Image Deblurring

PRESENTED BY 薛铭龙

Method

PART ONE

DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks

Main Content

Problem : This work is on blind motion deblurring of a single photograph.

Method : We present DeblurGAN, an end-to-end learned method for motion deblurring. The learning is based on a conditional GAN and the content loss.

- We propose a loss and architecture which obtain state-of-the art results in motion deblurring, while being 5x faster than the fastest competitor.
- We present a method based on random trajectories for generating a dataset for motion deblurring training in an automated fashion from the set of sharp image.
- We present a novel dataset and method for evaluation of deblurring algorithms based on how they improve object detection results.

Motion Deblurring

Image Deblurring

The common formulation of non-uniform blur model is the following:

$$I_B = k(M) * I_S + N \qquad (1)$$

where I_B is a blurred image, k(M) are unknown blur kernels determined by motion field M. I_S is a sharp image, ∗ denotes the convolution, N is an additive noise.

The family of deblurring problems is divided into two types:

- **Blind : deblurring**
- Non-blind : the blur kernels k(M) are known

Finding a blur function for each pixel is an **ill-posed** problem, and most of the existing algorithms rely on heuristics, image statistics and assumptions on the sources of the blur.

Deblur Generative Part Structure

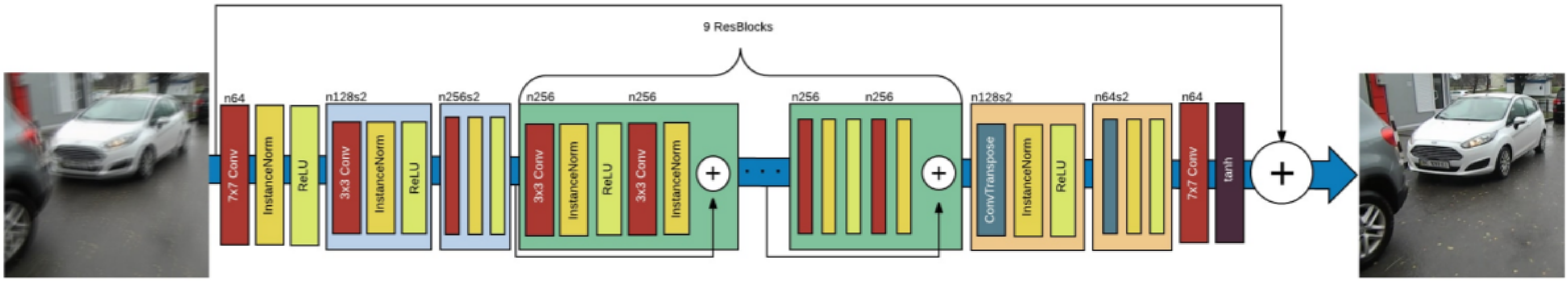


Figure 3: DeblurGAN generator architecture. DeblurGAN contains two strided convolution blocks with stride $\frac{1}{2}$, nine residual blocks [13] and two transposed convolution blocks. Each ResBlock consists of a convolution layer, instance normalization layer, and ReLU activation.

With the success of deep learning, over the last few years, there appeared some approaches based on convolutional neural networks (CNNs).

Ramakrishnan et al. use the combination of pix2pix framework and densely connected convolutional networksto perform blind kernel-free image deblurring. Such methods are able to deal with different sources of the blur.

Deblur Critic(Discriminator) Part Structure

$$W(P_{real}\|P_{fake}) = \inf_{\gamma\sim\prod(P_r,P_g)} E_{(x,y)\sim\gamma}[\|(x-y)\|] \quad (2)$$

Kantorovich-Rubinstein duality

$$W(P_{real}\|P_{fake}) = \sup_{\|f\|_L\le 1} E_{x\sim P_r}[f(x)]-E_{x\sim P_\theta}[f(x)] \quad (3)$$

根据 lipschitz 条件：

$$\|f(a)-f(b)\| \le K\|a-b\| \quad (4)$$

$$K\cdot W(P_{real}\|P_{fake}) = \sup_{\|f\|_L\le K} E_{x\sim P_r}[f(x)]-E_{x\sim P_\theta}[f(x)] \quad (5)$$

我们最终可用带参函数来估计上述方程的上界：

$$\max_{w\in W} E_{x\sim P_r}[f_w(x)]-E_{x\sim P_\theta}[f_w(x)] \le \sup_{\|f\|_L\le K} E_{x\sim P_r}[f(x)]-E_{x\sim P_\theta}[f(x)] \quad (6)$$

最终的生成器优化目标为：

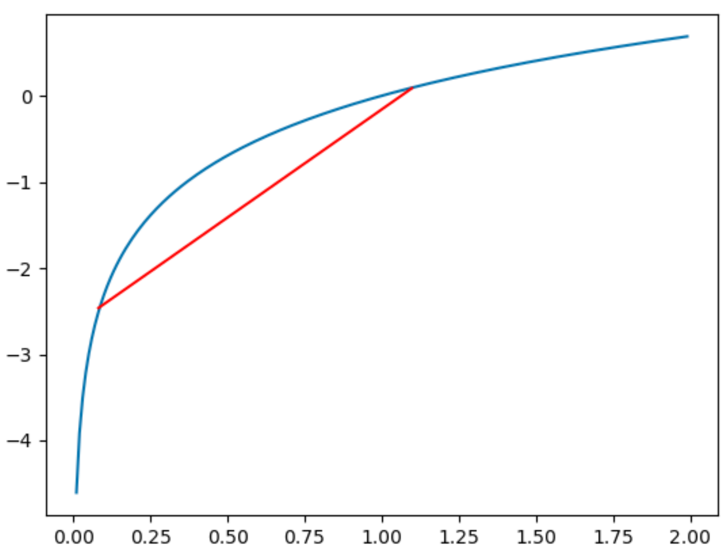
$$\max_{w\in W} E_{x\sim P_r}[f_w(x)]-E_{x\sim P_\theta}[f_w(x)] \quad (7)$$

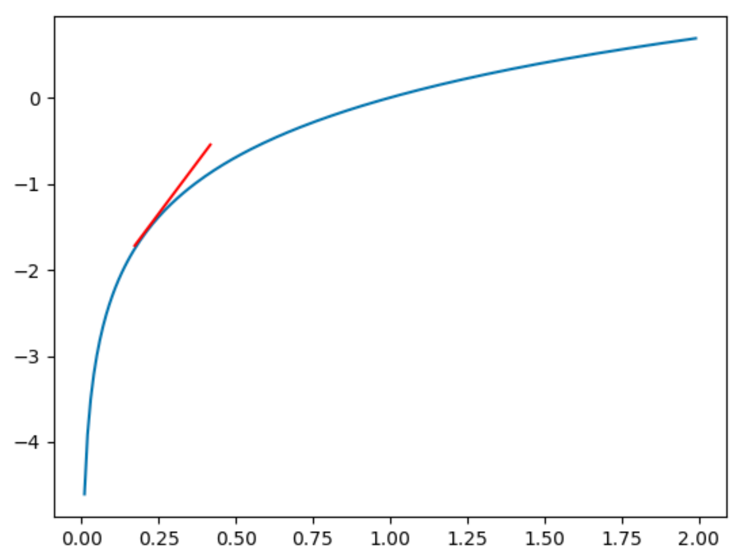
WGAN Lipschitz continuous

- Add weight clipping to $[-c, c]$

- Gradient Penalty term:

$$\lambda \underset{\tilde{x}\sim P_{\tilde{x}}}{E}\left[\left(\left\|\nabla_{\tilde{x}} D(\tilde{x})\right\|_2 - 1\right)^2\right] \tag{8}$$





The proposed deblur method

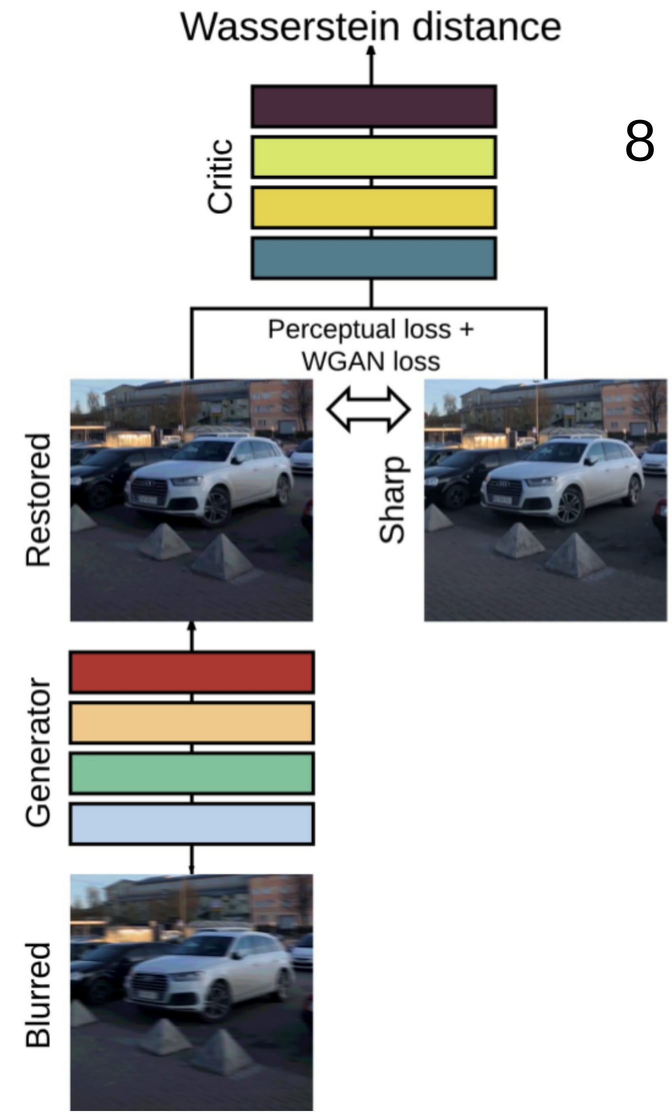
Loss Function

$$L = \underbrace{\underbrace{L_{GAN-GP}}_{\textit{adversarial loss}} + \underbrace{\lambda \cdot L_X}_{\textit{content loss}}}_{\textit{total loss}} \qquad (8)$$

- Adversarail Loss : We use WGAN-GP as the critic function, which is shown to be robust to the choice of generator architecture .
- Content Loss : Perceptual loss is a simple L2-loss, but based on the difference of the generated and target image CNN feature maps. It is defined as following:

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

- where $\phi_{i,j}$ is the feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer within the VGG19 network, pretrained on ImageNet, $W_{i,j}$ and $H_{i,j}$ are the dimensions of the feature maps. The perceptual
- loss focuses on restoring general content while adversarial loss focuses on restoring texture details.



8

Figure 4: DeblurGAN training. The generator network takes the blurred image as input and produces the estimate of the sharp image. The critic network takes the restored and sharp images and outputs a distance between them. The total loss consists of the WGAN loss from critic and the perceptual loss [17]. The perceptual loss is the difference between the VGG-19 [34] *conv3.3* feature maps of the sharp and restored images. At test time, only the generator is kept.

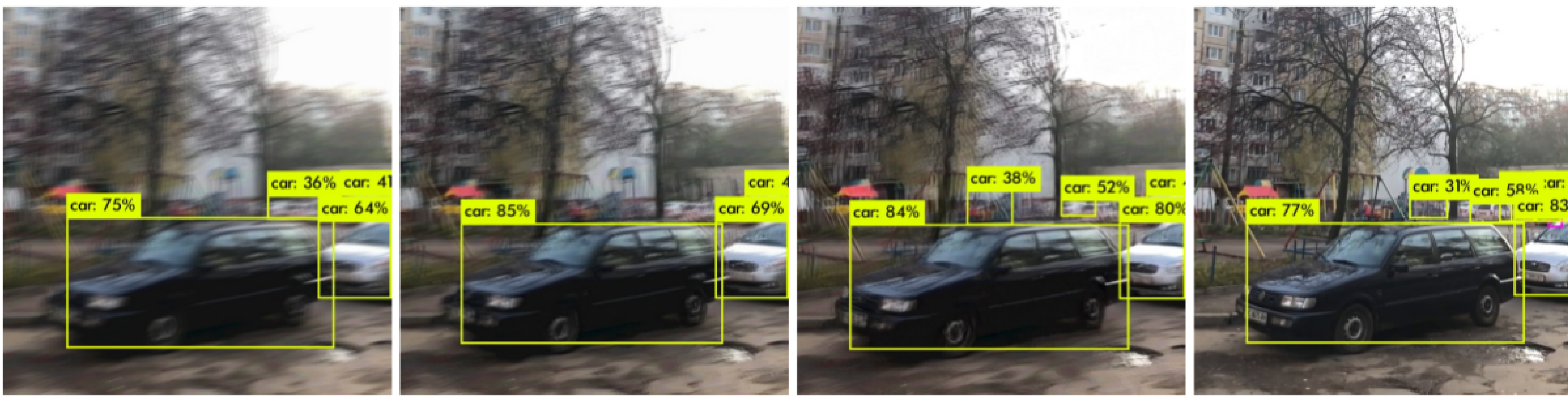
Experiment

PART

Two

Not Finish Running Yet!

empirically better results on validation. The training phase took 6 days for training one *DeblurGAN* network.



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

Figure 9: YOLO object detection before and after deblurring

Not Finish Running Yet!

```
model [ConditionalGANModel] was created
create web directory ./checkpoints/experiment_name/web...
#training images = 256
PSNR on Train = 9.132084
```

1/6 images

```
model [ConditionalGANModel] was created
create web directory ./checkpoints/experiment_name/we
#training images = 2302
PSNR on Train = 10.864766
```

Original images

```
saving the model at the end of epoch 60, iters 138120
End of epoch 60 / 300     Time Taken: 2771 sec
PSNR on Train = 15.865719
(epoch: 61, iters: 80, time: 1.545) G_GAN: -37.327 G_L1: 476.804 D_real+fake: -7.283
```

1/6 images

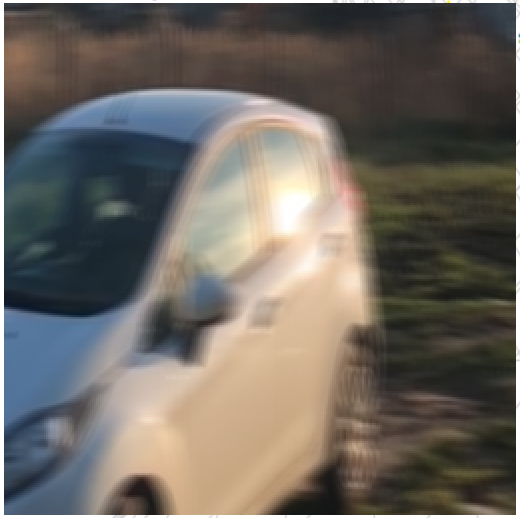
```
(epoch: 57, iters: 2288, time: 1.660) G_GAN: 30.020 (
20.825
End of epoch 57 / 300     Time Taken: 2974 sec
PSNR on Train = 13.776959
```

Original images

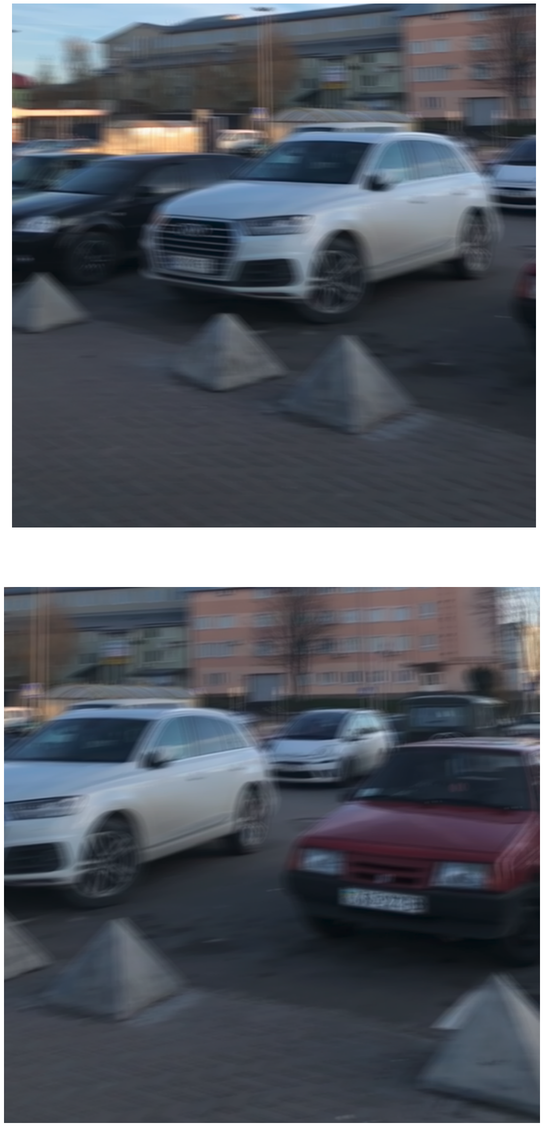
We have set the training images to 1/6 compared to the original images, but it doesn’t work


Original

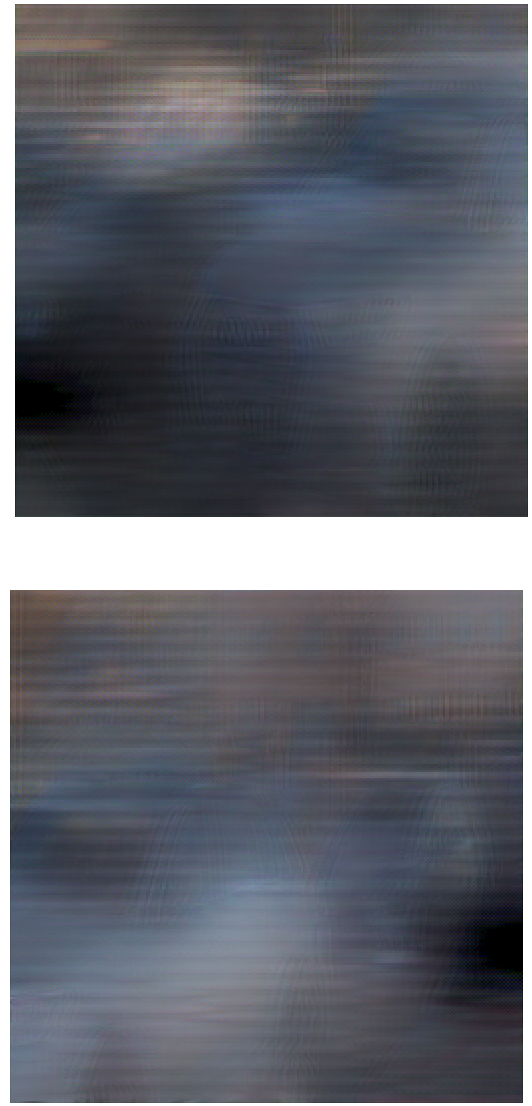

Generated

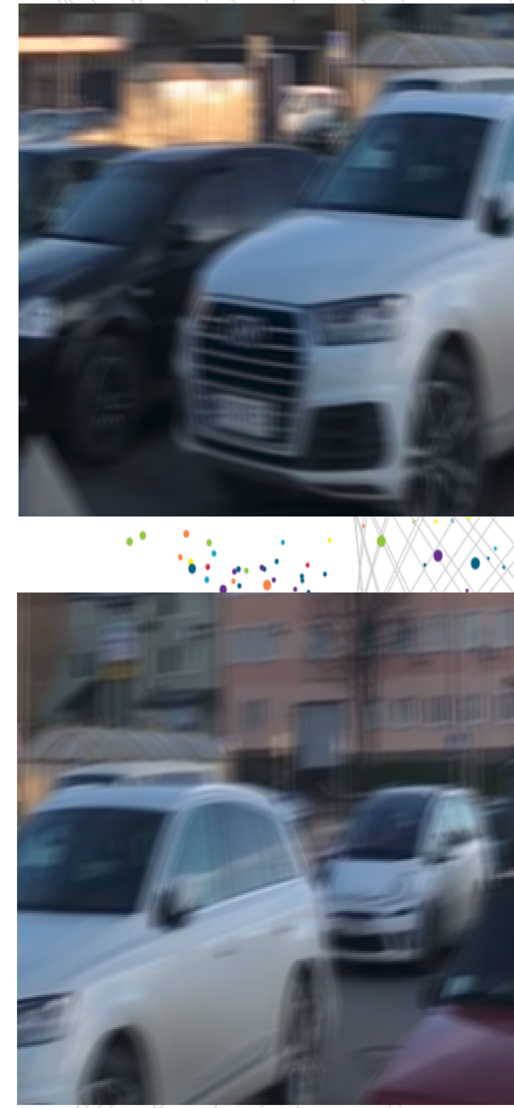
Real



Original



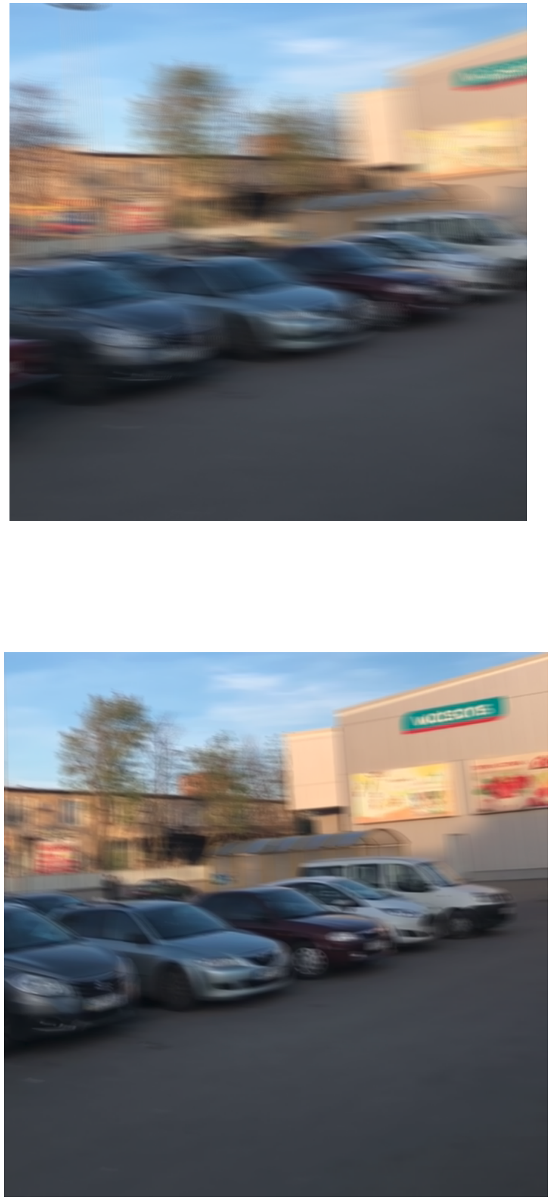
Generated



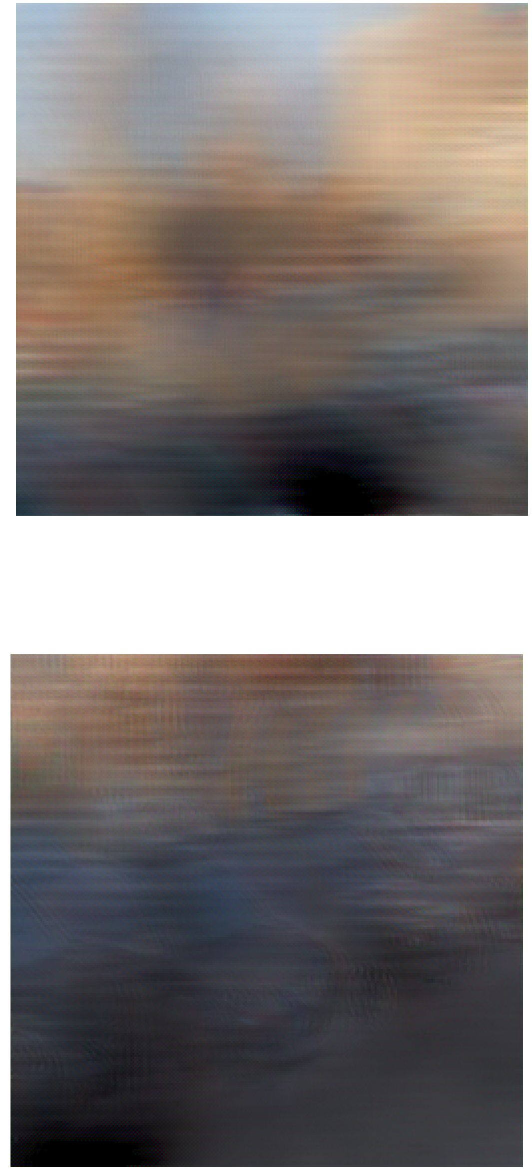
Real

Original **Generated** **Real**

Original Generated Real



Original



Generated

Real

THANK YOU FOR WATCHING

PRESENTED BY Xue Minglong