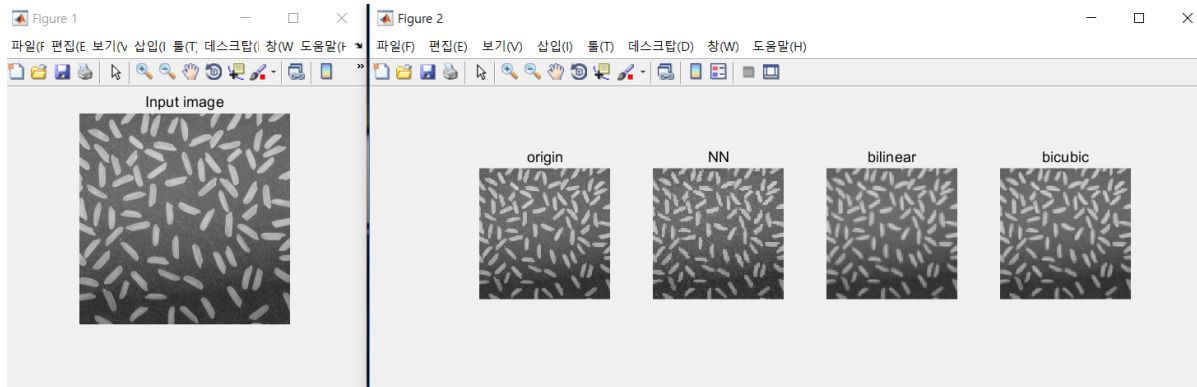


LAB2. Imresize

LAB. 2-1



소스 파일은 첨부파일에 있습니다.

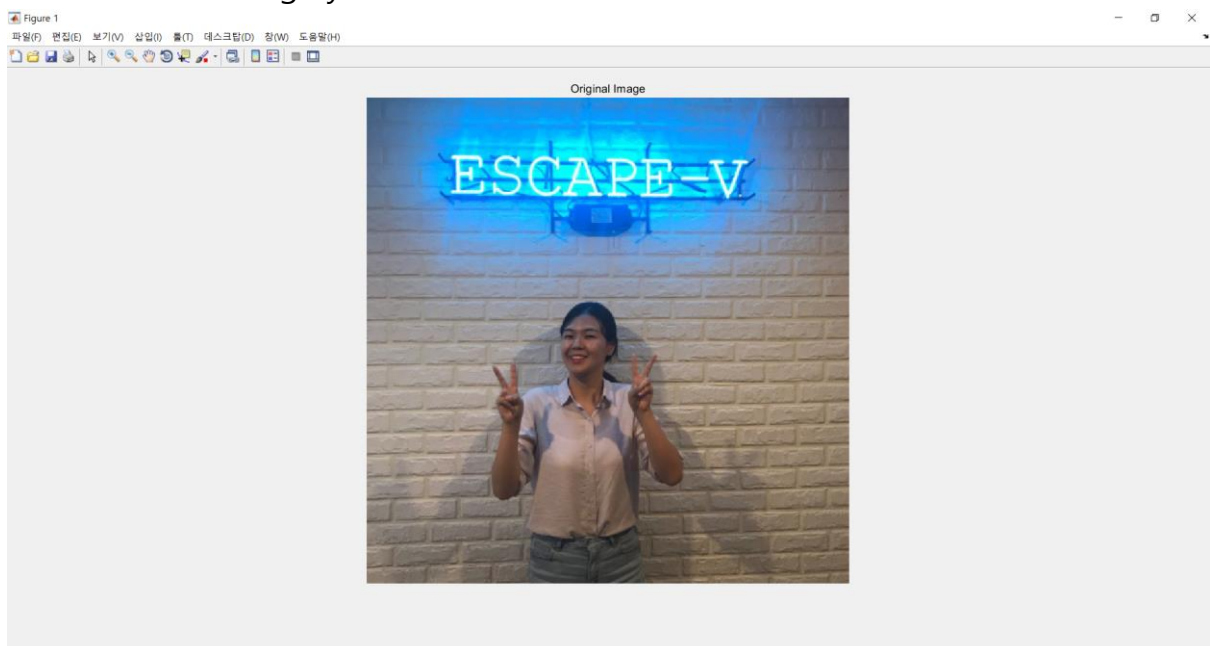
LAB. 2-2

Resize your photo to 0.85 times smaller and 1.24 times larger than the original one.

1) Make resize functions ([myResizeNN](#), [myResizeBil](#)) for NN, Bilinear interpolation (your own code not built-in)

2) Apply your functions to your photo.

원본 사진은 아래와 같다. 아래에 시행한 interpolation 은 출력 시 3 채널로 출력되는 오류가 있어 gray로 변환하여 진행하였다.



Nearest Neighbor interpolation을 실시하였다.

—Scale : 0.85 , time : (경과 시간은 0.260304초입니다.)



—Scale : 1.24 , time : (경과 시간은 0.230420초입니다.)



Bilinear interpolation을 실시하였다.

—Scale : 0.85 , time : (경과 시간은 0.207138초입니다.)

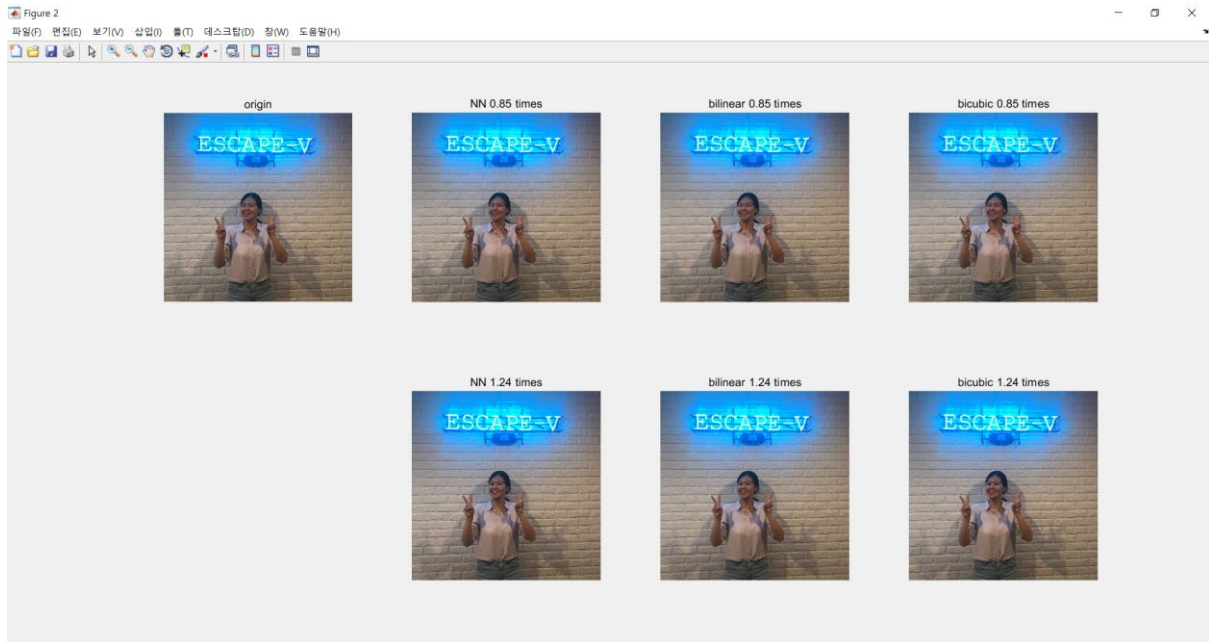


—Scale : 1.24 , time : (경과 시간은 0.287819초입니다.)



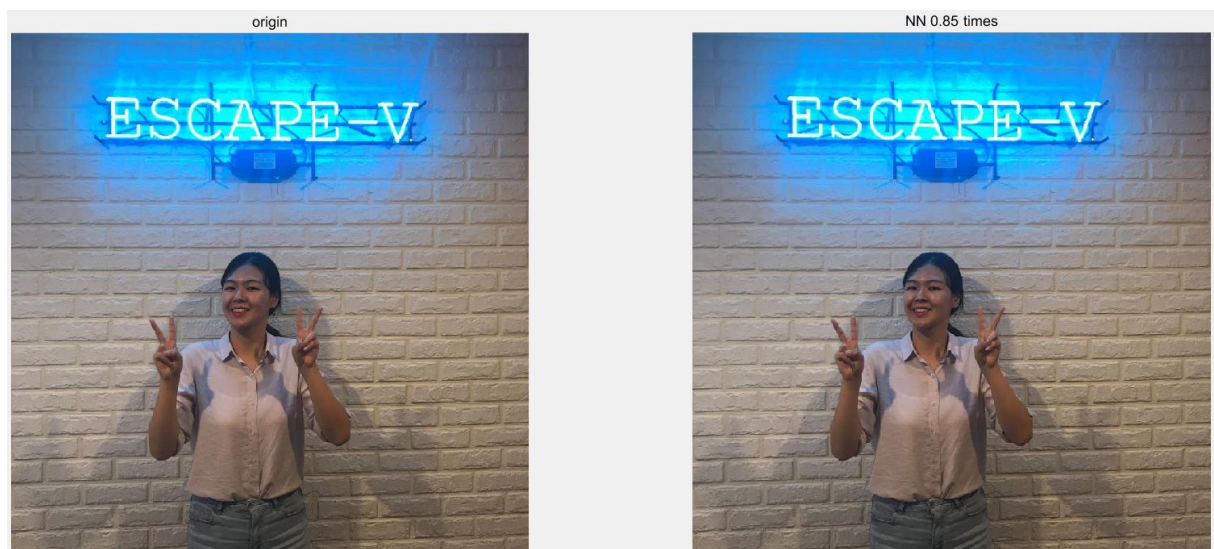
3) Compare them with the results obtained by *imresize* in matlab.

아래는 matlab built-in 코드로 interpolation을 실시하였다.

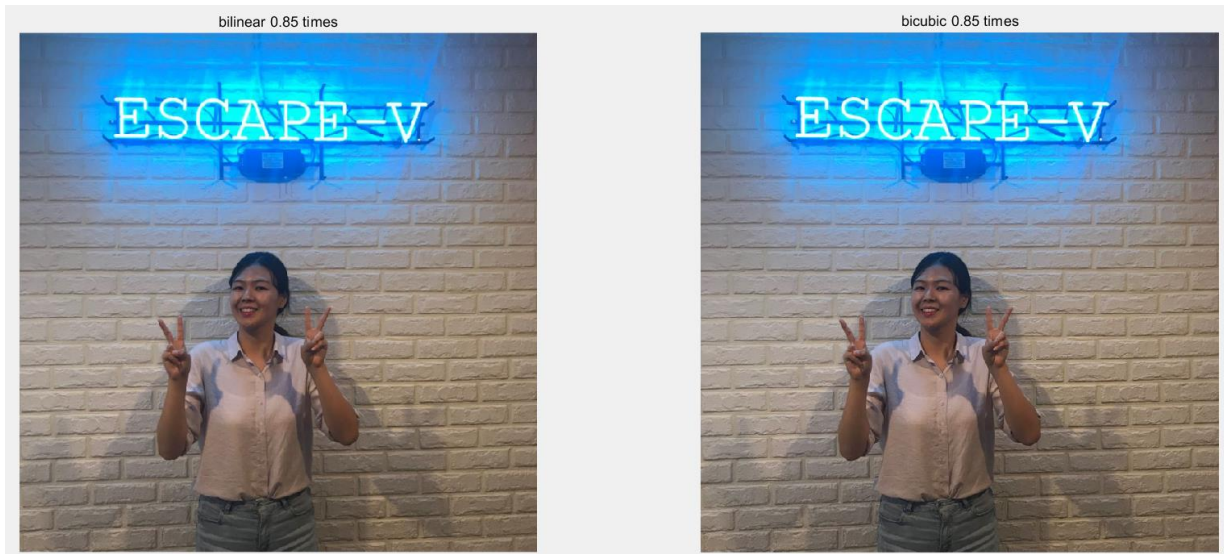


잘 안보여 확대하여 캡처해보았다.

Resize my photo to **0.85** times larger than original one.

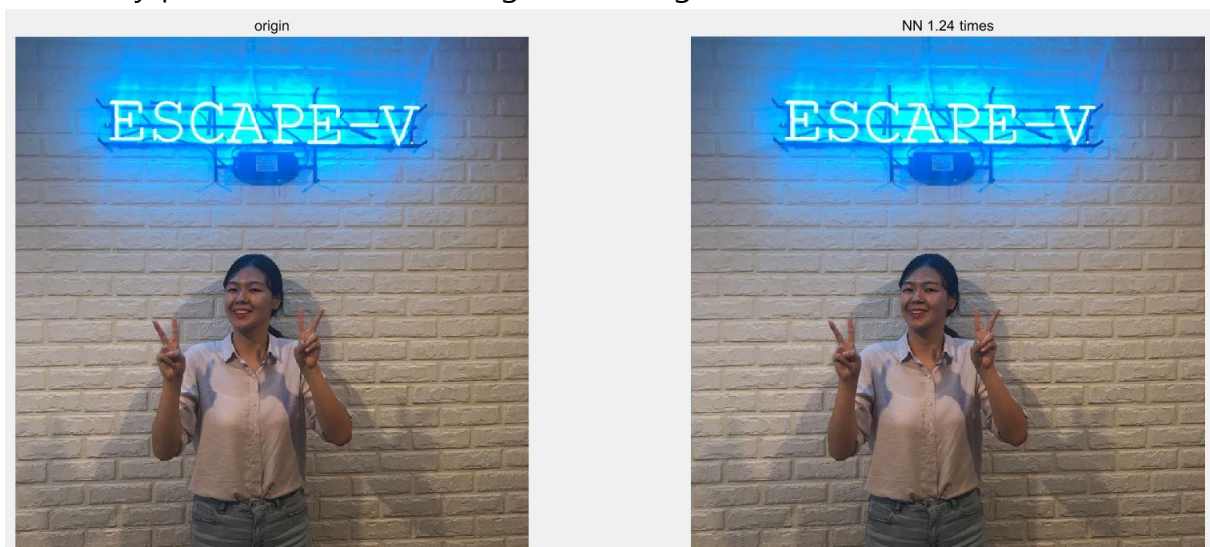


NN을 0.85배 시행했을 시 확대하여 보면 많이 깨진 것을 알 수 있다.



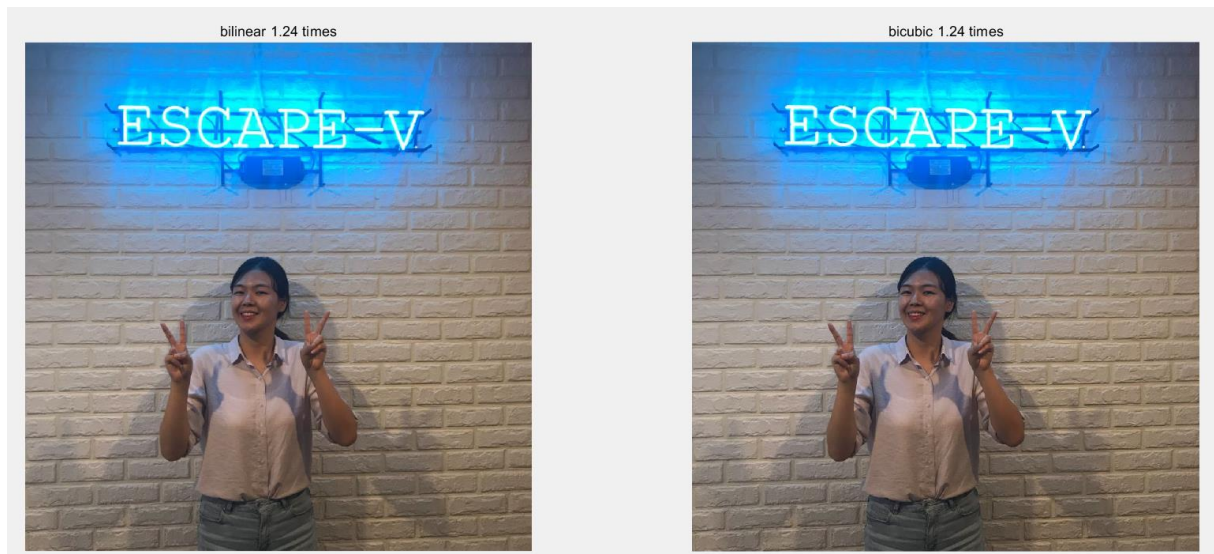
NN보다 bilinear가 훨씬 부드러워졌지만, bicubic과 비교하여 보면 조금 덜 부드럽다.

Resize my photo to **1.24** times larger than original one.



원본 사진보다는 1.24배를 했을 때도 깨진 것을 볼 수 있다.

이는 픽셀을 쪼갬기 때문이다. 얼굴이나 손가락 부분을 자세히 보면 NN 1.24 times는 네모 픽셀이 더 잘 보인다. 원본보다는 덜 자연스럽다.



Bilinear을 시행할 경우 거의 원본과 비슷했다. NN보다 부드러운 픽셀 모습이 보였고, bicubic과 비교하여서는 큰 차이점을 보지 못했다.

```
>> I=imread('VV.jpg'); %load an gray image
figure(1); imshow(I); title('Input image');
tic
Inn=imresize(I,0.85,'nearest');
Ibil=imresize(I,0.85,'bilinear');
Ibic=imresize(I,0.85,'bicubic');
figure(2);
subplot(2,4,1); imshow(I); title('origin');
subplot(2,4,2); imshow(Inn); title('NN 0.85 times');
subplot(2,4,3); imshow(Ibil); title('bilinear 0.85 times');
subplot(2,4,4); imshow(Ibic); title('bicubic 0.85 times');

Inn2=imresize(I,1.24,'nearest');
Ibil2=imresize(I,1.24,'bilinear');
Ibic2=imresize(I,1.24,'bicubic');

%subplot(2,4,1); imshow(I); title('origin');
subplot(2,4,6); imshow(Inn2); title('NN 1.24 times');
subplot(2,4,7); imshow(Ibil2); title('bilinear 1.24 times');
subplot(2,4,8); imshow(Ibic2); title('bicubic 1.24 times');
toc
경과 시간은 0.676345초입니다.
```

Tic toc으로 시간을 측정했을 시 **0.676345초**가 걸렸다.

4) Analyze the results in terms of interpolation methods and image quality.

내가 만든 interpolation 함수와 matlab의 내장함수인 interpolation 함수를 비교하여 보았을 때, 각각의 original 사진에 대해 조금씩 quality가 떨어진 것을 확인할 수 있었다. 내가 만든 interpolation 함수에 사용된 Original data가 조금 깨진 것을 사용하여 matlab 내장함수와는 performance가 떨어져 보인다. 하지만, 각각의 Original data와 비교하여 보았을 때는 많은 차이가 없었다. 그리고 matlab 내장 함수끼리 비교하여 보았을 때는 $NN \ll \text{Bilinear} < \text{Bicubic}$ 순으로 화질이 좋아보였다. NN은 가까운 곳으로 밝기를 정한다면, Bilinear은 가중치까지 고려하였다. Bicubic은 더욱 복잡한 식을 사용하였기 때문에 점차 부드러워지는 것을 볼 수 있다. 하지만 Bicubic은 많이 복잡하여 잘 사용하지 않는다고 한다.

Lab에 시행한 코드마다 tic toc으로 시간을 측정해보았을 때,
내가 만든 interpolation은

경과 시간은 0.260304초입니다. (NN / SCALE : 0.85)

경과 시간은 0.230420초입니다. (NN / SCALE : 1.24)

경과 시간은 0.207138초입니다. (BILINEAR / SCALE : 0.85)

경과 시간은 0.287819초입니다. (BILINEAR / SCALE : 1.24)

인 것을 고려해보면, 전체 시간은 0.9856784 초가 걸렸다.

MATLAB built-in function을 사용했을 시 전체 시간은 0.676345초가 걸렸다.

경과 시간은 0.676345초입니다. (NN / SCALE : 0.85) + (NN / SCALE : 1.24) +
(bilinear / SCALE : 0.85) + (bilinear / SCALE : 1.24) + (bicubic / SCALE : 0.85) +
(bicubic / SCALE : 1.24)

확실히 matlab의 built-in function의 수행시간이 짧았다.