

SRCNN(Super-Resolution Convolutional Neural Network)

Dong, Chao, et al. "Image super-resolution using deep convolutional networks." IEEE transactions on pattern analysis and machine intelligence 38.2 (2016): 295-307.

ISL

안재원

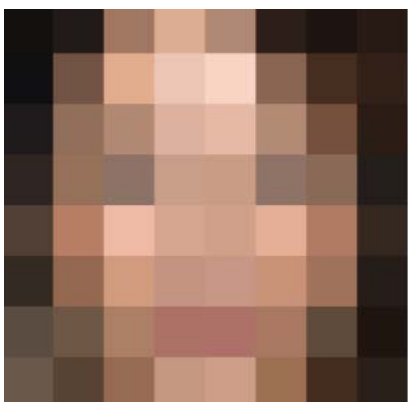
CONTENTS

- Intro
- Sparse-Coding-Based Methods
- Convolutional Neural Networks for Super-Resolution
- Experiments

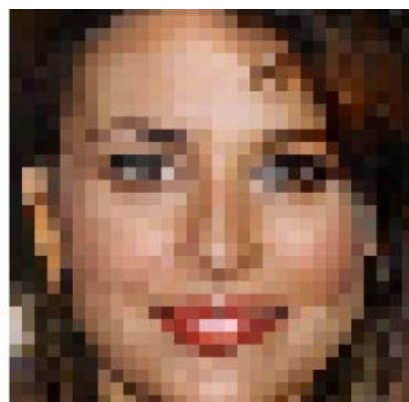
Intro

SR, Super-Resolution

- SR, Super-Resolution : 저 해상도의 영상을 고 해상도의 영상으로 복원하는 기술.
- 조건이 부족한 ill-posed한 문제의 한 종류로 해답이 유일하지 않은 문제



저 해상도 영상

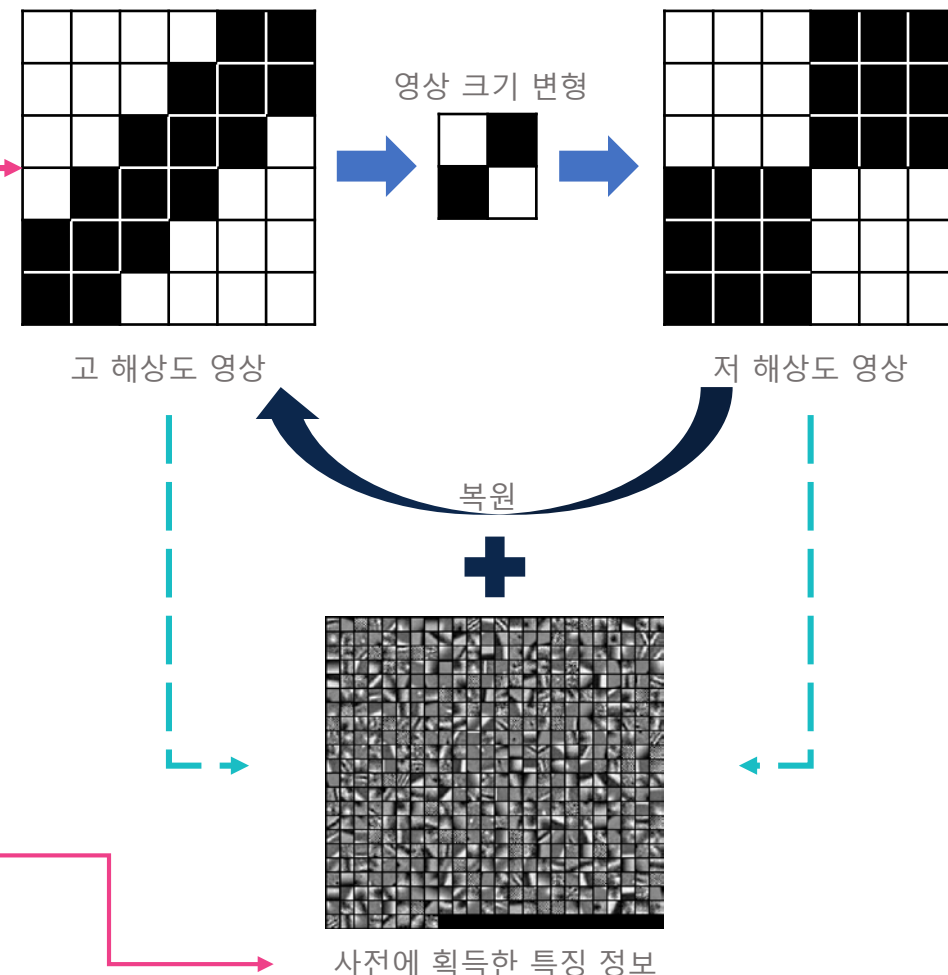


고 해상도 영상

- Example기반의 접근 방법이 State-of-the-art로 알려져 있다.
- Example을 어떻게 해석하고 활용하는지에 따라 다양한 방법이 있다.(SC, ANR, A+ etc..)

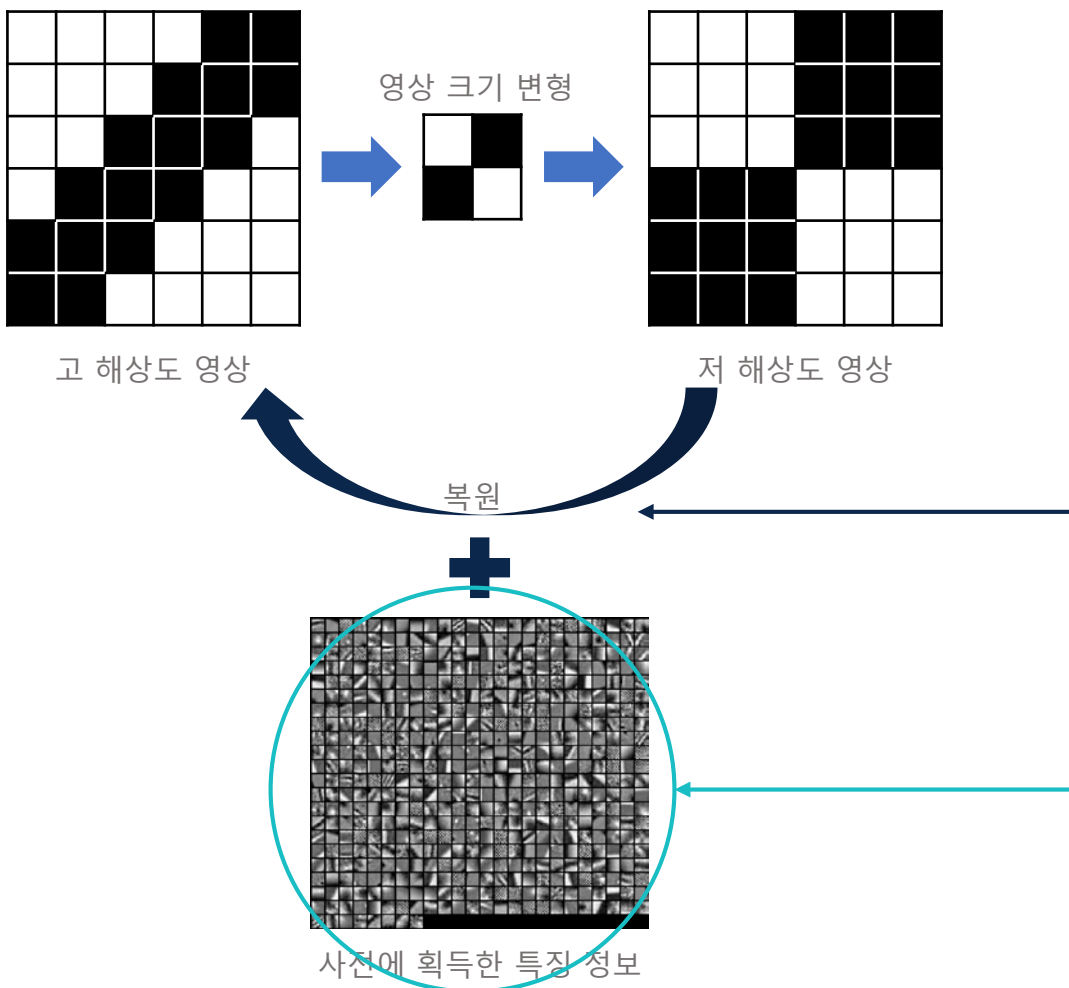
<기존 SR 기법의 한계>

- 전처리, 후처리 등의 추가적인 작업이 필요하며, 최적화의 한계가 있다.
- 흑백 위주의 단일 채널에 대한 연구가 진행 되어 왔으며, 다 채널을 동시에 처리하는 연구 사례가 많지 않다.
- 많은 연산 시간을 필요로 한다.



1. 사전에 획득한 특징 정보가 없을 때의 성능 문제.
2. 수작업(hand-made)으로 생성된 특징 정보의 한계
3. 특징 정보의 깊이

- Example기반의 Super-Resolution 방법 중 Sparse-coding 기반의 방법의 경우 진행 과정을 Convolution연산으로 해석이 가능하다.



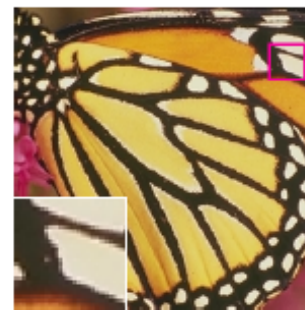
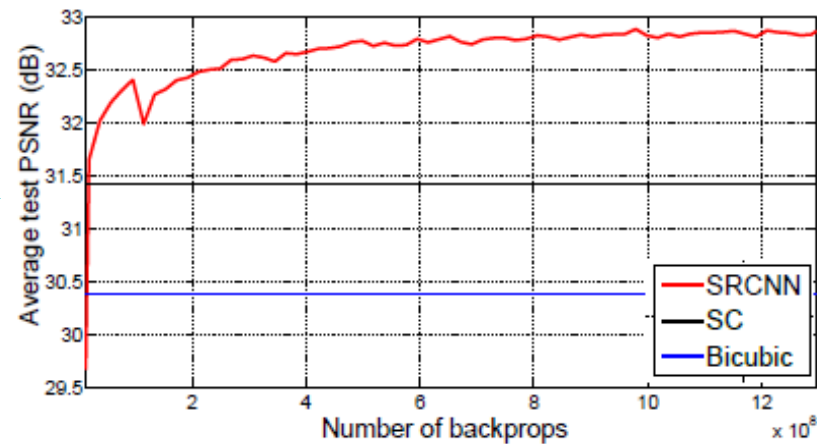
- 사전에 연구되어진 내용을 통해 CNN이 특징 검출 기능이 우수함을 알고 있다.

Intro

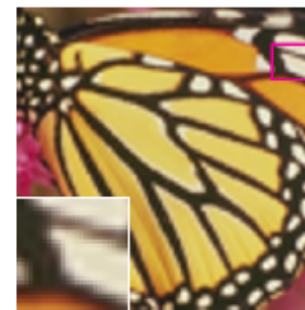
SRCNN의 장점

- 3개의 Convolution layer를 사용하지만, 기존 방법 대비 높은 성능을 보인다.
- 3개의 Convolution layer만을 사용하기 때문에 CPU에서 On-line으로 동작하더라도, 기존 방법 보다 더 빠르다.
- layer수를 늘리거나, loss function을 변경해 사용 할 수 있다. (확장성)

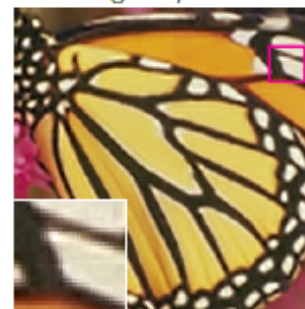
1. Feed-forward로 연산이 진행된다.(기존 방법은 반복 연산 과정이 포함됨)
2. 전처리, 후처리, 최적화 과정이 필요 없다.



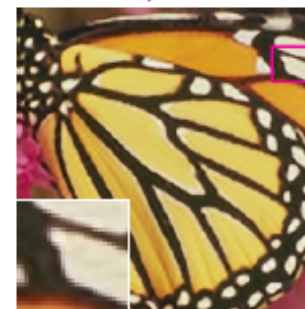
Original / PSNR



Bicubic / 24.04 dB



SC / 25.58 dB

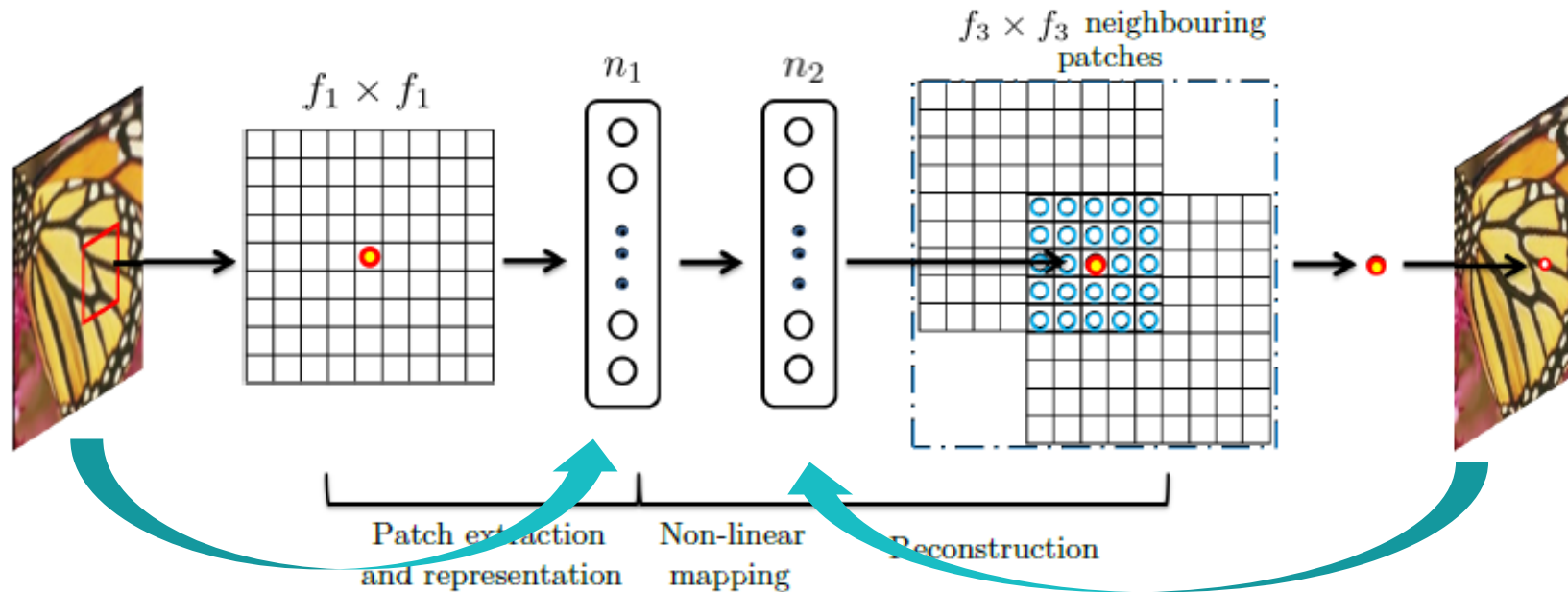


SRCNN / 27.95 dB

Sparse-Coding-Based Methods

Sparse-Coding-Based Methods

Sparse-Coding-Based Methods를 SRCNN관점으로 풀어 그린 그림

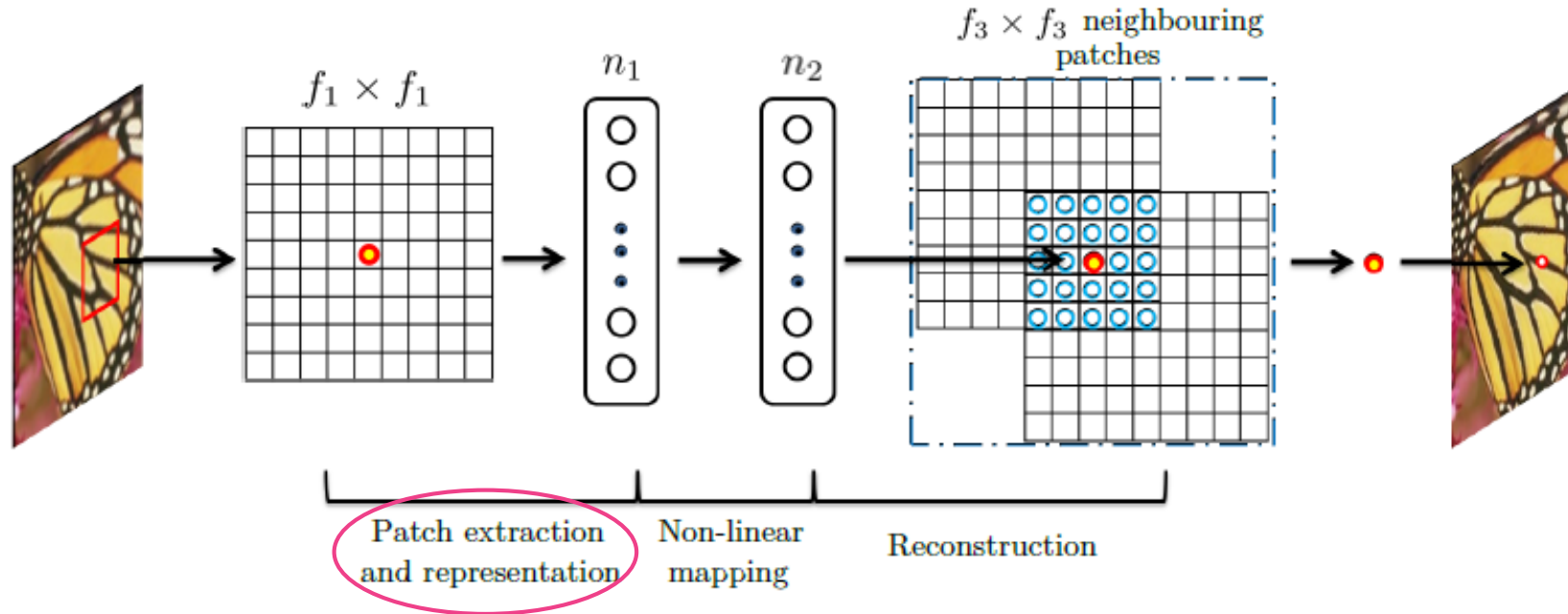


- 크게 3단계로 구성되어 있다.
- SR이 진행되는 흐름을 나타낸 것으로 사전에 진행되는 필터 생성 과정은 생략 되어 있다.

Sparse-Coding-Based Methods

Patch extraction and representation

Sparse-Coding-Based Methods를 SRCNN관점으로 풀어 그린 그림

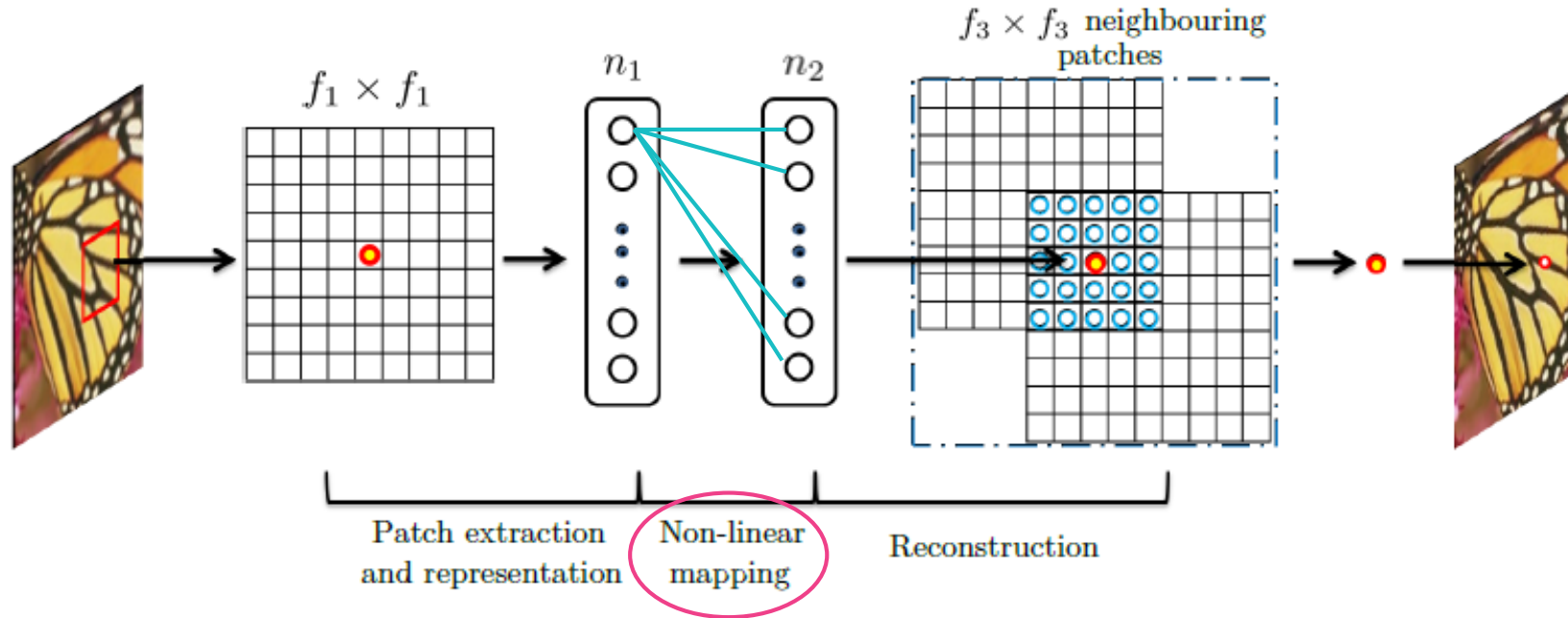


- n_1 개의 필터를 이용해 입력 영상(Low-Resolution)의 특징을 검출한다.
- 필터에 대하여 컨볼루션(convolution)연산을 사용한다.

Sparse-Coding-Based Methods

Non-linear mapping

Sparse-Coding-Based Methods를 SRCNN관점으로 풀어 그린 그림



- Low-resolution 영상에서 n_1 개의 필터를 이용해 검출한 결과와 High-resolution에서 검출되는 n_2 개의 특징 간의 상관 관계를 확인한다.
- Coding-solver를 이용해 상관 관계를 확인하며, Coding-solver의 동작에 따라 다양한 기법들이 있다.
- 반복적으로 동작하기 때문에 많은 연산 시간을 소비한다.

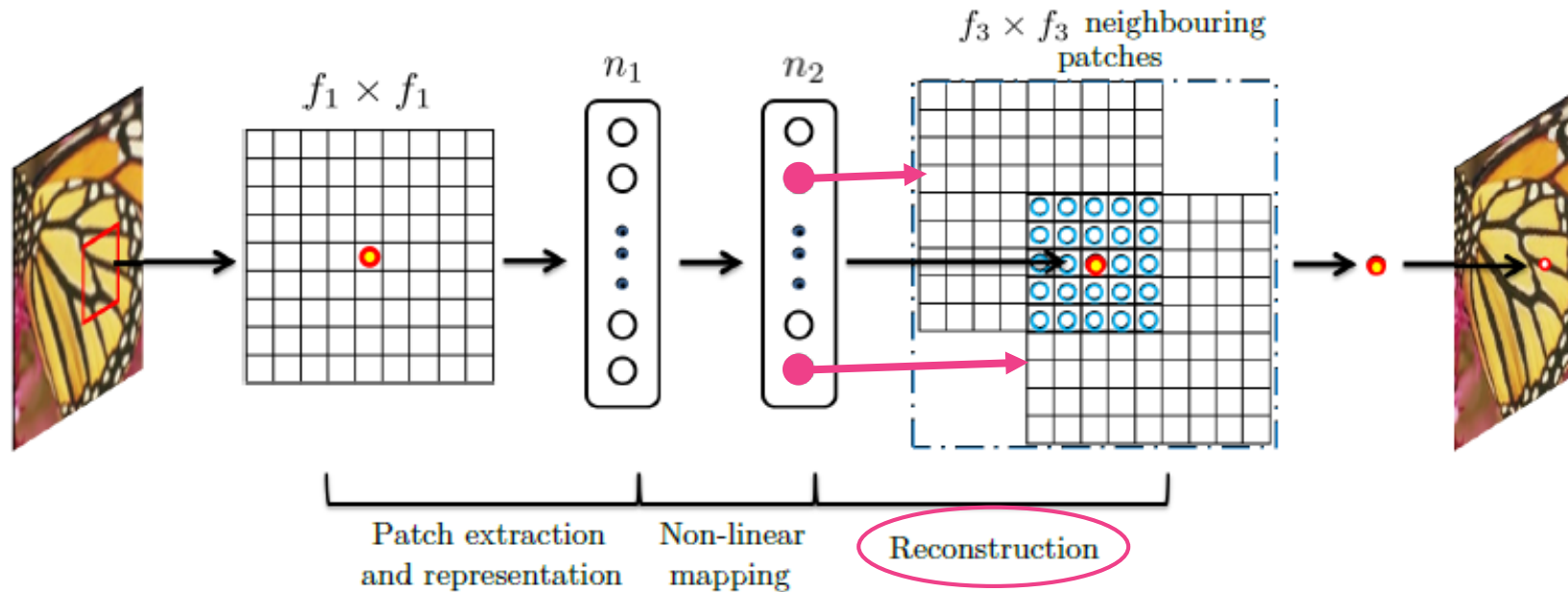


SRCNN에서는 Feed-forward로 연산이 진행되기 때문에 매우 빠르게 처리할 수 있다.

Sparse-Coding-Based Methods

Reconstruction

Sparse-Coding-Based Methods를 SRCNN관점으로 풀어 그린 그림

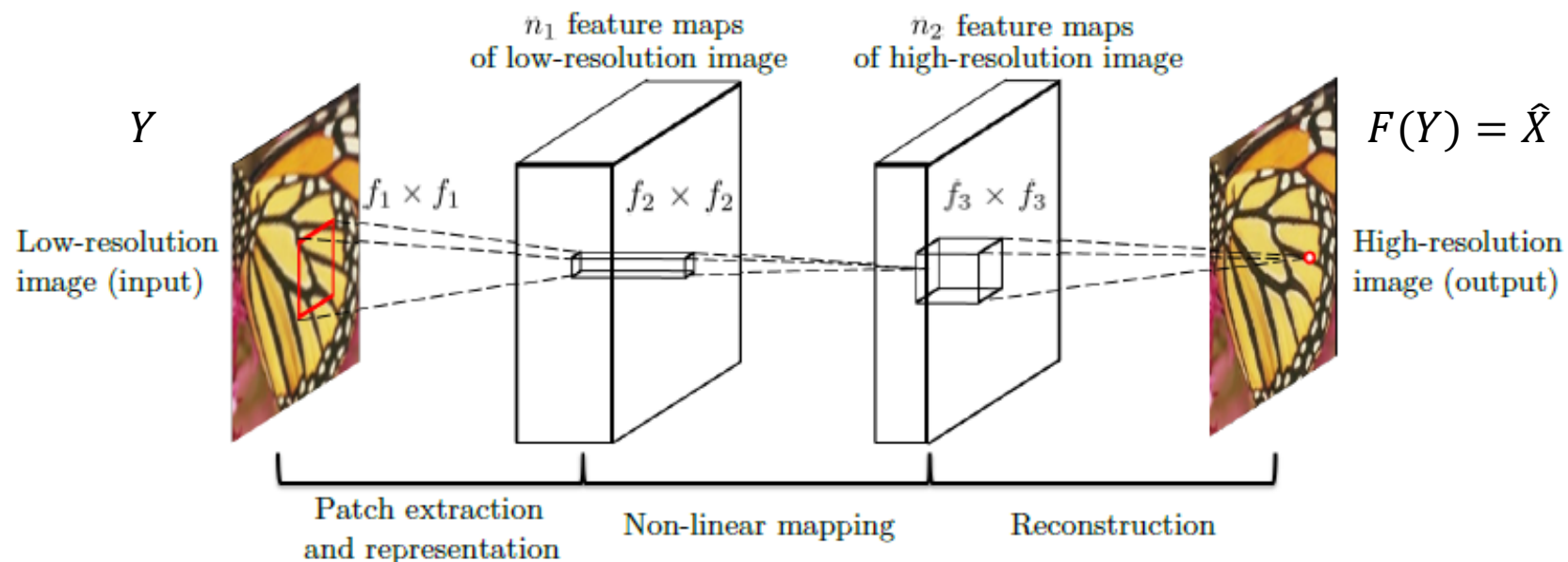


- 각 픽셀 위치에 대하여 활성화 되는 패치가 겹쳐지는 영역의 평균을 통해 High-resolution의 픽셀 값을 구한다.
- 다른 과정과 달리 Linear하다.

Convolutional Neural Networks for Super-Resolution

Convolutional Neural Networks for Super-Resolution

SRCNN의 네트워크 구조

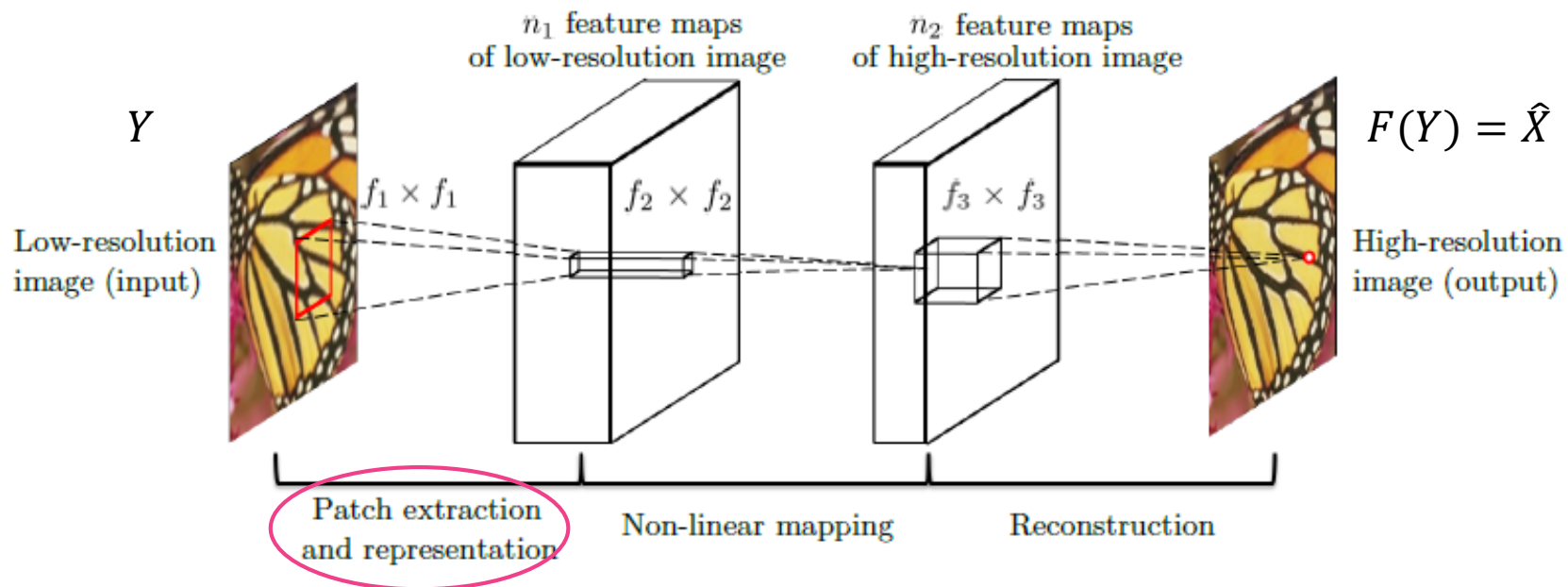


- 3개의 Convolutional layer로 구성되어 있다.
- 입력 영상 Y 를 이용해 SR과정을 거친(F) 결과 $F(Y)$ 를 구한다.
- 특징 검출, 변환, 최적화의 모든 작업을 Network를 이용해 구현한 end-to-end 형식으로 구성되어 있다.

Convolutional Neural Networks for Super-Resolution

Patch extraction and representation

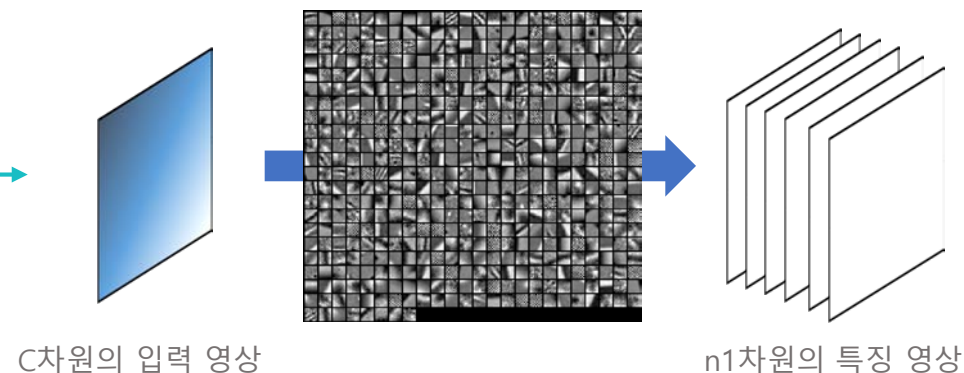
SRCNN의 네트워크 구조



- 패치 검출 과정으로 PCA, DCT 등의 사전에 학습된 네트워크를 사용하기도 한다.
- n_1 개의 패치를 검출한다.
- 다음의 수식으로 정의된다.

$$F_1(Y) = \max(0, W_1 * Y + B_1)$$

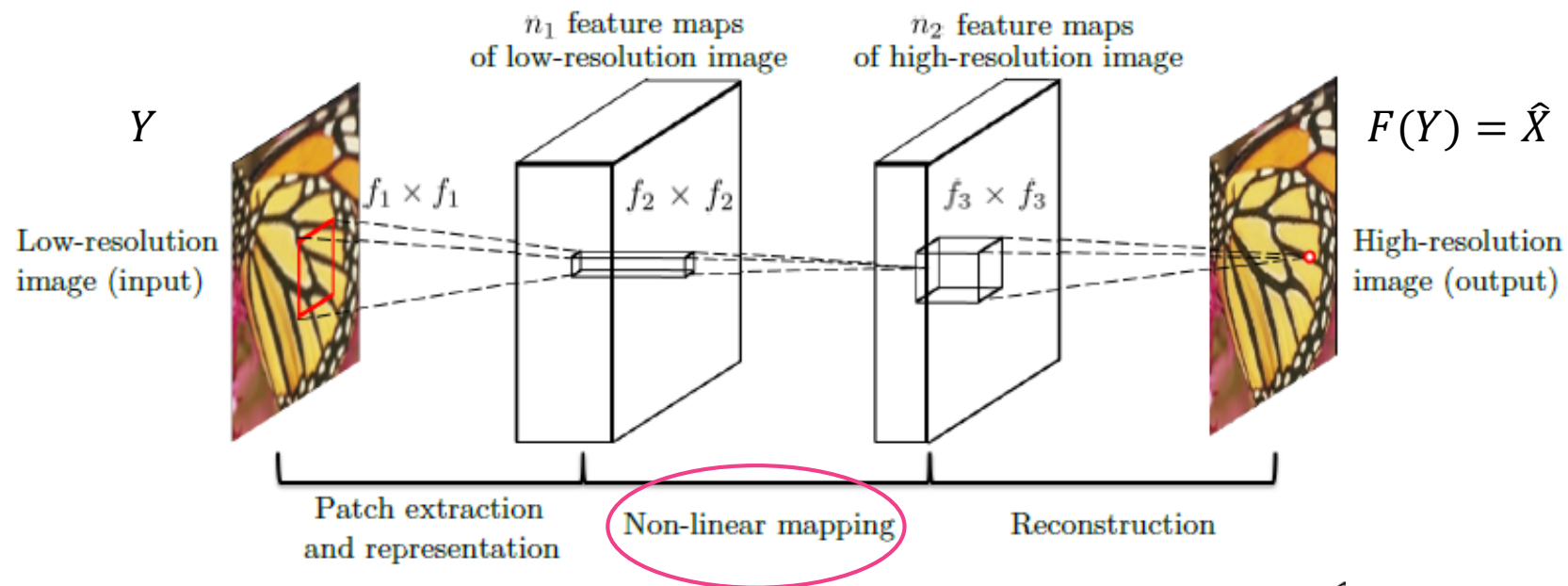
↑ ReLU



Convolutional Neural Networks for Super-Resolution

Non-linear mapping

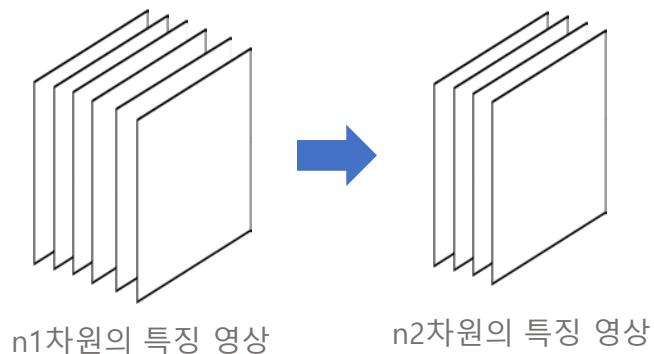
SRCNN의 네트워크 구조



- 첫 번째 Layer에서 검출된 특징들의 관계를 이용해 High-Resolution 결과를 생성하기 위한 정보를 생성한다.
- 필요에 따라 Layer 수를 늘릴 수 있으나, 1층으로 충분하다.
- f_2 의 크기는 보통 1을 사용한다.(1x1 convolution)처럼 동작한다.
- n_2 의 크기는 보통 n_1 보다 작은 값을 사용한다. ↪ 검출할 수 있는 정보가 적을 것이라 판단되기 때문에
- 다음의 수식으로 정의된다.

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2)$$

↪ ReLU



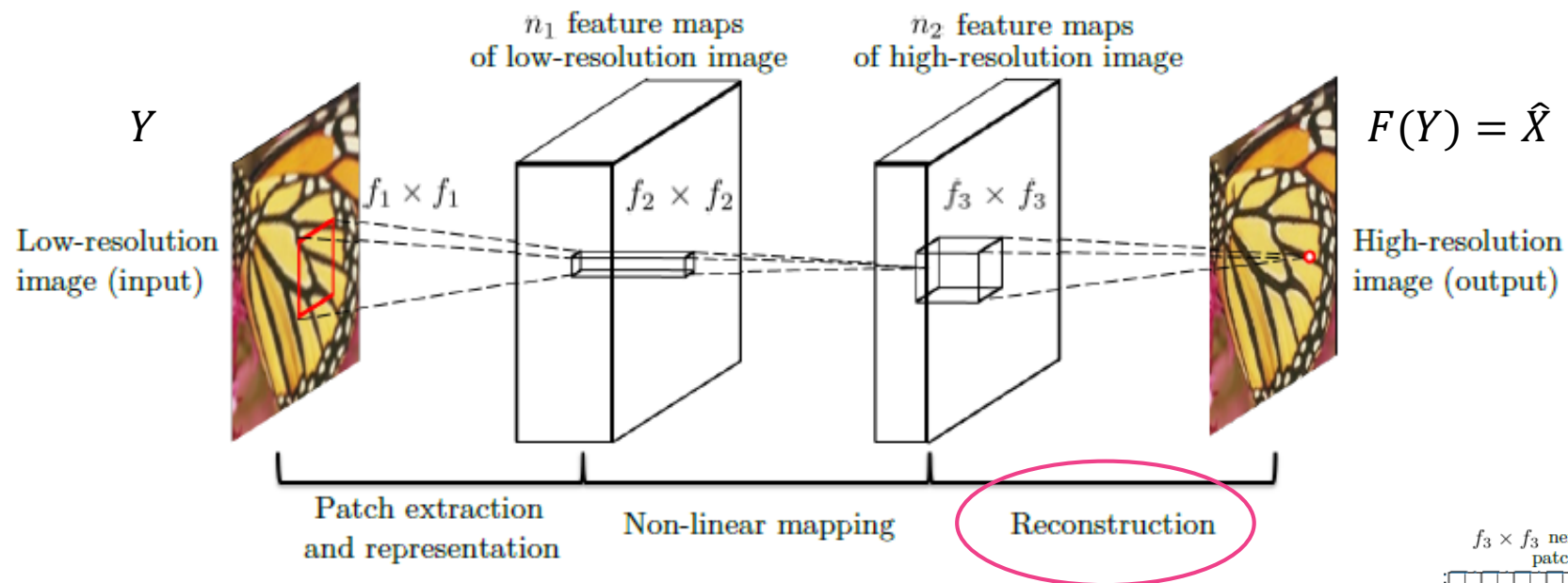
n_1 차원의 특징 영상

n_2 차원의 특징 영상

Convolutional Neural Networks for Super-Resolution

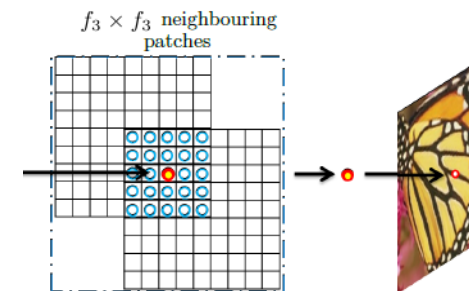
Reconstruction

SRCNN의 네트워크 구조



- 앞서 진행된 두 단계와 달리 Linear한 연산이 수행된다.
- n_2 차원의 특징 맵이 검출된 시점에서 충분히 Image domain의 정보가 반영될 수 있기 때문에 단순 평균을 취하는 동작을 하길 원한다.
- 다음의 수식으로 정의된다.

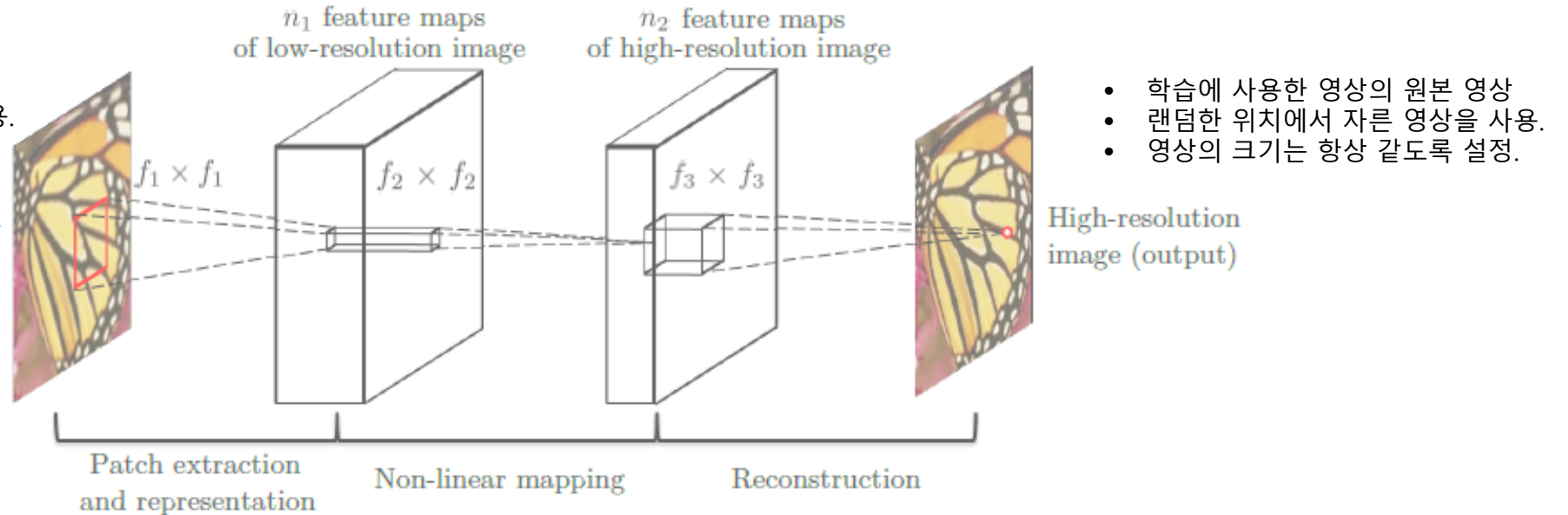
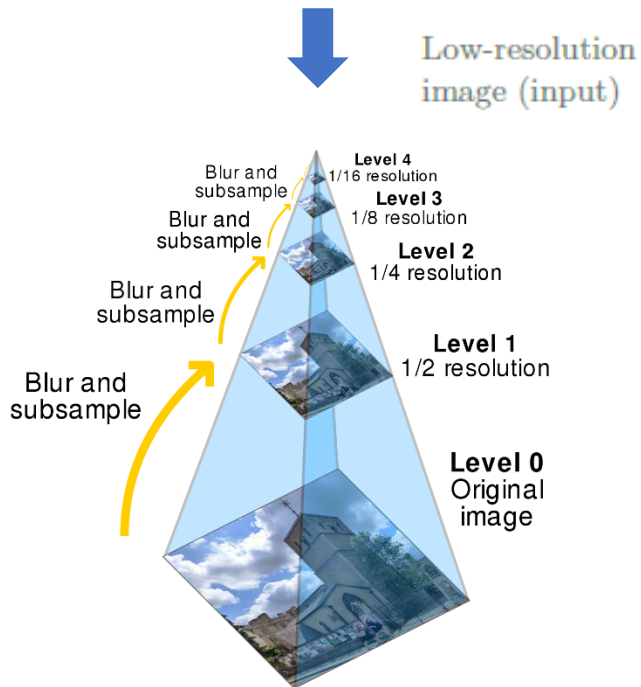
$$F(Y) = W_3 * F_2(Y) + B_3$$



Experiments

Training

- 학습 영상의 잘린 영상을 사용
- Upscaling factor에 맞춰 블러링된 영상 사용.
- 블러링은 가우시안 커널 사용.



- 학습에 사용한 영상의 원본 영상
- 랜덤한 위치에서 자른 영상을 사용.
- 영상의 크기는 항상 같도록 설정.

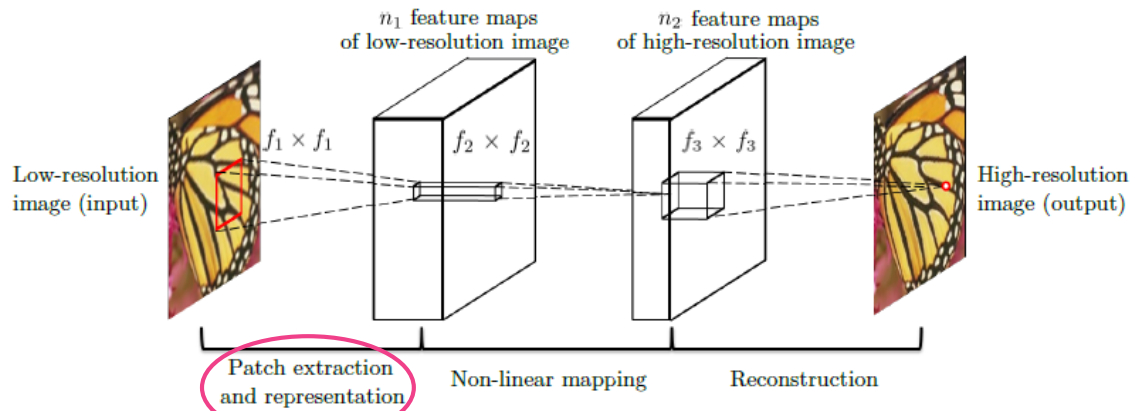
- 단계에 따라 다른 Training rate 적용.
- Border effects를 피하기 위해 Padding은 사용하지 않음.
- Padding을 하지 않기 때문에 정답보다 작은 영상이 출력되지만 겹치는 영역에 맞춰 에러를 계산함.
- Loss는 MSE(Mean Squared Error)를 사용함.

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^n \|F(\mathbf{Y}_i; \Theta) - \mathbf{X}_i\|^2$$

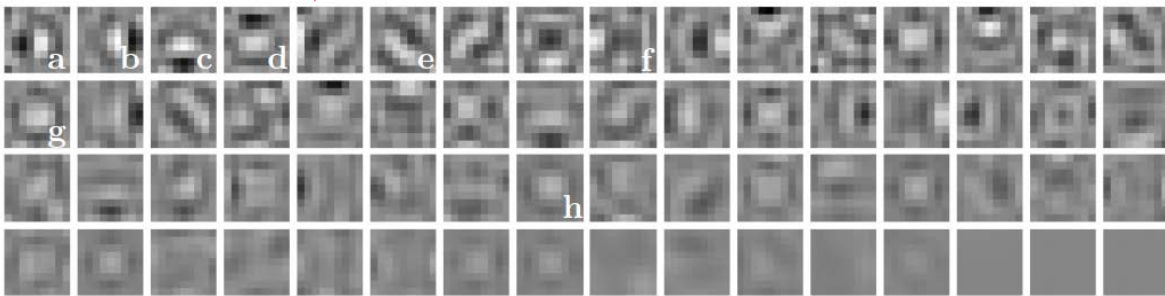
↳ MSE를 Loss로 사용하면, PSNR의 관점에서 좋은 결과가 얻어지도록 학습되지만, 다른 성능 지표도 높은 결과를 보인다.

Experiments

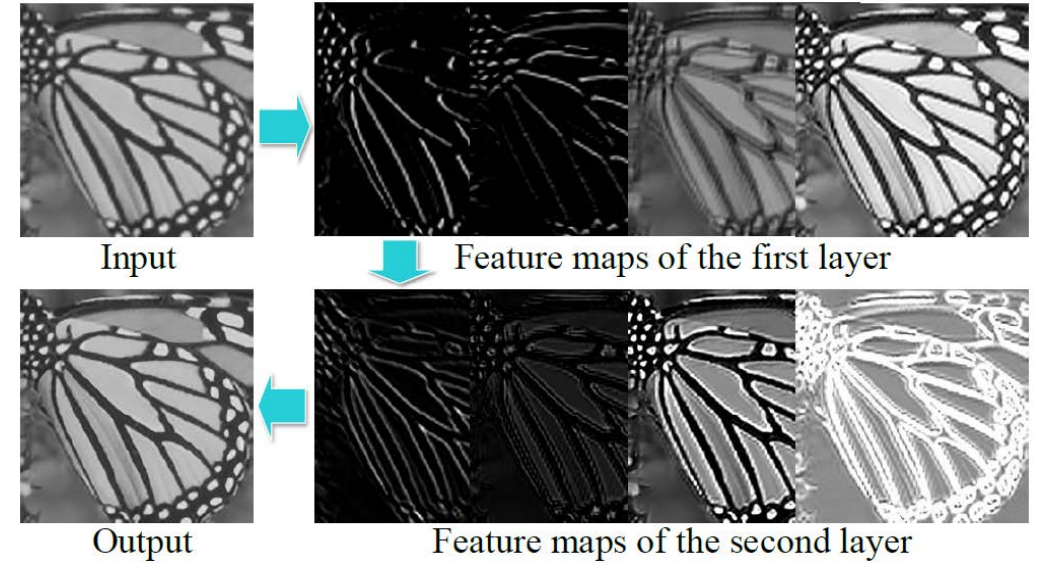
Training



- 특징 검출에 사용되는 Convolution mask



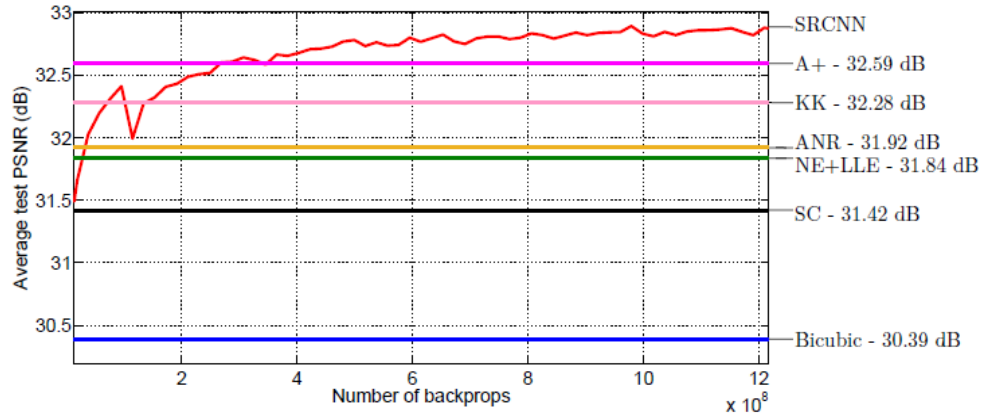
각 layer에서 출력되는 영상



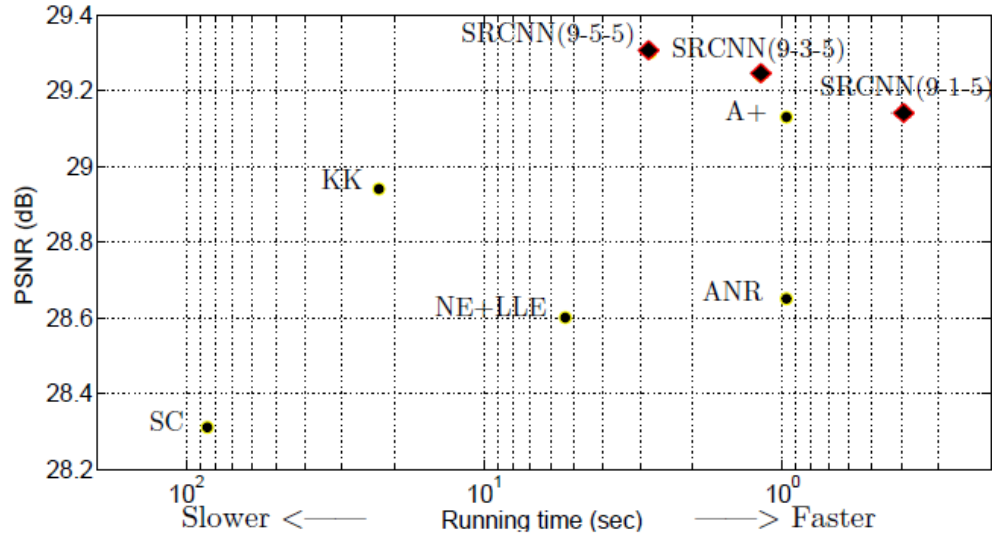
Experiments

Training result

- 각 기법 별 PSNR 결과



- 각 기법 별 연산 시간과 PSNR 비교



- 각 기법 별 성능 지표 결과

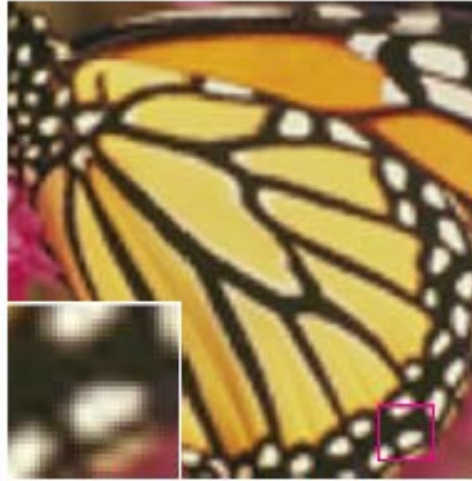
Eval. Mat	Scale	Bicubic	SC [50]	NE+LLE [4]	KK [25]	ANR [41]	A+ [41]	SRCNN
PSNR	2	33.66	-	35.77	36.20	35.83	36.54	36.66
	3	30.39	31.42	31.84	32.28	31.92	32.59	32.75
	4	28.42	-	29.61	30.03	29.69	30.28	30.49
SSIM	2	0.9299	-	0.9490	0.9511	0.9499	0.9544	0.9542
	3	0.8682	0.8821	0.8956	0.9033	0.8968	0.9088	0.9090
	4	0.8104	-	0.8402	0.8541	0.8419	0.8603	0.8628
IFC	2	6.10	-	7.84	6.87	8.09	8.48	8.05
	3	3.52	3.16	4.40	4.14	4.52	4.84	4.58
	4	2.35	-	2.94	2.81	3.02	3.26	3.01
NQM	2	36.73	-	42.90	39.49	43.28	44.58	41.13
	3	27.54	27.29	32.77	32.10	33.10	34.48	33.21
	4	21.42	-	25.56	24.99	25.72	26.97	25.96
WPSNR	2	50.06	-	58.45	57.15	58.61	60.06	59.49
	3	41.65	43.64	45.81	46.22	46.02	47.17	47.10
	4	37.21	-	39.85	40.40	40.01	41.03	41.13
MSSSIM	2	0.9915	-	0.9953	0.9953	0.9954	0.9960	0.9959
	3	0.9754	0.9797	0.9841	0.9853	0.9844	0.9867	0.9866
	4	0.9516	-	0.9666	0.9695	0.9672	0.9720	0.9725

Experiments

SR Result#1



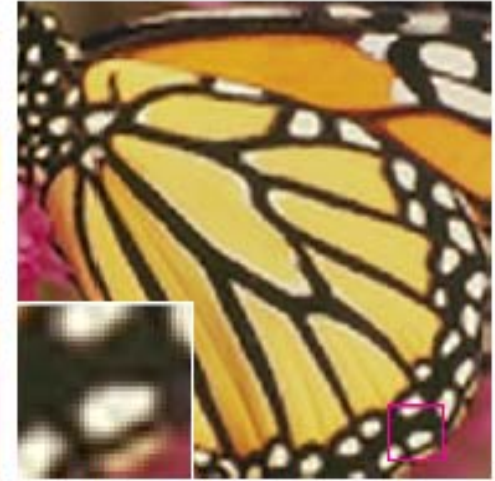
Original / PSNR



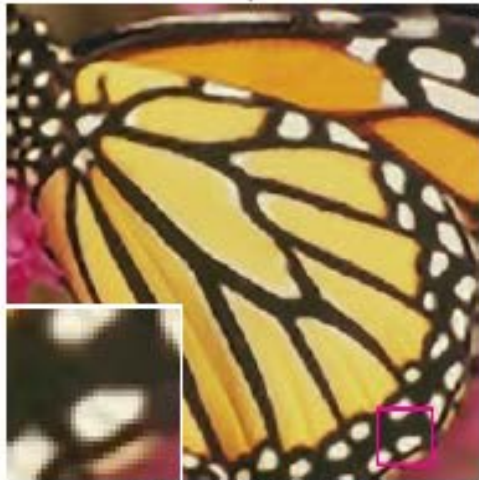
Bicubic / 24.04 dB



SC / 25.58 dB



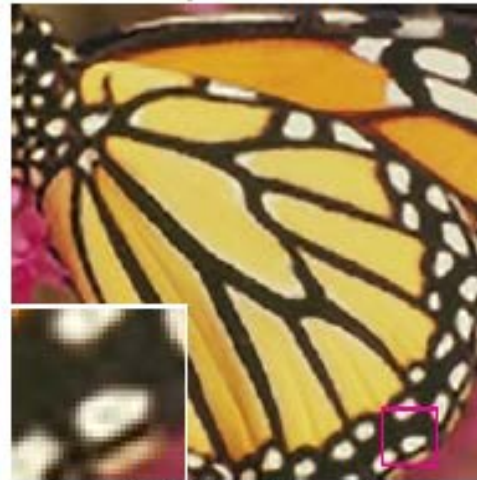
NE+LLE / 25.75 dB



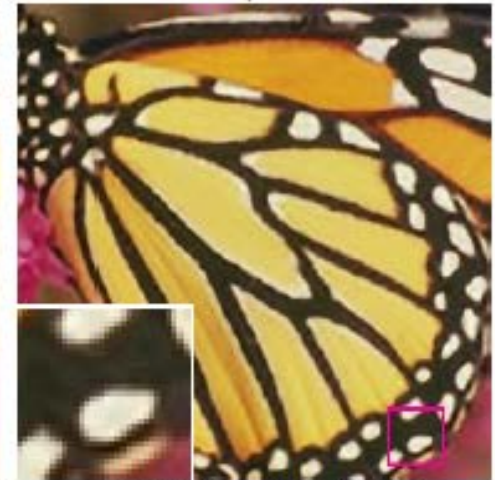
KK / 27.31 dB



ANR / 25.90 dB



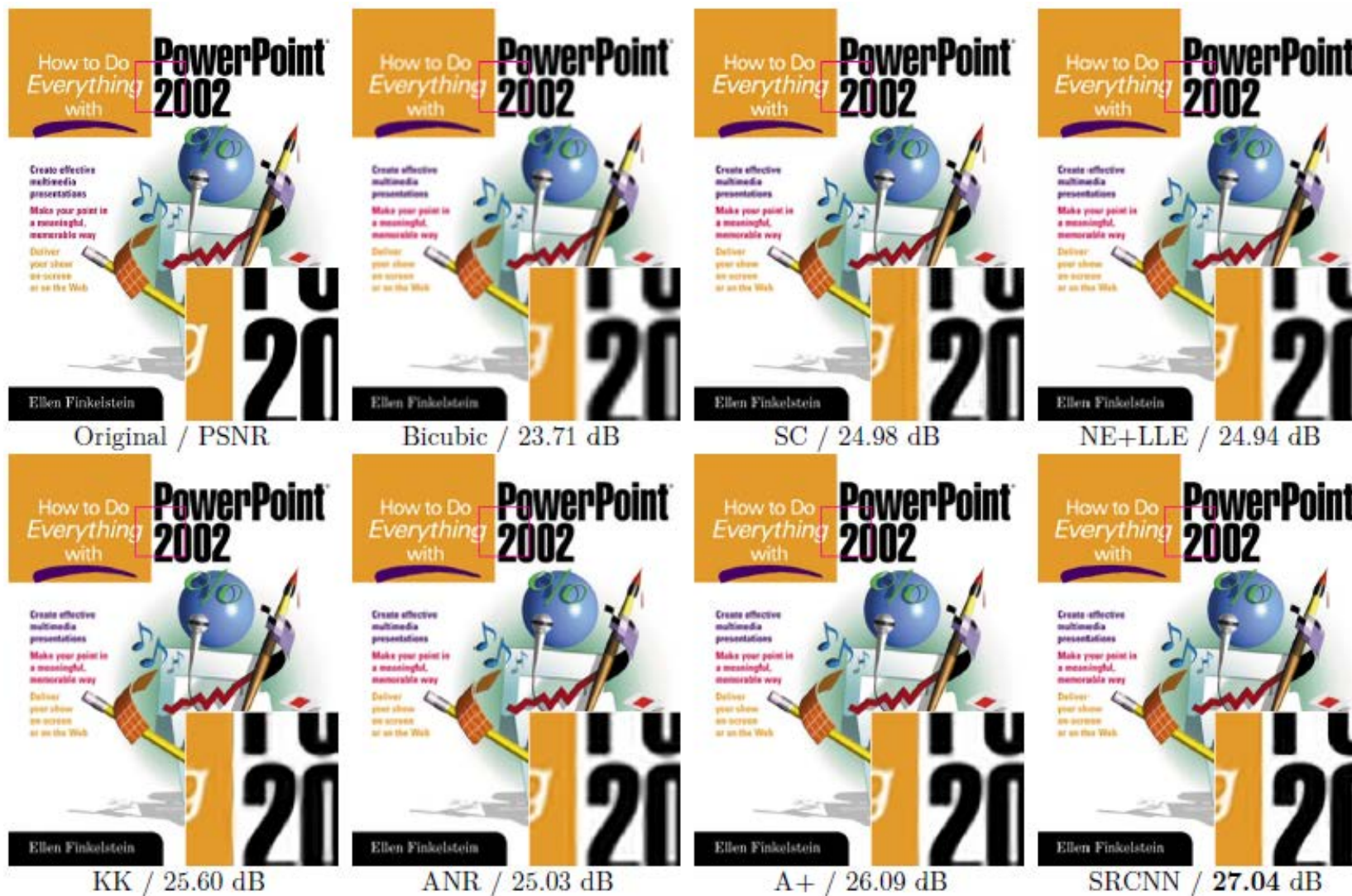
A+ / 27.24 dB



SRCNN / 27.95 dB

Experiments

SR Result#2



Experiments

SR Result#3



Original / PSNR



Bicubic / 26.63 dB



SC / 27.95 dB



NE+LLE / 28.31 dB



KK / 28.85 dB



ANR / 28.43 dB



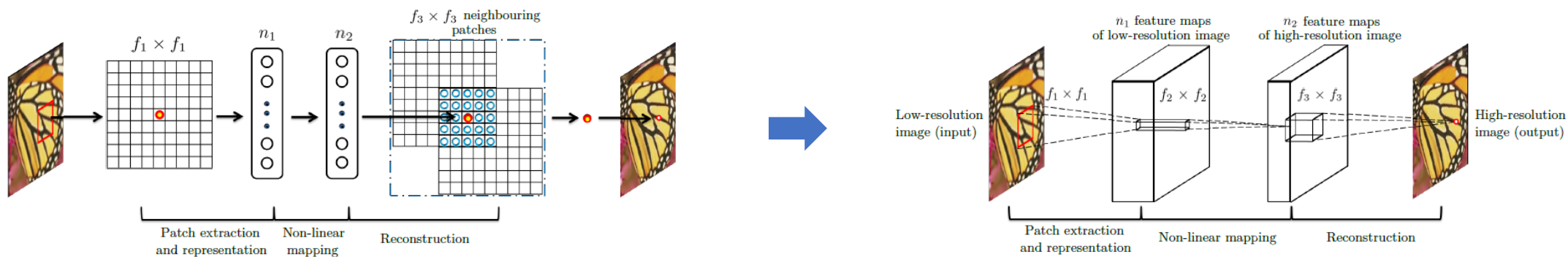
A+ / 28.98 dB



SRCNN / 29.29 dB

Conclusion

Conclusion



- 기존의 Convolution한 특성을 갖는 Sparse-coding 기반의 SR(Super-resolution)을 CNN을 이용해 구현하고, 그 결과를 확인 하였다.
- 전처리, 후처리, 최적화 기능이 모두 포함된 end-to-end 방식의 Network 구조를 구성하고, 그 결과를 확인 하였다.
- 매우 간단한 구조(3층)를 이용해 State-of-the-art의 성능을 보이는 것을 확인 하였다.
- 더 깊은 네트워크 구성으로 성능 향상을 기대해 볼 수 있다.

Q & A

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \end{aligned}$$

- 손실이 적을 수록 높은 값을 갖는다.
- MSE 가 작아진다 -> 분모가 작아진다 -> log안의 숫자가 커진다 ->

