

Differentiable Histogram Loss Functions for Intensity-based Image-to-Image Translation

Mor Avi-Aharon^{ID}, Assaf Arbelle^{ID}, and Tammy Riklin Raviv^{ID}, Member, IEEE

Abstract—We introduce the HueNet - a novel deep learning framework for a differentiable construction of intensity (1D) and joint (2D) histograms and present its applicability to paired and unpaired image-to-image translation problems. The key idea is an innovative technique for augmenting a generative neural network by histogram layers appended to the image generator. These histogram layers allow us to define two new histogram-based loss functions for constraining the structural appearance of the synthesized output image and its color distribution. Specifically, the color similarity loss is defined by the Earth Mover’s Distance between the intensity histograms of the network output and a color reference image. The structural similarity loss is determined by the mutual information between the output and a content reference image based on their joint histogram. Although the HueNet can be applied to a variety of image-to-image translation problems, we chose to demonstrate its strength on the tasks of color transfer, exemplar-based image colorization, and edges → photo, where the colors of the output image are predefined. The code is available at <https://github.com/mor-avi-aharon-bgu/HueNet.git>.

Index Terms—Intensity histogram loss functions, mutual information loss, histogram layers, earth movers distance, image-to-image translation.

I. INTRODUCTION

DEEP Neural Networks have dramatically improved the state-of-the-art in many practical domains [1], [2]. Although numerous loss functions have been proposed, metrics based on intensity histograms, which represent images by their color distributions [3], [4] or joint histograms of image pairs are seldom considered nowadays. The main obstacle seems to be the histogram construction which, in its standard form, is not a differentiable operation and therefore cannot be smoothly incorporated into a deep learning framework.

To exploit the strengths of histogram-based tools, we propose a novel generative deep learning method dubbed the *HueNet*, which enables a differentiable construction of intensity (1D) and joint (2D) histograms. Histogram construction is made possible by an innovative network architecture composed of pre-designed

Manuscript received 3 July 2021; revised 6 March 2023; accepted 3 May 2023. Date of publication 24 May 2023; date of current version 5 September 2023. This work was partially supported by the Israel Science Foundation under Grant 2497/19. Recommended for acceptance by M. Cheng. (*Corresponding author: Tammy Riklin Raviv.*)

The authors are with the School of Electrical and Computer Engineering, Ben-Gurion University, Beer-Sheva 84105, Israel (e-mail: moraviah@post.bgu.ac.il; arbellea@post.bgu.ac.il; rrtammy@bgu.ac.il).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TPAMI.2023.3278287>, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3278287

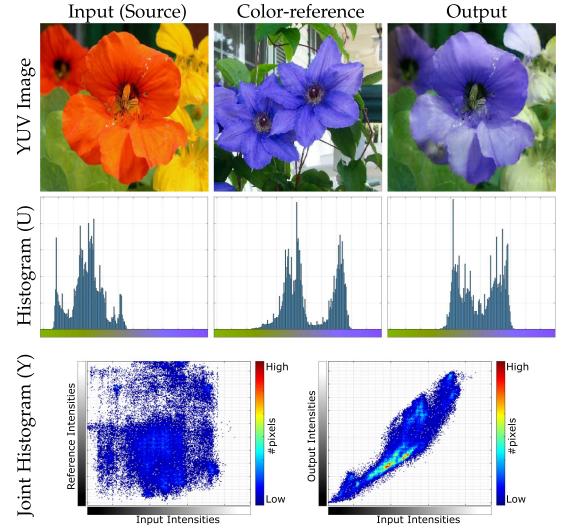


Fig. 1. Color transfer exemplified using the YUV color-space. First row: The output image has the structure of the input image but the colors of a color-reference image. Second row: The respective intensity histograms of the images in row 1. Third row: Color-coded joint histograms of the input and the color reference (left) images and the input and output images (right). The intensity and joint histograms refer to the U-channels and the Y-channels of the images, respectively.

histogram layers appended to the image generator. The outputs of these histogram layers allow us to define two novel differentiable loss functions to constrain the color distribution of the output image and its structural similarity to a reference image. Specifically, we define an Earth Mover’s Distance loss applied to two intensity histograms and a mutual information loss applied to a joint histogram of an image pair. It is important to note that the histogram layers are not learned but rather remain fixed throughout the training. They, however, take part in the back-propagation process in which the image generator weights are updated based on the histogram-based loss functions.

Embedding 1D histograms into a neural network was proposed in [5] for semantic segmentation and object detection, and for raw \leftrightarrow RGB in [6]. In both works the learnable histogram layers were intermediate layers and were not utilized for defining a loss function but rather to exploit global context.

Color and intensity histograms are useful representations for image-to-image translation tasks, in particular when the desired, output images do not necessarily exist. Consider for example the color transfer problem, as exemplified in Fig. 1 (first row), where the aim is to paint the input image with the colors of

a reference image. For this kind of unpaired learning task, neither of the standard loss functions based on a pixel-by-pixel comparison, e.g., mean-square error (MSE) or ℓ_1 , can be used. Classical methods for color transfer were based on the concept of histogram matching, where the main idea was to adapt a color histogram of a given image to the target image [7], [8], [9]. The portrait lighting transfer of Shu et al. [10] utilized the mass transport approach based on color histogram matching and facial geometry. In this work, we exploit intensity histogram matching to constrain similarity of color distributions using a neural network as an optimizer. The HueNet is trained to perform color-matching by minimizing a color-similarity loss defined by the EMD between the output and a reference image histograms, as shown in the second row of Fig. 1.

Constraining the content (or structure) of the output image to resemble that of the source image is a non-trivial task given their different color distributions. In a seminal work, Viola and Wells [11] used a cost function based on mutual information (MI) for multi-modal image registration, where the source and the target images had different intensities. The MI between two images was calculated based on their joint histogram, as illustrated in Fig. 1 (third row). In the context of image registration and texture synthesis [12], [13], the joint matrix is called a co-occurrence matrix. One of the HueNet key ideas is the adaptation of the MI constrain used in classical methods for matching a pair of images with different color distributions to deep neural networks. Specifically, the unique HueNet architecture inherently constructs the joint histogram between the network output and the source image. This allows us to define a corresponding MI loss function and use it to enhance image-to-image translation tasks. To the best of our knowledge, the MI of an image pair has never been used in a similar manner. This is also the first time a differentiable construction of joint histograms is incorporated into a deep learning framework.

Recent image synthesis methods and image-to-image translation, in particular, are mostly based on deep learning frameworks. Since the main aim is to generate realistic examples, adversarial frameworks were shown to be very effective [14]. In their Pix2Pix framework, Isola et al. [15] performed image-to-image translation (e.g., colorization of grayscale images and edges \rightarrow photo) by using adversarial loss as well as ℓ_1 loss between corresponding pixels in the network's output and the desired target image. In this sense, Pix2Pix is a supervised method and obviously cannot be applied to problems (such as color transfer) where the desired target image does not exist. Moreover, as discussed in [15] the images generated by using ℓ_1 loss tend to have grayish or brownish colors when there is any uncertainty regarding which of several plausible color values a pixel should take. Specifically, ℓ_1 is minimized by choosing the median of the conditional probability density function over all possible colors.

To address the issue of insufficient color diversity due to uncertainty [16] proposed a class-based colorization method in which the loss of each pixel in an image of a particular class was weighted by the frequency of its color in that class. Larsson et al. [17] proposed to use spatially localized multilayer slices (hypercolumns) as per-pixel descriptors for the prediction

of the respective pixel-wise color distributions. The HistoGAN of Afifi et al. [18] presented visually compelling results by exploiting the features of target histograms to control GAN-generated images as well as recoloring real images. A different approach was presented in [19], [20] for exemplar-based colorization where a reference color image was used to control the output colors. Alternatively, colorization by user scribbles was proposed in [21], [22] for user-defined output colors.

The problem of color transfer (which is an unpaired version of image colorization) was recently addressed in [23] using a two-step pipeline for deep semantic correspondences (via VGG19) between the input and the reference images followed by local color transfer in the image domain. In [24] a differential loss function to constrain the transformation from the input to the output to be locally affine in color-space has been introduced. While these methods provide visually appealing results they require some structural and semantic similarity of the reference with respect to the input image. Moreover, semantic information in the form of image segmentation is required as well. A different approach to image-to-image translation in an unpaired setting was presented in Zhu et al. seminal CycleGAN [25]. The key idea of [25] is to use cycle-consistent adversarial networks to enable style and color transfer (e.g., summer to winter) when the desired output image cannot be used for training. CycleGAN presents compelling results, but since in many cases the cyclic consistency constraint is not sufficient, additional supervision and loss functions are often required. Moreover, the CycleGAN can be only tested for the domain transfer it was trained on.

The HueNet presents a conceptual alternative to existing image-to-image translation methods. It does not require the extraction of semantic features; neither does it need a reference color image with semantic similarity to the input image. Although, similar to other approaches, the HueNet utilizes an adversarial loss to generate realistic images ensuring, for example, green grass and blue sky and not the other way around, it does not exclusively rely on it. Instead, the HueNet exploits two classical and fundamental image processing notions via its innovative architecture which enables a differentiable construction of the output image histograms. We stress that unlike other deep learning approaches the histograms are not predicted but actually calculated by a smart engineering of the structure and the weights of the histogram layers. This is the main contribution of our framework. The HueNet generator is trained such that the color distributions of the synthesized output image are constrained by reducing the distance (using EMD loss) between pairs of intensity histograms. In a similar manner, the structural appearance of this output image is controlled by maximizing the MI based on the joint histogram of an image pair. Obviously, once the training is completed these histogram layers (like the discriminator) are no longer required.

We demonstrate the strengths of the HueNet on different paired and unpaired image-to-image translation as illustrated in Fig. 2, using several publicly available datasets. These include color transfer for the flowers [26] and the indoor-outdoor scene [24] datasets, exemplar-based image colorization for the summer-winter dataset [25] and color-reference edges \rightarrow photo for the shoes [27] and bags [28] datasets. Note that we present

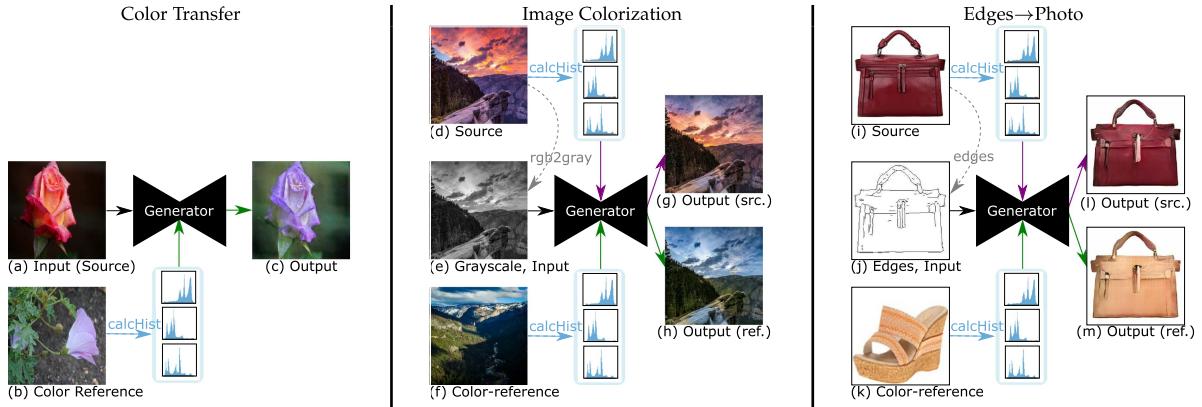


Fig. 2. *Image-to-image translation tasks*. Color transfer (left), exemplar-based image colorization (middle), and edges → photo (right). The inputs for all tasks consist of a content image (an edge map in the case of edge → photo) and histograms of the three color channels of the color reference image. The output for all the tasks is an image with the content of the input image and the colors of the color-reference image. For example, (l) and (m) are two possible outputs of the edges → photo, for the input histogram of either (i) or (k), respectively. Images (g) and (l) were deliberately generated with the colors of the source images, to enable visual comparison between the real images and the images synthesized by the HueNet.

non-standard nuances for the well-studied image colorization and edges → photo tasks by constraining the color distributions of the output images to resemble pre-selected color histograms extracted from some color reference images or a color pallet.

II. METHODS

In this section, we review the main principles underlying the differentiable construction of 1D and 2D intensity histograms (Section II-A). We then define the HueNet loss functions (Section II-B). The network architecture, and the implementation details are presented in Sections II-C and II-D.

A. Differentiable Histograms Construction

1) Color Spaces: To address image-to-image translation problems we could theoretically have been chosen any color space, but we decided on the YUV and HSV color spaces. The YUV color space is composed of one luma component (Y) and two chrominance components, called U and V. The Y channel is in the range [0,1] and the range of the U and the V channels is [-.5, .5]. For practical reasons we mapped all channel values to [-1, 1]. The HSV color space separates the color component of the image (hue) from its shade (saturation) and black/white ratio (value). Note that the hue channel is cyclic. In Section II-B.1a we present our approach to accommodate this case. In the following sections, we refer to each color channel as a gray-level image.

2) Differentiable 1D Histogram Formulation: Images acquired by digital cameras have three color channels, each with a discrete range of K intensity values. The intensity distribution of each channel can be described by an intensity histogram obtained by counting the number of pixels in each intensity value. Considering synthesized images that can take any value in the continuous range $[-1, 1]$, we define the intensity of an image pixel $x \in \Omega$, in a particular channel as $I(x) \in [-1, 1]$. We use the Kernel Density Estimation (KDE) [29], [30] for estimating

the gray level density f_I of an image's channel I as follows:

$$\hat{f}_I(g) = \frac{1}{NW} \sum_{x \in \Omega} \mathcal{K}\left(\frac{I(x) - g}{W}\right) \quad (1)$$

where $g \in [-1, 1]$, $\mathcal{K}(\cdot)$ is the kernel, W is the bandwidth and $N = |\Omega|$ is the number of pixels in the image. A valid kernel should be a non-negative real-valued and symmetrical integrable function. We choose to define the kernel $\mathcal{K}(\cdot)$ as the derivative of the logistic regression function $\sigma(z)$ as follows

$$\mathcal{K}(z) = \frac{\partial \sigma(z)}{\partial z} = \sigma'(z) = \sigma(z)(1 - \sigma(z)), \quad (2)$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$. The KDE is applied with a reflection boundary correction (see Appendix, available online, for more details).

For the construction of a smooth and differentiable image histogram, we partition the interval $[-1, 1]$ into K sub intervals $\{B_k\}_{k=0}^{K-1}$, each interval with length $L = \frac{2}{K}$ and center $\mu_k = -1 + L(k + \frac{1}{2})$, then $B_k = [-1 + kL, -1 + (k + 1)L]$. We then define the probability of pixel in the image to belong to a certain gray level interval (the value of a normalized histogram bin) as

$$P_I(k) \triangleq \Pr(g \in B_k) = \int_{B_k} \hat{f}_I(g) dg \quad (3)$$

By solving the integral we get

$$P_I(k) = \frac{1}{N} \sum_{x \in \Omega} \sigma\left(\frac{I(x) - \mu_k}{L}\right) \Big|_{\mu_k + L/2}^{\mu_k - L/2} \quad (4)$$

The function $P_I(k)$ which provides the value of the k^{th} bin in a differentiable histogram can be rewritten as follows:

$$P_I(k) = \frac{1}{N} \sum_{x \in \Omega} \Pi_k(I(x)), \quad (5)$$

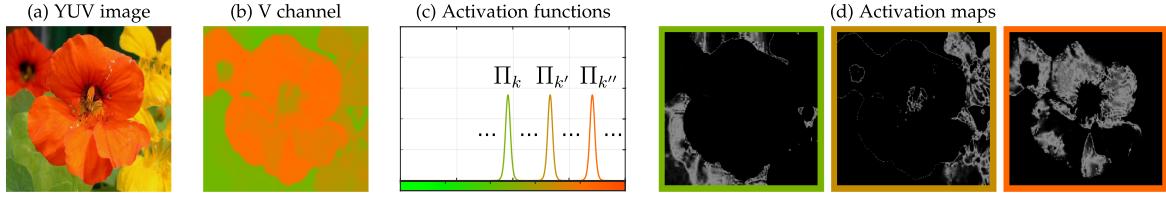


Fig. 3. *Activation functions and maps.* (b) One (out of three) color channel of the image on the left (a). (c) Three out of K activation functions. (d) Three out of K activation maps, each of which is generated by an application of the respective Π_k activation function to the output color channel shown in (b). Note that pixels with values closer to k have higher values in the k^{th} activation map.

where,

$$\Pi_k(z) \triangleq \sigma\left(\frac{z - \mu_k + L/2}{W}\right) - \sigma\left(\frac{z - \mu_k - L/2}{W}\right). \quad (6)$$

$\Pi_k(z)$ is a differentiable approximation of the Rect function. Fig. 3 illustrates the application of three (out of K) activation functions Π_k on a grayscale image. The resulting K channels are used for the construction of the corresponding gray level histogram. Specifically, the k^{th} histogram bin is obtained by a summation of the k^{th} channel values. The set of K channels can be viewed as smooth 1-hot approximations of the pixel values in a gray-level image. Note that the support of Π_k is over the gray-level range and unlike a convolutional kernel is not spatial. A differentiable histogram \mathbf{h} of a gray-level image I is defined as follows:

$$\mathbf{h} = \{\mu_k, P_I(k)\}_{k=0}^{K-1} \quad (7)$$

3) *Differentiable Joint Histogram Formulation:* The joint histogram of two gray-level images, each with K discrete gray levels, is a $K \times K$ matrix constructed such that its (k, l) entry counts the number of times that a pixel with a gray level value of k in one image corresponds (i.e., has the same image coordinates) to a pixel with a gray level value of l in the other. The joint gray-level density is obtained by normalizing the joint gray-level histogram. Consider two images $I_1, I_2 : \Omega \rightarrow [-1, 1]$ with continuous pixel values. Their joint gray-level density can be defined using multivariate KDE as follows:

$$\hat{f}_{I_1, I_2}(g_1, g_2) = \frac{1}{N} |\mathbf{W}|^{-1/2} \sum_{x \in \Omega} \mathcal{K}\left(\mathbf{W}^{-1/2}(\mathbf{I}(x) - \mathbf{g})\right) \quad (8)$$

where, $\mathbf{I}(x) = [I_1 \quad I_2]^T$, $\mathbf{g} = [g_1 \quad g_2]^T$, \mathbf{W} is the bandwidth (or smoothing) 2×2 matrix and $\mathcal{K}(\cdot, \cdot)$ is the symmetric 2D kernel function. As in the 1D case (2), we choose the kernel $\mathcal{K}(\cdot, \cdot)$ as the derivative of the logistic regression function $\sigma(z)$ for each of the two variables separately:

$$\mathcal{K}(z_1, z_2) = \sigma'(z_1)\sigma'(z_2) \quad (9)$$

We define the bandwidth matrix \mathbf{W} as $\begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix}$. We define the probability of corresponding pixels in I_1 and I_2 to belong to

the intensity intervals B_{k_1} and B_{k_2} , respectively, as follows:

$$\begin{aligned} P_{I_1, I_2}(k_1, k_2) &\triangleq \Pr(I_1(x) \in B_{k_1}, I_2(x) \in B_{k_2}) \\ &= \int_{B_{k_1}} \int_{B_{k_2}} \hat{f}_{I_1, I_2}(g_1, g_2) dg_1 dg_2 \end{aligned} \quad (10)$$

By calculating the double integral:

$$\begin{aligned} P_{I_1, I_2}(k_1, k_2) &= \frac{1}{N} \sum_{x \in \Omega} \sigma\left(\frac{I_1(x) - g_1}{W}\right) \Big|_{\mu_{k_1} + L/2}^{\mu_{k_1} - L/2} \\ &\quad \cdot \sigma\left(\frac{I_2(x) - g_2}{W}\right) \Big|_{\mu_{k_2} + L/2}^{\mu_{k_2} - L/2} \end{aligned} \quad (11)$$

By using the definition of Π_k from (6), we can express the value of the k_1, k_2 -th bin in the joint histogram as

$$P_{I_1, I_2}(k_1, k_2) = \frac{1}{N} \sum_{x \in \Omega} \Pi_{k_1}(I_1(x)) \Pi_{k_2}(I_2(x)) \quad (12)$$

This equation can be also written using matrix notation. We define a $K \times N$ matrix \mathbf{P}_j ($j = 1, 2$) where each of its K rows is a flatten activation map, generated from a gray level image I_j . A differentiable joint histogram \mathbf{J} of two images I_j , $j = 1, 2$ can be constructed via matrix multiplication as follows:

$$\mathbf{J}(I_1, I_2) = \frac{1}{N} \mathbf{P}_1 \mathbf{P}_2^T \quad (13)$$

4) Histogram Layers: 1D Histogram Layer.

A histogram layer of size K (one for each color channel) is constructed from the corresponding output image layer of the network generator. Specifically, the value of the k^{th} unit in a histogram layer is the sum of the entries (normalized by N) of the respective k^{th} activation map (5). An activation map k of a color channel l is obtained by the application of the k^{th} activation function to the l^{th} output image channel, see Figs. 3 and 4(a).

Joint Histogram Layer. After generating K activation maps for each of the output channels, we construct three matrices of size $K \times N$, by reshaping the $H \times W$ maps into $N \times 1$ vectors. The construction is performed by the HueNet joint histogram layers. Applying a similar process to the source image, we can now construct three joint histograms via three matrix multiplications (13), corresponding to the Y, U, and V channels. Fig. 4 illustrates the main ideas.

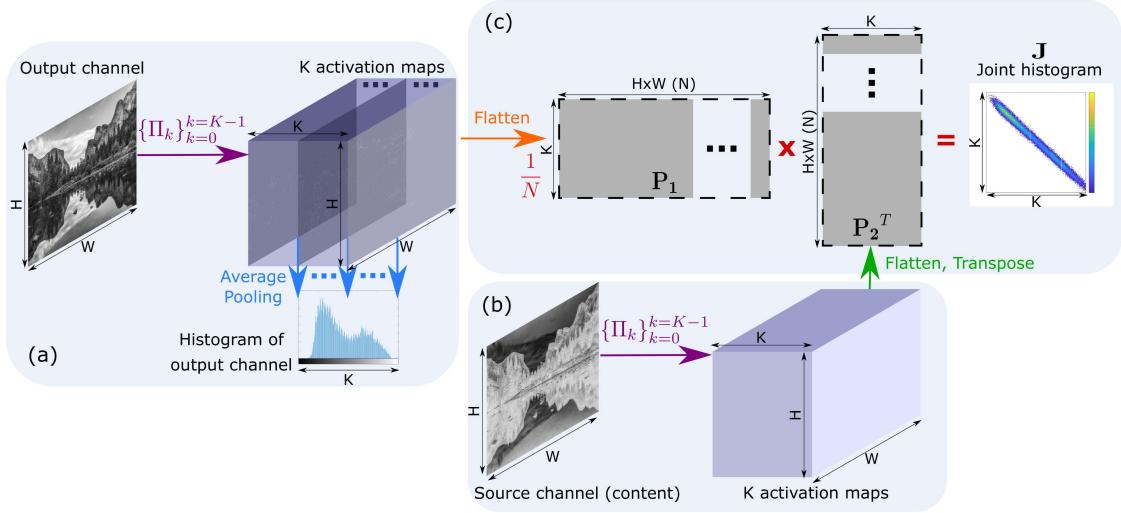


Fig. 4. *Histogram layers illustrated for one (out of three) output image channels.* Intensity and joint histograms are constructed using specified histogram layers of the HueNet. The weights of these layers are fixed but take part in the back-propagation process. (a) K activation maps ($H \times W$ matrices) are generated by the application of K activation functions $\{\Pi_k(\cdot)\}_{k=0}^{K-1}$ to the HueNet output image channel. Averaging the values of each of the K activation maps provides the respective histogram bins. (b) The histograms of the source image and the color reference image (not shown in this illustration) are obtained in a similar manner, i.e., via the construction of K activation maps and averaging each to form the respective histogram bin. (c) The MI loss is defined based on the joint histogram J of the source and the output images to constrain structural similarity. The joint histogram is constructed as part of the network in a differentiable manner using the activation maps. Specifically, activation maps of both the source and the output images are flatten to form two $K \times N$ matrices. Multiplication of these two matrices provides the joint histogram.

B. Loss functions

1) *Earth Mover's Distance:* Defining the distance between pairs of histograms as the MSE between their corresponding bins disregards the fact that nearby bins are similar. We use instead the EMD [31] (also known as the Wasserstein metric [32]) which is a cross-bin distance metric, to define the distance between two image histograms. Let \mathbf{h}_1 and \mathbf{h}_2 be the histograms of the images I_1 and I_2 , respectively. Note that when \mathbf{h}_1 and \mathbf{h}_2 have the same overall mass, the EMD is a true metric [31]. Moreover, when the compared histograms are also 1D, EMD has been shown to be equivalent to Mallows distance, which has a closed-form solution [33]. Werman et al. [34] showed that the EMD is equal to the ℓ_1 distance between the cumulative histograms (further details in the Appendix, available online). As in [35] we use the Euclidean distance for its faster convergence and easier gradient descent optimization [36], [37].

$$\text{EMD}^2(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=0}^{K-1} (\text{CDF}_i(\mathbf{h}_1) - \text{CDF}_i(\mathbf{h}_2))^2, \quad (14)$$

where $\text{CDF}_i(\mathbf{h}_j)$ is the i -th element of the cumulative density function of \mathbf{h}_j .

a) *Cyclic EMD:* This section describes the adaptation of the EMD measure for the case of a cyclic channel (the hue channel in the HSV colorspace). Werman et al. [34] showed that the EMD is equal to the ℓ_1 distance between the cumulative histograms. They also proved that matching two cyclic histograms by only examining cyclic permutations is optimal. Therefore, the cyclic $\mathcal{D}_{\text{EMD}_c^2}$ distance can be expressed as

$$\text{EMD}_{\text{Cyclic}}^2(\mathbf{h}_1, \mathbf{h}_2) = \min_{l=0, \dots, K-1} \text{EMD}^2(T(\mathbf{h}_1, l), T(\mathbf{h}_2, l)) \quad (15)$$

where the operator $T(\mathbf{h}_j, l)$ shifts by l all elements in \mathbf{h}_j under the modulo operation. Formally,

$$T(\mathbf{h}_j, l)[k] = \mathbf{h}_j[(k + l) \bmod K]. \quad (16)$$

In practice, we construct cyclic permutations of the constructed histogram by matrix multiplication of \mathbf{h}_j and circulant matrix (a special kind of Toeplitz matrix), to obtain the transformation described in the equation above.

2) *Mutual Information:* The MI of two images I_1 and I_2 can be formulated as follows:

$$\mathcal{I}(I_1, I_2) = H(I_1) - H(I_1|I_2) \quad (17)$$

where $H(I_1)$ is the entropy of I_1 and $H(I_1|I_2)$ is the conditional entropy of I_1 for a given I_2 . Maximizing the MI between the source and the output encourages their structural similarity. We assume that the entropy of the source image $H(I_1)$ is constant and minimizes the conditional entropy $H(I_1|I_2)$ rather than maximizing the MI as follows:

$$H(I_1|I_2) = - \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} P_{I_1, I_2}(k_1, k_2) \log \frac{P_{I_1, I_2}(k_1, k_2)}{P_{I_2}(k_2)} \quad (18)$$

where, P_{I_2} is the output image histogram, defined in (5), and P_{I_1, I_2} is the joint histogram discussed in Section II-A3.

3) *Adversarial Loss:* In addition to the EMD and MI loss functions we use an adversarial loss to encourage realism and natural assignment of the image colors thus ensuring, for example, greenish leaves and red petals and not the other way around. Let G denote the HueNet generator where D is its discriminator. For paired (colorization, edges \rightarrow photo) and unpaired (color transfer) tasks we use conditional and unconditional GAN loss functions, respectively, similar to [15]. The objective of the

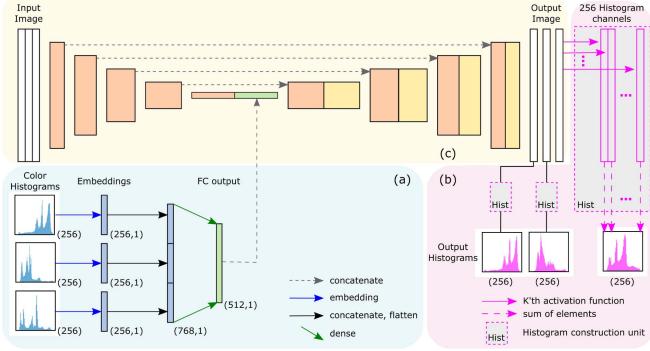


Fig. 5. The HueNet network is composed of an image generator (yellow) augmented by input (light blue) and output (pink) histogram layers. The input to the encoder part of the generator is either a gray-scale image (for image colorization), an edge map (for edge → photo), or a different color image (for color transfer). The color reference histograms are fed (one for each channel) into the embedding layers, followed by a fully connected layer and a concatenation with the code layer of the generator. The three output layers of the generator (which together compose the three channels of the synthesized output image) are used for the construction of 1D and 2D (joint) intensity histogram layers. Fig. 4 provides a detailed illustration of the histogram construction.

conditional GAN can be expressed as:

$$\arg \min_G \max_D \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] \quad (19)$$

where x is the input image, y is the desired output (real image), and z is noise in the form of dropout. We denote the HueNet adversarial loss function for all tasks by \mathcal{L}_{ADV} .

4) *Complete HueNet Loss*: We define \mathcal{L}_{EMD} , \mathcal{L}_{MI} as the normalized sums of the loss functions of each of the color output channels, according to (14), (18) respectively. The complete loss \mathcal{L} of the HueNet generator is a weighted sum of three loss functions:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{EMD}} + \beta \mathcal{L}_{\text{MI}} + \gamma \mathcal{L}_{\text{ADV}} \quad (20)$$

where \mathcal{L}_{EMD} , \mathcal{L}_{MI} , \mathcal{L}_{ADV} are the intensity similarity loss using EMD, the structural similarity loss using MI and the adversarial loss, respectively. The scalars α, β, γ are hyperparameters that were tuned using the validation sets.

C. HueNet Network Architecture

Fig. 5 illustrates the generator architecture as well as the augmented input and output histogram layers. The HueNet network architecture is composed of an image generator (a modified version of the U-Net [38]) augmented by input and output histogram layers. The input to the encoder is a content image. In addition, channel reference histograms are fed (each separately) into the embedding layers, followed by a fully connected layer and a concatenation with the code layer of the generator. Embedding the reference histogram within the network generator allows us to control the color distribution of the output image. The three output channels of the generator (which together composed the synthesized output image) are used for the construction of 1D and 2D intensity histogram layers. The histogram construction is illustrated in Fig. 4. Similar to [15], we use the convolutional ‘PatchGAN’ classifier [39] as a discriminator for the construction of the adversarial loss.



Fig. 6. *Color transfer results on Oxford flowers102*. Left-most column: Input images, Second column: color reference images (only their intensity histograms are provided to the network). Third column: HuNet results obtained for painting the input image with the color-reference image. Fourth column: HuNet results without the MI loss \mathcal{L}_{MI} . Fifth column: color transfer results by Reinhard [7]. The user is advised to zoom-in to better evaluate the visual results.

D. Implementation Details

To optimize our networks, we alternate between one gradient descent step on the discriminator and one step on the generator. We use stochastic gradient descent (SGD) minibatch and apply the Adam solver [40], with a learning rate of 0.0002, and momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. For histograms construction we use $K = 256$ bins, $W = L/2.5$. The hyperparameters of the HueNet loss function (20) are $\alpha = 100$ for color transfer and edges → photo; $\alpha = 20$ for image colorization; $\beta = 1$, $\gamma = 1$ for all tasks.

III. EXPERIMENTAL RESULTS

We demonstrate the strengths of the HueNet for different paired and unpaired image-to-image translation tasks using several publicly available datasets. These include color transfer for the flowers [26] and the indoor-outdoor scenes [24] datasets (Section III-A), exemplar-based image colorization for the summer-winter dataset [25] (Section III-B), and color-reference edges → photo for the shoes [27] and bags [28] datasets (Section III-C). More results are available in Figs. 2, 3, and 4 in the Appendix, available online. Note we present non-standard nuances for the image colorization and edges → photo tasks by constraining the color distributions of the output images to resemble pre-selected color histograms extracted from some color reference images (not provided as input) or a color pallet.

A. Color Transfer

To demonstrate the color transfer task we used two different datasets, namely the Oxford 102 Category Flower (Oxford flowers102) dataset [26] and the indoor-outdoor scene dataset [24]. The Oxford 102 Flower dataset consists of 8189 images, which



Fig. 7. *HueNet* color transfer results for the flowers dataset. The results are presented in two panels. Left-most columns: Input images, Second columns: color reference images (only their intensity histogram are provided to the network). Third columns: HuNet results obtained for painting the input images with the colors of the respective color-reference images. Right-most columns: HuNet results without the MI loss (\mathcal{L}_{MI}). The reader is advised to zoom-in to better evaluate the visual results.

were randomly divided into 7370 and 819 images for training and test, respectively. The indoor-outdoor scenes dataset is composed of only 120 images. For fair comparison with [24] and [23] we kept aside the image pairs used there for testing, and used randomly selected pairs (other than these) for training.

Qualitative Comparison. Figs. 6 and 7 present color transferred images of the flowers dataset obtained with and without the MI loss, to demonstrate its contribution to the overall image quality and similarity to the source. Fig. 6 also presents a visual comparison to Reinhard [7]. The color transfer results of the HueNet as well as those obtained by [24] and [23] for the indoor-outdoor scenes dataset are presented in Fig. 8. Since the problem addressed is unpaired the results are qualitative and subjective. As can be seen there is no specific method that outperforms the others. We also note that the proposed color-transfer is fully unsupervised and does not use the segmentation masks of the input images. Moreover, we could have used transfer-learning and pre-train the HueNet with any pair of color images. However, for fair comparisons we restricted our training to the indoor-outdoor scenes dataset.

Perceptual Realism. To evaluate the 'realism' of our color transfer results we devised a questionnaire for human observers, in which we presented either real (reference) or our color transferred (output) images in random order. The participants

TABLE I
CONFUSION MATRIX OF THE QUESTIONNAIRE RESULTS PRESENTING THE AVERAGE PERCENTAGE (%) OF PARTICIPANTS WHO MARKED THE REAL OR THE SYNTHESIZED IMAGE AS 'REAL' OR 'FAKE'

	Real	Fake
Real image	65.8	34.2
Output (color transfer)	51.9	48.1

were asked to mark each image a 'Real' or 'Fake'. We used 24 images, of which 12 were real and 12 were the HueNet color transfer outputs.

Table I presents the confusion matrix of the hit/miss percentages for 100 participants. As shown in the Table, the HueNet color transfer results misled (on average) the questionnaire participants on roughly half of the cases. The questionnaire can be accessed at <https://forms.gle/mTyd591Aq8u3XjWF6>.

Structural Similarity. The color transfer problem is an unpaired learning task. Therefore, a comparison of the network output to a real image with similar colors and structure is impossible. Instead, we quantitatively demonstrate the structural similarity between the output and the source image by calculating their normalized MI [41]. For comparison, we trained the HueNet without the MI loss. The results shown in Table II imply that

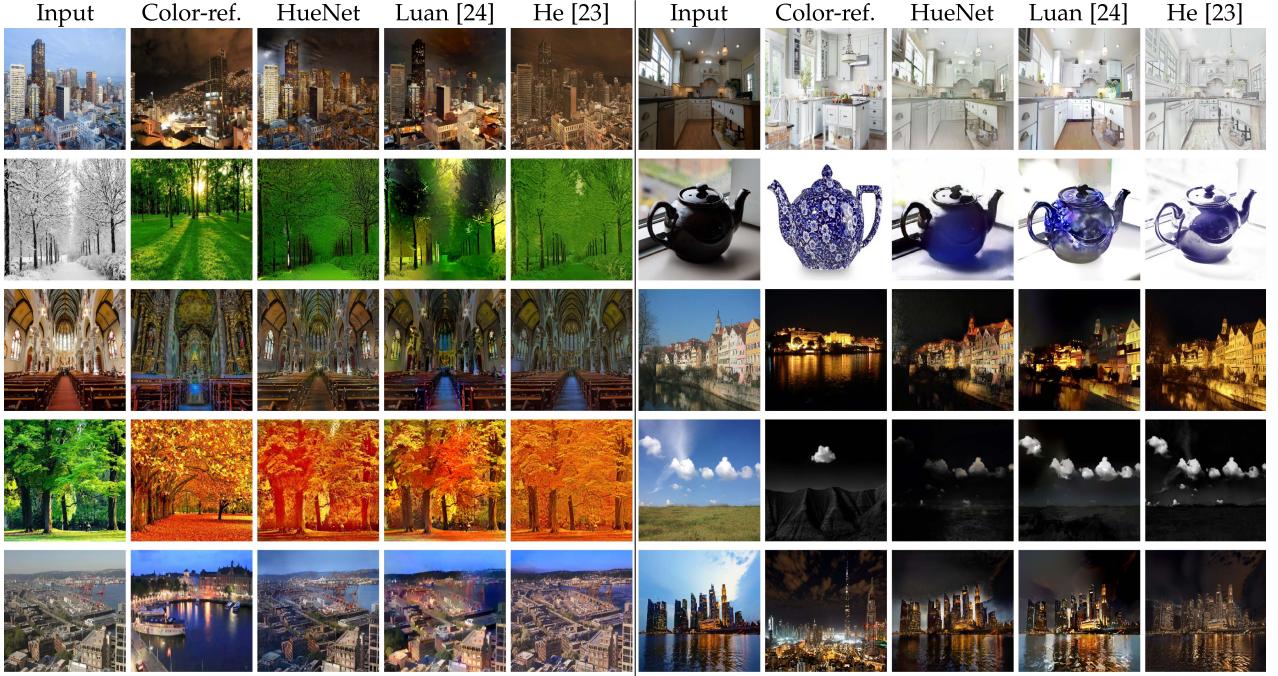


Fig. 8. *HueNet* color transfer results for the indoor-outdoor scenes dataset. The results are presented in two panels. Left-most columns: Input images, Second columns: color reference images (only their intensity histogram are provided to the network). Third columns: color-transfer HuNet results. Comparisons to Luan et al. [24] and to He et al. [23] are shown in the fourth and the fifth columns, respectively. Further results are shown in Fig. 3 of the Appendix, available online. The user is advised to zoom-in to better evaluate the visual results.

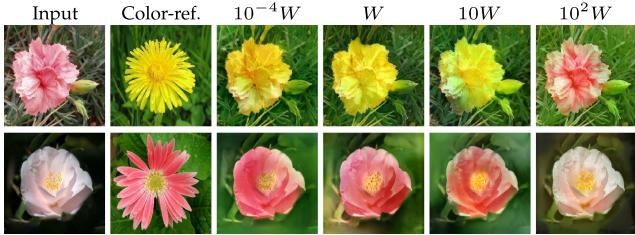


Fig. 9. An ablation study for the kernel bandwidth selection. We set the KDE bandwidth $W = L/2.5$ for all tasks. To support this choice we tested different values of bandwidth for the color transfer task.

TABLE II
STRUCTURAL SIMILARITY BETWEEN THE NETWORK OUTPUTS AND THE CORRESPONDING INPUT IMAGES FOR THE *COLOR TRANSFER* PROBLEM IS DEMONSTRATED BY CALCULATING THEIR NORMALIZED MI METRIC. THE TABLE COMPARES THE NORMALIZED MI MEASURES CALCULATED FOR THE HUENET WITH AND WITHOUT THE MI LOSS FOR EACH OF THE YUV CHANNELS

	Y	U	V
HueNet	0.178	0.141	0.167
w/o \mathcal{L}_{MI}	0.066	0.035	0.044

the output images obtained at test time for the network trained without MI loss were less consistent with the input in terms of their structure. The contribution of the MI loss is also depicted in columns 3-4 of Figs. 6 and 4 in the Appendix, available online.

Failure Modes. The MI loss facilitates color coherency in semantically meaningful regions without requiring an image segmentation. However, there are a few cases in which the

desired color-consistency is not completely preserved. For example, the purple-green background flower in Fig. 1 or the yellow-gray car shown in the right panel, third row of Fig. 3 in the Appendix, available online. We also note that in some cases when the luminance (Y channel) of the color reference image is high the output image may contain a shed that is not seen in the reference, for example the purplish appearance of the flower shown in row 4 of Fig. 6.

KDE Bandwidth Testing. We set the KDE bandwidth to $W = L/2.5$, where $L = \frac{2}{K}$ is the length of the sub-interval (the entire interval on which the constructed, differentiable histogram is defined was set to $[-1, 1]$). To support this choice we performed an ablation study in which we trained our framework with different values of W . We observed that the HueNet is robust for a wide range of bandwidths. However, as shown in Fig. 9 the performances degrade for extreme low values of W , possibly since this significantly affect the functions' derivatives. A wider bandwidth implies a wider range of colors. Specifically, when the bandwidth is set to $10^2 W$ it covers the entire histogram range. In that case the histogram-based loss functions are no longer effective and the output is very similar to the input regardless of the color-reference image.

Cross-Dataset Testing. We also tested the HueNet for color transfer of the flower images using reference color histograms of images from the summer-winter Yosemite dataset, as shown in Fig. 4 in the Appendix, available online. Note that the HueNet produces high-quality output images, even though the color distribution statistics of the summer-winter Yosemite dataset are quite different from those of the Oxford flowers102 dataset on which it was trained.

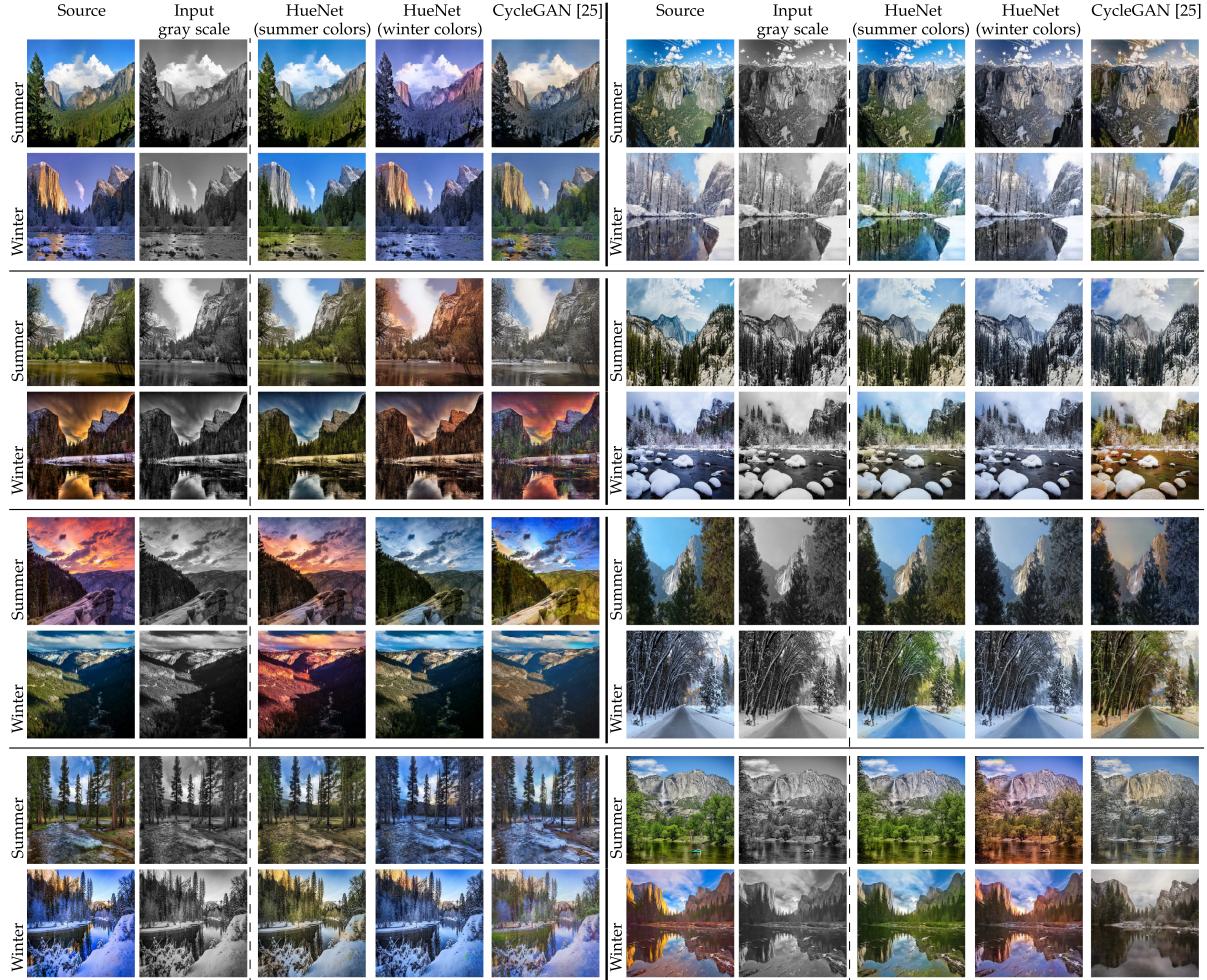


Fig. 10. *Image colorization with target ‘summer’ and ‘winter’ color distributions.* The first columns of both panels present the source images which are either summer (odd rows) or winter (even rows) images. The HueNet input was composed of a gray-scale image (second column of each panel) generated from the source, and a color histogram of either a color reference image or the source image. The color reference images of the odd rows are the source images of the even rows and vice versa. The third and the fourth columns of both panels present the HueNet outputs; i.e., colorization of the input images with the colors of the source and the color reference images, respectively. A comparison of the source images (the first column of each panel) to the images generated using the gray-scale inputs and the color histograms of the source images (the respective third column) enables further visual assessment. These images were generated using the gray-scale inputs and the color histograms of the source images. The CycleGAN [25] results for the summer-to-winter domain transfer are presented in the right-most column of each panel. The user is advised to zoom-in to better evaluate the visual results.



Fig. 11. *Edges → shoe ablation study results.* Left-most column: The original images used for generating the edge maps (HueNet input) shown in the Second column. Third column: The proposed HueNet output. Fourth-sixth columns: The network output without the EMD loss (fourth col.) the MI loss (fifth column) or the adversarial loss (sixth col.).

B. Image Colorization

To assess image colorization we used the summer/winter Yosemite dataset, prepared by [25] with Flickr API. We split the train and test images as in [25]. The HueNet input was a gray scale image. Its output was the same image colored as though it were taken in either the summer or the winter.

Fig. 10 presents the visual image colorization results on the summer/winter Yosemite dataset [25] as well as a comparison to CycleGAN [25]. The qualitative results show the ability of the HueNet to synthesize realistic and more colorful images compared to CycleGAN. Note that the results obtained by CycleGAN are exclusively for a summer-to-winter (or winter-to-summer) domain transfer. In other words, the CycleGAN input is the source image and not a gray-scale image. Moreover, in contrast to the HueNet, when CycleGAN is trained to generate



Fig. 12. *Edges → shoe and edges → bag* results are shown in the left and the right panels, respectively. The original image (first column of each panel) was used to generate the edge map (second column of each panel). The input to HueNet was composed of an edge map and a color histogram of either the color reference image (third column of each panel) or the source image. Note that the original image (source) itself was not provided as input. The fourth and the fifth columns of each panel present the HueNet synthesized images with the colors of the reference and the source images, respectively. Comparison of each of the original, source images to the corresponding synthesized image using the source edge map and its color distribution is shown here for sanity check. The right-most columns of each panel present the corresponding Pix2Pix [15] output. Much more edges → shoe results with a reference color pallet are shown in Fig. 2 in the Appendix, available online. The reader is advised to zoom-in to better evaluate the visual results.

summer images it cannot generate winter images (and vice versa) at test time .

C. Edges → Photo

We used two different datasets from [27] and [28] to demonstrate the edges → shoe and edges → bag problems. We divided the datasets into training and test sets as in [15]. In training, we used the source images for the reference color distribution and to calculate the MI loss. We tested the network by synthesizing color images based on the same edge map but with different pre-selected color histograms.

Quantitative Comparison.

Table III presents the quantitative results obtained by the HueNet for the edges → photo task and their comparison to Pix2Pix. In this experiment, the output images were synthesized with the colors of the original source images, from which the input edge maps were generated. The similarity between the source and the output images was evaluated by the MSE and the Fréchet inception distances (FID) [42]. Our method outperformed the Pix2Pix for both measures.

Ablation Studies.

We ran ablation studies to demonstrate the contribution of each of the HueNet loss functions to the overall output image

TABLE III

QUANTITATIVE COMPARISON OF EDGES → PHOTO TASK PERFORMANCE. IN THIS EXPERIMENT, THE SOURCE IMAGES WERE ALSO USED AS COLOR REFERENCE IMAGES. THE MSE AND THE FID MEASURES WERE CALCULATED FOR SOURCE-OUTPUT IMAGE PAIRS. OUR METHOD OUTPERFORMED THE PIX2PIX METHOD SINCE ITS OUTPUT COLORS CANNOT BE CONTROLLED.

Method	edges → shoe		edges → bag	
	MSE	FID	MSE	FID
Pix2Pix [15]	.034 (.017)	107.27	.059 (.023)	93.54
HueNet	.018 (.01)	66.9	.041 (.02)	75.05

TABLE IV

QUANTITATIVE ABLATION STUDY ON THE EDGES → SHOE DATASET. THE SCORES PRESENT THE MSE (STD) BETWEEN THE SOURCE AND THE GENERATED IMAGES OBTAINED USING EITHER THE PROPOSED HUENET OR THE HUENET WITHOUT EITHER OF ITS LOSS FUNCTIONS

HueNet	w/o \mathcal{L}_{EMD}	w/o \mathcal{L}_{MI}	w/o \mathcal{L}_{ADV}
MSE	.018 (.010)	.053 (.013)	.024 (.012)

quality. Fig. 11 exemplifies the influence of removing either of these loss functions from the training for the edges → photo problem. The figure shows that MI and ADV alone (setting $\lambda_{\text{ADV}} = 0$ in (20)) are not sufficient to produce realistic, detailed images. Using only the MI and ADV loss functions without the EMD loss (setting $\lambda_{\text{EMD}} = 0$ in (20)), does not allow the network produce output images with the colors of the reference. Finally, training exclusively with the ADV and EMD loss functions (without the MI loss) introduces visual artifacts. The MI loss is important for preserving the structural similarity to the source image (regions with the same color). This observation is quantitatively supported in Table IV.

Qualitative Comparison.

Fig. 12 visually presents the edges → shoe (left panel) and edges → bag (right panel) results as well as a comparison to Pix2Pix. For each panel, the left-most columns present the original images that were used to generate the edge maps (second columns) provided as input to the HueNet. The third columns in each panel present the color reference images. The fourth and fifth columns present the HueNet output painted either with the source colors (fourth columns) or with the color reference colors (fifth columns). Painting the output images with the colors of the source was done deliberately to enable the reader to visually compare between the HueNet synthesized images and the real images. Note that the colors of the output images are determined by the color histograms of the images and not by the color images themselves. The right-most columns present Pix2Pix results.

IV. CONCLUSION

We presented a novel generative deep learning framework, dubbed the HueNet, and demonstrated its strengths on different paired and unpaired image-to-image translation tasks, namely, color transfer, exemplar-based image colorization, and edges → photo with color reference. A key contribution of the HueNet is its innovative architecture which enables the differentiable construction of intensity (1D) and joint (2D) histogram-based

loss functions. These loss functions constrain the color distribution of the structural appearance of the synthesized image. Specifically, the EMD loss is applied to the output image and a color-reference histograms whereas the MI loss is applied to the joint histogram of the output and the content-reference image. The results presented for all applications are realistic and can confuse human observers, as we found on the perceptual realism test (Table I) and visual comparisons. Moreover, since the tools we developed are not tailored to specific applications they can be successfully applied to other computer vision problems. Possible future applications include, multi-modal image registration, image-to-image modality transfer (e.g., CT → MRI, heat → RGB), photo enhancement, and changing illumination conditions.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Hoboken, NJ, USA: Prentice Hall, 2008.
- [4] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed., Berlin, Germany: Springer, 2010.
- [5] Z. Wang, H. Li, W. Ouyang, and X. Wang, “Learnable histogram: Statistical context features for deep neural networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 246–262.
- [6] S. Nam and S. Joo Kim, “Modelling the scene dependent imaging in cameras with a deep neural network,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1717–1725.
- [7] E. Reinhard, M. Adhikmin, B. Gooch, and P. Shirley, “Color transfer between images,” *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, Jul./Aug. 2001.
- [8] L. Neumann and A. Neumann, “Color style transfer techniques using hue, lightness and saturation histogram matching,” in *Computational Aesthetics*. Princeton, NJ, USA: Citeseer, 2005, pp. 111–122.
- [9] F. Pitie, A. Kokaram, and R. Dahyot, “N-dimensional probability density function transfer and its application to color transfer,” in *Proc. IEEE 10th Int. Conf. Comput. Vis.*, 2005, pp. 1434–1439.
- [10] Z. Shu, S. Hadap, E. Shechtman, K. Sunkavalli, S. Paris, and D. Samaras, “Portrait lighting transfer using a mass transport approach,” *ACM Trans. Graph.*, vol. 37, pp. 1–15, 2018.
- [11] P. Viola and W. M. Wells III, “Alignment by maximization of mutual information,” *Int. J. Comput. Vis.*, vol. 24, no. 2, pp. 137–154, 1997.
- [12] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
- [13] M. Bharati, J. Liu, and J. MacGregor, “Image texture analysis: Methods and comparisons,” *Chemometrics Intell. Lab. Syst.*, vol. 72, no. 1, pp. 57–71, 2004.
- [14] I. Goodfellow et al., “Generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [15] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5967–5976.
- [16] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 649–666.
- [17] G. Larsson, M. Maire, and G. Shakhnarovich, “Learning representations for automatic colorization,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 577–593.
- [18] M. Afifi, M. Brubaker, and M. Brown, “HistoGAN: Controlling colors of GAN-generated and real images via color histograms,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7937–7946.
- [19] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, “Deep exemplar-based colorization,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–16, 2018.
- [20] C. Xiao et al., “Example-based colourisation via dense encoding pyramids,” *Comput. Graph. Forum*, vol. 39, no. 1, pp. 20–33, 2020.

- [21] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5400–5409.
- [22] R. Zhang et al., "Real-time user-guided image colorization with learned deep priors," 2017, *arXiv: 1705.02999*.
- [23] M. He, J. Liao, D. Chen, L. Yuan, and P. V. Sander, "Progressive color transfer with dense semantic correspondences," *ACM Trans. Graph.*, vol. 38, no. 2, pp. 1–18, 2019.
- [24] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6997–7005.
- [25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.
- [26] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. Indian Conf. Comput. Vis. Graph. Image Process.*, 2008, pp. 722–729.
- [27] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 192–199.
- [28] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.
- [29] R. A. Davis, K.-S. Lii, and D. N. Politis, "Remarks on some nonparametric estimates of a density function," in *Selected Works of Murray Rosenblatt*. Berlin, Germany: Springer, 2011, pp. 95–100.
- [30] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [31] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.
- [32] R. L. Dobrushin, "Prescribing a system of random variables by conditional distributions," *Theory Probability Appl.*, vol. 15, no. 3, pp. 458–486, 1970.
- [33] E. Levina and P. Bickel, "The earth mover's distance is the mallows distance: Some insights from statistics," in *Proc. IEEE 8th Int. Conf. Comput. Vis.*, 2001, pp. 251–256.
- [34] M. Werman, S. Peleg, and A. Rosenfeld, "A distance metric for multi-dimensional histograms," *Comput. Vis. Graph. Image Process.*, vol. 32, no. 3, pp. 328–336, 1985.
- [35] L. Hou, C. Yu, and D. Samaras, "Squared earth mover's distance-based loss for training deep neural networks," 2016. [Online]. Available: <http://arxiv.org/abs/1611.05916>
- [36] D. G. Luenberger et al., *Linear and Nonlinear Programming*. Berlin, Germany: Springer, 1984, vol. 2.
- [37] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for ℓ_1 -regularized loss minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1865–1892, 2011.
- [38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.
- [39] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 702–716.
- [40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [41] C. Studholme, D. Hawkes, and D. Hill, "Normalized entropy measure for multimodality image alignment," in *Medical Imaging 1998: Image Processing*, K. M. Hanson Ed., vol. 3338, Bellingham, WA, USA: SPIE, 1998, pp. 132–143.
- [42] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.



Mor Avi-Aharon received the BSc and MSc degrees in electrical and computer engineering (magna cum laude) from Ben-Gurion University.



Assaf Arbelle received the BSc (magna cum laude), MSc (magna cum laude), and PhD degrees in electrical and computer engineering from Ben-Gurion University, Israel, in 2013, 2016, and 2020, respectively.



Tammy Riklin Raviv (Member, IEEE) received the BSc degree in physics, the MSc degree in computer science (both magna cum laude) from the Hebrew University of Jerusalem, and the PhD degree from the School of Electrical Engineering of Tel-Aviv University. She is an associate professor with the School of Electrical and Computer Engineering, Ben-Gurion University. During 2010–2012 she was a research fellow with Harvard Medical School and the Broad Institute of MIT and Harvard. In 2008–2010 she was a post-doctorate associate with the Computer Science and Artificial Intelligence laboratory (CSAIL), MIT.