수치해석

HW03

담당교수 : 심동규 교수님

학 번 : 2021202058

성 명 : 송채영

1. Introduction

이번 과제는 256x256 크기의 이미지를 512x512 의 크기로 upscaling 하는 것이다. 이를 위해 Adaptive Interpolation 을 사용한다. Interpolation 된 이미지와 원본 고해상도 이미지 간의 오차가 적어지도록 하는 interpolation filter set 을 도출한 후 도출한 filter set 을 활용하여 interpolation 을 수행해본다. Interpolation 한 결과와 Ground Truth 폴더에 있는 512x512 크기의 이미지를 PSNR(Peak Signal-to-Noise Ratio)을 통해 평가해본다.

2. Adaptive Interpolation Method

Adaptive interpolation method(적응 보간 방법)은 입력 이미지의 픽셀을 관련성 있는 그룹으로 나눈 후 각 그룹에 대해 최적의 수평, 수직, 대각선 반픽셀 보간 필터를 도출하는 방법을 말한다. 그룹별로 데이터의 특성을 고려하여 수평, 수직, 대각선 방향에서의 최적의 중간 값을 찾아내어 보간을 수행한다. 각 그룹에 최적화된 보간 필터를 적용하여 이미지의 픽셀을 부드럽게 만들며 정확성을 향상시킬 수 있다.

Pixel classification 은 입력 이미지의 정수 픽셀을 해당 픽셀을 중심으로 하는 주변 활동성 및 방향성을 고려하여 여러 그룹으로 분류하는 과정이다. 이를 위해 특정 픽셀을 중심으로 하는 주변 영역의 활동성과 방향성을 계산하게 된다. 활동성 맵핑(양자화)의 경우 Mask-V 와 Mask-H 와 nxn pixel 을 내적 연산한 결과를 더하여 Max 부터 Min 까지 나열한 후 0~4 사이의 그룹으로 그룹핑 해야 한다. 방향성을 결정하는 경우 Mask-V, Mask-H, Mask-D135, Mask-D45 와 nxn pixel 을 내적 연산한 결과를 비교하여 가장 큰 값에 해당하는 방향의 그룹이 결정된다. 0 도에 해당하는 Mask-H 의 경우 그룹 1, 45 도에 해당하는 Mask-D45 의 경우 그룹 2, 90 도에 해당하는 Mask-V 의 경우 그룹 3, 135 도에 해당하는 Mask-D135 의 경우 그룹 4 에 해당하며 그 외는 0 에 해당한다. 활동성 맵핑에 해당하는 그룹을 A, 방향성 결정에 해당하는 그룹을 D 라고 한다면, A x 5 + D 를 통해 0 에서 24 사이의 그룹으로 정규화한다.

Filter set optimization 은 integer-pixel 그룹마다 interpolated 픽셀과 ground truth 픽셀 간 평균 오차를 최소화하는 각 H, V, D half pixel interpolation filter 를 추출하는 것을 말한다. 평균 제곱 오차를 최소화하는 방법으로 Least square optimization 방법을 활용하는데, Lease square optimization 은 주어진 데이터와 모델간의 차이를 최소화하는 파라미터를 찾는 통계적인 최적화 방법을 말한다.

$$F = (X^T X)^{-1} X^T Y$$

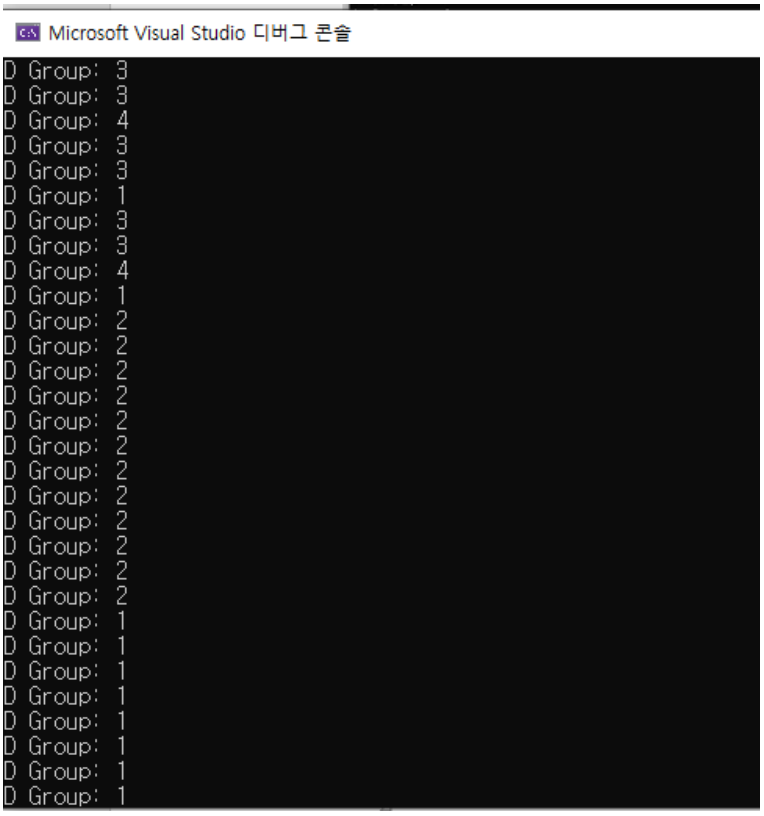
위 식을 사용하여 최소 제곱 최적화를 통해 선형 회귀 모델의 계수를 추정하는 과정을 나타낼 수 있다.

3. Experiments

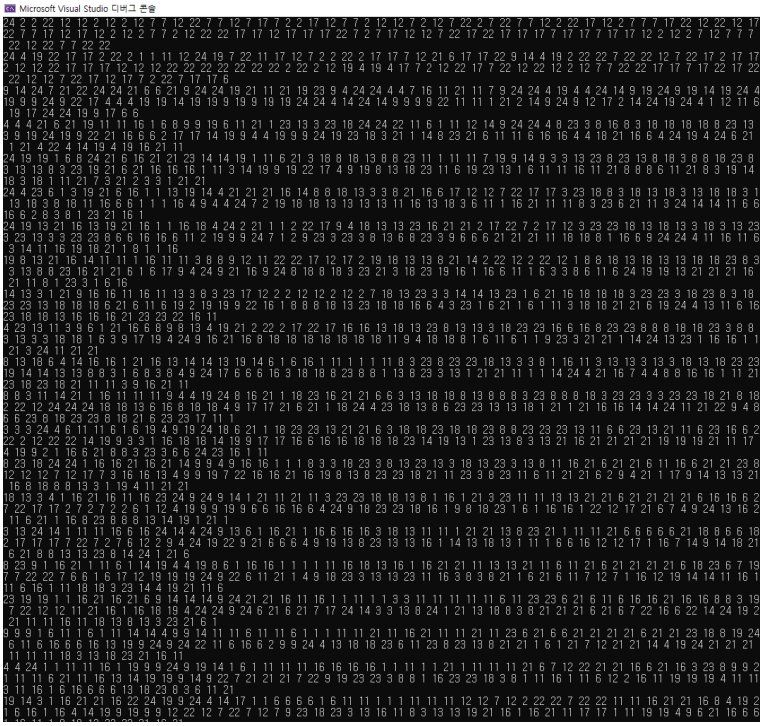
우선 각 pixel 마다 7x7 만큼 뽑아내서 pixel classification 을 진행해야 하므로 padding 을 진행하였다.

상하좌우로 3 개의 픽셀이 추가적으로 필요했기 때문에 padding value 를 3 으로 해주었다. 결과를 살펴보면 세 사진 모두 패딩이 잘 된 것을 확인할 수 있다.

이어서 pixel classification 을 수행하였다. 우선 활동성 부분이다. 패딩된 이미지를 바탕으로 7x7 pixel 을 뽑은 후 Mask-V 와 Mask-H 와 내적한 결과를 더하였다. 그 후 Mix 값과 Min 값을 나열한 후 0 부터 4 사이의 그룹으로 그룹핑을 하였다. 방향성 부분은 패딩된 이미지를 바탕으로 7x7 pixel 을 뽑은 후 Mask-V, Mask-H, Mask-D135, Mask-D45 와 내적 연산을 하였다. 정확히 이야기 하면 요소별 곱셈을 진행하였다. 그 후 값이 가장 큰 값을 선택하였다. 만약 0 도에 해당하는 Mask-H 가 가장 크다면 1 에 저장을, 45 도에 해당하는 Mask-D45 가 가장 크다면 2 에 저장을, 90 도에 해당하는 Mask-V 가 가장 크다면 3 에 저장을, 135 도에 해당하는 Mask-D135 가 가장 크다면 4 에 저장을 하도록 하였다. 해당되지 않는 이외의 각도는 0 에 저장하도록 하였다. 그룹핑이 골고루 됐는지를 확인하기 위해 각 그룹을 출력해본 결과를 아래에 첨부하겠다.



A*5 + D 를 적용한 후 0 에서 24 사이의 그룹으로 그룹핑 한 결과도 출력해보았다.



256x256 만큼의 결과가 출력되는 것으로 자세히 보이지는 않지만 0 에서 24 사이로 그룹핑이 잘 된 것을 확인할 수 있었다.

이후 fileter set optimization 부분이다. Y 값 까지는 구했지만, F 값을 계산한 후 F 와 7x7 요소별 곱셈한 결과를 512x512 공간에 저장한 후 512x512 의 원본 데이터와 비교해서 PSNR 구하는 부분을 다 구현하지 못하였다. 따라서 출력 결과가 없기 때문에 구현한 부분까지의 코드를 줄글로 설명하겠다.

간단하게 세개의 중첩 루프를 활용하여 25 의 그룹에 대해 7x7 크기의 block 을 추출한 후 해당 블록에 대해 최소 제곱 최적화를 수행하여 세 가지 방향의 픽셀인 h, v, d 를 뽑아내는 과정까지 구현하였다. Inverse matrix 와 transpose 그리고 내적 연산을 하는 함수를 구현한 후 main 함수에서 각각을 저장한 후 연산에 써주었다.

4. Conclusion

우선 이번 과제를 끝까지 진행하지 못한 점이 아쉬웠다. 또한 그룹핑을 하고 filtter set optimization 을 수행하면서 중간 중간 과정을 디버깅을 통해 확인하여야 했는데, 디버깅하는 과정이 생각보다 어려웠던 것 같다. 위에서 뽑아본 그룹핑을 한 결과도 0 에서 4 사이, 그리고 0 에서 24 사이의 숫자로 그룹핑이 잘 됐고 골고루 숫자가 나온다 정도만 알 수 있었지, 그 이상은 알지 못하였다. 또한 Adaptive interpolation method 자체에 대한 자료도 거의 없어 ppt 만으로 이해하는데 한계점도 있었던 것 같다. 특히 inverse metrix 를 계산하고 transpose matrix 를 계산하여 F 를 구하는 과정에서도 이게

제대로 계산이 됐는지 알 방법이 없어 시간이 많이 걸렸던 것 같다. 다 구현을 하지는 못하였지만, 예상한 결과에 대해 서술해보겠다. 우선 지난 2 차 과제에서 구현해본, bicubic 이나 six-tab 보다 훨씬 좋은 성능을 얻을 수 있을 것이라고 예상을 하고 과제를 시작하였다. 하지만 주변 친구들의 PSNR 결과가 26, 27 정도 나왔다는 사실을 듣고 예상 외의 결과라고 생각했다. PSNR 면에서 큰 차이는 없었지만, 사진을 직접 눈으로 보니 지난 과제로 수행했던 interpolation 방법들 보다 화질면에서도 더 좋게 나오고 원본 이미지와 거의 유사하게 나온 것 같았다. 다만 흰 반점들이 조금씩 보이는 것으로 보아 그룹핑을 하는 과정에서 그룹핑이 안 된 픽셀들이 존재하는 것 같았다. 그룹핑을 할 때 최대한 동일한 개수를 가지도록 하는 것이 좋은 결과를 나오도록 만들 것이라는 것도 알 수 있었다. 학기가 끝나고 완성하지 못 한 코드를 다시 구현해볼것이다.