

Multi scale Super-Resolution with fewer parameters

by using Asymmetric convolution and Share net

경희대학교 산업경영공학과 2015100915 김태호

<https://github.com/hwangtoemat>

요 약

Super-Resolution(이하 SR)은 저해상도(Low Resolution, 이하 LR) 이미지를 고해상도(High Resolution, 이하 HR) 이미지로 변환시키는 문제를 의미한다. Super-Resolution을 deep learning으로 해결하는 방법은 여러가지로 나눌 수 있는데, 각각의 방법에서 모델의 파라미터 수를 줄이는 것은 항상 중요한 문제이다. 따라서 기존 모델의 구조를 바꾸지 않고 Asymmetric convolution을 사용하여 기존 모델의 성능을 유지하면서 모델용량을 약 35%정도 줄였다. 또 각각 scale별 모델은 전부 학습시킬 필요없이 직접적으로 Upscale하는 부분을 제외한 나머지 부분은 Share하면서 파라미터의 수를 줄인다.

1. 서 론

Super-Resolution을 구현한 deep learning 모델은 2014년 SRCNN부터 2019년 DPSR, MSRN까지 매년 계속해서 발전된 모델들이 제시되어왔다. 점점 좋은 성능의 모델들이 제시되어 왔지만 그만큼 모델의 파라미터 개수 또한 늘어나며 연산량이 늘어나고 메모리 소비량이 커지며 모델은 점점 무거워져 갔다.[1] 따라서 발전된 모델이 있더라도 이를 모바일이나 일반 노트북과 같은 환경에서는 구현하기 힘들어졌으며 이러한 모델들을 실생활에 적용하기 위해 반드시 모델을 가볍게 압축시킬 필요가 있다.

모델을 압축시킬 때 가장 중요한 점은 기존 모델의 성능을 크게 감소시키지 않는 선에서 모델의

파라미터 개수, 연산 속도만 줄여야 한다는 것이다.

pre-upsampling의 방법으로 구현하는 모델의 경우 이미 upsampling을 한 후 연산을 하기 때문에 다른 방법에 비해 연산량이 많기 때문에 메모리의 부담이 크다. 따라서 이를 줄이기 위해 이 논문에서는 기존의 일반적인 convolution layer를 Asymmetric convolution으로 변경하여[2] 모델의 구조는 유지 하면서 파라미터의 개수를 획기적으로 줄여 기존 모델의 성능은 유지하면서 모델을 가볍게 만들어 주는 방법을 제안한다.

post-upsampling의 방법으로 구현하는 모델의 경우 일반적으로 한가지 모델은 한가지 scale에 대해서만 학습할 수 있기 때문에 다른 scale의 HR을 만들기 위해서는 처음부터 다시 학습해야 하는

단점이 있었고 이는 간단히 말해 3가지 scale의 HR변환이 필요하면 3가지 모델로 3배의 연산을 진행해야 하는 번거로움이 있는 것이다. 따라서 이 논문에서는 하나의 모델을 Extraction Block, Upscale Block, Refine Block 세 개로 나누어 Extraction Block과 Refine Block은 각각의 scale에 상관없이 share해주고 Upscale Block만 바꾸어 주는 방법을 제시한다. 이는 기존의 n개의 scale을 다루기 위해서 n배의 연산이 필요한 것과 비교하면 모델의 전체 용량을 크게 줄이는 효과가 있다.

2. 관련 연구

2.1 VDSR

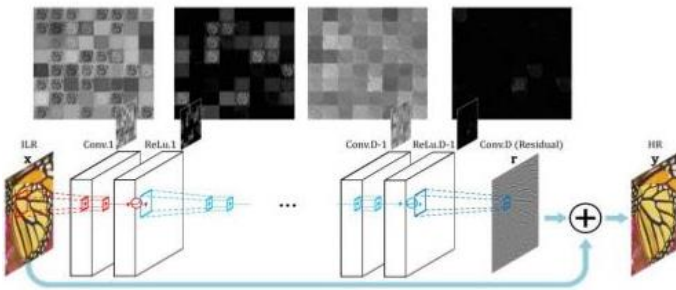


Fig 1. VDSR의 구조

VDSR[3]은 기존의 얇은 층을 가진 SR모델에 skip connection을 사용한 residual 기법의 도입으로 최초로 20층의 깊은 모델을 제시한 모델이다. VDSR은 input layer를 지난 값을 18층의 residual layer를 통과시킨 residual 값과 더해주어 SR을 만드는 방식이다. 깊은 층을 통해 기존의 모델보다 훨씬 좋은 성능을 냈다.

2.2 Asymmetric Convolution

Asymmetric convolution은 convolution 연산 시 kernel의 형태가 정사각형이 아닌 비대칭의 직사각형 형태를 갖는 convolution 연산을 의미한다. 이때 이를 이용하여 기존의 3x3연산을 3x1, 1x3으로 대체하면서 input된 값의 인식하는 구역의 넓이는 유지하면서 파라미터의 수를 크게 줄일 수 있다.[2]

Kernel Size	No. of Layers	No. of Filters	Image Patch Size	No. of Parameters
3 × 3	1	10	28 × 28	900
3 × 1 and 1 × 3	2	10	28 × 28	600
5 × 5	1	10	28 × 28	2500
5 × 1 and 1 × 5	2	10	28 × 28	1000
7 × 7	1	10	28 × 28	4900
7 × 1 and 1 × 7	2	10	28 × 28	1400
9 × 9	1	10	28 × 28	8100
9 × 1 and 1 × 9	2	10	28 × 28	1800
11 × 11	1	10	28 × 28	12,100
11 × 1 and 1 × 11	2	10	28 × 28	2200

Table 1. 일반적인 Convolution 과 Asymmetric convolution의 파라미터 개수 차이[2]

이를 통해 기존의 다른 모델에서 하지 못한 7x7, 9x9의 큰 구역에 대한 인식을 진행할 수도 있을 것이다.

2.3 MSRN

MSRN은 N개의 residual block에서 나온 output값을 전부 bottleneck layer에서 concat하여 1x1 크기의 커널로 병목된 특징들의 각 pixel위치의 의미를 얻어내어 upscale하는 SR모델이다.

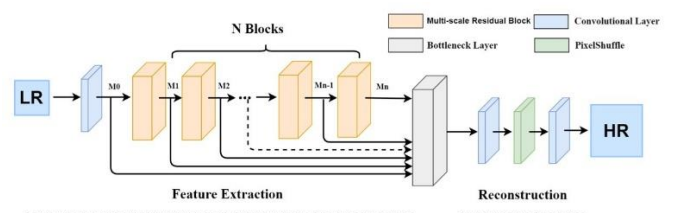


Fig 2. MSRN 구조

3. 제안 방법

3.1 VDSR에 Asymmetric convolution 적용

이 논문에서는 기존의 VDSR 내부의 residual layer에서 3x3 convolution 연산을 3x1, 1x3으로 나누어 모델의 파라미터 개수를 줄인 Asym_VDSR을 제안한다.

3.2 3개의 Block으로 나누어 multiscale 구현

이 논문에서는 하나의 모델을 'Extraction Block', 'Upscale Block', 'Refine Block'으로 나누어 'Extraction Block'과 'Refine Block'은 scale에 상관없이 share해주고 Upscale Block만 scale별로 다르게 학습하는 방법을 제안한다.

위의 아이디어로 설계한 모델인 MASRN은 'Extraction Block'에 여러 size의 kernel을 사용하여 여러 scale의 변환에서 공통적으로 사용할 수 있도록 하였고, bottleneck 구조[4]를 갖도록 하였다. 'Upscale block'에서는 PixelShuffle[9]를 사용하여 upsample하였다. 마지막으로 'Refine Block'에서는 여러 사이즈의 커널을 사용하는 residual block을 병렬의 형태로 사용하여 여러 scale에서 공통적으로 사용할 수 있도록 하였다.

4. 실험

4.1 Dataset

Dataset은 NITIRE 2017 SISR challenge에서

주어진 DIV2K를 사용했다. 이 dataset은 2K resolution의 영상으로 구성되어있는데, 학습용 800장, validation용 100장으로 이루어져 있다. Asym_VDSR을 학습할 때는 48x48사이즈로 200개로 randomcrop하여 사용하였고 MASRN을 학습할 때는 input할 LR값이 48x48사이즈가 되도록 각각의 scale마다 size가 다르게 HR을 Randomcrop하여 학습을 진행하였다.

Test시에는 set5 dataset을 사용하였다.

4.2 Asym_VDSR과 VDSR의 성능 비교

시간의 문제로 두 모델을 끝까지 학습시키지는 못했지만 같은 train data로 같은 Epoch를 학습시켰기때문에 두 모델간의 차이를 비교해 보겠다.

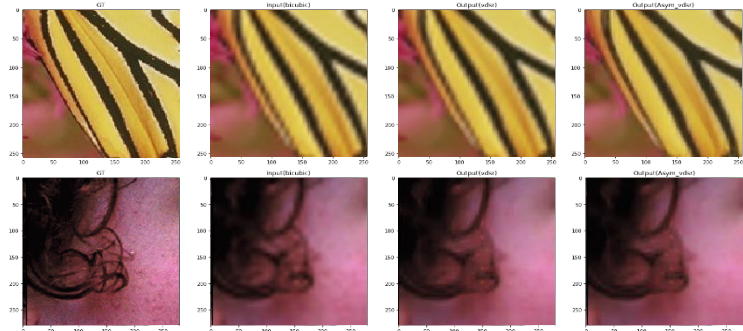


Fig 3. 왼쪽부터 GT, bicubic, VDSR, Asym_VDSR이다.

각 결과값들을 확대하여 보면 두 모델사이에 큰 차이는 느낄 수 없었다. 오히려 이미지 축소과정에서 생긴 계단현상은 Asym_VDSR에서 더 잘 해결하는 것처럼 보였다.

두 모델의 용량은 VDSR이 2.68Mb, Asym_VDSR이 1.78MB로 약 35% 정도

감소하였고 파라미터의 개수도 비슷하게 감소하였다.

Dataset (set5)	Model	Processing time(CPU)	Processing time(GPU)	PSNR
Head (scale_4)	VDSR	1.88854	0.06073	31.37733
	Asym_VDSR	1.74734	0.03661	31.24078
Butterfly (scale_4)	VDSR	1.68059	0.06537	22.95911
	Asym_VDSR	1.51420	0.04373	22.66058

Table 2. 붉은 글씨가 가장 좋은 값

두 모델간의 차이를 살펴보면 어떤 환경에서도 Asym_VDSR이 더 빠른 연산 속도를 보였고 특히 GPU에서는 평균 약 1.7배 빠른 것으로 나타났다. PSNR은 VDSR의 값이 더 높지만 눈으로 보았을 때 오히려 Asym_VDSR의 출력값이 경계선 부분을 더 부드럽게 만들어 준 것을 볼 수 있다. 이 실험에서 사용한 VDSR이 용량이 큰 모델이 아니기 때문에 체감상 용량이나 연산속도차이가 작아 보이지만 이 방법을 다른 무거운 모델에 적용시키면 확실한 퍼포먼스 차이를 느낄 수 있을 것이다.

4.3 MASRN 성능 실험

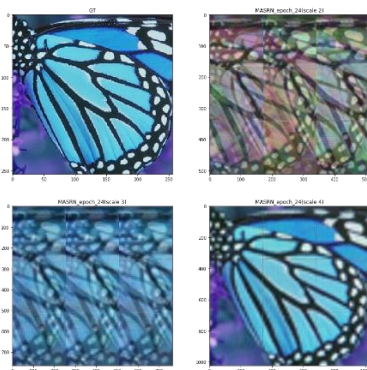


Fig 4. 왼쪽위부터 오른쪽으로 GT, Scale 2, Scale 3, Scale 4

위의 그림을 보면 알 수 있듯이 실험에서 좋은 성능을 내지 못하였다. 정확한 이유는 알 수 없지만 scale 2, 와 scale 3에서는 여러 장의 이미지가 겹쳐 있는 듯한 모습을 보이고 유일하게 scale 4에서만 형태를 유지하는 모습을 보였다. 이는 정확한 이유는 알 수 없지만 이후 각층의 feature map을 확인해보면서 모델을 수정해 나갈 것이다.

5. 결론 및 향후 연구

이 논문에서는 여러 형태의 SR모델에서 어떻게 하면 모델을 가볍게 할지 두 가지 방법을 제시하였다. 첫번째로 제시한 Asymmetric convolution을 사용하는 방법은 모델의 연산시간, 파라미터 수를 획기적으로 줄 일 수 있으며 모델의 성능도 큰 차이를 보이지 않은 훌륭한 방법이라고 생각한다. 이를 잘 활용하면 어떠한 환경에서도 실시간 영상처리가 가능한 모델을 만들 수 있을 것이라고 생각한다. 두번째로 제시한 Share Net은 아직 연구가 더 필요하다고 생각한다. 따라서 계속해서 모델의 구조를 바꾸면서 모델을 개선해 나갈 것이다.

참 고 문 헌

- [1] “Deep Learning for Image Super-resolution: A Survey” Zhihao Wang, Jian Chen, Steven C.H. Hoi, Fellow, IEEE arXiv:1902.06068v1 2019
- [2] “Multi-Scale Inception Based Super-Resolution Using Deep Learning Approach” Wazir Muhammad and Supavadee Aramvith, MDPI 2019
- [3] “Accurate Image Super-Resolution Using Very Deep Convolutional Networks” Jiwon Kim, Jung Kwon Lee and Kyoung Mu Lee IEEE 2016
- [4] “Multi-scale Residual Network for Image Super-Resolution” Juncheng Li, Faming Fang ECCV 2018
- [5] “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning” Christian Szegedy, Sergey Ioffe, arXiv:1602.07261v2 [cs.CV] 23 Aug 2016
- [6] “Enhanced Deep Residual Networks for Single Image Super-Resolution” Bee Lim, Sanghyun Son arXiv:1707.02921v1 [cs.CV] 10 Jul 2017
- [7] “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network” Christian Ledig, Lucas Theis, IEEE 2017
- [8] “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks” Xintao Wang, Ke Yu, Shixiang Wu, arXiv:1809.00219v2 [cs.CV] 17 Sep 2018
- [9] “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network” Wenzhe Shi, Jose Caballero, arXiv:1609.05158v2 [cs.CV] 23 Sep 2016