# Crafting a Toolchain for Image Restoration by Deep Reinforcement Learning: Supplementary Material

Ke Yu [1] Chao Dong[2] Liang Lin[2,3] Chen Change Loy[1]
[1]CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong
[2]SenseTime Research [3]Sun Yat-sen University
{yk017, ccloy}@ie.cuhk.edu.hk {dongchao, linliang}@sensetime.com

## 1. Framework Details

**Tools Architecture.** As shown in Figure 1, the architectures of two kinds of tools are presented. A convolutional layer, denoted as *Conv*($k$,$n$,$s$), has $n$ filters of kernel size $k \times k$ and $s$ represents the stride of convolution. The 3-layer CNN is similar to an SRCNN [1], except that we shrink the filter number and apply residual learning as in [3]. The 8-layer CNN contains 2 residual blocks [2] in the middle, as depicted in Figure 1, and its parameter number is about 2.3 times as that of the 3-layer tool.

**Agent Architecture.** As can be observed in Figure 2. The agent consists of 3 modules, named feature extractor, one-hot encoder and LSTM, respectively. The feature extractor, containing 4 convolutional layers and 1 fully-connected layer, takes the current distorted image $\mathbf{I}_t$ as input and it outputs a 32 dimensional feature. The one-hot encoder generates an $N$ dimensional encoded vector from previous value vector $\tilde{\boldsymbol{v}}_t$ with the last dimension discarded because the stopping action could not be chosen at the previous step. The outputs of first two modules are concatenated as the input of LSTM. The LSTM has a hidden size of 50, following which a fully-connected layer finally generates the $N+1$ dimensional value vector $\boldsymbol{v}_t$.

**Complexity.** The complexity of each module in *RL-Restore* is presented in Table 1. It is observed that the agent has more parameters than a tool, however, the computational complexity of the agent is much smaller than that of a tool because the convolutional layers in the agent have a stride of 2, leading to smaller features and cheaper computational cost. In Table 1, 'Theoretical' means that each tool is equally chosen without considering the stopping action. 'Practical' represents the real case while testing on the moderate set, where the average chosen ratios of the 3-layer CNN, 8-layer CNN and stopping action are 27.2%, 53.7% and 19.1%, respectively. Although the agent prefers to choose complex tools, the practical complexity is still smaller than that in theory thanks to the stopping action.

## 2. Additional Qualitative Evaluation

**Details of Test Sets.** The test sets are divided into 3 groups – mild, moderate and severe. As shown in Table 2, we divide each type of distortion into 10 levels according to the degradation degree. A distorted image in the test set consists of all three types of distortions – Gaussian blur, Gaussian noise and JPEG compression. The level of such a mixed distortion is measured by adding up the level of each individual distortion. For example, the level of a mixed of Gaussian blur $\sigma = 1.7$, Gaussian noise $\sigma = 11$ and JPEG compression $Q = 16$ is the sum of each individual distortion level, *i.e.*, $4 + 3 + 9 = 16$. As the level of each distortion ranges from 1 to 10, the level of a mixed distortion is between 3 and 30. Table 3 shows that a mixed distortion is categorized into different groups according to its level. In the aforementioned example, the mixed distortion with level 16 belongs to the moderate group. As discussed in our paper, training data only involve moderate distortions while testing is conducted on mild, moderate and severe datasets.

**More Qualitative Results.** Additional visual results on different test sets are presented in Figure 3, 4 and 5. The right two columns in each figure are failure cases. As can be observed in all figures, our results generally have sharper edges and fewer artifacts compared with VDSR baselines. However, there are still some failure cases, which are caused by improper toolchains selected by the agent. For example, in Figure 3, the agent decides to stop after only one step, leaving some noise or blurring not addressed, while in Figure 4 and 5, improper toolchains enlarge the noise and thus lead to unsatisfactory artifacts, especially along edges.

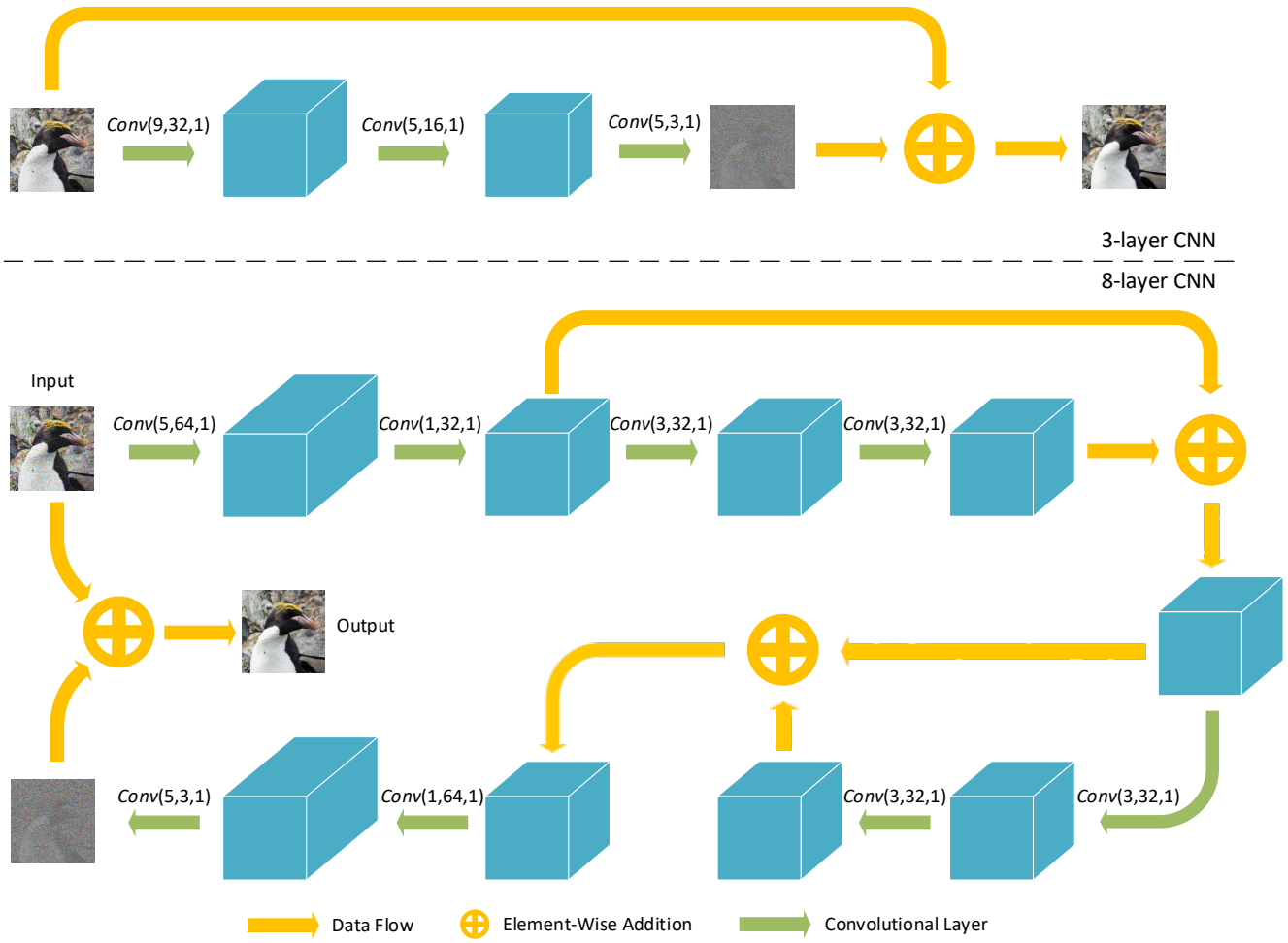


Figure 1. Network architecture of the tools.

Table 1. The complexity of each module in *RL-Restore* as well as VDSR baselines.

| Model | Component | Module | Parameters($\times 10^5$) | Computations($\times 10^9$) |
|---|---|---|---|---|
| *RL-Restore* | Agent | Feature Extractor | 0.681 | 0.0140 |
| | | One-Hot Encoder | N/A | N/A |
| | | LSTM | 0.195 | 0.0000195 |
| | Tools | 3-Layer CNN | 0.218 | 0.0864 |
| | | 8-Layer CNN | 0.506 | 0.201 |
| | | Theoretical | 0.362 | 0.144 |
| | | Practical | 0.331 | 0.131 |
| | Total | Theoretical | 1.96 | 0.474 |
| | | Practical | 1.87 | 0.436 |
| VDSR-s | N/A | N/A | 2.09 | 0.828 |
| VDSR | N/A | N/A | 6.67 | 2.65 |

Table 2. Degradation levels of individual distortion.

| Degradation Level | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Gaussian Blur ($\sigma$) | [0, 0.5] | [0.5, 1] | [1, 1.5] | [1.5, 2] | [2, 2.5] | [2.5, 3] | [3, 3.5] | [3.5, 4] | [4, 4.5] | [4.5, 5] |
| Gaussian Noise ($\sigma$) | [0, 5] | [5, 10] | [10, 15] | [15, 20] | [20, 25] | [25, 30] | [30, 35] | [35, 40] | [40, 45] | [45, 50] |
| JPEG (Q) | [80, 100] | [60, 80] | [50, 60] | [40, 50] | [35, 40] | [30, 35] | [25, 30] | [20, 25] | [15, 20] | [10, 15] |

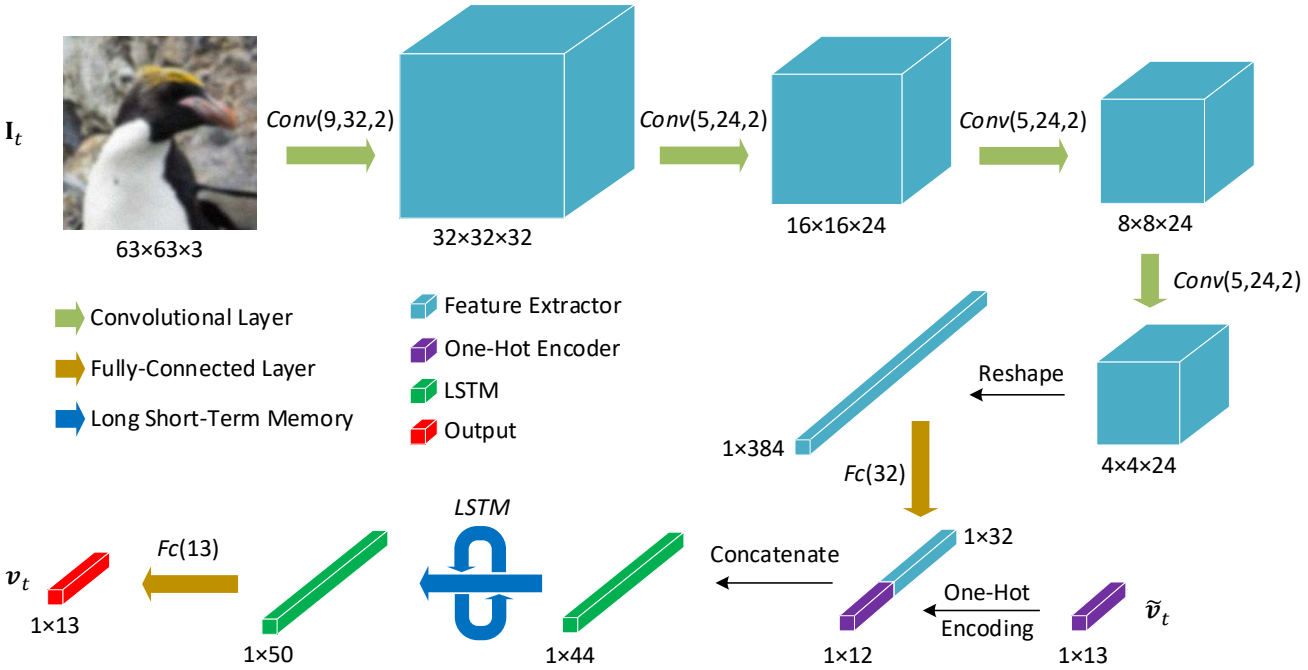


Figure 2. Network architecture of the agent.

Table 3. Categories of mixed distortions.

| Distortion Category | Extremely Mild | Mild | Moderate | Severe | Extremely Severe |
|---|---|---|---|---|---|
| Sum of Individual Level | [3, 10] | [11, 13] | [14, 19] | [20, 22] | [23, 30] |

## References


[1] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2016. 1

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[3] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1

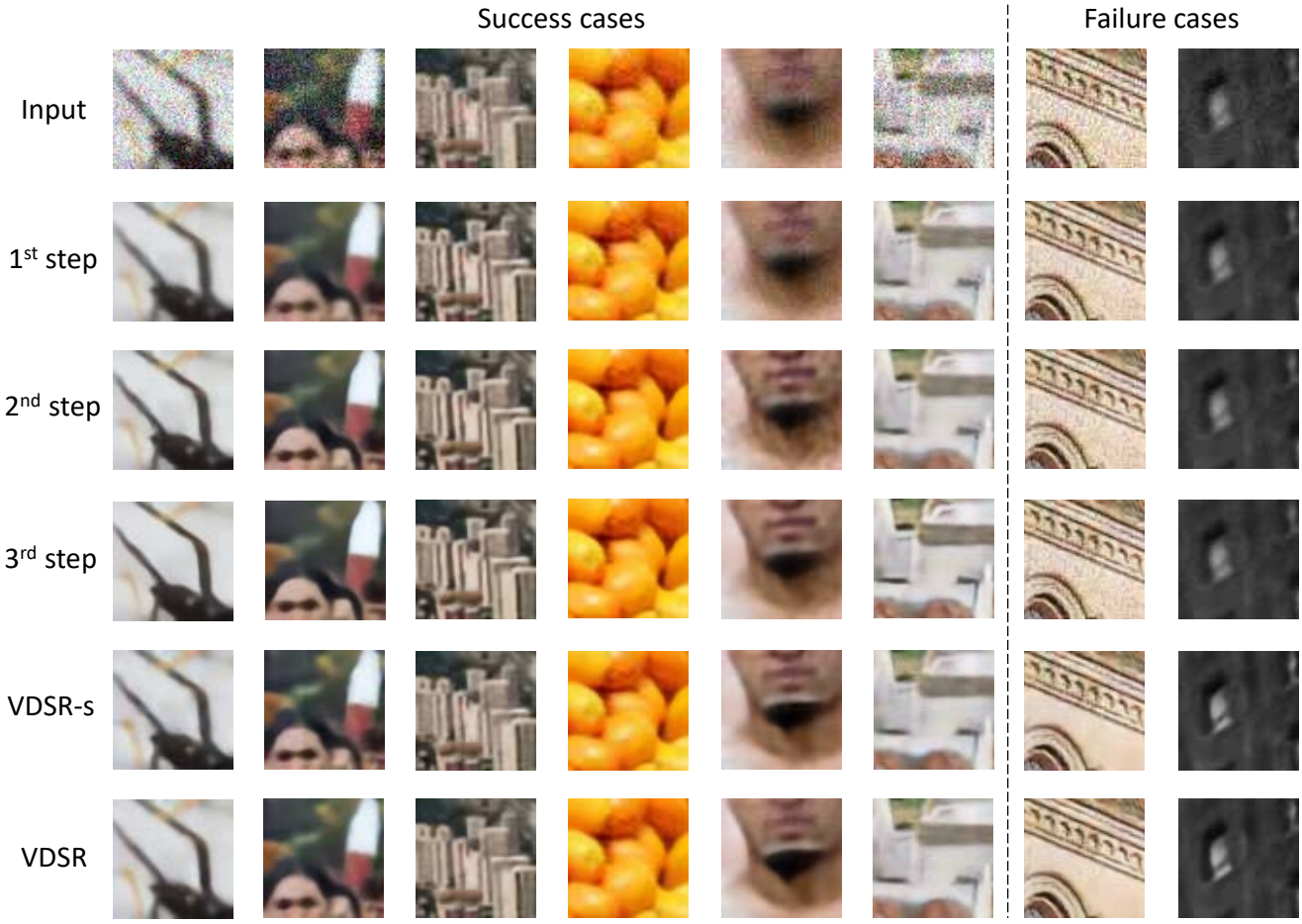

Figure 3. Qualitative results on mild set.

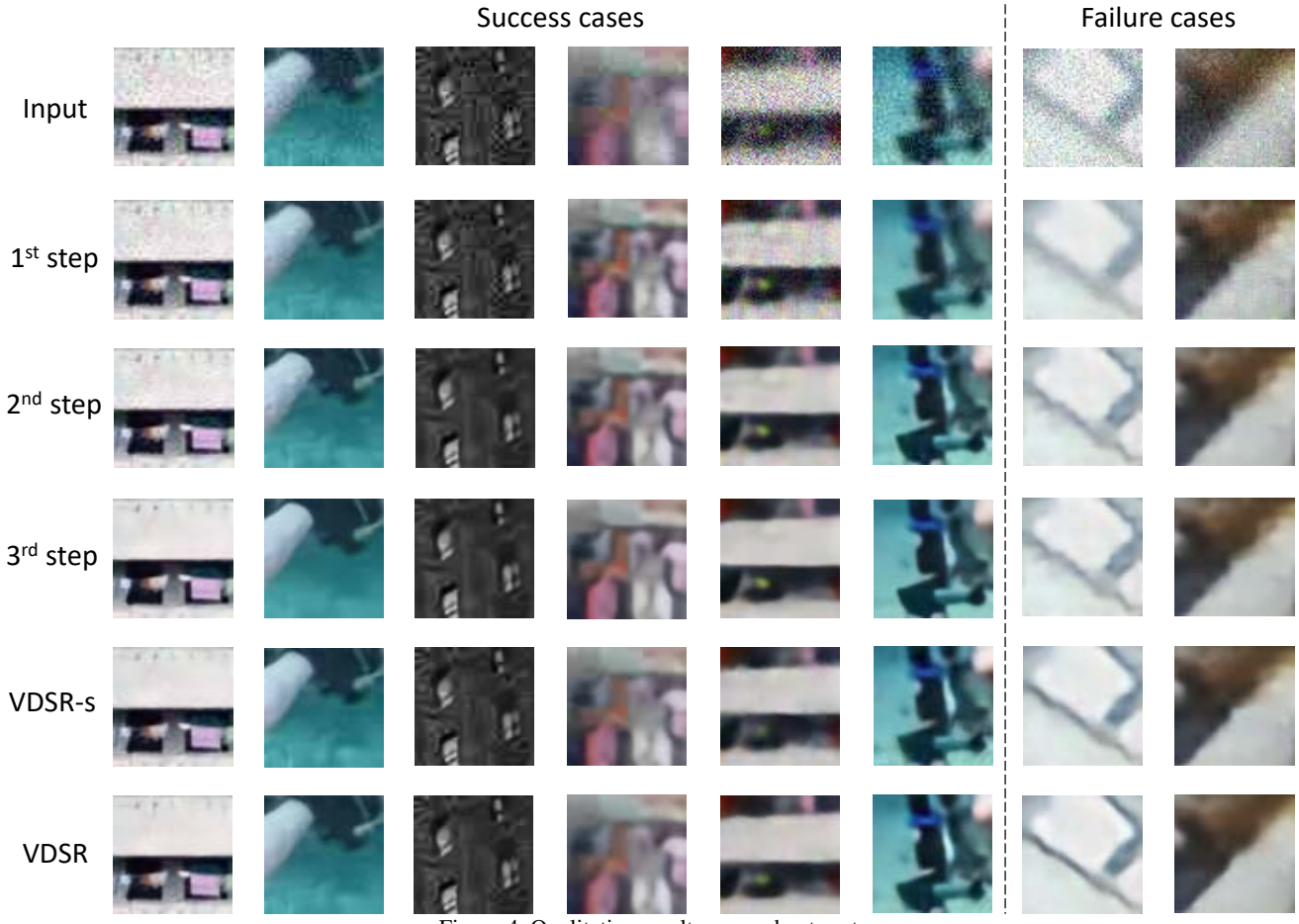

Figure 4. Qualitative results on moderate set.

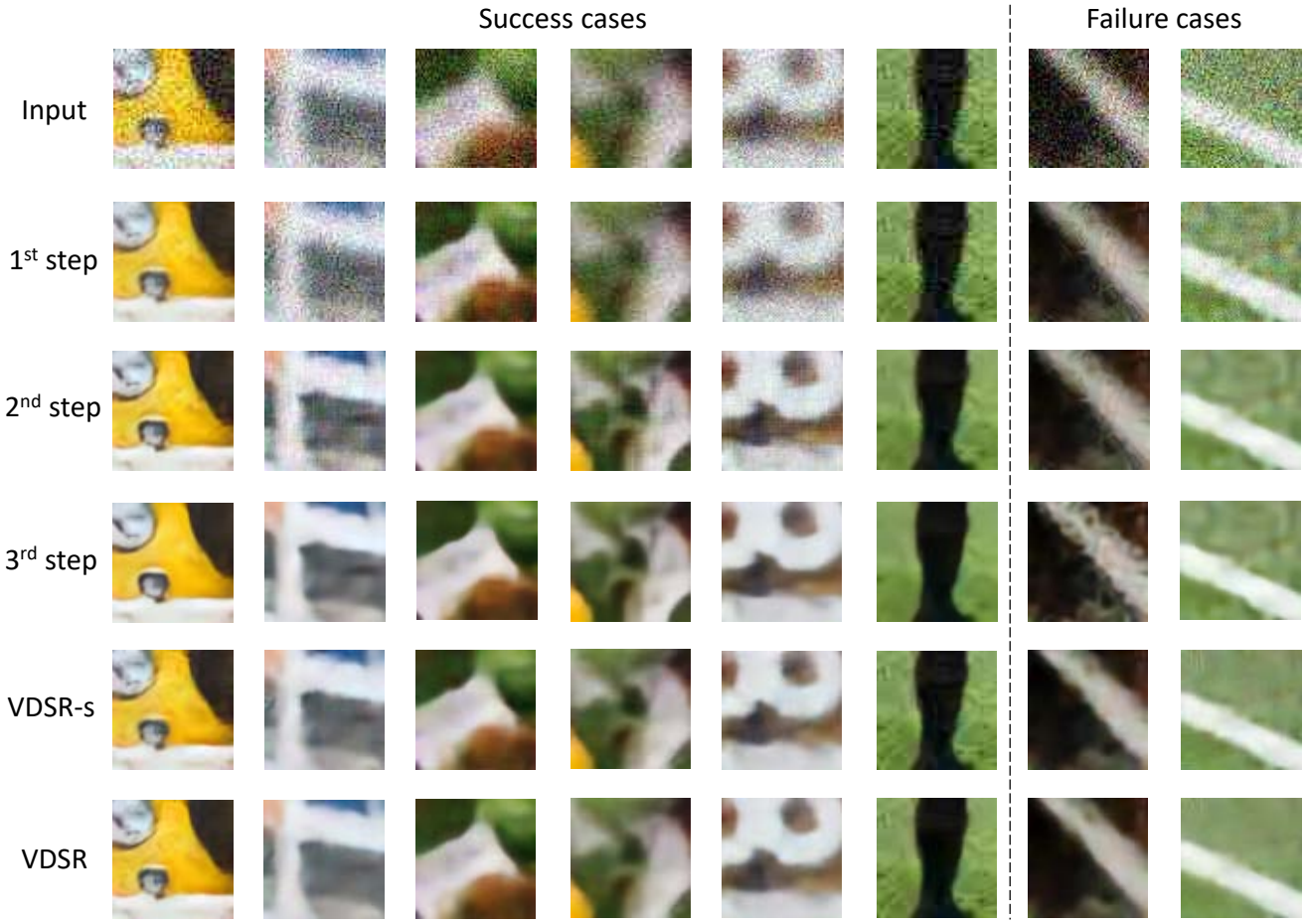

Figure 5. Qualitative results on severe set.