Deep Color Management and Image Quality Enhancement for Industrial Printers

Alberto Vitto[υ], **Fabio Lanzi**[υ], **Simone Calderara**[υ],
Andrea Mariani[δ], **Elena Pellesi**[δ], **Rita Cucchiara**[υ]
υ : University of Modena and Reggio Emilia; δ : Digital Design SRL
υ : <name>.<surname>@unimore.it; δ : <name><surname>@ddsrl.com

Abstract

Today, a rather challenging task in the design industry is related to the automation of color management and image enhancement process for printing on design surfaces. Such automation would notably speed up the work of graphic designers in editing and preparing images before the printing process, favoring at the same time a standardization of the output. This paper proposes novel methods that tackle the highlighted problem with Deep Learning techniques, using convolutional neural networks to reproduce creative operations which were previously a prerogative of expert designers. Obtained results are encouraging and could trace the path to future improvements.

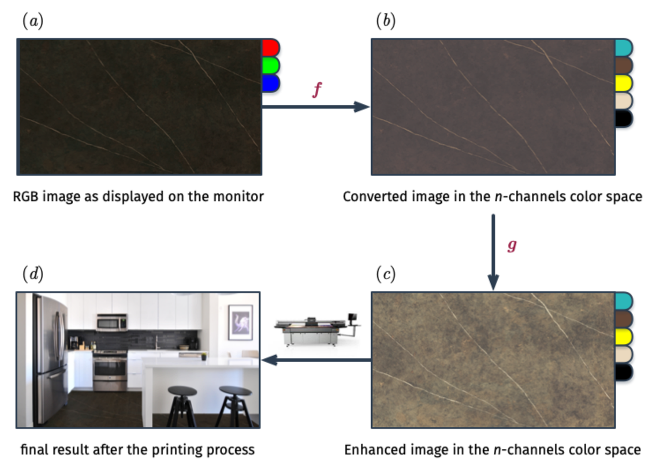

Figure 1: Scheme of the color management and image enhancement task for industrial printers. (*a*) $\mathbf{x}'$: RGB image as it is seen on the monitor; (*b*) $\mathbf{x}''$: CONVERTED image in the n-channels profile, where $n \geq 4$; (*c*) $\mathbf{x}''_{+}$: ENHANCED n-channels image; (*d*) final result after the printing process. Note how $\mathbf{x}'$ must go through the transformations f (color-space conversion) and g (enhancement) so that the printed result matches the image displayed on the monitor.

1 Introduction

This work will present the study of a Deep Learning based system able to enhance the visual quality of industrial design surfaces according to specific properties dictated by color profiles. This work has been developed in collaboration with *Digital Design Srl*, a leading design company located in Fiorano Modenese (Modena) specialized in creating high quality design surfaces for different applications, in particular ceramic tile printing.

In the design surfaces industry the color management process is crucial considering that the aspect of images seen from a monitor does not necessarily correspond to the actual outlook once they will be printed. In this context, the designer's job is to modify the image so that, once printed, it is consistent with the quality and level of detail seen on the monitor (or on paper) by the customer. This operation is currently done manually inside *Photoshop* (or other advanced image editing tools), and it is particularly complex and time-consuming; moreover, such process is not deterministic as it's based on the designer's experience and sensitivity. See Figure 1.

In our specific case study, the process starts from the scan of very large surfaces of different materials like marble, wood and cement, generating super-high-resolution RGB images whose average dimensions can often be larger than 25000×50000 (around 3 GB each). These images are then manipulated within *Photoshop*, by first applying a special custom color profile (depending on the industrial printer that will manage the printing process); secondly, the converted images are manually enhanced to ensure the best possible quality of the final printed product.

Usually, industrial printers work with specific multichannel output color profiles that map RGB images in multichannel color spaces, from the well-known CMYK to more complex multichannel color spaces. The color-space conversion in these multichannel spaces is lossy, meaning that often it can't reproduce the full range of colors in the initial high-resolution RGB image. The sensibility and experience of the designer are required to preserve and enhance color properties as they are asked from the customers; however, as mentioned above, these operations, carried out through complex procedures like simulating the actual rendering of the surface once it will be printed in production, are time-consuming and non-deterministic. Given these motivations, the objective of this work is to try to design a system able to help graphic designers in the context of images enhancement and color spaces conversion.

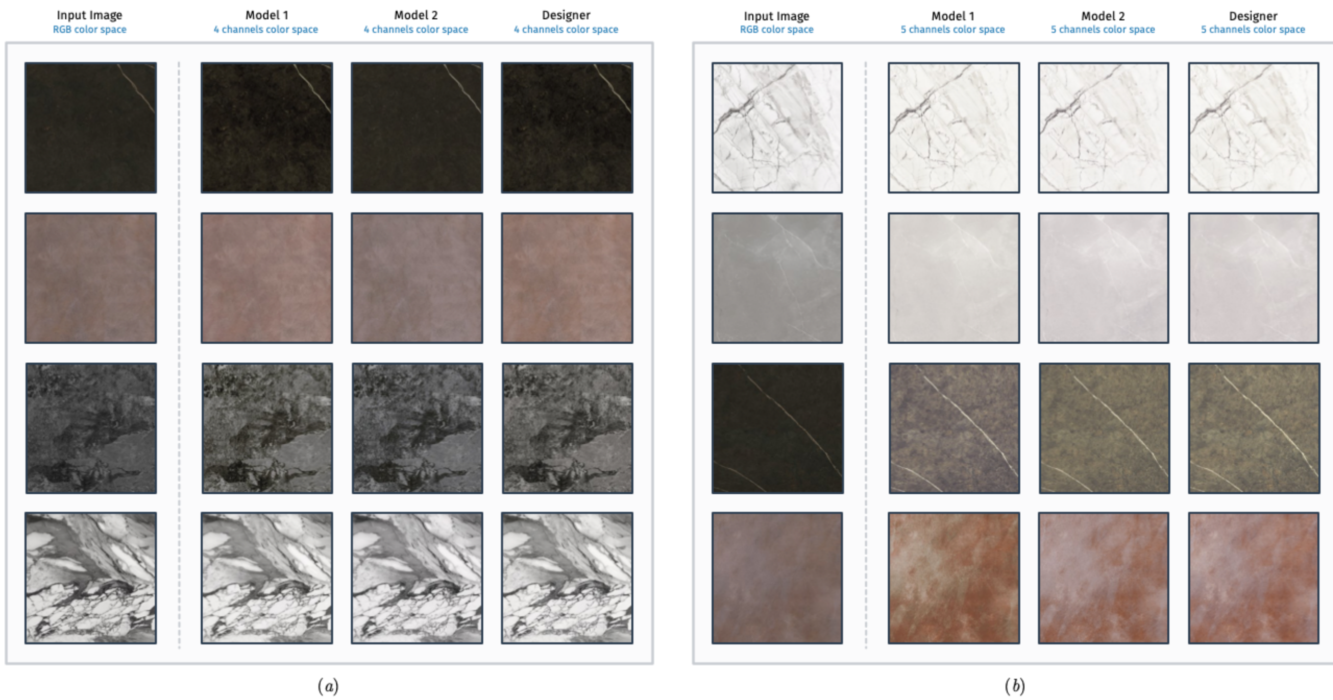

Figure 2: (a) Qualitative results with 4-channel color space. (b) Qualitative results with 5-channel color space. In both cases, we show the RGB image, CONVERTED and ENHANCED image by designer, CONVERTED and ENHANCED image generated by Model 1, CONVERTED and ENHANCED image generated by Model 2.

2 Related Works

As highlighted in Section 1, the purpose of this paper is to tackle image enhancement and color space conversion tasks using Deep Learning techniques and, to the best of our knowledge, as it is a rather specific task, no similar work exists in literature. Therefore, this work could be considered a first step towards this specific field of application.

For the design of our system, we took inspiration from the well-known *U-Net* [Ronneberger *et al.*, 2015] architecture, widely used by the community to tackle similar Computer Vision tasks, such as image-to-image translation [Laxman *et al.*, 2021], colorization [Di *et al.*, 2021], super-resolution [Hu *et al.*, 2019] and general-purpose image enhancement [Huang *et al.*, 2019].

3 Method Overview

In this section we formalize the main operations that characterize the color management and image quality enhancement task.

The first operation is the application of a color profile; it can be viewed as a function f that, given an image in the source RGB space, returns the corresponding image in the multichannel destination space defined by the profile. The second operation represents the enhancement work carried out by the graphic designer once the image has been converted in a multichannel space and it can be modeled with a function g.

Formally, the whole process is as follows:

$$(g \circ f)(\mathbf{x}') = g(f(\mathbf{x}')) = g(\mathbf{x}'') = \mathbf{x}''_{+} \tag{1}$$

where:

- $\mathbf{x}'$: original image in RGB space;
- $\mathbf{x}''$: CONVERTED image in multichannel space;
- $\mathbf{x}''_{+}$: ENHANCED image in multichannel space;
- f: color-space conversion function;
- g: enhancement function.

In the following subsections, we describe two different models (Model 1 and Model 2 in Figure 3) which aim to automate and speed up the process described above, focusing on the operations that are currently performed manually.

3.1 Model 1

The first model takes as input an already converted image $\mathbf{x}''$ and tries to approximate only the g function. The output of this model is a raw image in a multichannel color space which cannot be directly handled by *Photoshop*, because the file format required to save this kind of images, called .psb, is proprietary and it's extremely complex to manipulate.

Therefore we came up with a custom solution using the *LittleCMS*[1] open source library, containing functions for conversions between color spaces, which made it possible to bring

[1] *LittleCMS* library → https://github.com/mm2/Little-CMS

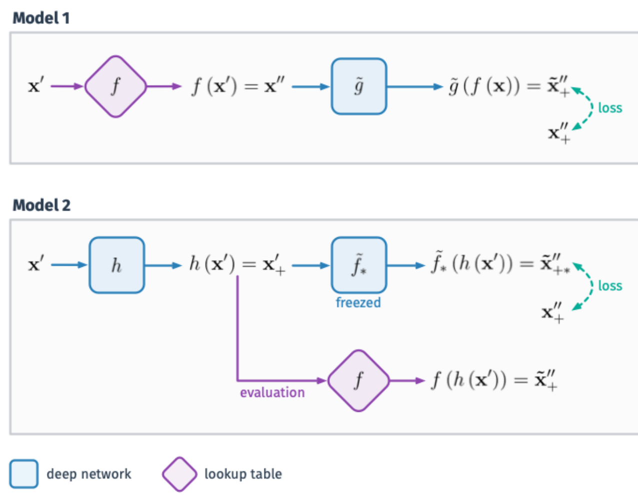

Figure 3: Diagram of the two proposed solutions, namely Model 1 and Model 2, showing the inputs and outputs of each of the main component blocks. The blocks modeled by neural networks are shown in blue, while the blocks consisting of lookup tables are shown in purple.

back the output in RGB space and save it in whichever image format desired. The peculiarity of this RGB image is that, once opened in *Photoshop* and converted with the proper color profile, no other actions are needed: the enhancement work have been carried out by the model, while the conversion is left to *Photoshop*. The complete process can be visualized as ($\rightarrow$ indicates temporal consequence):

$$f(\mathbf{x}') = \mathbf{x}'' \rightarrow \widetilde{g}(\mathbf{x}'') \rightarrow \mathcal{L}(\widetilde{\mathbf{x}''_+}, \mathbf{x}''_+) \tag{2}$$

where:

- $\widetilde{g}$: approximation of the enhancement function.
- $\mathcal{L}$: loss function, used to train the network, between ground truth ENHANCED image ($\mathbf{x}''_+$) and predicted ENHANCED image ($\widetilde{\mathbf{x}''_+}$).

3.2 Model 2

The second model tries to follow the path of Model 1 by applying the color profile f during the training phase using a differentiable network that approximates it. The idea is that an RGB image is modified during the passing inside the custom *U-Net* model that does the enhancement and a small pretrained 3-layer convolutional network put in the last layer emulates the application of the color profile that it would occur inside *Photoshop*: in this way the main network is aware of how the image will look like once converted with the true color profile in *Photoshop* and can therefore make adjustments and enhancements accordingly.

A new function h is introduced which applies an enhancement over an RGB image input image and outputs an RGB ENHANCED image. This is opposed to the real process described in (1) where the enhancement process is applied after the conversion. It's important to note we have hypothesized the assumption that

$$(g \circ f)(\mathbf{x}') \simeq (f \circ h)(\mathbf{x}') \tag{3}$$
$$g(f(\mathbf{x}')) \simeq f(h(\mathbf{x}')) \tag{4}$$

meaning the swap of the operations conducts to the same result. This constraint is forced at loss level during training.

For this solution, we distinguish 2 different operating modes: a training mode, where $\widetilde{f}_*$ is the approximation of f, necessary to allow back-propagation of gradient, which is imported as a pretrained model with freezed weights; an evaluation mode, where $\widetilde{f}_*$ is no longer needed because true f is directly applied inside *Photoshop*. The better approximation of $\widetilde{f}_* \simeq f$, the better results in test mode.

So, in training mode we have:

$$\widetilde{f}_*(h(\mathbf{x}')) \rightarrow \widetilde{f}_*(\mathbf{x}'_+) \rightarrow \mathcal{L}_{\text{train}}(\widetilde{\mathbf{x}''_{+}}_*, \mathbf{x}''_+) \tag{5}$$

where:

- h: enhancement function in RGB space;
- $\widetilde{f}_*$: approximation of the conversion function;
- $\mathcal{L}_{\text{train}}$: loss metric used in train mode; it requires $\widetilde{f}_*$ to be differentiable.

While, during the evaluation phase, we have:

$$f(h(\mathbf{x}')) \rightarrow f(\mathbf{x}'_+) \rightarrow \mathcal{L}_{\text{eval}}(\widetilde{\mathbf{x}''_+}, \mathbf{x}''_+) \tag{6}$$

where:

- $\mathcal{L}_{\text{eval}}$: loss metric used in test mode; do not require f to be differentiable.

4 Experiments and Results

An overview with qualitative results of the methods applied to both 4 and 5-channel images is presented in Figure 2, where it's clear how predicted and real output look similar in different scenarios. To better evaluate the quality of the obtained results we decided to develop a simple survey to be submitted to some graphic designers of *Digital Design Srl*. The main reason of this choice was that the metrics used for the test phase, namely L1, PSNR [Horé e Ziou, 2010] and SSIM [Wang *et al.*, 2004], were useful only to have a general idea of the quality and the correctness of the models: in the contexts where there is an improvement process, carried out through human intervention, these metrics could be misleading and it's unlikely they would provide a judgment in line with a human would give.

For these reasons 8 expert graphic designers were asked to participate in a survey developed to check the quality of the results. The survey has been released as `.zip` file composed of 20 folder, each with one real enhanced and converted image by a designer, one generated enhanced and converted by Model 1 and one generated enhanced and converted by Model 2. In each of the 3 images the conversion has been applied

Class	Score		
	Model 1	Model 2	Designer
-0.5: "Poor"	36	56	38
1: "Good"	85	74	79
2: "Excellent"	39	30	43
	145	**106**	**146**

Table 1: Results of survey for 4-channel color profile images. Each image was rated by participants according to one of the following scoring classes: -0.5 ("Poor"), 1 ("Good"), or 2 ("Excellent"). The table shows the total number of votes for each score class, while the last row shows the final score obtained by our 2 models and the expert designer.

Class	Score		
	Model 1	Model 2	Designer
-0.5: "Poor"	60	60	40
1: "Good"	61	79	75
2: "Excellent"	39	21	45
	109	**91**	**145**

Table 2: Results of survey for 5-channel color profile images. Each image was rated by participants according to one of the following scoring classes: -0.5 ("Poor"), 1 ("Good"), or 2 ("Excellent"). The table shows the total number of votes for each score class, while the last row shows the final score obtained by our 2 models and the expert designer.

with the real color profile inside *Photoshop* thus generating `.psb` format images; the latter aspect was also useful to hide the origin of the images and therefore designers did not have a clue about what was the real one and the generated ones and could give an opinion only based on their perception and experience; they had the freedom to open and compare each image in any way they wanted. The task was then to assign a score to each of the 3 images in the folders using *Google Forms*[2], namely "Excellent", "Good" or "Poor": then, each score was converted to a numeric value, as shown in Table 1 and Table 2.

Table 1 and Table 2 show the survey results for 4 and 5-channel images respectively. These experiments allowed us to quantitatively evaluate the results of our models, which generally showed scores comparable to those of human designers. It was surprising to see how Model 1 got almost the same number of "Excellent" in both 4 and 5-channel images and it was also interesting to note in 4-channel survey (Table 1) how the images manually enhanced by designer were rated with a higher number of "Poor" (38) with respect to the ones generated from Model 1 (36). It's important to highlight that these survey didn't take into account the time spent in generating the output: our models are approximately 10 times faster in generating the `ENHANCED` output with respect to an average designer and this significantly improves productivity.

[2] `https://docs.google.com/forms`

5 Conclusions and Future Perspectives

In this paper, we studied a topic that was unexplored in an academic environment while having a strong industrial application.

The color management and image quality enhancement of design surfaces is crucial to grant top quality in printing over design surfaces, but it's also time-consuming as it requires complex manual operations. We strongly believe that our system can be of great help to graphic designers, as it provides them with an automatic tool to generate ready-to-use high-quality images. In addition, the opinions collected through the survey described in Section 4 show how our system can produce results that are often indistinguishable from those of a real designer, even in the eyes of a domain-expert operator.

Given the quality of the final result and the industrial value of our system, we will certainly proceed with the study of this problem. In particular, we plan to focus on the ability to manage full-resolution images without downscaling, which is an extremely complex task given the enormous resolutions typically used in this specific sector.

References

[Di *et al.*, 2021] Yide Di, Xiaoke Zhu, Jin Xin, Qiwei Dou, Wei Zhou, e Qing Duan. Color-unet++: A resolution for colorization of grayscale images using improved unet++. *Multimedia Tools and Applications*, 80:1–20, 11 2021.

[Horé e Ziou, 2010] Alain Horé e Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[Hu *et al.*, 2019] Xiaodan Hu, Mohamed Naiel, Alexander Wong, Mark Lamm, e Paul Fieguth. Runet: A robust unet architecture for image super-resolution. In *Conference on Computer Vision and Pattern Recognition*, 06 2019.

[Huang *et al.*, 2019] Jie Huang, Pengfei Zhu, Mingrui Geng, Jiewen Ran, Xingguang Zhou, Chen Xing, Pengfei Wan, e Xiangyang Ji. *Range Scaling Global U-Net for Perceptual Image Enhancement on Mobile Devices: Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 230–242. European Conference on Computer Vision, 01 2019.

[Laxman *et al.*, 2021] Kumarapu Laxman, Shiv Ram Dubey, Baddam Kalyan, e Satya Raj Vineel Kojjarapu. Efficient high-resolution image-to-image translation using multi-scale gradient u-net. *arXiv preprint arXiv:2105.13067*, 2021.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, e Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, volume 9351, pages 234–241, 10 2015.

[Wang *et al.*, 2004] Zhou Wang, Alan Bovik, Hamid Sheikh, e Eero Simoncelli. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13:600 – 612, 05 2004.