BigColor: Colorization using a Generative Color Prior for Natural Images –Supplemental Document–

Geonung Kim[1], Kyoungkook Kang[1], Seongtae Kim[1], Hwayoon Lee[1], Sehoon Kim[2], Jonghyun Kim[2], Seung-Hwan Baek[1], Sunghyun Cho[1]

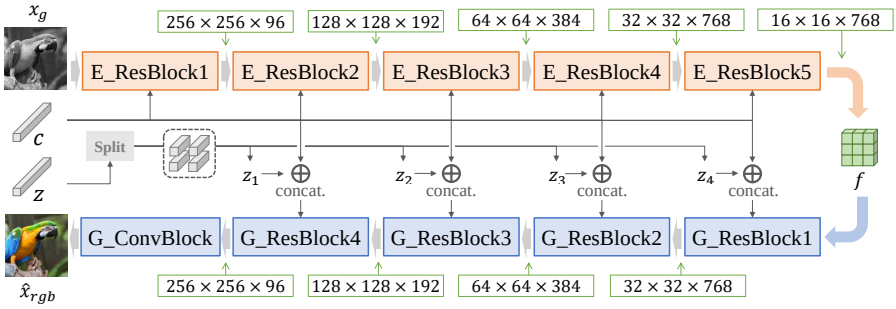
[1]POSTECH [2]Samsung Electronics

In this Supplemental Document, we present additional details, analysis, and results. Specifically, we provide:

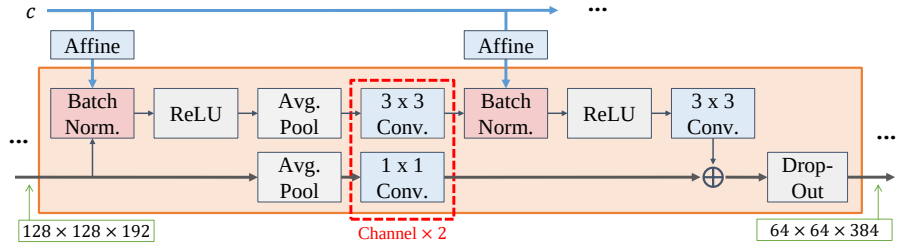
- Details on the network architecture,
- User study details,
- Additional discussion on training set,
- Additional ablations,
- Additional discussion on the color enhancement augmentation,
- Additional examples of the luminance replacement,
- Qualitative and quantitative comparisons,
- Additional uncurated examples,
- Additional examples of the multi-modal colorization, and
- Additional discussions on the limitations.

S1 Details on Network Architecture

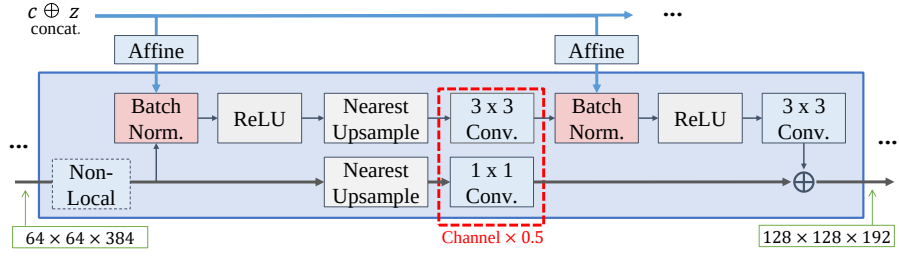
Fig. S1 shows the detailed network architecture of BigColor. We designed our encoder architecture by inverting the original BigGAN generator [2]. Our encoder consists of several ResBlocks, which are designed based on the ResBlock of the generator with a couple of modifications. Specifically, while the generator of BigGAN has non-local layers in a few ResBlocks, we exclude them and adopt an additional drop-out layer at the end of each ResBlock for performance improvement. For the generator, we adopt the network architecture of the fine-scale layers of the BigGAN generator. We initialize the generator with a pretrained BigGAN model on the ImageNet-1K training set [5]. The random vector z fed to the generator has a dimension of 68, which splits into four 17-dim vectors. We concatenate each split vector with the class code c of dimension 128, resulting in a 145-dim vector. Our code will be made public upon the acceptance of the paper.



(a) Encoder-generator overview



(b) Details of E_ResBlock3



(c) Details of G_ResBlock3

Fig. S1. (a) Overview of BigColor architecture. Orange and blue boxes indicate encoder and generator blocks, respectively. We denote the output feature dimension at each layer shown in the green boxes. The split operation divides the random code z into four segments, which are then concatenated with the class code c. (b) Details of E_ResBlock3 shown in (a). (c) Details of G_ResBlock3 shown in (a). The red dotted box indicates that the number of feature channels is doubled or halved in these operations.

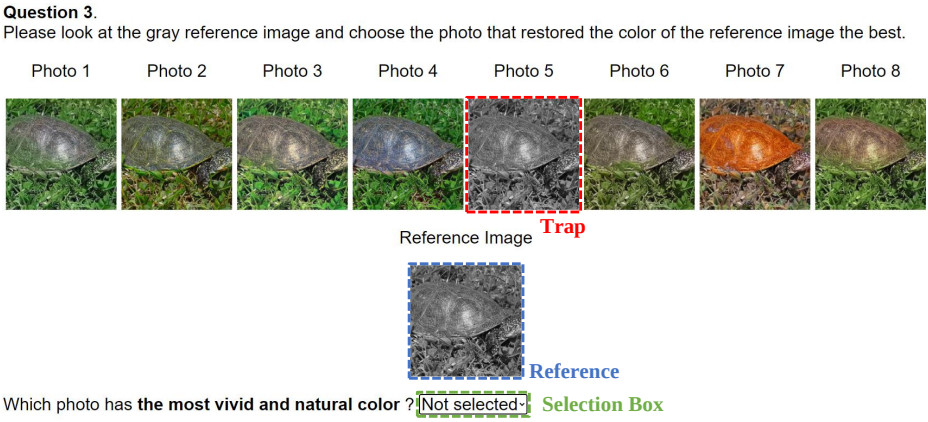


Fig. S2. Our user study interface. Top row shows the colorization results of different methods including a *trap* image denoted as a red box, which aims to filter out insincere users. Bottom-center shows the input grayscale image, denoted as a blue box. The selection box, denoted as green, provides an interface to choose the best one.

S2 User Study

We designed an Amazon-Mechanical-Turk interface for our user study as shown in Fig. S2. For an input grayscale image, a user should select the most preferred colorization image among the results with different methods. In total, there are 100 test samples. We shuffled the order of the results to remove any bias in the sequence. We also intentionally added the input grayscale image as a trap option in the colorization results for the sanity check of the user study. We excluded all the subjects who selected the trap image at least one time. In this way, we selected 33 valid participants out of 200 total subjects. Tab. S1 shows the full preference scores of all of the participants and statistics that summarize the user scores. The results show that the participants generally prefer to the results of BigColor over the others.

Table S1. The preference percentage to colorization methods for 33 participants in the user study. Most subjects prefer the colorization result of BigColor to the other methods. BigColor achieves the best percentage for all criterion: min, max, lower quartile(Q1), higher quartile(Q3), mean,and median.

User	CIC	ChromaGAN	InstColor	DeOldify	ColTran	ToVivid	BigColor
1	0.13	0.09	0.06	0.09	0.24	0.13	**0.26**
2	0.10	0.07	0.06	0.04	0.15	0.17	**0.41**
3	0.12	0.07	0.08	0.06	0.22	0.13	**0.32**
4	0.06	0.07	0.07	0.09	0.18	0.13	**0.40**
5	0.12	0.10	0.07	0.07	**0.25**	0.14	**0.25**
6	0.11	0.07	0.05	0.21	0.16	0.10	**0.30**
7	0.09	0.13	0.11	0.15	0.17	0.05	**0.30**
8	0.13	0.12	0.09	0.13	0.18	0.15	**0.20**
9	0.05	0.09	0.14	0.23	0.17	0.07	**0.25**
10	0.04	0.15	0.10	0.10	0.16	0.14	**0.31**
11	0.06	0.05	0.06	0.16	**0.26**	0.15	**0.26**
12	0.20	0.03	0.08	0.13	**0.24**	0.16	0.16
13	0.09	0.10	0.11	0.21	0.14	0.08	**0.27**
14	0.01	0.09	0.19	**0.28**	0.20	0.10	0.13
15	0.12	0.11	0.06	0.10	0.18	0.12	**0.31**
16	0.05	0.03	0.03	0.10	0.23	0.13	**0.43**
17	0.04	0.06	0.06	0.09	0.23	0.11	**0.41**
18	0.12	0.08	0.09	0.07	**0.32**	0.07	0.25
19	0.04	0.21	0.11	0.19	0.13	0.10	**0.22**
20	0.03	0.07	0.06	0.08	0.22	0.14	**0.40**
21	0.03	0.05	0.05	0.11	0.29	0.11	**0.36**
22	0.05	0.15	0.11	0.12	0.19	0.11	**0.27**
23	0.15	0.11	0.14	0.08	0.18	0.11	**0.23**
24	0.03	0.06	0.07	0.09	0.30	0.09	**0.36**
25	0.09	**0.20**	0.14	0.15	0.12	0.12	0.18
26	0.05	0.14	0.11	0.20	0.10	0.11	**0.29**
27	0.09	0.10	0.11	**0.21**	0.16	0.14	0.19
28	0.05	0.05	0.04	0.10	0.23	0.11	**0.42**
29	0.05	0.06	0.05	0.12	0.28	0.09	**0.35**
30	0.21	0.10	0.06	0.13	0.13	0.10	**0.27**
31	0.13	0.09	0.09	0.18	**0.19**	**0.19**	0.13
32	0.02	0.13	0.11	0.09	0.20	0.12	**0.33**
33	0.09	**0.20**	0.13	**0.20**	0.13	0.09	0.16
Max	0.21	0.15	0.14	0.28	0.32	0.19	**0.43**
Q3	0.12	0.12	0.11	0.18	0.23	0.14	**0.35**
Mean	0.08	0.10	0.09	0.13	0.20	0.12	**0.28**
Median	0.09	0.09	0.08	0.12	0.19	0.11	**0.27**
Q1	0.05	0.07	0.06	0.09	0.16	0.10	**0.23**
Min	0.01	0.03	0.03	0.04	0.10	0.05	**0.13**

S3 Additional Assessment

S3.1 With and Without Less Colorful Training Images

In our main manuscript, we present results obtained using a training set where we exclude 10% of images with low colorfulness score [3]. In this section, we show an additional experimental result obtained from the full training set including training images with low colorfulness scores. Tab. S2 shows that BigColor trained with the full training set also outperforms the baseline methods, which clearly proves the effectiveness of our approach. Also, BigColor trained without images with low colorfulness scores slightly outperforms the full dataset-based model, which shows that excluding less colorful images indeed helps improve the colorization performance.

Table S2. Qualitative comparisons with previous automatic colorization methods and BigColor models using the different training datasets. BigColor still achieves state-of-the-art performance for all metrics without the filtered dataset.

Methods	Colorful ↑	FID ↓	Classification ↑
CIC [9]	33.036	11.322	69.976
ChromaGAN [7]	26.266	8.209	70.374
InstColor [6]	25.507	7.890	68.422
DeOldify [1]	23.793	3.487	72.364
ColTran [4]	34.485	3.793	67.210
ToVivid [8]	35.128	4.078	73.816
BigColor (colorful dataset)	**40.006**	**1.243**	**76.516**
BigColor (full dataset)	<u>39.774</u>	<u>1.473</u>	<u>76.320</u>

S3.2 Additional Ablations

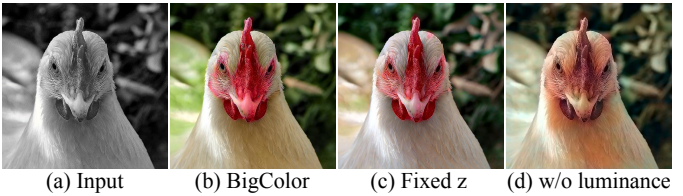
We conduct additional ablation studies using 10% of the ImageNet training images amounting to 100 image classes.

Fixing z Fixing the latent code z makes our color-synthesis process deterministic, resulting in the incapability of multi-modal synthesis capability. It reduces the colorfulness score from 33.3 to 26.2, while the fidelity of colorization remains, as shown in Tab. S3 and Fig. S3 (c). We conjecture that this is because fixing z decreases the representation space of BigColor, which is originally spanned by (f, c, z).

Learning without Luminance For the colorization task where an input image already has the luminance components, it may be enough to estimate the color components without the luminance. On the other hand, our approach estimates RGB color values for an input image in order to effectively exploit the pretrained GAN generator, which is trained in the RGB color space. To validate this, we modify BigColor to synthesize only color components *ab* in the *Lab* color space. Fig. S3 (d) and Tab. S3 shows that learning in the RGB space results in significantly higher-quality results, indicating that training BigColor in the RGB domain is essential to fully exploit the pretrained GAN prior.

Table S3. Quantitative results with the fixed z and without the luminance, demonstrating their contributions to the final performance.

Methods	Colorful ↑	FID ↓	Classification ↑
BigColor (fixed z)	26.169	5.787	81.44
BigColor (w/o luminance)	29.024	20.971	73.68
BigColor	**33.299**	**5.714**	**81.44**



(a) Input (b) BigColor (c) Fixed z (d) w/o luminance

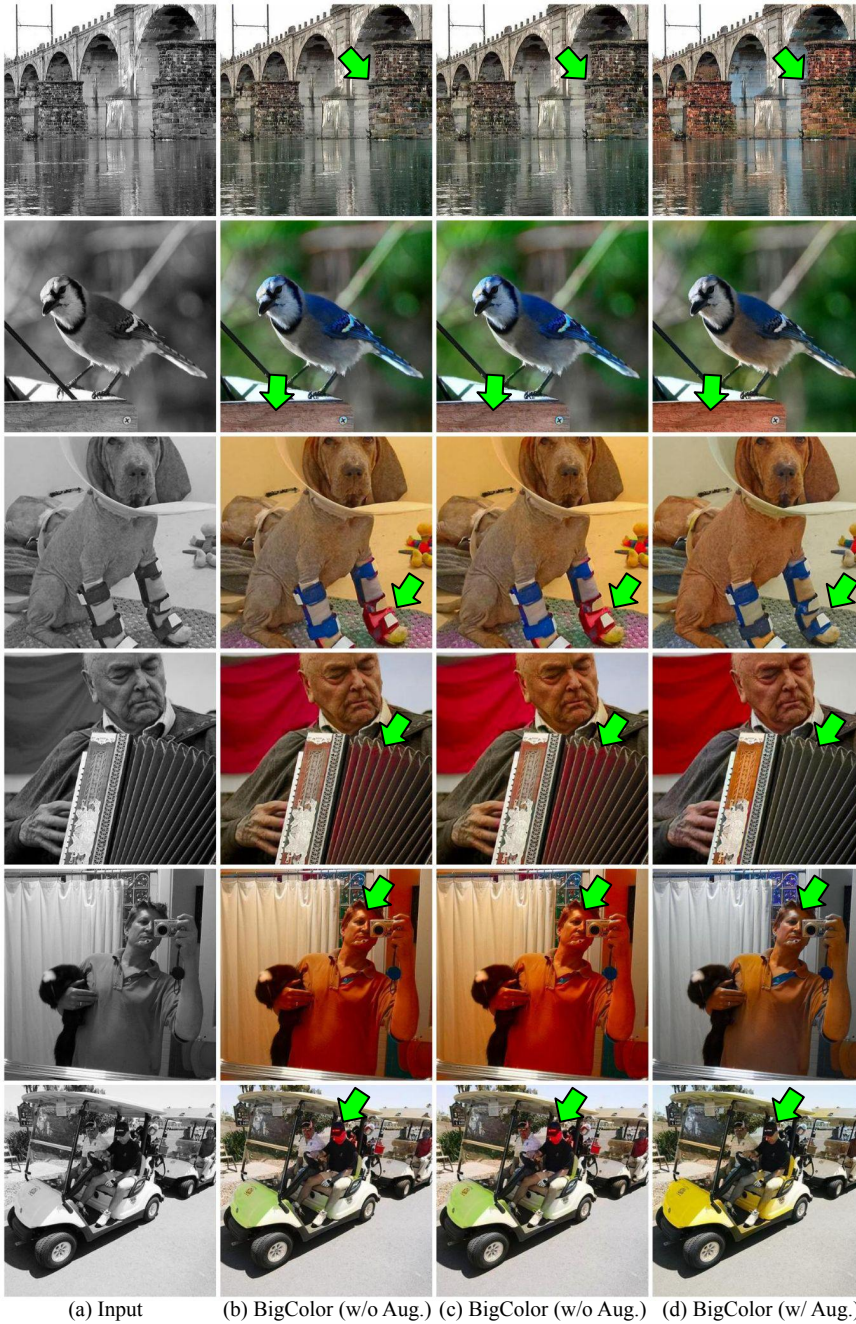
Fig. S3. Qualitative results with (c) the fixed z and (d) without the luminance, given (a) an input gray-scale image. (b) BigColor achieves the best qualitative result.

S3.3 Impact of Color Augmentation

Our color augmentation strategy enhances the vividness and semantic correctness of synthesized color images by altering the colors of the real images in the training set. Specifically, our color augmentation applies to the real color images in the training set, which is fed to the discriminator during training, only by a small amount so that it does not introduce unwanted distortion to the color distribution of real images. On the other hand, it still makes the colors of semantically different regions in the training images more distinguishable during the training phase. In consequence, it helps the generator learn to synthesize semantically more correct and vivid colors. In this section, we qualitatively demonstrate the effect of our color augmentation. For the quantitative evaluation, we refer the readers to Tab. 5 in the main manuscript.

Fig. S4 shows a qualitative comparison of our color augmentation strategy with two baselines: BigColor without our augmentation and BigColor with post color processing. For the post color processing, we apply the same color color-balancing method used in BigColor directly to the output of the generator.

The results of BigColor without the color augmentation and with the post color processing (Fig. S4(b) and (c)) have similar colors while the results in (c) are slightly more vivid. This implies that our color augmentation scheme does not change the color distribution of the real images in the training set much. Also, compared to our results in (d), the results without the color augmentation have less vivid and semantically inaccurate colors, e.g., the dull colors of the bricks and wood on the first and second rows, the red color on the accordion on the fourth row, and the red human face on the last row. While the post processing slightly boosts the color vividness, its results still have the exact same artifacts as it is applied at the end of the colorization process. On the other hand, our results have more vivid and semantically correct colors, showing that our color augmentation scheme helps the generator synthesize more vivid and accurate colors.



(a) Input (b) BigColor (w/o Aug.) (c) BigColor (w/o Aug.) + post Aug. (d) BigColor (w/ Aug.)

Fig. S4. Our color augmentation scheme enhances the vividness and semantic correctness of colorization results. Note that color augmentation as post-processing fails to achieve these improvements.

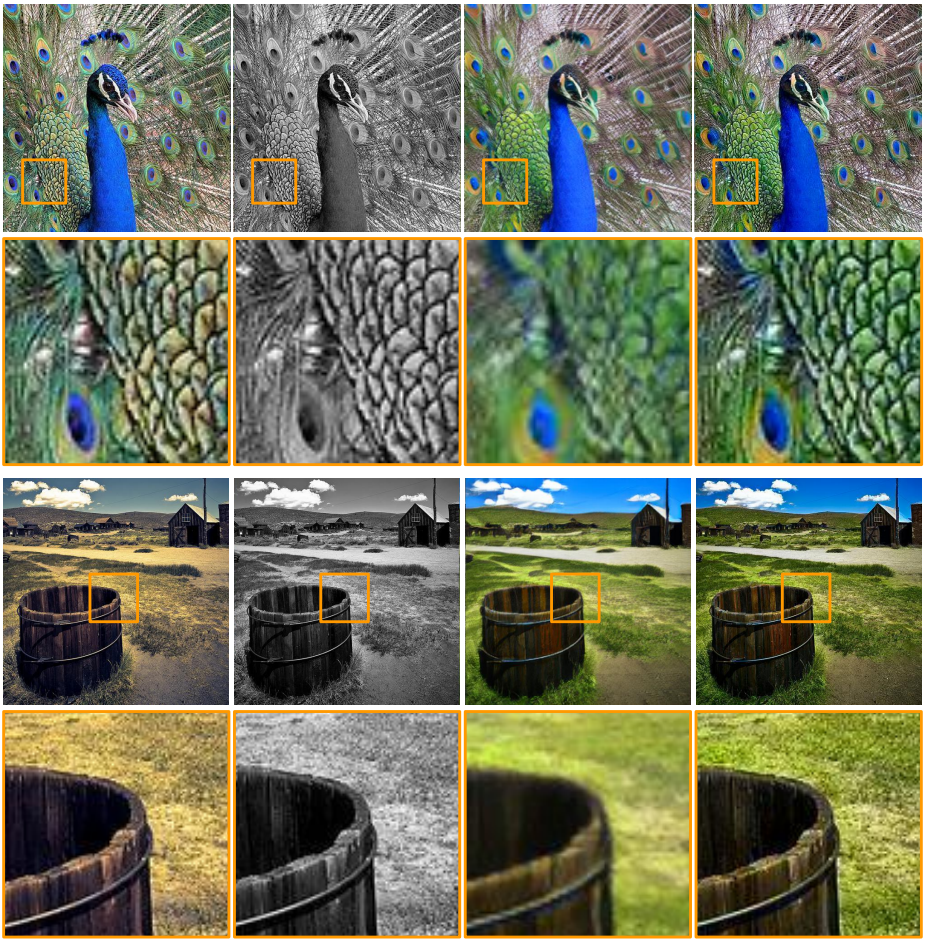


Fig. S5. Effect of replacing luminance. We replace the luminance of generator output with the luminance of the input image to boost high-frequency details on the output of the generator. It is better seen in the enlarged images in the second and fourth rows.

S3.4 Luminance Replacement

BigColor brings the high-frequency spatial details of the input grayscale image via luminance replacement. Fig. S5 shows that the luminance replacement enables us to recover high-frequency spatial details while maintaining the high-quality colors synthesized from our generator.

S4 Additional Comparisons

In this section, we provide additional qualitative and quantitative comparisons against recent colorization methods [8,1,9,7,6,4]. For the previous methods, we used the official codes from the authors except for ToVivid [8], which does not provide code at the time of our submission. For the results of ToVivid, we requested the authors and obtained the results for our input images. We use the *stable* version of DeOldify [1].

As ColTran [4] and ToVivid [8] can only deal with images of a specific size, we conducted our experiments as follows. For all the other methods except for ColTran and ToVivid, we resized the input images keeping the aspect ratio so that the smaller between the width and height is 256. We then colorize the input and crop the center regions to make the output resolution 256×256. Regarding ToVivid, we resized the input images to 256×256 ignoring the aspect ratio, obtained their colorization results, and resized the results to recover the original aspect ratios while keeping the smaller between the width and height still 256. Then, we cropped the center regions of size 256×256. Regarding ColTran, we resized the input images keeping the aspect ratio so that the smaller between the width and height is 256, cropped the center regions of size 256×256, and performed colorization.

Qualitative Comparisons We show additional qualitative comparisons in Fig. S6, Fig. S7, and Fig. S8. BigColor outperforms all the compared methods for challenging scenes with diverse semantics and structures.

Comparison on Challenging Images As described in the main manuscript, we constructed a curated dataset consisting of 100 complex images based on the number of people. Fig. S9 shows representative examples of simple images and complex images with respect to the number of people. Note that the complex images not only contain many people, but also have higher image complexity in general. Tab. S4 and Fig. S10 show quantitative and qualitative comparisons of different colorization methods. Again, BigColor achieves state-of-the-art quantitative and qualitative performance.

Table S4. BigColor is robust for colorizing complex images compared to the previous colorization methods, achieving the best performance in terms of FID and colorfulness score.

Metric	CIC	ChromaGAN	InstColor	DeOldify	ColTran	ToVivid	BigColor
FID ↓	111.317	114.336	92.589	66.575	116.997	90.904	**65.729**
Colorful ↑	31.080	24.789	22.534	24.303	35.065	36.198	**42.077**

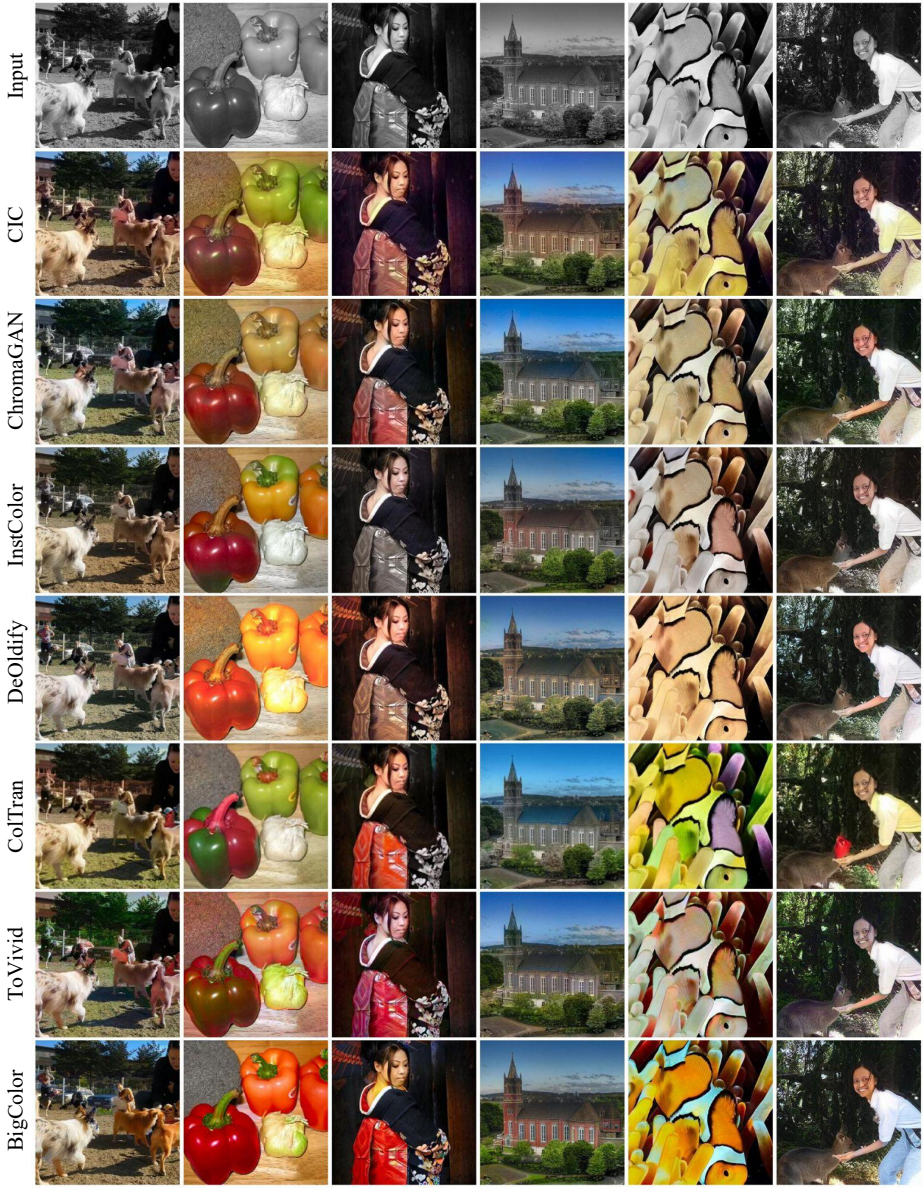


Fig. S6. Qualitative comparison of different colorization methods on the ImageNet1K validation set [5].

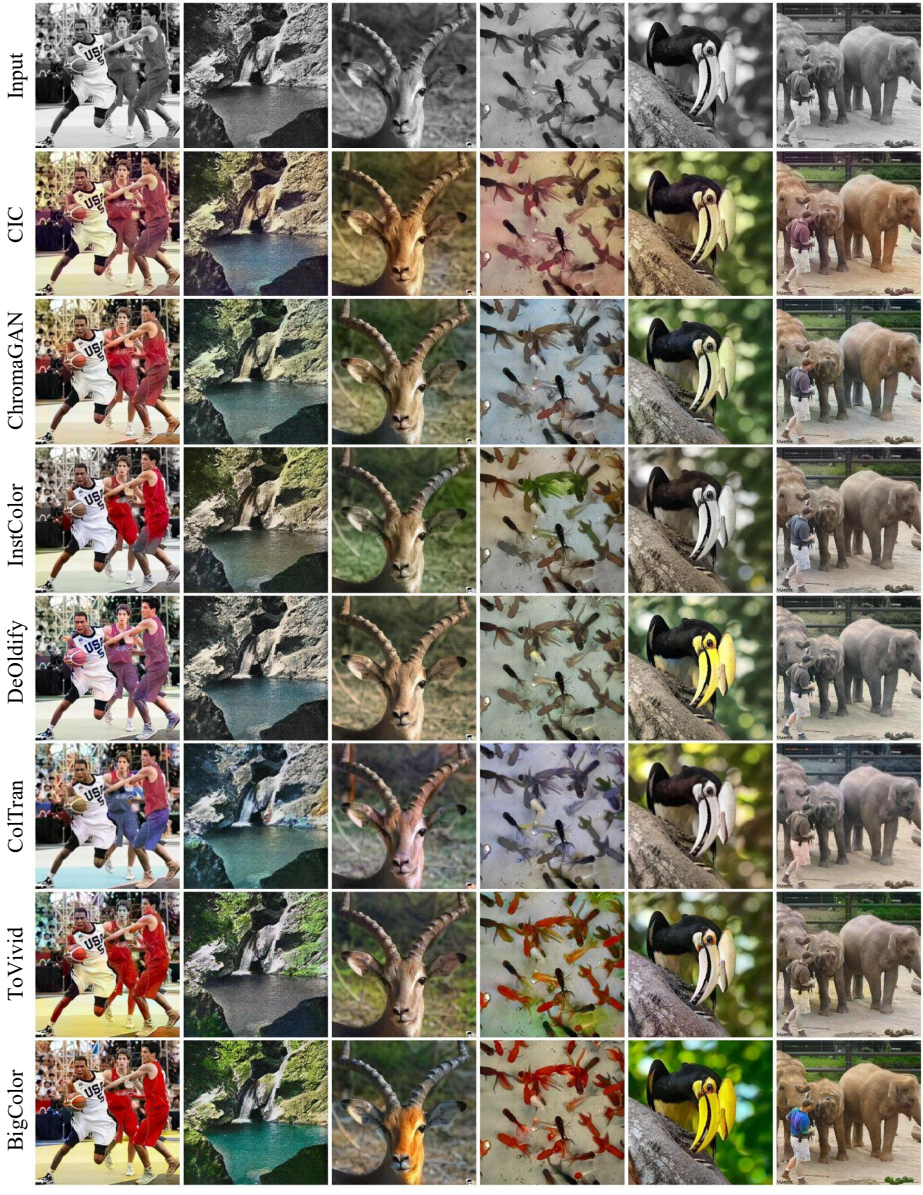


Fig. S7. Qualitative comparison of different colorization methods on the ImageNet1K validation set [5].

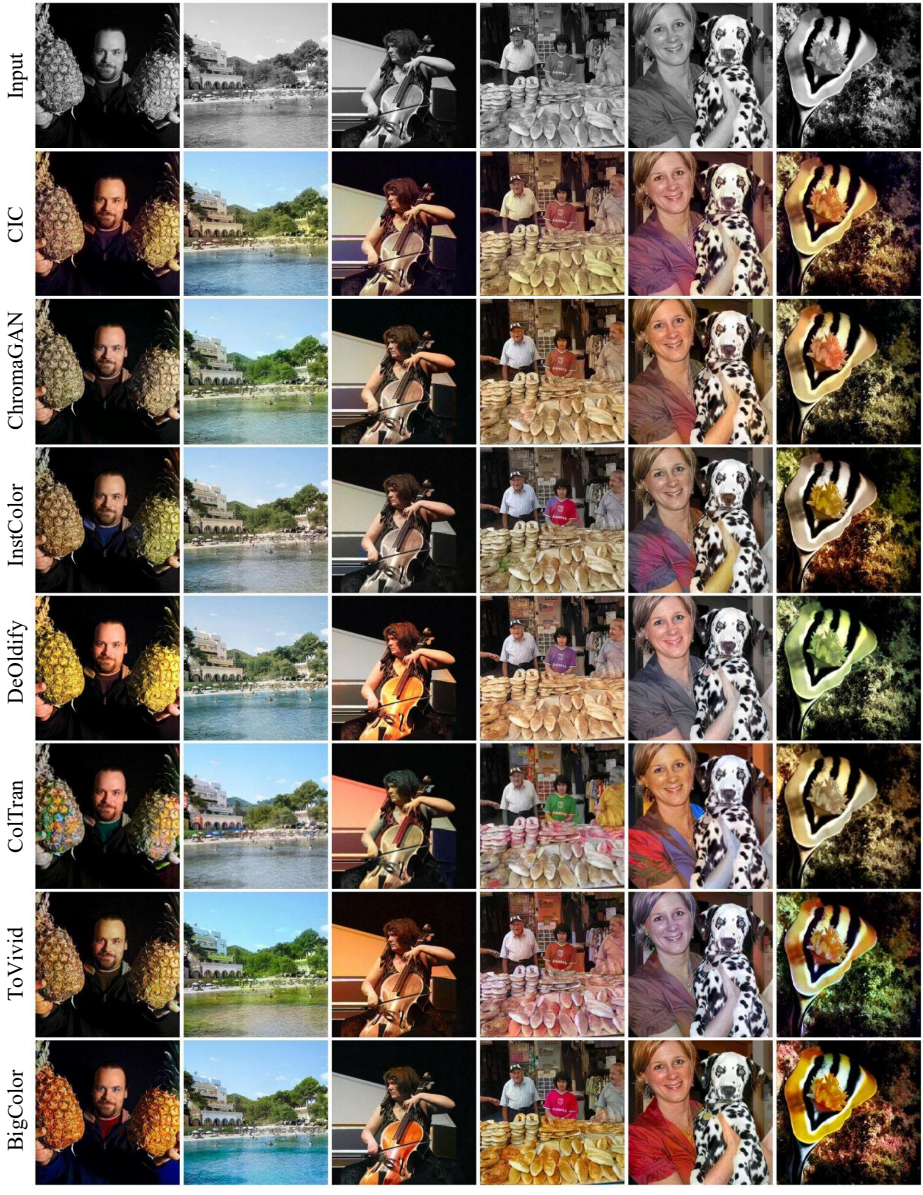
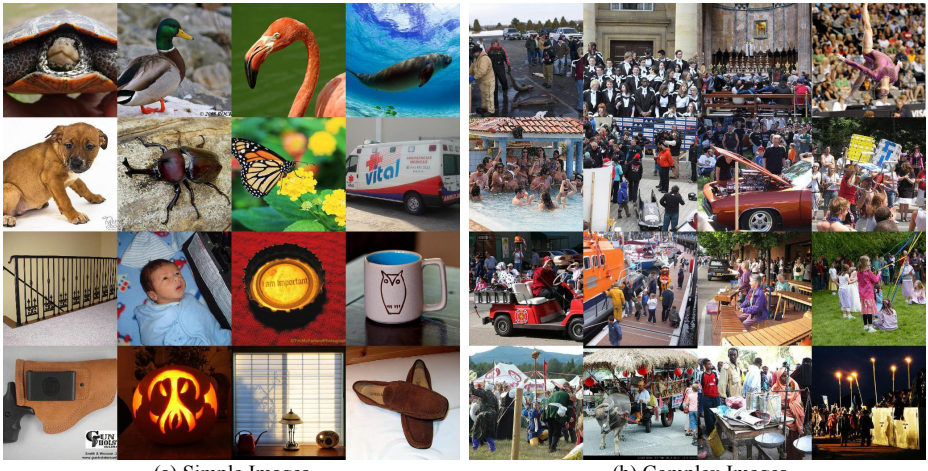


Fig. S8. Qualitative comparison of different colorization methods on the ImageNet1K validation set [5].



(a) Simple Images (b) Complex Images

Fig. S9. Our curated simple images and complex images in the ImageNet1K dataset [5]. Although we assume the proportionality between the number of people and image complexity, the complex images often contain various objects that make the image more complicated and diverse.

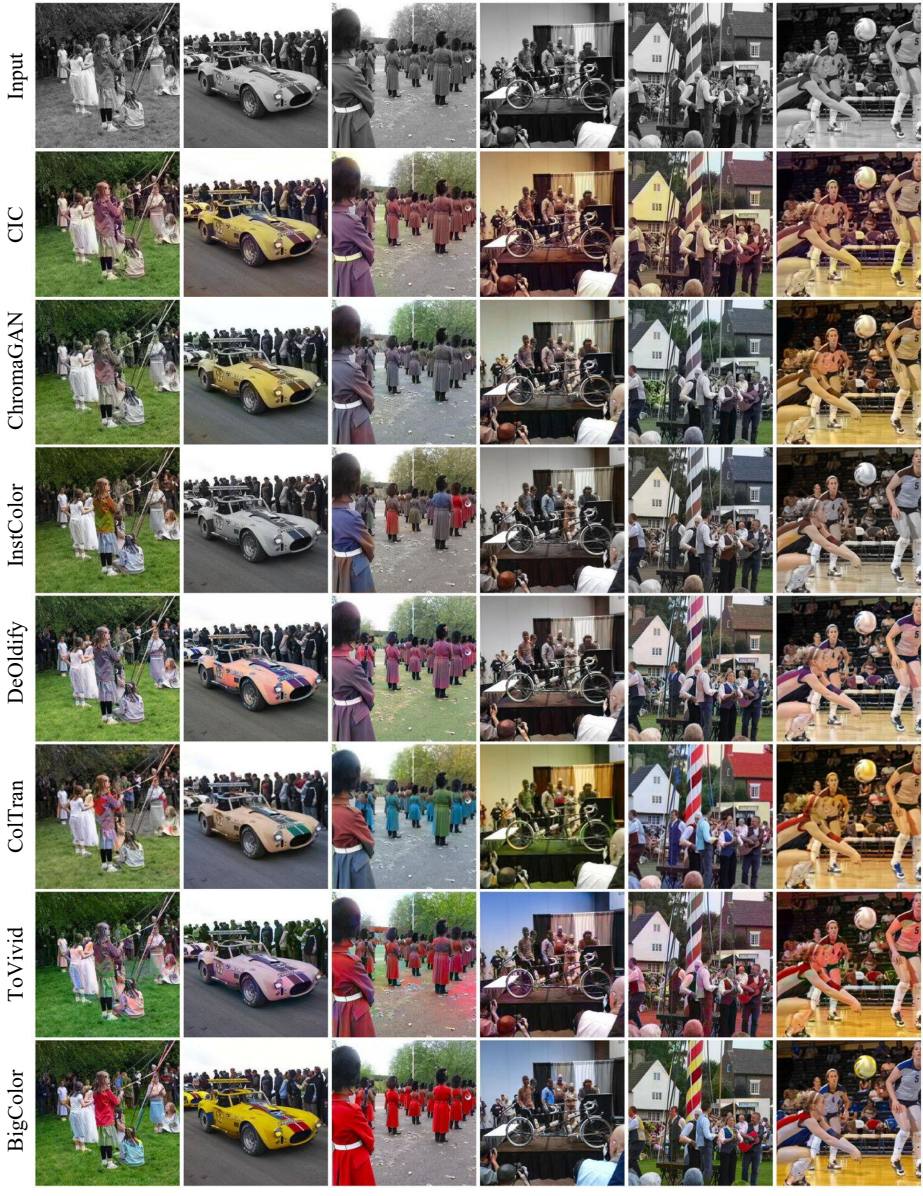


Fig. S10. Qualitative comparison of different colorization methods on complex images.

S5 Additional Results

Uncurated Examples BigColor enables robust colorization for in-the-wild images. We show our colorization results on 100 *uncurated* images in Fig. S11 and Fig. S12.

S5.1 Multi-modal Colorization

BigColor allows us to synthesize diverse colors for an input grayscale image. Fig. S13 shows additional results of multi-modal colorization by using different random codes z.

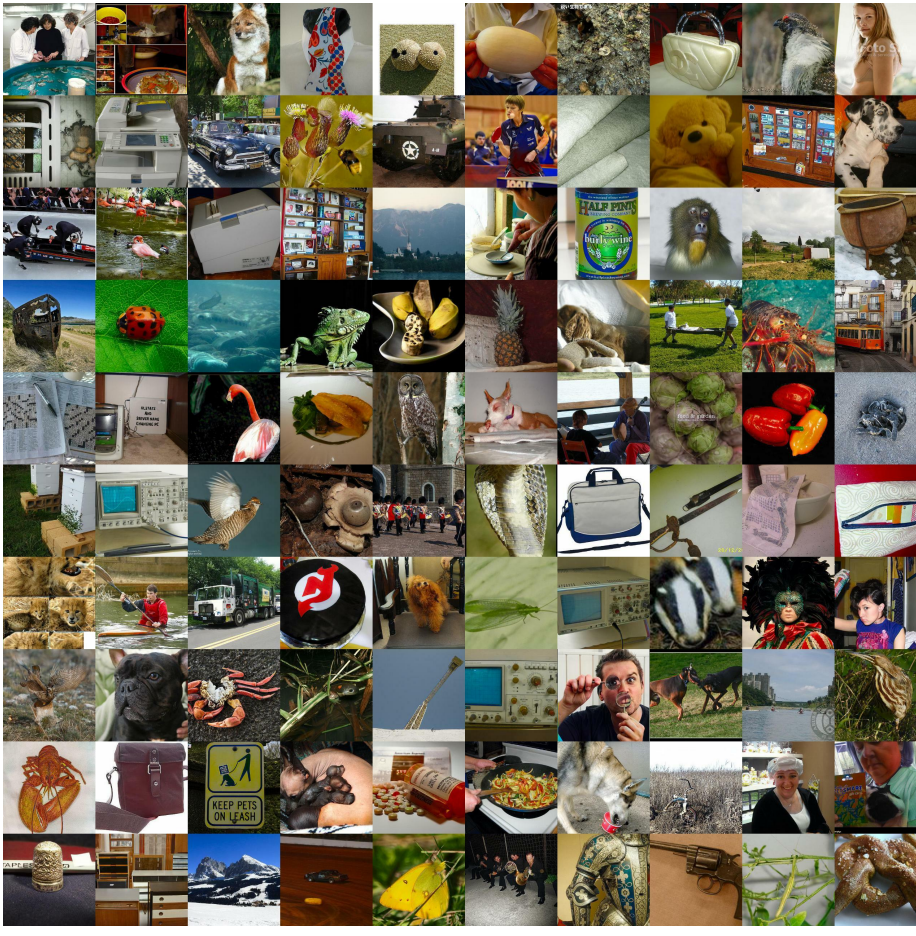


Fig. S11. Uncurated colorization results of BigColor, showing robust performance of BigColor to various samples.

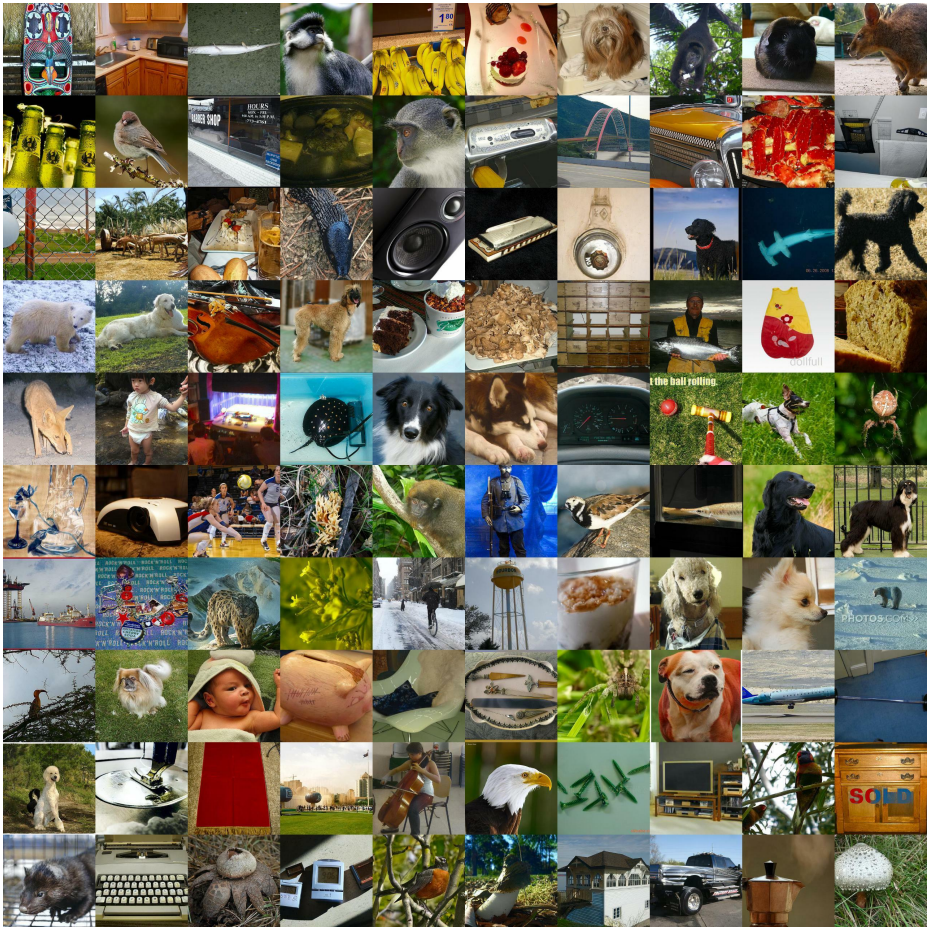


Fig. S12. Uncurated colorization results of BigColor, showing robust performance of BigColor to various samples.

Fig. S13. BigColor successfully produces multi-modal colorization results by sampling different random codes z.

S5.2 Limitations

Fig. S14 shows three failure cases of BigColor. First, BigColor may fail to colorize very small regions such as the ear in the first row in the figure. This is because the spatial resolution of extracted feature f is small as 16×16. Second, BigColor may struggle with complex images that contain objects with significantly different object classes such as toys and humans. Lastly, BigColor may fail to handle input images considerably different from training images. Old grayscale photographs could fall into such category as they often deviate from the training-data grayscale distribution due to their unique film sensitivity and chemical development recipes.

Fig. S14. Limitation of BigColor. Top to bottom: missing tiny region, mixed image classes, domain gap with training images.

References

1. Antic, J.: Deoldify. `https://github.com/jantic/DeOldify` (2019)
2. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019)
3. Hasler, D., Suesstrunk, S.E.: Measuring colorfulness in natural images. In: Human vision and electronic imaging VIII. vol. 5007, pp. 87–95. International Society for Optics and Photonics (2003)
4. Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer. In: International Conference on Learning Representations (2021)
5. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
6. Su, J.W., Chu, H.K., Huang, J.B.: Instance-aware image colorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7968–7977 (2020)
7. Vitoria, P., Raad, L., Ballester, C.: Chromagan: Adversarial picture colorization with semantic class distribution. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2445–2454 (2020)
8. Wu, Y., Wang, X., Li, Y., Zhang, H., Zhao, X., Shan, Y.: Towards vivid and diverse image colorization with generative color prior. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14377–14386 (2021)
9. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision. pp. 649–666. Springer (2016)