https://doi.org/10.2352/issn.2169-2629.2021.29.7

GamutNet: Restoring Wide-Gamut Colors for Camera-Captured Images

Hoang Le[1*], Taehong Jeong[2*], Abdelrahman Abdelhamed[3], Hyun Joon Shin[4], and Michael S. Brown[1]
[1] York University, Toronto, Canada [2] MAXST, Seoul, Republic of Korea [3] Samsung AI Center, Toronto, Canada
[4] Ajou University, Suwon, Republic of Korea
[*] These authors contributed equally.

Abstract

Most cameras still encode images in the small-gamut sRGB color space. The reliance on sRGB is disappointing as modern display hardware and image-editing software are capable of using wider-gamut color spaces. Converting a small-gamut image to a wider-gamut is a challenging problem. Many devices and software use colorimetric strategies that map colors from the small gamut to their equivalent colors in the wider gamut. This colorimetric approach avoids visual changes in the image but leaves much of the target wide-gamut space unused. Non-colorimetric approaches stretch or expand the small-gamut colors to enhance image colors while risking color distortions. We take a unique approach to gamut expansion by treating it as a restoration *problem. A key insight used in our approach is that cameras internally encode images in a wide-gamut color space (i.e., ProPhoto) before compressing and clipping the colors to sRGB's smaller gamut. Based on this insight, we use a software-based camera ISP to generate a dataset of 5,000 image pairs of images encoded in both sRGB and ProPhoto. This dataset enables us to train a neural network to perform wide-gamut color restoration. Our deep-learning strategy achieves significant improvements over existing solutions and produces color-rich images with few to no visual artifacts.*

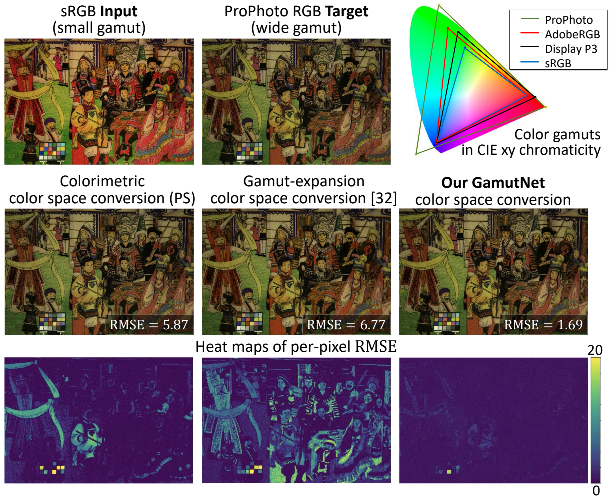

Figure 1. *(top) An input image encoded in its small-gamut sRGB color space and its ground-truth wide-gamut ProPhoto. Gamuts of common color spaces used on cameras (sRGB, Adobe RGB, Display P3, and ProPhoto) are shown. (middle) Conversion of the sRGB image to ProPhoto using a standard "colorimetric" conversion from Photoshop [10], a state-of-the-art gamut expansion method [5], and our GamutNet result. (last) Heat maps of per-pixel errors in terms of root-mean-squared-error (RMSE).*

Introduction

Most digital cameras still encode their captured images in the standard RGB (sRGB) color space [1]. Since sRGB's introduction, several wider gamut color spaces, such as Adobe RGB [2], Display P3[1], and ProPhoto RGB [4], have been proposed for use with improved display technology and image-editing software. For a simple comparison, sRGB is capable of encoding only $\sim$35% of all visible colors, the medium-gamut Adobe RGB and Display P3 color spaces encode $\sim$50% of all visible colors, and the wide-gamut ProPhoto color space encodes $\sim$90% of all visible colors.

A gamut conversion is required to map between color spaces. Converting from wide-gamut to small-gamut color spaces requires deciding how to contract and clip color values to lie within a smaller gamut boundary. Conversely, the restoration of a wide-gamut color space from a small-gamut one is more challenging as the correct target colors in the wider gamut are unknown (see Fig. 1). When converting an sRGB image to a wider-gamut color space, most software and devices employ a *colorimetric* strategy that strives for accurate color reproduction in the target color space. This conservative approach avoids color distortion, but leaves large regions in the wider-gamut unused. Non-colorimetric gamut-expansion methods (also referred to as saturation or perceptual intent methods), such as [5, 6, 7, 8], are designed to stretch the input gamut colors to fit the target gamut with the goal of enhancing the images and making them appear more vivid. However, in terms of restoration, their results may introduce unwanted color distortions.

We leverage an insight proposed in [9] which observed that a camera's image signal processor (ISP) units produce an internal version of a captured image in the wide-gamut ProPhoto RGB color space. This in-camera conversion of the raw-RGB sensor image to the ProPhoto RGB color space allows cameras to perform photo enhancement in a color-rich representation. The conversion to a gamut-limited output color space is applied as the last step of the camera pipeline. Unfortunately, cameras do not provide easy access to this wide-gamut image state. However, it is possible to use a software ISP to mimic the camera's hardware. This allows access to an image in both its wide-gamut and small-gamut color encoding. This access to paired small-gamut and wide-gamut images enables the color space conversion problem to be cast as a restoration problem where known ground truth (i.e., wide-gamut version) is available to learn a recovery model.

Contribution We describe an approach to leverage the in-camera processing pipeline to produce a dataset of 5,000 image pairs encoded in both the small-gamut sRGB and wide-gamut ProPhoto RGB color spaces. Using this dataset, we train a deep neural network, termed GamutNet, to restore sRGB image colors back to their wide-gamut ProPhoto RGB representation. We show that this deep-learning approach to color restoration can reduce errors by almost 50% over existing methods. Our dataset, code, and trained model will be made publicly available.

[1] Display P3 was introduced by Apple and is based on the DCI-P3 color space [3].

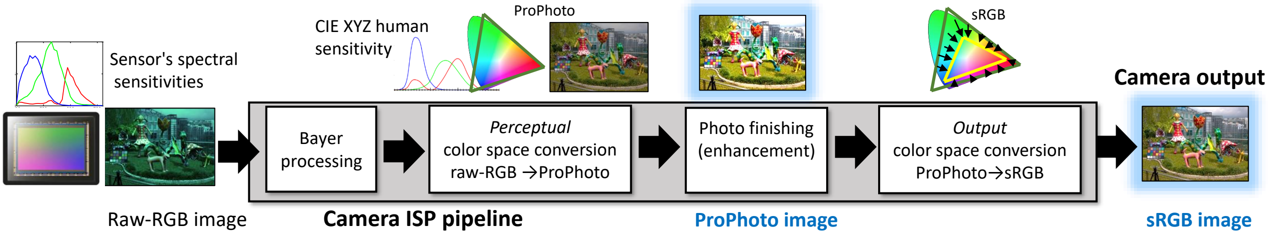

Figure 2. *This figure illustrates a typical imaging pipeline implemented by a camera's image signal processor (ISP) hardware. An essential step in the pipeline is converting the sensor-specific raw-RGB colors to a perceptual color space based on CIE XYZ (i.e., ProPhoto). The ISP then enhances the wide-gamut ProPhoto-encoded image. The enhanced ProPhoto image is finally converted to its output color space. While some cameras allow an option to save the image in a medium-gamut color space, such as Adobe RGB or Display P3, most cameras default to sRGB. By emulating this camera pipeline, we can obtain training data from pairs of the same images encoded in small-gamut sRGB and wide-gamut ProPhoto (highlighted in blue).*

Preliminaries and Related Work

Addressing the gamut mismatch between different color spaces and devices is a well-studied topic (e.g., [11, 12, 13, 14, 15, 16]) Methods to convert color values from one color space to another can be categorized into two classes: (i) colorimetric reproduction and (ii) gamut matching. Colorimetric reproduction includes *absolute* and *media-relative colorimetric rendering intents* [17] that aim to reproduce the exact visual stimuli of the image in the target color space. Gamut matching, on the other hand, focuses on fully utilizing the color gamut of the target space. Gamut matching methods are often referred to as *perceptual* or *saturation-rendering intents*. These include gamut expansion methods (e.g., recent methods [7, 5]) that expand colors to fit the target gamut based on local features (e.g., local contrast) in the image.

Different from the discussed categories, the goal of our method is to *restore* the small-gamut encoding of an image to its wide-gamut encoding while maintaining colorimetric accuracy. In particular, we focus our attention on the saturated points found in the small-gamut image. These saturated points represent are potentially wide-gamut colors that were clipped to the small-gamut boundary when converting from their original wide-gamut encoding on the camera. Recovering these clipped color values is mathematically an ill-posed problem.

As described earlier, the key insight leveraged in this work is that the camera imaging pipeline internally encodes captured images to the wide-gamut ProPhoto color space as part of the internal processing on the hardware image signal processor (ISP). Specifically, one of the main tasks of an ISP is to convert the sensor-specific raw-RGB image, defined by the sensor's color filter array's spectral sensitivities, to a device-independent color space based on CIE XYZ [18]. In fact, the ProPhoto color space (initially called the Reference Output Medium Metric [ROMM] color space [4]) was designed explicitly for this purpose. The ProPhoto encoded image is subsequently enhanced by the ISP and finally saved to sRGB, as shown in Fig. 2. Cameras do not allow access to this internal image ProPhoto encoding; however, it is possible to use a software-ISP to mimic this functionality. As a result, we can produce a large dataset of ProPhoto encoded and sRGB encoded image pairs.

Recent work [9] introduced a similar idea that embedded a small number of samples from the ProPhoto image into the saved sRGB image as a comment. This metadata was used to estimate a mapping function that enables the mapping from the sRGB colors to their corresponding ProPhoto values. We aim to solve this problem without specialized metadata using a neural network framework.

Deep Gamut Restoration

We begin by first providing an overview of our method, followed by a description of generating images for our dataset. Finally, we provide details to the GamutNet architecture and its training process.

Method Overview

Given an image encoded in the sRGB color space, denoted as $\mathbf{x}_{\text{sRGB}} \in \mathbb{R}^{M\times 3}$, where M is the number of pixels, we want to map it into a wide-gamut image, denoted as $\mathbf{y}_{\text{ProPhoto}} \in \mathbb{R}^{M\times 3}$, in the ProPhoto color space. Note that our goal is to produce an image that is as colorimetrically accurate as possible in the wide-gamut color space. The colors that are most problematic are those that were out-of-gamut in the in-camera conversion from ProPhoto to sRGB. The color values of these pixels had to be *clipped* to the sRGB gamut boundary. As a result, we focus our network's capacity on these pixels and not the pixels with in-gamut colors.

With this in mind, we apply a pre-defined 3×3 linear transform to convert $\mathbf{x}_{\text{sRGB}}$ to a clipped version of ProPhoto. This is written as follows:

$$\tilde{\mathbf{x}}_{\text{ProPhoto}} = C\, \tilde{\mathbf{x}}_{\text{sRGB}}\ , \qquad (1)$$

where $C \in \mathbb{R}^{3\times 3}$ using the chromatic adaptation transform CIECAT02 [19] and $\tilde{\mathbf{x}}_{\text{sRGB}}$ represents the linearized sRGB values after a de-gamma operation is performed on $\mathbf{x}_{\text{sRGB}}$. The matrix C is applied to each RGB value in $\tilde{\mathbf{x}}_{\text{sRGB}}$.

This linear mapping converts all original in-gamut color values in a colorimetrically accurate manner. However, the colors of pixels initially clipped to the sRGB gamut, are constrained at the boundary of the sRGB gamut in the ProPhoto space (see visualization in Fig. 3). We referred to this transformed image as $\tilde{\mathbf{x}}_{\text{ProPhoto}}$. Our goal is to process these out-of-gamut pixels to restore them to their original wide gamut value.

Out-of-gamut pixels Since the input sRGB image $\mathbf{x}_{\text{sRGB}}$ was generated through a conversion from its original ProPhoto encoding, we classify the pixels in $\mathbf{x}_{\text{sRGB}}$ into two classes: (1) the *in-gamut* pixels that were not clipped because they are inside the sRGB gamut; and (2) the possibly *out-of-gamut* pixels that have been potentially clipped down.

When the color of a pixel is on the sRGB gamut's boundary, the color may either be initially at the boundary or be clipped to the boundary. Therefore, these pixels are regarded as potentially out-of-gamut. We consider the pixels with one or more possibly saturated values in their three RGB channels as potentially out-of-gamut, with two exceptions of pure white and pure black colors, which are preserved during the color space conversion.

On the other hand, all non-saturated pixels—pixels with color values in the exclusive range $(0,255)$—are considered in-gamut.

Based on this simple definition, we compute a binary *out-of-gamut mask* for each input image to let our gamut mapping network focus on learning how to restore the out-of-gamut pixels.

The out-of-gamut mask $\mathbf{m}$ is computed based on the input sRGB image $\mathbf{x}_{\text{sRGB}}$ as

$$\mathbf{m} = \begin{cases} 0 & \text{where } \mathbf{x}_{\text{sRGB}} \text{ is black, white, or in-gamut} \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

The task of our network is to restore the out-of-gamut colors of pixels in the sRGB space to their original color in the ProPhoto space with the help of the in-gamut neighbors. Since we have the colorimetrically correct conversion for in-gamut pixels, our network focuses on estimating the amount of the saturation of each pixel instead of its original color, so we can add the estimated saturation (or residual) to the linearly mapped ProPhoto image to get the restored image—that is:

$$\hat{\bar{\mathbf{y}}}_{\text{ProPhoto}} = \mathbf{m} \odot f_\theta(\tilde{\mathbf{x}}_{\text{ProPhoto}}, \text{inv}(\mathbf{m})) + \tilde{\mathbf{x}}_{\text{ProPhoto}}, \tag{3}$$

where f_θ is our network, called *GamutNet*, with parameters θ, $\tilde{\mathbf{x}}_{\text{ProPhoto}}$ is the input image linearly mapped to the ProPhoto space, $\odot$ indicates point-wise masking operation, and $\text{inv}(\cdot)$ indicates binary mask inversion.

Given a dataset of N image pairs

$$\mathscr{D} = \left\{ \left(\tilde{\mathbf{x}}^{(i)}_{\text{ProPhoto}}, \bar{\mathbf{y}}^{(i)}_{\text{ProPhoto}} \right) \right\}_{i=1}^{N} , \tag{4}$$

we train our neural network by minimizing the L_1 loss between the estimated linear ProPhoto image and its ground-truth counterpart, over the out-of-gamut pixels

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \left| \mathbf{m}^{(i)} \odot \left(\hat{\bar{\mathbf{y}}}^{(i)}_{\text{ProPhoto}} - \bar{\mathbf{y}}^{(i)}_{\text{ProPhoto}} \right) \right| , \tag{5}$$

where $|\cdot|$ indicates the L_1 norm. Fig. 3 shows our gamut mapping framework and the architecture of *GamutNet*.

Dataset

To produce our dataset, we first gather a large number of camera RAW images to be processed through a software camera emulator. For this, we use RAW images from the MIT-Adobe FiveK [20], NUS [21], Cube+ [22], and RAISE [23] datasets. These datasets collectively represented 16,599 images captured from different cameras and of a wide variety of scene content.

We use the Adobe Camera RAW SDK to serve as our software ISP as done in [18]. The Adobe Camera RAW SDK mimics the steps of a typical ISP. We can modify the SDK to output the ProPhoto RGB after image enhancement and the final sRGB image. Access to these internal image encodings in the SDK is detailed in [18]. The image enhancement steps shown in Fig. 2 control the photo-finishing applied to the image using a combination of a 3D lookup table and a 1D lookup table. These combined lookup tables form a particular "picture style." In this paper, we use both the *Adobe Vivid* picture style and *Adobe Standard* picture style to render the images. We can think of this as a virtual camera that essentially integrates these two picture styles into a meta style. In the end, we generated 33,198 pairs of input (sRGB) and target (ProPhoto) images.

Before training, we need to exclude images that contain only a few out-of-gamut pixels. We also want to avoid images that result in overly saturated sRGB (i.e., most of the colors were out of gamut). Images with only a few out-of-gamut pixels do not provide sufficient information to train our model. Similarly, overly saturated images force the network to learn without in-gamut neighbors' information. To cull such images, we compute the ratio p of the number of the out-of-gamut pixels to the total number of pixels of each image:

$$p = n_s \, / \, (n_g + n_s) \, , \tag{6}$$

where n_s and n_g are the number of the out-of-gamut pixels and that of the in-gamut pixels of the image, respectively. After excluding images with p greater than 50%, we select 5,000 images with the largest p to form our final dataset. We randomly split the 5,000 images into training, validation, and testing sets of 3,000, 1,000, and 1,000 images, respectively.

Network Architecture and Training

We designed our DNN network to predict the residual between our input clipped-ProPhoto ($\tilde{\mathbf{x}}_{\text{ProPhoto}}$) image and the target image ($\bar{\mathbf{y}}_{\text{ProPhoto}}$). This means the residual represents the color difference between the linear-sRGB colors transformed into the ProPhoto color space with a clipped signal and its ProPhoto RGB counterpart (the original signal). To help the network focus on restoring the out-of-gamut-pixels' values only with the in-gamut pixels' aid, we concatenate the inverted out-of-gamut mask to the input image as illustrated in Fig. 3.

The out-of-gamut mask is also applied during the loss calculation to prevent the in-gamut pixels from contributing to the loss. This is because colorimetrically, the in-gamut pixels in the input image are almost identical to their counterpart in the ground-truth image and yield negligible loss. Using the out-of-gamut mask, we can filter out such trivial cases and focus our model on restoring out-of-gamut color values in the wider-gamut ProPhoto color space. We use L_1 loss for optimization, which is commonly used in many image restoration problems.

Results

We first evaluate our method's ability to perform gamut restoration from sRGB to ProPhoto. The evaluation uses the testing partition of our dataset described in the "Dataset" section. Note that all errors are reported only on the out-of-gamut colors.

ProPhoto Gamut Restoration

For our first evaluation, we compare our method with three representative approaches for color space conversion. The first is the recent gamut-extension method by Zamir et al. [5] that extends an image's color gamut for use on wide-gamut displays. This work performs an optimization procedure based on a novel perception model applied to any target color space. We note that the goal of Zamir et al.'s method *is not* accurate colorimetric reproduction. However, Zamir et al.'s work represents the state-of-the-art in gamut expansion that serves as an approach that would likely be used with modern displays.

The second approach we compare with is Adobe Photoshop's color conversion utilities. In particular, we use Adobe Photoshop's *absolute* and *relative* colorimetric color conversion utilities. These represent the most common strategy found in other software or display devices to minimize colorimetric errors. The difference between the two approaches is that for the relative colorimetric approach, the white-point is shifted slightly from D65 to D50 to match the white-point definition in ProPhoto. The white-point shift has little effect on color values in the final mapped image.

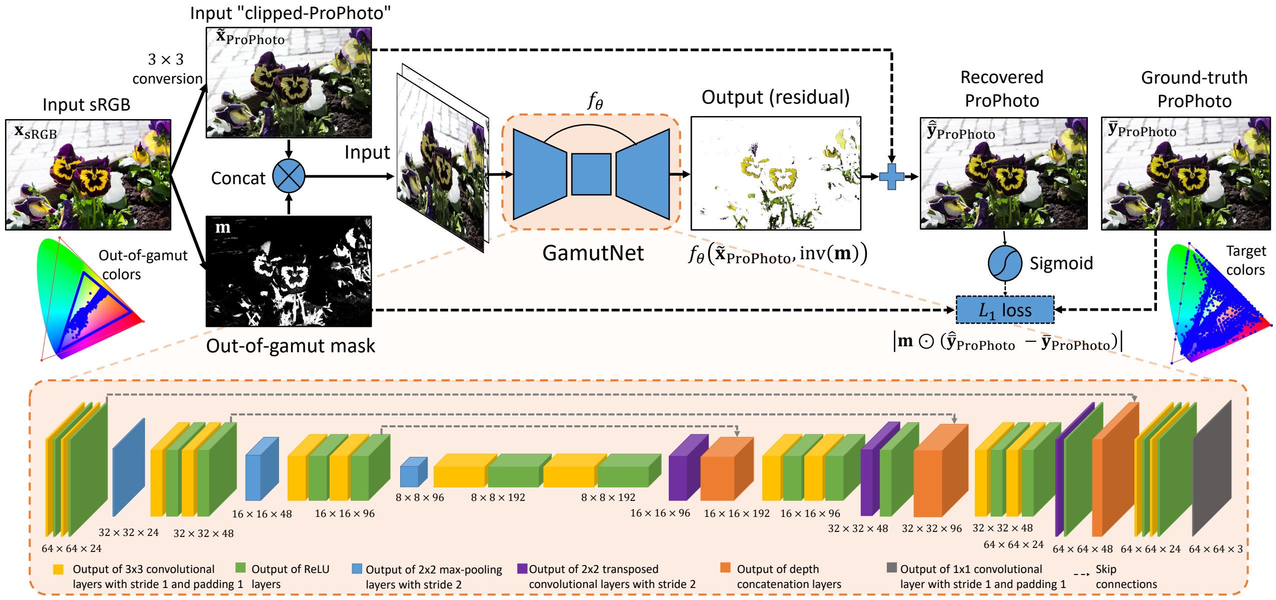

Figure 3. An overview of our method. We first convert the input sRGB image into a linear "clipped"-ProPhoto image using a 3×3 linear transformation; we then compute the out-of-gamut mask; the "clipped"-ProPhoto image and mask are concatenated as input to GamutNet; GamutNet predicts the residual pixel corrections and adds them to the input clipped ProPhoto image. Our L_1 loss is computed between our recovered ProPhoto output and its corresponding ground truth on the out-of-gamut pixels only. Our GamutNet architecture is shown in the bottom part. The out-of-gamut colors are shown in CIE-xy chromaticity plots for the input and target. Note that the chromaticity plot is a projection of the 3D color values, and therefore some out-of-gamut pixels will fall inside the 2D-triangle of the gamut's projection.

Table 1. This table shows the RMSE and ΔE_{00} results of color space conversion between sRGB and ProPhoto on 1,000 images with known ProPhoto color values. Methods used are Zamir et al. [5] gamut-expansion, Photoshop's color conversion feature (absolute and relative colorimetric), a linear transform, and our GamutNet result.

	RMSE				ΔE_{00}			
Methods	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3
Zamir et al.	9.533	4.405	6.730	10.127	6.101	3.836	5.720	8.315
Photoshop, Absolute	5.892	2.475	3.958	5.927	3.509	2.293	3.111	4.106
Photoshop, Relative	5.891	2.475	3.958	5.927	3.509	2.293	3.111	4.106
CIECAT02	5.289	2.083	3.383	5.243	3.244	2.061	2.861	3.867
Ours	**2.812**	**1.319**	**1.719**	**2.500**	**2.209**	**1.141**	**1.611**	**2.514**

Finally, for completeness, we also show the results of the 3×3 color space transform from linear-sRGB to linear-ProPhoto. This last method serves as the input to our GamutNet method.

Table 1 shows the comparison of all methods in terms of root mean squared error (RMSE) and ΔE_{00} computed over the out-of-gamut pixels. ΔE_{00} is a color-based metric that accounts for non-uniformity of perceptual color appearance. The lower the ΔE_{00} score, the more similar colors are considered perceptually. The table shows the mean score over the 1,000 test images, the top 25% quantile, the 50% quantile (medium), and the bottom 25% quantile for both RMSE and ΔE_{00}. Our GamutNet approach provides much better results for all metrics.

Subjective results are shown in Fig. 4. We also include results from Apple's *ColorSync* software [24]. Unlike Photoshop, we were not able to batch process the 1,000 test images with ColorSync. As a result, we included examples from ColorSync for visual comparisons only in our figures and not in Table 1. Fig. 4 shows several images each with the following: input sRGB; Photoshop (relative colorimetric) conversion; ColorSync's conversion; our GamutNet conversion; and (5) the ground-truth ProPhoto target. Below each image, we show a heat map of the RMSE error ranging from [0-20] and a CIE-xy chromaticity diagram showing the out-of-gamut sRGB colors after conversion. We can see that GamutNet gives the best quantitative results on both metrics. The chromaticity diagram also reveals that GamutNet shifts the out-of-gamut colors to be much more similar to their ground-truth positions.

Additional Evaluations

We trained and evaluated our model on an additional dataset, Cube+ [22]. This dataset contains 1,707 images captured by a single DSLR camera, a Canon EOS 550D. To render sRGB and ProPhoto RGB images, we used a different picture style (i.e., photo-finishing style) than the dataset used in the previous setup. Among the camera-matching styles, which are specific to a particular camera, we used the *Landscape* style, which fits well with the contents of Cube+, which consists mostly of outdoor scenes. After splitting the dataset, we trained our model with the same settings as the previous setup. Table 2 shows the evaluation metrics computed on 341 test images from that dataset; again results are computed for out-of-gamut pixels. Similar to the results using our dataset from the main paper, our GamutNet yields more than 50% improvement compared to existing approaches in terms of RMSE. We notice that the mean ΔE_{00} is a bit lower than on our

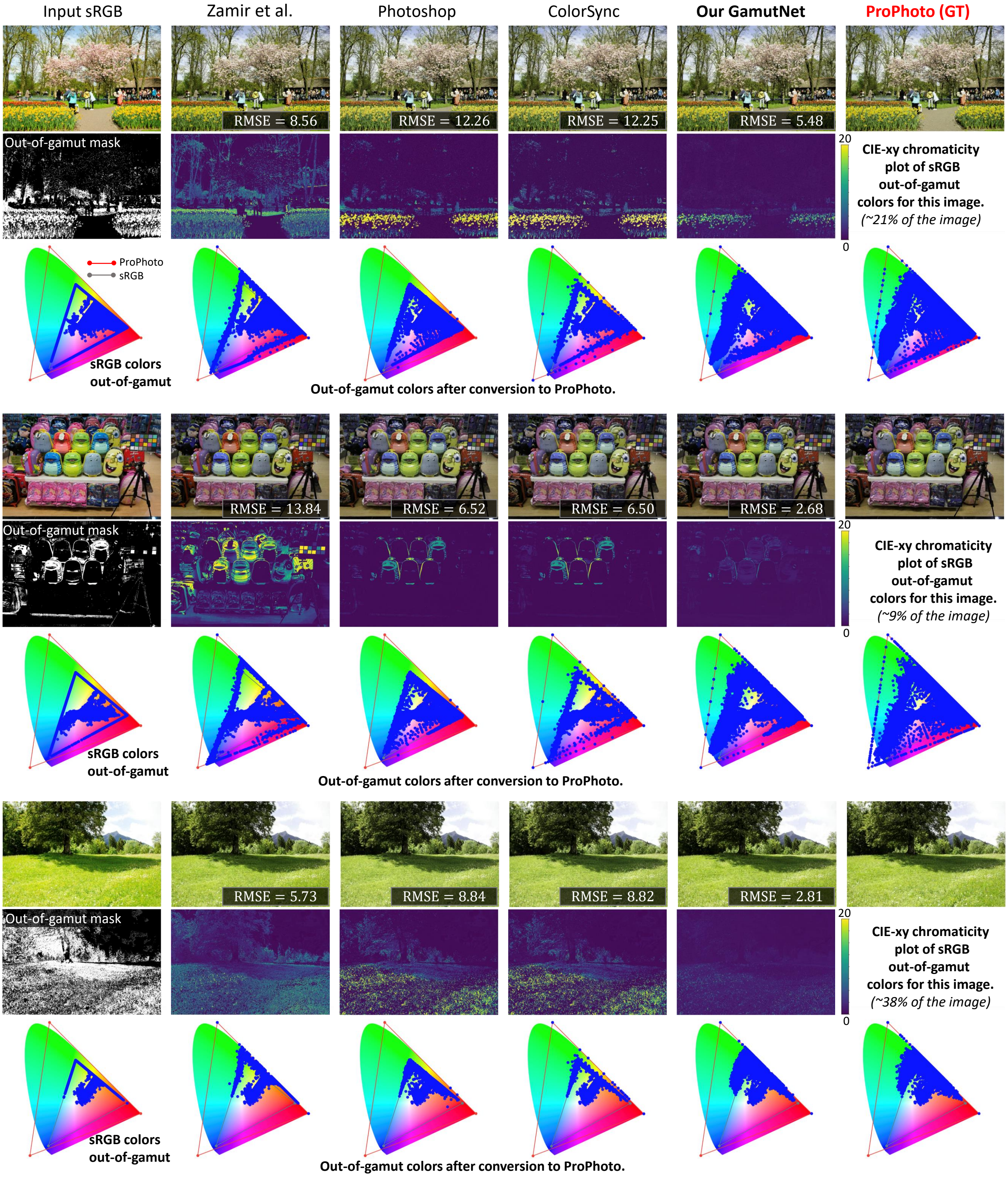

Figure 4. *Comparisons among the ProPhoto results of Zamir et al. [5] (gamut expansion), Photoshop [10], ColorSync [24], and our GamutNet. Heat maps of per-pixel RMSE are shown as well as plots of out-of-gamut colors on a CIE xy-chromaticity diagram showing the gamut for sRGB and ProPhoto. Photoshop and ColorSync are applying similar colorimetric conversion. The GamutNet provides the best color space conversion.*

Table 2. This table shows the mean RMSE and ΔE_{00} of color space conversion between sRGB and ProPhoto on an additional dataset. Methods used are Zamir et al.'s gamut-expansion [5], Photoshop's color conversion feature (relative colorimetric) [10], a linear transform, and our GamutNet.

Methods	RMSE	ΔE_{00}
Zamir et al.	16.959	3.807
Photoshop	5.419	1.463
CIECAT02	5.631	1.621
Ours	**2.608**	**1.364**

Table 3. This table compares the results obtained by using our models trained and evaluated in different scenarios. The first and the last rows correspond to the control cases, and the other is the experimental case used to assess our model's ability to generalize across datasets.

Trained on	Evaluated on	RMSE	ΔE_{00}
Main	Main	2.812	2.209
Main	Additional	3.121	1.696
Additional	Additional	2.608	1.364

dataset. We believe this is because the test dataset includes many images containing a relatively small number of saturated pixels.

To see how well our model generalizes, we evaluated the model on the described additional dataset. This heterogeneous evaluation is shown in the second row of Table 3. The other two rows are included for comparison. Regarding the two cases evaluated on the additional dataset, the experimental case is worse than the control case by 19.67% and 24.34% in terms of RMSE and ΔE_{00}, respectively, which is not a significant drop in performance. Moreover, compared to the results of CIECAT02 (third row in Table 2), which is the input, the experimental case is superior by more than 44%. Although the mean ΔE_{00} is increased by a fraction (~ 0.075), it is perceptually insignificant.

Summary

We have presented a DNN-based approach, termed GamutNet, to address wide-gamut color restoration from a small-gamut encoding (sRGB). The proposed DNN approach is enabled by the insight that cameras internally represent images in the wide-gamut ProPhoto color space. Working from this insight, we produced a new dataset of 5,000 images encoded in the small-gamut sRGB color space and the wide-gamut ProPhoto color space. This sRGB/ProPhoto paired training data allows us to treat the problem as one of restoration versus enhancement where we have a ground-truth target signal to recover. We showed that GamutNet can improve the colorimetric restoration by close to 50%. Our code and dataset will be made publicly available.

Acknowledgement

This work is supported by the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) programme and an NSERC Discovery Grant.

References

[1] International Electrotechnical Commission. IEC 61966-2-1 default RGB colour space – sRGB. *IEC*, 1999. 1

[2] *Adobe RGB Color Image Encoding*, 1998 (accessed: Nov.15.2020). 1

[3] Society of Motion Picture and Television Engineers. SMPTE Recommended Practice - D-Cinema Quality — Reference Projector and Environment. *RP 431-2:2011*, pages 1–14, 2011. 1

[4] Kevin E. Spaulding, Geoffrey J. Woolfe, and Edward J. Giorgianni. Reference input/output medium metric rgb color encodings. In *Color Imaging Conference*, 2000. 1, 2

[5] Syed Waqas Zamir, Javier Vazquez-Corral, and Marcelo Bertalmio. Vision models for wide color gamut imaging in cinema. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Nov 2019. 1, 2, 3, 4, 5, 6

[6] Byoung-Ho Kang, Maeng-Sub Cho, Jan Morovic, and M. R. Luo. Gamut extension modelling based on observer experimental data. *Color and Imaging Conference*, 2001(1):158–162, 2001. 1

[7] Syed Waqas Zamir, Javierzamir2019vision Vazquez-Corral, and Marcelo Bertalmío. Gamut extension for cinema. *IEEE Transactions on Image Processing*, 26(4):1595–1606, 2017. 1, 2

[8] Javier Vazquez-Corral and Marcelo Bertalmıo. Gamut mapping for visual attention retargeting. *Color and Imaging Conference*, 2017(25):313–316, 2017. 1

[9] Hoang Le, Mahmoud Afifi, and Michael S Brown. Improving color space conversion for camera-captured images via wide-gamut metadata. *Color and Imaging Conference*, 2020(28):193–198, 2020. 1, 2

[10] Adobe Inc. Photoshop. Last accessed 16 Nov, 2020. 1, 5, 6

[11] J. Morovic and M. R. Luo. Evaluating gamut mapping algorithms for universal applicability. *Color Research & Application*, 26:85 – 102, 12 2000. 2

[12] John J McCann. Color gamut mapping using spatial comparisons. In *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI*, 2000. 2

[13] Chae-Soo Lee, Yang-Woo Park, Seok-Je Cho, and Yeong-Ho Ha. Gamut mapping algorithm using lightness mapping and multiple anchor points for linear tone and maximum chroma reproduction. *Journal of Imaging Science and Technology*, 45(3):209–223, 2001. 2

[14] Ján Morovič. *Color gamut mapping*. John Wiley & Sons, 2008. 2

[15] Arne Bakke, Ivar Farup, and Jon Hardeberg. Evaluation of algorithms for the determination of color gamut boundaries. *Journal of Imaging Science and Technology*, 54(5):50502–1, 2010. 2

[16] Jens Preiss, Mark D Fairchild, James A Ferwerda, and Philipp Urban. Gamut mapping in a high-dynamic-range color space. In *Color Imaging XIX: Displaying, Processing, Hardcopy, and Applications*, 2014. 2

[17] International Color Consortium. White paper 9: Common color management workflows and rendering intent usage, 2005. 2

[18] Hakki Can Karaimer and Michael S. Brown. A software platform for manipulating the camera imaging pipeline. In *ECCV*, 2016. 2, 3

[19] Nathan Moroney, Mark D Fairchild, Robert W.G. Hunt, Changjun Li, M Ronnier Luo, and Todd Newman. The CIECAM02 color appearance model. In *Color and Imaging Conference*, 2002. 2

[20] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR*, 2011. 3

[21] Dongliang Cheng, Dilip K Prasad, and Michael S Brown. Illuminant estimation for color constancy: Why spatial-domain methods work and the role of the color distribution. *JOSA-(A)*, 31(5):1049–1058, 2014. 3

[22] Nikola Banic and Sven Loncaric. Unsupervised learning for color constancy. *CoRR*, 2017. 3, 4

[23] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. RAISE: A raw images dataset for digital image forensics. In *ACM Multimedia*, 2015. 3

[24] Apple Inc. ColorSync utility. 4, 5