SRGAN 을 이용한 업스케일링

Upscaling using SRGAN

First A. 박주찬

컴퓨터공학부, 한국기술교육대학교, 충청남도 천안시, green261535@gmail.com

Second B. 김지원

컴퓨터공학부, 한국기술교육대학교, 충청남도 천안시, rnemwkqm9@gmail.com

ABSTRACT

영상을 업스케일링하면 결과 영상의 texture detail 이 떨어지게 된다. 최근에는 영상을 업스케일링 한 결과가 객관적 평가와 주관적 평가에서 높은 점수를 받는 방법들이 연구되고 있다. 본 논문은 SRGAN 을 사용하여 이미지를 업스케일링 하는 기법의 결과에 대해 기술한다. DIV2k 라는 고화질 이미지 dataset 을 가지고 학습하고, 검증 데이터를 통해 모델을 평가한다. 모델의 효용성에 대한 평가 지표는 검증 데이터와 결과 이미지간의 PSNR 과 SSIM 점수이고, 기존 이미지 업스케일링에 주로 사용되는 bicubic interpolation 과 비교한다. 평가 결과를 바탕으로 SRGAN 을 사용한 업스케일링의 효용성에 대해 고찰한다.

1 Introduction: About this Template

최근에는 데스크탑, 모바일, TV 등의 디스플레이 기술 발달로 고해상도, 초고해상도의 영상을 제공할 수 있다. 그러나 과거에 생성된 영상들과 같이 저해상도를 가지는 영상들은 디스플레이가 고해상도를 지원함에도 불구하고 여전히 저해상도를 가진다. 이런 문제를 해결하기 위해 일반적으로 업스케일링(upscaling) 영상처리 알고리즘인 보간법(interpolation)을 사용한다. 그러나 보간법은 주어진 샘플들 사이의 값을 추정하는 기법인데, 알고리즘의 특성상 영상의 뭉개짐 현상, 즉 texture detail 이 떨어져 육안으로 보는 데 어려움이 있다.

이 문제를 해결하기 위해 최근 기계학습 연구들은 MSE(Meas Squared reconstruction Error)를 최소화함으로써 SR(Super Resolution)문제를 최적화했다. 이 연구들은 높은 PSNR(Peak Signal-to-Noise Ratios)를 가지지만, 여전히 texture detail 이 부족한 문제가 발생한다.

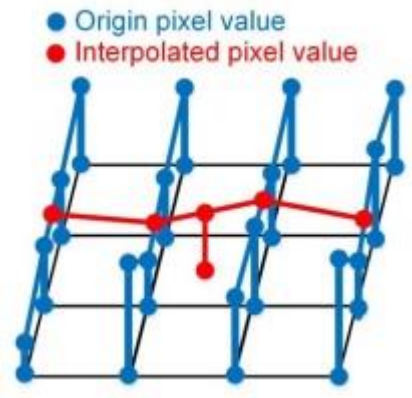

Figure 1: 인접 16 개 화소를 이용하여 보간을 수행하는 3 차회선 보간법(bicubic interpolation)[11]

본 논문에서는 영상의 업스케일링에 이미지 개선 기계학습 알고리즘의 한 종류인 SRGAN(Super Resolution GAN)을 사용하여 영상의 화질 개선에 대해 고찰하고자 한다. 보간법을 사용했을 경우의 이미지 그리고 SRGAN 을 사용한 이미지를 비교하여 효용성을 검증한다.

2 Related work

저해상도의 영상을 업스케일링 하는 알고리즘은 일반적으로 보간법이 사용된다. 대표적인 보간법으로는 Bilinear, Bicubic, Lanczos 등이 있고, 대부분의 프로그램과 소프트웨어에서 사용되는 기법이다. Feature1 은 인접한 16 개 화소를 이용하여 보간을 수행하는 bicubic 알고리즘이다. 각 열에 있는 화소의 값을 참조하여 중간 화소의 값을 구하고, 중간 화소의 값들로부터 다시 새로운 화소의 값을 얻는다. Feture1 의 붉은 점들이 계산을 통해 얻어낸 새로운 화소의 값이다. 보간법은 smoothing 과 유사한 결과를 도출하기 때문에 높은 texture detail 을 기대하기 힘들다.

SR 문제에 대해서 SISR(Single Image Super Resolution)분야에서 최초로 CNN 을 적용한 연구는 SRCNN(dong et al.)[2]으로, 여러가지 단점이 있지만 최신 모델들이 가지는 중요한 요소들은 모두 갖추고 있다. SRCNN 이후 Deep Learning 을 사용한 SR 모델들에 대한 연구가 활발히 진행되고 있다.

Very Deep SR network(VDSR)[7]은 VGG network 구조를 차용하고, 깊은 모델의 학습을 위해 global residual learning 과 gradient clipping 을 동반한 adjustable high learning rate 를 제안했다.

SRGAN 은 2017 년도에 나온 논문이며, 이후 SRGAN 을 기반으로 둔 여러 논문들이 출시됐다. SRGAN 에서 VGG network 을 사용해 Generator loss 를 측정하지만, VGG network 는 깊은 모델에 적합하지 않기 때문에 보다 local 하게 residual learning 을 수행하는 ResNet 을 사용한 신경망인 SRResNet[8]과 ResNet 을 SISR 문제에 좀 더 적합하도록 개선한 Enhanced Deepresidual learning for SISR(EDSR)[9]이 연구되었다. SRResNet 은 batch normalization 을 사용하므로 깊은 모델을 안정적으로 학습할 수 있었고, 비슷한 맥락에서 feature map 끼리 concatenation 을 하는 DenseNet 을 사용한 모델로는 Residual DenseNet[10] 등이 연구되었다.

보간법이나 기계학습 등의 방법으로 영상을 업스케일링 한 후 그 결과들을 비교할 때, 얼마나 원본 영상에 충실하게 복원되었는지를 객관적인 지표로 비교할 방법이 필요하다. 일반적인 방법으로 객관적인 지표인 PSNR 과 SSIM(Structural Similarity)을 사용하여 결과 영상 화질의 열화 정도를 얻을 수 있다.

3 Task and Model

SRGAN 은 GAN(Generative Adversarial Network)과 동일한 신경망 구조를 사용한다. GAN 에 대해서는 [1]에서 살펴볼 수 있다. GAN 과 동일한 신경망 구조를 사용하기 때문에 perceptual similarity 를 주목한 loss function 을 사용한다.

3.1 Model Definition

SRGAN 모델은 Figure2 에서 볼 수 있듯이 Generator Network 와 Discriminator Network 로 이루어진다.

Generator Network 의 핵심 부분은 동일한 레이아웃의 B residual blocks 으로 이루어져 있다. 해당 block 의 레이아웃은 Gross 와 Wilber 가 제안한 형태의 레이아웃을 사용한다[12]. B residual block 에서 3x3 크기의 커널과 64 개의 feature map 을 가지는 convolutional layer 와 batch-normalization[13] layer 를 거친 후 활성 함수(activation function)으로 ParametricReLU[14]를 사용한다. 이후 convolutional layer, PixelShuffler, PReLU 로 이루어진

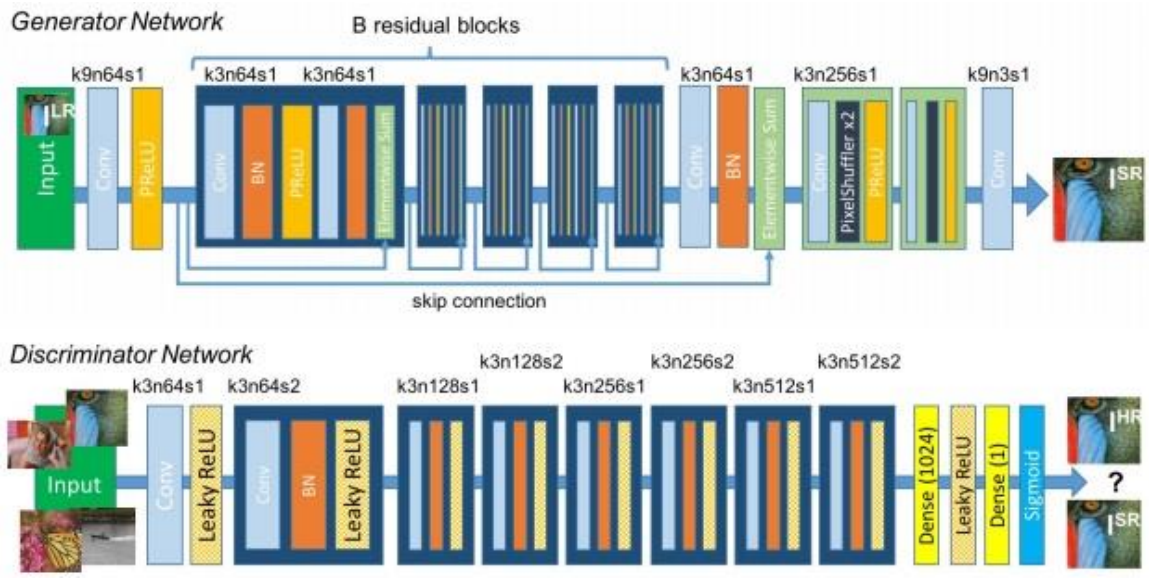

Figure 2: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.[8]

block 을 거치는데, 이 block 은 두 개의 학습된 sub-pixel convolution layer 로 입력 영상의 해상도를 높인다[15].

Generator Network 를 거쳐 생성된 SR 영상과 원본 영상을 구별하기 위해 Discriminator Network 를 학습시킨다. Feture2 에 있는 Discriminator Network 은 Radford et al.[16]에서 제안된 구조를 따른다. 활성 함수로는 LeakyReLU(α = 0.2)를 사용하여 max pooling 을 피한다. Discriminator Network 에서의 convolutional layer 도 마찬가지로 3x3 의 커널을 가진다. VGG network[17]에서와 같이 처음에는 64 개의 channels 을 가지고 마지막엔 512 개의 channels 을 가지도록 한다. channels 의 수가 두 배가 될 때마다 이미지의 해상도를 stride 의 값을 2 로 주어 반으로 줄이며 convolution 을 진행한다. 얻어진 512 개의 channels 을 가진 Deep feature 는 두 개의 dense layer 와 sigmoid 활성 함수를 거쳐 확률 값을 얻어낸다. 이 확률 값을 통해 Discriminator Network 는 입력 영상이 SR 영상인지 원본 영상인지를 판단한다.

3.2 Loss function

일반 GAN 의 Discriminator loss 는 그대로 사용하였고, Generator loss 는 SRGAN 논문의 Perceptual loss function[5]을 가져왔다. 여기에 추가적으로 WGAN 논문에서 제안한 Total variation loss function[6]을 추가하여 loss function 을 만들었다. Total variation loss

function 을 쓰는 이유는 GAN 에서 이미지를 생성하는 경우 이미지를 좀 더 부드럽고 자연스럽게 해주기 때문이다.

3.2.1 Generator loss

Generator loss 는 Perceptual loss + Total variation loss 이다.

Perceptual loss

$$l^{SR} = \underbrace{l_X^{SR}}_{\text{Content loss}} + \underbrace{10^{-3} l_{Gen}^{SR}}_{\text{Adversarial loss}} \tag{1}$$

Total variation loss

$$l_{Gen}^{TV} \tag{2}$$

1. Content loss

$$l_X^{SR} = l_{MSE}^{SR} + \mathrm{l}_{VGG/i,j}^{SR} \tag{3}$$

식 (3)의 l_{MSE}^{SR}는 Pixel-wise MSE loss 이다. 이 loss 의 문제점은 높은 PSNR 을 얻을 수 있지만, 사람이 보기에 부자연스러울 수 있다는 단점이 있다. 따라서 pre-trained 된 VGG net 을 사용해서 feature map 에서의 Euclidean distance 를 계산하는 $\mathrm{l}_{VGG/i,j}^{SR}$ 를 추가로 사용한다.

$$l_{MSE}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \left(I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y} \right)^2 \tag{4}$$

$$l_{VGG/i,j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}\left(G_{\theta_G}(I^{LR}) \right)_{x,y} \right)^2 \tag{5}$$

$\phi_{i,j}$: VGG19 에서 i 번째 maxpooling layer 를 거치기 전 j 번째 conv 에서 얻어진 feature map 을 가리킨다.

2. Adversarial loss

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}\left(G_{\theta_G}(I^{LR}) \right) \tag{6}$$

3. Total variation loss

$$l_{Gen}^{TV} = \frac{1}{r^2 W(H-1)} \left(\sum_{y=2}^{rH} G_{\theta_G}(I^{LR})_{x,y} - \sum_{y=1}^{rH-1} G_{\theta_G}(I^{LR})_{x,y} \right)^2 + \frac{1}{r^2 (W-1)H} \left(\sum_{x=2}^{rW} G_{\theta_G}(I^{LR})_{x,y} - \sum_{x=1}^{rW-1} G_{\theta_G}(I^{LR})_{x,y} \right)^2 \quad (7)$$

3.2.2 Discriminator loss

일반 GAN 의 Discriminator loss 를 그대로 사용했다.

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR} \sim p_{\text{train}}(I^{HR})} \left[\log D_{\theta_D}(I^{HR}) \right] + \mathbb{E}_{I^{LR} \sim p_G(I^{LR})} \left[\log \left(1 - D_{\theta_D} \left(G_{\theta_G}(I^{LR}) \right) \right) \right] \quad (8)$$

4 Experiments

Data Augmentation 으로는 Training 고해상도 이미지를 88, 128, 168 크기의 이미지로 랜덤하게 추출하여 Target(HR image)으로 두고, HR image 를 BICUBIC interpolation 방법으로 크기를 4 배 또는 2 배로 줄여 학습 모델의 Input 으로 사용하여 학습시켰다. Optimizer 는 Adam 을 사용하였고, learning rate 는 0.001, beta1 과 beta2 는 0.9, 0.999 으로 두었다.

4.1 Data

학습데이터와 검증 데이터는 DIV2K dataset 를 사용했다. 학습데이터는 총 800 개의 고해상도 이미지이며 검증 데이터는 100 개의 고해상도 이미지이다.

4.2 Evaluation Measures

평가도구로는 PSNR(Peak Signal to Noise Ratio)과 SSIM((Structure Similarity Index)을 사용하였다. PSNR 이란 최대 신호대 잡음비이며 주로 영상 또는 동영상 화질 손실 정보를 평가할 때 사용한다. 손실이 적을수록 높은 값을 갖는다. PSNR 은 단순히 원본 이미지와 왜곡 이미지 사이의 수치적 차이로 이미지 품질을 평가하는 기법이다. 하지만 [3], [4]의 여러 연구를 통해 PSNR 기법은 인간의 시각체계에 적합하지 않는 이미지를 평가 방법이라는 것을 보여주었다. 대안으로 나온 평가 기법으로는 SSIM(Structure Similarity Index)이며, SSIM 은 이미지의 전체적인 구조적 관점에서 인식하여 품질을 평가한다. 이렇게 PSNR 과 SSIM 두가지를 가지고 평가 지표로 사용하였다.

4.3 Evaluation Setting

4.3.1 PSNR 의 수식

$$\begin{aligned}\mathrm{PSNR} &= 10\log_{10}\left(\frac{(MAX)^2}{MSE}\right)\\ &= 20\log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right)\\ &= 20\log_{10}(MAX) - 10\log_{10}(MSE)\end{aligned} \tag{9}$$

4.3.2 SSIM 의 수식

SR 이미지와 HR 이미지간의 평균 휘도 비교:

$$l(SR,HR) = \frac{2\mu_{SR}\mu_{HR} + C_1}{\mu_{SR}^2 + \mu_{HR}^2 + C_1} \tag{10}$$

SR 이미지와 HR 이미지의 표준편차값을 이미지의 대비로 정의하고, 두 이미지의 표준편차값을 비교:

$$c(SR,HR) = \frac{2\sigma_{SR}\sigma_{HR} + C_2}{\sigma_{SR}^2 + \sigma_{HR}^2 + C_2} \tag{11}$$

SR 이미지와 HR 이미지간의 왜곡된 구조를 다음과 같이 비교:

$$s(SR,HR) = \frac{2\sigma_{SR,HR} + C_3}{\sigma_{SR}\sigma_{HR} + C_3} \tag{12}$$

(9), (10), (11)식을 묶어 SSIM 의 수식은 다음과 같다.

$$\begin{aligned}SSIM(SR,HR) &= l(SR,HR)c(SR,HR)s(SR,HR)\\ &= \frac{(2\mu_{SR}\mu_{HR} + C_1)(2\sigma_{SR,HR} + C_2)}{(\mu_{SR}^2 + \mu_{HR}^2 + C_1)(\sigma_{SR}^2 + \sigma_{HR}^2 + C_2)}\end{aligned} \tag{13}$$

C_1, C_2, C_3는 아래와 같이 정의(L 은 default 값):

$$\begin{aligned}C_1 &= (0.01 * L)^2\\ C_2 &= (0.03 * L)^2\\ C_1 &= \frac{C_2}{2}\end{aligned} \tag{14}$$

4.4 Results

Table 1: DataSet : DIV2k(2x upsample)

	BICUBIC PSNR	BICUBIC SSIM	SRGAN PSNR	SRGAN SSIM
Crop size(88)	28.3654	0.8877	29.0084	0.8938
Crop size(128)	28.3654	0.8877	**29.0127**	0.8949
Crop size(168)	28.3654	0.8877	29.0124	**0.8951**
Crop size(88) without tv loss	.	.	28.9112	0.8895
Crop size(128) without tv loss	.	.	28.9228	0.8906
Crop size(168) without tv loss	.	.	28.9172	0.8901

Table 2: DataSet : DIV2k(4x upsample)

	BICUBIC PSNR	BICUBIC SSIM	4x PSNR	4x SSIM
Crop size(88)	24.5025	0.7408	25.0117	0.7541
Crop size(128)	24.5025	0.7408	**25.0189**	**0.7558**
Crop size(168)	24.5025	0.7408	25.0168	0.7554
Crop size(88) without tv loss	.	.	24.8942	0.7482
Crop size(128) without tv loss	.	.	24.8347	0.7494
Crop size(168) without tv loss	.	.	24.8742	0.7491

Figure 1: SRGAN 4x upsampling image (data : BSDS100)

Figure 1: BICUBIC 4x upsampling image (data : BSDS100)

4.5 Analysis

보통 이미지를 학습시킬 때 입력 이미지의 크기를 크게 해주면 더 나은 성능을 보인 결과들이 많았다. 그래서 SRGAN 모델의 입력으로 넣을 이미지의 사이즈가 화질 개선 성능에 영향을 미치는지 확인을 해보았다. Bicubic interpolation 방법은 입력 이미지 사이즈와 관계없이 PSNR 과 SSIM 값이 일정하게 나왔으며, SRGAN 을 통해 학습시킨 결과도 입력 이미지가 크거나 작거나 대체적으로 성능이 비슷한 것을 볼 수 있었다. 이미지 분류 문제일 경우에는 입력 이미지의 사이즈가 성능에 영향을 미친다고 볼 수 있지만, 화질 개선 문제일 경우에는 입력 이미지의 사이즈가 성능에 영향을 미치지 않는다는 결론을 도출할 수 있었다. 이미지를 2 배와 4 배 확대하는 두가지 경우에 대해서 실험결과 2 배를 확대하는 것보다 4 배를 확대했을 때 성능이 확 떨어지는 것을 확인할 수 있었으며 이것은 당연한 결과라고 생각한다. TV loss 가 과연 꼭 필요한지에 대해서도 실험을 해본 결과 성능이 크게 차이가 나지 않았지만, TV loss 를 사용할 경우 성능이 약간 증가한 것을 볼 수 있었다.

5 Conclusion

이미지를 화질 개선할 때, 딥러닝을 사용하면 효과적이라고 할 수 있다. 이미지 화질 개선 Task 에는 입력 이미지의 크기가 성능에 영향을 미치지 않았다. GAN 기반의 모델이 이미지 화질개선 하는데 있어 수치적인 성능은 크게 높이진 못하지만 사람이 봤을 때 이미지가 자연스럽고 매끄럽다는 장점을 지니고 있다. 반면에 GAN 기반의 모델의 단점으로는 Generator 가 이미지를 자연스럽게 만들려고 하기 때문에 화질 개선한 이미지가 정확한 이미지가 아닐 수 있다. 그러므로 이미지의 정보가 중요한 Task 에서 입력 이미지의 화질을 개선하려고 SRGAN 을 사용한다고 해서 성능이 좋아질 수도 있지만 아닐 가능성도 크며 해볼만한 연구일 것 같다. 결론으로는 이미지 또는 영상의 화질이 좋지 않을 때, SRGAN 을 이용해 이미지의 화질을 개선한다면 사람 눈에 보기에 자연스러운 이미지를 얻을 수 있으며, TV loss 를 같이 사용한다면 이미지가 좀 더 부드러워지는 효과를 얻을 수 있다.

REFERENCES

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

[2] Dong, C., Loy, C. C., He, K., & Tang, X. (2014, September). Learning a deep convolutional network for image super-resolution. In *European conference on computer vision* (pp. 184-199). Springer, Cham.

[3] Z. Wang and A. C. Bovik, "A universal image quality index," IEEE Signal Processing Letter, Vol. 9, No. 3, pp. 81-84, Mar. 2002

[4] Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it?," IEEE Signal Process. Mag. 26(1) pp. 98-117, 2009.

[5] Ledig, Christian & Theis, Lucas & Huszar, Ferenc & Caballero, Jose & Cunningham, Andrew & Acosta, Alejandro & Aitken, Andrew & Tejani, Alykhan & Totz, Johannes & Wang, Zehan & Shi, Wenzhe. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. 105-114. 10.1109/CVPR.2017.19.

[6] Gulrajani, Ishaan, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin and Aaron C. Courville. "Improved Training of Wasserstein GANs." *NIPS* (2017).

[7] Kim, J., Kwon Lee, J., & Mu Lee, K. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1646-1654).

[8] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).

[9] Lim, B., Son, S., Kim, H., Nah, S., & Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 136-144).

[10] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2472-2481).

[11] 노유리, & 이성길. (2018). GPU 기반 후처리 효과에 대한 업스케일링의 효용성 실험. *정보과학회논문지*, *45*(7), 618-625.

[12] Gross, S., & Wilber, M. Training and investigating residual nets (2016). *URL http://torch. ch/blog/2016/02/04/resnets. html*.

[13] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[14] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).

[15] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel

convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1874-1883).

[16] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

[17] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.