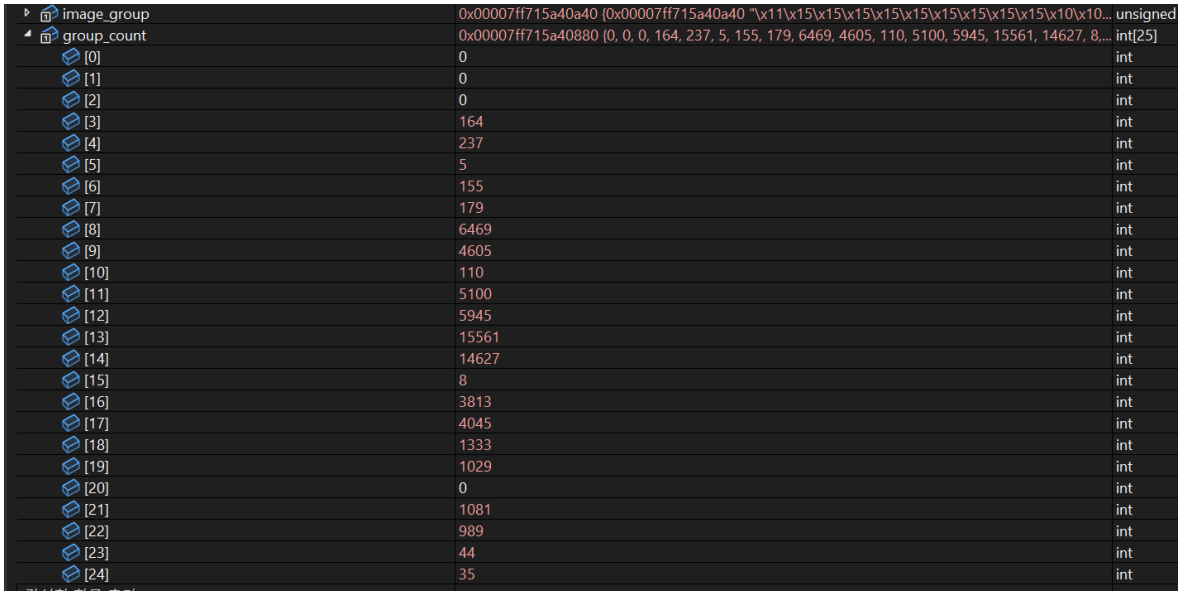
## Adaptive Interpolation Method

input image의 pixel을 중심으로 주변 픽셀들의 활동성, 방향성에 따라 pixel마다 그룹을 형성하고 각 그룹에 최적화된 filter를 생성하여 이를 통해 사진의 지역적인 특성을 고려한 interpolation 기법이다. 영상압축, 이미지 interpolation에 많이 사용되며 특성을 고려하기에 고정된 필터를 이용해 interpolation 하는 방법보다 더 좋은 성능을 뽑아낼 수 있다.

## Experiments

먼저 pixel classification을 진행한다.

사진마다 각각 활동성의 최댓값, 최솟값이 다르기 때문에 고정적인 어떤 범위로 사진들을 분류하는 것은 사진에 adaptive하지 않다. 따라서 vertical한 특성을 뽑아내는 필터와 horizontal한 특성을 뽑아내는 필터를 이용하여 모든 픽셀에 대해 요소별 곱셈을 진행하고 매 픽셀 연산시 이를 더한 값을 배열에 저장해 두었다. 모든 픽셀에 대한 연산이 끝나면 배열을 순회하여 가장 큰 값과 가장 작은 값을 찾아내고 이를 이용하여 등간격으로 범위를 나누어 0~4의 값에 매핑하였다. 이렇게 활동성을 뽑아내고 방향성은 이미지와 연산하였을 때 가장 큰 값을 가지게 끔 하는 필터의 방향성을 따라가게끔 구현하였다. 또한 정반대의 방향의 연산이 같을 때 방향성이 없다고 가정하고 이를 방향성 0에 매핑하였다. 이로서 0~4의 크기인 활동성과 방향성을 구했고 이를 이용해 0~24까지의 그룹으로 분류를 완료하였다.




분류된 그룹들의 모습이다. 주로 0부터 24까지에서 중심부에 픽셀들이 많이 위치하는 것을 확인할 수 있다.

다음은 이미지를 512x512사진에 2칸씩 떨어뜨려 펼친 후 각각 픽셀들의 오른쪽 아래 대각선 방향 세가지 픽셀을 interpolation하기 위한 adaptive filter를 만든다.

least square optimization을 이용하면 최종적으로 adaptive filter를 아래와 같이 구할 수 있다.

adaptive filter(F) = = $(X^TX)^{-1}X^TY$

전체 그룹 수 25번만큼 반복문을 돌고 한번 돌때마다 adaptive filter 7x7이 3개(H,V,D)씩 만들어진다.

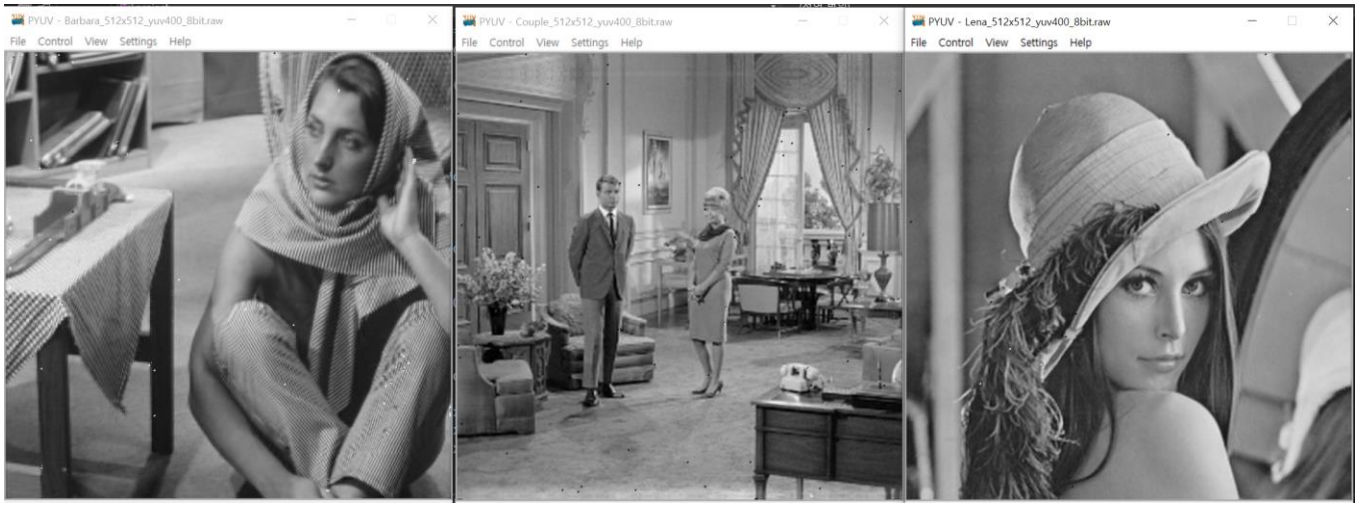
해당 반복문 안에서 위에서 분류한 그룹현황을 기반으로 배열의 크기를 동적할당하여 같은 그룹에 해당하는 픽셀 총 수(i) x 49(7x7을 1-D로 풀어냄)을 저장해두고 이를 X로 사용한다.

정답데이터 Y는 원본 이미지 512x512에서 interpolation할 픽셀과 대응되는 픽셀값으로 채운다.

따라서 i x 1의 형태를 가진다.

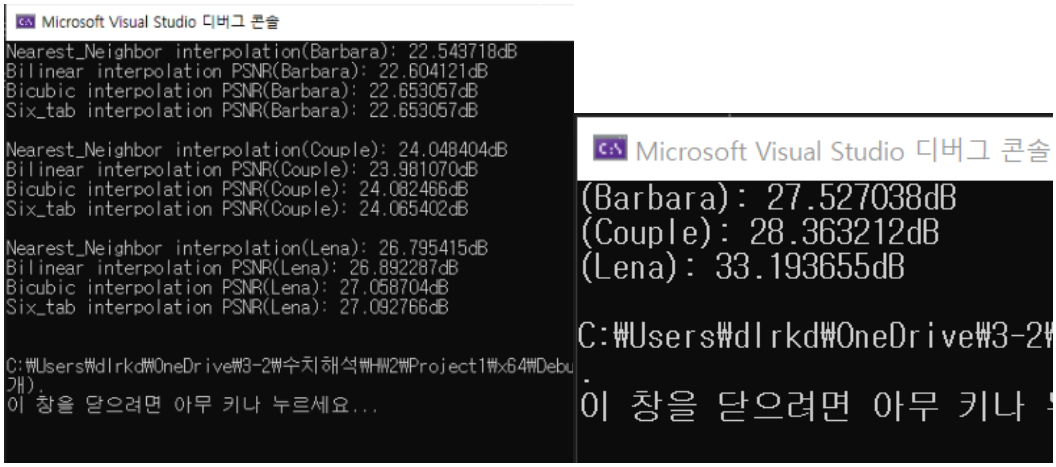
전체 연산을 진행하면 $X^T$ = 49 x i이고 $X$는 i x 49이므로 $X^TX$ = 49 x 49 형태이며 이것의 역행렬 또한 49 x 49 이다. 이를 다시 $X^T$와 내적하면 49 x i가 되고 이를 Y와 내적하면 49 x 1의 형태 이를 2-D로 바꾸면 7x7의 F를 구할 수 있는 것이다. 이때 정답데이터 Y의 배열만 H,V,D 각각 다르게 해주어 동일한 X와 연산을 진행하면 세 개의 F를 구할 수 있다.

최종적으로 interpolation 하는 값은 XF로 구할 수 있다. 이를 통해 interpolation을 완료하고 실제 이미지를 확인해본다.

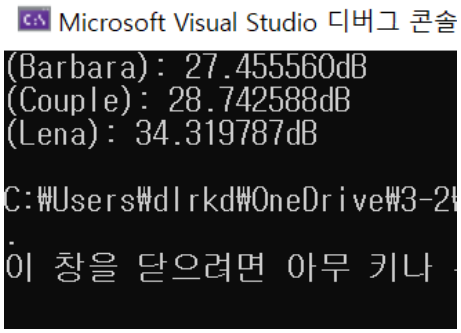



결과적으로 전반적인 이미지는 interpolation이 잘 되었으나 자세히 보면 경계부분에서 살짝 값이 이상해서 0 또는 255로 interpolation이 되는 경우가 생겼다.

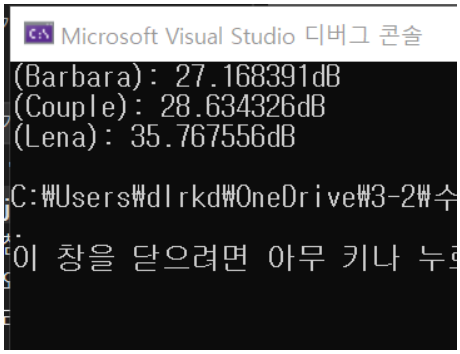
PSNR을 이전 과제 128x128에서 interpolation한 경우랑 비교해보자.



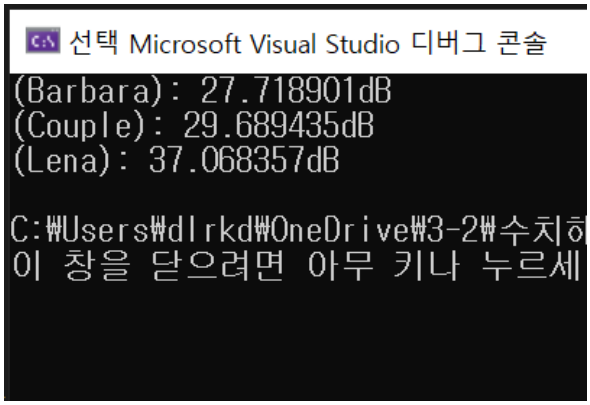

확실히 128x128에서 interpolation한 것보다 256x256으로 interpolation한 psnr이 전체적으로 좋아지긴 하였다. 하지만 이는 interpolation 전 데이터의 정보량이 훨씬 많기 때문에 당연한 결과이다. 따라서 추가적으로 adaptive interpolation임을 고려해서 데이터셋의 분포를 확인하고 코드를 다소 수정하여 성능을 끌어 올려보았다.



활동성을 vertical과 horizontal로만 측정하지 않고 4가지 필터와 내적한 전체 값을 활용하였을 때는 Barbara, Couple에선 약간 감소했으나 Lena에서 좋은 성능을 보였다.




패딩을 0으로 하고 그룹이 중앙쪽에 몰리는 현상을 고려하여 범위를 등간격으로 하지 않고 중앙쪽 범위를 더 세부적으로 하기 위해 중앙 범위를 3으로 나누어 활동성을 분류한 결과 다른 부분에서는 오히려 역효과였으나 Lena사진에서 눈에 띄게 PSNR이 증가한 것을 확인할 수 있었다.




추가적으로 경계에서 발생하는 interpolation문제에 대해 interpolation한 값이 많이 픽셀값을 많이 벗어나 interpolation이 잘 되지 않을 때는 bilinear interpolation 방식을 이용하여 interpolation하도록 조건을 추가하였다. 최종적으로 PSNR을 위와 같이 뽑아낼 수 있었다.

경계부분에서의 픽셀이 안정화된 모습

# Conclusion

Adaptive interpolation 방식을 구현하면서 전반적인 흐름은 픽셀을 그룹화하고 그룹마다 최적화된 필터로 interpolation한다였지만 픽셀을 어떤 기준으로 나눌 것인가? 그룹마다 행해지는 최적필터를 구하는 과정속에서 어떻게 error를 줄일 것인가? 필터를 이용해 interpolation한 후 추가적으로 할 수 있는 interpolation은 무엇이 있을까? 와 같은 여러 고민을 해볼 수 있었다. 먼저 어떤 기준으로 나눌지는 강의 자료를 참고하여 구현했지만 더 세부적으로 방향성을 고려했다면 더 이미지의 경계부분에서 interpolation이 제대로 안되는 픽셀의 빈도를 줄일 수 있었을 것으로 예상한다. 또 활동성을 나누는 범위를 등간격으로 하지 않는다는 접근은 PSNR 측면에서 성능의 향상을 확인할 수 있었고 이는 사진의 특성에 따라 다양하게 적용할 수 있어보였다. 또 최적 필터를 구하는 과정에서 least square optimazation을 사용했으나 다른 알고리즘을 사용할 수도 있을 것 같고 패딩의 값을 zero로 하니 좀 더 경계가 뚜렷해져서 인지 성능이 좋아지는 경험 또한 하게 되었다. 마지막으로 interpolation을 마친 후 예외적인 픽셀에는 다른 interpolation을 추가 적용하여 PSNR을 올리고 사진을 더 매끄럽게 할 수 있었다. 이번 프로젝트를 통해 이미지 interpolation 성능의 향상에 있어서 효과를 보는 사진도 있었고 그렇지 않은 사진도 있었는데 이는 사진의 특성을 파악하는 것이 interpolation 기법을 선택하는데 있어서 꼭 필요한 선행작업임을 알게 되었다.