

PixelRL: Fully Convolutional Network with Reinforcement Learning for Image Processing

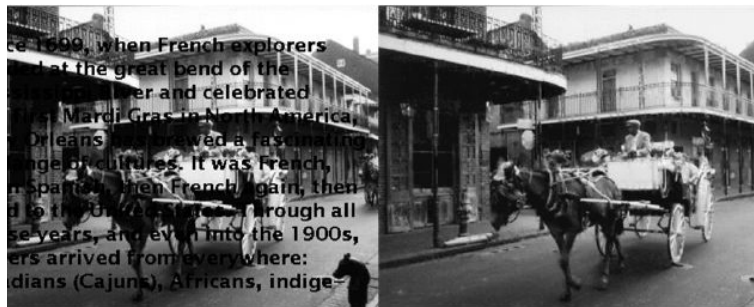
Image Processing

Computer vision focuses on making sense of what the machines see.

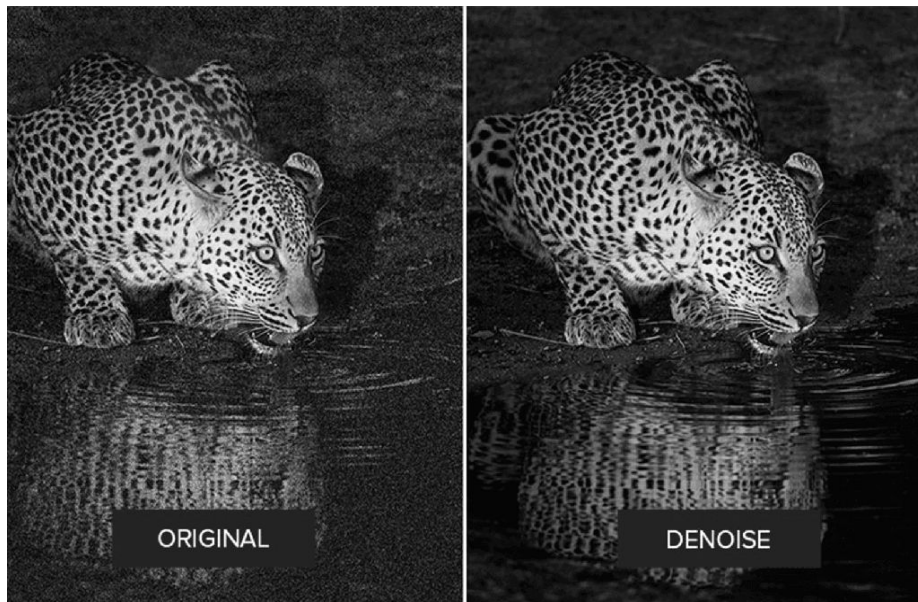
Image processing, on the other hand, transform images in many ways such as smoothing, filtering, enhancing, inpainting, blurring, etc..



color enhancing



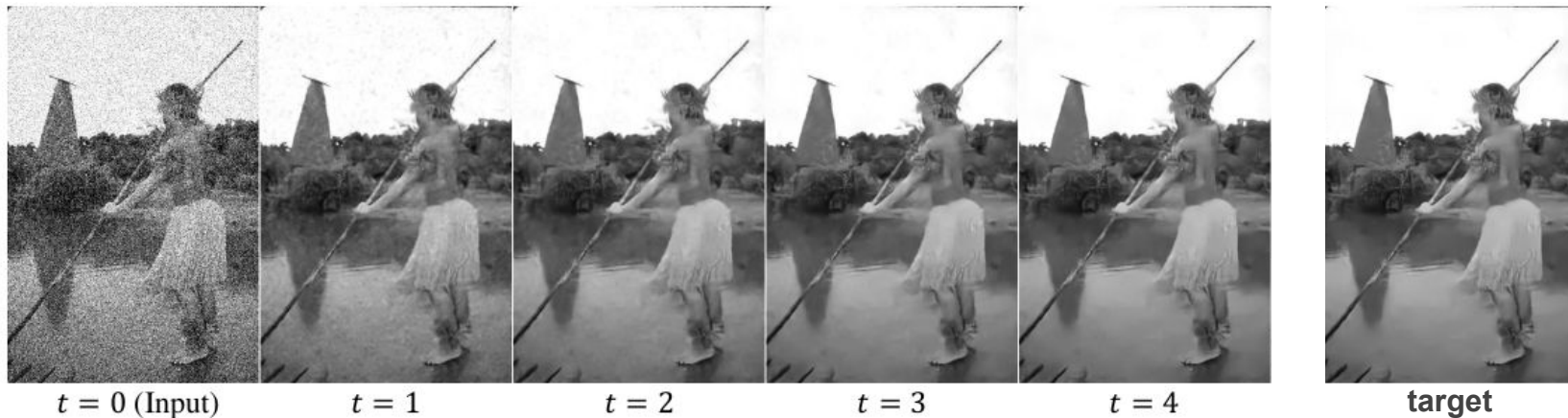
inpainting



denoising

Fully Convolutional Network with Reinforcement Learning

Denoising



$s^{(0)} = I$ is a input noisy image, the agents iteratively remove the noises by executing actions.

$$r_i^{(t)} = (I_i^{target} - s_i^{(t)})^2 - (I_i^{target} - s_i^{(t+1)})^2,$$

that maximize the mean of the total expected rewards at all pixels:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} E_{\pi} \left(\sum_{t=0}^{\infty} \gamma^t \bar{r}^{(t)} \right), \quad (5)$$

$$\bar{r}^{(t)} = \frac{1}{N} \sum_{i=1}^N r_i^{(t)}, \quad (6)$$

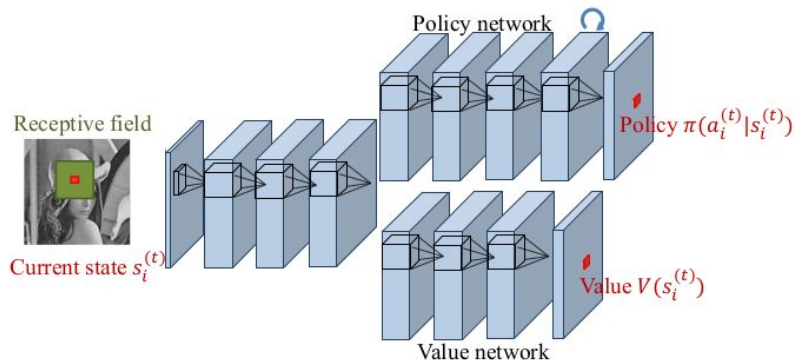
where $\bar{r}^{(t)}$ is the mean of the rewards $r_i^{(t)}$ at all pixels.

TABLE I: Actions for image denoising and restoration.

	action	filter size	parameter
1	box filter	5x5	-
2	bilateral filter	5x5	$\sigma_c = 1.0, \sigma_S = 5.0$
3	bilateral filter	5x5	$\sigma_c = 0.1, \sigma_S = 5.0$
4	median filter	5x5	-
5	Gaussian filter	5x5	$\sigma = 1.5$
6	Gaussian filter	5x5	$\sigma = 0.5$
7	pixel value += 1	-	-
8	pixel value -= 1	-	-
9	do nothing	-	-

Fully Convolutional Network with Reinforcement Learning

Architecture



Shared network				Policy network			
Conv +ReLU	Conv +ReLU	Conv +ReLU	Conv +ReLU	Conv +ReLU	Conv +ReLU	ConvGRU	Conv +Softmax
3x3, 1, 64	3x3, 2, 64	3x3, 3, 64	3x3, 4, 64	3x3, 3, 64	3x3, 2, 64	3x3, 1, 64	3x3, 1, A
				Value network			
				Conv +ReLU	Conv +ReLU	Conv	
				3x3, 3, 64	3x3, 2, 64	3x3, 1, 1	

Fig. 1: Network architecture of the fully convolutional A3C. The numbers in the table denote the filter size, dilation factor, and output channels, respectively.

Objective

$$r_i^{(t)} = (I_i^{target} - s_i^{(t)})^2 - (I_i^{target} - s_i^{(t+1)})^2,$$

that maximize the mean of the total expected rewards at all pixels:

$$\pi^* = \operatorname{argmax}_{\pi} E_{\pi} \left(\sum_{t=0}^{\infty} \gamma^t \bar{r}^{(t)} \right), \quad (5)$$

$$\bar{r}^{(t)} = \frac{1}{N} \sum_{i=1}^N r_i^{(t)}, \quad (6)$$

where $\bar{r}^{(t)}$ is the mean of the rewards $r_i^{(t)}$ at all pixels.

A3C

$$R_i^{(t)} = r_i^{(t)} + \gamma V(s_i^{(t+1)}),$$

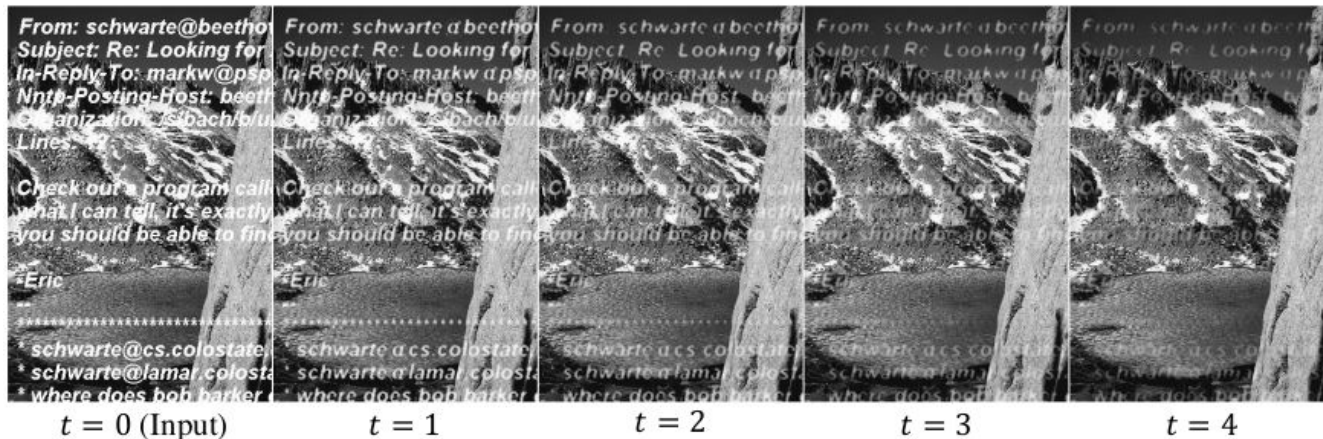
$$d\theta_v = \nabla_{\theta_v} \frac{1}{N} \sum_{i=1}^N \left(R_i^{(t)} - V(s_i^{(t)}) \right)^2,$$

Improving: Reward Map Convolution

$$R_i^{(t)} = r_i^{(t)} + \gamma \sum_{j \in \mathcal{N}(i)} w_{i-j} V(s_j^{(t+1)}),$$

Fully Convolutional Network with Reinforcement Learning

Inpainting



target

Color Enhancing



$t = 0$ (Input)

$t = 1$

$t = 2$

$t = 3$

target

Datasets

BSD68. contains **493** images (428 train and 65 test)



dl-image-enhance. contains **112** images (67 train and 45 test)



Evaluation

Color Enhancing

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2.$$

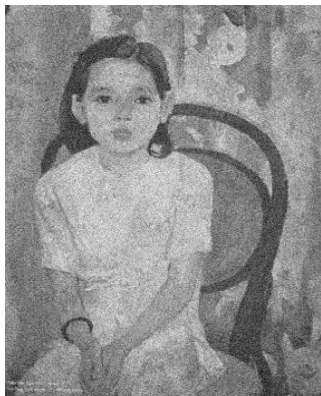
The PSNR (in dB) is defined as

Denoising and Inpainting

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Demo & Fail cases

Denoising



Demo & Fail cases

Inpainting

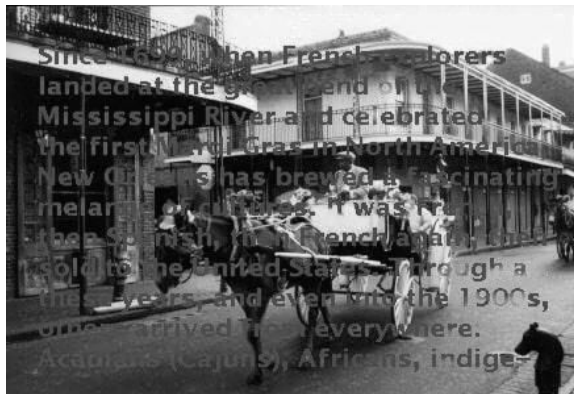
In *global optimization*, the true global solution of the optimized is found; the compromise is efficiency. The worst-case complexity of optimization methods grows exponentially with the problem size, so that in practice, for the particular problem instances encountered so far, this favorable situation does occur, but for very large problems, with a few tens of variables, can take a very long time (days) to solve.

Global optimization is used for problems with a small number of variables, where the value of finding the true global optimum is very high. One example is a engineering design is *worst-case analysis* of a high value or safety critical system. Here the variables are parameters, that can be manufacturing, or with the engineering conditions. The objective function is a utility function, i.e. smaller values are better than larger values, and the constraints are knowledge of the possible parameter values. The optimization problem of finding the *worst-case* values for the parameters, i.e. value is acceptable, so can certify the system as safe or reliable (the parameter relations).

A local optimization method can rapidly find a set of parameters, but it is guaranteed to be the absolute worst possible. If the method finds parameter values that yield unacceptable performance, it can determine that the system is not reliable. But a local method cannot certify the system as reliable; it can only fail to find values. A global optimization method, in contrast, will find the

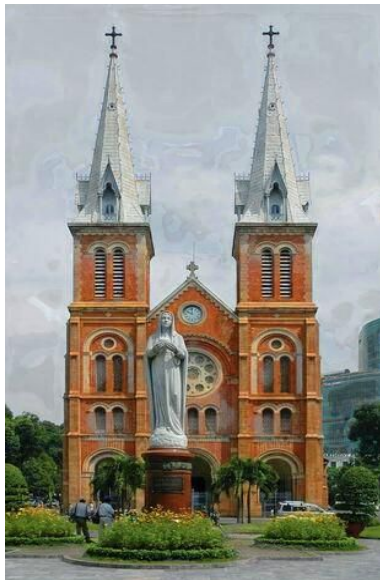
Since 1699, when French explorers landed at the great bend of the Mississippi River and celebrated the first Mardi Gras in North America, New Orleans has brewed a fascinating melange of cultures. It was French, then Spanish, then French again, then sold to the United States. Through all these years, and even into the 1900s, others arrived from everywhere: Acadians (Cajuns), Africans, indige-

Since 1699, when French explorers landed at the great bend of the Mississippi River and celebrated the first Mardi Gras in North America, New Orleans has brewed a fascinating melange of cultures. It was French, then Spanish, then French again, then sold to the United States. Through a these years, and even into the 1900s, others arrived from everywhere. Acadians (Cajuns), Africans, indige-



Demo & Fail cases

Color Enhancing



References

PixelRL: Fully Convolutional Network with Reinforcement Learning for Image Processing