Spatially Consistent Transformer for Colorization in Monochrome-Color Dual-Lens System

Xuan Dong, Chang Liu, Xiaoyan Hu, Kang Xu, Weixin Li*

***Abstract*—We study the colorization problem in monochrome-color dual-lens camera systems, i.e. colorizing the gray image from the monochrome camera using the color image from the color camera as reference. In related methods, cost volume based CNN methods achieve the state-of-the-art results, but they are costly in GPU memory due to building the 4D cost volume. Recently, some slice-wise cross-attention based methods are proposed for related problems. The slice-wise cross-attention has much less costs in GPU memory but directly using them for this colorization problem cannot generate competing results. We make use of the non-local computation property of cross-attention to propose a transformer based method. To overcome the limitations of straight-forward slice-wise cross-attention, we propose the spatially consistent cross-attention (SCCA) block to encourage pixels of slices across different epipolar lines in the gray image to find spatially consistent correspondence with pixels of the reference color image. And, to further reduce the memory cost while keeping the colorization accuracy, we design a pyramid processing strategy to cascade a series of SCCA blocks with smaller slice size and perform the colorization from coarse to fine. To extract more powerful image features, we use several regional self-attention (RSA) blocks with U-style connections. Experimental results show that we outperform the state-of-the-art methods largely on the synthesized datasets of Cityscapes, Sintel, and SceneFlow, and the real monochrome-color dual-lens dataset.**

***Index Terms*—Spatially Consistent Cross-Attention, Pyramid Processing, Transformer.**

I. Introduction

Monochrome-color dual-lens camera systems have been widely used in popular smart phones, e.g. Huawei P40, Mate40, etc. Between the monochrome and color cameras, there exist different hardwares, e.g. the color filter array, and different software modules, e.g. white balance, demosaic, etc. As a result, the monochrome camera has better light efficiency [1], [2] than the color camera. Thus, as shown in Fig. 1, the gray image $\mathbf{I}^{\mathbf{M}} \in \mathbb{R}^{h\times w}$, where h is the image height and w is the image width, from the monochrome camera has higher quality, i.e. signal-noise ratio, than the color image $\mathbf{R}^{\mathbf{C}} \in \mathbb{R}^{h\times w\times 3}$ from the color camera but lacks color information. And, by recovering colors of $\mathbf{I}^{\mathbf{M}}$ using $\mathbf{R}^{\mathbf{C}}$ as reference, the colorization result $\mathbf{I}^{\mathbf{C}*} \in \mathbb{R}^{h\times w\times 3}$ will have higher quality than $\mathbf{R}^{\mathbf{C}}$. Different from the other kinds of colorization problems, e.g. automatic [3], [4], scribble-based [5], [6], reference-based [7], [8], etc., in the monochrome-color dual-lens colorization problem, the recovered colors of $\mathbf{I}^{\mathbf{C}*}$ should be of high quality and faithful to the physical colors of the scenes according to the requirement of camera systems. And, the pixels of $\mathbf{R}^{\mathbf{C}}$ that locate in the same epipolar line with each pixel of $\mathbf{I}^{\mathbf{M}}$ can provide strong clues of color information. So, when searching for corresponding pixels in $\mathbf{R}^{\mathbf{C}}$, the search range can be the 1D epipolar line instead of the 2D whole image, making the solving of the problem easier and less computational costly.

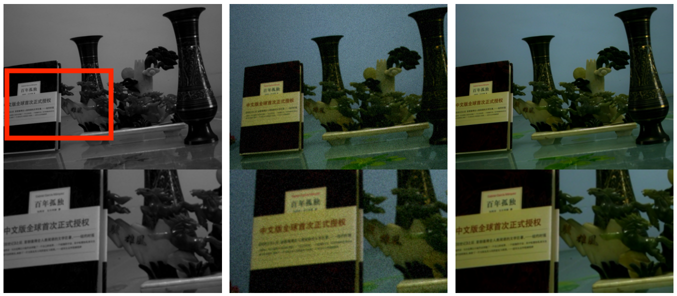

(a) Input pair of gray image $\mathbf{I}^{\mathbf{M}}$ and color image $\mathbf{R}^{\mathbf{C}}$. (b) Colorization result $\mathbf{I}^{\mathbf{C}*}$.

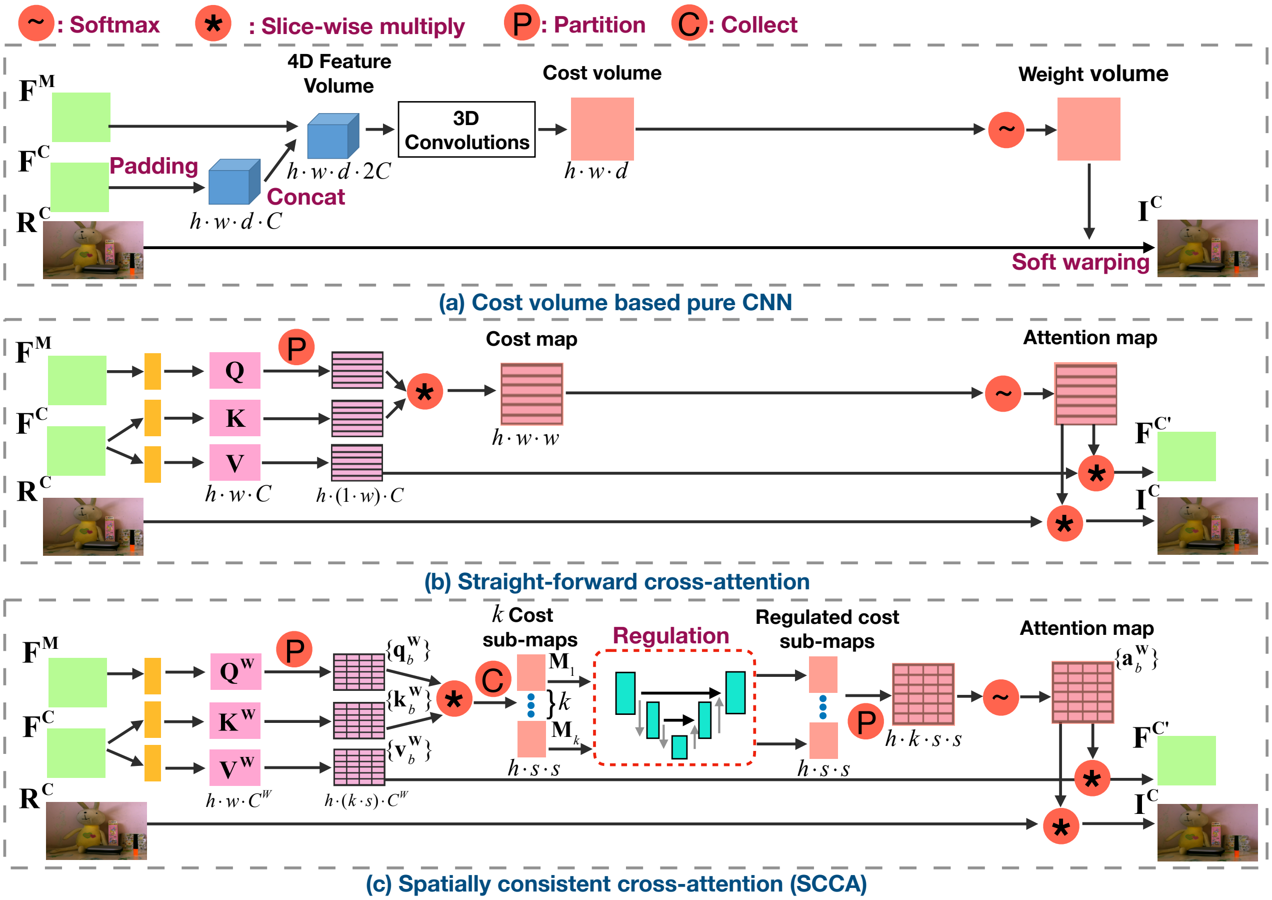
Fig. 1: The gray image $\mathbf{I}^{\mathbf{M}}$ and color image $\mathbf{R}^{\mathbf{C}}$ in the input pair are shot by the monochrome and color cameras, respectively. We propose a spatially consistent transformer to learn to colorize $\mathbf{I}^{\mathbf{M}}$ using $\mathbf{R}^{\mathbf{C}}$ as reference, and get the colorization result $\mathbf{I}^{\mathbf{C}*}$. The region marked with the red box is shown in the second line.

In the literature, 1) the first kind of solution is the hand-crafted methods. Jeon et al. [1] use a traditional hand-crafted stereo matching method to search for the disparity of pixels between the pair of images, warp the color image, and use post-processing to correct wrongly colorized regions due to wrongly estimated disparity and occlusions. But the hand-crafted methods need a lot of manually pre-defined parameters, which are not always robust in practice. 2) With the success of convolutional neural networks (CNN) in various computer vision and image processing problems, the second kind of solution is the CNN based deep learning methods. Because the convolution operation is quite local, to deal with the large-displacement problem, the existing methods, e.g. Dong et al. [9], [10], have to build the 4D cost volume and regulate it with 3D convolutions to obtain the correspondences, and use them to perform soft warping of the reference color image to get the colorization result. As shown in Fig. 2 (a), the size of image features is $h \times w \times C$, where C is the feature channel number. And they need to add another dimension to model the correspondence relationships of pixels between gray and

*Corresponding Author. Email: weixinli@buaa.edu.cn.

Xuan Dong, Chang Liu, Xiaoyan Hu, Kang Xu are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, P. R. China. Weixin Li is with the School of Computer Science and Engineering, Beihang University, Beijing, China and Zhongguancun Laboratory, Beijing, P. R. China.

Manuscript received xxxx; revised xxxx.



(a) Cost volume based pure CNN

(b) Straight-forward cross-attention

(c) Spatially consistent cross-attention (SCCA)

Fig. 2: Related solutions and our idea of the spatially consistent cross-attention (SCCA) structure (best viewed in color).

color images with the 1D search range d. The value of d needs to be large enough to deal with the pixel with the maximum displacement, resulting in the 4D cost volume with the high memory cost of $O(h \cdot w \cdot d \cdot C)$. 3) Recently, cross-attention based transformer structure provides a non-local deep learning computation framework and is successfully used in related problems of stereo matching and super resolution [11], [12]. As shown in Fig. 2 (b), the image is partitioned into h slices and each slice, with the slice size of $1 \times w$, contains one line of pixels. And in each slice, the correspondence relationships of every pair of pixels are estimated by the slice-wise cross-attention, achieving the memory cost of $O(h{\cdot}w{\cdot}w)$. But, in the current slice-wise cross-attention process, each line of pixels is treated as isolated data and the results across neighboring epipolar lines may be inconsistent in some un-certain regions, e.g. textureless regions.

We propose a transformer based method to make use of the non-local computation property of cross-attention. To overcome the limitations of the straight-forward slice-wise cross-attention, our insights are that 1) the correspondence relationships of neighboring slices across different epipolar lines can communicate with each other during the cross-attention process so as to obtain spatially consistently attended weights and thus result in spatially consistent colorization results. And 2) in each SCCA block, instead of separating each row into a slice like the straight-forward cross-attention, we separate the image into slices with the slice size of $1 \times s$ where s is smaller than the image width w, and we use the pyramid processing strategy to combine a series of SCCA blocks of different resolutions to reduce the memory cost from $O(h \cdot w \cdot w)$ to $O(h \cdot w \cdot s)$ without affecting the colorization accuracy.

Based on our insights, we propose the spatially consistent transformer. The overall structure is shown in Fig. 3. In the feature extraction part, we cascade a series of regional self-attention (RSA) blocks with U-style skip connections in multiple scales to encourage the features to have global and local information. In the colorization part, 1) we propose the spatially consistent cross-attention (SCCA) block, as shown in Fig. 2 (c). Different from the straight-forward cross-attention in Fig. 2 (b), after performing the slice-wise multiplication of $\mathbf{q}_b^\mathbf{W}$ and $\mathbf{k}_b^\mathbf{W}$ to get the cost values $\mathbf{c}_b^\mathbf{W} \in \mathbb{R}^{s \times s}$ which contains the correspondence costs of every pair of pixels in the slice b, we collect $\mathbf{c}_b^\mathbf{W}$ across all h lines to build k cost sub-maps ($\mathbf{M}_1 \in \mathbb{R}^{h \times s \times s}$ to $\mathbf{M}_k \in \mathbb{R}^{h \times s \times s}$, and $k = \frac{w}{s}$). And we perform multi-scale convolutions for each cost sub-map via a U-Net to make cost values of slices across different lines communicate with each other, so as to obtain spatially consistently regulated cost values. 2) We take a pyramid processing way. At each pyramid level i, the soft warping results $\mathbf{F}_i^{\mathbf{C}'}$ and $\mathbf{I}_i^\mathbf{C}$ from $\mathbf{F}_i^\mathbf{C}$ and $\mathbf{R}_i^\mathbf{C}$ are obtained by the SCCA block, respectively. In addition, to correct errors due to occlusions and large-displacement, we propose a feature correction CNN block to get the corrected feature $\mathbf{F}_i^{\mathbf{C}''}$ from

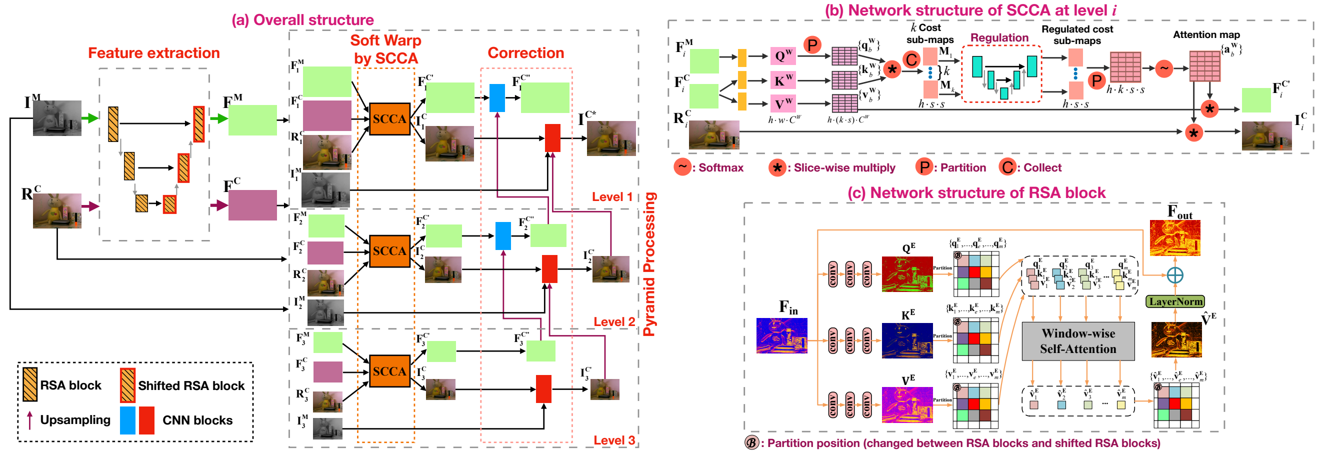


Fig. 3: The overall structure of our spatially consistent transformer (a), and the detailed network structures of the SCCA (b) and RSA blocks (c) (best viewed in color).

$\mathbf{F}_i^{\mathbf{C}'}$ with the upsampled result of $\mathbf{F}_{i+1}^{\mathbf{C}''}$ as guidance, and an image correction CNN block to get the corrected image $\mathbf{I}_i^{\mathbf{C}'}$ from $\mathbf{I}_i^{\mathbf{C}}$ with $\mathbf{I}_i^{\mathbf{M}}$ and the upsampled result of $\mathbf{I}_{i+1}^{\mathbf{C}'}$ as guidance.

Experiments are conducted on both synthesized datasets, i.e. CityScapes [13], Sintel [14], and SceneFlow [15], and real monochrome-color dual-lens dataset [16]. And the results show that we outperform the comparison methods largely.

The key contributions of this paper include that 1) we propose the SCCA to encourage pixels of slices across neighboring epipolar lines to obtain spatially consistent correspondence with pixels of the reference color image, so as to improve the colorization accuracy. 2) We propose the spatially consistent transformer to use small slice size and cascade a series of SCCA blocks with the pyramid processing strategy to reduce the memory cost while keeping the colorization accuracy.

II. Related Works

In related dual-lens colorization methods, as mentioned in Sec. I, Jeon et al. [1] use a stereo matching based method, but hard warping the reference color image by the estimated disparity cannot generate accurate colorization results, especially in occlusion regions. The works in [2], [9], [10], [16] make use of soft warping and propose the cost volume based pure CNN model for the colorization. But, building the 4D feature volume and regulating it with 3D convolutions have high memory costs.

Besides dual-lens colorization, there exist a set of other colorization tasks, including automatic [3], [4], [17]–[24], scribble-based [5], [6], [25], [26], domain-specific [27]–[35], text-based [36]–[39], diverse [40]–[42], reference-based [7], [8], [27], [30], [43]–[48], and video colorizations [49]–[52]. But the challenge of these tasks usually lies in that the reference color information is rough-grained, e.g. sparse human scribbles, text descriptions, reference images from other scenes, or even no reference at all in automatic colorization. So, related methods mostly focus on matching the rough-grained reference color information with the input gray image and exploiting clues of textures, structures, semantic meaning, etc. in the input gray image itself to generate the results. But, they usually do not consider how to estimate the fine-grained colors if fine-grained and strong related color reference exists. As a result, using these methods in our problem usually results in rough-grained colorization results. In some other colorization tasks, e.g. diverse, domain-specific, and video colorizations, the data have specific property, e.g. sketch images, SAR images, cartoon images, etc. And their optimization goals are different from ours, e.g. different colorization results in diverse colorization, temporally consistent results in video colorization, etc. Due to these differences, their methods are not proper for our problem.

For correspondence searching in related tasks, e.g. stereo matching [11], stereo super resolution [12], etc., existing methods include hand-crafted ones that use feature matching and matching cost aggregation like [53], pure CNN methods, e.g. cost volume based methods [54], and recently proposed cross-attention based methods [11], [12]. The work in [11] performs post-processing to use the spatial consistency property of neighboring pixels to correct the error disparity values by results of neighboring pixels, but the cross-attention blocks do not consider the spatially consistency of neighboring pixels across different epipolar lines. The axial attention [55], which is similar with the example (b) of Fig. 2, i.e. straight-forward cross-attention, also lets pixels of differently partitioned slices only communicate with the ones within the same slice. We notice that, within the cross-attention computation, regulating the cost values of slices across neighboring epipolar lines is more direct and beneficial to generate spatially consistent results. So, in this paper, we propose the spatially consistent cross-attention and build the model with pyramid processing strategy .

Vision transformers have achieved promising results in various vision tasks, including high-level tasks [56], [56], [57] of image classification, object detection, etc., and low-level tasks of single image super resolution [58], automatic colorization [59], etc. Most of them are proposed to provide a stronger backbone than traditional CNN structures for the visual feature extraction, e.g. global self-attention based methods of ViT [56], DeiT [60], and regional self-attention based methods like Swin Transformer [57]. Several works, e.g. IPT [58] and

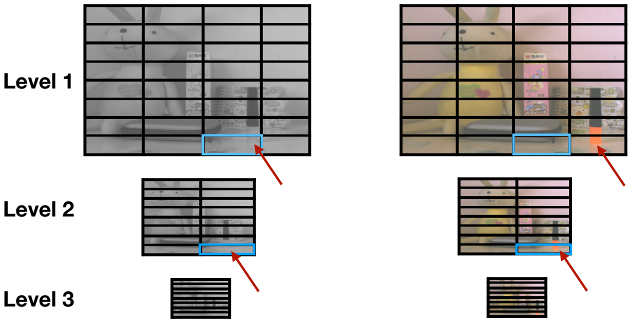


Fig. 4: An example to explain why the pyramid processing can solve the large-displacement problem (Pyramid levels are 3 for simplification). At level 1, the pixels of the marker pen (pointed by the red arrow) in the marked slice (marked in blue) of the gray image cannot find corresponding pixels in the color image by the slice-wise cross-attention because they are partitioned into another slice. But, at level 2, the pixels of the marker pen are partitioned into the same slice between the two images. And at level 3, the slice width is equal to the image width, so any large-displacement pixel can find its corresponding pixel.

Colorization Transformer [59], use these transformer backbones for low level tasks and achieve good results. Inspired by these successful use of vision transformers, especially in low-level tasks, we adopt the transformer techniques and propose a spatially consistent transformer according to the specific requirement of colorization in dual-lens systems.

III. Method

The overall structure of our model is shown in Fig. 3.

We use cross-attention to search for the correspondences between the input pair of monochrome and color images because the non-local deep learning structure in cross-attention, i.e. computing the attention weights between the query and key features and using the estimated attention weights to soft-warp the value feature, provides a natural implementation of the correspondence searching. In comparison with the popular memory-costly cost volume based CNN structure, cross-attention is more efficient and can avoid building the costly 4D cost volume. While there exist several works to use cross-attention for correspondence searching, e.g. stereo super resolution method PASSR [12], cross-attention based stereo matching [11], they always let the computation of each slice be performed separately without any communication. In comparison, during the cross-attention computation, we insert a U-Net to let the attention weights between different slices communicate with each other. This can encourage the estimated attention weights to be more spatially consistent so as to generate more accurate correspondences. In the spatially consistent cross-attention (SCCA) block, as shown in Fig. 2 (c), for each slice b, after we perform slice-wise multiplication between the query feature $\mathbf{q}_b^{\mathbf{W}}$ and the key feature $\mathbf{k}_b^{\mathbf{W}}$ to get the original slice-wise cost values $\mathbf{c}_b^{\mathbf{W}} \in \mathbb{R}^{s\times s}$ which contains the correspondence costs of every pair of pixel in the slice b, we collect the cost values $\mathbf{c}_b^{\mathbf{W}}$ across all h epipolar lines to get in total k cost sub-maps, denoted as $\mathbf{M}_1 \in \mathbb{R}^{h\times s\times s}$ to $\mathbf{M}_k \in \mathbb{R}^{h\times s\times s}$. Each cost sub-map $\mathbf{M}_p \in \mathbb{R}^{h\times s\times s}$ can be seen as the feature map of the 2D image with the size of $h\times s$, and the feature dimension with the size of s contains the correspondence costs along the search range s. By performing 2D convolutions for $\mathbf{M}_p$, for each element (y, x) and its neighboring elements $(y+\Delta y, x+\Delta x)$ in different lines or/and columns, $\mathbf{M}_p(y, x)$ and $\mathbf{M}_p(y+\Delta y, x+\Delta x)$ will be convolved and thus their correspondence costs along the search range will be convolved, resulting in the communication of correspondence costs of pixels across different epipolar lines. We perform multi-scale convolutions for each cost sub-map $\mathbf{M}_p$ via a U-Net to regulate the correspondence costs to be spatially consistent, so as to help generate spatially consistent outputs $\mathbf{F}^{\mathbf{C}'}$ and $\mathbf{I}^{\mathbf{C}}$.

In the soft warp and correction parts of Fig. 3, we design a pyramid based pipeline to combine a series of SCCA blocks in different resolutions to soft warp $\mathbf{R}^{\mathbf{C}}$ to the colorization result $\mathbf{I}^{\mathbf{C}*}$ using $\mathbf{F}^{\mathbf{M}}$ and $\mathbf{I}^{\mathbf{M}}$ as guidance. To overcome the large-displacement problem, when using the SCCA block for the correspondence searching, a straight-forward method is to set the slice size as $s = w$, but this will lead to the cost to be $O(hww)$. To further reduce the cost, we use the pyramid based pipeline and the slice size s can be set to be smaller than w, so as to reduce the cost to $O(hws)$. The slice width of SCCA blocks across all levels i is set as $s = \frac{w}{16}$ to achieve efficient computation within the pyramid processing. At each pyramid level i, the core part is the proposed SCCA block. Using $\mathbf{F}_i^{\mathbf{M}}$ as guidance, it soft warps $\mathbf{F}_i^{\mathbf{C}}$ and $\mathbf{R}_i^{\mathbf{C}}$ to generate $\mathbf{F}_i^{\mathbf{C}'}$ and $\mathbf{I}_i^{\mathbf{C}}$. The slice size in SCCA is set as $1\times s$ across all levels i ($s = \frac{w}{16}$ in this paper). The slice height is set to be 1 due to the 1D relative-displacement of pixels between $\mathbf{I}^{\mathbf{M}}$ and $\mathbf{R}^{\mathbf{C}}$. As explained in Fig. 4, at the top pyramid level, the slice width s is much smaller than the image width w and helps reduce the memory cost a lot. And at the bottom pyramid level, s is equal to the image width so as to be large enough to deal with the large-displacement problem. To repair errors due to occlusions and limited search range using the $1\times s$ slice, $\mathbf{F}_i^{\mathbf{C}'}$ and $\mathbf{I}_i^{\mathbf{C}}$ are corrected by the feature correction CNN block and image correction CNN block, respectively.

In the feature extraction part of Fig. 3, instead of using traditional CNN layers for the feature extraction, inspired by SWIN, ViT etc., we extract features $\mathbf{F}^{\mathbf{M}} \in \mathbb{R}^{h\times w\times C^E}$ and $\mathbf{F}^{\mathbf{C}} \in \mathbb{R}^{h\times w\times C^E}$ of the input gray image $\mathbf{I}^{\mathbf{M}} \in \mathbb{R}^{h\times w}$ and the reference color image $\mathbf{R}^{\mathbf{C}} \in \mathbb{R}^{h\times w\times 3}$ respectively by using a series of regional self-attention (RSA) blocks, where h is the image height, w is the image width, and C^E is the feature channel number. The advantage of RSA in comparison with CNN is that it is better to extract non-local features of the images and is also good at extracting local features of the images. The RSA block is similar with the Swin block [57] and we further revise it from the following two aspects. 1) Similar with [61], we use convolutional layers instead of patch embedding to extract the query, key, and value features from the input because convolution is better to extract local features of neighboring pixels for low level vision tasks. And 2) we use the U-style short connections to cascade the RSA blocks in multiple scales, so that the features can capture both non-local/global and local information.

A. Spatially Consistent Cross-Attention (SCCA)

SCCA blocks are used similarly in all the pyramid levels. For simplification, at any pyramid level i, we name the input of $\mathbf{F}_i^{\mathbf{M}}$, $\mathbf{F}_i^{\mathbf{C}}$ and $\mathbf{R}_i^{\mathbf{C}}$ as $\mathbf{F}^{\mathbf{M}}$, $\mathbf{F}^{\mathbf{C}}$ and $\mathbf{R}^{\mathbf{C}}$ in this subsection, respectively.

As shown in Fig. 2 (c), first, we use three CNN blocks to extract the query feature $\mathbf{Q}^{\mathbf{W}}$ from $\mathbf{F}^{\mathbf{M}}$, the key feature $\mathbf{K}^{\mathbf{W}}$ and value feature $\mathbf{V}^{\mathbf{W}}$ from $\mathbf{F}^{\mathbf{C}}$, respectively. Each CNN block has three 3×3 convolutional layers and ReLu. Second, $\mathbf{Q}^{\mathbf{W}}$, $\mathbf{K}^{\mathbf{W}}$ and $\mathbf{V}^{\mathbf{W}}$ are partitioned into slices with the slice size of $1 \times s$. And each line of pixels is partitioned into $k = \frac{w}{s}$ slices. Third, the original cost values of each slice b, i.e. $\mathbf{c}_b^{\mathbf{W}}$, is calculated by

$$\mathbf{c}_b^{\mathbf{W}} = \mathbf{q}_b^{\mathbf{W}} \mathbf{k}_b^{\mathbf{W}(T)} / \sqrt{C^W} + \mathbf{B}^{\mathbf{W}}, \tag{1}$$

where we follow [57] to add the relative position bias $\mathbf{B}^{\mathbf{W}}$ and feature dimension C^W in this equation. The original cost values $\mathbf{c}_b^{\mathbf{W}}$ of slices across h epipolar lines are collected to k cost sub-maps, i.e. $\mathbf{M}_1 \in \mathbb{R}^{h \times s \times s}$ to $\mathbf{M}_k \in \mathbb{R}^{h \times s \times s}$, and we use a U-Net to process the k cost sub-maps separately so that the cost values can be regulated with context information across different epipolar lines. This processing helps to obtain spatially consistent cost values, i.e. pixels of neighboring epipolar lines in the input gray image have similar correspondence with pixels in the color image. Then, the regulated cost sub-maps are partitioned to slices with the size of $1 \times s$ again, which is denoted as $\widehat{\mathbf{c}}_b^{\mathbf{W}}$. And we obtain the attention values $\mathbf{a}_b^{\mathbf{W}}$ of each slice b by

$$\mathbf{a}_b^{\mathbf{W}} = SoftMax(\widehat{\mathbf{c}}_b^{\mathbf{W}}), \tag{2}$$

and the slice-wise soft warping for the partitioned slices $\mathbf{f}_b^{\mathbf{C}}$ of $\mathbf{V}^{\mathbf{W}}$ is performed by

$$\mathbf{f}_b^{\mathbf{C}'} = \mathbf{a}_b^{\mathbf{W}} \mathbf{f}_b^{\mathbf{C}}. \tag{3}$$

The soft warping results $\mathbf{f}_b^{\mathbf{C}'}$ of all slices are collected to obtain the soft warped feature $\mathbf{F}^{\mathbf{C}'}$. Similarly, the partitioned slices $\mathbf{r}_b^{\mathbf{C}}$ of $\mathbf{R}^{\mathbf{C}}$ are soft warped by

$$\mathbf{i}_b^{\mathbf{C}} = \mathbf{a}_b^{\mathbf{W}} \mathbf{r}_b^{\mathbf{C}}, \tag{4}$$

and the soft warped results $\mathbf{i}_b^{\mathbf{C}}$ of all slices are collected to obtain the soft warped image $\mathbf{I}^{\mathbf{C}}$.

B. Pyramid Processing

We build n pyramid levels (n is set as 5 in this paper) in the proposed pyramid processing pipeline. The pyramid processing is performed level by level to get the final result. And the partition slice size in the SCCA block is set as $1 \times s$ (s is set as $\frac{w}{16}$, where w is the image width) across all pyramid levels. The combination of n and s ensures that 1) at the bottom pyramid level, the width of the partitioned slice in the SCCA block, i.e. s, is equal to the image width so as to solve the large-displacement problem. 2) At the top pyramid level, s is smaller than w so as to reduce the memory cost.

As shown in Fig. 3, at each pyramid level i, with the features $\mathbf{F}_i^{\mathbf{M}}$ and $\mathbf{F}_i^{\mathbf{C}}$, and the image $\mathbf{R}_i^{\mathbf{C}}$ as input, the SCCA block generates the soft warped feature of $\mathbf{F}_i^{\mathbf{C}}$, i.e. $\mathbf{F}_i^{\mathbf{C}'}$, and the soft warped image of $\mathbf{R}_i^{\mathbf{C}}$, i.e. $\mathbf{I}_i^{\mathbf{C}}$.

To correct errors of $\mathbf{F}_i^{\mathbf{C}'}$, the upsampled feature of $\mathbf{F}_{i+1}^{\mathbf{C}''}$ from level $i+1$ is used as the guidance, and the feature correction CNN block (constructed by three 3×3 convolutional layers and ReLu), denoted as f, processes them by $\mathbf{F}_i^{\mathbf{C}''} = f(U(\mathbf{F}_{i+1}^{\mathbf{C}''}), \mathbf{F}_i^{\mathbf{C}'})$, where U denotes the upsampling process using bilinear interpolation by the ratio of 2, and $\mathbf{F}_i^{\mathbf{C}''}$ is the corrected feature.

To correct errors of $\mathbf{I}_i^{\mathbf{C}}$, similarly, we use an image correction CNN block to get the corrected image $\mathbf{I}_i^{\mathbf{C}'}$. Here, we use $\mathbf{F}_i^{\mathbf{C}''}$ and $\mathbf{F}_i^{\mathbf{M}}$ as the feature guidance, and the gray image $\mathbf{I}_i^{\mathbf{M}}$ and the corrected color image $\mathbf{I}_{i+1}^{\mathbf{C}'}$ from level $i+1$ as the image guidance to perform the correction. And $\mathbf{I}_i^{\mathbf{C}'} = g_2(U(\mathbf{I}_{i+1}^{\mathbf{C}'}), \mathbf{I}_i^{\mathbf{M}}, g_1(\mathbf{F}_i^{\mathbf{M}}, \mathbf{F}_i^{\mathbf{C}''}, \mathbf{I}_i^{\mathbf{C}}))$, where both g_1 and g_2 are constructed by three 3×3 convolutional layers and ReLu. The corrected image $\mathbf{I}_{i+1}^{\mathbf{C}'}$ from level $i+1$ is used as guidance for solving the large-displacement problem and the gray image $\mathbf{I}_i^{\mathbf{M}}$ is used as guidance for solving the occlusion problem.

C. Feature Extraction

Regional self-attention (RSA) blocks are cascaded with the U-style short connections for feature extraction.

In each RSA block, given the input feature $\mathbf{F}_{\mathbf{in}}$, we use three CNN blocks to extract the query, key and value feature maps $\mathbf{Q}^{\mathbf{E}}$, $\mathbf{K}^{\mathbf{E}}$, and $\mathbf{V}^{\mathbf{E}}$, respectively. Each CNN block has three 3×3 convolutional layers and ReLu. Then, we partition the feature maps into non-overlapped windows (with the window size of $a \times a = 16 \times 16$ in this paper), and get the features $\mathbf{q}_e^{\mathbf{E}}$, $\mathbf{k}_e^{\mathbf{E}}$, $\mathbf{v}_e^{\mathbf{E}}$ of each window e. We follow [57] to add the relative position bias $\mathbf{B}^{\mathbf{E}}$ and feature dimension C^E into the softmax, and the self-attention within each window is performed by

$$\hat{\mathbf{v}}_e^{\mathbf{E}} = SoftMax(\mathbf{q}_e^{\mathbf{E}} \mathbf{k}_e^{\mathbf{E}(T)} / \sqrt{C^E} + \mathbf{B}^{\mathbf{E}}) \mathbf{v}_e^{\mathbf{E}}. \tag{5}$$

Then we concatenate $\hat{\mathbf{v}}_e^{\mathbf{E}}$ over all windows to get $\hat{\mathbf{V}}^{\mathbf{E}}$, and get the output feature $\mathbf{F}_{\mathbf{out}}$ of this block by $\mathbf{F}_{\mathbf{out}} = LN(\hat{\mathbf{V}}^{\mathbf{E}}) + \mathbf{F}_{\mathbf{in}}$, where LN denotes a LayerNorm layer.

Among different RSA blocks, the Swin shift [57] of the window partition positions is used for the marked block in Fig. 3 so that different sets of neighboring pixels can be included into the same windows among different blocks. In this way, the pixels can have more chances to communicate with all their neighboring pixels.

D. Loss

When training the model on synthesized datasets, where the ground-truth color image $\mathbf{GT}$ of the input gray image $\mathbf{I}^{\mathbf{M}}$ exists, we use SSIM [62] as the metric to design the training loss L_s to measure the quality of the colorization result $\mathbf{I}^{\mathbf{C}*}$ in Cb and Cr color channels:

$$L_s = 1 - \frac{1}{2}(SSIM(\mathbf{I}_{Cb}^{\mathbf{C}*}, \mathbf{GT}_{Cb}) + SSIM(\mathbf{I}_{Cr}^{\mathbf{C}*}, \mathbf{GT}_{Cr})). \tag{6}$$

When training the model on real monochrome-color dual-lens datasets, where $\mathbf{GT}$ does not exist, we follow [10], [16] to perform the colorization in a cycle way. And we evaluate the

TABLE I: Layers descriptions of our model. ‘conv’, ‘cat’, and ‘pooling’ are short for convolutional layer, concatenate, and average pooling (with stride=2), respectively.

Input	Input Dim.	Output	Output Dim.	Layers Description
RSA at level j in Feature Extraction				
$\mathbf{F_{in}}$	$h_j{\cdot}w_j{\cdot}C^E$	$\mathbf{Q^E}$	$h_j{\cdot}w_j{\cdot}C^E$	(3x3 conv, ReLu)×3
$\mathbf{F_{in}}$	$h_j{\cdot}w_j{\cdot}C^E$	$\mathbf{K^E}$	$h_j{\cdot}w_j{\cdot}C^E$	(3x3 conv, ReLu)×3
$\mathbf{F_{in}}$	$h_j{\cdot}w_j{\cdot}C^E$	$\mathbf{V^E}$	$h_j{\cdot}w_j{\cdot}C^E$	(3x3 conv, ReLu)×3
$\mathbf{Q^E}$/$\mathbf{K^E}$/$\mathbf{V^E}$	$h_j{\cdot}w_j{\cdot}C^E$	$\mathbf{q}_e^\mathbf{E}$/$\mathbf{k}_e^\mathbf{E}$/$\mathbf{v}_e^\mathbf{E}$	$a \cdot a \cdot C^E$	Partition
$\mathbf{q}_e^\mathbf{E}$,$\mathbf{k}_e^\mathbf{E}$,$\mathbf{v}_e^\mathbf{E}$	$a \cdot a \cdot C^E$	$\hat{\mathbf{v}}_e^\mathbf{E}$	$a \cdot a \cdot C^E$	Eq. (5)
$\hat{\mathbf{v}}_e^\mathbf{E}$	$a \cdot a \cdot C^E$	$\hat{\mathbf{V}}^\mathbf{E}$	$h_j{\cdot}w_j{\cdot}C^E$	Collect
SCCA at level i, where $k = \frac{w_i}{s}$				
$\mathbf{F^M}$	$h_i{\cdot}w_i{\cdot}C^W$	$\mathbf{Q^W}$	$h_i{\cdot}w_i{\cdot}C^W$	(3x3 conv, ReLu)×3
$\mathbf{F^C}$	$h_i{\cdot}w_i{\cdot}C^W$	$\mathbf{K^W}$	$h_i{\cdot}w_i{\cdot}C^W$	(3x3 conv, ReLu)×3
$\mathbf{F^C}$	$h_i{\cdot}w_i{\cdot}C^W$	$\mathbf{V^W}$	$h_i{\cdot}w_i{\cdot}C^W$	(3x3 conv, ReLu)×3
$\mathbf{Q^W}$/$\mathbf{K^W}$/$\mathbf{V^W}$	$h_i{\cdot}w_i{\cdot}C^W$	$\mathbf{q}_b^\mathbf{W}$/$\mathbf{k}_b^\mathbf{W}$/$\mathbf{v}_b^\mathbf{W}$	$1{\cdot}s{\cdot}C^W$	Partition
$\mathbf{q}_b^\mathbf{W}$,$\mathbf{k}_b^\mathbf{W}$	NA	$\mathbf{c}_b^\mathbf{W}$	$s{\cdot}s$	Eq. (1)
$\mathbf{c}_b^\mathbf{W}$	$s{\cdot}s$	cost sub-maps $\mathbf{M}_p$	$h_i{\cdot}s{\cdot}s$	collect
cost sub-maps $\mathbf{M}_p$	$h_i{\cdot}s{\cdot}s$	regulated sub-maps	$h_i{\cdot}s{\cdot}s$	U-Net
U-Net in SCCA				
cost sub-map $\mathbf{M}_p$	$h_i{\cdot}s{\cdot}s$	r1	$\frac{h_i}{2}\cdot\frac{s}{2}{\cdot}2s$	(3x3 conv, ReLu)×2,pooling
r1	$\frac{h_i}{2}\cdot\frac{s}{2}{\cdot}2s$	r2	$\frac{h_i}{4}\cdot\frac{s}{4}{\cdot}4s$	(3x3 conv, ReLu)×2,pooling
r2	$\frac{h_i}{4}\cdot\frac{s}{4}{\cdot}4s$	r3	$\frac{h_i}{8}\cdot\frac{s}{8}{\cdot}8s$	(3x3 conv, ReLu)×2,pooling
r3	$\frac{h_i}{8}\cdot\frac{s}{8}{\cdot}8s$	r4	$\frac{h_i}{4}\cdot\frac{s}{4}{\cdot}4s$	(3x3 conv, ReLu)×2,upsample
cat(r4,r2)	$\frac{h_i}{4}\cdot\frac{s}{4}{\cdot}8s$	r5	$\frac{h_i}{2}\cdot\frac{s}{2}{\cdot}2s$	(3x3 conv, ReLu)×2,upsample
cat(r5,r1)	$\frac{h_i}{2}\cdot\frac{s}{2}{\cdot}4s$	r6	$h_i{\cdot}s{\cdot}s$	(3x3 conv, ReLu)×2,upsample
cat(r6,$\mathbf{M}_p$)	$h_i{\cdot}s{\cdot}2s$	regulated cost sub-map	$h_i{\cdot}s{\cdot}s$	(3x3 conv, ReLu)×2

first-time colorization results by the structure similarity loss $L_{structure}$ between $\mathbf{I}_Y^{\mathbf{C}*}$ and $\mathbf{I^M}$:

$$L_{structure} = 1 - SSIM_{cs}(\mathbf{I}_Y^{\mathbf{C}*}, \mathbf{I^M}), \tag{7}$$

where $SSIM_{cs}$ is the revised SSIM metric (setting the coefficient of luminance dimension as 0, and the coeffients of contrast and structure dimensions as 1 in SSIM). We also evaluate the second-time colorization results $\mathbf{I^{Cycle}}$ by the cycle consistency loss L_{cycle} between $\mathbf{I^{Cycle}}$ and the input color image $\mathbf{R^C}$ in the Cb and Cr color channels:

$$L_{cycle} = 1 - \frac{1}{2}(SSIM(\mathbf{I}_{Cb}^{\mathbf{Cycle}}, \mathbf{R}_{Cb}^{\mathbf{C}}) + SSIM(\mathbf{I}_{Cr}^{\mathbf{Cycle}}, \mathbf{R}_{Cr}^{\mathbf{C}})). \tag{8}$$

Finally, the total loss for the real monochrome-color dual-lens datasets L_r is $L_r = \frac{1}{2}(L_{structure} + L_{cycle})$.

IV. Experimental Results

A. Datasets

Our experiments are performed on both synthesized datasets, including Cityscapes [13], Sintel [14], and SceneFlow [15], and real monochrome-color dual-lens dataset in [2]. Following [1] and [2], to simulate the real monochrome-color dual-lens systems, the original color images in synthesized datasets are distorted with two setups, named Setup1 and Setup2 (by adding Gaussian noises with the standard deviation of $0.03\sqrt{\kappa}$ and $0.07\sqrt{\kappa}$ respectively, where κ represents the noise-free signal intensity). PSNR (Peak Signal-to-Noise Ratio) and SSIM [62] are used as the evaluation metrics for objective evaluation.

B. Implementation Details

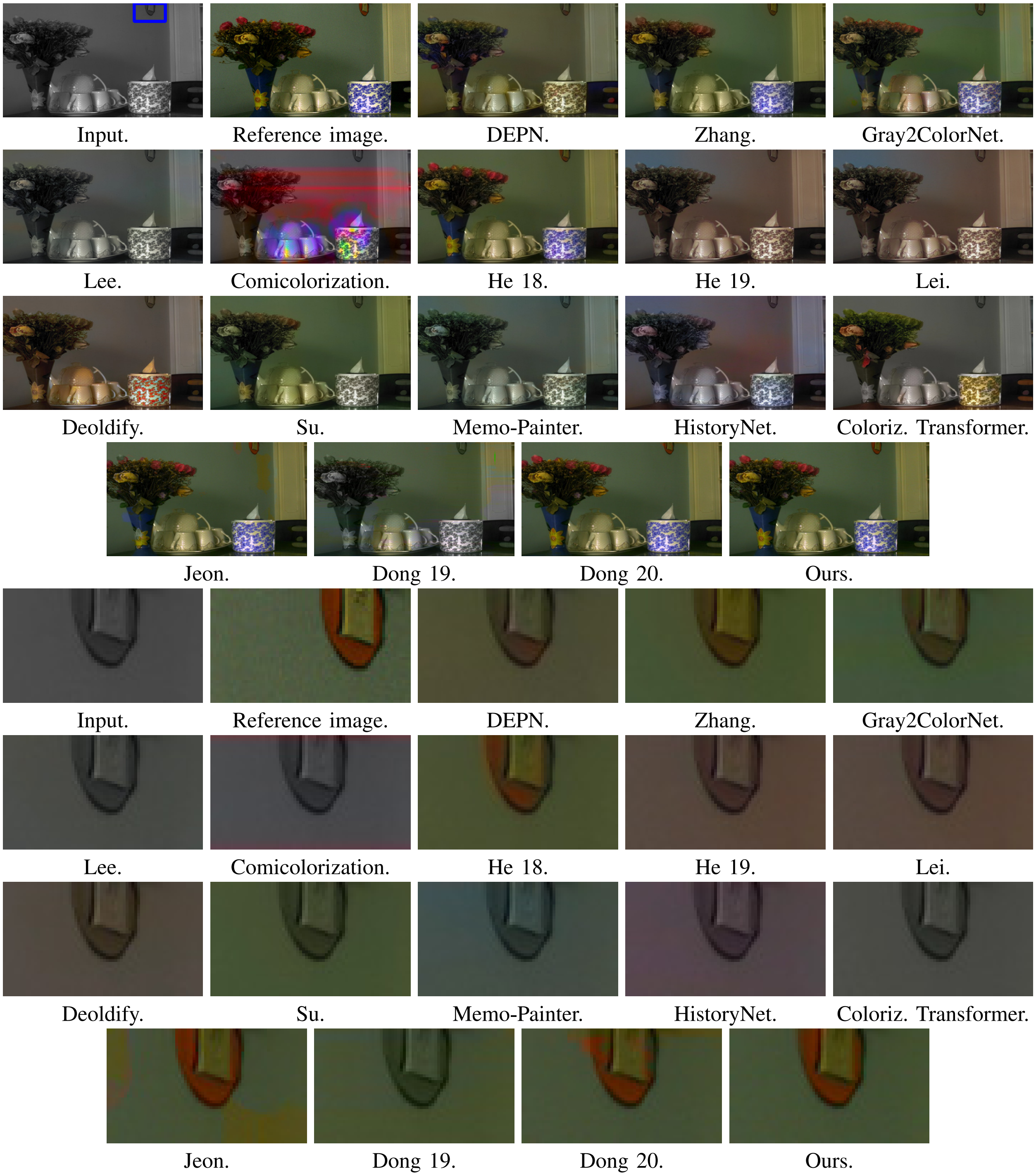
The detailed layers descriptions of our model are provided in Table I. We implement our model using MindSpore. We use an Nvidia 3090 card to train the models using the conventional Adam optimizer with b1 = 0.9, b2 = 0.999 for 200 epochs on each dataset. The initial learning rate is set as 0.0001 and decayed by half every 20 epoch. The batchsize is set as 1. Training our model roughly takes 1 day for 200 epochs. In each dataset, 80% randomly selected images are used for training and the left 20% images are used for testing.

C. Comparison Methods

We compare with state-of-the-art reference-based colorization algorithms, i.e. the methods of Lu et al. (Gray2ColorNet) [48], Lee et al. [27], Furusawa et al. (Comicolorization) [30], He et al. 2018 [8], and He et al. 2019 [46], and Zhang et al. [63], deep learning based automatic colorization algorithms, i.e. the methods of Su et al. [22], Yoo et al. (Memo-Painter) [23], Xiao et al. (DEPN) [24], Lei et al. [52], DeOldify [64], Jin et al. (HistoryNet) [65], and Kumar et al. (Colorization Transformer) [59] and state-of-the-art monochrome-color dual-lens colorization algorithms, i.e. the methods of Jeon et al. [1], Dong et al. 2019 [2], Dong et al. 2020 [16]. For fair comparison, the learning based colorization methods are fine-tuned on each dataset with the same training setting of ours.

D. Results

The quantitative evaluation results on the three synthesized datasets are shown in Tables II and III. Some qualitative example results on the real monochrome-color dual-lens dataset



Input. Reference image. DEPN. Zhang. Gray2ColorNet.

Lee. Comicolorization. He 18. He 19. Lei.

Deoldify. Su. Memo-Painter. HistoryNet. Coloriz. Transformer.

Jeon. Dong 19. Dong 20. Ours.

Input. Reference image. DEPN. Zhang. Gray2ColorNet.

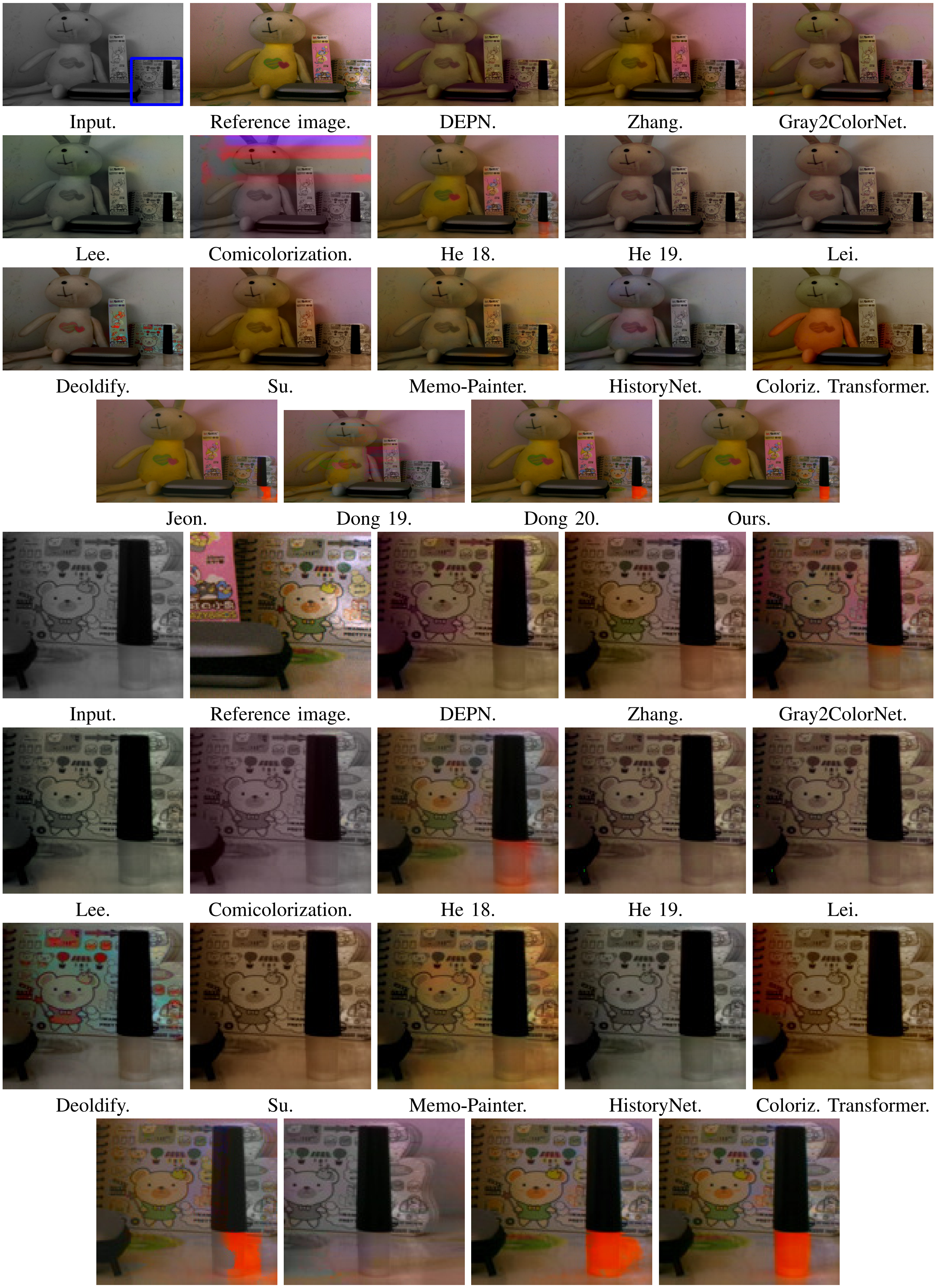
Lee. Comicolorization. He 18. He 19. Lei.

Deoldify. Su. Memo-Painter. HistoryNet. Coloriz. Transformer.

Jeon. Dong 19. Dong 20. Ours.

Fig. 5: Example results of different methods on the real monochrome-color dual-lens dataset. The region marked in blue is enlarged for each image.

[2] and synthesized datasets are shown in Figs. 5 to 8. Due to the lack ground-truth colors of the input gray images from the monochrome camera, direct objective evaluation is not possible on the real monochrome-color dual-lens dataset. To perform quantitative evaluation on the real monochrome-color dual-lens dataset, we follow [16] to let the colorization methods perform cycle colorization, i.e. firstly colorizing the input gray image using the input color image as reference to get the first-time colorization result, and secondly colorizing the de-colored map of the input color image using the first-time colorization result as reference to get the second-time colorization result. And we evaluate the second-time colorization results with the Cb and Cr color channel values of the input color images as ground-truth. The PSNR and SSIM values are shown in Table IV. The results are not the direct evaluation of the colorization quality but may provide some indirect verification of the colorization quality of different methods. Besides the theoretical analysis of the costs of cost volume based pure CNN, straight-forward slice-wise cross-attention and our method in Sec. I, we also provide the practical costs in



Input. Reference image. DEPN. Zhang. Gray2ColorNet.

Lee. Comicolorization. He 18. He 19. Lei.

Deoldify. Su. Memo-Painter. HistoryNet. Coloriz. Transformer.

Jeon. Dong 19. Dong 20. Ours.

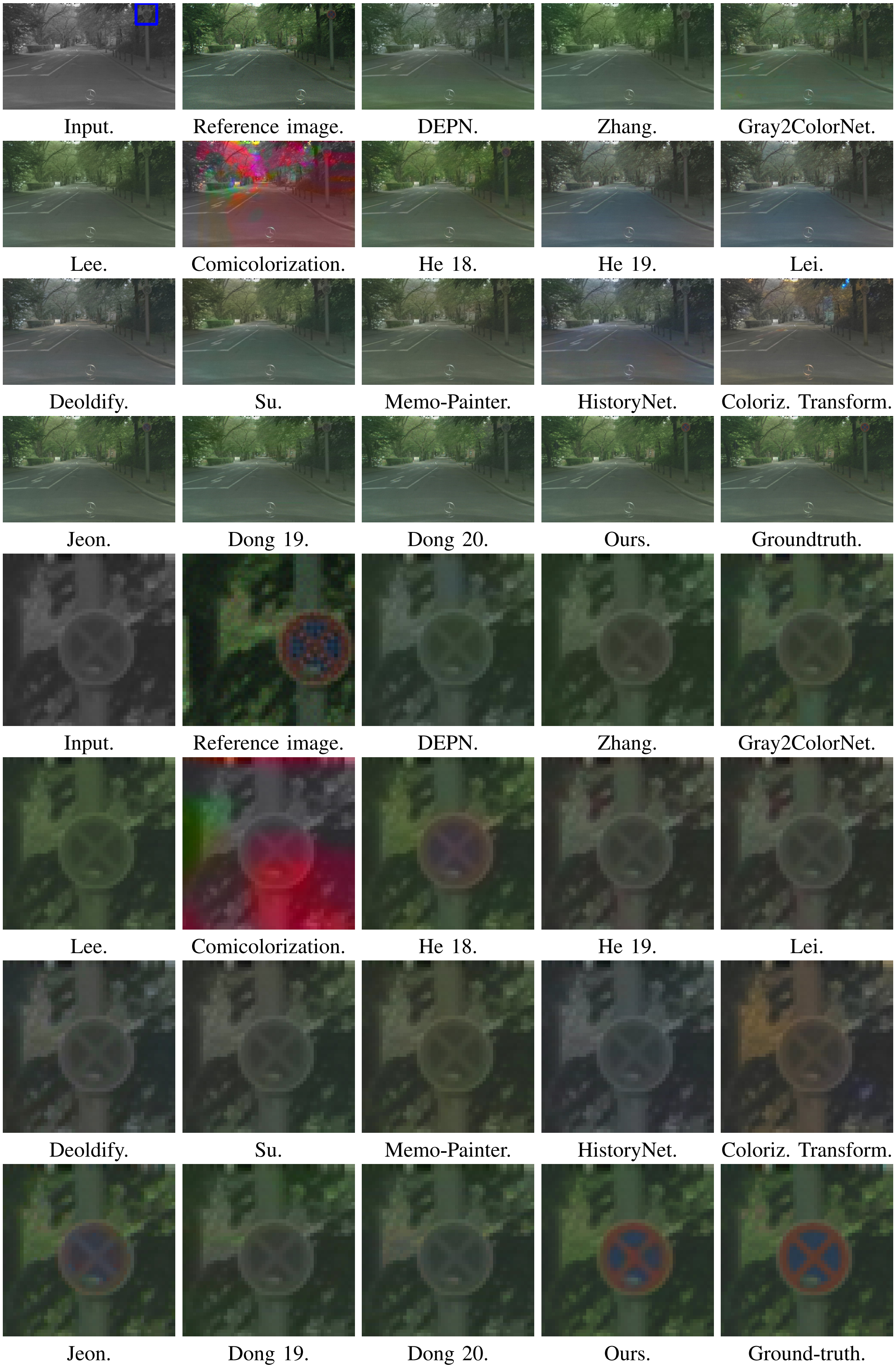
Input. Reference image. DEPN. Zhang. Gray2ColorNet.

Lee. Comicolorization. He 18. He 19. Lei.

Deoldify. Su. Memo-Painter. HistoryNet. Coloriz. Transformer.

Jeon. Dong 19. Dong 20. Ours.

Fig. 6: Example results of different methods on the real monochrome-color dual-lens dataset. The region marked in blue is enlarged for each image.



Input. Reference image. DEPN. Zhang. Gray2ColorNet.

Lee. Comicolorization. He 18. He 19. Lei.

Deoldify. Su. Memo-Painter. HistoryNet. Coloriz. Transform.

Jeon. Dong 19. Dong 20. Ours. Groundtruth.

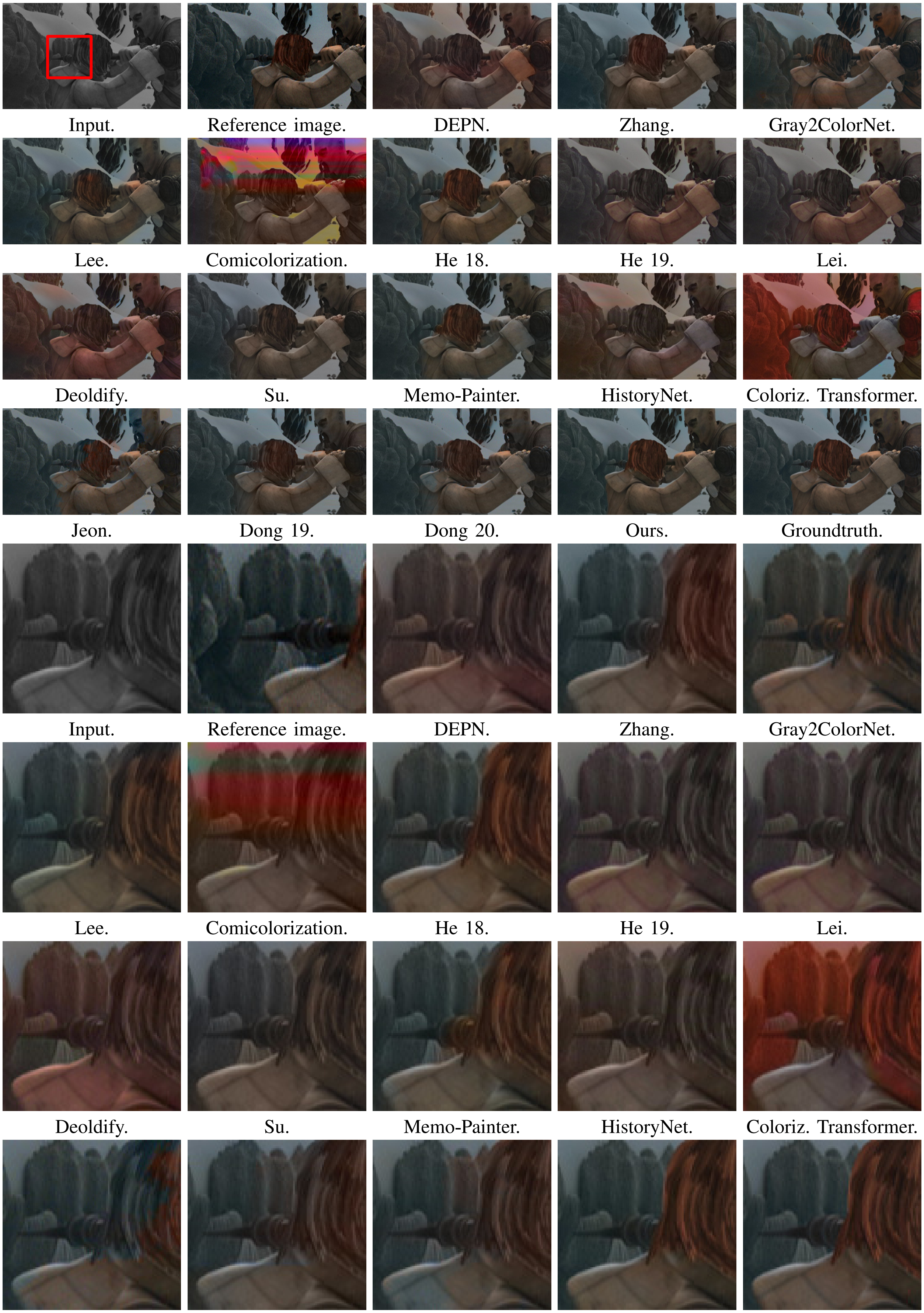
Input. Reference image. DEPN. Zhang. Gray2ColorNet.

Lee. Comicolorization. He 18. He 19. Lei.

Deoldify. Su. Memo-Painter. HistoryNet. Coloriz. Transform.

Jeon. Dong 19. Dong 20. Ours. Ground-truth.

Fig. 7: Example colorization results of comparison methods and ours on CityScapes dataset. The region marked in box is enlarged for each image.



Input. Reference image. DEPN. Zhang. Gray2ColorNet.

Lee. Comicolorization. He 18. He 19. Lei.

Deoldify. Su. Memo-Painter. HistoryNet. Coloriz. Transformer.

Jeon. Dong 19. Dong 20. Ours. Groundtruth.

Input. Reference image. DEPN. Zhang. Gray2ColorNet.

Lee. Comicolorization. He 18. He 19. Lei.

Deoldify. Su. Memo-Painter. HistoryNet. Coloriz. Transformer.

Jeon. Dong 19. Dong 20. Ours. Ground-truth.

Fig. 8: Example colorization results of comparison methods and ours on Sintel dataset. The region marked in box is enlarged for each image.

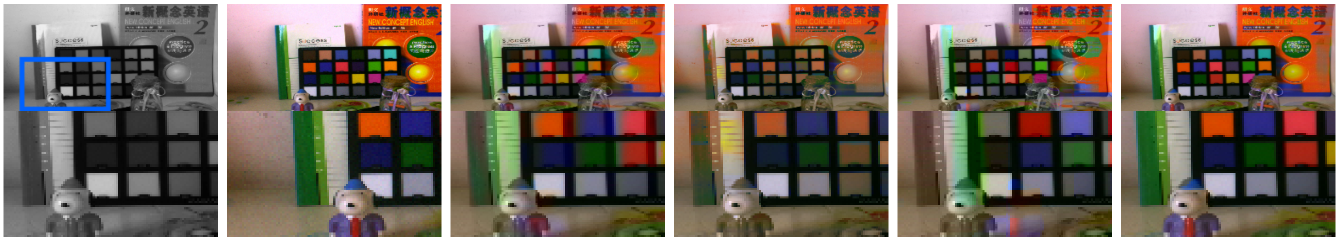

(a) Input pair of images. (b) No RSA. (c) No regulation. (d) Only top level. (e) Ours.

Fig. 9: Example ablation study results on the real monochrome-color dual-lens dataset.

TABLE II: Average PSNR (dB)/SSIM values of different methods in the three synthesized datasets on Setup1.

	CityScapes	Sintel	SceneFlow
Gray2ColorNet [48]	39.97/0.969	39.13/0.949	34.95/0.939
Lee [27]	42.36/0.977	37.46/0.955	37.63/0.951
Comicolorization [30]	23.50/0.880	23.46/0.865	22.35/0.845
He18 [8]	41.99/0.976	38.92/0.959	38.82/0.951
He19 [46]	42.34/0.977	38.20/0.958	36.99/0.945
DEPN [24]	38.65/0.973	31.67/0.939	32.55/0.929
Zhang [63]	41.59/0.977	38.36/0.957	37.54/0.942
Su [22]	38.98/0.976	35.49/0.949	34.19/0.941
Memo-Painter [23]	42.06/0.977	40.50/0.964	40.40/0.962
Lei [52]	35.07/0.970	31.19/0.935	30.97/0.930
DeOldify [64]	36.53/0.968	30.40/0.920	31.57/0.925
HistoryNet [65]	33.88/0.968	29.67/0.931	29.87/0.923
Colorization Transformer [59]	30.00/0.951	26.56/0.906	28.78/0.919
Jeon [1]	39.01/0.909	36.35/0.906	36.34/0.889
Dong19 [2]	44.26/0.982	43.88/0.983	45.18/0.988
Dong20 [16]	45.22/0.983	43.95/0.984	45.50/**0.989**
Ours	**46.73/0.987**	**45.82/0.985**	**46.05**/0.985

TABLE III: Average PSNR (dB)/SSIM values on Setup2.

	CityScapes	Sintel	SceneFlow
Gray2ColorNet [48]	39.88/0.969	34.98/0.948	34.70/0.937
Lee [27]	42.36/0.977	37.46/0.955	37.62/0.951
Comicolorization [30]	22.97/0.871	23.29/0.863	22.31/0.841
He18 [8]	42.09/0.976	39.05/0.959	39.09/0.952
He19 [46]	41.89/0.977	38.00/0.957	36.98/0.944
DEPN [24]	38.56/0.973	31.64/0.939	32.51/0.928
Zhang [63]	41.37/0.976	38.15/0.956	37.19/0.941
Su [22]	38.98/0.976	35.49/0.949	34.19/0.941
Memo-Painter [23]	42.06/0.977	40.50/0.964	40.40/0.962
Lei [52]	35.07/0.970	31.19/0.935	30.97/0.930
DeOldify [64]	36.53/0.968	30.40/0.920	31.57/0.925
HistoryNet [65]	33.88/0.968	29.67/0.931	29.87/0.923
Colorization Transformer [59]	30.00/0.951	26.56/0.906	28.78/0.919
Jeon [1]	34.13/0.741	33.71/0.795	33.32/0.745
Dong19 [2]	43.21/0.979	42.71/0.977	44.16/0.984
Dong20 [16]	44.41/0.980	42.63/0.979	44.31/0.983
Ours	**46.52/0.986**	**45.60/0.984**	**45.19/0.984**

TABLE IV: Average PSNR (db) and SSIM values of the second-time colorization results of different colorization methods on the real monochrome-color dual-lens dataset.

	PSNR	SSIM
Gray2ColorNet [48]	33.67	0.9135
Lee [27]	29.54	0.8888
Comicolorization [30]	25.53	0.8521
He18 [8]	33.68	0.8986
He19 [46]	34.02	0.9042
DEPN [24]	27.84	0.8747
Zhang [63]	31.14	0.9061
Su [22]	34.12	0.9204
Memo-Painter [23]	35.80	0.9193
Lei [52]	25.53	0.8612
Deoldify [64]	25.87	0.8587
HistoryNet [65]	25.30	0.8623
Colorization Transformer [59]	24.11	0.8495
Jeon [1]	34.44	0.8319
Dong19 [2]	31.25	0.9067
Dong20 [16]	42.99	0.9707
Ours	**43.82**	**0.9759**

TABLE V: Computational and GPU memory costs of cost volume based pure CNN (CV based CNN), straight-forward slice-wise cross-attention (Straight CA) and ours.

Methods	Time(s)	FLOPs(G)	Memory(MB)
CV based CNN	0.788	360.78	18557
Straight CA	0.683	2822.9	3954
Ours	**0.267**	**16.56**	**1426**

Table V, including the processing time per image, FLOPs per image, and the GPU memory cost. The practical costs verify our analysis that our method has less costs in GPU memory and computation.

As the results show, automatic colorization methods [22]–[24], [52], [59], [65] do not perform well in our problem, because the reference color image which contains many useful color clues is not used at all during the colorization. The results of reference-based colorization algorithms [8], [27], [30], [46], [48], [63], are usually not accurate. It is because they usually assume that the reference color image is from different locations and/or shot at different time and the contents within the pair of gray and color images just share similar semantics. Due to different assumptions, these methods mostly focus on matching the rough-grained reference color information with the input gray image. But, they usually do not consider how to estimate the fine-grained colors if fine-grained and strongly related color reference exists. Dual-lens colorization algorithms [1], [2], [16] are not competing with ours too. The combination of stereo matching and warping in [1] is not suitable for colorization, especially in occlusion regions, because estimating correct colors and disparity are two different problems. The cost volume based CNN models in [2], [16] have big memory costs in building the 4D feature volume and thus they have to set a small number for the feature channel to enable to colorize large-displacement pixels, which limits the learning ability of the CNN. In comparison, the proposed SCCA exploits the spatial consistency property of neighboring pixels to overcome the limitations of straight-forward cross-attention, i.e. pixels across different epipolar lines lack communications. In addition, we reduce the search range in the SCCA blocks to lower the memory cost while keeping the colorization accuracy with the pyramid processing pipeline.

We also performed a user study for the comparisons on the real monochrome-color dual-lens dataset [2] with the help of 30 annotators in total. There are five annotation choices, including ‘Perfect’, ‘Few Errors’, ‘Partly Wrong’, ‘Mostly Wrong’, and ‘Totally Wrong’. Each annotator annotates the

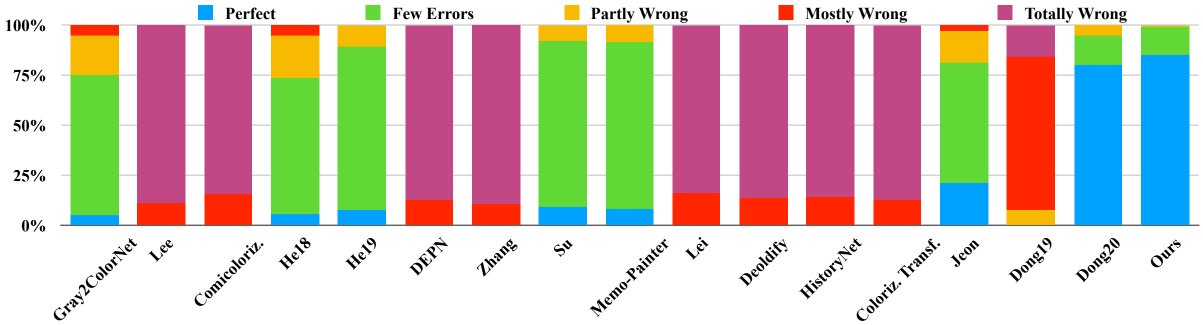
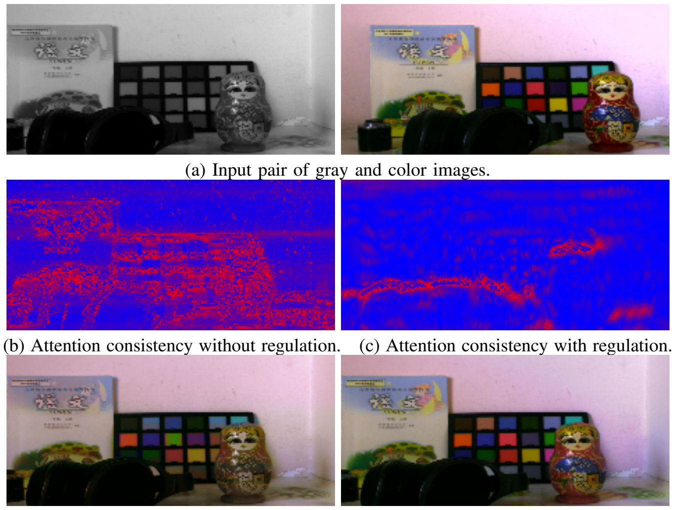


Fig. 10: User study results.

TABLE VI: Average PSNR/SSIM of the ablation study on Setup1.

	CityScapes	Sintel	SceneFlow
No RSA	45.67/0.984	44.24/0.983	45.29/0.983
No regulation	45.33/0.983	43.35/0.981	44.40/0.982
Only top level	37.12/0.957	36.36/0.952	36.59/0.962
Ours	**46.73/0.987**	**45.82/0.985**	**46.05/0.985**



(a) Input pair of gray and color images.

(b) Attention consistency without regulation. (c) Attention consistency with regulation.

(d) Colorization result without regulation. (e) Colorization result with regulation.

Fig. 11: An example to visualize the consistency of estimated attention values of pixels across different lines without and with the U-Net for the regulation. In (b) and (c), blue reflects small differences (i.e. high consistency) of the estimated attention values of each pixel in $\mathbf{a}_b^{\mathbf{W}}$ with its neighboring pixel in the next epipolar line. And red reflects big differences (i.e. poor consistency).

colorization results of the 16 comparing methods and ours. The whole set of images during the annotation includes 100 pairs which are randomly selected from the real monochrome-color dual-lens dataset. To judge outlier annotators, each annotator is asked to randomly re-annotate some results and we see the annotation as an outlier if the score differences are beyond one score level. The user study results, as shown in Fig. 10, show that our method gets higher perceptual scores than the others.

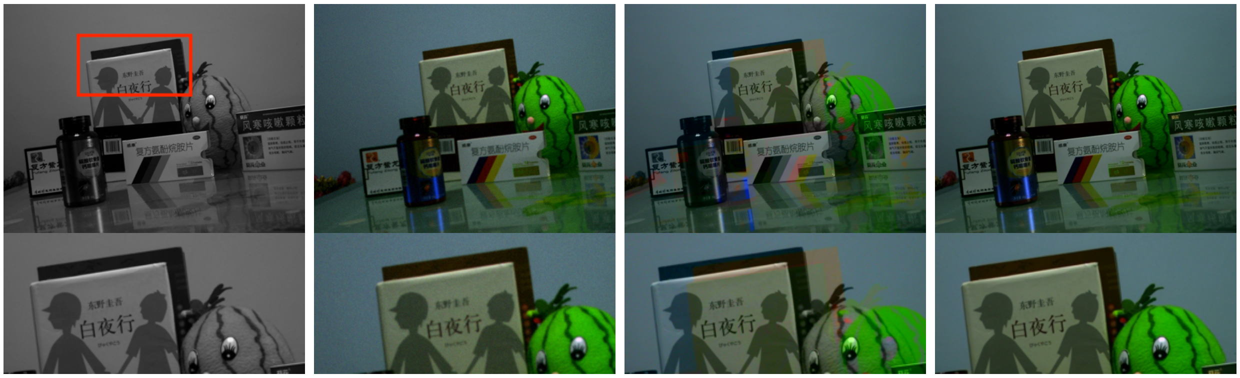
E. Ablation Study

We try different model variants in the proposed method, so as to evaluate the importance of the key modules of the proposed method. (1) We remove the RSA blocks and replace each RSA block with a three-layer CNN, named 'No RSA'. (2) We do not perform the regulation via the U-Net in the SCCA block, named 'No regulation'. (3) We only perform the processing at the top pyramid level without performing the other pyramid levels, named 'Only top level'. Quantitative results and some example results are shown in Table VI and Fig. 9.

The degradation of the colorization quality of 'No RSA' verifies that the RSA block is more powerful than traditional CNN for the feature extraction. The results of 'No regulation' are not competing with the proposed method because the regulation via the U-Net in the SCCA block can encourage pixels across neighboring lines in the input gray image to obtain consistent correspondence with pixels of the reference color image, and thus result in better colorization quality. In Fig. 11, we also visualize the consistency of estimated attention values of pixels across different lines without and with the U-Net for the regulation. For each pixel x_1 that belongs to the slice b_1 and its neighboring pixel x_2 in the next line that belongs to the slice b_2, we read their attention values in $\mathbf{a}_{b1}^{\mathbf{W}}$ and $\mathbf{a}_{b2}^{\mathbf{W}}$, respectively, calculate the absolute differences, and average them along the correspondence search dimension to obtain the visualization value of the pixel x_1. From Fig. 11, the visualized intermediate results also verify that the regulation by the U-Net can help obtain spatially consistent correspondence for neighboring pixels. The results of 'Only top level' are much lower than the proposed method. Because the slice size in the SCCA block is set as $1 \times s$ and s is much smaller than the image width, only performing the colorization on the top pyramid level will lead to limited search range of many pixels in the input gray image for the corresponding pixels in the reference color image. Since the corresponding pixels are out of the search range, the results are inevitably poor. In comparison, because our method performs the processing with multiple pyramid levels and the colorization of pixels with large-displacement can be solved by the other pyramid levels, the proposed pyramid processing pipeline can achieve high colorization quality.

V. Conclusions

We propose the spatially consistent transformer to solve the colorization problem in monochrome-color dual-lens systems.



(a) Input pair of gray image and color image . (b) Two inputs overlaid. (c) Our result.

Fig. 12: The gray image and color image in the input pair (a) are shot by the monochrome and color cameras, respectively. The overlaid result (b) (obtained by concatenating Y channel of the input gray image and $CbCr$ channels of the input color image) has obvious errors because the displacements of pixels between the input pair are not solved. The proposed spatially consistent transformer gets the colorization result (c) with correct colors. The region marked with the red box is shown in the second line.

Several regional self-attention (RSA) blocks with U-style connections are used for extracting features of input images. And we propose the spatially consistent cross-attention (SCCA) block to exploit the commonly used spatial consistency property of neighboring pixels to help obtain spatially consistent colorization results. A series of SCCA blocks are cascaded in a pyramid processing way to achieve an efficient and effective colorization framework. Experimental results show that the proposed spatially consistent transformer outperforms the state-of-the-art methods largely.

The limitations of the proposed method are that it can hardly work for the other kinds of colorization tasks, e.g. reference-based colorization, scribble-based colorization, etc., where the reference color image cannot provide plenty of useful color clues for the pixels of the input gray image in every epipolar line.

VI. Acknowledge

This work is sponsored by the National Nature Science Foundation of China (No. 61802026, 61806016, and 62076032) and CAAI-Huawei MindSpore Open Fund.

References

[1] H. G. Jeon, J. Y. Lee, S. Im, H. Ha, and I. S. Kweon, "Stereo matching with color and monochrome cameras in low-light conditions," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4086–4094, 2016.

[2] X. Dong, W. Li, X. Wang, and Y. Wang, "Learning a deep convolutional network for colorization in monochrome-color dual-lens system," *AAAI Conference on Artificial Intelligence*, 2019.

[3] R. Zhang, P. Isola, and A. Efros, "Colorful image colorization," *European Conference on Computer Vision*, pp. 649–666, 2016.

[4] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "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*, vol. 35, no. 4, 2016.

[5] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM transactions on graphics*, vol. 23, no. 3, pp. 689–694, 2004.

[6] R. Zhang, J. Zhu, P. Isola, X. Geng, A. Lin, T. Yu, and A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.

[7] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," *ACM transactions on graphics*, vol. 21, no. 3, pp. 277–280, 2002.

[8] M. He, D. Chen, J. Liao, P. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM SIGGRAPH*, 2018.

[9] X. Dong, W. Li, X. Hu, X. Wang, and Y. Wang, "A colorization framework for monochrome-color dual-lens systems using a deep convolutional network," *IEEE Transactions on Visualization and Computer Graphics, Early Access*, 2020.

[10] X. Dong, C. Liu, W. Li, X. Hu, X. Wang, and Y. Wang, "Self-supervised colorization towards monochrome-color camera systems using cycle cnn," *IEEE Transactions on Image Processing Early Access*, 2021.

[11] Z. Li, X. Liu, N. Drenkow, A. Ding, F. X. Creighton, R. H. Taylor, and M. Unberath, "Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers," *arXiv preprint arXiv:2011.02910*, 2020.

[12] L. Wang, Y. Wang, Z. Liang, Z. Lin, J. Yang, W. An, and Y. Guo, "Learning parallax attention for stereo image super-resolution," *CVPR*, 2019.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

[14] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," *European Conference on Computer Vision*, pp. 611–625, 2012.

[15] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D.Cremers, A.Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4040–4048, 2016.

[16] X. Dong, W. Li, X. Wang, and Y. Wang, "Cycle-cnn for colorization towards real monochrome-color camera systems," *AAAI Conference on Artificial Intelligence*, 2020.

[17] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," *ECCV*, pp. 577–593, 2016.

[18] G. Ozbulak, "Image colorization by capsule networks," *CVPR Workshop*, 2019.

[19] Y. Jin, B. Sheng, P. Li, and P. Chen, "Broad colorization," *TNNLS*, 2021.

[20] S. Wan, Y. Xia, L. Qi, Y. Yang, and M. Atiquzzaman, "Automated colorization of a grayscale image with seed points propagation," *TMM*, 2020.

[21] J. Zhao, J. Han, L. Shao, and C. Snoek, "Pixelated semantic colorization," *IJCV*, vol. 128, no. 3, pp. 818–834, 2020.

[22] J. Su, H. Chu, and J. Huang, "Instance-aware image colorization," *CVPR*, pp. 7968–7977, 2020.

[23] S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, and J. Choo, "Coloring with limited data: Few-shot colorization via memory augmented networks," *CVPR*, pp. 11 283–11 292, 2019.

[24] C. Xiao, C. Han, Z. Zhang, J. Qin, T. Wong, G. Han, and S. He, "Example-based colourization via dense encoding pyramids," *Computer Graphics Forum*, vol. 39, no. 12, pp. 1–14, 2019.

[25] Y. Ci, X. Ma, Z. Wang, H. Li, and Z. Luo, “User guided deep anime line art colorization with conditional adversarial networks,” *ACM MM*, pp. 1536–1544, 2018.

[26] Y. Xiao, J. Wu, J. Zhang, P. Zhou, Y. Zheng, C. Leung, and L. Kavan, “Interactive deep colorization and its application for image compression,” *TVCG*, 2019.

[27] J. Lee, E. Kim, Y. Lee, D. Kim, J. Chang, and J. Choo, “Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence,” *CVPR*, pp. 5801–5810, 2020.

[28] L. Zhang, C. Li, T. Wong, Y. Ji, and C. Liu, “Two-stage sketch colorization,” *TOG*, 2018.

[29] W. Chen and J. Hays, “Sketchgan: Towards diverse and realistic sketch to image synthesis,” *CVPR*, 2018.

[30] C. Furusawa, K. Hiroshiba, K. Ogaki, and Y. Odagiri, “Comicolorization: semi-automatic manga colorization,” *SIGGRAPH Asia*, 2017.

[31] M. Xie, C. Li, X. Liu, and T. Wong, “Manga filling style conversion with screentone variational autoencoder,” *TOG*, 2020.

[32] S. Chen, J. Zhang, L. Gao, Y. He, S. Xia, M. Shi, and F. Zhang, “Active colorization for cartoon line drawings,” *TVCG*, 2020.

[33] P. Wang and V. Patel, “Generating high quality visible images from sar images using cnns,” *IEEE Radar Conference*, pp. 570–575, 2018.

[34] L. Zhang, C. Li, E. Serra, Y. Ji, T. Wong, and C. Liu, “User-guided line art flat filling with split filling mechanism,” *CVPR*, 2021.

[35] X. Jin, Z. Li, K. Liu, D. Zou, X. Li, X. Zhu, Z. Zhou, Q. Sun, and Q. Liu, “Focusing on persons: Colorizing old images learning from modern historical movies,” *ACM MM*, 2021.

[36] C. Zou, H. Mo, C. Gao, R. Du, and H. Fu, “Language-based colorization of scene sketches,” *TOG*, 2019.

[37] H. Bahng, S. Yoo, W. Cho, D. K. Park, Z. Wu, X. Ma, and J. Choo, “Coloring with words: Guided image colorization through text-based palette generation,” *ECCV*, pp. 431–447, 2018.

[38] V. Manjunatha, M. Iyyer, J. B. Graber, and L. Davis, “Learning to color from language,” *North American Chapter of the Association for Computational Linguistics*, 2018.

[39] H. Kim, H. Y. Jhoo, E. Park, and S. Yoo, “Tag2pix: Line art colorization using text tag with secat and changing loss,” *ICCV*, 2019.

[40] A. Deshpande, J. Lu, M. Yeh, M. Chong, and D. Forsyth, “Learning diverse image colorization,” *CVPR*, pp. 6837–6845, 2017.

[41] Y. Wu, X. Wang, Y. Li, H. Zhang, X. Zhao, and Y. Shan, “Towards vivid and diverse image colorization with generative color prior,” *Arxiv*, 2021.

[42] S. Messaoud, D. Forsyth, and A. Schwing, “Structural consistency and controllability for diverse colorization,” *ECCV*, 2018.

[43] R. Ironi, D. Cohen-Or, and D. Lischinski, “Colorization by example,” *Rendering Techniques*, pp. 201–210, 2005.

[44] R. K. Gupta, A. Y. S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong, “Image colorization using similar images,” *ACM international conference on Multimedia*, pp. 369–378, 2012.

[45] M. He, J. Liao, L. Yuan, and P. Sander, “Neural color transfer between images,” *Arxiv*, 2017.

[46] M. He, J. Liao, D. Chen, L. Yuan, and P. Sander, “Progressive color transfer with dense semantic correspondences,” *ACM Transactions on Graphics*, vol. 38, no. 13, 2019.

[47] Z. Xu, T. Wang, F. Fang, Y. Sheng, and G. Zhang, “Stylization-based architecture for fast deep exemplar colorization,” *CVPR*, pp. 9363–9372, 2020.

[48] P. Lu, J. Yu, X. Peng, Z. Zhao, and X. Wang, “Gray2colornet: Transfer more colors from reference image,” *ACM MM*, pp. 3210–3218, 2020.

[49] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, “Tracking emerges by colorizing videos,” *ECCV*, 2018.

[50] Y. Zhao, L. Po, W. Yu, Y. Rehman, M. Liu, Y. Zhang, and W. Ou, “Vcgan: Video colorization with hybrid generative adversarial network,” *TMM*, 2021.

[51] S. Iizuka and E. Serra, “Deepremaster: Temporal source-reference attention networks for comprehensive video enhancement,” *TOG*, 2019.

[52] C. Lei and Q. Chen, “Fully automatic video colorization with self-regularization and diversity,” *CVPR*, 2019.

[53] C. Liu, J. Yuen, and A. Torralba, “Sift flow: Dense correspondence across scenes and its applications,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[54] K. Alex, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, “End-to-end learning of geometry and context for deep stereo regression,” *International Conference on Computer Vision*, 2017.

[55] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, “Axial attention in multidimensional transformers,” *Arxiv*, 2019.

[56] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[57] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” *International Conference on Computer Vision (ICCV)*, 2021.

[58] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, “Pre-trained image processing transformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 299–12 310.

[59] M. Kumar, D. Weissenborn, and N. Kalchbrenner, “Colorization transformer,” *ICLR*, 2021.

[60] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[61] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[62] Z. Wang, A. C. Bovik, H. Sheikh, and E. P. Simoncelli, “Image quality assessment from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[63] B. Zhang, M. He, J. Liao, P. Sander, L. Yuan, A. Bermak, and D. Chen, “Deep exemplar-based video colorization,” *CVPR*, 2019.

[64] J. Antic, “Deoldify: A deep learning based project for colorizing and restoring old images (and video!),” *Online*, 2019.

[65] X. Jin, Z. Li, K. Liu, D. Zou, X. Li, X. Zhu, Z. Zhou, Q. Sun, and Q. Liu, “Focusing on persons: Colorizing old images learning from modern historical movies,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1176–1184.