BigColor: Colorization using
a Generative Color Prior for Natural Images

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]

POSTECH[1]
{k2woong92,kkang831,seongtae0205,hwayoon2,shwbaek,s.cho}@postech.ac.kr
Samsung Electronics[2]
{sh0264.kim,jh015.kim}@samsung.com

Abstract. For realistic and vivid colorization, generative priors have
recently been exploited. However, such generative priors often fail for
in-the-wild complex images due to their limited representation space.
In this paper, we propose BigColor, a novel colorization approach that
provides vivid colorization for diverse in-the-wild images with complex
structures. While previous generative priors are trained to synthesize
both image structures and colors, we learn a generative color prior to
focus on color synthesis given the spatial structure of an image. In this
way, we reduce the burden of synthesizing image structures from the gen-
erative prior and expand its representation space to cover diverse images.
To this end, we propose a BigGAN-inspired encoder-generator network
that uses a spatial feature map instead of a spatially-flattened BigGAN
latent code, resulting in an enlarged representation space. BigColor en-
ables robust colorization for diverse inputs in a single forward pass, sup-
ports arbitrary input resolutions, and provides multi-modal colorization
results. We demonstrate that BigColor significantly outperforms existing
methods especially on in-the-wild images with complex structures.

Keywords: Colorization, GAN Inversion, Generative Color Prior

1 Introduction

Image colorization aims to hallucinate the chromatic dimension of a grayscale
image and has been studied for decades in computer vision and graphics. Its
application includes not only modernizing classic black-and-white films but also
providing artistic control over grayscale imagery with diverse color distributions
[25,39,4,34,20].

Early works propagate user-annotated color strokes based on pixel affin-
ity [22,13,38,28,36] or find similar regions in reference images to mimic the ref-
erence color distributions [4,6,9]. With the advent of deep learning, data-driven
colorization approaches have rapidly advanced by adopting neural networks to
learn a mapping from grayscale images to trichromatic images. This trend was
sparked by using a convolutional neural network (CNN) and a regression loss
such as mean-squared error (MSE) [39,32,31,1], which unfortunately suffers from
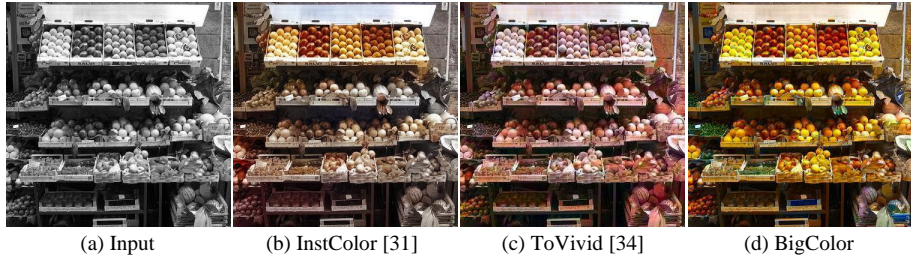
(a) Input   (b) InstColor [31]   (c) ToVivid [34]   (d) BigColor

Fig. 1. We achieve robust colorization for in-the-wild images using a generative color prior. (a) For an input image with complex spatial structures, existing colorization methods suffer from (b) desaturated color and (c) unnatural color distribution. (d) In contrast, BigColor synthesizes natural colors consistent with the input structure using a learned generative color prior.

desaturated colors as shown in Fig. 1 (b), as the MSE loss encourages to find an average of plausible color images corresponding to an input image.

To synthesize vivid colors, high-quality representations learned in pretrained generative adversarial network (GAN) models have recently been exploited as generative priors for image colorization [34,37,33,27,8]. Adopting GAN inversion, these methods invert an input grayscale image to a latent code of a pretrained GAN model by minimizing the structural discrepancy between the input grayscale image and the generated color image from the latent code. While GAN inversion allows us to utilize the learned generative prior of natural images, it also inherits a notable problem of existing GAN models: limited representation space. Thus, existing colorization methods using generative priors fail to handle in-the-wild images with complex structures and semantics, resulting in desaturated and unnatural colors as shown in Fig. 1(c).

In this paper, we propose BigColor, a novel image colorization method that synthesizes vivid and natural colors for in-the-wild images with complex structures. For vivid colorization, we adopt the GAN-inversion approach by using a pretrained BigGAN [2], which is a state-of-the-art class-conditional generative model. As directly using the BigGAN model hampers colorization performance for in-the-wild images due to its limited representation space, we offload the burden of the BigGAN model that was responsible for synthesizing both structures and colors to focus on color synthesis. This offloading strategy allows us to learn a generative color prior that can cover in-the-wild images with complex structures.

Specifically, we learn a generative color prior with an encoder-generator neural network. Unlike conventional GAN-inversion colorization methods, our encoder extracts a spatial feature map describing the structure of an input image better than using a spatially-flattened latent code in BigGAN. As a spatial feature map has a higher spatial resolution than an original BigGAN latent code, the representation space of the entire network can be enlarged, i.e., we can map features to a wider range of natural images. We then design our generator to directly exploit the spatial feature by using the fine-scale network layers adopted

from the multi-scale BigGAN generator. We jointly train the encoder and generator networks to encourage the network to focus on color synthesis by making use of the spatial feature. As our network is fully convolutional and departs from using a fixed-size flattened latent code of BigGAN, BigColor can process images with arbitrary sizes which were not feasible for conventional GAN-inversion colorization methods that use the original latent codes of GANs [27,34,8,37,33]. Also, BigColor allows us to synthesize multi-modal colorization results by using different condition vectors for the network. We assess BigColor with extensive experiments including a user study and demonstrate that BigColor outperforms previous methods across all tested scenarios in particular for in-the-wild images.

2   Related Work

Optimization-based Colorization Early colorization methods utilize color annotations from users and propagate them to neighbor pixels based on pixel affinity by solving constrained optimization problems [22,13,38,28,36]. Data-driven colorization methods find reference color images with similar semantics to an input grayscale image and use the reference color distributions via optimization [24,4,6,9]. Unfortunately, the optimization-based approaches demand dense user annotations or accurate reference matching, failing to provide robust and automatic colorization.

Colorization with Regression Networks Learning a mapping function from a grayscale image to a color image has been extensively studied with the advent of neural networks. Regression-based neural networks minimize average reconstruction error, resulting in desaturated colors [5,7,21,14]. Vivid color synthesis then became one of the core challenges in network-based image colorization methods. Notable examples in this line of research include optimizing over a quantized color space [39], detection-guided colorization [31], adversarial training [32,1], and global reasoning using a transformer [20]. While significant progress has been made, it is still challenging to synthesize vivid and natural colors for in-the-wild grayscale images with complex structures.

Colorization with Generative Prior GANs have recently achieved remarkable success in learning low-dimensional latent representations of natural color images, enabling synthesizing high-fidelity natural images [17,18,2]. This success has led to using the learned generative prior for image restoration such as deblurring [33,37], super-resolution [3,26,27], denoising [33,37], and colorization [33,37,27,34,8]. Most previous approaches are limited to handling a single class of images, such as human faces using StyleGAN [17,18], due to the limited representation space of modern GAN models.

Recently, a few attempts [27,34] have been made to colorize natural images of multiple classes using a pretrained BigGAN generator [2]. Specifically, deep generative prior (DGP) [27] jointly optimizes the BigGAN latent code and the pretrained BigGAN generator to synthesize a color image via GAN inversion. The representation space of the DGP is still not enough to cover complex images because of the difficulty in synthesizing both structures and colors from the
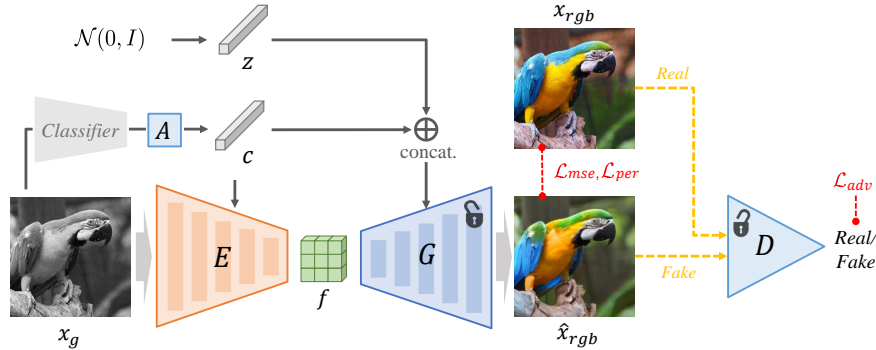
Fig. 2. We extract the spatial feature f of the input image x_g using a class-conditioned convolutional encoder E. The generator G, which is initialized with the fine levels of the pretrained BigGAN [2], takes the spatial feature f as inputs and synthesizes a colorized image $\hat{x}_{rgb}$ conditioned on the control parameters of the class code c and the random sample z. A is a class embedding layer that transforms a one-hot class vector to a class code c. Jointly training the encoder-generator model with a pretrained BigGAN discriminator D enables us to learn the generative color prior with an enlarged representation space.

generator. Wu et al. [34] attempted to bypass the structural mismatch between a GAN-inverted color image and an input grayscale image by warping the synthesized color features into the input grayscale. Nonetheless, considerable mismatches between a GAN-inverted and an input image cannot be fully resolved, and thus produce colorization artifacts. In contrast to the previous methods, Big-Color effectively enlarges the representation space by using an encoder-generator architecture that uses spatial features. This allows us to handle diverse images with complex structures.

3   Colorization using a Generative Color Prior

In this section, we describe the framework of BigColor and our strategy to learn a generative color prior. BigColor has an encoder-generator network architecture, where the encoder E estimates a spatial feature map f from an input grayscale image x_g, and the generator G synthesizes a color image $\hat{x}_{rgb}$ from the feature f. Note that different from conventional GAN-based colorization methods, we do not rely on the spatially-flattened latent code of BigGAN, but instead use a spatial feature map f that has a larger dimension. In order to exploit the effectiveness of the BigGAN architecture for image synthesis [2], we design the encoder E and the generator G by using the fine-scale layers of the BigGAN generator. Also, we use two control variables for conditioning the encoder and the generator: the class code c and the random code z sampled from a normal distribution. The class code c enables class-specific feature extraction for effective colorization and the random code z accounts for the multi-modal nature of image colorization.
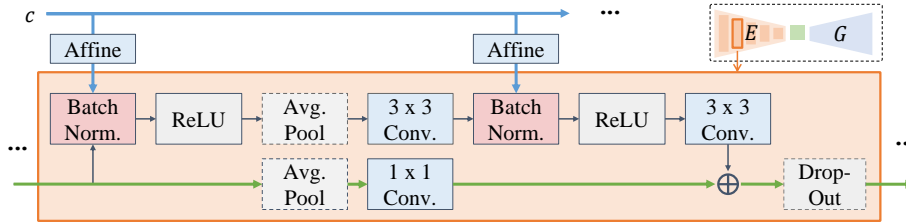
Fig. 3. We design our encoder E by inverting the fine layers of the BigGAN generator [2], consisting of the five encoder blocks shown in the top right. Each encoder block denoted as an orange box extracts the spatial features conditioned on the class code c. The first and the last encoder blocks do not have average pooling and dropout respectively.

In the spirit of adversarial learning, we also adopt a pretrained BigGAN discriminator D. We jointly train the encoder E, the generator G, and the discriminator D, resulting in an enlarged representation space where the generator G takes the responsibility of synthesizing color on top of the spatial feature f extracted from the encoder E. See Fig. 2 for an overview of BigColor. In the following, we describe each component of BigColor and the training scheme in detail.

3.1   Encoder

Our encoder takes an input grayscale image x_g and estimates a spatial feature map f, which is fed to the generator. For an input image size of 256×256, our spatial feature f has the spatial resolution of 16×16 with the channel size of 768. To successfully extract the spatial feature f, we design our encoder inspired by an inversion of the BigGAN generator as shown in Fig. 3. The encoder consists of five blocks, where all the blocks except for the first have average pooling layers to reduce the spatial size of an input feature. We also adopt dropout layers except for the last block for better generalization on test-case inputs.

To extract class-specific spatial structures, we inject the class information of an input image into the encoder. Specifically, we obtain the scale and bias parameters of the batch-normalization layers through an affine transformation of the BigGAN class code $c \in \mathbb{R}^{128 \times 1}$ [2]. We adopt the BigGAN's class embedding layer (A in Fig. 2) to obtain the class code c from a class vector in the form of the one-hot vector representation. The class vector can be either provided by the user or estimated using an off-the-shelf classifier. In our experiments, we use a 1,000-dimensional vector for the class vector representing ImageNet-1K classes. More details on the architecture can be found in the Supplemental Document. In summary, our encoder E extracts the class-specific spatial feature map f that contains the structure information of an input image x_g as

$$f = E(x_g; c). \tag{1}$$

BigColor

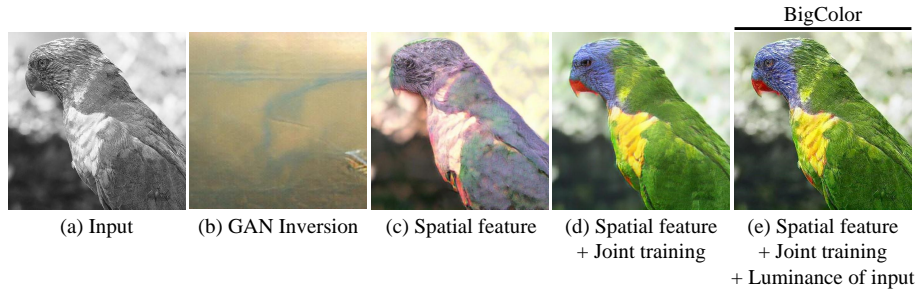|(a) Input|(b) GAN Inversion|(c) Spatial feature|(d) Spatial feature + Joint training|(e) Spatial feature + Joint training + Luminance of input|

Fig. 4. (a)&(b) Colorization with conventional GAN inversion often fail to invert in-the-wild images. (c) We exploit the spatial feature of the input and the fine-scale layers of the pretrained BigGAN generator, effectively enlarging the representation space. (d) Jointly optimizing the encoder-generator module improves the representation coverage and provides vivid and natural colorization. (e) We boost high-frequency details by replacing the luminance of synthesized image with the input luminance.

3.2   Generator

Our generator G synthesizes colors given the spatial feature f of the input grayscale image x_g. Analogously to the encoder design, we design and initialize our generator G using the fine-scale layers of the pretrained BigGAN generator, specifically from the third to the last layers. The generator G uses two condition variables of the class vector c and the random vector z sampled from a normal distribution. We concatenate the class vector c and the random vector z as an input to the generator G as in the original BigGAN architecture [2]. Our generator G synthesizes a color image $\hat{x}_{rgb}$ conditioned on the class and the random codes as

$$\hat{x}_{rgb} = G(f; c, z). \tag{2}$$

We note that unlike the original BigGAN generator that uses a spatially-flattend latent code, our generator G takes the *spatial* feature f as input. To restore high-frequency spatial details, we replace the luminance of the synthesized color image $\hat{x}_{rgb}$ with the luminance of the input grayscale image x_g in the CIELAB color space [39,32,31]. See Fig. 4(e).

Generative Color Prior We learn the generative color prior for colorizing in-the-wild images with complex structures using our generator G. To this end, we exploit our specific network architecture and training scheme. For the architecture, our generator G takes the fine-scale spatial feature map f as an input of which resolution is $16 \times 16 \times 768$ when the grayscale image has 256×256 resolution. The dimension of the feature f is higher than that of the original BigGAN latent code of which resolution is 119×1. Thus, we can effectively enlarge the representation space of our generator G compared to the conventional GAN-inversion colorization methods by utilizing the structural information provided in the large-dimensional feature f. Compare the colorization results of Fig. 4(b)&(c). We note that a similar finding was used in BDInvert [15], a re-

cent transform-robust GAN-inversion method using a spatial feature for Style-GAN [17,18].

In terms of training strategy, we initialize the generator G and the discriminator D with the corresponding layers of the ImageNet-pretrained BigGAN model. As such, we can leverage the learned structure-color distribution of natural images of the pretrained BigGAN. However, our generator G at the initial point is still not fully focusing on synthesizing colors as it was originally trained to synthesize both structure and color. We unlock the full potential of our network by jointly optimizing the encoder E, the generator G, and the discriminator D. The joint training allows the generator G to learn a generative color prior by focusing on synthesizing colors on top of the spatial feature f. The reduced learning complexity of the generator results in an enlarged representation space, covering in-the-wild natural images as demonstrated in Fig. 4(d).

Multi-modal Image Colorization Image colorization is an inherently ill-posed problem as multiple potential color images could explain a single grayscale image. We handle this multi-modal nature of image colorization by injecting the random code z sampled from a normal distribution into the generator G. Sampling multiple latent code z enables synthesizing diverse color images. Note that we do not provide the random code to the encoder as the multi-modal nature only applies to the color synthesis, not the spatial feature extraction.

3.3 Training Details

Adversarial Training We train our framework in an alternating manner for adversarial learning. We define our encoder-generator loss function $\mathcal{L}^G$ as a sum of three terms:

$$\mathcal{L}^G = \mathcal{L}^G_{mse} + \lambda_{per}\mathcal{L}^G_{per} + \lambda_{adv}\mathcal{L}^{\mathcal{G}}_{adv}, \tag{3}$$

where $\mathcal{L}^G_{mse}$ and $\mathcal{L}^G_{per}$ are the MSE reconstruction losses that penalize the color and perceptual discrepancies between the synthesized image $\hat{x}_{rgb}$ and the ground truth image x_{rgb}. For the perceptual loss $\mathcal{L}^G_{per}$, we use the VGG16 [30] features at 1st, 2nd, 6th, and 9th layers. $\mathcal{L}^{\mathcal{G}}_{adv}$ is the adversarial loss, specifically the class-conditional hinge loss [23] defined as $\mathcal{L}^{\mathcal{G}}_{adv} = -D(\hat{x}_{rgb}, c)$. We use the balancing weights λ_{per} and λ_{adv} set as 0.2 and 0.03 respectively. For discriminator training, we also use the hinge loss [23]

$$\mathcal{L}^{\mathcal{D}}_{adv} = -\min(0, -1 + D(x_{rgb}, c)) + \min(0, -1 - D(\hat{x}_{rgb}, c)). \tag{4}$$

Color Augmentation To promote synthesizing vivid color, we apply a simple color augmentation to the real color images fed to the discriminator. Specifically, we scale chromaticity of images in YUV color space as $\{U, V\} \leftarrow \{1.2U, 1.2V\}$. This color augmentation makes colors of semantically different regions in training images more distinguishable. As a result, it helps the generator learn to synthesize not only more vivid but also semantically more correct colors, which is not achievable by direct augmentation of generator output as will be shown in Sec. 4.2.

Table 1. Quantitative comparison with other colorization methods using the three metrics of colorfulness [10], FID [12], and classification accuracy [39]. BigColor outperforms all previous work with significant margins. *Aug.* denotes our color-augmentation scheme. The bold and underlined scores are the best and 2nd best results.

	Colorful ↑ [10]	FID ↓ [12]	Classification ↑
CIC [39]	33.036	11.322	69.976
ChromaGAN [32]	26.266	8.209	70.374
InstColor [31]	25.507	7.890	68.422
DeOldify [1]	23.793	3.487	72.364
ColTran [20]	34.485	3.793	67.210
ToVivid [34]	35.128	4.078	73.816
BigColor (w/o Aug.)	<u>36.157</u>	<u>1.288</u>	<u>76.302</u>
BigColor (w/ Aug.)	**40.006**	**1.243**	**76.516**

4   Experiments

Implementation We train our model on 1.2M color images of the ImageNet 1K [29] training set after excluding 10% original images with low colorfulness scores [10]. We generate grayscale images based on a conventional linear-combination method[1]. We resize and crop the training images to be 256×256. For training, we use the Adam optimizer [19] with the coefficients of $\beta_1 = 0.0$ and $\beta_2 = 0.999$. The learning rates are set to 0.0001 for the encoder-generator module and 0.00003 for the discriminator with the decay rate of 0.9 per epoch. We also use the exponential moving average [16] with the coefficient of $\beta = 0.999$ for model parameter update. We set the batch size to 60 and train the entire model for 12 epochs.

4.1   Evaluation

We evaluate the effectiveness of BigColor on the ImageNet-1K validation set of 50K images [29] that have complex spatial structures.

Comparison with Other Colorization Methods We compare BigColor to recent automatic colorization methods including CIC [39], ChromaGAN [32], DeOldify [1], InstColor [31], ColTran [20] and ToVivid [34]. Fig. 5 shows that BigColor qualitatively outperforms all the methods on six challenging images. BigColor successfully colorizes the complex structures of human faces, penguin heads, food, and buildings with semantically-natural and vivid colors. The two notable state-of-the-art methods of ToVivid [34] and ColTran [20] suffer from unnatural colorization as shown on the penguins and the human face due to their limited representation space. This clearly demonstrates the effectiveness of our learned generative color prior to in-the-wild images. See the Supplemental Document for more qualitative results.

---

[1] $L = 0.2989R + 0.5870G + 0.1140B$, where L is the grayscale intensity and R, G, B are the trichromatic color intensities.
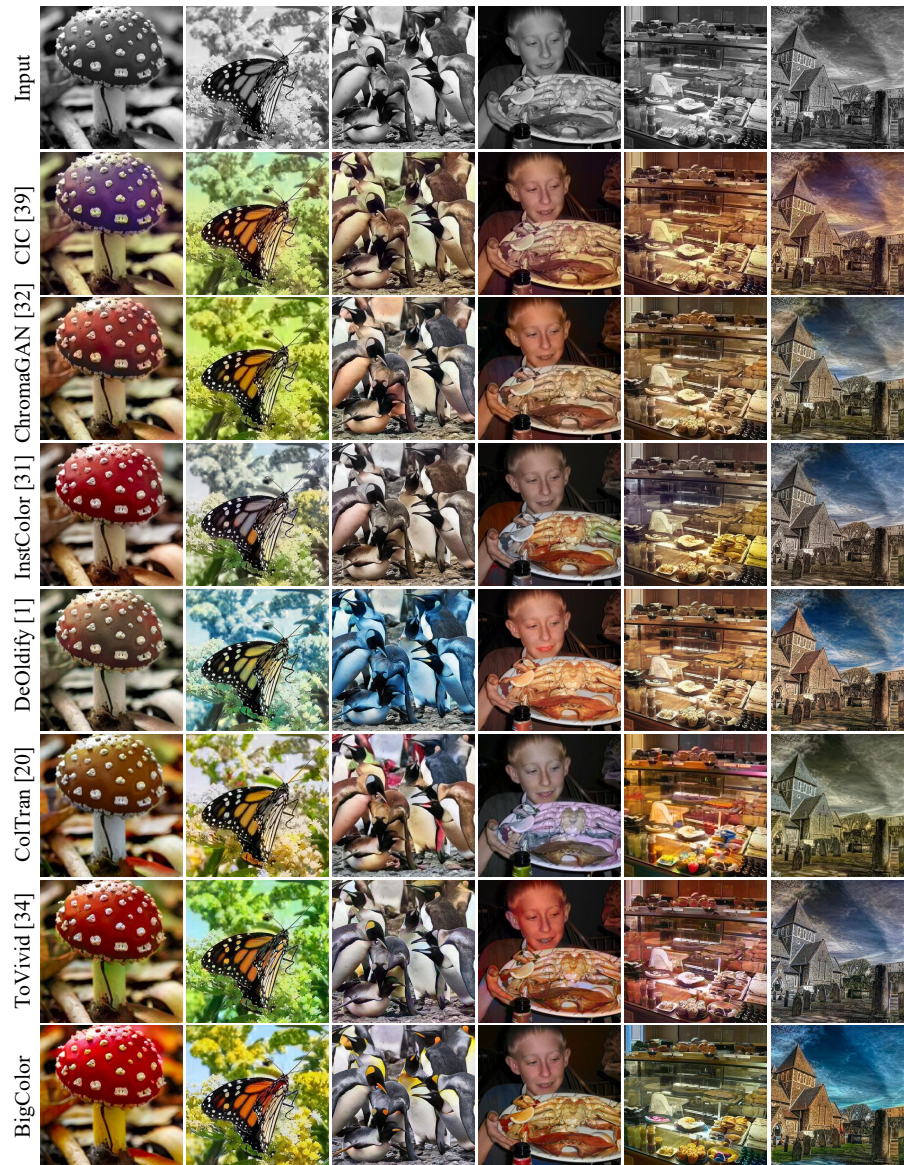
Fig. 5. Qualitative comparison with other colorization methods. For in-the-wild images with complex structure, our method synthesizes natural and vivid color images while the other methods suffer from desaturated and unnatural color distributions.

We further evaluate BigColor using the three quantitative metrics of colorfulness, FID, and classification accuracy commonly used in the image colorization field. Colorfulness measures the overall colorfulness of an image based on psychological experiments [10]. FID describes the distributional distance between the

Table 2. BigColor is robust for colorizing complex images compared to the previous colorization methods, achieving the best performance in terms of classification accuracy with a marginal performance drop similar to the real ground-truth color images.

	Classification acc.		*Performance drop*
Dataset	Whole	Complex	Whole → Complex
Ground Truth	78.530	76.000	-2.530
CIC [39]	69.976	64.000	-5.976
ChromaGAN [32]	70.374	60.000	-10.374
InstColor [31]	68.422	65.000	-3.422
DeOldify [1]	72.364	68.000	-4.364
ColTran [20]	67.210	65.000	-2.210
ToVivid [34]	73.816	65.000	-8.816
BigColor	**76.516**	**74.000**	-2.516

real color images and synthesized color images [12]. The classification accuracy measures whether a classifier trained on natural color images, specifically the pretrained ResNet50-based classifier [11], can predict the correct classes of synthesized color images which were used in CIC [39]. Tab. 1 shows that BigColor outperforms the previous methods with significant margins across all tested metrics with and without the color-augmentation scheme.

In-the-wild Images with Complex Structures We test the robustness of BigColor specifically on challenging in-the-wild images with complex structures. To this end, we select 100 challenging images selected from the ImageNet1K validation set which contain as many humans as possible using an off-the-shelf object detector [35], assuming the proportionality between the number of people and the image complexity. On the curated dataset with 100 samples, Tab. 2 shows the classification accuracy of the synthesized color images for all the methods. BigColor again achieves the best performance with only a 2.5% accuracy drop from the whole-data evaluation. Our performance drop of 2.5% is at the same level of the ground-truth case, where real color images are used to obtain the classification accuracy. We refer to the Supplemental Document for further quantitative and qualitative evaluations.

User Study We conducted a user study to investigate the perceptual preference of colorization methods using Amazon Mechanical Turk (AMT). Specifically, 33 subjects are presented with 100 input and colorized images randomly selected from the ImageNet 1K validation set. The subjects choose the best-restored color image among the results obtained with different methods [39,32,31,1,20,34]. Fig. 6 shows that users clearly prefer BigColor over the state-of-the-art methods. More details can be found in the Supplemental Document.

4.2   Ablation Study

We conduct extensive ablation studies to assess BigColor in details by using 10% of the ImageNet training images amounting to 100 image classes.
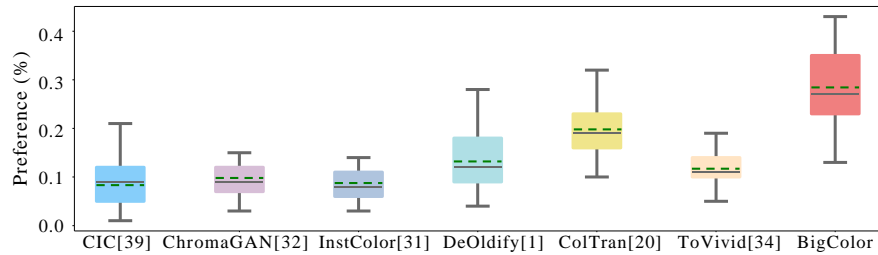
Fig. 6. We conduct a user study to evaluate the preference for colorization results. In all tested metrics, our method outperforms the other methods. The dashed green line and the bold gray line inside the bars are the mean and the median respectively.
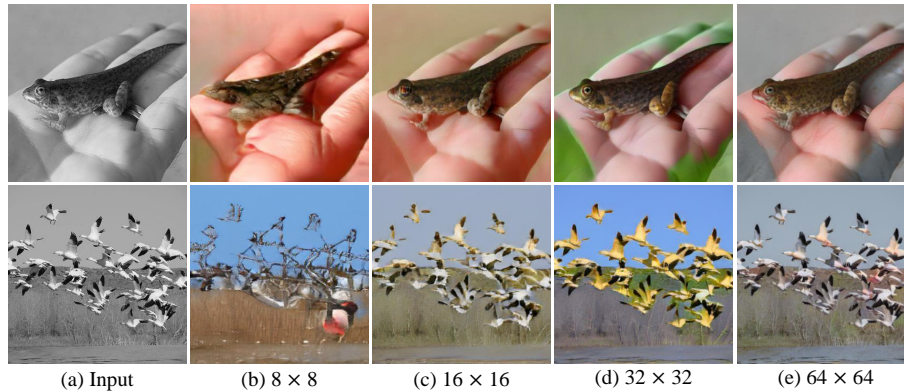

(a) Input   (b) 8 × 8   (c) 16 × 16   (d) 32 × 32   (e) 64 × 64

Fig. 7. The resolution of the spatial feature f plays an important role for maintaining a balance between keeping the spatial structure of the input and providing the degree of freedom for synthesizing color. We empirically chose 16×16 as the best configuration.

Resolution of the Spatial Feature We evaluate the impact of the resolution of the spatial feature f. Fig. 7 shows the colorization results with varying spatial resolutions of the feature f from 8×8 to 64×64. As the spatial resolution increases, the synthesized color images can exploit more structural information of the input image for colorization. However, a large spatial resolution could harm the colorization results as it reduces the capacity of the generator with fewer layers. We chose 16×16 as the spatial resolution of the feature f.

Initialization with a Pretrained Generative Prior We initialize our generator and discriminator using the BigGAN pretrained model in order to leverage the learned structure-color distribution of natural images. Fig. 8 and Tab. 3 show that the pretrained initialization improves performance over the training-from-scratch alternatives with random initialization. Specifically, we test all four combinations of the generator-discriminator initialization settings with and without the pretrained initialization. The qualitative and quantitative results indicate that BigColor successfully exploits the pretrained information in the BigGAN
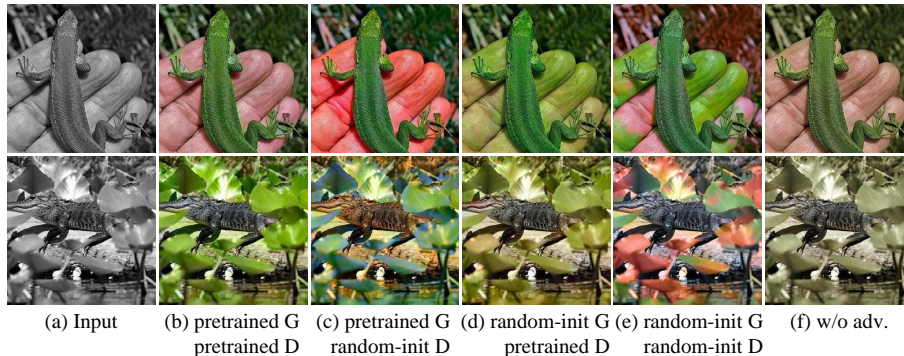
| (a) Input | (b) pretrained G pretrained D | (c) pretrained G random-init D | (d) random-init G pretrained D | (e) random-init G random-init D | (f) w/o adv. |

Fig. 8. We evaluate the impact of initializing the generator G and the discriminator D with the pretrained BigGAN model. Compared to the random initialization, pretrained initialization results in vivid and natural colorization. Also, adversarial training is critical to achieving vivid colorization without desaturation.

Table 3. Initialization with the pretrained BigGAN model provides quantiatively better results in terms of FID and classification accuracy.

Model	G	D	G	D	G	D	G	D	w/o adv.
Pretrained	✓	✓	✓	-	-	✓	-	-	
FID [12]	**5.714**		8.058		6.852		7.201		7.692
Class. Acc.	**81.44**		78.96		80.78		80.60		75.52

generator and the discriminator. We also confirmed the importance of including the adversarial loss to achieve vivid colorization.

Encoder Architecture We considered two main factors for designing our encoder architecture: extracting image structure and exploiting class information. We found that the residual blocks and class-conditioned batch normalization in the original BigGAN generator are essential for robust image colorization as shown in Tab. 4. Specifically, residual blocks transfer structural information and the class-conditioned batch normalization extracts the class-specific spatial feature.

Color Augmentation We experimentally evaluate the impact of color augmentation on the real color images fed to the discriminator. To this end, we compare the FID score and the classification accuracy on 1000 classes of the ImageNet with and without the color augmentation, which shows clear improvements in both metrics as shown in Tab. 5. We also test applying the color augmentation on the synthesized image from the generator as a post-processing after training. This does not consider image semantics, resulting in unnatural colorization as indicated by the FID and the classification scores. In contrast, augmenting the discriminator input enables us to effectively learn the vivid and semantically correct color distribution of the real images. More discussion with qualitative examples of the color augmentation is provided in the Supplemental Document.

Table 4. We analyze our encoder architecture in details to provide insight on the importance of each encoder component: batch normalization (BN), class-conditioned batch normalization (CBN), residual learning (RL). The encoder with residual path and class-conditioned batch normalization shows the best result in terms of FID.

	w/o BN	w/ BN	w/ CBN
w/o RL.	6.523	7.286	5.974
w/ RL.	5.980	5.854	**5.714**

Table 5. Augmenting the real color image fed to the discriminator improves the colorization performance measured in FID and classification accuracy. Our experiments also confirm that directly augmenting the synthesized color degrades the colorization performance. Disc. and Gen. denote the color augmentation on the real color image fed to the discriminator and the generated color image respectively.

Color Aug.	Disc.	Gen.	Disc.	Gen.	Disc.	Gen.	Disc.	Gen.
	✓	-	✓	✓	-	-	-	✓
FID	**1.243**		1.604		1.288		1.621	
Class. Acc.	**76.516**		76.282		76.302		76.238	

4.3   Multi-modal Colorization

BigColor is capable of synthesizing diverse colorization results for an input grayscale image as shown in Fig. 9. We can sample random code z that is injected into the generator to synthesize diverse color images. In addition, we can also alter the class vector c to generate class-specific colorization results, for instance by using the class codes of different classes of birds to colorize an input bird image as shown in the second row in Fig. 9.

4.4   Black-and-White Photo Restoration

Fig. 10 shows the colorization results of BigColor for old monochromatic photographs with arbitrary resolutions and aspect ratios. Note that BigColor is not limited to a specific input resolution owing to the convolutional spatial feature f with a variable spatial resolution. In contrast, conventional GAN-inversion methods [34,27] use a spatially-flattend latent code, enforcing the spatial resolution to be fixed.

5   Conclusion

We propose BigColor, a robust image colorization method using a generative color prior for in-the-wild images with complex structures. We exploit the spatial structure of an input grayscale image using a convolutional encoder, effectively enlarging the representation space of a generator compared to the conventional colorization methods using GAN inversion. Jointly optimizing the encoder-generator module with a discriminator allows us to learn a generative color prior where the generator focuses on synthesizing colors on top of the extracted spatial-structure feature. We extensively assess BigColor in qualitative and quantitative manners and demonstrate that BigColor outperforms existing state-of-the-art methods.
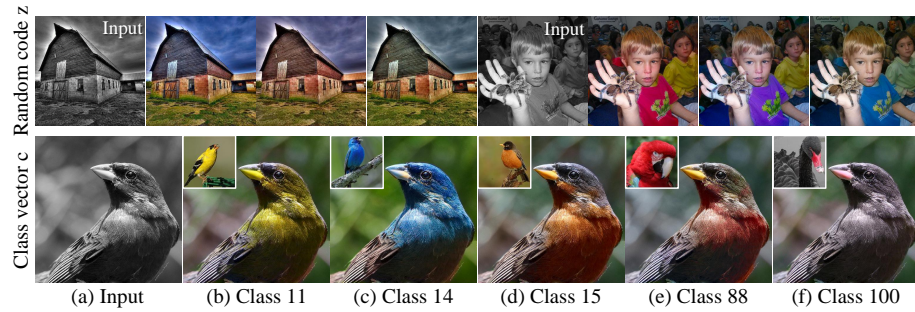
Random code z

Class vector c

(a) Input   (b) Class 11   (c) Class 14   (d) Class 15   (e) Class 88   (f) Class 100

Fig. 9. BigColor supports multi-modal image colorization by sampling the random code z or using different class vectors c which can be estimated from the reference images shown in the insets. The class indices estimated from the reference images are shown below each of the colorization results.

Fig. 10. We apply BigColor to old monochromatic photographs of diverse resolutions and aspect ratios. Left-top to right-bottom: *Albert Einstein at Princeton University, Charlie Chaplin on the movie 'The Kid'(1921), Marilyn Monroe, Photo by Ansel Adams of Yosemite.*

Limitations Our method is not free from limitations. The spatial resolution of the extracted feature f determines the structural details that can be maintained for the color synthesis procedure. Thus, tiny regions might be overlooked in the colorization process. Also, we rely on the BigGAN class code which may not be perfectly estimated for challenging 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. Chan, K.C., Wang, X., Xu, X., Gu, J., Loy, C.C.: Glean: Generative latent bank for large-factor image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14245–14254 (2021)
4. Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: European conference on computer vision. pp. 126–139. Springer (2008)
5. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE international conference on computer vision. pp. 415–423 (2015)
6. Chia, A.Y.S., Zhuo, S., Gupta, R.K., Tai, Y.W., Cho, S.Y., Tan, P., Lin, S.: Semantic colorization with internet images. ACM Transactions on Graphics (TOG) **30**(6), 1–8 (2011)
7. Deshpande, A., Rock, J., Forsyth, D.: Learning large-scale automatic image colorization. In: Proceedings of the IEEE international conference on computer vision. pp. 567–575 (2015)
8. Gu, J., Shen, Y., Zhou, B.: Image processing using multi-code gan prior. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3012–3021 (2020)
9. Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H.: Image colorization using similar images. In: Proceedings of the 20th ACM international conference on Multimedia. pp. 369–378 (2012)
10. 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)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
13. Huang, Y.C., Tung, Y.S., Chen, J.C., Wang, S.W., Wu, J.L.: An adaptive edge detection based colorization algorithm and its applications. In: Proceedings of the 13th annual ACM international conference on Multimedia. pp. 351–354 (2005)
14. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Transactions on Graphics (ToG) **35**(4), 1–11 (2016)
15. Kang, K., Kim, S., Cho, S.: Gan inversion for out-of-range images with geometric transformations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13941–13949 (2021)
16. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
17. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)

18. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8110–8119 (2020)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer. In: International Conference on Learning Representations (2021)
21. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: European conference on computer vision. pp. 577–593. Springer (2016)
22. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM SIGGRAPH 2004 Papers, pp. 689–694 (2004)
23. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
24. Liu, X., Wan, L., Qu, Y., Wong, T.T., Lin, S., Leung, C.S., Heng, P.A.: Intrinsic colorization. In: ACM SIGGRAPH Asia 2008 papers, pp. 1–9 (2008)
25. Luo, X., Zhang, X., Yoo, P., Martin-Brualla, R., Lawrence, J., Seitz, S.M.: Time-travel rephotography. ACM Transactions on Graphics (Proceedings of ACM SIG-GRAPH Asia 2021) **40**(6) (12 2021)
26. Menon, S., Damian, A., Hu, S., Ravi, N., Rudin, C.: Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 2437–2445 (2020)
27. Pan, X., Zhan, X., Dai, B., Lin, D., Loy, C.C., Luo, P.: Exploiting deep generative prior for versatile image restoration and manipulation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
28. Qu, Y., Wong, T.T., Heng, P.A.: Manga colorization. ACM Transactions on Graphics (TOG) **25**(3), 1214–1220 (2006)
29. 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)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
31. 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)
32. 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)
33. Wang, X., Li, Y., Zhang, H., Shan, Y.: Towards real-world blind face restoration with generative facial prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9168–9178 (2021)
34. 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)
35. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. `https://github.com/facebookresearch/detectron2` (2019)
36. Xu, K., Li, Y., Ju, T., Hu, S.M., Liu, T.Q.: Efficient affinity-based edit propagation using kd tree. ACM Transactions on Graphics (TOG) **28**(5), 1–6 (2009)
37. Yang, T., Ren, P., Xie, X., Zhang, L.: Gan prior embedded network for blind face restoration in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 672–681 (2021)

38. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. IEEE transactions on image processing **15**(5), 1120–1129 (2006)
39. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision. pp. 649–666. Springer (2016)