SCSNet: An Efficient Paradigm for Learning Simultaneously Image Colorization and Super-Resolution

Anonymous AAAI submission
Paper ID 528

Abstract

In the practical application of restoring low-resolution gray-scale images, we generally need to run three separate processes of image colorization, super-resolution, and dows-sampling operation for the target device. However, this pipeline is redundant and inefficient for the independent processes, and some inner features could have been shared. Therefore, we present an efficient paradigm to perform ***S***imultaneously Image ***C***olorization and ***S***uper-resolution (SCS) and propose an end-to-end SCSNet to achieve this goal. The proposed method consists of two parts: colorization branch for learning color information that employs the proposed plug-and-play *Pyramid Valve Cross Attention* (PV-CAttn) module to aggregate feature maps between source and reference images; and super-resolution branch for integrating color and texture information to predict target images, which uses the designed *Continuous Pixel Mapping* (CPM) module to predict high-resolution images at continuous magnification. Furthermore, our SCSNet supports both automatic and referential modes that is more flexible for practical application. Abundant experiments demonstrate the superiority of our method for generating authentic images over state-of-the-art methods, *e.g.*, averagely decreasing FID by 1.8↓ and 5.1 ↓ compared with current best scores for automatic and referential modes, respectively, while owning fewer parameters (more than ×2↓) and faster running speed (more than ×3↑).

1 Introduction

In some practical scenarios, *e.g.*, restoration of old photos and artistic creation of gray-scale draft, we can only obtain Low-Resolution (LR) gray-scale images and hope to get more attractive High-Resolution (HR) colorful images. As shown in the top part of Figure 1, the current solution pipeline cascades different methods by mainly three stages: 1) Using automatic or referential colorization model to color the gray-scale image for obtaining visually appealing RGB images. 2) Leveraging Single Image Super-Resolution (SISR) method for learning a nonlinear mapping to reconstruct HR images from LR inputs. 3) Down-sampling the generated HR images to the appropriate resolution for the target device. However, this pipeline is redundant and inefficient for practical use, where image colorization and super-resolution could have shared some common features by one unified network. Also, the device-adapted down-sampling operation in the last stage means that there is redundancy calculation in the SISR stage, which generates HR images at a fixed magnification (usually higher than needed) rather than device-required magnification. As shown in the bottom part of Figure 1, we focus on solving the above problems and designing an efficient paradigm to achieve *Simultaneously image Colorization and Super-resolution* (SCS) task by one unified network. Figure 2 shows our authentic and diversified generation results in both automatic and referential modes, as well as the qualitative and quantitative comparison results with State-Of-The-Art (SOTA) pipelines. Concretely, we propose a novel efficient SCSNet that contains colorization and super-resolution branches.

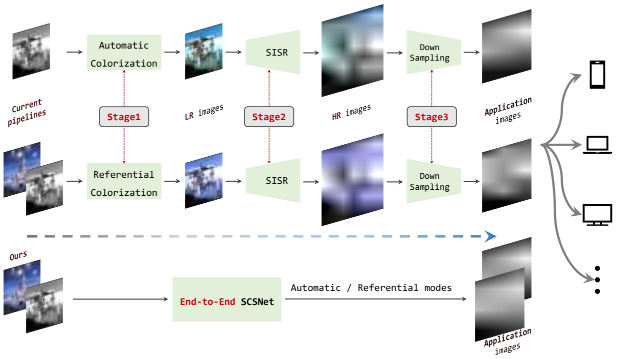

Figure 1. **Comparison between current pipelines and ours for the SCS task**. Current automatic and referential pipelines require three stages: 1) image colorization; 2) image super-resolution; and 3) down-sampling for target devices in different scenarios. Our end-to-end *SCSNet* supports both automatic and referential modes.

For colorization branch, it learns how to predict two missing channels information from the given gray-scale image. Image colorization mainly falls into automatic and referential modes depending on the availability of the reference image. The automatic mode only requires LR gray-scale image that seems intuitive but suffers from poor chromaticity of the generated images, because each semantic object can have various colors and the network tends to average output if applying inappropriate training strategy (Zhang, Isola, and Efros 2016). The referential mode requires an additional reference image for providing semantic color information that

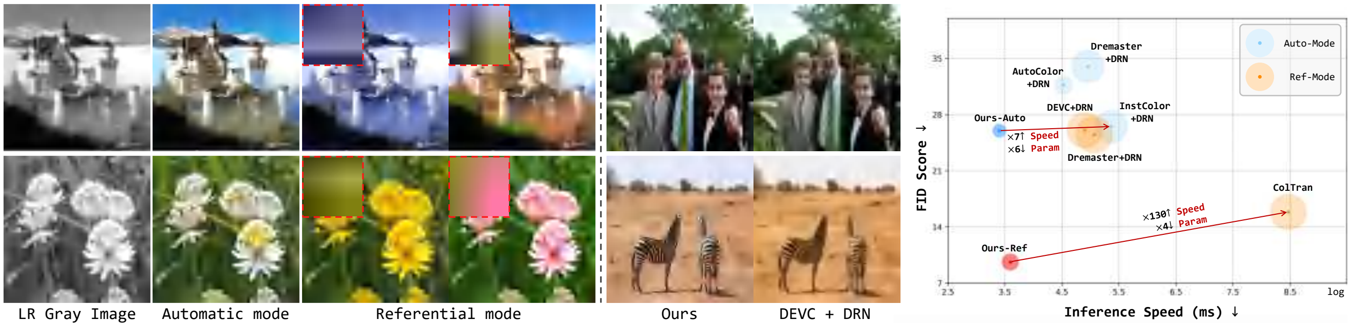

Figure 2. **Samples for simultaneously image colorization and super-resolution with 128×128 gray inputs.** The left part shows our various results for 4× under *automatic* and *referential* modes, while the middle part is for 4× compared with SOTA referential pipeline. Right figure shows the efficiency comparison among our method and SOTA methods, and the circle size represents the parameter number of each method.

is more controllable. A key point in the referential process is how to reasonably merge color information from the referential image into the source image. Some works (He et al. 2018; Zhang et al. 2019a) propose to calculate the correlation matrix to characterize bidirectional similarity between source and reference images, and approaches (Zhao et al. 2019; Xu et al. 2020; Kumar, Weissenborn, and Kalchbrenner 2021) use direct concatenation, AdaIN operation (Huang and Belongie 2017), or transformer module to aggregate information. However, the information interaction of current methods can be error-prone and may lack visual interpretation. Inspired by self-attention (Zhang et al. 2019b), we redesign a plug-and-play *Pyramid Valve Cross Attention* (PVCAttn) module that applies interpretable valves to control the information flow and fuses features at multiple scales. Also, our SCS paradigm supports both modes controlled by a reference switch in the PVCAttn module.

For super-resolution branch, it learns how to reconstruct HR images from LR images. In general, SISR technology is employed to post-process the generated images for better visualization, and almost all current SISR methods only carry out fixed magnification (Wang et al. 2018; Guo et al. 2020), which goes against the natural world with a continuous visual expression. Unlike recent Meta-SR (Hu et al. 2019) that attempts continuous magnification by predicting convolutional weights for each pixel, we propose a more efficient *Continuous Pixel Mapping* (CPM) module to realize arbitrary magnification in a continuous space, without extra down-sampling operation.

So far, we have completed a novel and efficient paradigm to restore colorful HR images in a continuous magnification and make the following three contributions:

- We propose an efficient SCSNet paradigm to perform the SCS task in an end-to-end manner firstly, and abundant experiments demonstrate the superiority of our approach for generating authentic and colorful images.
- A novel plug-and-play *PVCAttn* module is proposed to effectively aggregate color information between source and reference images in an explicable way.
- An elaborate *CPM* module is designed to realize continuous magnification, which is more computation-friendly and suitable for practical application.

2 Related Work

2.1 Image Colorization

Before the advent of CNN-based approaches, Li *et al*. (Deshpande, Rock, and Forsyth 2015) train a quadratic objective function in the chromaticity maps to colorize images. Subsequently, learning-based approaches almost dominate the automatic image colorization (Anwar et al. 2020; Cheng, Yang, and Sheng 2015; Zhang, Isola, and Efros 2016; Larsson, Maire, and Shakhnarovich 2016; Iizuka, Simo-Serra, and Ishikawa 2016). Cheng *et al*. (Cheng, Yang, and Sheng 2015) propose to extract multiple-level feature descriptors to regress pixel values, while Zhang *et al*. (Zhang, Isola, and Efros 2016) quantize the chrominance space into bins. Later works (Deshpande et al. 2017; Messaoud, Forsyth, and Schwing 2018) leverage VAE to learn a low dimensional embedding of color fields, while GAN-based methods (Cao et al. 2017; Vitoria, Raad, and Ballester 2020) introduce adversarial training to generate diverse and authentic colorful images. Moreover, I2C (Su, Chu, and Huang 2020) uses an off-the-shelf object detector to obtain extra object-level features, while Lei *et al*. (Lei and Chen 2019) design a two-stage network successively for colorization and refinement. Even though the above automatic methods perform well, they are uncontrollable and cannot generate various images once trained. This work also considers diversity and controllability when designing the network while retaining the benefits of automatic image colorization.

Differently, referential image colorization requires additional information to guide the generation process. Zou *et al*. (Zou et al. 2019) propose a SECat network that inputs a gray-scale line art and color tag information to produce a quality colored image. Some GAN-based methods (Furusawa et al. 2017; Xian et al. 2018; Sun et al. 2019) use adversarial training to improve the rationality of generated images, while works (Huang, Liao, and Kwong 2020; Lee et al. 2020; Xian et al. 2018) take gray sketch image as input and color it with the aid of the reference image condition. He *et al*. (Iizuka and Simo-Serra 2019) propose a similarity sub-net to compute the bidirectional similarity map between source and reference images. Considering the limitation of one-stage network, methods (Xu et al. 2020; Zhang et al. 2019a) design the coarse-to-fine net-

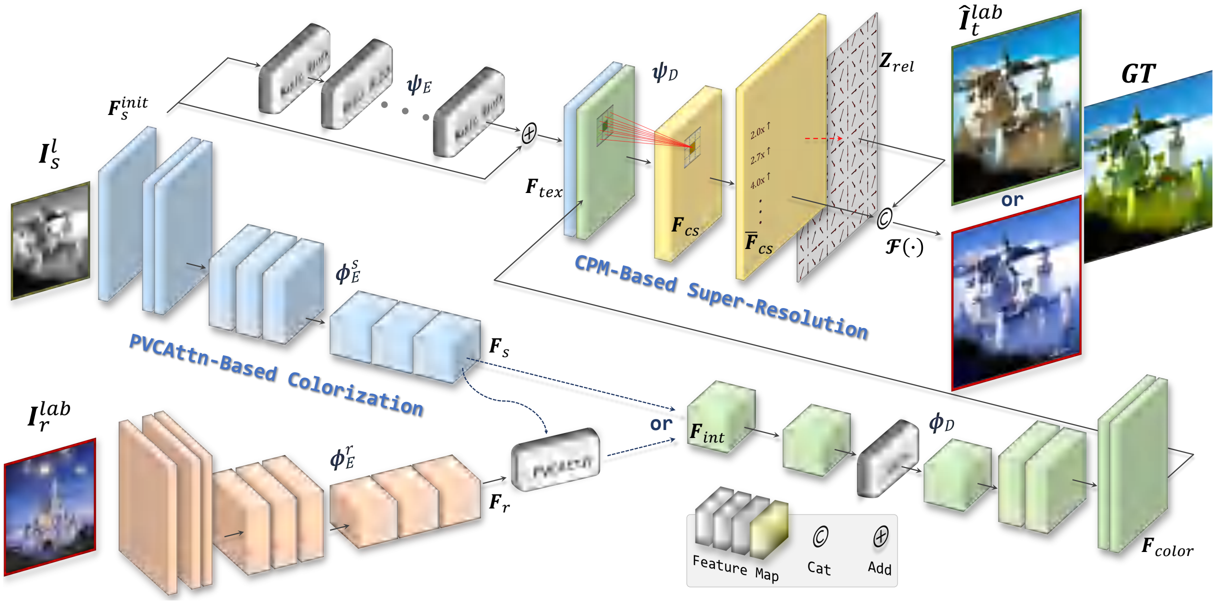

Figure 3. **Overview of the proposed SCSNet**, which consists of a *PVCAttn-based Colorization* branch (ϕ) and a *CPM-based Super-resolution* branch (ψ). Given a low-resolution gray source image $\boldsymbol{I}_s^l$ and a colorful reference image $\boldsymbol{I}_r^{lab}$, the encoders $\boldsymbol{\phi}_E^s$ and $\boldsymbol{\phi}_E^r$ in colorization branch are used to extract corresponding deep features $\boldsymbol{F}_s$ and $\boldsymbol{F}_r$, respectively. PVCAttn module aggregates $\boldsymbol{F}_s$ and $\boldsymbol{F}_r$ to form $\boldsymbol{F}_{int}$ if choosing the referential mode, otherwise $\boldsymbol{F}_{int}$ equaling $\boldsymbol{F}_s$. Subsequent colorization decoder $\boldsymbol{\phi}_D$ restores the color feature to the original resolution $\boldsymbol{F}_{color}$. For super-resolution branch, the encoder $\boldsymbol{\psi}_E$ extracts the residual feature of initial feature $\boldsymbol{F}_s^{init}$, while the CPM module in decoder $\boldsymbol{\psi}_D$ uses mapping function $\boldsymbol{\mathcal{F}}(\cdot)$ to generate target HR image $\hat{\boldsymbol{I}}_t^{lab}$ in a continuous space.

work to improve the performance. Nevertheless, how to reasonably aggregate the referential feature is still a big challenge (Zhao et al. 2019; Xu et al. 2020). Works (Zhang et al. 2019a; Huang, Liao, and Kwong 2020) propose to obtain the correlation matrix whose elements characterize the similarity between the source and reference images. Recently, Gray2ColorNet (Lu et al. 2020) design an attention gating mechanism-based color fusion network, and Kumar *et al*. (Kumar, Weissenborn, and Kalchbrenner 2021) firstly introduce the transformer (Vaswani et al. 2017) structure. However, the above methods are effortless to select incorrect referential information prone to produce visual artifacts, *e.g*., color shift and color patch. To alleviate the problems, we propose a novel PVCAttn module to more effectively aggregate information between source and reference images.

2.2 Single Image Super-Resolution

Since Dong *et al*. (Dong et al. 2015) propose SRCNN for SISR, many CNN-based methods (Lim et al. 2017; Zhang et al. 2018a,b; Ledig et al. 2017; Wang et al. 2018; Guo et al. 2020) with good effects have been proposed. EDSR (Lim et al. 2017) improves performance significantly by removing unnecessary batch normalization in conventional residual networks and designing a new multi-scale deep super-resolution system. Later RCAN (Zhang et al. 2018a) and RDN (Zhang et al. 2018b) improve the residual block, and works (Ledig et al. 2017; Wang et al. 2018) further introduce adversarial loss during the training phase that greatly improves the model's performance. To solve the problem of real-world image matching, works (Cai et al. 2019; Zhang et al. 2019c) contribute new datasets where paired real-world LR-HR images on the same scene are captured. Recently, Guo *et al*. (Guo et al. 2020) propose a novel dual regression scheme for paired and unpaired data, which forms a closed-loop to provide additional supervision. The above methods have achieved good results, but they can only carry out fixed factors for SISR, not producing a continuous display for practical application. Different from Meta-SR that attempts continuous magnification by predicting convolutional weights for each pixel, we design a more efficient *Continuous Pixel Mapping* head to directly regress pixel value with local relative coordinate in a continuous space.

3 Approach

In this paper, a novel efficient paradigm is proposed to complete both automatic and referential image colorization along with SISR simultaneously by one end-to-end network. As depicted in Figure 3, the proposed SCSNet consists of a PVCAttn-based colorization branch for restoring the color information, as well as a CPM-based super-resolution branch for generating high-resolution target image in a continuous space. An initial convolution firstly increases the channel dimension of the low-resolution gray-scale source image: $\boldsymbol{I}_s^l \in \mathbb{R}^{1\times H_s\times W_s} \to \boldsymbol{F}_s^{init} \in \mathbb{R}^{64\times H_s\times W_s}$.

For the colorization branch, encoders $\boldsymbol{\phi}_E^s$ and $\boldsymbol{\phi}_E^r$ are employed to extract corresponding deep features:

$$\begin{aligned}\boldsymbol{F}_s &= \phi_E^s(\boldsymbol{F}_s^{init}) \in \mathbb{R}^{256\times H_s/4\times W_s/4},\\ \boldsymbol{F}_r &= \phi_E^r(\boldsymbol{I}_r^{lab}) \in \mathbb{R}^{256\times H_s/4\times W_s/4}.\end{aligned} \tag{1}$$

We design the branch with two patterns: the automatic mode that directly maps source image feature to output (*i.e*., $\boldsymbol{F}_s \to$

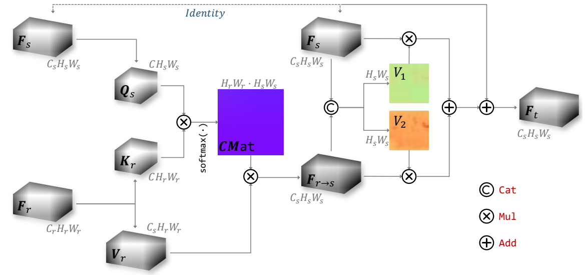

Figure 4. **Schematic diagram of VCAttn**. The module takes the source feature $\boldsymbol{F}_s$ and the reference feature $\boldsymbol{F}_r$ as input and outputs the aggregated target feature map $\boldsymbol{F}_t$ that has the same dimension with $\boldsymbol{F}_s$. *Identity* means the shared feature map.

$\boldsymbol{F}_{int}$), and the referential mode that employs the proposed plug-and-play PVCAttn module to aggregate both source and reference image features:

$$\boldsymbol{F}_{int} = \text{PVCAttn}(\boldsymbol{F}_s, \boldsymbol{F}_r) \in \mathbb{R}^{256\times H_s/4\times W_s/4}. \quad (2)$$

Subsequently, decoder ϕ_D restores the color information $\boldsymbol{F}_{int}$ to the original resolution $\boldsymbol{F}_{color} \in \mathbb{R}^{64\times H_s\times W_s}$ via a self-attention layer and several convolution layers.

For super-resolution branch, encoder ψ_E extracts the residual texture feature $\boldsymbol{F}_{tex} \in \mathbb{R}^{64\times H_s\times W_s}$ from the initial feature map $\boldsymbol{F}_s^{init}$ via concatenated basic blocks. Each basic block contains two convolution layers along with a skip operation. A subsequent 3×3 convolution is used to aggregate $\boldsymbol{F}_{tex}$ and $\boldsymbol{F}_{color}$, indicated as $\boldsymbol{F}_{cs} \in \mathbb{R}^{256\times H_s\times W_s}$. Finally, the CPM module employs mapping function $\boldsymbol{\mathcal{F}}(\cdot)$ to regress target HR image $\hat{\boldsymbol{I}}_t^{lab} \in \mathbb{R}^{3\times H_s*p\times W_s*p}$, and p represents any magnification that can be a decimal, while $\overline{\boldsymbol{F}}_{cs} \in \mathbb{R}^{2\times H_s*p\times W_s*p}$ is obtained according to $\boldsymbol{F}_{cs}$. The detailed structure of SCSNet can be found in Appendix C.

3.1 Pyramid Valve Cross Attention

In order to more effectively aggregate feature information between source and reference images, we propose a novel *Valve Cross Attention* (VCAttn) module. As illustrated in Figure 4, the purpose of VCAttn is to select the reference feature $\boldsymbol{F}_r \in \mathbb{R}^{C_r\times H_r\times W_r}$ reasonably to the source feature $\boldsymbol{F}_s \in \mathbb{R}^{C_s\times H_s\times W_s}$. Similar to SAttn (Zhang et al. 2019b), three convolution operations are used to extract query features $\boldsymbol{Q}_s \in \mathbb{R}^{C\times H_s\times W_s}$, key features $\boldsymbol{K}_r \in \mathbb{R}^{C\times H_r\times W_r}$, and value features $\boldsymbol{V}_r \in \mathbb{R}^{C_s\times H_r\times W_r}$, respectively. Then, $\boldsymbol{Q}_s$ and $\boldsymbol{K}_r$ are employed to calculate the correlation matrix $\boldsymbol{CMat}$, which further multiplies $\boldsymbol{V}_r$ to obtain $\boldsymbol{F}_{r\to s}$. Subsequently, the designed valve structure fuses $\boldsymbol{F}_s$ and $\boldsymbol{F}_{r\to s}$ to obtain valve maps $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, which are used to control the information flux of $\boldsymbol{F}_s$ and $\boldsymbol{F}_{r\to s}$. To further improve the representation of VCAttn, we design a pyramid VCAttn module. As shown in Figure 5, pyramid feature maps are sent into corresponding VCAttn modules after pre-convolving, and the concatenated feature map goes through a post-convolution to obtain the final output.

3.2 Continuous Pixel Mapping

In order to generate target images at any magnification, we model the discrete feature mapping in continuous pixel space and propose an efficient super-resolution head named *Continuous Pixel Mapping*. As shown in Figure 6, we model the feature $\boldsymbol{F}^{x,y}$ of each point x, y with two parts: main feature $\overline{\boldsymbol{F}}_{cs}^{x,y}$ obtained by bilinear interpolation around the neighborhood four points, as well as the local coordinate feature $\boldsymbol{Z}_{rel}^{x,y}$ that describes continuous local spatial information. We model each point in the target image by its local relative coordinate to the nearest point in the original resolution image (*i.e.*, anchor point) for providing fine-grained guidance for each location. Since the coordinate is continuous that it can be infinitely interpolated, and is independent of the image resolution, $\boldsymbol{F}^{x,y}$ can be modeled in a continuous space. Note that we align the corner when obtaining the main feature of each point. As for local coordinate feature $\boldsymbol{Z}_{rel}^{x,y}$, we look for its corresponding anchor point in the original feature $\boldsymbol{F}_{cs} \in \mathbb{R}^{256\times H_s\times W_s}$ and calculate local coordinate feature $\boldsymbol{Z}_{rel}^{x,y}$ in the following formula:

$$\begin{aligned} \boldsymbol{Z}_{rel}^{x} &= \text{mod}(x, x_{unit})/x_{unit} * 2 - 1, \\ \boldsymbol{Z}_{rel}^{y} &= \text{mod}(y, y_{unit})/y_{unit} * 2 - 1, \end{aligned} \quad (3)$$

where $x_{unit} = 1/W_s, y_{unit} = 1/H_s$, mod is remainder operation, and $\boldsymbol{Z}_{rel}^{x/y}$ are in range -1 to +1, *i.e.*, (-1,-1) for the upper left corner while (+1,+1) for the lower right corner.

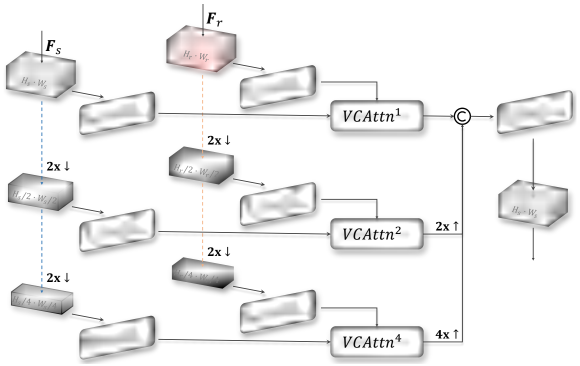

Figure 5. **Schematic diagram of PVCAttn**. Pre-convolved pyramid feature maps are processed by multiple VCAttn modules.

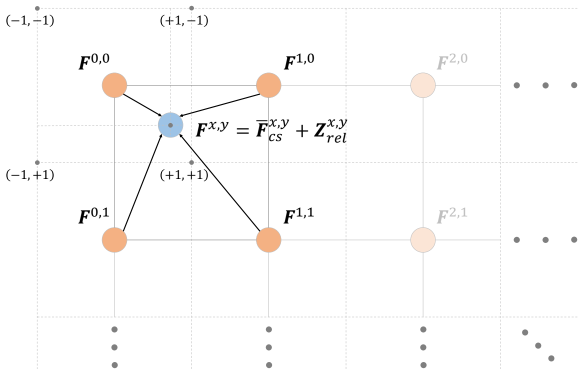

Figure 6. **Schematic diagram of CPM**. Taking four points on the feature map as an example, the feature $\boldsymbol{F}^{x,y}$ of the point x, y is modeled as the fusion of two parts: one is the main feature $\overline{\boldsymbol{F}}_{cs}^{x,y}$ obtained by bilinear interpolation with corner alignment, the other is the local relative coordinate $\boldsymbol{Z}_{rel}^{x,y}$ to the nearest anchor point.

Finally, continuous pix mapping function $\boldsymbol{\mathcal{F}}(\cdot)$ that contains four linear layers maps the feature to target image $\hat{\boldsymbol{I}}_t^{lab}$.

3.3 Objective Functions

During the training stage of SCSNet, we only adopt three losses: *Content Loss* $\mathcal{L}_C$ to monitor image quality at the pixel level, *Perceptual Loss* $\mathcal{L}_P$ to ensure semantic similarity, and *Adversarial Loss* $\mathcal{L}_{Adv}$ to improve image quality and authenticity. The full loss $\mathcal{L}_{all}$ is defined as follow:

$$\mathcal{L}_{all} = \lambda_C \mathcal{L}_C + \lambda_P \mathcal{L}_P + \lambda_{Adv} \mathcal{L}_{Adv}, \tag{4}$$

where $\lambda_C = 10$, $\lambda_P = 5$, and $\lambda_{Adv} = 1$ represent weight parameters to balance different terms.

Content Loss. The first term $\mathcal{L}_C$ calculates the ℓ_1 error between the generated target image $\hat{\boldsymbol{I}}_t^{lab}$ and ground truth $\boldsymbol{I}_t^{lab}$:

$$\mathcal{L}_C = ||\hat{\boldsymbol{I}}_t^{lab} - \boldsymbol{I}_t^{lab}||_1. \tag{5}$$

Perceptual Loss. The second term $\mathcal{L}_P$ calculates semantic errors between the generated target image $\hat{\boldsymbol{I}}_t^{lab}$ and the ground truth image $\boldsymbol{I}_t^{lab}$:

$$\mathcal{L}_P = \mathbb{E}\left[\sum_{l=1}^{5} w_l \cdot ||\phi_l(\hat{\boldsymbol{I}}_t^{lab}) - \phi_l(\boldsymbol{I}_t^{lab})||_1\right], \tag{6}$$

where $\phi_l(\cdot)$ represents the activation map extracted at the $convl\_1$ layer from the pre-trained VGG16 network, and w_l is the weight for layer_l.

Adversarial Loss. The third term $\mathcal{L}_{Adv}$ employs the standard relativistic discriminator (Jolicoeur-Martineau 2018) for adversarial training in order to ensure the authenticity of the generated images. Since the SCS task is typically a one-to-many problem, the adversarial loss greatly improves the model performance.

$$\begin{aligned}\mathcal{L}_{Adv}^G =& \mathbb{E}_{x\sim p_x}\left[(D(x) - \mathbb{E}_{\tilde{x}\sim p_{\tilde{x}}}[D(\tilde{x})] - 1)^2\right] + \\ & \mathbb{E}_{\tilde{x}\sim p_{\tilde{x}}}\left[(D(\tilde{x}) - \mathbb{E}_{x\sim p_x}[D(x)])^2\right], \\ \mathcal{L}_{Adv}^D =& \mathbb{E}_{\tilde{x}\sim p_{\tilde{x}}}\left[(D(\tilde{x}) - \mathbb{E}_{x\sim p_x}[D(x)] - 1)^2\right],\end{aligned} \tag{7}$$

where p_x and $p_{\tilde{x}}$ are real and generated image distributions.

4 Experiments

4.1 Datasets and Implementation Details

ImageNet-C. Considering the high requirement for both image colorization and super-resolution, we filtered out some of the low-quality images from ImageNet (Deng et al. 2009) to build a colorful and high-resolution dataset named ImageNet-C. It ends up with 407,041 training images and 16,216 validation images. In detail, we remove images with file sizes smaller than 80K and less color variation.

Other Datasets. CelebA-HQ (Karras et al. 2018) (30,000 images), Flowers (Nilsback and Zisserman 2008) (8,189 images), Bird (Yu et al. 2015) (479,548 images), and COCO (Lin et al. 2014) (98,246 images) datasets are used to assess different colorization methods (for testing), and they go through the same pre-selection process as ImageNet-C.

Evaluation Metrics. We use *Peak Signal-to-Noise Ratio* (PSNR) and *Structural Similarity* (SSIM) (Wang et al. 2004) to assess the generated images at pixel level, while *Fréchet Inception Distance* (FID) (Heusel et al. 2017) and *Image ColorfulNess* (CN) (Zerman, Rana, and Smolic 2019) to assess image distribution and colorfulness at semantic level. In order to fully evaluate various methods, we hire real people to score authenticity of images for human study.

Implementation Details. The image is processed in LAB color space, and the input resolution of source and reference images is 128×128. Consistent with DRN, the bicubic kernel is used to produce LR-HR pairs. We train the SCSNet with two modes alternately and apply random horizontal flip and elastic distortion (Simard et al. 2003) to reference images. Perceptual weights w_{1-5} in Eq. 6 are set as 1.0/32, 1.0/16, 1.0/8, 1.0/4, and 1.0, respectively. We use Adam (Kingma and Ba 2015) optimizer and set $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight-decay=$1e^{-4}$, and learning rate=$1e^{-4}$. SCSNet is trained for 50 epochs with batch-size=4 and output-resolution=512 (default ×4↑ setting). Resolution of the referential image for all experiments is set to 128, and all experiments run with 8 Tesla V100 GPUs.

4.2 Comparison with SOTAs

We conduct and discuss a series of qualitative and quantitative comparison experiments on several datasets. At present, there is no end-to-end model to perform image colorization and super-resolution simultaneously, so we choose some SOTA colorization methods (*i.e.*, AutoColor (Lei and Chen 2019), DRemaster (Iizuka and Simo-Serra 2019), InstColor (Su, Chu, and Huang 2020), DEVC (Zhang et al. 2019a), ColTran (Kumar, Weissenborn, and Kalchbrenner 2021)) along with concatenated super-resolution approaches (ESRGAN (Wang et al. 2018), DRN (Guo et al. 2020)) as our comparison methods. Concretely, we divide the above colorization methods into automatic and referential modes.

Qualitative Results. We conduct a series of qualitative experiments on ImageNet-C and COCO validation datasets to visually show the superiority of our approach for generating authentic and colorful images for the SCS problem. As shown in Figure 7, the left part shows automatic SCS results of different methods that use the low-resolution gray-scale images (first column) as input. All methods can distinguish semantic targets and color them, but our generated images look better in colorfulness and detail than other approaches. The right part shows results for various methods under the condition of an elastic reference image in resolution 128×128, which provides the color information that the real image should contain. All methods could transfer referential color well except ColTran, but our method can produce clearer and authentic images while maintaining color transfer. More results of our approach can be found in Appendix F. Note that ColTran inputs an extra HR gray-scale image for better clarity that is unfair for the SCS task. We reduce its input gray-scale image to the same resolution as other methods, and the output images get a little blurry (*c.f.* ColTran-LR in the penultimate column).

Quantitative Results. We choose image-level metrics to evaluate the effectiveness of different SOTA methods on

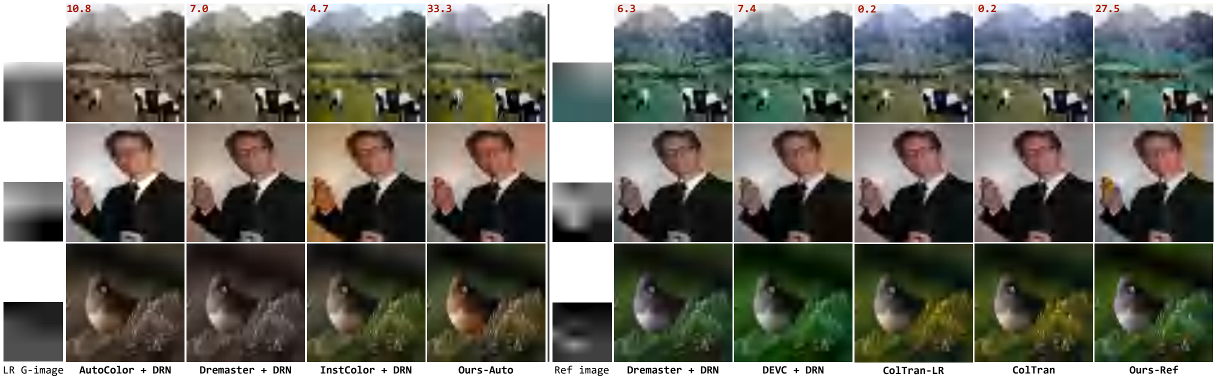

Figure 7. **Qualitative comparisons with SOTA methods on ImageNet-C and COCO datasets.** The left and right parts are *simultaneously image colorization and super-resolution* results in automatic and referential modes, respectively. The upper left number is the inference FPS speed of the corresponding method. Note that ColTran inputs an *extra* HR gray-scale image that is unfair for other approaches, so we reduce its input gray-scale image to the same resolution as other methods, *i.e*., ColTran-LR.

	Method	ImageNet-C		CelebA-HQ		Flowers		Bird		COCO		Average		Params	Speed
		FID ↓	CN ↑	FID ↓	CN ↑	FID ↓	CN ↑	FID ↓	CN ↑	FID ↓	CN ↑	FID ↓	CN ↑	(M)↓	(FPS)↑
	Real Test Dataset	-	5.522	-	4.344	-	5.783	-	5.006	-	5.222	-	5.175	-	-
Automatic	AutoColor + ESRGAN	31.725	3.390	25.637	3.488	64.107	3.210	42.956	3.190	38.904	3.417	40.666	3.339	25.432	16.910
	DRemaster + ESRGAN	33.908	2.998	34.775	3.054	74.601	2.678	45.505	2.833	37.249	3.078	45.208	2.928	65.516	9.217
	InstColor + ESRGAN	26.353	3.588	31.109	3.306	48.425	3.510	36.212	3.435	24.954	3.635	33.411	3.495	66.990	5.541
	AutoColor + DRN	31.666	3.389	**26.083**	3.489	63.891	3.210	43.034	3.189	39.208	3.417	40.776	3.339	18.560	10.781
	DRemaster + DRN	33.993	2.996	34.842	3.054	74.356	2.678	45.742	2.832	37.397	3.076	45.266	2.927	58.644	7.037
	InstColor + DRN	26.501	3.588	31.389	3.307	48.475	3.511	36.378	3.436	25.153	3.635	33.579	3.495	60.118	4.671
	Ours-Auto	**25.992**	**4.688**	27.809	**3.892**	**46.607**	**4.724**	**34.401**	**4.334**	**24.047**	**4.573**	**31.771**	**4.442**	**9.954**	**33.293**
Referential	DRemaster + DRN	25.498	3.990	29.702	3.426	41.506	4.311	31.843	3.505	35.267	3.722	32.763	3.791	73.987	6.311
	DEVC + DRN	26.050	4.288	49.126	3.570	39.426	4.516	42.444	3.700	36.702	3.938	38.750	4.002	69.570	7.435
	ColTran	15.860	4.692	**10.405**	4.215	21.595	5.003	18.580	4.529	18.391	4.679	16.966	4.624	70.697	0.209
	Ours-Ref	**9.632**	**5.288**	12.771	**4.388**	**9.776**	**5.743**	**13.526**	**4.822**	**13.812**	**4.974**	**11.903**	**5.043**	**15.358**	**27.466**

Table 1. **Image-level evaluation for SOTA methods on several datasets.** Since the SCS task is an ill-conditioned problem that each pixel has various semantic colors, more reasonable image-level FID and CN are used. **Bold** and underline represent optimal and suboptimal results.

Method	ImageNet-C		CelebA-HQ		COCO	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
DRemaster + DRN	19.343	0.811	25.559	0.915	21.039	0.845
InstColor + DRN	22.126	0.842	26.523	**0.923**	22.917	0.856
Ours-Auto	**22.807**	**0.856**	**27.160**	0.917	**23.341**	**0.872**
ColTran	20.734	0.845	24.495	0.914	22.787	0.857
DRemaster + DRN	24.671	0.871	28.582	0.928	26.663	0.901
Ours-Ref	**27.694**	**0.923**	**30.741**	**0.950**	**28.197**	**0.931**

Table 2. **Pixel-level evaluation for SOTA methods.** Top and bottom parts are for automatic and referential modes, respectively.

several datasets: FID for assessing image distribution while visual-friendly CN for colorfulness. Our approach is trained only on ImageNet-C without extra datasets, while other methods use corresponding pre-trained models that may use extra datasets for training. We randomly choose 5,000 images of each method for assessment (2,500 under automatic mode; 500 by self-referential elastic images; while 2,000 by randomly selecting other elastic images as reference images). Table 1 shows the results of different methods for several datasets on two modes, and we can summarize the following conclusions: **1)** The middle part illustrates that different SR methods have little difference in the results, so we choose SOTA DRN for SISR in the following experiments. **2)** Referential mode tends to get better results than automatic mode. **3)** Different datasets are slightly different in CN metric and our method obtains the highest CN score (*i.e*., 4.442 and 5.043 for two modes, increasing +0.947 and +0.419 than current best results), meaning that our approach can capture color information better and generate visual-appealing colorful images. **4)** Our method obtains the best FID scores on almost all datasets no matter in automatic mode or referential mode, meaning that the generated images by our method have a more consistent distribution of the real images. **5)** We further evaluate the parameter and running speed of all approaches, and our SCSNet has the fewest parameters (×6↓ than InstColor+DRN; ×4↓ than ColTran) and fastest running speed (×8↑ than InstColor+DRN; ×130↑ than ColTran; with batch size equaling one), which is more efficient for practical application. Furthermore, we use pixel-level PSNR and SSIM to evaluate generated images under automatic and self-referential modes. As shown in Table 2, our method consistently obtains better evaluation scores, meaning that the predicted images by SCSNet are more consistent with real images. Interestingly, the aforementioned ColTran inability to integrate referential image colors (*c.f*. Figure 7) is also reflected here, where it obtains worse pixel-level metric scores.

Comparison Methods	Authenticity (%)
Ours *v.s.* AutoColor + DRN	**77.3** *v.s.* 22.7
Ours *v.s.* DRemaster + DRN	**90.6** *v.s.* 9.4
Ours *v.s.* InstColor + DRN	**59.4** *v.s.* 40.6
Ours *v.s.* DRemaster + DRN	**82.7** *v.s.* 17.3
Ours *v.s.* DEVC + DRN	**95.1** *v.s.* 4.9
Ours *v.s.* ColTran	**68.6** *v.s.* 31.4

Table 3. Human study about the authenticity of generated images with different methods in automatic and referential modes.

$\mathcal{L}_C$	$\mathcal{L}_P$	$\mathcal{L}_{Adv}$	FID ↓	CN ↑	PSNR ↑	SSIM ↑
✓	✗	✗	16.290	4.128	26.172	0.898
✓	✓	✗	15.068	4.196	<u>27.456</u>	<u>0.918</u>
✓	✗	✓	<u>11.616</u>	<u>5.171</u>	26.536	0.906
✓	✓	✓	**9.632**	**5.288**	**27.694**	**0.923**

Table 4. Quantitative ablation study for different loss terms.

Human Study. Since SCS is an ill-conditioned problem, and each metric has its evaluation disadvantage, we further perform a human study for artificially evaluating the quality of generated images for different methods. Concretely, we randomly select 500 generated images (250:250 for two modes) of different approaches on the COCO dataset. Each image pair (*i.e.*, ours *vs.* each other method) is displayed for one second for each conner (50 totally), and the conner needs to select which image is more visually authentic. Table 3 illustrates that the generated images by our approach are preferred by real people, meaning that our method can generate more authentic images than SOTA methods.

4.3 Ablation Study and Further Assessment

Loss Functions. Following the afore-mentioned procedure for generating the validation images (under referential mode), we quantitatively evaluate the effectiveness of each loss function in Table 4 and draw a conclusion: Each loss function contributes to the model performance, and the model obtains the best metric score when all loss functions are applied. Qualitative results can be found in Appendix A.

Network Components. We perform quantitative experiments to evaluate the effectiveness of each component of our approach. Specifically, we modify a simple version of PVCAttn as *Basic Cross-Attention* (BCAttn) that removes pyramid structure and valves, which is used for a fair comparison with our PVCAttn. Results in Table 5 demonstrate the effectiveness of each component, and our approach obtains the highest metric scores when both proposed components are used. Moreover, the CPM module obtains competitive results even though it is designed for continuous magnification, and more result analysis can be found in Appendix B.

CPM Efficiency. We compare CPM module with Meta-SR (Hu et al. 2019) that also achieves continuous magnification, and results indicate that CPM is more efficient as it has fewer parameters and a ×2↑ faster running speed, *i.e.*, **0.35M** *vs.* 0.45M and **178FPS** *vs.* 92 FPS.

Interpretability of PVCAttn. In Figure 8, we visualize the attention maps of PVCAttn, focusing on a few points in the source image. Visualized attention maps indicate that each location pays more attention to semantically similar areas, and the location in the low-resolution feature map focuses on more average areas (*c.f.* right-bottom attention map).

Baseline	BCAttn	PVCAttn	CPM	FID ↓	CN ↑	PSNR ↑	SSIM ↑
✓	✗	✗	✗	17.541	4.763	25.517	0.887
✓	✓	✗	✗	16.635	4.767	26.173	0.896
✓	✗	✓	✗	15.334	4.863	26.619	<u>0.907</u>
✓	✗	✗	✓	<u>9.978</u>	<u>5.059</u>	<u>26.796</u>	0.905
✓	✗	✓	✓	**9.632**	**5.288**	**27.694**	**0.923**

Table 5. Quantitative ablation study of our approach with different components on the ImageNet-C dataset.

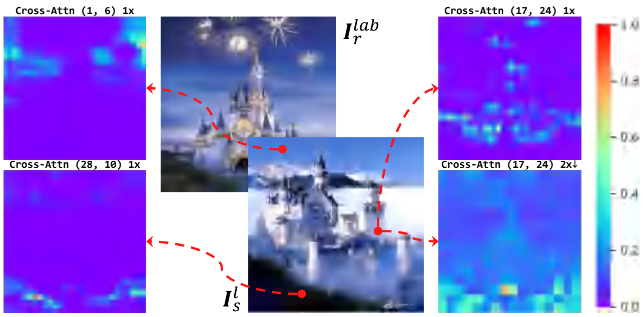

Figure 8. Cross attention for a set of reference points. The PVCAttn can match similar semantic information. The left part shows attention maps for different positions, while the right part shows pyramid attention maps for one position.

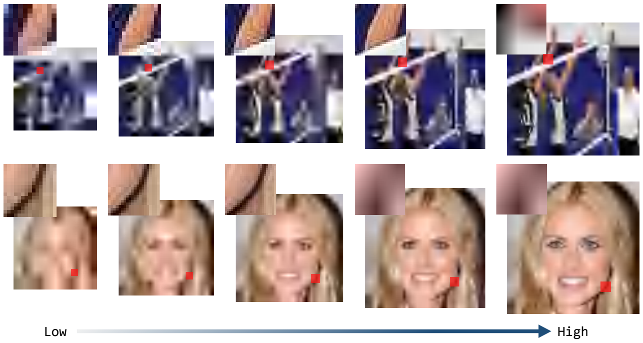

Figure 9. Image generation at continuous magnification. Magnified red areas are displayed in the upper left corner.

Multi-Magnification Generation. Benefit from the CPM module, SCSNet can generate target images at continuous magnification. As shown in Figure 9, the generated results have consistent color stability for different resolutions and smooth transitions for adjacent images. Dynamic video can be seen in the supplementary material.

5 Conclusion

In this paper, we propose an efficient paradigm to address SCS task and design an end-to-end SCSNet to complete this goal. Concretely, a *PVCAttn* module is designed to aggregate feature information between source and reference images effectively, while the *CPM* efficiently models the discrete pixel mapping in a continuous space to generate target images at arbitrary magnification. Extensive experiments demonstrate our approach's superiority for achieving the SCS task well and generating high-quality images.

References

Anwar, S.; Tahir, M.; Li, C.; Mian, A.; Khan, F. S.; and Muzaffar, A. W. 2020. Image colorization: A survey and dataset. *arXiv preprint arXiv:2008.10774*.

Cai, J.; Zeng, H.; Yong, H.; Cao, Z.; and Zhang, L. 2019. Toward real-world single image super-resolution: A new benchmark and a new model. In *ICCV*, 3086–3095.

Cao, Y.; Zhou, Z.; Zhang, W.; and Yu, Y. 2017. Unsupervised diverse colorization via generative adversarial networks. In *ECML-PKDD*, 151–166. Springer.

Cheng, Z.; Yang, Q.; and Sheng, B. 2015. Deep colorization. In *ICCV*, 415–423.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255. Ieee.

Deshpande, A.; Lu, J.; Yeh, M.-C.; Jin Chong, M.; and Forsyth, D. 2017. Learning diverse image colorization. In *CVPR*, 6837–6845.

Deshpande, A.; Rock, J.; and Forsyth, D. 2015. Learning large-scale automatic image colorization. In *ICCV*, 567–575.

Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2015. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2): 295–307.

Furusawa, C.; Hiroshiba, K.; Ogaki, K.; and Odagiri, Y. 2017. Comicolorization: semi-automatic manga colorization. In *SIGGRAPH Asia 2017 Technical Briefs*, 1–4.

Guo, Y.; Chen, J.; Wang, J.; Chen, Q.; Cao, J.; Deng, Z.; Xu, Y.; and Tan, M. 2020. Closed-loop matters: Dual regression networks for single image super-resolution. In *CVPR*, 5407–5416.

He, M.; Chen, D.; Liao, J.; Sander, P. V.; and Yuan, L. 2018. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, 37(4): 1–16.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*.

Hu, X.; Mu, H.; Zhang, X.; Wang, Z.; Tan, T.; and Sun, J. 2019. Meta-SR: A magnification-arbitrary network for super-resolution. In *CVPR*, 1575–1584.

Huang, J.; Liao, J.; and Kwong, S. 2020. Semantic Example Guided Image-to-Image Translation. *IEEE Transactions on Multimedia*.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 1501–1510.

Iizuka, S.; and Simo-Serra, E. 2019. DeepRemaster: temporal source-reference attention networks for comprehensive video enhancement. *ACM Transactions on Graphics (TOG)*, 38(6): 1–13.

Iizuka, S.; Simo-Serra, E.; and Ishikawa, H. 2016. 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.

Jolicoeur-Martineau, A. 2018. The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint arXiv:1807.00734*.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kumar, M.; Weissenborn, D.; and Kalchbrenner, N. 2021. Colorization Transformer. In *ICLR*.

Larsson, G.; Maire, M.; and Shakhnarovich, G. 2016. Learning representations for automatic colorization. In *ECCV*, 577–593. Springer.

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 4681–4690.

Lee, J.; Kim, E.; Lee, Y.; Kim, D.; Chang, J.; and Choo, J. 2020. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *CVPR*, 5801–5810.

Lei, C.; and Chen, Q. 2019. Fully automatic video colorization with self-regularization and diversity. In *CVPR*, 3753–3761.

Lim, B.; Son, S.; Kim, H.; Nah, S.; and Mu Lee, K. 2017. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 136–144.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Lu, P.; Yu, J.; Peng, X.; Zhao, Z.; and Wang, X. 2020. Gray2ColorNet: Transfer More Colors from Reference Image. In *ACM MM*, 3210–3218.

Martin, D.; Fowlkes, C.; Tal, D.; and Malik, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, 416–423. IEEE.

Messaoud, S.; Forsyth, D.; and Schwing, A. G. 2018. Structural consistency and controllability for diverse colorization. In *ECCV*, 596–612.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. In *ICLR*.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *ICVGIP*, 722–729. IEEE.

Simard, P. Y.; Steinkraus, D.; Platt, J. C.; et al. 2003. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*.

Su, J.-W.; Chu, H.-K.; and Huang, J.-B. 2020. Instance-aware image colorization. In *CVPR*, 7968–7977.

Sun, T.-H.; Lai, C.-H.; Wong, S.-K.; and Wang, Y.-S. 2019. Adversarial colorization of icons based on contour and color conditions. In *ACM MM*, 683–691.

Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.-H.; and Zhang, L. 2017. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 114–125.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Vitoria, P.; Raad, L.; and Ballester, C. 2020. ChromaGAN: adversarial picture colorization with semantic class distribution. In *WACV*, 2445–2454.

Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; and Change Loy, C. 2018. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, 0–0.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Xian, W.; Sangkloy, P.; Agrawal, V.; Raj, A.; Lu, J.; Fang, C.; Yu, F.; and Hays, J. 2018. Texturegan: Controlling deep image synthesis with texture patches. In *CVPR*, 8456–8465.

Xu, Z.; Wang, T.; Fang, F.; Sheng, Y.; and Zhang, G. 2020. Stylization-based architecture for fast deep exemplar colorization. In *CVPR*, 9363–9372.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zerman, E.; Rana, A.; and Smolic, A. 2019. Colornet-estimating colorfulness in natural images. In *2019 IEEE International Conference on Image Processing (ICIP)*, 3791–3795. IEEE.

Zeyde, R.; Elad, M.; and Protter, M. 2010. On single image scale-up using sparse-representations. In *ICCS*, 711–730. Springer.

Zhang, B.; He, M.; Liao, J.; Sander, P. V.; Yuan, L.; Bermak, A.; and Chen, D. 2019a. Deep exemplar-based video colorization. In *CVPR*, 8052–8061.

Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2019b. Self-attention generative adversarial networks. In *ICML*, 7354–7363. PMLR.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *ECCV*, 649–666. Springer.

Zhang, X.; Chen, Q.; Ng, R.; and Koltun, V. 2019c. Zoom to learn, learn to zoom. In *CVPR*, 3762–3770.

Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; and Fu, Y. 2018a. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 286–301.

Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. 2018b. Residual dense network for image super-resolution. In *CVPR*, 2472–2481.

Zhao, J.; Han, J.; Shao, L.; and Snoek, C. G. 2019. Pixelated semantic colorization. *International Journal of Computer Vision*, 1–17.

Zou, C.; Mo, H.; Gao, C.; Du, R.; and Fu, H. 2019. Language-based colorization of scene sketches. *ACM Transactions on Graphics (TOG)*, 38(6): 1–16.

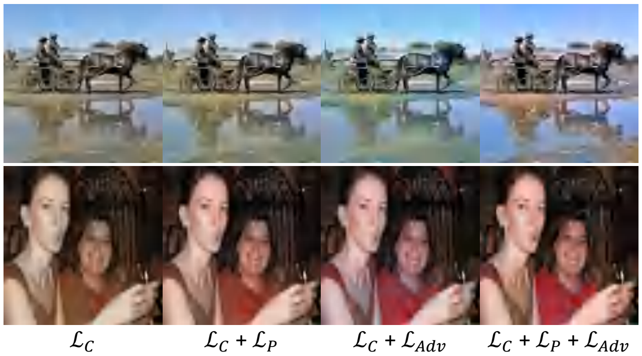
$\mathcal{L}_C$
$\mathcal{L}_C + \mathcal{L}_P$
$\mathcal{L}_C + \mathcal{L}_{Adv}$
$\mathcal{L}_C + \mathcal{L}_P + \mathcal{L}_{Adv}$

Figure 10. Qualitative comparisons for different loss terms

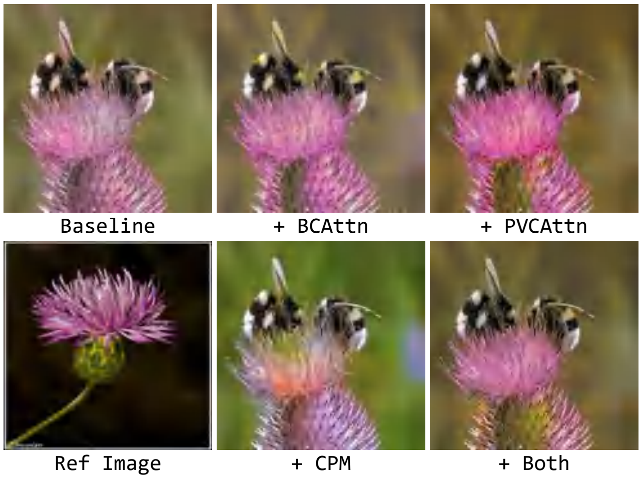

Figure 11. Qualitative comparisons with different components on ImageNet-C dataset. Please zoom in for more details.

5.1 A Further Qualitative Ablation Study

5.2 Ablation Study

In this section, we further conduct ablation studies qualitatively on the ImageNet-C dataset to analyze the contribution of each loss function and component.

Loss Functions. We conduct a qualitative experiment to assess the impact of each loss function intuitively. As shown in Figure 10, the first row shows the results of our baseline that only use the content loss $\mathcal{L}_C$. When we add perceptual loss $\mathcal{L}_P$ into the training process, the model gets better visual effects at a semantic level, *e.g.*, obtaining more realistic and consistent color information. At the same time, the adversarial loss $\mathcal{L}_{Adv}$ makes the network more concerned about the color characteristics of details, *e.g.*, reflections on water and skin tone. Our method achieves the best visual effects when all loss functions are used simultaneously, where filled color and semantic boundary can be intuitively felt.

Network Components. We perform a qualitative experiment to evaluate the effectiveness of each component for our approach. Specifically, we modify a simple version of PVCAttn as *Basic Cross-Attention* (BCAttn) that removes pyramid structure and valves, which is used for a fair comparison with our PVCAttn. As shown in Figure 11, the *BCAttn* provides better color transfer effect compared with the baseline, while our *PVCAttn* module help to generate more colorful images than BCAttn. Nevertheless, the above two modules improve image sharpness to a limited extent. The edges of objects become sharper when *CPM* module is added, but it is more likely to cause semantic coloring errors (*e.g.*, green on the flower). When both PVCAttn and CPM are used, the generated images have sharper edges and more reasonable semantic coloring results.

5.3 B Further Assessment of CPM

We further conduct a comparison experiment with RDN (Zhang et al. 2018b) and Meta-SR (Hu et al. 2019) to assess our CPM module on widely used Set14 (Zeyde, Elad, and Protter 2010), BSD100 (Martin et al. 2001), and DIV2K (Timofte et al. 2017) datasets. Similar to Meta-SR (Hu et al. 2019), we replace the SR head of RDN (Zhang et al. 2018b) with the CPM module and conduct experiments on X2, X3, and X4 settings. As shown in Table 6, experimental results indicate that the CPM module can obtain competitive results even though it is designed for continuous magnification of target images. With the CPM module, we only need to store one unified model instead of a model for each magnification, which greatly improves the practicality. Also, as depicted in Table 7, our CPM module is more efficient than Meta-SR as it has fewer parameters and a $\times 2\uparrow$ faster speed, *i.e.*, **0.35M** *vs*. 0.45M and **178FPS** *vs*. 92 FPS.

5.4 C SCSNet Architecture

The detailed network structure of SCSNet is shown in Table 8. Some notations: Conv (convolution layer), Deconv (deconvolution layer), IN (instance normalization layer), Linear (linear layer), UP (up-sampling layer), SN (Miyato et al. 2018) (spectral normalization), SAttn (self-attention layer) (Zhang et al. 2019b), Basic Block for SR, LReLU (LeakyReLU), M (the number of the input channel), N (the number of the output channel), K (kernel size), S (stride size), P (padding size), and p (magnification rate). Also, we attach the PyTorch implementation for the model in the first folder of the supplementary material.

5.5 D Training Dataset

The used images for training in the ImageNet-C dataset can be obtained in the supplementary material.

5.6 E Elastic Deformation Examples

Figure 12 shows some reference images with elastic deformation. In training, elastic deformation can eliminate some inherent structural information of reference images while preserving color information. Therefore, the designed PVCAttn module learns how to look for colors from the reference image to superimpose on the source image.

5.7 F More SCS Results

Figure 13 shows more generation results on Flowers (Nilsback and Zisserman 2008), CelebA-HQ (Karras et al. 2018), COCO (Lin et al. 2014), and ImageNet-C datasets.

5.8 G Failure Cases

Some failure cases can be viewed in Figure 14. Typically, the failures include the wrong colorization of boundaries, color shift, and improper colorization.

Method	Metric	Set14			B100			DIV2K		
		X2	X3	X4	X2	X3	X4	X2	X3	X4
Bicubic	PSNR ↑	30.240	27.550	26.000	29.560	27.210	25.960	31.350	28.490	26.920
	SSIM ↑	0.8688	0.7742	0.7227	0.8431	0.7385	0.6675	0.9076	0.8339	0.7774
RDN (Zhang et al. 2018b)	PSNR ↑	34.010	30.570	28.810	32.340	29.260	27.720	35.170	31.390	29.340
	SSIM ↑	0.9212	0.8468	0.7871	0.9017	0.8093	0.7419	0.9483	0.8931	0.8446
Meta-RDN (Hu et al. 2019)	PSNR ↑	**34.040**	30.550	28.840	**32.350**	29.300	27.750	**35.180**	31.420	29.360
	SSIM ↑	**0.9213**	0.8466	0.7872	0.9019	0.8096	**0.7423**	0.9484	**0.8935**	**0.8448**
CPM-RDN (Ours)	PSNR ↑	34.000	**30.610**	**28.910**	**32.350**	**29.310**	**27.770**	**35.180**	**31.450**	**29.380**
	SSIM ↑	0.9207	**0.8472**	**0.7880**	**0.9020**	**0.8101**	**0.7423**	**0.9486**	0.8930	0.8428

Table 6. Effect assessment of the CPM module with different methods on X2, X3, and X4 settings.

Table 7. Effect assessment of the CPM module with SOTA methods on X2, X3, and X4 settings.

Method	Params (M)	Speed (FPS)
Meta-SR	0.45	92
CPM-SR	**0.35** (-22.2% ↓)	**178** (+93.5% ↑)

Module	Input → Output Shape	Layer Information
Encoder $\boldsymbol{\phi}_E^s$	$(1, H_s, W_s) \rightarrow (64, H_s, W_s)$	Conv-(M-1, N-64, K-3x3, S-1, P-1), IN, LReLU
	$(64, H_s, W_s) \rightarrow (64, H_s, W_s)$	Conv-(M-64, N-64, K-3x3, S-1, P-1), IN, LReLU
	$(64, H_s, W_s) \rightarrow (64, H_s, W_s)$	Conv-(M-64, N-64, K-3x3, S-1, P-1), IN, LReLU
	$(64, H_s, W_s) \rightarrow (64, H_s/2, W_s/2)$	Conv-(M-64, N-64, K-3x3, S-2, P-1), IN, LReLU
	$(64, H_s/2, W_s/2) \rightarrow (128, H_s/2, W_s/2)$	Conv-(M-64, N-128, K-3x3, S-1, P-1), IN, LReLU
	$(128, H_s/2, W_s/2) \rightarrow (128, H_s/2, W_s/2)$	Conv-(M-128, N-128, K-3x3, S-1, P-1), IN, LReLU
	$(128, H_s/2, W_s/2) \rightarrow (128, H_s/4, W_s/4)$	Conv-(M-128, N-128, K-3x3, S-2, P-1), IN, LReLU
	$(128, H_s/4, W_s/4) \rightarrow (256, H_s/4, W_s/4)$	Conv-(M-128, N-256, K-3x3, S-1, P-1), IN, LReLU
	$(256, H_s/4, W_s/4) \rightarrow (256, H_s/4, W_s/4)$	Conv-(M-256, N-256, K-3x3, S-1, P-1), IN, LReLU
Encoder $\boldsymbol{\phi}_E^r$	$(3, H_r, W_r) \rightarrow (64, H_r, W_r)$	Conv-(M-3, N-64, K-3x3, S-1, P-1), IN, LReLU
	$(64, H_r, W_r) \rightarrow (64, H_r, W_r)$	Conv-(M-64, N-64, K-3x3, S-1, P-1), IN, LReLU
	$(64, H_r, W_r) \rightarrow (64, H_r, W_r)$	Conv-(M-64, N-64, K-3x3, S-1, P-1), IN, LReLU
	$(64, H_r, W_r) \rightarrow (64, H_r/2, W_r/2)$	Conv-(M-64, N-64, K-3x3, S-2, P-1), IN, LReLU
	$(64, H_r/2, W_r/2) \rightarrow (128, H_r/2, W_r/2)$	Conv-(M-64, N-128, K-3x3, S-1, P-1), IN, LReLU
	$(128, H_r/2, W_r/2) \rightarrow (128, H_r/2, W_r/2)$	Conv-(M-128, N-128, K-3x3, S-1, P-1), IN, LReLU
	$(128, H_r/2, W_r/2) \rightarrow (128, H_r/4, W_r/4)$	Conv-(M-128, N-128, K-3x3, S-2, P-1), IN, LReLU
	$(128, H_r/4, W_r/4) \rightarrow (256, H_r/4, W_r/4)$	Conv-(M-128, N-256, K-3x3, S-1, P-1), IN, LReLU
	$(256, H_r/4, W_r/4) \rightarrow (256, H_r/4, W_r/4)$	Conv-(M-256, N-256, K-3x3, S-1, P-1), IN, LReLU
Extractor	$(256, H_s/4, W_s/4) \rightarrow (256, H_s/4, W_s/4)$	PVCAttn-(M-256, N-256)
	$(256, H_s/4, W_s/4) \rightarrow (256, H_s/4, W_s/4)$	Conv-(M-256, N-256, K-3x3, S-1, P-1), IN, LReLU
	$(256, H_s/4, W_s/4) \rightarrow (256, H_s/4, W_s/4)$	Conv-(M-256, N-256, K-3x3, S-1, P-1), IN, LReLU
	$(256, H_s/4, W_s/4) \rightarrow (256, H_s/4, W_s/4)$	Conv-(M-256, N-256, K-3x3, S-1, P-1), IN, LReLU
	$(256, H_s/4, W_s/4) \rightarrow (256, H_s/4, W_s/4)$	SAttn(M-256, N-256)
Decoder $\boldsymbol{\phi}_D$	$(256, H_s/4, W_s/4) \rightarrow (256, H_s/2, W_s/2)$	UP-(M-256, N-256)
	$(256, H_s/2, W_s/2) \rightarrow (256, H_s/2, W_s/2)$	Conv-(M-256, N-256, K-3x3, S-1, P-1), IN, LReLU
	$(256, H_s/2, W_s/2) \rightarrow (128, H_s/2, W_s/2)$	Conv-(M-256, N-128, K-3x3, S-1, P-1), IN, LReLU
	$(128, H_s/2, W_s/2) \rightarrow (128, H_s, W_s)$	UP-(M-128, N-128)
	$(128, H_s, W_s) \rightarrow (128, H_s, W_s)$	Conv-(M-128, N-128, K-3x3, S-1, P-1), IN, LReLU
	$(128, H_s, W_s) \rightarrow (64, H_s, W_s)$	Conv-(M-128, N-64, K-3x3, S-1, P-1), IN, LReLU
Encoder $\boldsymbol{\psi}_E$	$(64, H_s, W_s) \rightarrow (64, H_s, W_s)$	Basic Block-(M-64, N-64) × 16
Decoder $\boldsymbol{\psi}_D$	$(128, H_s, W_s) \rightarrow (3, H_s * p, W_s * p)$	CPM-(M-128, N-3)

Table 8. Detailed SCSNet architecture.

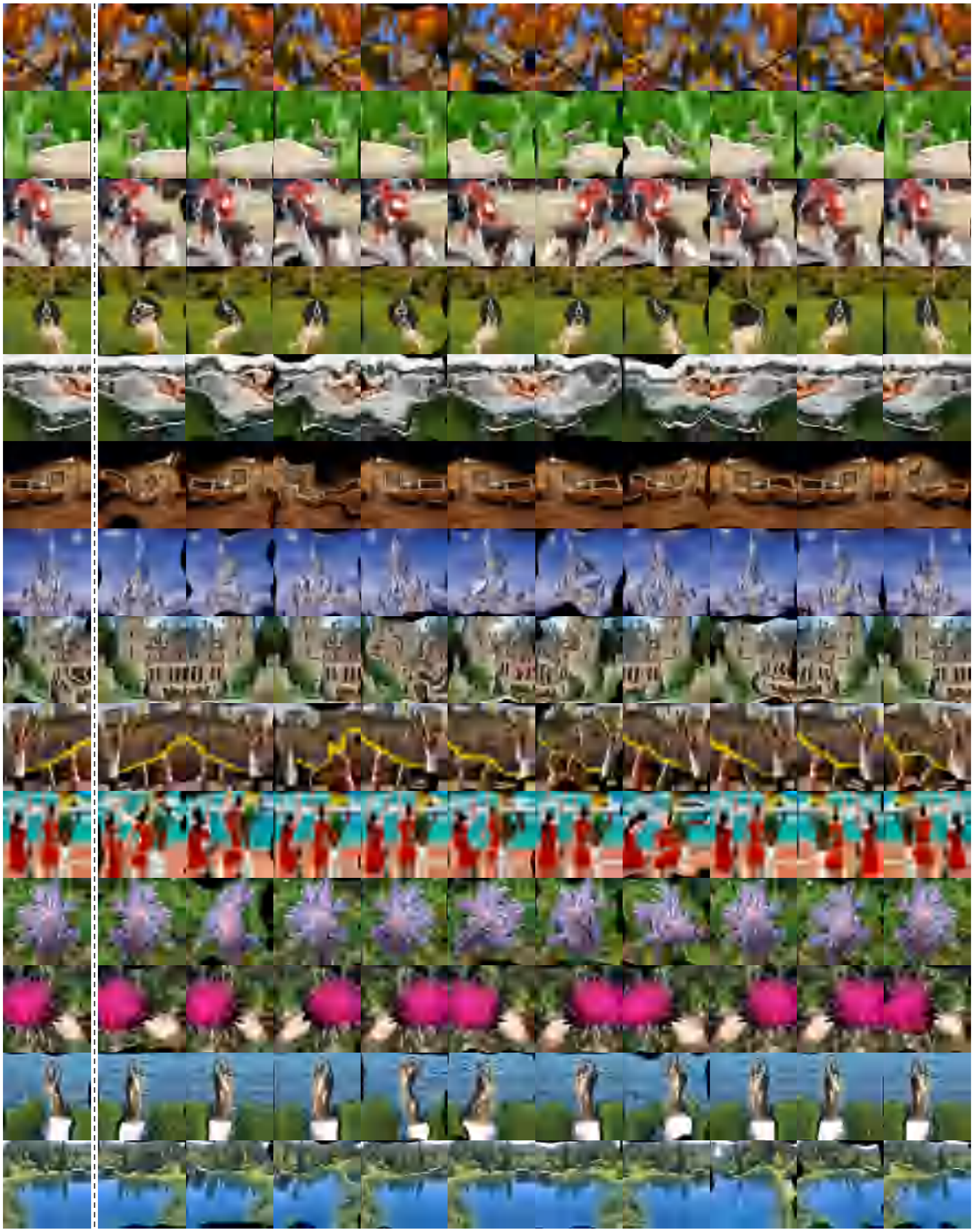

Figure 12. Elastic deformation results of the reference images during the training phase.

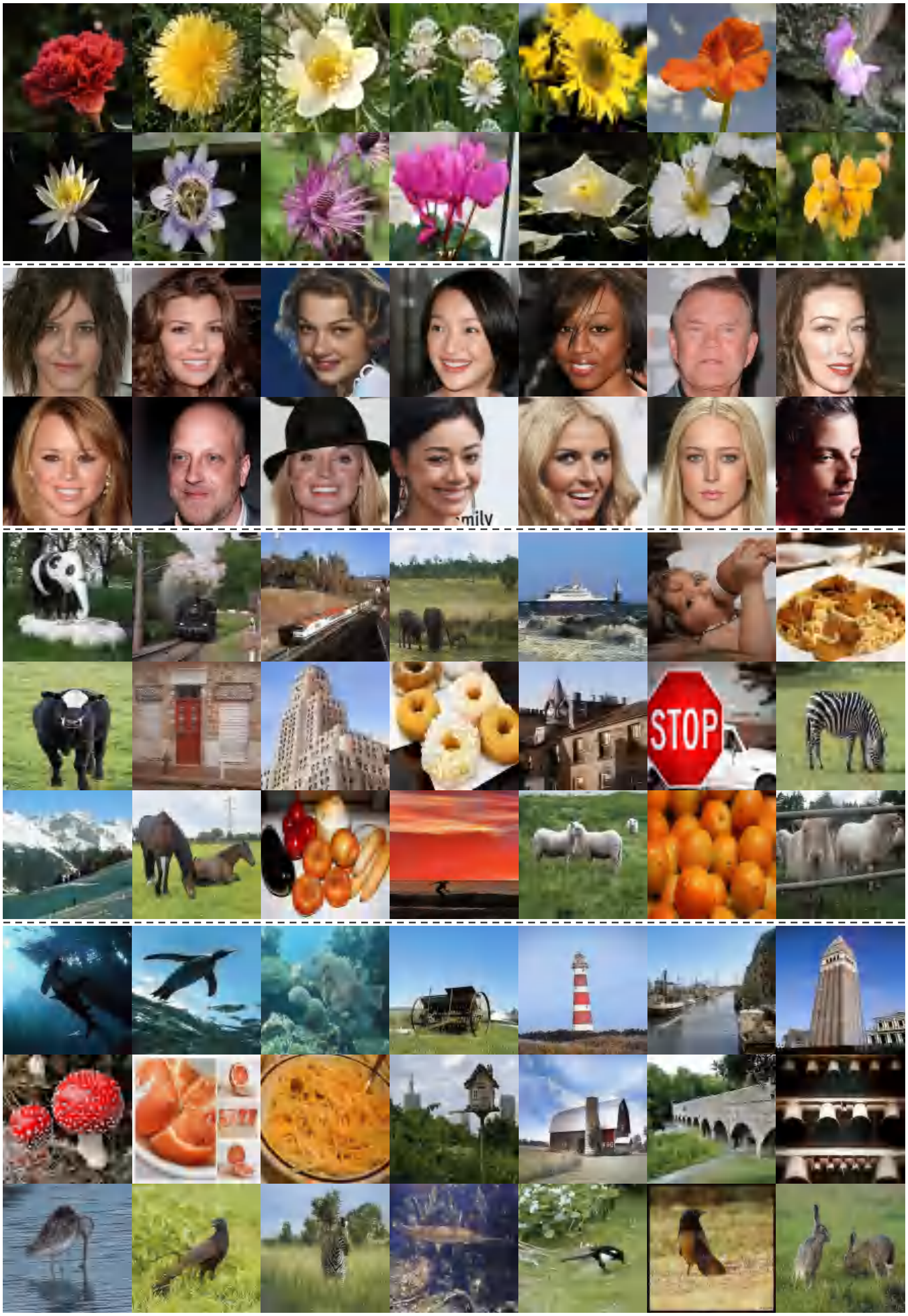

Figure 13. More generation results.

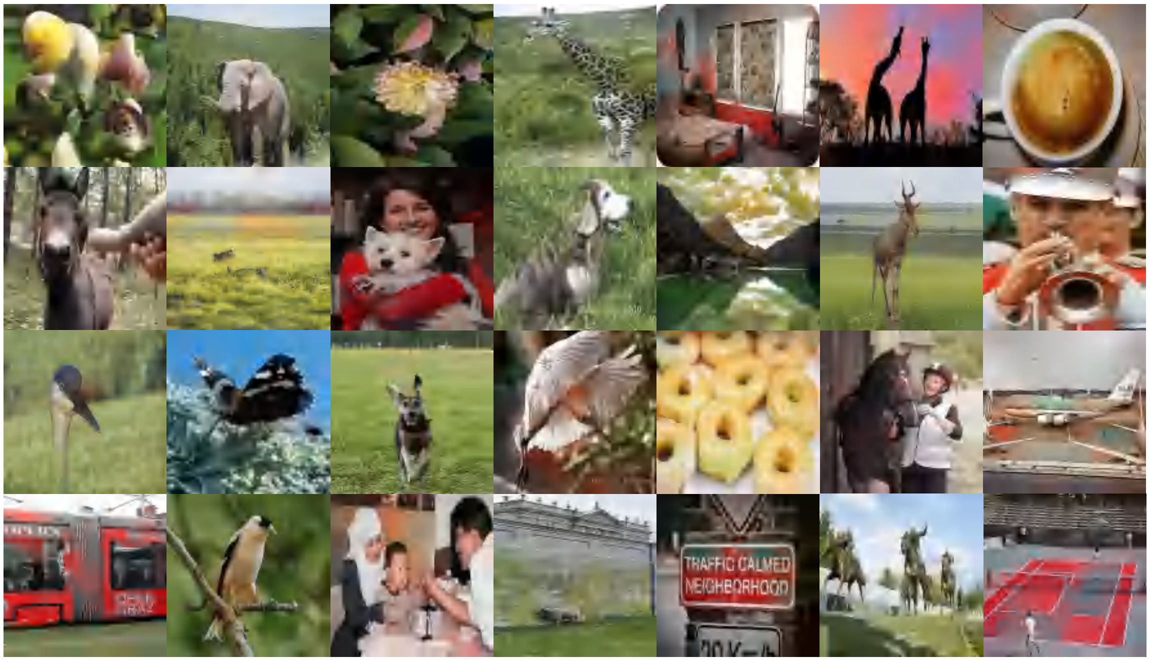

Figure 14. Some failure cases.