PCCNet

Liu, Xiaying; Yang, Ping; Telea, Alexandru C.; Kosinka, Jiri; Wu, Zizhao

Published in:
Computer Graphics International 2023

Publication date:
2023

PCCNet: A Few-Shot Patch-wise Contrastive Colorization Network

Xiaying Liu[1], Ping Yang[1], Alexandru C. Telea[2], Jiří Kosinka[3], and Zizhao Wu[1⋆]

[1] Hangzhou Dianzi University, China;
{xiayingliu,yangping,wuzizhao}@hdu.edu.cn
[2] Utrecht University, the Netherlands; a.c.telea@uu.nl
[3] University of Groningen, the Netherlands; j.kosinka@rug.nl

Abstract. Few-shot colorization aims to learn a model to colorize images with little training data. Yet, existing models often fail to keep color consistency due to ignored patch correlations of the images. In this paper, we propose PCCNet, a novel Patch-wise Contrastive Colorization Network to learn color synthesis by measuring the similarities and variations of image patches in two different aspects: inter-image and intra-image. Specifically, for inter-image, we investigate a patch-wise contrastive learning mechanism with positive and negative samples constraint to distinguish color features between patches *across* images. For intra-image, we explore a new intra-image correlation loss function to measure the similarity distribution which reveals structural relations between patches *within* an image. Furthermore, we propose a novel color memory loss that improves the accuracy of the memory module to store and retrieve data. Experiments show that our method allows the correctly saturated color to spread naturally over objects and also achieves higher scores in quantitative comparisons with related methods.

Keywords: Colorization · Contrastive Learning · Memory Networks.

1 Introduction

Image colorization aims to create color images from grayscale ones in the most natural manner possible. Existing methods [30, 12] typically build models that allow coloring any kind of grayscale images by training them on large, generic, image datasets such as ImageNet [1] or COCO [14]. However, such models tend to output an average, desaturated result, where the specific hues one would ideally get can be lost. Other artifacts include the inability to consider rare exemplars. For instance, flowers are often colorized red, while in reality they can have a wide spectrum of hues. Few-shot colorization aims to address this by enabling the model to remember what color an object should be with limited training data (few images of a few specific classes). In this category, Yoo *et al*. [28] proposed the MemoPainter colorization model, which can generate

⋆ Corresponding author

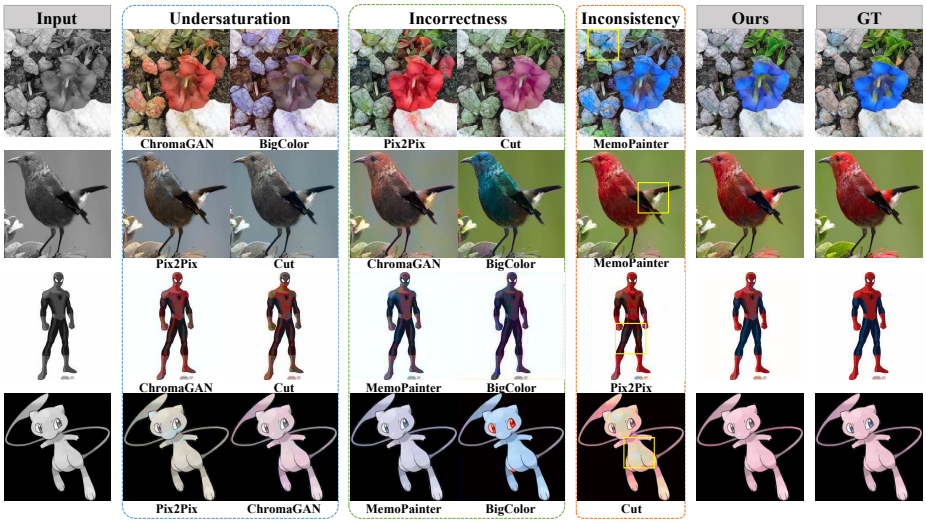


Fig. 1. Compared with existing colorization methods (ChromaGAN [24], BigColor [11], Pix2Pix [6], Cut [19] and MemoPainter [28]), PCCNet has distinctive superiority in ensuring color saturation, color correctness and color consistence.

compelling results. However, the color propagation *within* an image can appear unnatural due to the lack of attention to the image patches.

Three main challenges exist when performing few-shot colorization: Colors should be (1) *consistent*, *i.e.*, they should appear visually natural and the image should avoid mottled (mixed) hues and color bleeding artifacts. Also, colors should be (2) *correct*, meaning that the generated colors should match the ground-truth ones, even if the model was trained with only a few examples exhibiting some specific type of imagery and colors. Finally, colors should be (3) *saturated*, which means the colors should be bright rather than slightly gray.

In this work, we propose a novel patch-wise contrastive colorization network, dubbed PCCNet, to address the above three challenges; see Fig. 1. Specifically, to make sure the produced colors are *consistent* and also visually natural, our method investigates a patch-wise contrastive learning mechanism to learn the patch correlations *across* images. Further, our method explores a correlation loss function in order to learn the patch correlations *within* an image. More concretely, we measure intra-image patches correlations from the perspective of patch data distribution similarity. Additionally, the color information stored in the memory module guides the coloring process to remember the *correct* colors and ensure the generation of *saturated* colors.

To summarize, our main contributions are:

- We first apply patch-wise contrastive learning to image colorization with stronger supervision achieved by using a self-supervised approach.
- We investigate a patch-wise contrastive loss and an intra-image correlation loss to learn patch correlations in two aspects: inter-image and intra-image.

– We use a memory module with a new training strategy to remember the colors of rare objects, allowing for few-shot or one-shot learning.

2 Related Work

Many works have been proposed on learning automatic colorization models using large numbers of image pairs (grayscale or color). Zhang *et al*. [30] treated the regression colorization problem as a classification one to generate saturated results. Subsequent improvements included using two-branch dual-task structures [24, 29] and three-branch triple-task structures [9, 8] to add classification, fine-grained semantic parsing, or both, to the colorization process. Su *et al*. [21] applied object-level colorization to clearly separate objects and background. MemoPainter [28] used a memory network to help remember the color of objects. Hong *et al*. [4] proposed an iterative generative model to attain diversity and possible colorization. Recent methods [26, 11] used the rich color prior encapsulated in a GAN to guide the colorization network to generate diverse color images. Transformer-based architectures [12, 7] improved colorization by focusing on global information. In addition, the exemplar-based image colorization [13] and the line drawing with colorization [22] had also attracted the attention of scholars. In this context, our proposed PCCNet is an automatic colorization model which makes better use of the patch correlations focusing on solving a few-shot learning problem.

3 Proposed Method

Image colorization aims to recover the channels $(a, b) \in \mathbb{R}^{H\times W\times 2}$ from the channel $L \in \mathbb{R}^{H\times W\times 1}$, where (L, a, b) is specified in the CIE *Lab* color space. Our PCCNet consists of three main modules: patch sampling (P), color memory (M), and colorization (C), see Fig. 2.

3.1 Patch Sampling Module

The patch sampling module P consists of an encoder and an F sub-net (see Fig. 2). The encoder is identical to the one used by the colorization module (see Sec. 3.3) and also shares the latter's weights. Multi-scale features are obtained from the encoder and then fed to the F sub-net. We pass the *(a,b)* channels through P to obtain the embedding vectors $\mathbf{w}_k$ and $\mathbf{z}_k$ which represent image patches. The index k, $1 \leq k \leq 256$, indicates the location of the patches in the image. For each patch of the output image, there is a corresponding patch of the ground-truth image as its positive example, labeled as the positive pair $(\mathbf{w}, \mathbf{z}^+) \sim p_{\mathbf{wz}^+}$. For negative examples, the remaining $\mathbf{z}_k$ except for $\mathbf{z}^+$ are all negatives $\mathbf{z}^- \sim p_Z$.

We improve the performance in terms of optimizing the negative sampling strategy and removing negative-positive coupling (NPC) effects [27] of InfoNCE.

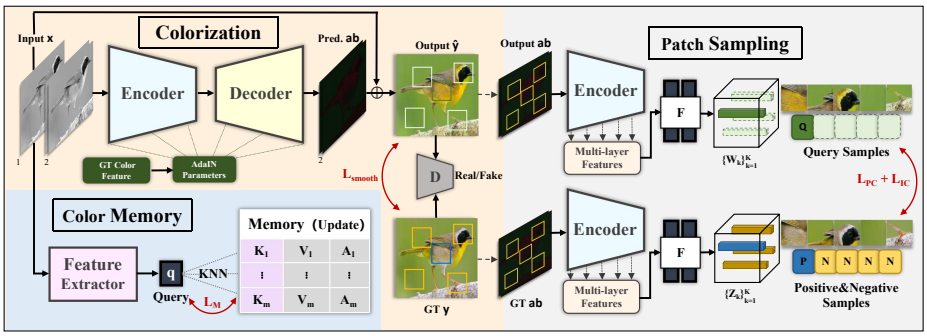
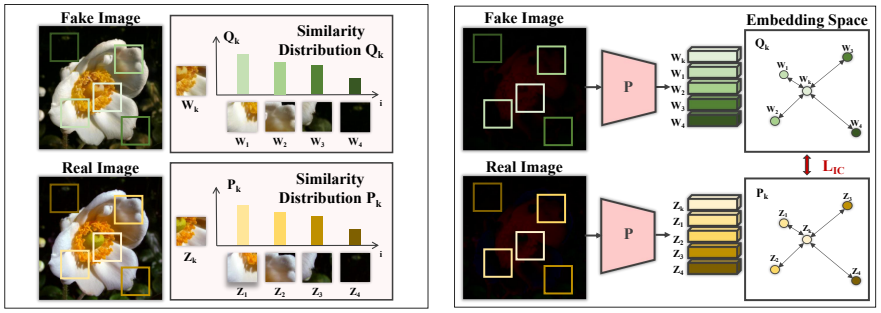


Fig. 2. Overview of the PCCNet framework. C (colorization) synthesizes a colorized image $\hat{y}$ form the grayscale input x in collaboration with P (patch sampling) and M (color memory). During training, our model continuously updates the memory while modulating C with the color feature of the ground truth, and also extracts patches from P to calculate two patch-wise losses. During testing, our model retrieves the nearest color feature from M as a condition to guide C.

There are some easy negatives, such as $\mathbf{z}_4$ in Fig. 3a, that do not contribute significantly to the comparison mechanism. To avoid being overly influenced by those easy-negatives, we use the method of hard-negative sampling modeled by the von Mises-Fisher distribution [20]. The distribution q_{Z^-} is defined as

$$q_{Z^-}(\mathbf{z}^-) \propto e^{\beta \mathbf{z}^+ \cdot \mathbf{z}^-} p_Z(\mathbf{z}^-), \tag{1}$$

where $\cdot$ denotes dot product and the *concentration parameter* β [16] controls the hardness of negative sampling.



(a) For complete color images (b) For (a, b) channels

Fig. 3. Patch correlation through similarity distribution.

Patch-wise Contrastive Loss L_{PC}. The NPC effect refers to the diminishing gradient of InfoNCE by easy negative and positive samples, which in turn makes training harder. To prevent the NPC effect, we formulate this loss, called L_{PC},

by using a decoupled InfoNCE loss, measuring the correlation between image patches, as

$$L_{\mathrm{PC}} = E_{(\mathbf{w},\mathbf{z}^+)\sim p_{\mathbf{wz}^+}}\left[-\log\frac{e^{\mathbf{w}\cdot\mathbf{z}^+/\tau}}{NE_{\mathbf{z}^-\sim q_{\mathbf{z}^-}}\left[e^{\mathbf{w}\cdot\mathbf{z}_i^-/\tau}\right]}\right], \tag{2}$$

where N denotes the number of negative examples and the *temperature parameter* τ is set to 0.07. In terms of implementation, we use

$$\begin{aligned} E_{\mathbf{z}^-\sim q_{Z^-}}\left[e^{\mathbf{w}\cdot\mathbf{z}^-/\tau}\right] &= E_{\mathbf{z}^-\sim p_Z}\left[e^{\mathbf{w}\cdot\mathbf{z}^-/\tau}\frac{q_{Z^-}(\mathbf{z}^-)}{p_Z(\mathbf{z}^-)}\right] \\ &= E_{\mathbf{z}^-\sim p_Z}\left[e^{\mathbf{w}\cdot\mathbf{z}^-/\tau}e^{\beta\mathbf{z}^+\cdot\mathbf{z}^-}\right]. \end{aligned} \tag{3}$$

Intra-image Correlation Loss L_{IC}. Patches at different positions have different representations. For example, in Fig. 3a, the central patch represents the stamen; other patches represent the petals or the background. Different patches have different similarity distributions. The (a, b) channels of this image (Fig. 3b) show a similar story. For the real-image (a, b) channels, any patch Z_k has a different similarity with the others, as captured by the similarity distribution P_k

$$P_k(i) = \frac{e^{\mathbf{z}_k\cdot\mathbf{z}_i}}{\sum_{j=1}^{K} e^{\mathbf{z}_k\cdot\mathbf{z}_j}}. \tag{4}$$

A patch $\mathbf{w}_k$ in the (a, b) channels of the generated image has the similarity distribution Q_k with the other patches $\mathbf{w}_i$ given by

$$Q_k(i) = \frac{e^{\mathbf{w}_k\cdot\mathbf{w}_i}}{\sum_{j=1}^{K} e^{\mathbf{w}_k\cdot\mathbf{w}_j}}, \tag{5}$$

which should match P_k as much as possible. Given the above, we define the intra-image correlation loss L_{IC} which measures intra-image patches correlations in terms of the Jensen-Shannon divergence

$$L_{\mathrm{IC}} = \sum_{k=1}^{K} \mathrm{JSD}(P_k \| Q_k). \tag{6}$$

3.2 Color Memory Module

The color memory module M consists of a feature extractor and a memory (see Fig. 2). Inspired by the memory module of [10], we design the memory of M as follows

$$\text{Memory} = (K_1, V_1, A_1), (K_2, V_2, A_2), \cdots, (K_m, V_m, A_m), \tag{7}$$

where K and V represent the spatial and color features extracted from the training data, A tracks the ages of the infrequent (K, V) pairs, and m denotes

the adjustable memory size. The model can retrieve the nearest color feature to guide the coloring process during testing, and the feature extractor extracts spatial features $s \in \mathbb{R}^{512}$ from grayscale images. The color features $c \in \mathbb{R}^{313}$ are obtained from color images by using a quantized (a, b) value method [30].

The ability to extract spatial features determines whether the corresponding color can be found. We redefine the color memory module loss for unsupervised training to better extract spatial features. The triplet loss used in MemoPainter [28] optimizes the within-class similarity and between-class similarity by means of average force. We use the circle loss [23], which adds the weights to control the gradient contribution of the positive and negative keys, so as to obtain a more discriminative model than using the triplet loss. We describe the specific definition of the color memory module loss in Sec. 3.4.

3.3 Colorization Module

The colorization module C is built upon the baseline of CGAN [17] which consists of a generator and a discriminator.

Generator G. The input x is a grayscale image, *i.e.*, the L channel, and the output $\hat{y}$ predicts the real image y. Our generator (Fig. 2) contains an encoder consisting of downsample blocks and residual blocks, and a decoder consisting of residual blocks and upsample blocks. We concatenate the L channel and a duplicate L channel before feeding it to the encoder. This ensures that the encoder can be used directly by P to extract multi-scale features. We use AdaIN [5] to treat color features c retrieved from the color memory module M as a guide for the colorization module. Specifically, we compute the affine scaling y_1 and shift y_2 of AdaIN by MLP from c and let it act on the AdaIN layer as

$$\mathrm{AdaIN}(x_{\mathrm{now}}, c) = y_1 \left(\frac{x_{\mathrm{now}} - \mu(x_{\mathrm{now}})}{\sigma(x_{\mathrm{now}})} \right) + y_2, \tag{8}$$

where μ and σ denote average and standard deviation of the output of the previous convolutional layer x_{now}.

Discriminator D. The discriminator uses color features as a condition to distinguish between real images and output images. We use the Markovian discriminator [6] to pay attention to image details, avoiding extreme outputs of the discriminator. Given that our input images are 256×256 pixels, we use a receptive field of size 70×70. The discriminator D outputs a matrix of size 32×32 in which each element represents the classification (true/fake) of each patch.

3.4 Objective Functions

Color Memory Loss L_M. For the unsupervised learning of the feature extractor, we use the circle loss defined as

$$L_M(S_n, S_p) = \log \left(1 + \sum_{i=1}^{K} \sum_{j=1}^{L} e^{\gamma(\alpha_n^j S_n^j - \alpha_p^i S_p^i)} \right), \tag{9}$$

where S_n is the between-class similarity, and S_p is the within-class similarity. Negative and positive keys $K[n_b]$ and $K[n_p]$ are given by $\mathrm{KL}(V[n_b] \parallel v) > \delta_1$ and $\mathrm{KL}(V[n_p] \parallel v) < \delta_1$ respectively, where δ_1 is the user preset color threshold. K and L are the number of positive, respectively negative, keys; $\gamma = 256$ is a scale factor; and α_p^i and α_n^j are the weights of positive and negative samples, respectively, defined as

$$\alpha_p^i = [O_p - S_p^i]_+ \quad \text{and} \quad \alpha_n^j = [S_n^j - O_n]_+, \tag{10}$$

where $[\cdot]_+$ denotes 'cut off at zero' to ensure positive values, $O_p = 1 + m$, $O_n = -m$, and m is a hyperparameter set to 0.25. Minimizing L_M allows the feature extractor to extract a query q with the smallest possible between-class similarity and the largest possible within-class similarity.

Colorization and Patch Sampling Loss. We combine four losses to jointly train C and P: (1) patch-wise contrastive loss L_{PC} (see Sec. 3.1); (2) intra-image correlation loss L_{IC} (see Sec. 3.1); (3) content loss L_{smooth}; and (4) adversarial loss L_{LSGAN}, as follows.

Content Loss L_{smooth}. We use the smooth L_1 metric between the generated $\hat{y}$ and the ground truth image y, *i.e.*,

$$L_{\mathrm{smooth}}(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{if} \quad |y - \hat{y}| \leqslant \delta_2 \\ \delta_2 |y - \hat{y}| - \frac{1}{2}\delta_2^2 & \text{otherwise,} \end{cases} \tag{11}$$

to make training more robust to outlier images.

Adversarial Loss L_{LSGAN}. LSGAN [15] uses a least squares loss to mitigate the vanishing gradient problem. We follow a LSGAN design with the discriminator and generator losses given by

$$\begin{aligned} L_{\mathrm{LSGAN}}^D =& \frac{1}{2} E_{y \sim P_{\mathrm{data}}}[(D(y, c) - 1)^2] + \frac{1}{2} E_{x \sim P_{\mathrm{data}}}[D(G(x, c), c)^2], \\ L_{\mathrm{LSGAN}}^G =& \frac{1}{2} E_{x \sim P_{\mathrm{data}}}[(D(G(x, c), c) - 1)^2], \end{aligned} \tag{12}$$

respectively. Putting it all together, the final loss functions of C and P amount to a generator loss L_G and a discriminator loss L_D which are defined as

$$\begin{aligned} L_G &= \lambda_1 L_{\mathrm{PC}} + \lambda_2 L_{\mathrm{IC}} + \lambda_3 L_{\mathrm{smooth}} + \lambda_4 L_{\mathrm{LSGAN}}^G, \\ L_D &= L_{\mathrm{LSGAN}}^D. \end{aligned} \tag{13}$$

4 Experiments

We now present datasets and metrics used to evaluate our method (Sec. 4.1), several implementation details (Sec. 4.2), and the results of our evaluation and comparisons (Sec. 4.3), and ablation studies (Sec. 4.4).

4.1 Datasets and Evaluation Metrics

We measure the performance of our model on four datasets, namely **Oxford102 Flowers** [18], **Bird100** [25], **Hero** [28], and **Pokemon**[4], where Hero performs few-shot colorization, and Pokemon performs one-shot or zero-shot colorization. The FID score [3] is used to evaluate the colorization image quality. LPIPS [31] evaluates the perceptual similarity which is more consistent with human perception. We provide the evaluation measured by PSNR as a reference, although experiments show that PSNR prefers grayish images. The colorfulness score [2] reflects the degree of color brightness.

4.2 Implementation Details

We implemented PCCNet with PyTorch and trained our models on an NVIDIA RTX 2060 GPU using the Adam optimizer with the learning rate decaying from 10^{-3} to 10^{-6}. We optimize the color memory module M first, then optimize the colorization and patch sampling modules C and P jointly. Please see the Supplementary Material for parameter settings.

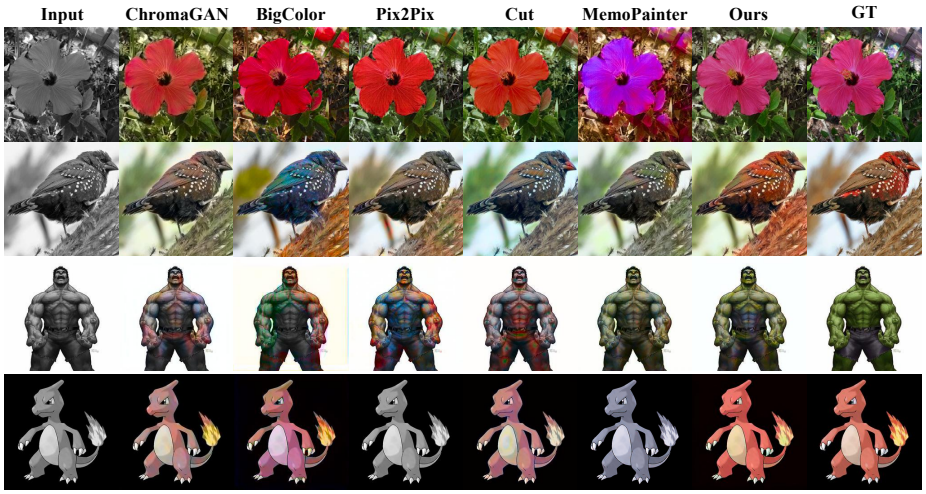


Fig. 4. Qualitative comparison of colorization results on four different datasets.

4.3 Evaluation

Qualitative Comparisons. We compare PCCNet with ChromaGAN [24], BigColor [11], Pix2Pix [6], Cut [19], and MemoPainter [28]. Fig. 4 shows that ChromaGAN and Cut generate images that are grayish, akin to old photos that are not restored well. BigColor and Pix2Pix produce more vibrant results, but the

[4] https://www.kaggle.com/kvpratama/pokemon-images-dataset

lack of supervision of image patches makes color transitions unnatural. Especially in the few-shot cases, *i.e.*, on the Hero datasets, severe unreasonable color fusion phenomena occur. In this case, MemoPainter yields better results due to color memory. However, the results still show color inconsistency and color deviations. Compared with the above methods, our method obtains better visual results. In line with the requirements listed in Sec. 1, our method improves the model's ability to maintain color consistency, color correctness, and color saturation. See the Supplementary Material for more qualitative results.

Table 1. Quantitative evaluation of colorization experiments

Dataset	Oxford102 Flowers				Bird100			
Metric	FID↓	LPIPS↓	PSNR↑	Colorful↑	FID↓	LPIPS↓	PSNR↑	Colorful↑
ChromaGAN	74.306	0.231	18.768	44.756	41.613	0.188	22.302	25.256
BigColor	59.464	0.217	19.031	65.287	38.866	0.229	18.633	48.883
Pix2Pix	46.899	0.217	19.182	52.319	44.495	0.205	21.546	23.413
Cut	44.174	0.213	18.956	59.674	45.266	0.218	20.193	27.146
MemoPainter	59.338	0.272	17.693	66.286	40.011	0.216	21.080	29.012
Ours	**41.206**	**0.211**	**20.142**	**66.868**	**29.836**	**0.181**	**22.321**	**36.278**
Dataset	Hero				Pokemon			
Metric	FID↓	LPIPS↓	PSNR↑	Colorful↑	FID↓	LPIPS↓	PSNR↑	Colorful↑
ChromaGAN	104.171	0.098	23.467	26.155	92.050	0.125	**21.811**	24.101
BigColor	143.791	0.147	21.238	21.841	111.534	0.168	18.233	34.626
Pix2Pix	133.237	0.109	21.981	**36.931**	94.265	0.140	21.217	10.897
Cut	130.329	0.116	22.052	24.709	90.807	0.136	20.762	28.180
MemoPainter	139.494	0.105	22.311	23.747	89.816	0.138	20.646	26.520
Ours	**90.252**	**0.081**	**24.372**	33.528	**64.023**	**0.124**	21.703	**32.416**

Quantitative Evaluation. Table 1 shows the results of our quantitative evaluation. For all datasets, our method achieves lower FID and LPIPS scores. This indicates that our method is able to generate more natural and realistic color images that are closer to the ground truth. The two exceptions to the above are Pix2Pix and ChromaGAN, where Pix2Pix achieves a higher Colorfulness score than our method on the Hero dataset and ChromaGAN achieves a higher PSNR score than our method on the Pokemon dataset. However, as mentioned earlier, Pix2Pix exhibits a 'color fusion' phenomenon for this dataset, and ChromaGAN exhibits a 'desaturation' phenomenon for the Pokemon dataset (see again Fig. 4 or more visual results in the Supplementary Material). While this leads to higher scores, we argue that the results of Pix2Pix and ChromaGAN are less similar to the ground truth than ours.

4.4 Ablation Study

As described in Sec. 3, our method has three main components which can be seen as the key heuristics we use to drive our colorization: the color memory

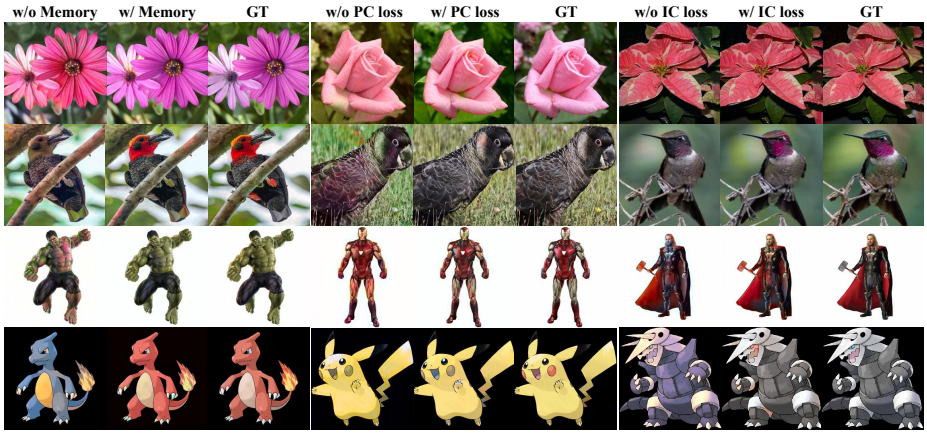


Fig. 5. Qualitative comparisons of ablation studies on PCCNet.

module, the patch-wise contrastive loss L_{PC}, and the intra-image correlation loss L_{IC}. To further understand the effect of these components, we performed a series of ablation studies. Figure 5 visually shows the effect of each of our modules. M helps to memorize the colors, and the network without M generates images with biased colors. Further adding L_{PC} and L_{IC} leads to better handling of the details of the images. Hence, we argue that each individual module is of added value to the final colorization. In Table 2, the results show the meaningful improvements by our three components. Adding each component to our network results in varying degrees of improvement on the Oxford102 Flowers dataset. The best results are seen when all components are added. See supplementary material for more ablation experiments.

5 Conclusion

We have presented PCCNet, a new deep learning model for colorization with improved color consistency, color correctness, and color saturation. Our framework relies on a patch-wise contrastive learning mechanism and an intra-image

Table 2. Quantitative comparisons for ablation studies.

Setting			Oxford102 Flowers			
Memory Module	PC Loss	IC Loss	FID↓	LPIPS↓	PSNR↑	Colorful↑
×	✓	✓	42.101	0.213	19.888	58.450
✓	×	✓	48.849	0.217	19.696	65.108
✓	✓	×	42.789	0.214	20.126	66.209
✓	✓	✓	**41.206**	**0.211**	**20.142**	**66.868**

correlation loss for distilling the learning process of the correlation between image patches in two aspects: inter-image and intra-image. We further drive our colorization by a memory module which favors the learning of outlier color patterns present in the training images. Qualitative and quantitative comparisons as well as ablation experiments illustrate its effectiveness on various benchmark datasets.

6 Acknowledgements

This work was partially supported by the Zhejiang Provincial Natural Science Foundation of China (LGF21F20012).

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition pp. 248–255 (2009)
2. Hasler, D., Süsstrunk, S.: Measuring colorfulness in natural images. In: IS&T/SPIE Electronic Imaging (2003)
3. 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. In: NIPS (2017)
4. Hong, K., Li, J., Li, W., Yang, C., Zhang, M., Wang, Y., Liu, Q.: Joint intensity-gradient guided generative modeling for colorization. ArXiv **abs/2012.14130** (2020)
5. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. IEEE International Conference on Computer Vision (ICCV) pp. 1510–1519 (2017)
6. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5967–5976 (2017)
7. Ji, X., Jiang, B., Luo, D., Tao, G., Chu, W., Xie, Z., Wang, C., Tai, Y.: Colorformer: Image colorization via color memory assisted hybrid-attention transformer. In: European Conference on Computer Vision (2022)
8. Jin, X., Li, Z., Liu, K., Zou, D., Li, X., Zhu, X., Zhou, Z., song Sun, Q., Liu, Q.: Focusing on persons: Colorizing old images learning from modern historical movies. Proceedings of the 29th ACM International Conference on Multimedia (2021)
9. Jin, Y., Sheng, B., Li, P., Chen, C.L.P.: Broad colorization. IEEE Transactions on Neural Networks and Learning Systems **32**, 2330–2343 (2020)
10. Kaiser, L., Nachum, O., Roy, A., Bengio, S.: Learning to remember rare events. ArXiv **abs/1703.03129** (2017)
11. Kim, G.Y., Kang, K., Kim, S.H., Lee, H., Kim, S., Kim, J., Baek, S.H., Cho, S.: Bigcolor: Colorization using a generative color prior for natural images. In: European Conference on Computer Vision (2022)
12. Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer. ArXiv **abs/2102.04432** (2021)

13. Li, H., Sheng, B., Li, P., Ali, R., Chen, C.L.P.: Globally and locally semantic colorization via exemplar-based broad-gan. IEEE Transactions on Image Processing **30**, 8526–8539 (2021)
14. Lin, T.Y., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision (2014)
15. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. IEEE International Conference on Computer Vision (ICCV) pp. 2813–2821 (2016)
16. Mardia, K.V., Jupp, P.E.: Directional Statistics. John Wiley & Sons, Inc. (2000)
17. Mirza, M., Osindero, S.: Conditional generative adversarial nets. ArXiv **abs/1411.1784** (2014)
18. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. Sixth Indian Conference on Computer Vision, Graphics & Image Processing pp. 722–729 (2008)
19. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision (2020)
20. Robinson, J., Chuang, C.Y., Sra, S., Jegelka, S.: Contrastive learning with hard negative samples. ArXiv **abs/2010.04592** (2020)
21. Su, J.W., kuo Chu, H., Huang, J.B.: Instance-aware image colorization. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 7965–7974 (2020)
22. Sun, Q., Chen, Y., Tao, W., Jiang, H., Zhang, M., Chen, K., Erdt, M.: A gan-based approach toward architectural line drawing colorization prototyping. The Visual Computer **38**, 1283–1300 (2021)
23. Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y.: Circle loss: A unified perspective of pair similarity optimization. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6397–6406 (2020)
24. Vitoria, P., Raad, L., Ballester, C.: Chromagan: Adversarial picture colorization with semantic class distribution. IEEE Winter Conference on Applications of Computer Vision (WACV) pp. 2434–2443 (2020)
25. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. california institute of technology (2011)
26. Wu, Y., Wang, X., Li, Y., Zhang, H., Zhao, X., Shan, Y.: Towards vivid and diverse image colorization with generative color prior. IEEE/CVF International Conference on Computer Vision (ICCV) pp. 14357–14366 (2021)
27. Yeh, C.H., Hong, C.Y., Hsu, Y.C., Liu, T.L., Chen, Y., LeCun, Y.: Decoupled contrastive learning. ArXiv **abs/2110.06848** (2021)
28. Yoo, S., Bahng, H., Chung, S., Lee, J., Chang, J., Choo, J.: Coloring with limited data: Few-shot colorization via memory augmented networks. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11275–11284 (2019)
29. Zhang, J., Xu, C., Li, J., Han, Y., Wang, Y., Tai, Y., Liu, Y.: Scsnet: An efficient paradigm for learning simultaneously image colorization and super-resolution. In: AAAI Conference on Artificial Intelligence (2022)
30. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European Conference on Computer Vision (2016)
31. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 586–595 (2018)