DeblurGAN

current topic in AI

Soojung Hong

June 17. 2021

Content

- DeblurGAN models

  [1] DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks

  https://arxiv.org/abs/1711.07064

  [2] DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better

  https://arxiv.org/abs/1908.03826

- DeblurGAN ver1, ver2 architecture / loss function comparison
- Deblurred image quality metrics (PSNR, SSIM)
- Probe histogram image experiment results
- Use case in HERE

Deblurred image example by DeblurGAN

✓ YOLO object detection performance

Blurred image

Deblurred image

Sharp image

[1] DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks

https://arxiv.org/abs/1711.07064

Common formulation of non-uniform blur model

$$I_B = k(M) * I_S + N,$$

- I_B : blurred image
- $k(M)$: unknown blur kernels determined by motion field M.
- I_S : sharp latent image, denotes the convolution,
- N : an additive noise.

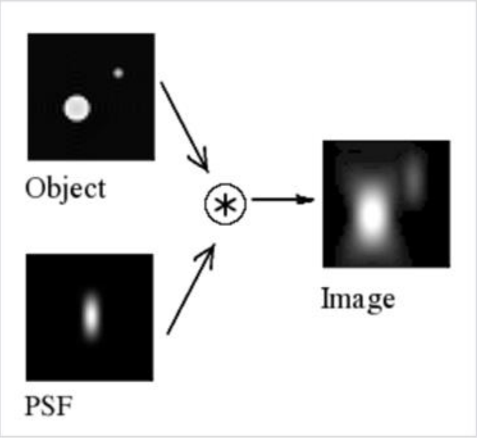


Image formation in a confocal microscope: central longitudinal (XZ) slice. The 3D acquired distribution arises from the convolution of the real light sources with the PSF.

✓ Two types of deblurring : Blind deblurring, Non-blind deblurring

DeblurGAN ver1 Architecture

CNN learns a residual correction IR to the blurred image IB, so IS = IB + IR.

IS is sharp image, IB is blur image, IR is residual correction

global skip connection

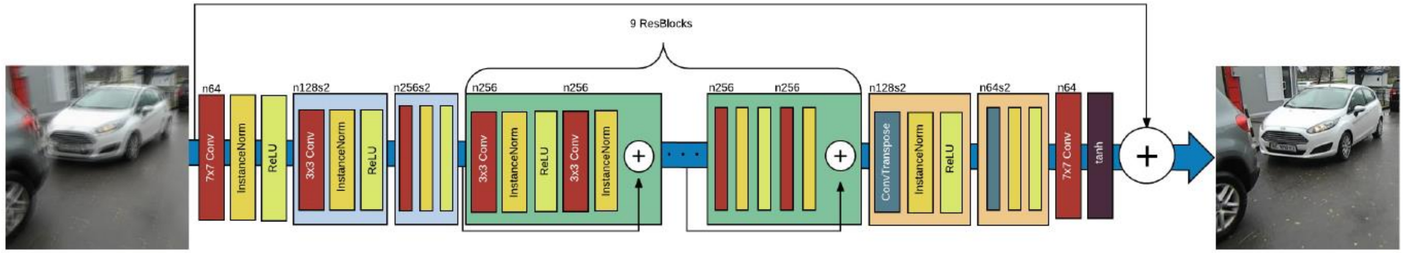


Figure 3: DeblurGAN generator architecture. DeblurGAN contains two strided convolution blocks with stride $\frac{1}{2}$, nine residual blocks [13] and two transposed convolution blocks. Each ResBlock consists of a convolution layer, instance normalization layer, and ReLU activation.

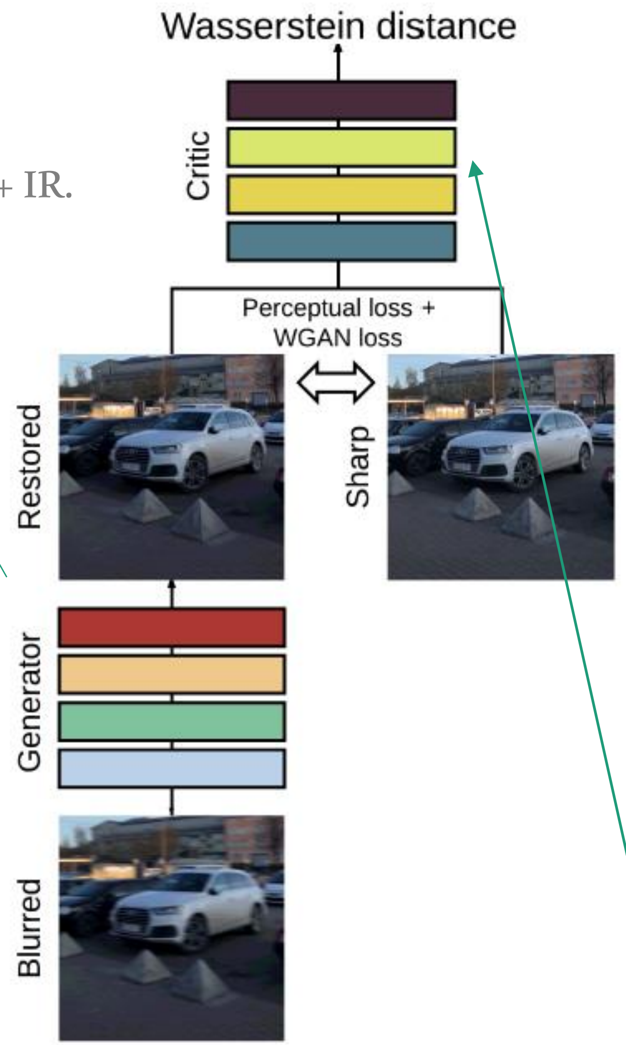
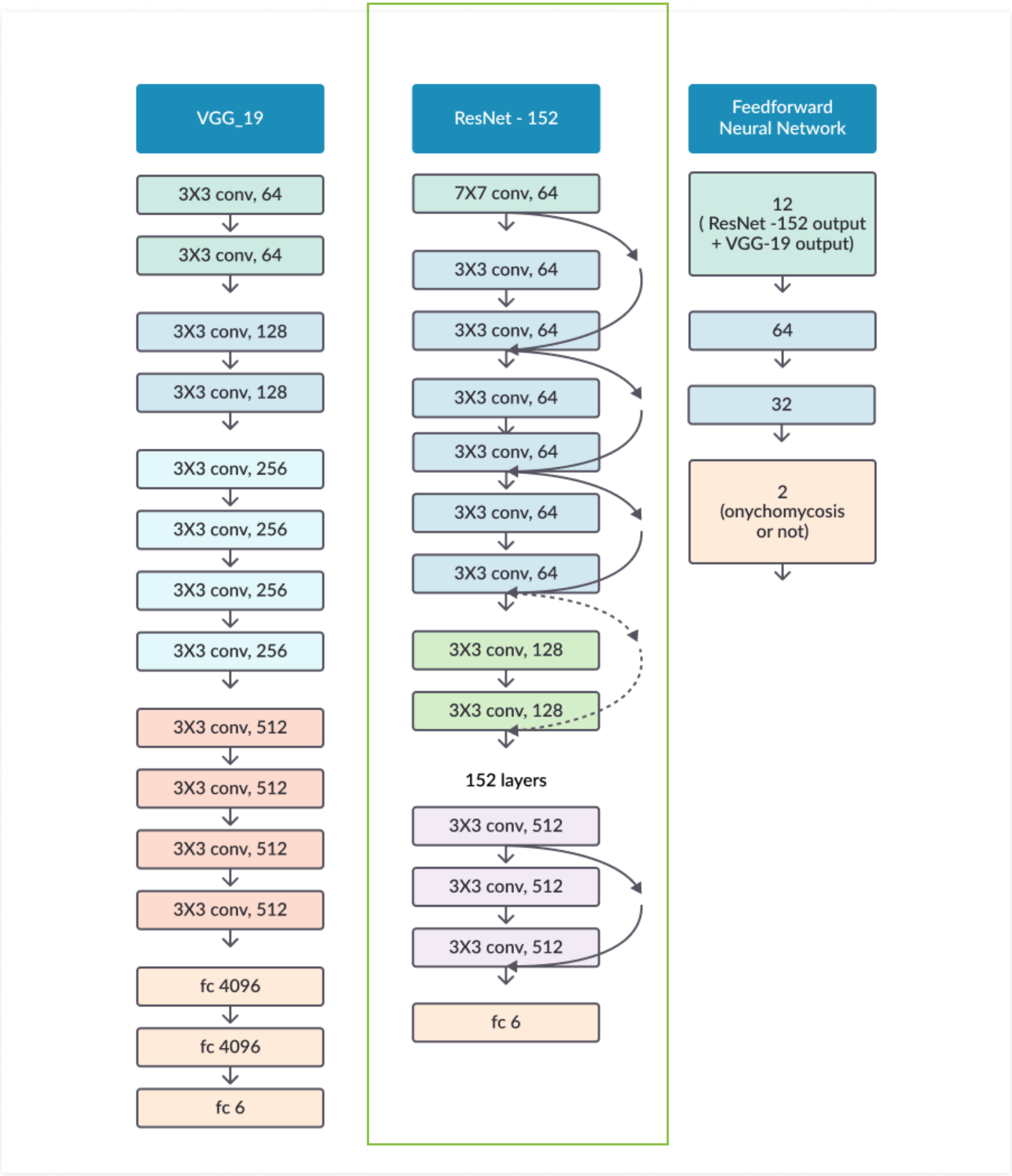


Figure 4: DeblurGAN training. The generator network

Critic network D architecture of critic network is identical to PatchGAN

DeblurGAN verı

Backbone network : ResNet



Standard GAN training : minimax objective

$$\min_G \max_D \underbrace{\mathbb{E}_{x\sim\mathbb{P}_r}[\log(D(x))] + \underbrace{\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[\log(1 - D(\tilde{x}))]}}$$

Discriminators try to **maximize**

(i.e. D(x) – the probability that real image is real is high)

D(G(z)) is the probability that fake image is real become low

(close to 0)

Generators try to minimize

(i.e. try to maximize the discriminator's

output for fake instance , D(G(z)), is

probably real (close to 1)

➔ Difficult to train GAN due to Mode Collapse, Gradient Vanishing/Exploding and unstable training

DeblurGAN ver1 loss

Loss function : combination of content loss and adversarial loss

$$\underbrace{\mathcal{L} = \underbrace{\mathcal{L}_{GAN}}_{adv\ loss} + \underbrace{\lambda \cdot \mathcal{L}_X}_{content\ loss}}_{total\ loss}$$

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}$$

Wasserstein GAN - GP

- ✓ WGAN-GP improve stability during training model
- ✓ WGAN-GP is robust to the choice of generator architecture
- ✓ Loss function helps to generate higher quality result

"content" loss function : L1 or MAE loss, L2 or MSE loss on raw pixels.

content loss can lead to the **blurry artifacts** on generated images

Instead, proposed **Perceptual loss (** a simple L2-loss,) but based on the difference of **the generated and target image CNN feature maps**

$$\mathcal{L}_X = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^S)_{x,y} - \phi_{i,j}(G_{\theta_G}(I^B))_{x,y})^2$$

Feature map by j-th convolution before ith maxpooling layer in VGG19 network

DeblurGAN ver 1

Method	Sun *et al.*	Nah *et al.*	Xu *et al.*	Whyte *et al.*	DeblurGAN		
Metric	[36]	[25]	[44]	[40]	*WILD*	*Synth*	*Comb*
PSNR	25.22	26.48	**27.47**	27.03	26.10	25.67	25.86
SSIM	0.773	0.807	0.811	0.809	**0.816**	0.792	0.802

(a) Blurred photo (b) Nah *et al.* [25] (c) DeblurGAN (d) Sharp photo

Figure 9: YOLO object detection before and after deblurring

Evaluation Metric (quality metric)

- PSNR (Peak Signal to Noise ratio)

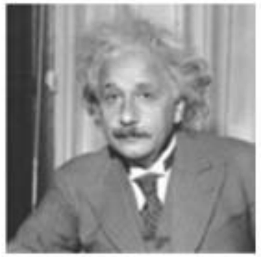
$$MSE = \frac{1}{wh}\sum_{i=0}^{h-1}\sum_{j=0}^{w-1}|I(i,j) - K(i,j)|^2$$

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

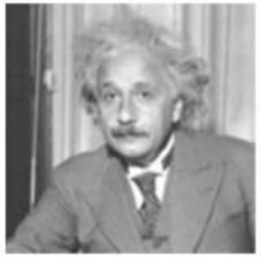
power of distorting noise

MAX$_f$ is the maximum signal value that exists in our original "known to be good" image

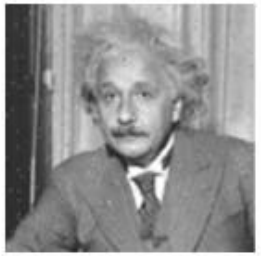
✓ strictly on numeric comparison and does not actually take into account any level of biological factors of the human vision system



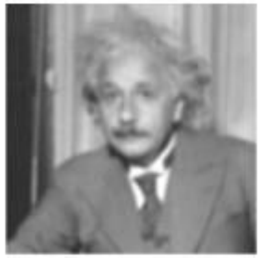
Original
SSIM=1



PSNR=26.547
SSIM=0.988



PSNR=26.547
SSIM=0.840



PSNR=26.547
SSIM=0.694

- SSIM (similarity structure Index measure)

$$\text{SSIM}(\mathbf{x},\mathbf{y}) = [l(\mathbf{x},\mathbf{y})]^{\alpha} \cdot [c(\mathbf{x},\mathbf{y})]^{\beta} \cdot [s(\mathbf{x},\mathbf{y})]^{\gamma}$$

✓ **SSIM** is more perceptual metric

✓ **SSIM** is a newer measurement tool based on three factors (i.e. luminance, contrast, and structure to better suit the workings of the human visual system)

DeblurGAN version 2 architecture

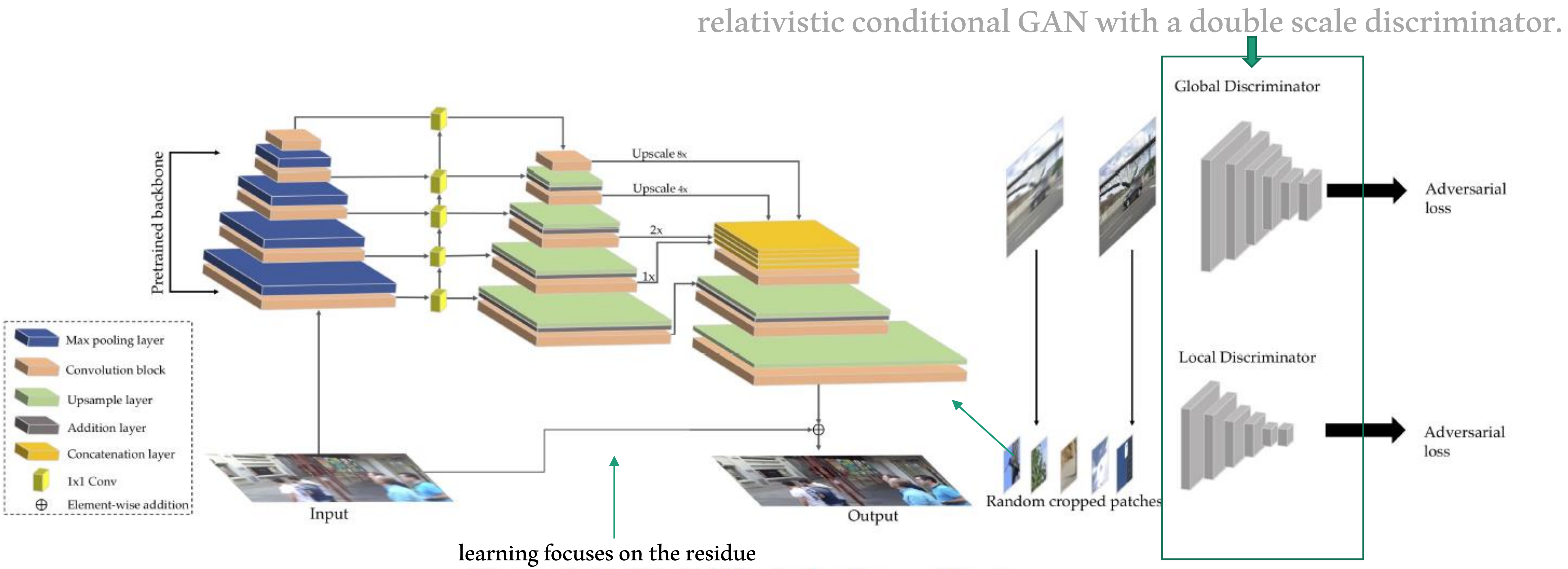


Figure 2: DeblurGAN-v2 pipeline architecture.

- ✓ Core building block in Generator : Feature Pyramid Network
- ✓ It can flexibly work with a wide range of backbones

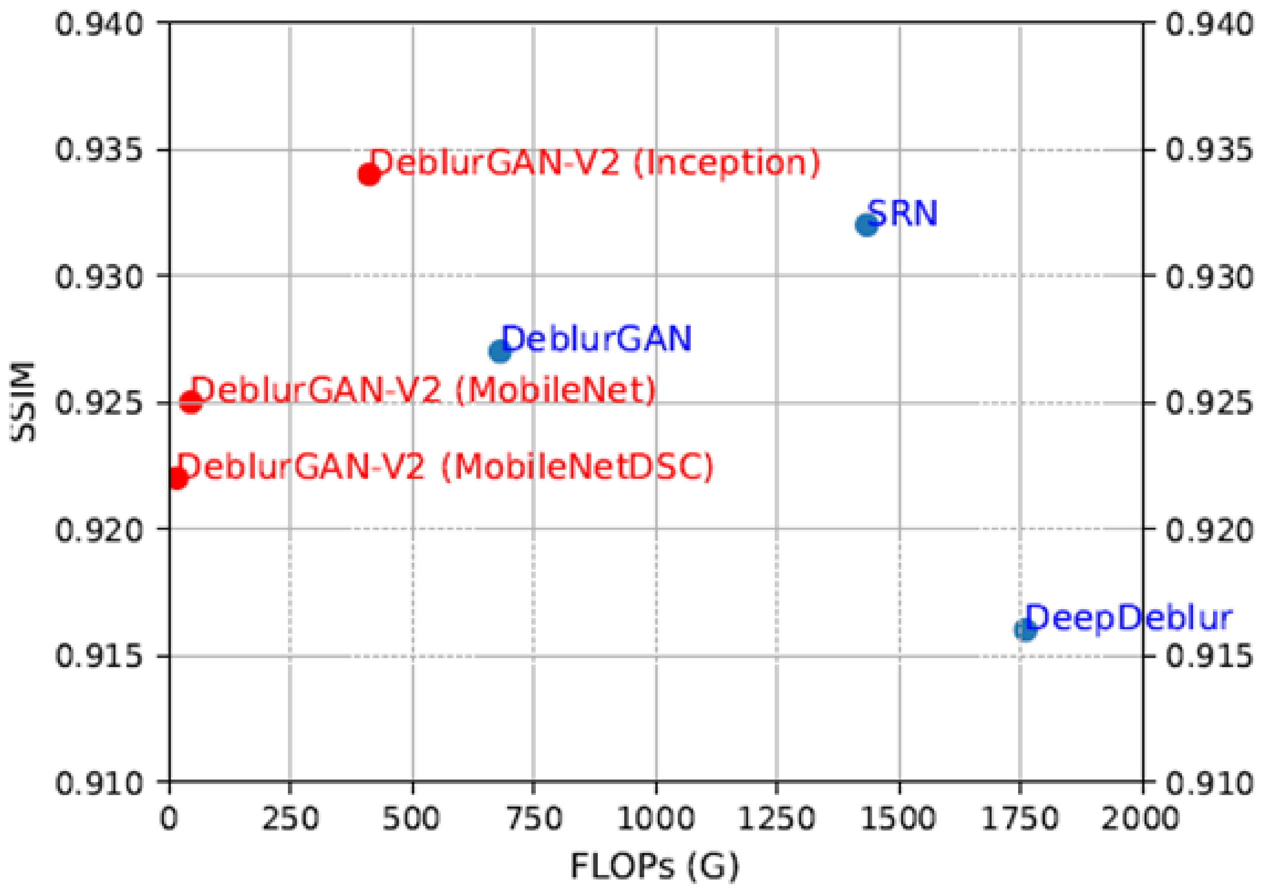
Inception-ResNet-v2, SEResNeXt, MobileNet V2 backbone and MobileNet-DSC (for mobile on device)

SSIM-FLOP

✓ Performance measure :

SSIM : structural similarity index measure

FLOPs : floating point operations per second



DeblurGAN ver2 loss

$$L_G = 0.5 * L_p + 0.006 * L_X + 0.01 * L_{adv}$$

Pixel-space loss (simplest L1 or L2 distance)

Lp term was not included in ver 1

It helps correct color and texture distortions

Relativistic GAN loss

$$L_D^{RaLSGAN} = \mathbb{E}_{x \sim p_{data}(x)}\left[(D(x) - \mathbb{E}_{z \sim p_z(z)} D(G(z)) - 1)^2\right] + \mathbb{E}_{z \sim p_z(z)}\left[(D(G(z)) - \mathbb{E}_{x \sim p_{data}(x)} D(x) + 1)^2\right]$$

Content loss

Again, using perceptual distance

(Euclidean loss on the VGG19 conv3 3 feature maps)

$$\mathcal{L}_X = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^S)_{x,y} - \phi_{i,j}(G_{\theta_G}(I^B))_{x,y})^2$$

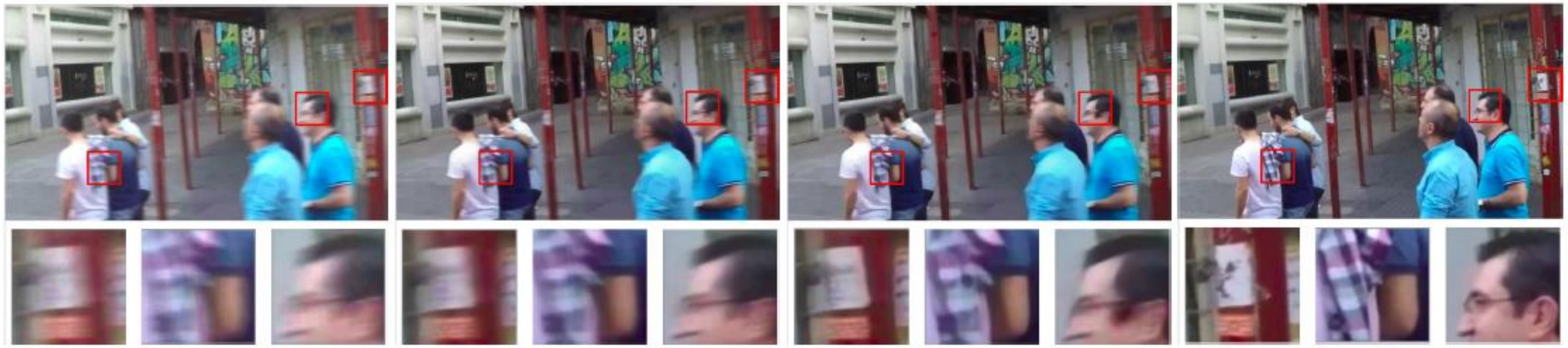
Feature map by j-th convolution before ith maxpooling layer in VGG19 network

- ✓ notably faster
- ✓ more stable compared to WGAN-GP objective
- ✓ empirically the generated results possess higher perceptual quality and overall sharper outputs

Evaluation

Method	Sun *et al.*	Nah *et al.*	Xu *et al.*	Whyte *et al.*	DeblurGAN		
Metric	[36]	[25]	[44]	[40]	*WILD*	*Synth*	*Comb*
PSNR	25.22	26.48	**27.47**	27.03	26.10	25.67	25.86
SSIM	0.773	0.807	0.811	0.809	**0.816**	0.792	0.802

DeblurGAN ver1



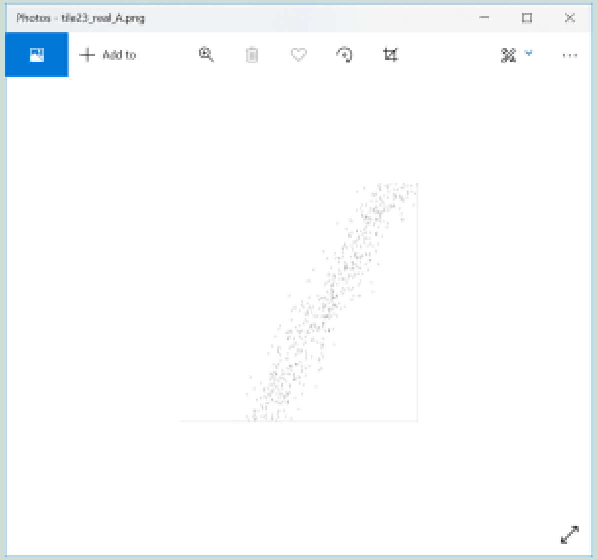
(a) Degraded photo (b) DeblurGAN (c) DeblurGAN-v2 (Inception-ResNet-v2) (d) Clean photo

Figure 6: Visual comparison example on the Restore Dataset.

	Sun *et al.*	Nah *et al.*	Xu *et al.*	DeblurGAN		
Metric	[36]	[25]	[44]	*WILD*	*Synth*	*Comb*
PSNR	24.6	28.3/**29.1***	25.1	27.2	23.6	**28.7**
SSIM	0.842	0.916	0.89	0.954	0.884	**0.958**
Time	20 min	4.33 s	13.41 s	**0.85** s		

more than 6x fewer parameters comparing to Multi-scale CNN heavily speeds up the inference.

Test image comparison between DeblurGAN ver1 & ver2

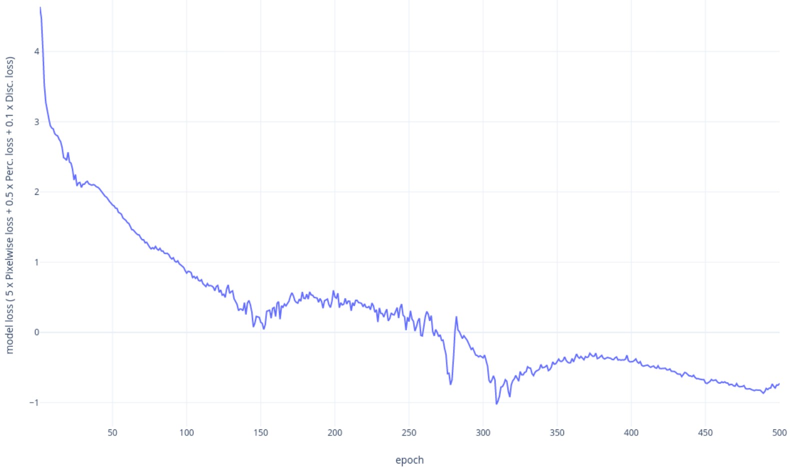
DeblurGAN ver1	DeblurGAN ver2
	

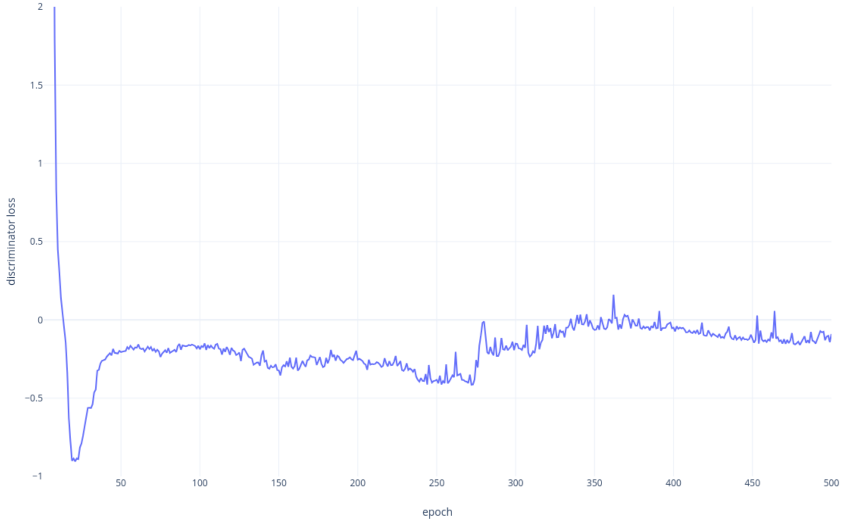
Comparison of training : ver1 vs ver2

Dataset	Model	Config (hyperparameter) Setting	PSNR		SSIM	
			Train	Test	Train	Test
Benchmarking Results#SF-L26-M5-LC	DeblurGAN v1	BS=1 CU=1 RL=1e-5 Epoch=500 Model loss=5 x Pixel-wise + 0.5 x Perc. loss + 0.1 x Disc. loss	29.91	21.75	0.9690	0.9037
		BS=1 CU=1 RL=5e-6 Epoch=500 Model loss=5 x Pixel-wise + 0.5 x Perc. loss + 0.1 x Disc. loss	26.16	21.39	0.9565	0.9014
	DeblurGAN v2	› configuration	25.37	23.56 ✓	0.968	0.9217 ✓

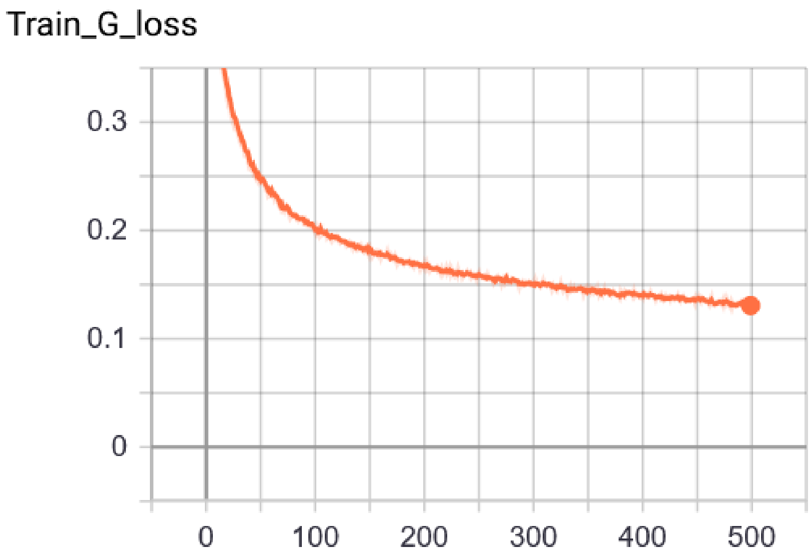
Training loss comparison : ver1 vs ver2

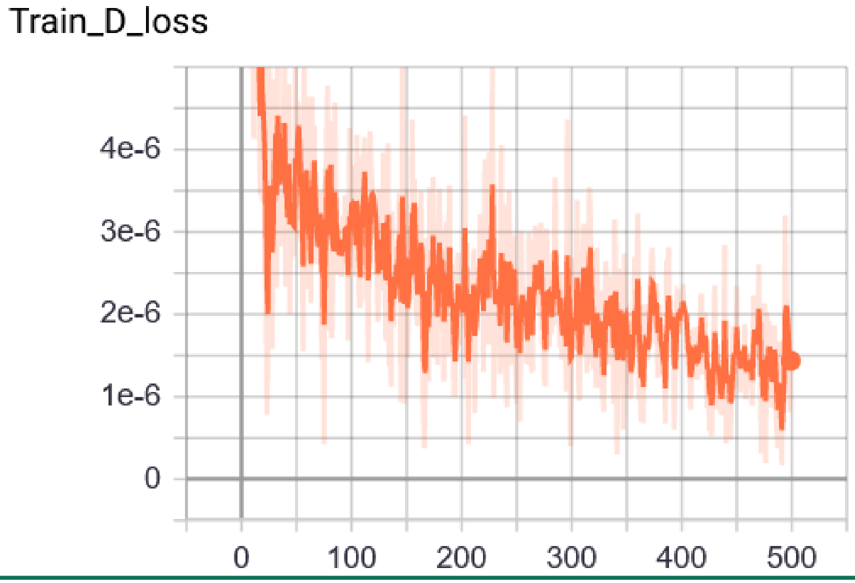
ver1





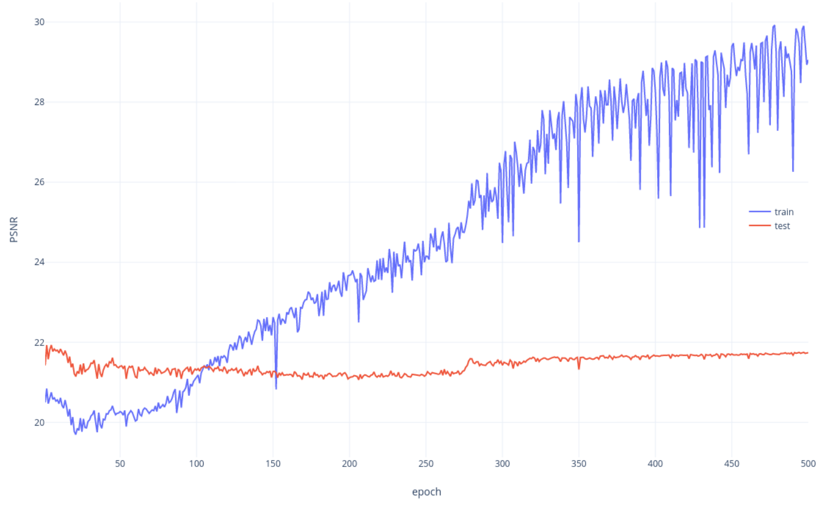
ver2

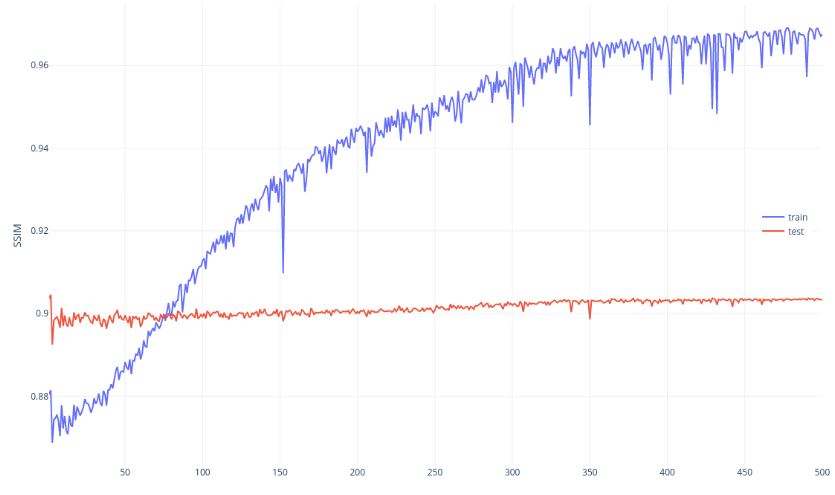




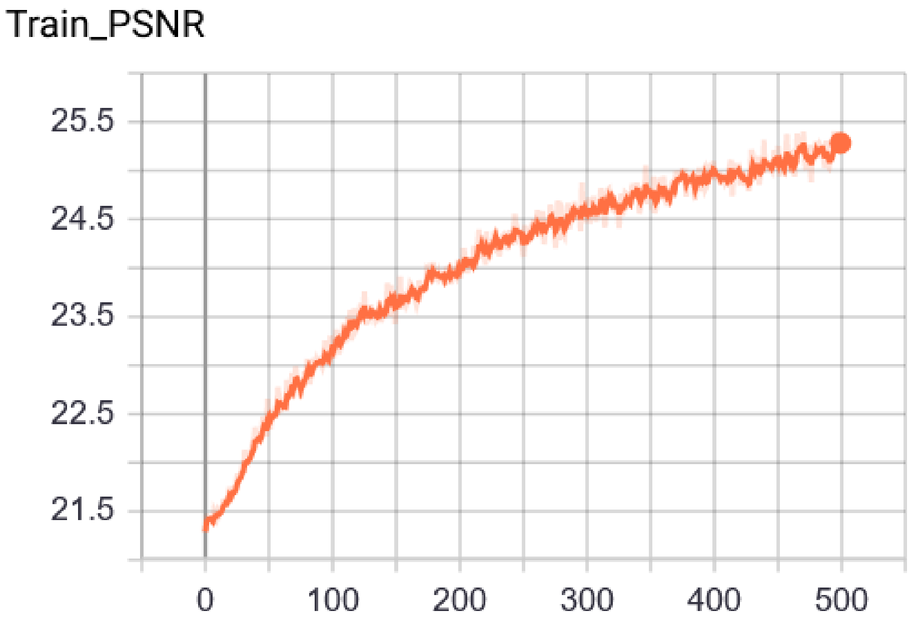
Comparison PSNR, SSIM quality metrics : ver1 vs ver2

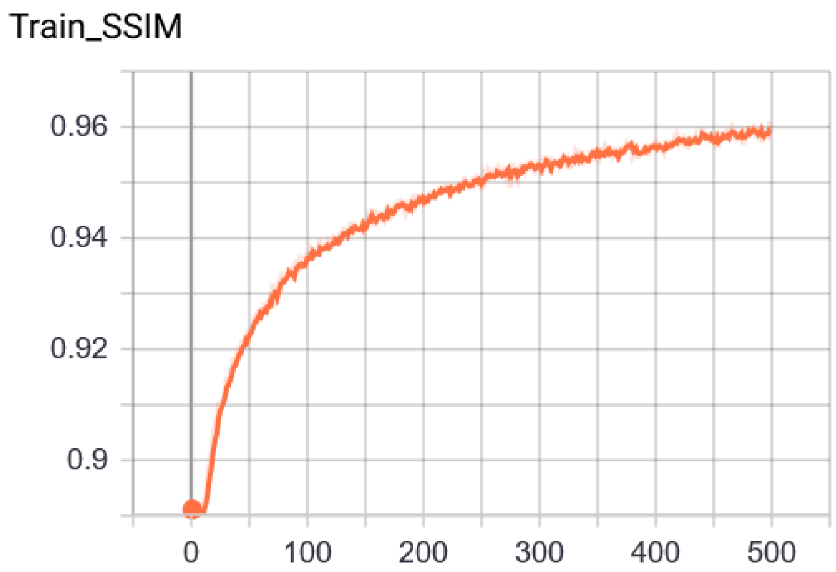
ver1





ver2





User case work for HERE

- Preprocessing for low quality Aerial image before object detection
- HERE SLI (street level imagery) object deblurring