# PCT-Net: Full Resolution Image Harmonization Using Pixel-Wise Color Transformations

Julian Jorge Andrade Guerreiro[1,*] Mitsuru Nakazawa[2] Björn Stenger[2]

[1]The University of Tokyo [2]Rakuten Institute of Technology, Rakuten Group, Inc.

julianguerreio@outlook.de, {mitsuru.nakazawa, bjorn.stenger}@rakuten.com

## Abstract

*In this paper, we present* PCT-Net*, a simple and general image harmonization method that can be easily applied to images at full-resolution. The key idea is to learn a parameter network that uses downsampled input images to predict the parameters for pixel-wise color transforms (PCTs) which are applied to each pixel in the full-resolution image. We show that affine color transforms are both efficient and effective, resulting in state-of-the-art harmonization results. Moreover, we explore both CNNs and Transformers as the parameter network, and show that Transformers lead to better results. We evaluate the proposed method on the public full-resolution iHarmony4 dataset, which is comprised of four datasets, and show a reduction of the foreground MSE (fMSE) and MSE values by more than 20% and an increase of the PSNR value by 1.4dB, while keeping the architecture light-weight. In a user study with 20 people, we show that the method achieves a higher B-T score than two other recent methods.*

## 1. Introduction

Cutting and pasting parts of an image into another image is an important editing task, also referred to as image compositing. However, creating a composite image by simply adding a foreground region to a different image will typically produce unrealistic results due to different conditions at the time the images were taken. In order to reduce this discrepancy between foreground and background, image harmonization aims to align the colors by modifying the foreground region.

A variety of approaches have been proposed to solve this task using traditional statistical techniques as well as deep learning methods. Nonetheless, most of the research [5,12–15,17,23,28,30] has solely focused on low-resolution images ($256 \times 256$ pixels), whereas high resolution images have become in fact the standard for most real use cases. Since most approaches are built on convolutional neural networks (CNN), they would theoretically be able to process images of any size. However, due to poor scaling, the computational cost required for high resolution images render them effectively impractical.

More recently, some methods have started exploring high-resolution image harmonization [6,18,22,34] by leveraging a network that takes a low-resolution image as its input, but instead of predicting the final image, further processes the image according to the output of the network. While this allows us to apply high resolution image harmonization based on a low-resolution input, current models are either simplifying the problem for the sake of efficiency or employ a series of complex operations to improve performance. Following the general dual branch approach, we propose a light-weight model capable of harmonizing images at high resolutions. As shown in Fig. 1, our method achieves significant improvements in terms of foreground-normalized MSE (fMSE), but only requires roughly the same or less parameters, depending on the backbone.

We are able to outperform state-of-the-art methods on full resolution image harmonization by interpolating the network output in parameter space instead of introducing interpolation errors in the image space. Following the reasoning by Xue *et al*. [34], we argue that the parameter space contains less high frequency components which cause higher interpolation errors during upsampling. In contrast to [34], we greatly reduce the complexity by introducing pixel-wise color transformations (PCT) and find that a simple affine transformation is sufficient to achieve significant improvements. We show that this idea can easily be applied to both, CNN-based and Transformer-based models, while outperforming current state-of-the-art models in terms of reconstruction error.

In our approach, the backbone network predicts a set of parameters for each input pixel. Since this is done in low resolution, we interpolate the parameter map to match the full resolution of the original composite image. We then

[*]work conducted during an internship at Rakuten Group, Inc.

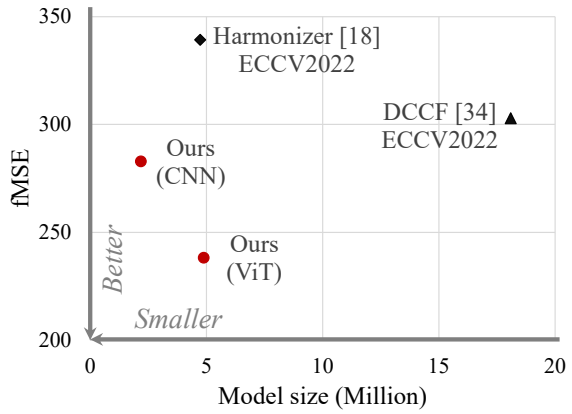


**Figure 1. Model size *vs*. performance (fMSE score) comparison.** *Even though our model size is smaller than others, our proposed models achieve better performance on full resolution images than prior work. The performance is calculated using the iHarmony4 dataset [7].*

apply the same, pre-determined PCT function to each pixel according to the predicted parameters. In order to find suitable parameters that represent an appropriate color transformation, the backbone network needs to consider spatial and semantic information within the image. When applying the PCT function, we change each pixel solely based on its value and the parameters predicted for that pixel position.

Our contributions can be summarized as follows:

- We introduce PCT-Net, an architecture based on a dual branch approach that is able to handle high-resolution images. It processes images at full resolution through a pixel-wise color transformation (PCT).
- To the best of our knowledge, we are the first to use a Transformer-based architecture for training and testing at full resolution. We further propose a novel training strategy for image harmonization where we do not resize the images, but instead evaluate the loss function on the full-resolution images.
- We demonstrate significant improvements in quantitative and qualitative performance compared to existing approaches, while retaining a light-weight and simple network architecture.

## 2. Related Work

**Image harmonization** describes the process of modifying the colors of the foreground of a composite image to blend with its background. Traditionally, this problem has been approached from either an image gradient-based view [16, 25, 31, 32] in order to seamlessly blend a foreground object with a background image, or from a statistical perspective by considering the different foreground and background color distributions [4, 26, 27, 35]. However, these approaches are limited by the information contained within a single image. To take advantage of the information in multiple images, prior work has used statistics from image sets in their approaches [21, 39].

Current methods are mostly based on neural networks, making use of an encoder-decoder CNN architecture based on U-Net [29], optionally enhanced by attention blocks [8], such as iSSAM [30]. While Cong *et al*. [5, 7] framed the problem as a domain gap task between foreground and background, Ling *et al*. [23] considered image harmonization as reducing the style discrepancy between inharmonious regions. Guo *et al*. [12–14] proposed a lighting-based approach that separately harmonizes an image after decomposing it into its reflection and illumination components. One of the methods employed a Transformer-based architecture, which improves the performance over architectures using CNNs. Based on the revision of architectures presented in [8, 33], Sofiiuk *et al*. [30] introduced semantic information by adding pre-trained HR-Net [36] features that are further processed by an encoder-decoder network. Unlike previous methods, Jiang *et al*. [17] proposed a framework that does not require any labeled training data at the expense of overall performance compared to supervised training. Hang *et al*. [15] introduced contrastive learning for image harmonization by combining two different contrastive regularization losses that can be used on top of any existing network. The above methods exclusively focus on low-resolution images, such as $256 \times 256$ pixels. While convolutional neural networks can be applied to larger images regardless of the training image size, the steep increase in computational cost renders existing approaches unsuitable for high resolution tasks.

To handle higher resolution images, most image harmonization approaches downscale the input image, *e.g*., to $256 \times 256$ pixels. Instead of predicting new pixel values, a different output is regressed by a network and is subsequently applied to the full resolution image. For example, Liang *et al*. [22] trained a network to produce a color curve based on a low-resolution image that is then applied to the full-resolution foreground image. Similarly, Ke *et al*. [18] proposed a model named *Harmonizer*, which uses an encoder network on a low-resolution image to regress coefficients controlling various filter operations, such as brightness and contrast. While these two light-weight approaches are easily adaptable to high resolutions, they are not able to account for local differences across the foreground regions compared to pixel-to-pixel transformations. Another recent approach combined pixel-to-pixel as well as RGB-to-RGB transformations using two separate branches [6].

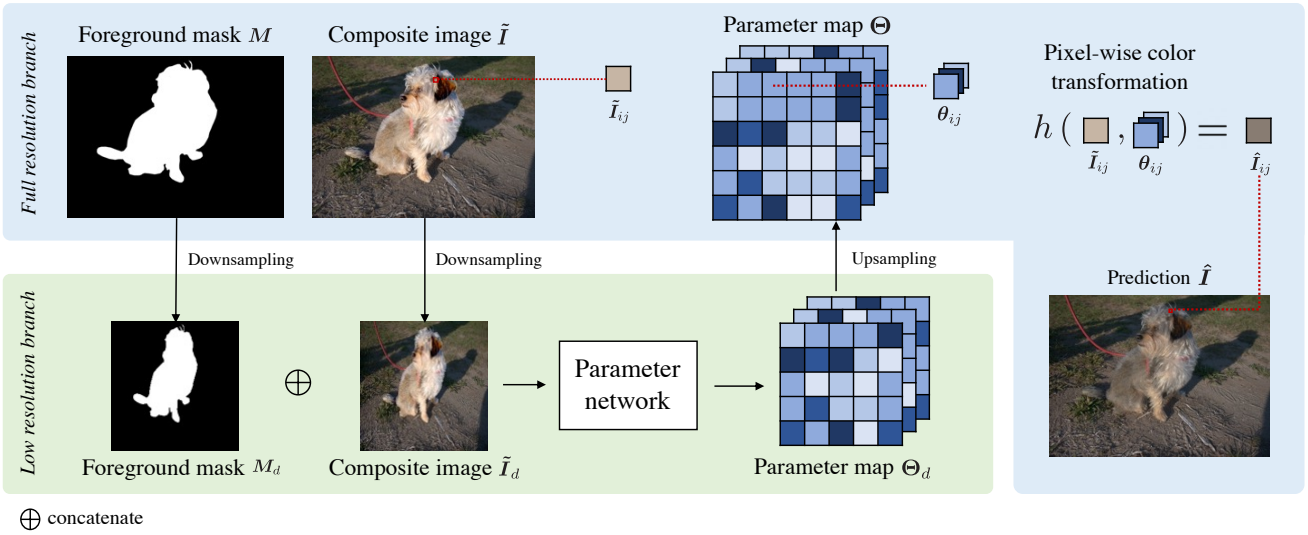


**Figure 2. Model overview.** *Our model is composed of two branches, a low-resolution (LR) and a full-resolution (FR) branch. An LR parameter map is obtained using the trained parameter network. The LR parameter map is upsampled to obtain the FR parameter map. To harmonize a full-resolution image, pixel values of the FR image foreground region $\tilde{\boldsymbol{I}}_{ij}$ are mapped using the pixel-wise color transformation $h(\cdot)$ taking the parameters of the FR parameter map. The pictures are taken from [7].*

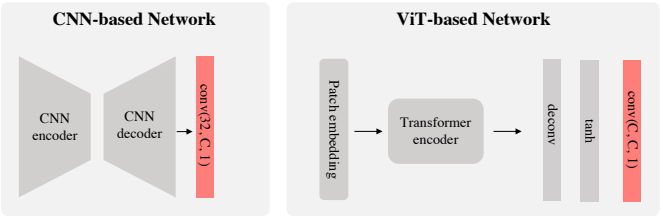


**Figure 3. Parameter network architectures.** *We propose two networks as backbones to predict the parameter map, a CNN (left) and a ViT (right). Instead of an image, we modify the architectures to output a parameter map of depth $C$. The added blocks are indicated in red.*

An encoder-decoder architecture processes a low resolution image, whereas the input mask and features from the encoder are input to a color mapping module [38] that predicts an RGB-to-RGB transformation. In the final step, the two branches are combined by a refinement module that is capable of producing a high-resolution image prediction. Xue *et al*. [34] proposed Deep Comprehensible Color Filter (DCCF), consisting of four neural color filters, the coefficients of which are predicted by a low-resolution branch for each pixel. Before applying the neural color filters to the full resolution composite image, the coefficient map is scaled to match the final image size.

**Image enhancement** aims to adjust an image to make it more suitable for future use, for instance by modifying the contrast or brightness of an image. Unlike image harmonization, image enhancement affects the entire image, but have followed comparable dual branch solutions to tackle high-resolution inputs. For example, Gharbi *et al*. [11] developed a deep bilateral learning approach leveraging local affine color transformations. Mohan *et al*. [24] introduced three different deep local parametric filters, which are combined to obtain the final enhanced image. Unlike our proposed PCT functions, the learned filters are not applied to each individual pixel.

## 3. PCT-Net

### 3.1. Problem Definition

Let us denote a composite RGB image as $\tilde{\boldsymbol{I}} \in \mathbb{R}^{H \times W \times 3}$ of width $W$ and height $H$, and its corresponding binary input mask as $\boldsymbol{M} \in \{0,1\}^{H \times W}$ which indicates the foreground region. Given $\tilde{\boldsymbol{I}}$ and $\boldsymbol{M}$ as input, the goal of image harmonization is to change the color of the foreground region to match the background image. If a ground truth image $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$ is available, harmonization can be defined as minimization of the error between the predicted image $\hat{\boldsymbol{I}} \in \mathbb{R}^{H \times W \times 3}$ and its ground truth.

### 3.2. Overview

An overview of the proposed model, PCT-Net, is shown in Fig. 2. PCT-Net consists of two different branches, *i.e*., a full resolution branch (FR) and a low resolution branch (LR). For the LR branch, we downscale the FR composite and the mask image to a lower resolution $(W_d, H_d)$, obtain-

ing $\tilde{\boldsymbol{I}}_d$ and $\boldsymbol{M}_d$. The low-resolution path is used for training a parameter network (see Subsection 3.3), which maps the LR input to a parameter map $\boldsymbol{\Theta}_d \in \mathbb{R}^{W_d \times H_d \times C}$, where $C$ is the number of parameters per pixel. The FR parameter map $\boldsymbol{\Theta} \in \mathbb{R}^{W \times H \times C}$ is obtained by upsampling $\boldsymbol{\Theta}_d$.

To modify the input image, we use a pixel-wise color transformation (PCT) function $h(\cdot)$ in the FR branch. For each pixel in the foreground region with coordinates $(i, j)$, the PCT function is applied using the pixel values of the composite image and the parameter map, *i.e.* $\hat{\boldsymbol{I}}_{i,j} = h\left(\tilde{\boldsymbol{I}}_{i,j}; \boldsymbol{\theta}_{i,j}\right)$. Different functions can be selected for the PCT, some of which are detailed in Subsection 3.4. During training of the parameter network in the LR branch, the loss is calculated in the FR branch based on the difference between the predicted image $\hat{\boldsymbol{I}}$ and the ground truth $\boldsymbol{I}$ (see Subsection 3.5). Unlike other approaches, PCT-Net enables full-resolution pixel-to-pixel image harmonization while retaining a light-weight architecture.

### 3.3. Parameter Network

The parameter network processes the LR images to output the parameter map $\boldsymbol{\Theta}_d$ that controls the pixel-wise color transformation in the FR branch. We evaluate two types of architectures for the parameter network, namely a CNN-based encoder-decoder network and a network based on a Visual Transformer (ViT) [9] (see Fig. 3).

As CNN-based network, we adopt iSSAM [30], which has been used as a backbone in recent work for image harmonization [15,34]. We replace the output layer with a $1 \times 1$ convolution layer to ensure the output provides $C$ parameters. For our Transformer-based network, we modify the Visual Transformer architecture [9] to act as an encoder and follow the model described in [12]. First, the input image is divided into patches which are subsequently projected to an embedding space. Then, positional encodings are added to the embedded patches and processed by the Transformer encoder consisting of multiple attention layers. In order to obtain the final feature map with the correct dimensions, the output of the patches from the Transformer encoder are reassembled and processed by a single deconvolution layer [37] and a $\tanh$ non-linearity. Analogous to the CNN-based network, a $1 \times 1$ convolution is applied as the final step to output the correct amount of parameters per pixel.

### 3.4. Pixel-wise Color Transformation

Here, we introduce different PCT functions, which are evaluated in Section 4. A straightforward function directly predicts new pixel values while disregarding the input pixel and interpreting the parameter output as the harmonized pixel as

$$h(\boldsymbol{p}; \boldsymbol{\theta}) = (\theta_1, \theta_2, \theta_3)^T, \tag{1}$$

where $\boldsymbol{p} = (p_1, p_2, p_3) \in \mathbb{R}^3$ represents a three-dimensional pixel value in the foreground region of the input image. This function essentially describes how previous methods directly predict harmonized pixel values.

Since color transformations can be expressed as an approximation using a linear transformation, we consider an affine transformation as a PCT function:

$$\begin{aligned} h(\boldsymbol{p}; \boldsymbol{\theta}) &= \boldsymbol{W}_{\boldsymbol{\theta}} \cdot \boldsymbol{p} + \boldsymbol{b}_{\boldsymbol{\theta}}, \\ \boldsymbol{W}_{\boldsymbol{\theta}} &= \begin{pmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & \theta_9 \end{pmatrix}, \quad \boldsymbol{b}_{\boldsymbol{\theta}} = \begin{pmatrix} \theta_{10} \\ \theta_{11} \\ \theta_{12} \end{pmatrix}. \end{aligned} \tag{2}$$

When the parameter map is upsampled from the LR to the FR branch to match the original image resolution, the parameters are interpolated instead of the pixels. Furthermore, we can apply an additional constraint to control the properties of our transformation, for example, by making the matrix $\boldsymbol{W}_{\boldsymbol{\theta}}$ symmetric.

We further investigate a polynomial PCT function inspired by the color correction method proposed in [10]. First the three-dimensional pixel $\boldsymbol{p}$ is transformed into a higher-dimensional representation $\boldsymbol{p}_{pol} \in \mathbb{R}^9$ as follows:

$$\boldsymbol{p}_{\text{pol}} = (p_1, p_2, p_3,\ p_1 p_2,\ p_1 p_3,\ p_2 p_3,\ p_1^2,\ p_2^2,\ p_3^2)^T. \tag{3}$$

Then we apply a linear transformation using a matrix $\boldsymbol{X}_{\boldsymbol{\theta}} \in \mathbb{R}^{3 \times 9}$ that projects the higher-dimensional representation back to the pixel space as $h(\boldsymbol{p}; \boldsymbol{\theta}) = \boldsymbol{X}_{\boldsymbol{\theta}} \boldsymbol{p}_{pol}$. While this allows for more flexibility and control for the parameter network, the large number of parameters per pixel leads to higher computational complexity.

### 3.5. Loss Function

To evaluate the difference between composite image $\hat{\boldsymbol{I}}$ and the ground truth $\boldsymbol{I}$ within the foreground mask $\boldsymbol{M}$, we adopt the foreground-normalized MSE loss [30] which considers the size of the foreground area as it discourages focusing disproportionately on images with a larger foreground:

$$\mathcal{L}_{\text{fMSE}} = \frac{\sum_{i,j} (\hat{\boldsymbol{I}}_{i,j} - \boldsymbol{I}_{i,j})^2}{\max\left(A_{\min}, \sum_{i,j} \boldsymbol{M}_{i,j}\right)}, \tag{4}$$

where $A_{\min}$ is a constant value, which we set as 1000 for our experiments to stabilize training. In addition, we leverage two regularization losses to support the learning process. The first loss is a self-style contrastive regularization $\mathcal{L}_{\text{CR}}$, which is proposed in [15] for Image Harmonization, to allow the network to learn not only from positive pairs between an composite image and the ground truth, but also from negative ones between various composite images. To that end, we create additional composite images as

described in [15] by applying color transformations to the foreground region of the ground truth:

$$\mathcal{L}_{\mathrm{CR}} = \frac{D\left(\boldsymbol{f}, \boldsymbol{f}^{+}\right)}{D\left(\boldsymbol{f}, \boldsymbol{f}^{+}\right) + \sum_{k=1}^{K} D\left(\boldsymbol{f}, \boldsymbol{f}^{-}\right)} + \frac{D\left(\boldsymbol{c}, \boldsymbol{c}^{+}\right)}{D\left(\boldsymbol{c}, \boldsymbol{c}^{+}\right) + \sum_{k=1}^{K} D\left(\boldsymbol{c}, \boldsymbol{c}^{-}\right)}. \tag{5}$$

In Eq. (5), $\boldsymbol{f}$, $\boldsymbol{f}^{+}$ and $\boldsymbol{f}^{-}$ are the feature vectors of the foreground region of the predicted image, the ground truth and the composite images, respectively. The feature vector extractor is a pre-trained VGG16 network in the same manner as [15]. The parameter $\boldsymbol{c}$ represents the foreground-background style consistency calculated by $\boldsymbol{c} = \mathrm{Gram}\left(\boldsymbol{f}, \boldsymbol{b}^{+}\right)$, where $\boldsymbol{b}^{+}$ is the feature vector of the background region of the ground truth. $\mathrm{Gram}\left(\cdot\right)$ denotes the Gram matrix. The values of $\boldsymbol{c}^{+}$ and $\boldsymbol{c}^{-}$ are calculated in the same way as $\boldsymbol{c}$. The number $K$ is the total number of composite images. Function $D\left(\cdot\right)$ is the $\mathcal{L}_1$ distance function. The details of this self-style contrastive loss are found in [15].

Additionally, we investigate a smoothing regularization, which aims to further constrain the solution space during training and encourages neighboring parameters to be closer to each other. The regularization function is as follows:

$$\mathcal{L}_{\mathrm{smooth}} = \frac{1}{C} \sum_{k} ||\nabla \boldsymbol{\Theta}_k||_2, \tag{6}$$

where $||\nabla \boldsymbol{\Theta}_k||_2$ denotes the magnitude of the spatial gradient for the $k$-th parameter. In this case, the regularization function imposes restrictions directly on the parameter space instead of the images and penalizes large variations between neighboring parameter values.

# 4. Experiments

## 4.1. Dataset

Following prior work on image harmonization, we use the iHarmony4 [7] dataset for experiments, which consists of four different subsets, namely HAdobe5k, HCOCO, HDay2night and HFlickr. The resolution of these subsets ranges from $312 \times 230$ to $6048 \times 4032$ pixels. Only the HAdobe5k subset is composed of images with width or height larger than $1024$ pixels. Additionally, the number of images in the subsets varies greatly. Overall, the subsets HAdobe5k, HCOCO, HDay2night and HFlickr contain 2160, 4283, 133 and 828 test images, respectively. The composite images in iHarmony4 were created synthetically by choosing a foreground region and modifying the color of that region.

## 4.2. Implementation Details

We conduct extensive experiments of our proposed model using two different backbone modules, that is, a CNN-based backbone proposed in [30] as iSSAM, and a Vision Transformer (ViT)-based backbone introduced in [12]. To train our CNN-based model, we follow the procedures in [34]. The model is trained for 180 epochs with a learning rate of $10^{-3}$. At the 160th epoch and the 175th epoch the rate is decreased by a factor of 10 respectively. In the case of our ViT-based model, we slightly change the number of epochs compared to [12] to 100, instead of 60 epochs. The model is accordingly trained with a learning rate of $10^{-4}$, while only switching from a constant learning rate to a linear learning rate after 50 epochs.

Our loss function consists of the fMSE loss as described in Subsection 3.5. For regularization, we use a smoothing term for our CNN-based approach and a contrastive term for our ViT-based model. The different choices for the regularization terms are motivated by our ablation study in Subsection 4.4. We re-implement contrastive regularization by following the original paper [15]. Compared to the fMSE loss, we weight our contrastive loss by $0.01$ in accordance with [15]. For the smoothness regularization, we empirically set the weighting coefficient to $0.1$.

While training our models, we further augment the full-resolution composite images by cropping them according to a random bounding box, the size of which lies between 0.7 and 1 of the original height and width. Moreover, we randomly flip the input images horizontally. Due to memory limitations, we resize all images so that the largest side does not exceed 2048 pixels. Apart from that, we do not resize the full-resolution images before using them in the FR branch. Unlike Xue *et al*., we therefore train our images in the original size with different aspect ratios. Both models are trained using an Adam [19] optimizer and a batch size of 4. Our models are based on the PyTorch framework and trained on NVIDIA® A100 GPUs.

## 4.3. Evaluation

We evaluate the performance of our approach on the test dataset of iHarmony4 and compare the results of other state-of-the-art methods. To show the effectiveness of our method on both, a CNN architecture and a Transformer-based architecture, we consider two different backbones.

As quantitative performance metrics for image harmonization, we calculate the mean squared error (MSE), peak signal-to-noise ratio (PSNR) and the foreground mean squared error (fMSE) for each image following other image harmonization papers and take the average across each subset. While the MSE has been used as an important evaluation metric, we argue that due to varying foreground sizes within the dataset, the MSE is skewed towards images with large foreground regions. Therefore, the fMSE offers a more balanced understanding of overall quality. Further considerations in terms of evaluation metrics are detailed in the supplementary material.

Unlike most existing image harmonization methods, we are mainly interested in the performance of image harmonization on high resolution images, which can only be found in the HAdobe5k subset. While most of the prior work only evaluates their method on the standard low resolution of $256 \times 256$ pixels, we provide results on the original full image resolution. For comparison, we adopt Ke *et al*. [18] and Xue *et al*. [34], which, to the best of our knowledge, are the only methods that also evaluate on full-resolution images. While the method presented by Liang *et al*. [22] is able to handle full-resolution images, results for full-resolution images were not provided.

Xue *et al*. [34] did not report fMSE scores. Using the code provided by the authors, we thus calculated the fMSE scores for comparability, which is represented with an asterisk in Table 1. The comparison results with other prior work on the $256 \times 256$ pixels low-resolution images are summarized in the supplementary material.

Table 1 shows the results of our proposed networks and the two existing methods on each subset as well the entire iHarmony4 test set. Both of our proposed methods clearly outperform other approaches across the entire dataset. Out of both our backbones, the ViT-based one demonstrates better results than the CNN-based model. Compared to Xue *et al*., which achieved the best results out of both previous methods, our ViT-based model improves fMSE and MSE by more than 20% on the entire test set, while increasing the PSNR by more than 1.4dB. Nonetheless, our models show worse scores on the HDay2night subset compared to Ke *et al*., which can be attributed to the fact that the amount of available data is low and the images mainly show webcam footage of landscapes, which is very different from the rest of the dataset.

In Figure 4, we illustrate the improved performance of our approach on one example image. Compared to the two other methods, both our networks are able to align the color of the napkin fairly close to the ground truth image. In our example image, there are two reference objects that could be used to infer a suitable color. While the results for Ke *et al*. and Xue *et al*. do not indicate that the network used the other napkins as a reference, our results suggest that our method might posses such capabilities.

Figure 5 shows another example, where our networks fail to improve the composite image. We can see that the image itself presents a quite challenging setting. In order to correctly harmonize the window, a deeper understanding regarding the behavior of light and how it interacts with different materials such as glass is necessary. This particular example seems to expose a lack of such an understanding across all the approaches.

In addition to considering the average error, we also investigate how many images experience a decline in terms of fMSE, MSE and PSNR after applying an image harmonization model. The results are shown in Table 2. While Xue *et al*. only demonstrate slightly lower success rates compared to our approach improving composite images, the model proposed by Ke *et al*. and our CNN-based model are not able to decrease the MSE or PSNR in more than one fourth of all cases. This indicates that both models are capable of improving some composite images quite well, yet lack the capability to apply these improvements to certain images in the test set.

### 4.4. Ablation Studies

In order to verify the effectiveness of our design choices, we conduct several ablation studies in this subsection. The base model is trained using only the $\mathcal{L}_{\text{fMSE}}$ loss with an affine transformation for the PCT function.

First, we verify the effectiveness of our approach based on parameter map interpolation over upsampling from a predicted low-resolution image. For the straightforward upsampling, we first predict a low-resolution image using the LR branch, and use bilinear interpolation to obtain the full-resolution image. To measure the performance difference of these two approaches with regards to image resolution, we use not only the original full resolution iHarmony4 dataset, but also the one downsampled into $256 \times 256$ pixels, as has been the standard in prior harmonization work [18, 34]. The results are summarized in Table 3. The simple upsampling approach is comparable with ours on low resolution images, however, it performs worse on full resolution images. In conclusion, we confirm that performance degradation on full resolution images is significantly reduced when using parameter map interpolation.

We investigate two different regularization techniques introduced in Subsection 3.5. Table 4 shows the results of experiments conducted with different loss functions. For the CNN-based PCT-Net, smoothing the parameter map improves performance, while it does not improve the performance for the ViT-based approach. This behavior might be caused by decoder layers in the Transformer model, which we investigate in the supplementary material. Instead, the Transformer-based model benefits from adding a contrastive regularization.

Lastly, we also investigate the influence of our PCT function on the overall performance. We consider three different PCT functions listed in Table 5, where *affine symmetric* describes an affine transformation with a symmetric matrix. Whereas the polynomial PCT function does not provide better results compared to the simpler affine PCT function, we can see slight improvements if the affine transformation matrix is restricted to be symmetric. We provide further analysis of different PCT functions in the supplementary material.

**Table 1. Quantitative performance on the full-resolution iHarmony4 dataset.** *Ke* et al.*'s and Xue* et al.*'s results are taken from the original papers [18,34]. Since [34] does not provide fMSE values, we evaluate DCCF by running the provided code on the images, denoted as Xue* et al. *[34]*. Our models achieve lower errors on all datasets, except for the HDay2night dataset. The best results are marked in bold, the second best are underlined.*

| Method | HAdobe5k subset | | | HCOCO subset | | | HDay2night subset | | | HFlickr subset | | | All | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | fMSE↓ | MSE↓ | PSNR↑ | fMSE↓ | MSE↓ | PSNR↑ | fMSE↓ | MSE↓ | PSNR↑ | fMSE↓ | MSE↓ | PSNR↑ | fMSE↓ | MSE↓ | PSNR↑ |
| Composite images | 2148.42 | 54.46 | 28.16 | 1079.71 | 73.03 | 33.54 | 1502.99 | 113.07 | 33.96 | 1646.29 | 270.99 | 28.23 | 1462.45 | 177.99 | 31.24 |
| Ke *et al*. [18] | 196.12 | 24.37 | 37.80 | 374.96 | 20.93 | 37.69 | **640.74** | **37.28** | 37.15 | 479.26 | 69.19 | 33.37 | 339.23 | 27.62 | 37.23 |
| Xue *et al*. [34] | N/A | 23.34 | 37.75 | N/A | 17.07 | 38.66 | N/A | 55.76 | **37.40** | N/A | 64.77 | 33.60 | N/A | 24.65 | 37.87 |
| Xue *et al*. [34]* | 196.19 | 23.98 | 37.67 | 317.80 | 17.37 | 38.37 | 716.47 | 55.09 | 37.35 | 437.82 | 65.16 | 33.46 | 302.89 | 25.34 | 37.60 |
| **Ours (CNN)** | 168.56 | 21.14 | 39.10 | 297.34 | 16.93 | 38.81 | 740.42 | 50.53 | 36.84 | 431.82 | 64.19 | 33.76 | 282.77 | 24.05 | 38.29 |
| **Ours (ViT)** | **149.39** | **19.35** | **39.97** | **245.67** | **12.45** | **39.85** | 700.65 | 46.47 | 37.25 | **357.53** | **45.79** | **34.87** | **238.27** | **18.80** | **39.28** |

**(a)** *Input image* **(b)** *Mask image* **(c)** *Ke* et al. *[18]* **(d)** *Xue* et al. *[34]* **(e)** *Ours (CNN)* **(f)** *Ours (ViT)* **(g)** *Ground truth*

**Figure 4. Examples of successful harmonization.** *The top and bottom images are original and zoomed-in images on the mask region, respectively. Given the input composite image shown in (a) and the foreground mask image shown in (b), Ke* et al.*'s, Xue* et al.*'s and our methods generate the harmonized images shown in (d)–(f). Our results, especially Ours (ViT), are closer to the ground truth image shown in (g). Original images are taken from [7].*

**(a)** *Input image* **(b)** *Mask image* **(c)** *Ke* et al. *[18]* **(d)** *Xue* et al. *[34]* **(e)** *Ours (CNN)* **(f)** *Ours (ViT)* **(g)** *Ground truth*

**Figure 5. Examples of failed harmonization.** *The image layout follows that of Fig. 4. It can be seen that all methods fail to harmonize the composite image well. Original images are taken from [7].*

**Table 2. Percentage of images that did not improve upon applying image harmonization**. *A lower percentage is better. The best results are marked in bold, the second best are underlined. Like Table 1, Xue* et al.*'s results are calculated by running the code provided by the authors.*

| Method | fMSE | MSE | PSNR |
|---|---|---|---|
| Ke *et al*. [18] | 14.33% | 26.74% | 27.70% |
| Xue *et al*. [34]* | <u>6.24%</u> | <u>6.14%</u> | <u>6.24%</u> |
| **Ours (CNN)** | 11.52% | 26.18% | 26.18% |
| **Ours (ViT)** | **4.89%** | **5.73%** | **5.73%** |

**Table 3. Performance comparison between ours and straightforward upsampling from low-resolution prediction.** *We evaluate the performance on the iHarmony4 test set using full resolution and images downsampled to* $256 \times 256$ *pixels. Even on full-resolution images, our methods are able to achieve performance comparable to the downsampled case by interpolating the parameter map instead of the images.*

| Method | Downsampled images | | | Full-resolution images | | |
|---|---|---|---|---|---|---|
| | fMSE↓ | MSE↓ | PSNR↑ | fMSE↓ | MSE↓ | PSNR↑ |
| Upsampling (CNN) | **264.69** | 24.44 | 38.19 | 557.50 | 44.03 | 35.19 |
| **Ours (CNN)** | 268.22 | **23.94** | **38.63** | **296.84** | **25.32** | **38.05** |
| Upsampling (ViT) | **219.53** | **18.21** | 39.41 | 505.92 | 35.64 | 37.60 |
| **Ours (ViT)** | 228.00 | 19.34 | **39.52** | **250.30** | **20.03** | **38.98** |

**Table 4. Effect of loss functions on performance.** *We evaluate different loss functions on the iHarmony4 test set. Adding a regularization function improves performance of both our networks. However, the CNN-based architecture only benefits from adding a smoothness regularization, while the Transformer-based approach performs best with a contrastive regularization.*

| Backbone | Loss | fMSE↓ | MSE↓ | PSNR↑ |
|---|---|---|---|---|
| CNN | $\mathcal{L}_{\text{fMSE}}$ | 296.84 | 25.32 | 38.05 |
| | $\mathcal{L}_{\text{fMSE}}+\mathcal{L}_{\text{smooth}}$ | **282.77** | **24.05** | **38.29** |
| | $\mathcal{L}_{\text{fMSE}}+\mathcal{L}_{\text{CR}}$ | 301.03 | 26.07 | 38.00 |
| ViT | $\mathcal{L}_{\text{fMSE}}$ | 250.30 | 20.03 | 38.98 |
| | $\mathcal{L}_{\text{fMSE}}+\mathcal{L}_{\text{smooth}}$ | 259.97 | 20.62 | 38.88 |
| | $\mathcal{L}_{\text{fMSE}}+\mathcal{L}_{\text{CR}}$ | **238.27** | **18.80** | **39.28** |

### 4.5. User Study

For a qualitative evaluation, we conducted a user study where twenty people were asked to select the better harmonized image given two different choices. For the evaluation, we prepared 26 high-resolution composite images using foreground objects from the BIG dataset [3] and the RealHM dataset [17] and adding them to suitable background images from [1]. Given the composite images, we obtained harmonized images with three methods, that is, our best model with the ViT backbone, Xue *et al*.'s [34] and Ke *et al*.'s [18] method. As in prior work, we adopt the Bradley-Terry [2] model, where 78 pairwise comparisons among the three methods are evaluated. Following [20], we average the B-T score across all images. The results are shown in Table 6, showing that the quantitative improvement of our method also translates to improvements in perceptual quality. The images used in this study are available in the supplementary material.

**Table 5. Effect of PCT function on performance.** *We investigate the impact of different PCT functions on performance by evaluating on the iHarmony4 test set. Affine PCT functions outperform polynomial PCT functions by a clear margin.*

| Backbone | PCT | fMSE↓ | MSE↓ | PSNR↑ |
|---|---|---|---|---|
| CNN | affine | 296.84 | 25.32 | 38.05 |
| | affine symmetric | **290.57** | **25.03** | **38.14** |
| | polynomial | 333.09 | 29.51 | 37.44 |
| ViT | affine | **250.30** | 20.03 | 38.98 |
| | affine symmetric | 250.79 | **19.94** | **39.03** |
| | polynomial | 267.05 | 20.95 | 38.67 |

**Table 6. User study results.** *The B-T score is calculated according to [20]. A higher score indicates higher preference. The output of our model (ViT) results in higher preference compared to two recent state-of-the-art methods. The study was conducted with 20 people, each person rated 26 composite images using pairwise comparisons.*

| Ke *et al*. [18] | Xue *et al*. [34] | **Ours (ViT)** |
|---|---|---|
| 0.98 | 0.93 | **1.09** |

## 5. Conclusion

In this paper, we proposed a light-weight and efficient image harmonization method that leverages the fact that interpolating within a function parameter space, rather than the image space introduces fewer errors. We investigated two different backbones: a CNN-based model and a ViT-based model. For both backbones, we observe significant improvements using pixel-wise color transformations. In experiments, we show that the approach is effective for harmonizing full-resolution images and achieves state-of-the-art results in terms of fMSE, MSE, and PSNR on the iHarmony4 dataset. Furthermore, in a user study involving 20 participants and 26 images with three different harmonization methods, we show that our approach also improves the perceptual quality. Overall, the simplicity of our approach allows for future investigation of different backbone models and color transformation functions.

## References


[1] unsplash.com. https://unsplash.com. 8

[2] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 8

[3] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 8

[4] Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. Color harmonization. In *ACM SIGGRAPH 2006 Papers*, pages 624–630. 2006. 2

[5] Wenyan Cong, Li Niu, Jianfu Zhang, Jing Liang, and Liqing Zhang. Bargainnet: Background-guided domain translation for image harmonization. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 1, 2

[6] Wenyan Cong, Xinhao Tao, Li Niu, Jing Liang, Xuesong Gao, Qihao Sun, and Liqing Zhang. High-resolution image harmonization via collaborative dual transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18470–18479, June 2022. 1, 2

[7] Wenyan Cong, Jianfu Zhang, Li Niu, Liu Liu, Zhixin Ling, Weiyuan Li, and Liqing Zhang. Dovenet: Deep image harmonization via domain verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8394–8403, June 2020. 2, 3, 5, 7

[8] Xiaodong Cun and Chi-Man Pun. Improving the harmony of the composite image by spatial-separated attention module. *IEEE Transactions on Image Processing*, 29:4759–4771, 2020. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[10] Graham D Finlayson, Michal Mackiewicz, and Anya Hurlbert. Color correction using root-polynomial regression. *IEEE Transactions on Image Processing*, 24(5):1460–1470, 2015. 4

[11] Michaël Gharbi, Jiawen Chen, Jonathan T. Barron, Samuel W. Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Trans. Graph.*, 36(4), jul 2017. 3

[12] Zonghui Guo, Zhaorui Gu, Bing Zheng, Junyu Dong, and Haiyong Zheng. Transformer for image harmonization and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–19, 2022. 1, 2, 4, 5

[13] Zonghui Guo, Dongsheng Guo, Haiyong Zheng, Zhaorui Gu, Bing Zheng, and Junyu Dong. Image harmonization with transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14870–14879, October 2021. 1, 2

[14] Zonghui Guo, Haiyong Zheng, Yufeng Jiang, Zhaorui Gu, and Bing Zheng. Intrinsic image harmonization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16367–16376, June 2021. 1, 2

[15] Yucheng Hang, Bin Xia, Wenming Yang, and Qingmin Liao. Scs-co: Self-consistent style contrastive learning for image harmonization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19710–19719, June 2022. 1, 2, 4, 5

[16] Jiaya Jia, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Drag-and-drop pasting. *ACM Transactions on graphics (TOG)*, 25(3):631–637, 2006. 2

[17] Yifan Jiang, He Zhang, Jianming Zhang, Yilin Wang, Zhe Lin, Kalyan Sunkavalli, Simon Chen, Sohrab Amirghodsi, Sarah Kong, and Zhangyang Wang. Ssh: A self-supervised framework for image harmonization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4832–4841, October 2021. 1, 2, 8

[18] Zhanghan Ke, Chunyi Sun, Lei Zhu, Ke Xu, and Rynson WH Lau. Harmonizer: Learning to perform white-box image and video harmonization. In *European Conference on Computer Vision*, pages 690–706. Springer, 2022. 1, 2, 6, 7, 8

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 8

[21] Jean-Francois Lalonde and Alexei A Efros. Using color compatibility for assessing image realism. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2

[22] Jingtang Liang, Xiaodong Cun, and Chi-Man Pun. Spatial-separated curve rendering network for efficient and high-resolution image harmonization. In *European Conference on Computer Vision*, pages 334–349. Springer, 2022. 1, 2, 6

[23] Jun Ling, Han Xue, Li Song, Rong Xie, and Xiao Gu. Region-aware adaptive instance normalization for image harmonization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9361–9370, June 2021. 1, 2

[24] Sean Moran, Pierre Marza, Steven McDonagh, Sarah Parisot, and Gregory Slabaugh. Deeplpf: Deep local parametric filters for image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12826–12835, 2020. 3

[25] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, pages 313–318. 2003. 2

[26] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1434–1439. IEEE, 2005. 2

[27] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 2

[28] Xuqian Ren and Yifan Liu. Semantic-guided multi-mask image harmonization. In *European Conference on Computer Vision*, pages 564–579. Springer, 2022. 1

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 2

[30] Konstantin Sofiiuk, Polina Popenova, and Anton Konushin. Foreground-aware semantic representations for image harmonization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1620–1629, January 2021. 1, 2, 4, 5

[31] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image harmonization. *ACM Transactions on Graphics (TOG)*, 29(4):1–10, 2010. 2

[32] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. In *European Conference on Computer Vision*, pages 31–44. Springer, 2010. 2

[33] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3789–3797, July 2017. 2

[34] Ben Xue, Shenghui Ran, Quan Chen, Rongfei Jia, Binqiang Zhao, and Xing Tang. DCCF: Deep comprehensible color filter learning framework for high-resolution image harmonization. In *European Conference on Computer Vision*, pages 300–316. Springer, 2022. 1, 3, 4, 5, 6, 7, 8

[35] Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. Understanding and improving the realism of image composites. *ACM Transactions on graphics (TOG)*, 31(4):1–10, 2012. 2

[36] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 173–190. Springer International Publishing, 2020. 2

[37] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, 2010. 4

[38] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2058–2073, 2022. 3

[39] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Learning a discriminative model for the perception of realism in composite images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3943–3951, 2015. 2