Attention-guided Image Compression by Deep Reconstruction of Compressive Sensed Saliency Skeleton

Supplementary Material

1. Least-norm solution

In Section 3.4, we hope to make the minimum adjustment to the output image I_g (or to the critical pixel set $\mathbf{c}_g$) so that the adjusted image can satisfy the CS constraint. This forms the following optimization problem:

$$\text{minimize} \quad ||\delta|| \tag{1}$$
$$\text{subject to} \quad H \cdot (\mathbf{c}_g + \delta) = \mathbf{y} \tag{2}$$

In Eq. 2, H is the CS sampling matrix and $\mathbf{c}_g$ is the critical pixel set in the restored image I_g, so the above formulation can be rewritten as:

$$\text{minimize} \quad ||\delta|| \tag{3}$$
$$\text{subject to} \quad H \cdot \delta = \mathbf{y} - H \cdot \mathbf{c}_g \tag{4}$$

Let $\hat{\mathbf{y}} = \mathbf{y} - H \cdot \mathbf{c}_g$, we get:

$$\text{minimize} \quad \delta^T \delta \tag{5}$$
$$\text{subject to} \quad H \cdot \delta - \hat{\mathbf{y}} = 0 \tag{6}$$

The above optimization problem can be solved by method of Lagrange multipliers. By introducing Lagrange multiplier λ, we can get the Lagrangian function:

$$L(\delta, \lambda) = \delta^T \delta + \lambda^T (H \cdot \delta - \hat{\mathbf{y}}) \tag{7}$$

The optimality conditions are

$$\nabla_\delta L = 2\delta + H^T \lambda = 0 \tag{8}$$
$$\nabla_\lambda L = H \cdot \delta - \hat{\mathbf{y}} = 0 \tag{9}$$

From first condition, we can get:

$$\delta = -\frac{1}{2} H^T \lambda \tag{10}$$

Substitute it into the second condition to get:

$$-\frac{1}{2} H H^T \lambda - \hat{\mathbf{y}} = 0 \tag{11}$$

Because CS sampling matrxi H is a full row rank,fat matrix, we can get:

$$\lambda = -2(HH^T)^{-1} \hat{\mathbf{y}} \tag{12}$$

By Substituting λ into Eq. 10, we get the final solution:

$$\delta_* = H^T (HH^T)^{-1} \cdot \hat{\mathbf{y}} \tag{13}$$
$$= H^T (HH^T)^{-1} \cdot (\mathbf{y} - H \cdot \mathbf{c}_g) \tag{14}$$

2. Auto-encoder architecture

We use the same U-Net like architecture for both pixel domain and DCT domain auto-encoders in the restoration network $\mathcal{G}$. As shown in Fig. 1, the auto-encoder network contains one input convolutional layer, one output convolutional layer and 16 residual blocks. In the encoder stage, input image is convolved by the input convolutional layer, and then pass through four basic size-invariant reisual blocks. Next, two downsampling residual blocks are adopted to downsample the feature size to eliminate redundant information. In the decoder stage, two upsampling residual blocks are adopted to recover the spatial size of feature maps to the original level. Finally, four basic size-invariant residual blocks and one output convolutional layer are adopted to produce the restored image.

3. CS sampling matrix

AGDL compression system performs Compressed sensing on the critical pixel set $\mathbf{c}$ with a full row rank, fat CS sampling matrix H (far fewer rows than columns):

$$\mathbf{y} = H \cdot \mathbf{c} \tag{15}$$

where $H \in \mathcal{R}^{m \times n}$, $\mathbf{c} \in \mathcal{R}^{n \times 1}$, $\mathbf{y} \in \mathcal{R}^{m \times 1}$, n is the length of the critical pixel vector, m is the length of the CS measurements, $m \ll n$.

We choose the random Gaussian matrix to be the CS sampling(measurement) matrix. The entries of a Gaussian matrix are independent and follow a normal distribution with expectation $\mu = 0$ and variance σ^2. The probability density function of a normal distribution is:

$$f(x/\mu, \sigma^2) = \frac{1}{\sqrt{(2\sigma^2 \pi)}} exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{16}$$

The variance σ^2 is set to $\frac{1}{m}$, that is to say, each entry of the CS sampling matrxi H, denoted by $H_{i,j}$, is independently sampled from the Gaussian distribution $\mathcal{N}(0, \frac{1}{m})$, where m is the length of the CS measurements.

How to determine the length of the critical pixel vector n and the length of the CS measurements m are discussed in the next section.

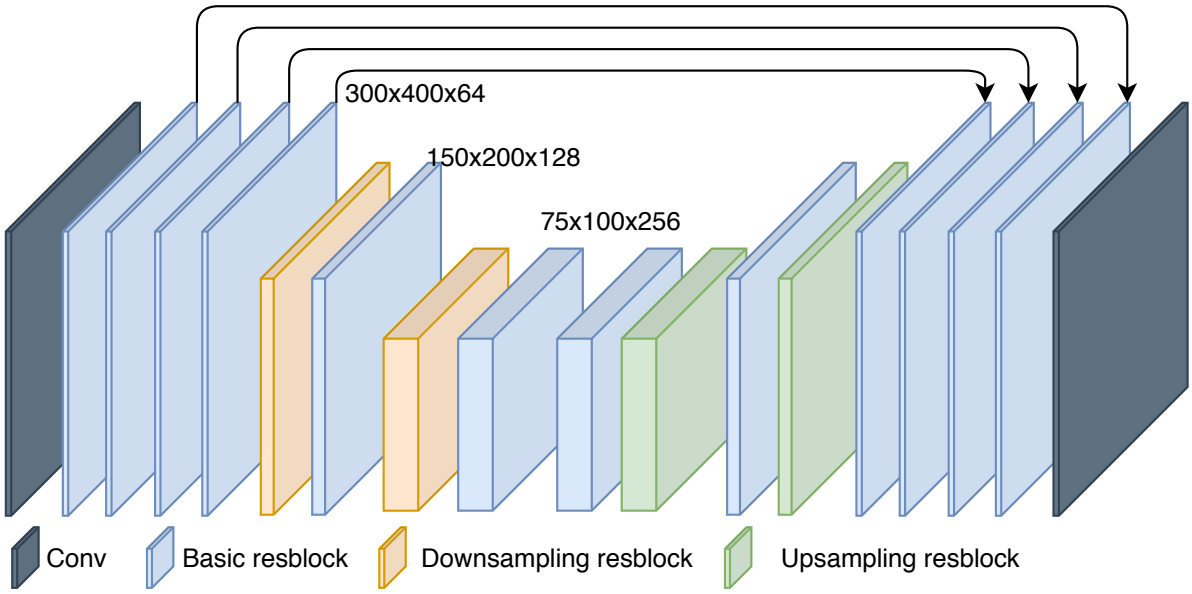

Figure 1: Network architecture for both pixel domain and DCT domain auto-encoders.

4. Ablation study

In AGDL, there are two parameters need to be predefined: the length of critical pixel vector n (or the size of the critical pixel set) and the length of the CS measurements m. In our experiemnts, We tried different combinations of n and m to find which one is the best. A total of four combinations are tested:

- $n = 1200, m = 240$
- $n = 1200, m = 480$
- $n = 2400, m = 480$
- $n = 2400, m = 960$

It is noteworthy that increasing n and m will sample more details, but it will also increase the total bit rate. The ROI RD curves of different combinations of n and m on DUTS-TE dataset are shown in Fig. 2. We can see that when the bit rate is low, $n = 1200, m = 240$ is the winner among all four combinations in rate-distortion performance. But when the bit rate is higher than 0.6bpp, the combination $n = 2400, m = 960$ is the best choice.

In the comparison experiments (both quantitative and qualitative results), we choose $n = 1200, m = 240$ as the default combination to train the AGDL system.

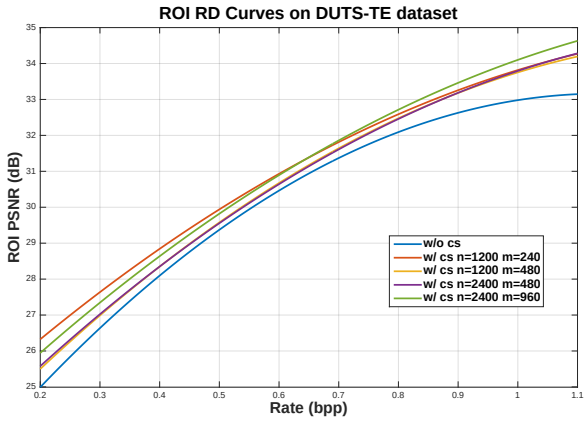

Figure 2: ROI RD curves of different combinations of n and m on DUTS-TE dataset.

5. Complete visual comparisons

The complete visual comparisons of all competing methods (J2K ROI, MWCNN, IDCN, DMCNN, QGAC) are shown in this section.

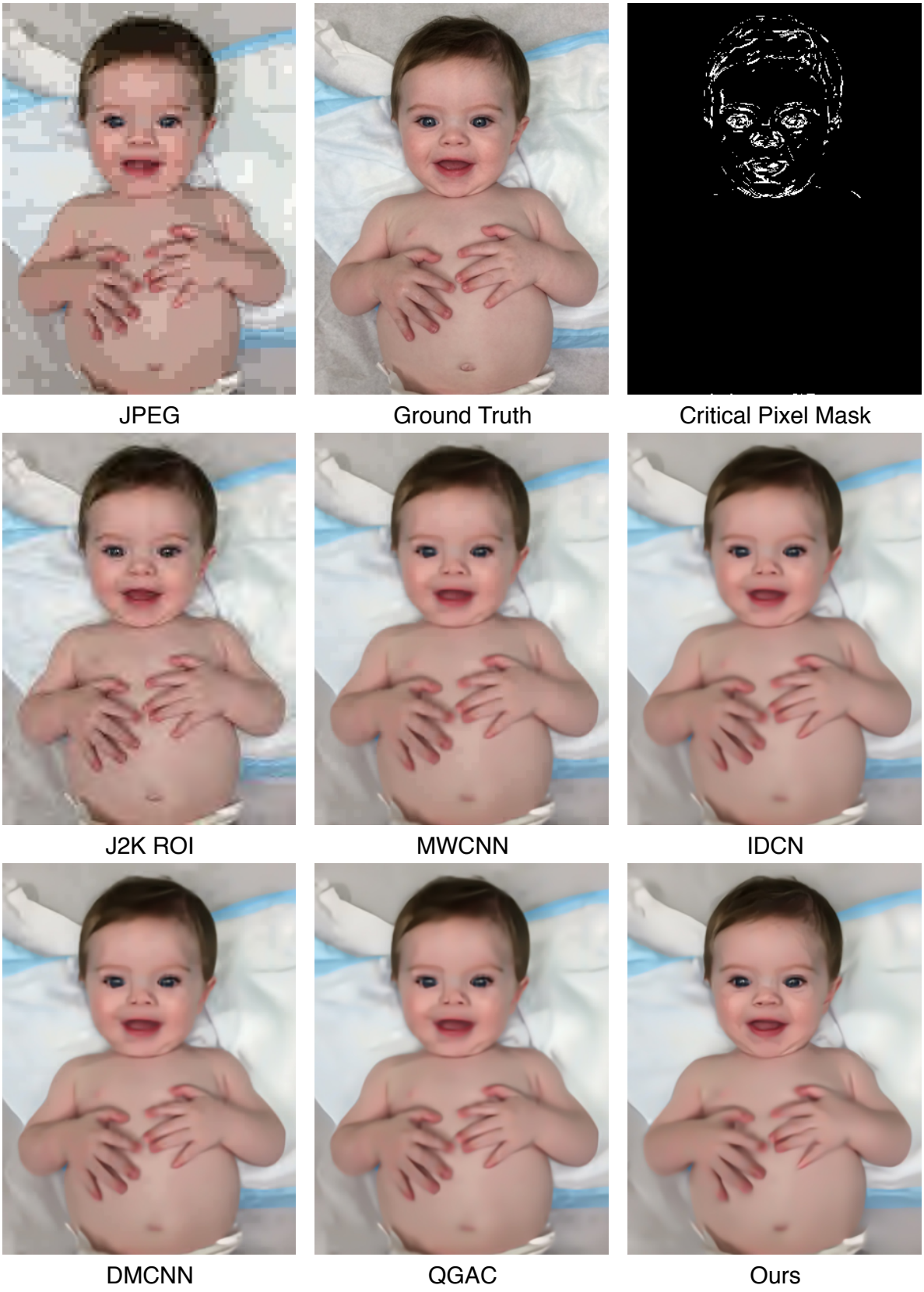

Figure 3: Visual comparisons of different methods on portraits in the same bit rate.

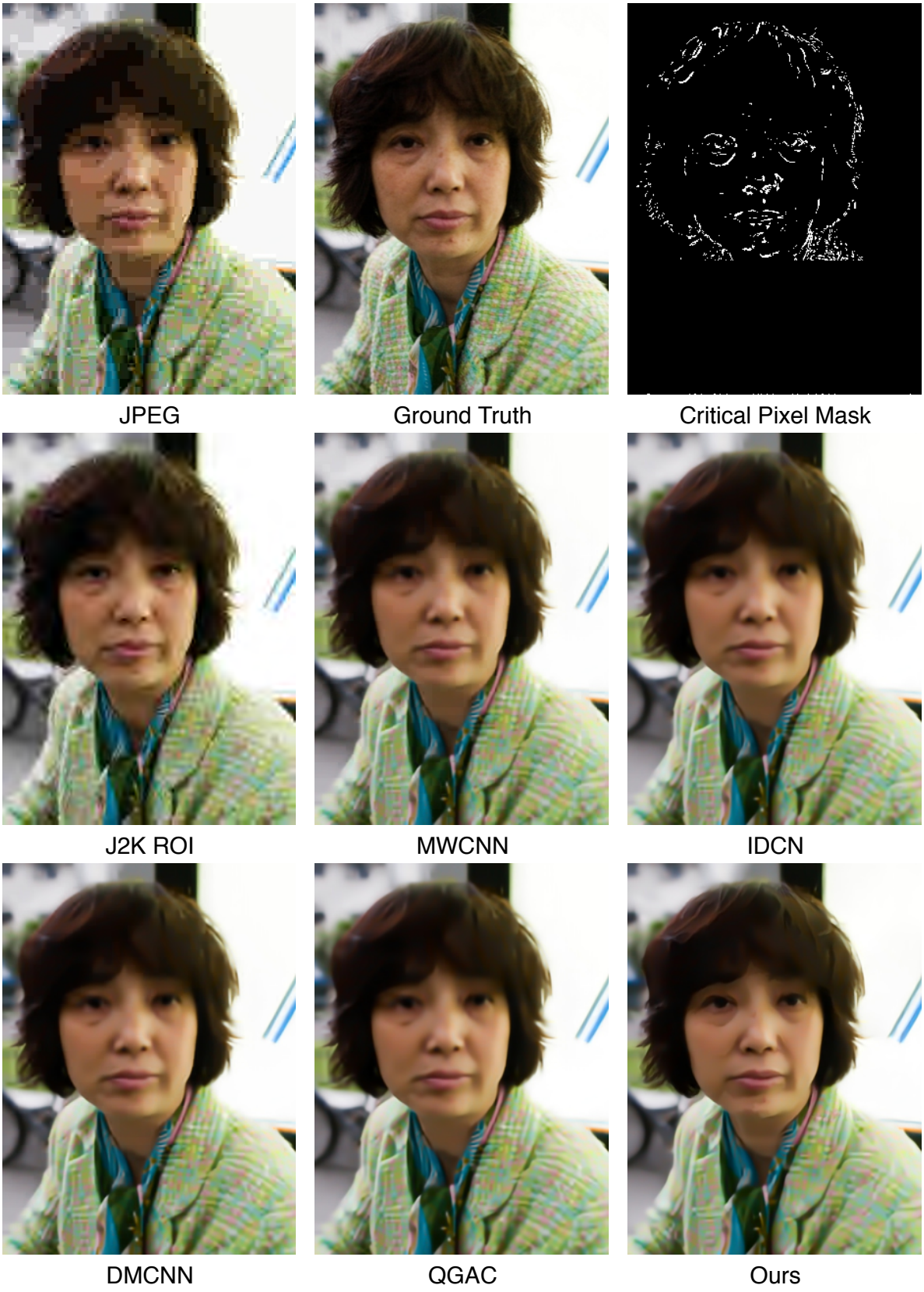

Figure 4: Visual comparisons of different methods on portraits in the same bit rate.

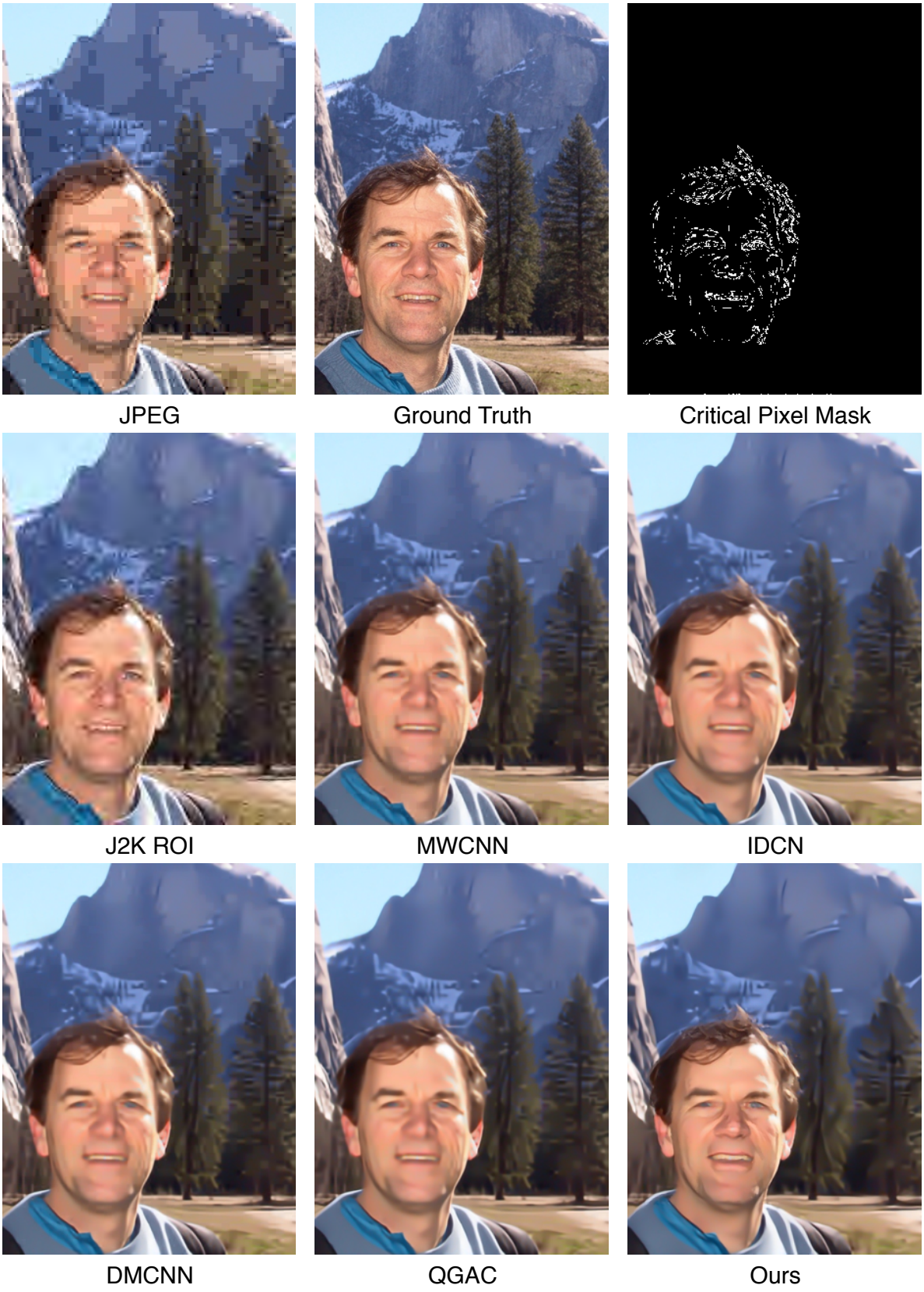

Figure 5: Visual comparisons of different methods on portraits in the same bit rate.

Figure 6: Visual comparisons of different methods on portraits in the same bit rate.

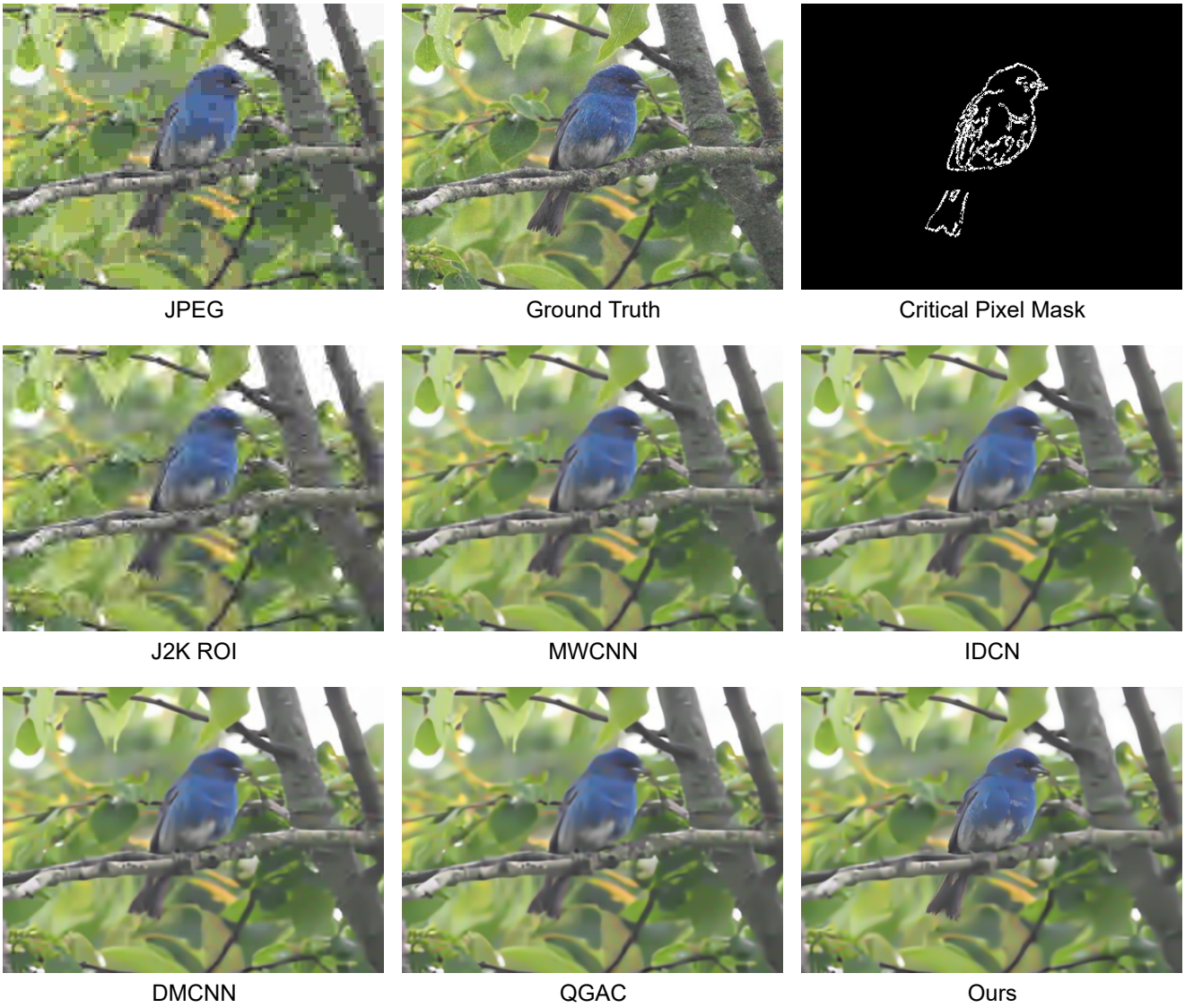

Figure 7: Visual comparisons of different methods on general objects in the same bit rate.

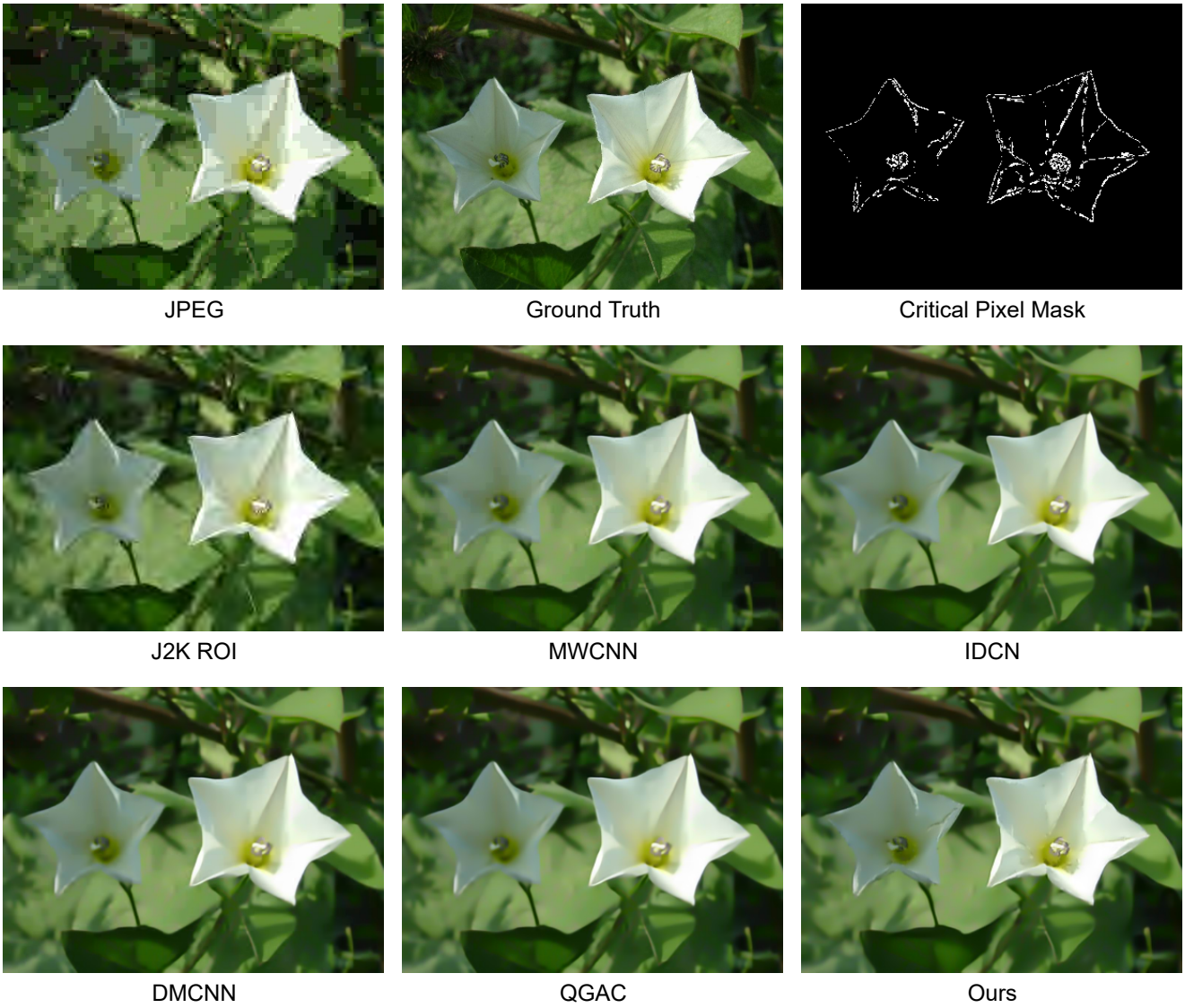

Figure 8: Visual comparisons of different methods on general objects in the same bit rate.

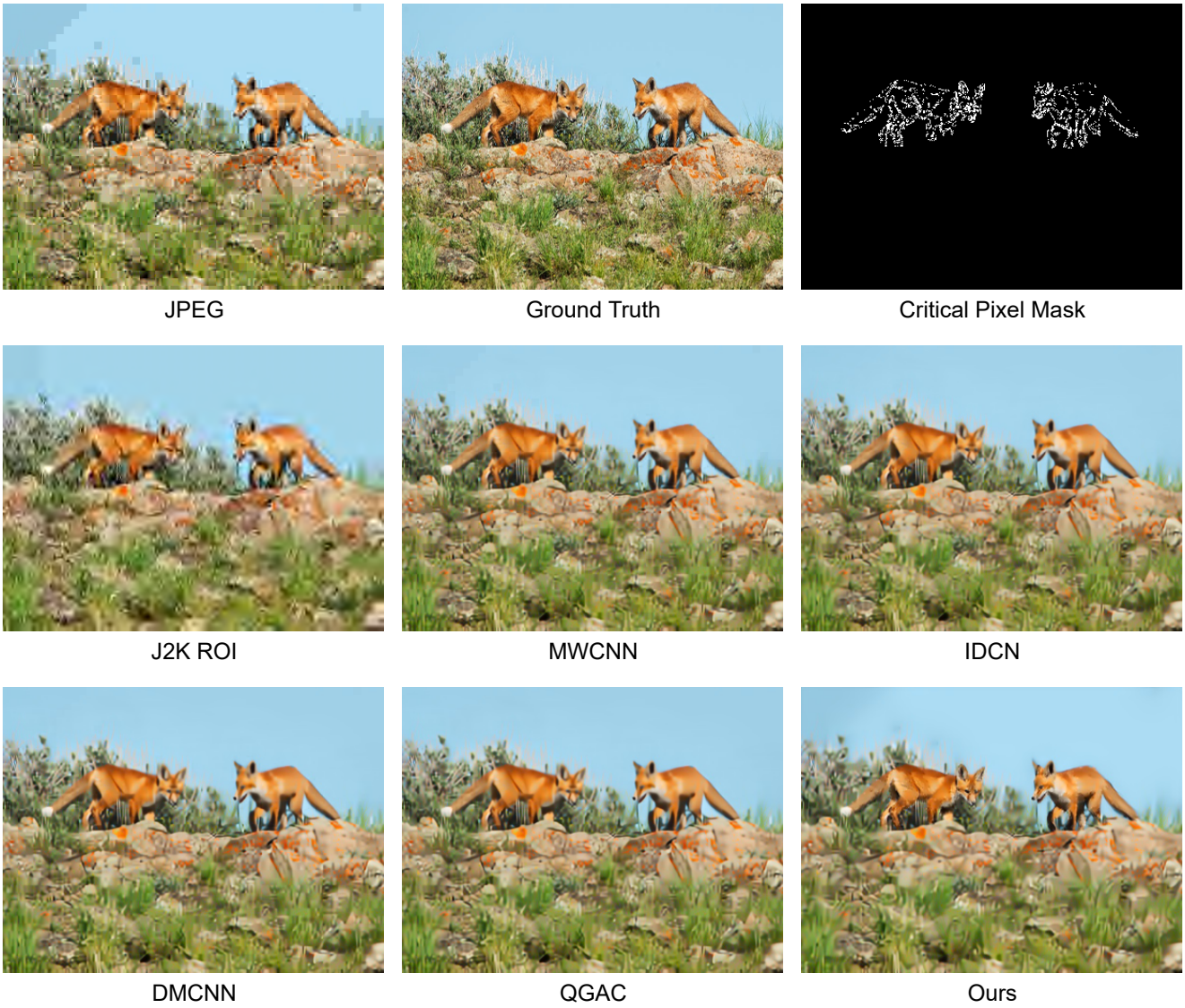

Figure 9: Visual comparisons of different methods on general objects in the same bit rate.

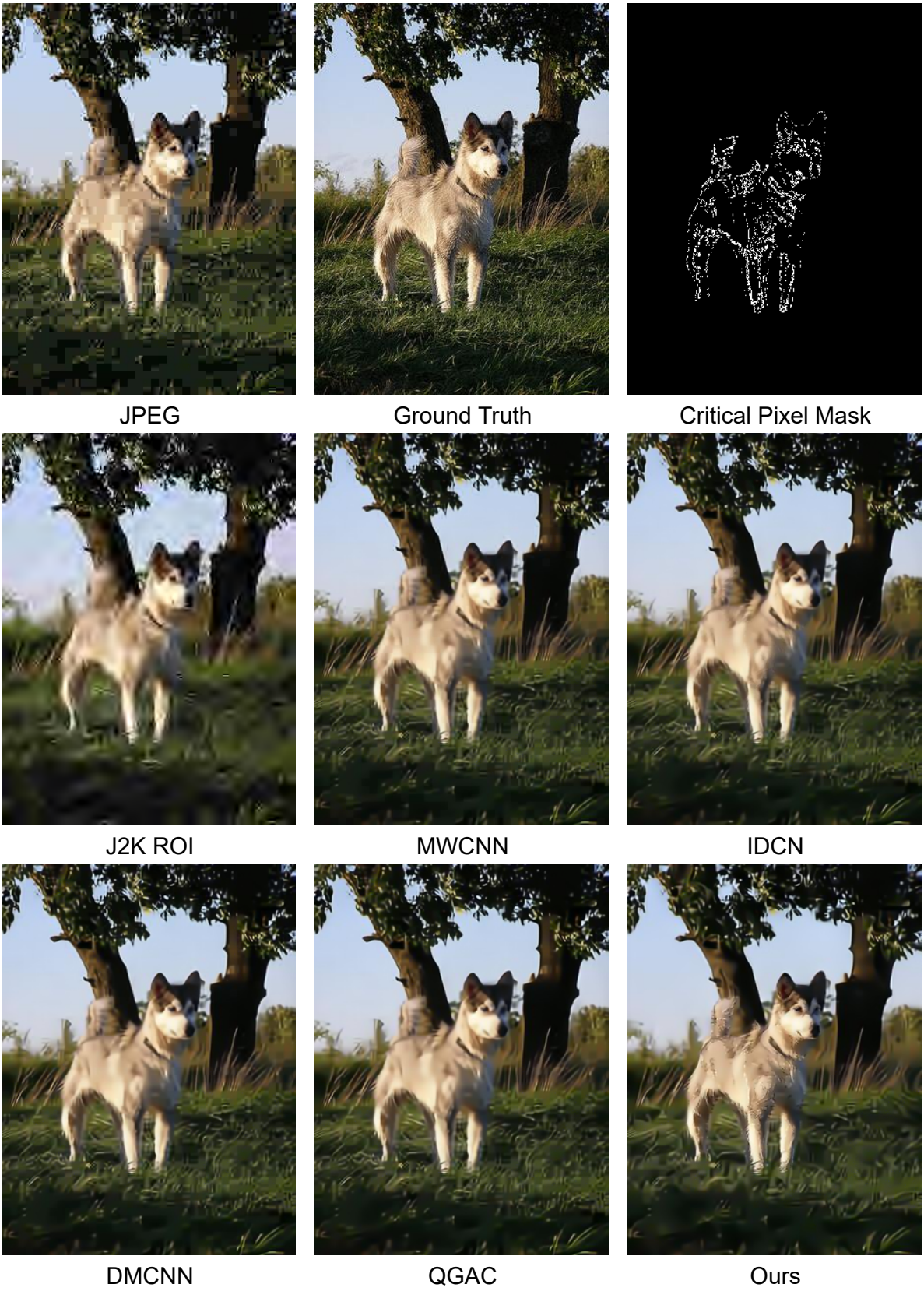

Figure 10: Visual comparisons of different methods on general objects in the same bit rate.

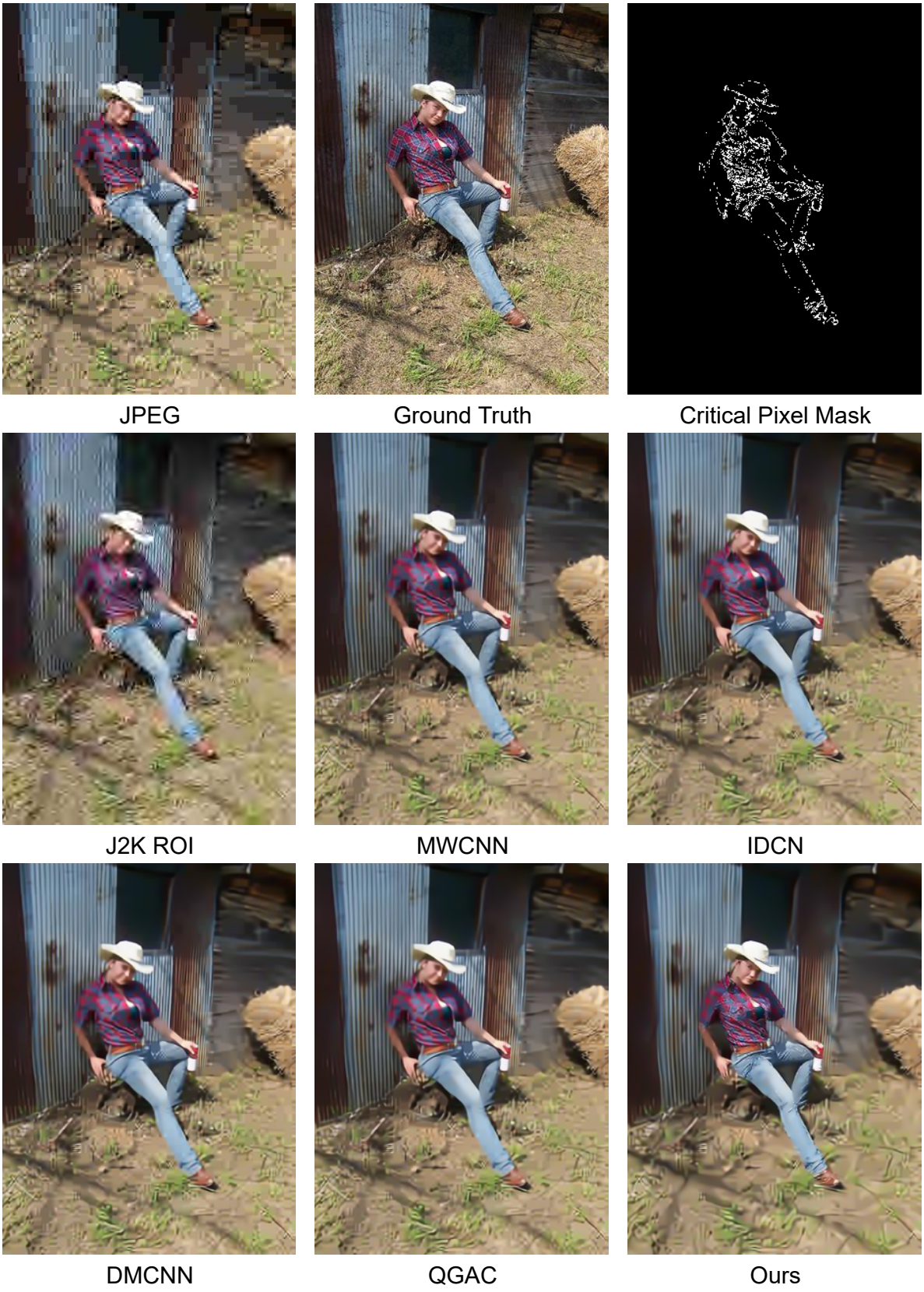

Figure 11: Visual comparisons of different methods on general objects in the same bit rate.