Learning Representations for Automatic Colorization
Supplementary Material

Gustav Larsson[1], Michael Maire[2], and Gregory Shakhnarovich[2]

[1]University of Chicago [2]Toyota Technological Institute at Chicago
larsson@cs.uchicago.edu, {mmaire,greg}@ttic.edu

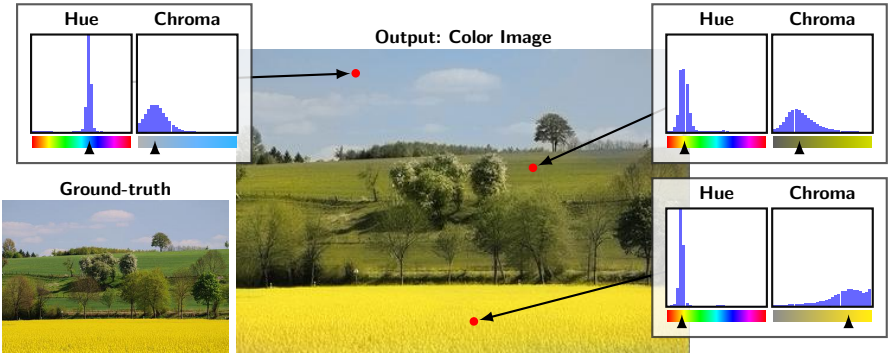


Fig. 1: **Histogram predictions.** Example of predicted hue/chroma histograms.

Supplementary Section 1 provides additional training and evaluation details. This is followed by more results and examples in Supplementary Section 2.

1 Supplementary details

1.1 Re-balancing

To adjust the scale of the activations of layer l by factor m, without changing any other layer's activation, the weights $\mathbf{W}$ and the bias $\mathbf{b}$ are updated according to:

$$\mathbf{W}_l \leftarrow m\mathbf{W}_l \qquad \mathbf{b}_l \leftarrow m\mathbf{b}_l \qquad \mathbf{W}_{l+1} \leftarrow \frac{1}{m}\mathbf{W}_{l+1} \tag{1}$$

The activation of $\mathbf{x}_{l+1}$ becomes:

$$\mathbf{x}_{l+1} = \frac{1}{m}\mathbf{W}_{l+1}\mathrm{ReLU}(m\mathbf{W}_l\mathbf{x}_l + m\mathbf{b}_l) + \mathbf{b}_{l+1} \tag{2}$$

The m inside the ReLU will not affect whether or not a value is rectified, so the two cases remain the same: (1) negative: the activation will be the corresponding feature in $\mathbf{b}_{l+1}$ regardless of m, and (2) positive: the ReLU becomes the identity function and m and $\frac{1}{m}$ cancel to get back the original activation.

We set $m = \frac{1}{\sqrt{\hat{\mathbb{E}}[X^2]}}$, estimated for each layer separately.

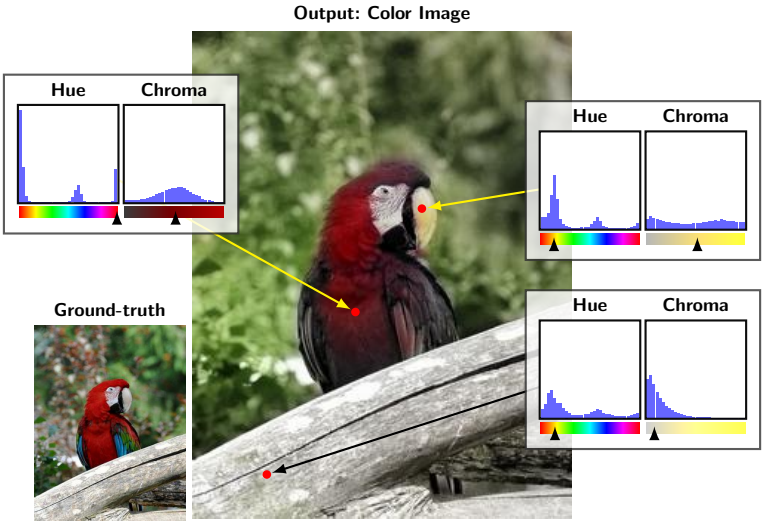


Fig. 2: **Histogram predictions.** Example of predicted hue/chroma histograms.

1.2 Color space $\alpha\beta$

The color channels $\alpha\beta$ ("ab" in [2]) are calculated as

$$\alpha = \frac{B - \frac{1}{2}(R+G)}{L+\epsilon} \qquad \beta = \frac{R-G}{L+\epsilon} \tag{3}$$

where $\epsilon = 0.0001$, $R, G, B \in [0,1]$ and $L = \frac{R+G+B}{3}$.[1]

1.3 Error metrics

For M images, each image m with N_m pixels, we calculate the error metrics as:

$$\text{RMSE} = \frac{1}{\sum_{m=1}^{M} N_m} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \sqrt{\left\| \left[\mathbf{y}_{\alpha\beta}^{(m)}\right]_n - \left[\hat{\mathbf{y}}_{\alpha\beta}^{(m)}\right]_n \right\|^2} \tag{4}$$

$$\text{PSNR} = \frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{N_m} -10 \cdot \log_{10} \left(\frac{\|\mathbf{y}_{\text{RGB}}^{(m)} - \hat{\mathbf{y}}_{\text{RGB}}^{(m)}\|^2}{3N_m} \right) \tag{5}$$

Where $\mathbf{y}_{\alpha\beta}^{(m)} \in [-3,3]^{N_m \times 2}$ and $\mathbf{y}_{\text{RGB}}^{(m)} \in [0,1]^{N_m \times 3}$ for all m.

[1] We know that this is how Deshpande *et al.* [2] calculate it based on their code release.

Hue	Chroma	CF	RMSE	PSNR
Sample	Sample		0.426	21.41
Mode	Mode		0.304	23.90
Expectation	Expectation		0.374	23.13
Expectation	Expectation	✓	0.307	24.35
Expectation	Median		0.342	23.77
Expectation	Median	✓	**0.299**	**24.45**

Table 1: **ImageNet/cval1k.** Comparison of various histogram inference methods for hue/chroma. Mode/mode does fairly well but has severe visual artifacts. (CF = Chromatic fading)

1.4 Lightness correction

Ideally the lightness L is an unaltered pass-through channel. However, due to subtle differences in how L is defined, it is possible that the lightness of the predicted image, $\hat{L}$, does not agree with the input, L. To compensate for this, we add $L-\hat{L}$ to all color channels in the predicted RGB image as a final corrective step.

2 Supplementary results

2.1 Validation

A more detailed list of validation results for hue/chroma inference methods is seen in Table 1.

2.2 Examples

We provide additional samples for global biasing (Figure 3) and SUN-6 (Figure 4). Comparisons with Charpiat *et al.* [1] appear in Figures 5 and 6. Examples of how our algorithm can bring old photographs to life in Figure 7. More examples on ImageNet (ctest10k) in Figures 8 to 11 and Figure 12 (failure cases). Examples of histogram predictions in Figures 1 and 2.

References

1. Charpiat, G., Bezrukov, I., Altun, Y., Hofmann, M., Schölkopf, B.: Machine learning methods for automatic image colorization. In: Computational Photography: Methods and Applications. CRC Press (2010)
2. Deshpande, A., Rock, J., Forsyth, D.: Learning large-scale automatic image colorization. In: ICCV (2015)
3. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. ACM Transactions on Graphics (TOG) 21(3) (2002)

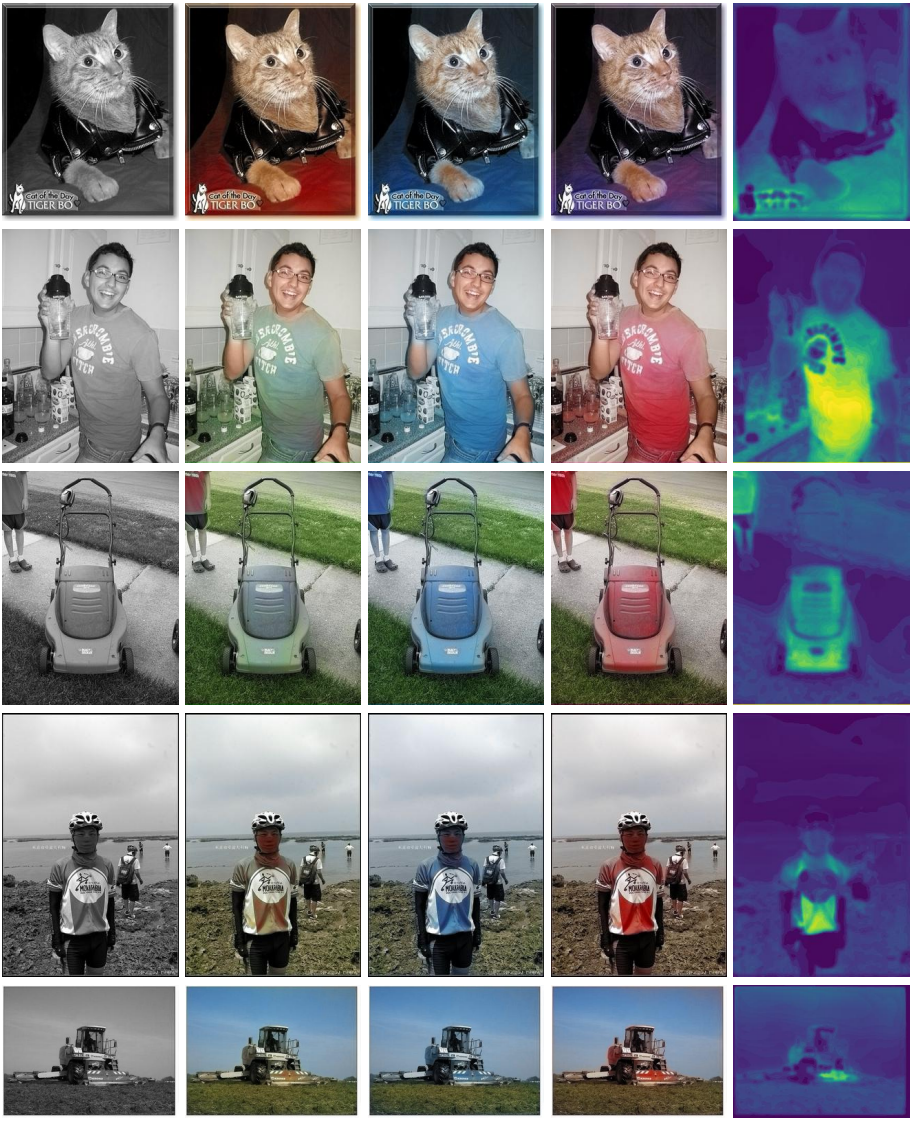


Fig. 3: **Sampling multiple colorizations.** From left: graylevel input; three colorizations sampled from our model; color uncertainty map according to our model.

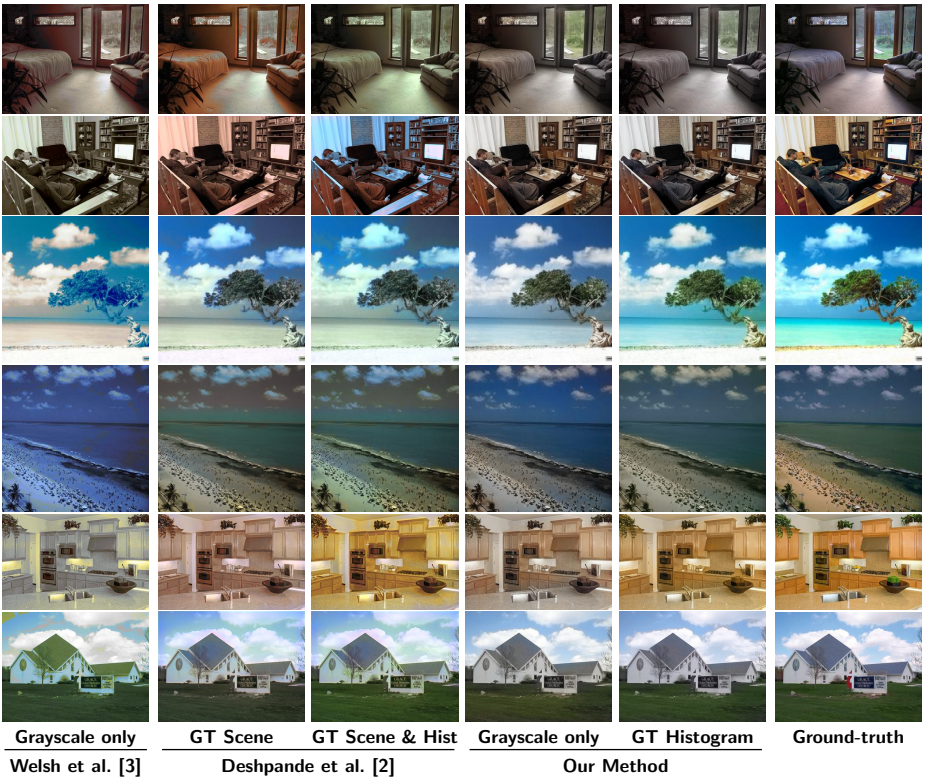


Fig. 4: **SUN-6.** Additional qualitative comparisons.

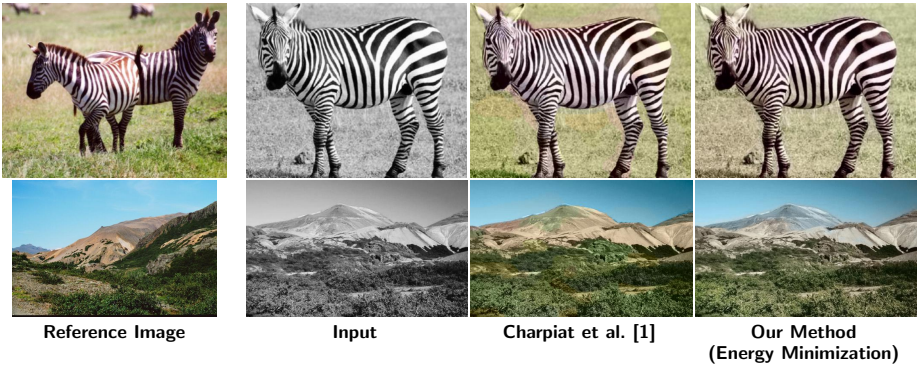


Fig. 5: **Transfer.** Comparison with Charpiat *et al.* [1] with reference image. Their method works fairly well when the reference image closely matches (compare with Figure 6). However, they still present sharp unnatural color edges. We apply our histogram transfer method (Energy Minimization) using the reference image.

Fig. 6: **Portraits.** Comparison with Charpiat *et al.* [1], a transfer-based method using 53 reference portrait paintings. Note that their method works significantly worse when the reference images are not hand-picked for each grayscale input (compare with Figure 5). Our model was not trained specifically for this task and we used no reference images.

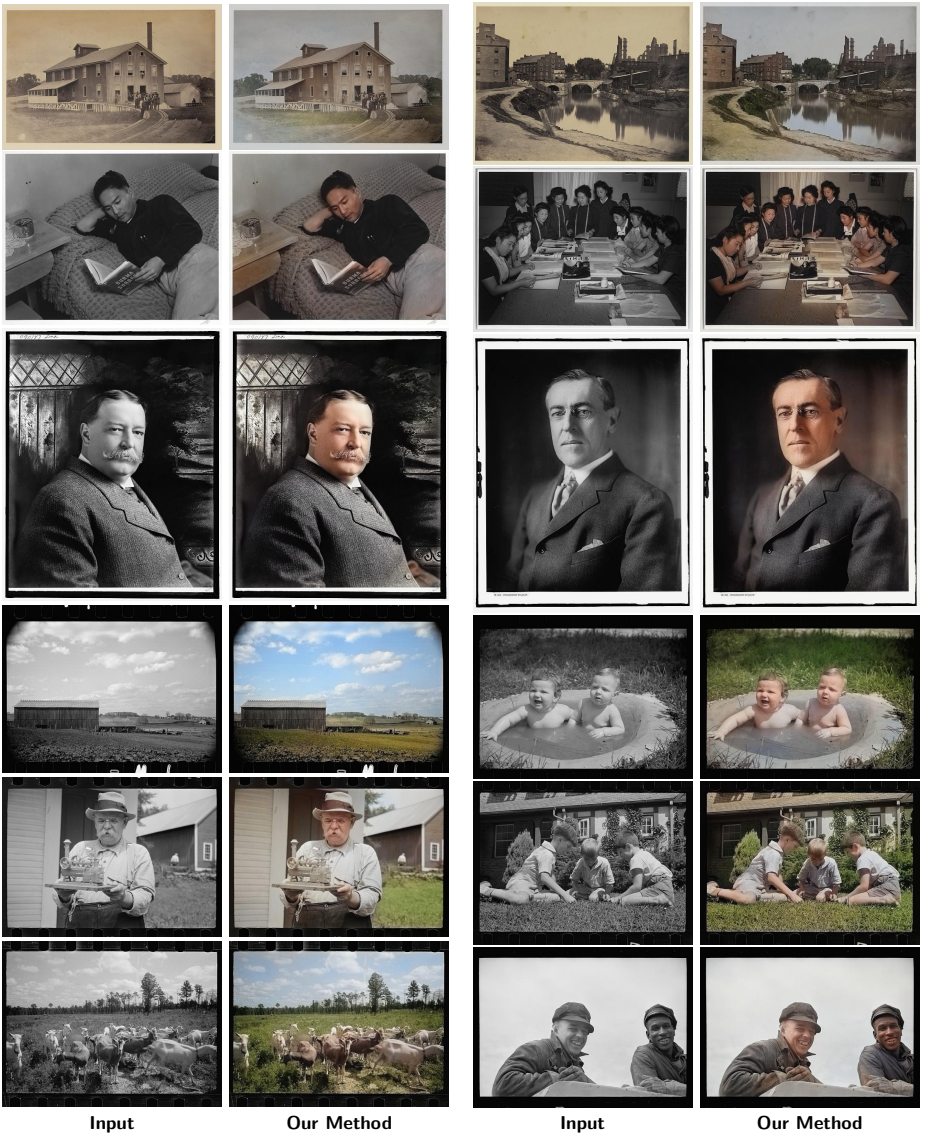


Fig. 7: **B&W photographs.** Old photographs that were automatically colorized. (Source: Library of Congress, `www.loc.gov`)

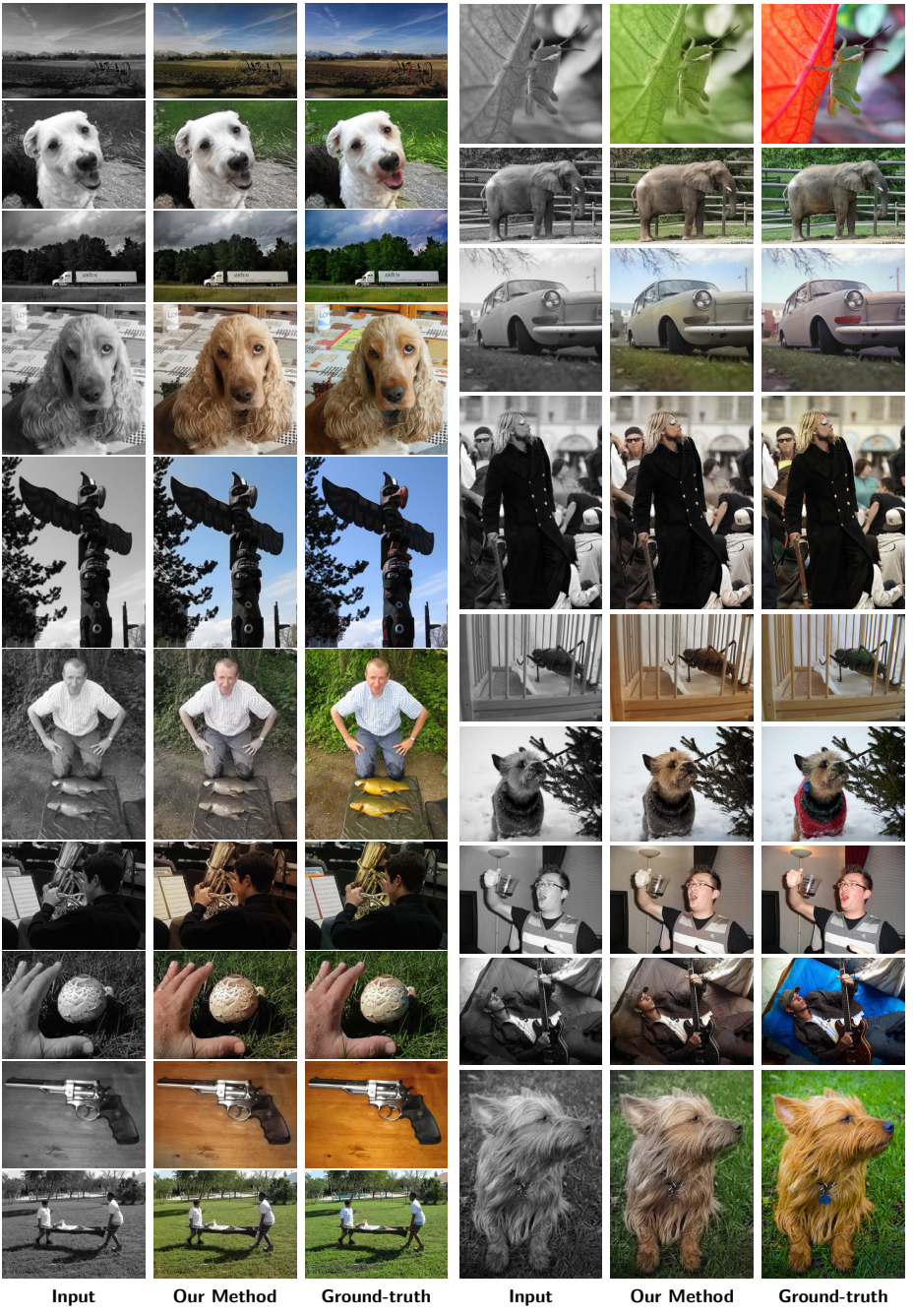


Fig. 8: **Fully automatic colorization results on ImageNet/ctest10k.**

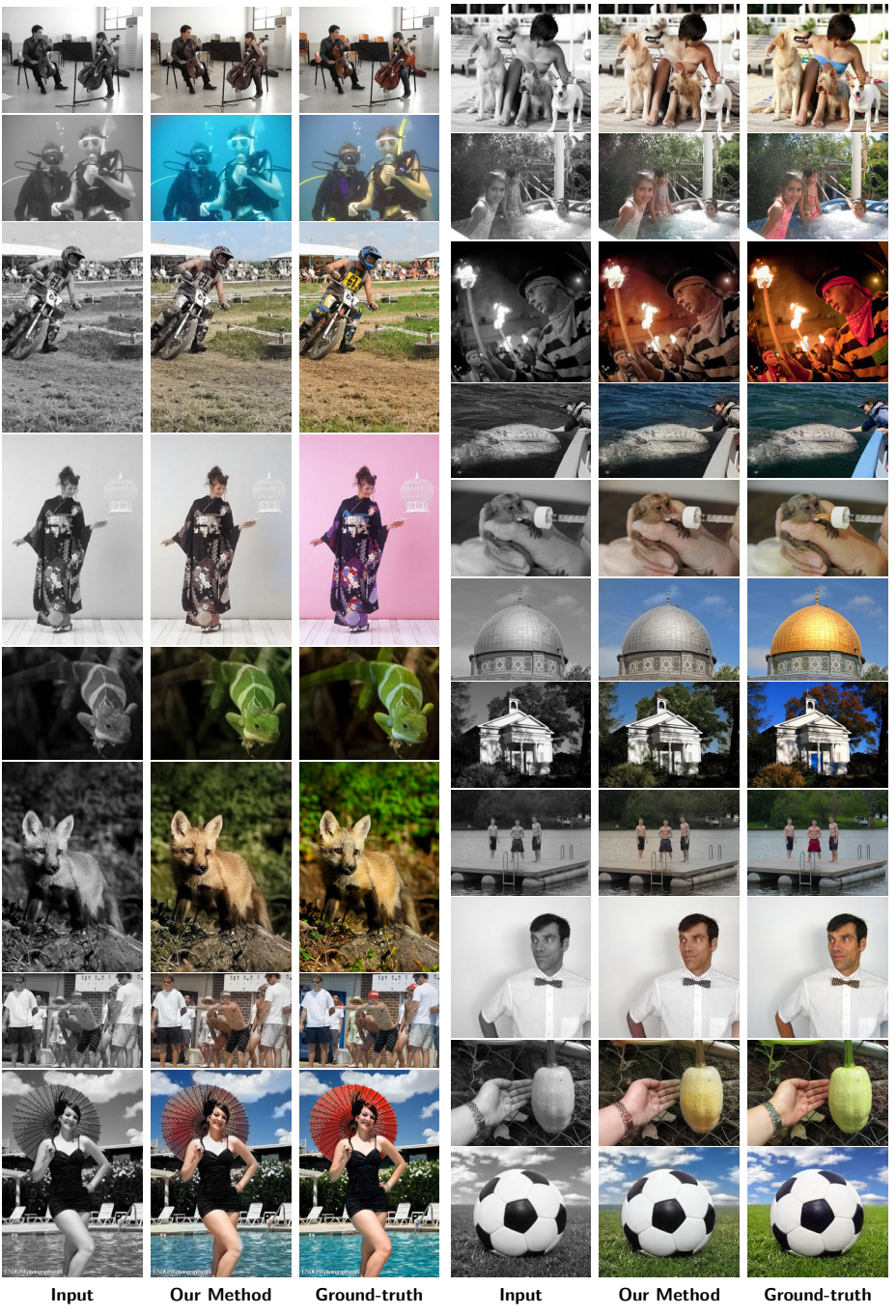


Fig. 9: **Fully automatic colorization results on ImageNet/ctest10k.**

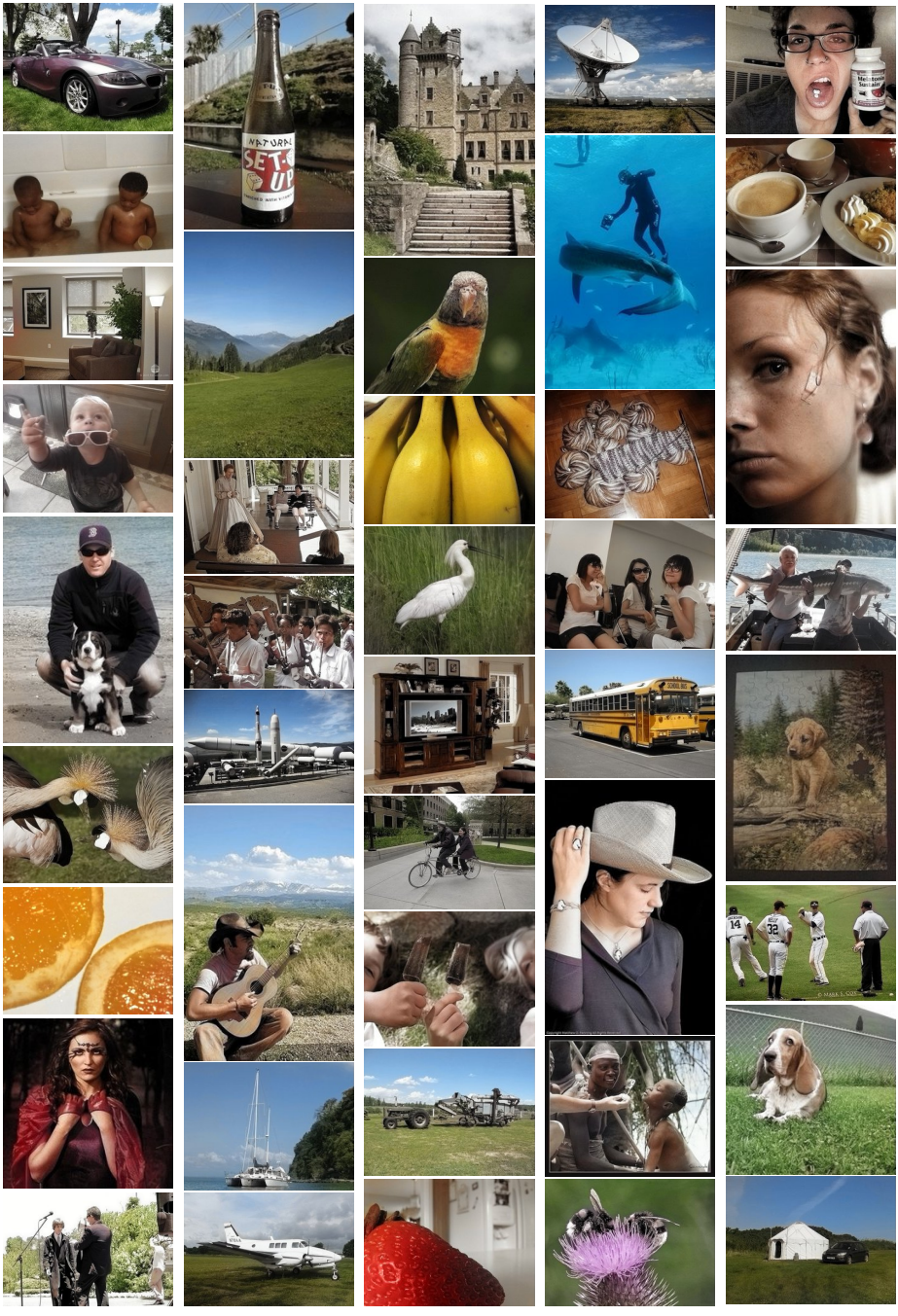


Fig. 10: **Fully automatic colorization results on ImageNet/ctest10k.**

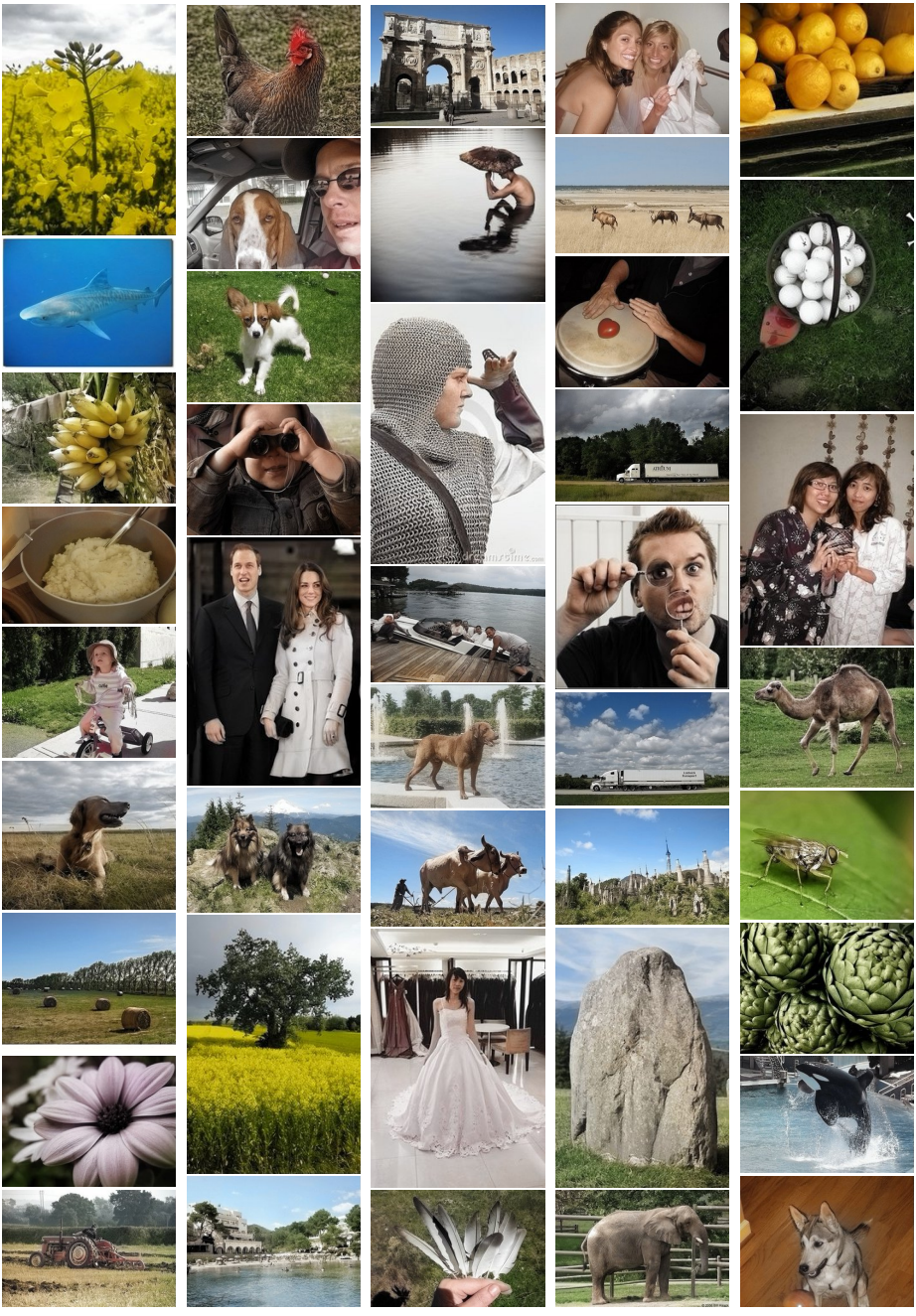
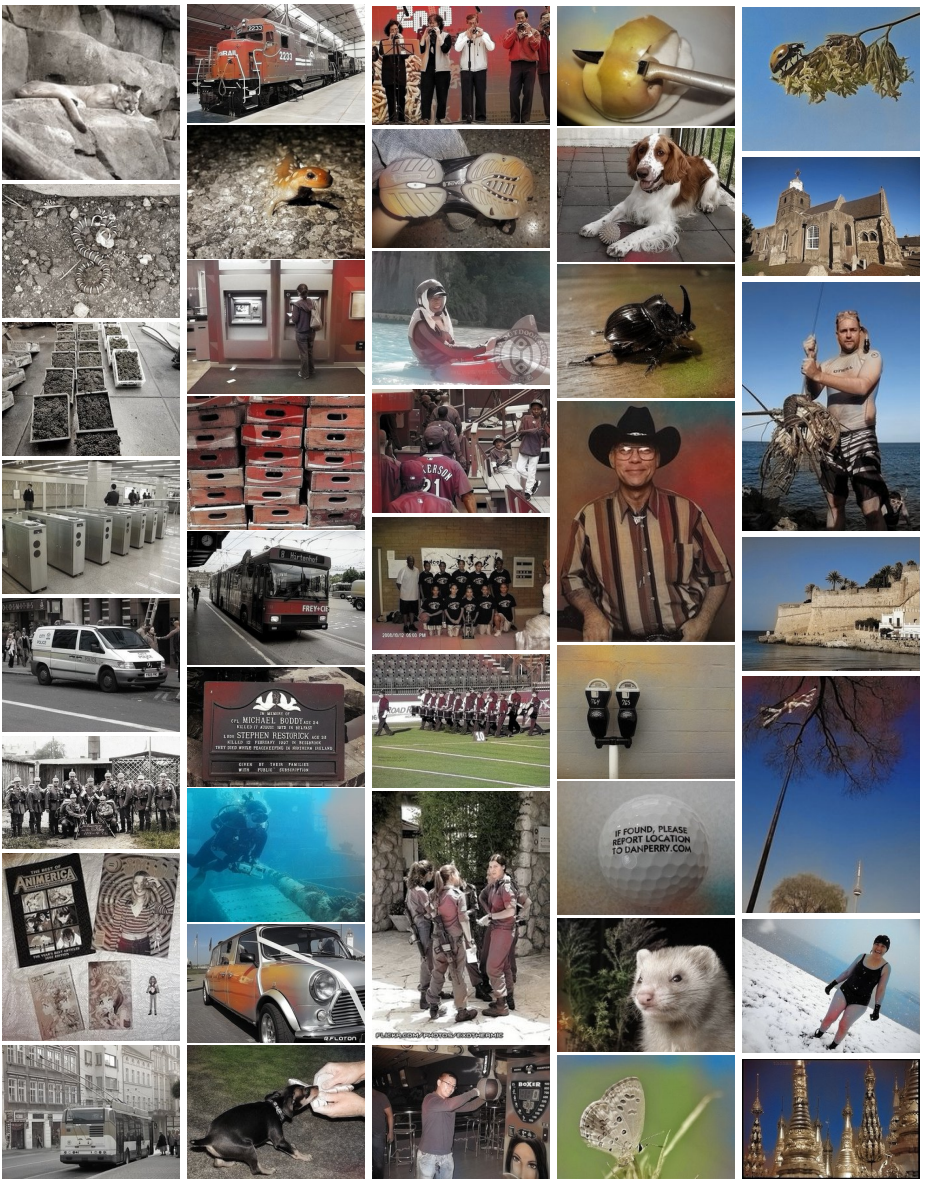


Fig. 11: **Fully automatic colorization results on ImageNet/ctest10k.**



Too Desaturated **Inconsistent Chroma** **Inconsistent Hue** **Edge Pollution** **Color Bleeding**

Fig. 12: **Failure cases.** Examples of the five most common failure cases: the whole image lacks saturation (*Too Desaturated*); inconsistent chroma in objects or regions, causing parts to be gray (*Inconsistent Chroma*); inconsistent hue, causing unnatural color shifts that are particularly typical between red and blue (*Inconsistent Hue*); inconsistent hue and chroma around the edge, commonly occurring for closeups where background context is unclear (*Edge Pollution*); color boundary is not clearly separated, causing color bleeding (*Color Bleeding*).