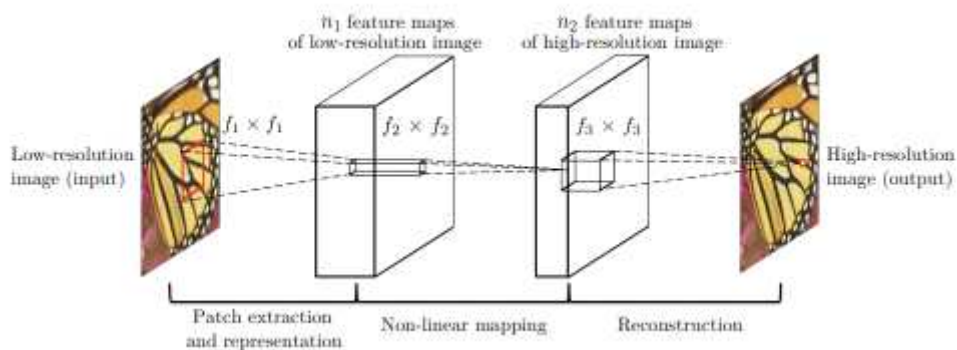


논문리뷰 - SRCNN

Image Super-Resolution Using Deep Convolutional Networks

이 논문은 2015년에 발표된 논문으로 SISR에 최초로 딥러닝을 적용한 논문이다. SR 문제에 많은 사람들이 딥러닝을 적용한 계기가 되기도 하였다. 위 논문의 방법은 저해상도, 고해상도 이미지 간 매핑을 직접 학습한다. 이때 매핑이란, 저해상도 이미지를 입력으로 받아 고해상도 이미지를 출력하는 Deep CNN으로 표현된다.



먼저 위를 자세히 알아보기 전에 위 논문에서의 용어 정리를 살짝 하고 가자면,
 Y = interpolated image(low resolution image)이며, X 와 size가 같아야한다.

X = high resolution image

우리의 목표는 Y 를 mapping한 F 안에 넣어 $F(Y)$ 가 X 와 비슷해지는 것이다.

1. Formulation

저해상도 이미지가 있을 때 bicubic interpolation으로 원하는 크기까지 upscale을 진행한다. 사진의 과정을 통해 interpolated image를 mapping하는 F 를 학습하고자 한다.

1-1. Patch extraction and representation

Y 에서 여러 patch들을 추출하고, 각 patch들을 vector로 나타낸다. 나타낸 vector는 feature map이다. 첫 번째 layer는 F_1 으로 표현한다.

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

W_1 : filter (size : $c \times f_1 \times f_1 \times n_1$)

B :biases

c : number of channels in the input image

f_1 : the spatial size of a filter

n ; the number of filters

이후 ReLU를 적용한다.

1-2. Non-linear mapping

한 vector 내에 다른 vector로 nonlinearly mapping을 한다. 각 mapped vector는

high-resolution patch를 표현한다,

첫 번째 layer는 n_1 차원 feature를 뽑는다. 이 과정에서 n_1 차원 벡터들이 n_2 차원 벡터로 mapping 한다.

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

Non-linearity를 높이고 싶으면 conv layer를 더 쌓으면 되지만 compleity가 높아지게 되고 결국 학습 시간이 늘어난다.

1-3. Reconstruction

앞서 만들어진 patch들을 모아 high-resolution image를 만든다. 이때 만들어진 이미지는 X 와 비슷할 것이다.

기존의 전통적인 방법들은 최종 이미지를 만들기 위해 patch들을 평균을 냈는데 평균 내는 행위는 벡터를 flatten이라고 하는거라서 마지막 layer는 다음과 같이 만들어졌다.

$$F(\mathbf{Y}) = W_3 * F_2(\mathbf{Y}) + B_3$$

세가지 방법 모두 conv layer 형태로 만들어졌다.

Relationship to Sparse-Coding-Based Methods

Sparse Coding Based SR 방법들은 CNN이라고 본다.

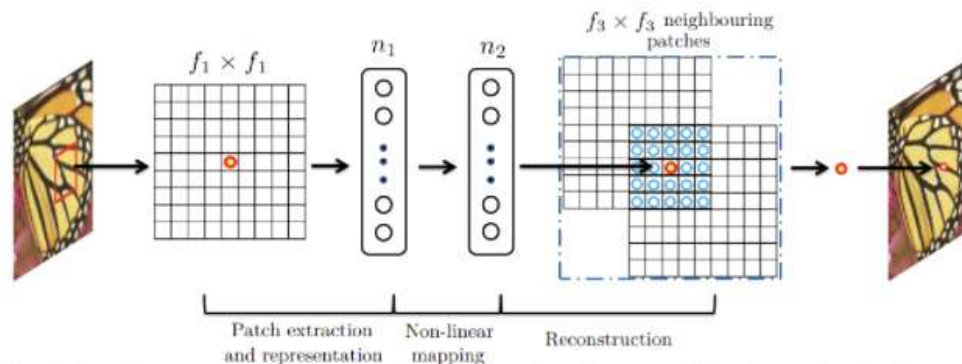


Fig. 3. An illustration of sparse-coding-based methods in the view of a convolutional neural network.

LOSS = MSE

Learning Rate

Loss	Learning Rate (layer)	Optimizer
MSE	10^{-4} (1 st , 2 nd), 10^{-5} (3 rd)	SGD

Optimizer = SGD

FSRCNN(Fast-SRCNN)

: SRCNN의 속도를 높이는 데 중점을 뒀다. 이전의 다른 learning-based 방법들과 비교했을 때 충분히 빠르지만 큰 이미지에서의 속도는 만족스럽지 않다.

그래서 네트워크구조를 살펴보다가 속도를 제한하는 두 가지 원인을 발견했다.

첫째, LR 이미지는 bicubic upsampling되어 네트워크의 입력으로 넣어진다.

즉 이미지 크기를 n 배 키웠다면 n^2 만큼의 계산량이 늘어난다.

따라서 input 이미지를 upsampling 하지 않고 그대로 사용한다면 그만큼 빨라질 것이다.

둘째, SRCNN에서 input patch들은 high-dimensional LR feature space & HR feature space에 투영된다.

파라미터수가 많을수록 속도는 줄어들이지만 그와 동시에 성능은 좋아진다.

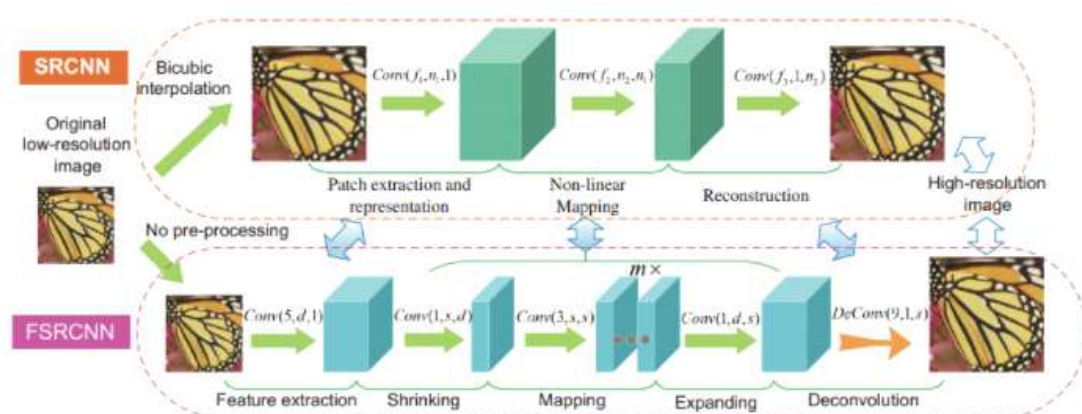
문제는 '기존의 성능을 유지하며 어떻게 네트워크의 크기를 줄이느냐'이다.

위에서 언급한 두 가지 문제를 다음과 같이 해결함으로써 FSRCNN을 설계했다.

첫 번째 문제를 해결하기 위해 bicubic interpolation을 deconvolution layer로 바꿨다.

두 번째 문제를 해결하기 위해 mapping layer의 시작과 끝에 shrinking & expanding layer를 추가했다.

출처: <https://sofar-sogood.tistory.com/entry/FSRCNN-리뷰-Accelerating-the-Super-Resolution-Convolutional-Neural-Network-ECCV-16> [작심삼일]



Reference

<https://d-tail.tistory.com/6>

<https://sofar-sogood.tistory.com/entry/SRCNN-Image-Super-Resolution-Using-Deep-Convolutional-Networks-ECCV-14>