

# Deep Exemplar-based Colorization\*

Mingming He<sup>\*1</sup>, Dongdong Chen<sup>\*2</sup> †, Jing Liao<sup>3</sup>, Pedro V. Sander<sup>1</sup>, and Lu Yuan<sup>3</sup>  
<sup>1</sup>Hong Kong UST, <sup>2</sup>University of Science and Technology of China, <sup>3</sup>Microsoft Research



**Figure 1:** Colorization results of black-and-white photographs. Our method provides the capability of generating multiple plausible colorizations by giving different references. Input images (from left to right, top to bottom): Leroy Skalstad/pixabay, Peter van der Slujs/wikimedia, Bollywood Hungama/wikimedia, Lorri Lang/pixabay, Aamir Mohd Khan/pixabay, Official White House Photographer/wikimedia, Anonymous/wikimedia and K. Krallis/wikimedia.

## Abstract

We propose the first deep learning approach for exemplar-based local colorization. Given a reference color image, our convolutional neural network directly maps a grayscale image to an output colorized image. Rather than using hand-crafted rules as in traditional exemplar-based methods, our end-to-end colorization network learns how to *select*, *propagate*, and *predict* colors from the large-scale data. The approach performs robustly and generalizes well even when using reference images that are unrelated to the input grayscale image. More importantly, as opposed to other learning-based colorization methods, our network allows the user to achieve customizable results by simply feeding different references. In order to further reduce manual effort in selecting the references, the system automatically recommends references with our proposed image retrieval algorithm, which considers both semantic and luminance information. The colorization can be performed fully automatically by simply picking the top reference suggestion. Our approach is validated through a user study and favorable quantitative comparisons to the-state-of-the-art methods. Furthermore, our approach can be naturally extended to video colorization. Our code and models will be freely available for public use.

## 1 Introduction

The aim of image colorization is to add colors to a gray image such that the colorized image is perceptually meaningful and visually appealing. The problem is ill-conditioned and inherently ambiguous since there are potentially many colors that can be assigned to the gray pixels of an input image (*e.g.*, leaves may be colored in green, yellow, or brown). Hence, there is no unique correct solution and human intervention often plays an important role in the colorization process.

Manual information to guide the colorization is generally provided in one of two forms: user-guided scribbles or a sample reference

image. In the first paradigm [Levin et al. 2004; Yatziv and Sapiro 2006; Huang et al. 2005; Luan et al. 2007; Qu et al. 2006], the manual effort involved in placing the scribbles and the palette of colors must be chosen carefully in order to achieve a convincing result. This often requires both experience and a good sense of aesthetics, thus making it challenging for an untrained user. In the second paradigm [Welsh et al. 2002; Irony et al. 2005; Tai et al. 2005; Charpiat et al. 2008; Liu et al. 2008; Chia et al. 2011; Gupta et al. 2012; Bugeau et al. 2014], a color reference image similar to the grayscale image is given to facilitate the process. First, correspondence is established, and then colors are propagated from the most reliable correspondences. However, the quality of the result depends heavily on the choice of reference. Intensity disparities between the reference and the target caused by lighting, viewpoint, and content dissimilarity can mislead the colorization algorithm.

A more reliable solution is to leverage a huge reference image database to search for the most similar image patch/pixel for colorization. Recently, deep learning techniques have achieved impressive results in modeling large-scale data. Image colorization is formulated as a regression problem and deep neural networks are used to directly solve it [Cheng et al. 2015; Deshpande et al. 2015; Larsson et al. 2016; Iizuka et al. 2016; Zhang et al. 2016; Isola et al. 2017; Zhang et al. 2017]. These methods can colorize a new photo fully automatically without requiring any scribbles or reference. Unfortunately, none of these methods allow multi-modal colorization [Charpiat et al. 2008]. By learning from the data, their models mainly use the dominant colors they have learned, hindering any kind of user controllability. Another drawback is that it must be trained on a very large reference image database containing all potential objects.

More recent works attempt to achieve the best of both worlds: controllability from interaction and robustness from learning. Zhang et al. [2017] and Sangkloy et al. [2016] add manual hints in the form of color points or strokes to the deep neural network in order to suggest possibly desired colors for the scribbles provided by users. This greatly facilitates traditional scribble-based interactions and achieves impressive results with more natural colors learned from the large-scale data. However, the scribbles are still essential for achieving high quality results, so a certain amount of trial-and-error is still involved.

\*Supplemental material: [http://www.dongdongchen.bid/supp/deep\\_exam\\_colorization/index.html](http://www.dongdongchen.bid/supp/deep_exam_colorization/index.html)

†\* indicates equal contribution. This work was done when Mingming He and Dongdong Chen were interns at Microsoft Research Asia.



**Figure 2:** Our goal is to selectively propagate the correct reference colors (indicated by the dots) for the relevant patches/pixels, and predict natural colors learned from the large-scale data when no appropriate matching region is available in the reference (indicated by the region outlined in red). Input images (from left to right): Julian Fong/flickr and Ernest McGay, Jr/flickr.

In this paper, we suggest another type of hybrid solution. We propose the first deep learning approach for exemplar-based local colorization. Compared with existing colorization networks [Cheng et al. 2015; Iizuka et al. 2016; Zhang et al. 2016], our network allows control over the output colorization by simply choosing different references. As shown in Fig. 1, the reference can be similar or dissimilar to the target, but we can always obtain plausible colors in the results, which are visually faithful to the references and perceptually meaningful.

To achieve this goal, we present the first convolutional neural network (CNN) to directly select, propagate and predict colors from an aligned reference for a gray-scale image. Our approach is qualitatively superior to existing exemplar-based methods. The success comes from two novel sub-networks in our exemplar-based colorization framework.

First, the *Similarity sub-net* is a pre-processing step which provides the input of the end-to-end colorization network. It measures the semantic similarity between the reference and the target using a VGG-19 network pre-trained on the gray-scale image object recognition task. It provides a more robust and reliable similarity metric to varying semantic image appearances than previous metrics based on low-level features.

Then, the *Colorization sub-net* provides a more general colorization solution for either similar or dissimilar patch/pixel pairs. It employs multi-task learning to train two different branches, which share the same network and weight but are associated with two different loss functions: 1) *Chrominance loss*, which encourages the network to selectively propagate the correct reference colors for relevant patch/pixel, satisfying chrominance consistency; 2) *Perceptual loss*, which enforces a close match between the result and the true color image of high-level feature representations. This ensures a proper colorization learned from the large-scale data even in cases where there is no appropriate matching region in the reference (see Fig. 2). Therefore, our method can greatly loosen restrictive requirements on a good reference selection as required in other exemplar-based methods.

To guide the user towards efficient reference selection, the system recommends the most likely reference based on a proposed image retrieval algorithm. It leverages both high-level semantic information and low-level luminance statistics to search for the most similar images in the ImageNet dataset [Russakovsky et al. 2015]. With the help of this recommendation, our approach can serve as a fully automatic colorization system. The experiments demonstrate that our automatic colorization outperforms existing automatic methods quantitatively and qualitatively, and even produces comparably high quality results to the-state-of-the-art interactive methods [Zhang et al. 2017; Sangkloy et al. 2016]. Our approach can also be extended to video colorization.

Our contributions are as follows: (1) The first deep learning ap-

proach for exemplar-based colorization, which allows controllability and is robust to reference selection. (2) A novel end-to-end double-branch network architecture which jointly learns faithful local colorization to a meaningful reference and plausible color prediction when a reliable reference is unavailable. (3) A reference image retrieval algorithm for reference recommendation, with which we can also attain a fully automatic colorization. (4) A method capable of transferability to unnatural images, even though the network is trained purely on a natural image dataset. (5) An extension to video colorization.

## 2 Related work

Next, we provide an overview of the major related works of each of the major algorithm categories.

### 2.1 Scribble-based colorization

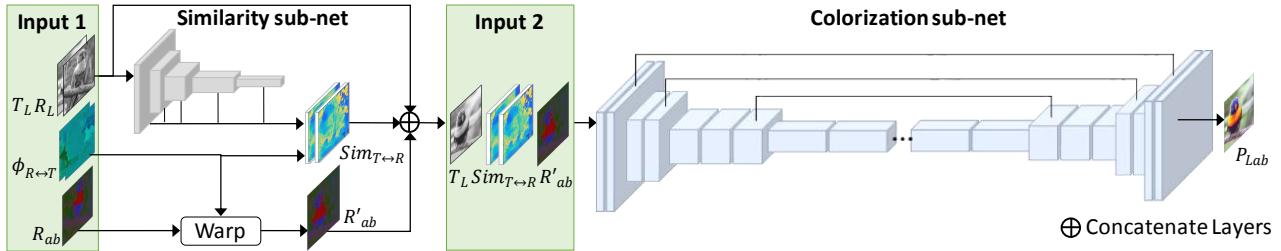
These methods focus on propagating local user hints, for instance, color points or strokes, to the entire gray-scale image. The color propagation is based on some low-level similarity metrics. The pioneering work of Levin et al. [2004] assumed that adjacent pixels with similar luminance should have similar color, and then solved a Markov Random Field for propagating sparse scribble colors. Further advances extended similarity to textures [Qu et al. 2006; Luan et al. 2007], intrinsic distance [Yatziv and Sapiro 2006], and exploited edges to reduce color bleeding [Huang et al. 2005]. The common drawback of such methods is intensive manual work and professional skills for providing good scribbles.

### 2.2 Example-based colorization

These methods provide a more intuitive way to reduce extensive user effort by feeding a very similar reference to the input grayscale image. The earliest work [Welsh et al. 2002] transferred colors by matching global color statistics, similar to Reinhard et al. [2001]. The approach yielded unsatisfactory results in many cases since it ignored spatial pixel information. For more accurate local transfer, different correspondence techniques are considered, including segmented region level [Irony et al. 2005; Tai et al. 2005; Charpiat et al. 2008], super-pixel level [Gupta et al. 2012; Chia et al. 2011], and pixel level [Liu et al. 2008; Bugeau et al. 2014]. However, finding low-level feature correspondences (e.g., SIFT, Gabor wavelet) with hand-crafted similarity metrics is susceptible to error in situations with significant intensity and content variation. Recently two works utilize deep features extracted from a pre-trained VGG-19 network for reliable matching between images that are semantically-related but visually different, and then leverage it to style transfer [Liao et al. 2017] and color transfer [He et al. 2017]. However, all of these exemplar-based methods have to rely on finding a good reference, which is still an obstacle for users, even when some semi-automatic retrieval methods [Liu et al. 2008; Chia et al. 2011] are used. By contrast, our approach is robust to any given reference thanks to the capability of our deep network to learn natural color distributions from large-scale image data.

### 2.3 Learning-based colorization

Several techniques rely entirely on learning to produce the colorization result. Deshpande et al. [2015] defined colorization as a linear system and learned its parameters. Cheng et al. [2015] concatenated several pre-defined features and fed them into a three-layer fully connected neural network. Recently, some end-to-end learning approaches [Larsson et al. 2016; Iizuka et al. 2016; Zhang et al. 2016; Isola et al. 2017] leveraged CNN to automatically ex-



**Figure 3:** System pipeline (inference stage). The system consists of two sub-networks. The Similarity sub-net works as a pre-processing step using Input 1 which includes two luminance channels  $T_L$  and  $R_L$  from the target and the reference respectively, bidirectional mapping functions  $\phi_{R \leftrightarrow T}$  and two chrominance channels  $R_{ab}$  from the reference. It computes the bidirectional similarity maps  $sim_{T \leftrightarrow R}$  and the aligned reference chrominance  $R'_{ab}$ , which, along with  $T_L$ , form Input 2 for the Colorization sub-net. The Colorization sub-net is an end-to-end CNN to predict the chrominance channels of the target, which are then combined with  $T_L$  to generate the final colorized result  $P_{Lab}$ .

tract features and predict the color result. The key difference in those networks is the loss function (*e.g.*, image reconstruction  $L_2$  loss [Iizuka et al. 2016], classification loss [Larsson et al. 2016; Zhang et al. 2016], and  $L_1 + GAN$  loss for considering the multi-modal colorization [Isola et al. 2017]). All of these networks are learned from large-scale data and do not require any user intervention. However, they only produce a single plausible result for each input, even though colorization is intrinsically an ill-posed problem with multi-modal uncertainty [Charpiat et al. 2008].

## 2.4 Hybrid colorization

To achieve desirable color results, Zhang et al. [2017] and Sangkloy et al. [2016] proposed a hybrid framework that inherits the controllability from scribble-based methods and the robustness from learning-based methods. Zhang et al. [2017] uses provided color points while Sangkloy et al. [2016] adopts strokes. Instead, we incorporate the reference rather than user-guided points or strokes into the colorization network, since we believe that giving a similar color example is a more intuitive way for untrained users. Furthermore, the reference selection can be achieved automatically using our image retrieval system.

## 3 Exemplar-based Colorization Network

Our goal is to colorize a target grayscale image based on a color reference image. More specifically, we aim to apply a reference color to the target where there is semantically-related content, and fall back to a plausible colorization for the objects or regions with no related content in the reference. To achieve this goal, we address two major challenges.

First, it is difficult to measure the semantic relationship between the reference and the target, especially given that the reference is in color while the target is a grayscale image. To solve this problem, we use a gray-VGG-19, trained on image classification tasks only using the luminance channel to extract their own features, and compute their feature's differences.

Second, it is still challenging to select reference colors and propagate them properly by defining hand-crafted rules based on the similarity metrics. Instead, we propose an end-to-end network to learn selection and propagation simultaneously. Oftentimes both steps are not enough to recover all colors, especially when the reference is not very related to the target. To address this issue, our network would instead predict the dominant colors for misaligned objects from the large-scale data.

Fig. 3 illustrates the system pipeline. Our system uses the CIE Lab color space, which is perceptually linear. Thus, each image can be

separated into a luminance channel  $L$  and two chrominance channels  $a$  and  $b$ . The input of our system includes a grayscale target image  $T_L \in \mathcal{R}^{H \times W \times 1}$ , a color reference image  $R_{Lab} \in \mathcal{R}^{H \times W \times 3}$ , and the bidirectional mapping functions between them. The bidirectional mapping function is a spatial warping function defined with bidirectional correspondences. It returns the transformed pixel location given a source location "p". The two functions are respectively denoted as  $\phi_{T \rightarrow R}$  (mapping pixels from  $T$  to  $R$ ) and  $\phi_{R \rightarrow T}$  (mapping pixels from  $R$  to  $T$ ), where  $H$  and  $W$  are the height and width of the input images. For simplicity, we assume the two input images are of the same dimensions, although this is not necessary in practice. Our network consists of two sub-networks. The Similarity sub-net computes the semantic similarities between the reference and the target, and outputs bidirectional similarity maps  $sim_{T \leftrightarrow R}$ . The Colorization sub-net takes  $sim_{T \leftrightarrow R}$ ,  $T_L$  and  $R'_{ab}$  as the input, and outputs the predicted  $ab$  channels of the target  $P_{ab} \in \mathcal{R}^{H \times W \times 2}$ , which are then combined with  $T_L$  to get the colorized result  $P_{Lab}$  ( $P_L = T_L$ ). Details of the two sub-networks are introduced in the following sections.

### 3.1 Similarity Sub-Network

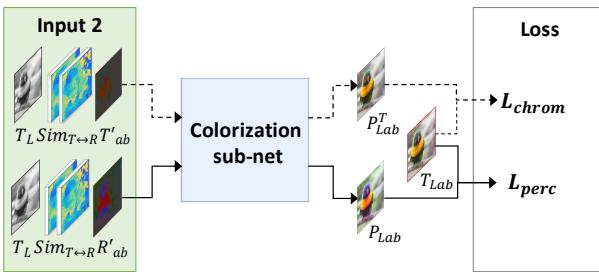
Before calculating pixel-level similarity, the two input images  $T_L$  and  $R_{Lab}$  have to be aligned. The bidirectional mapping functions  $\phi_{R \rightarrow T}$  and  $\phi_{T \rightarrow R}$  can be calculated with a dense correspondence algorithm, such as SIFTFlow [Liu et al. 2011], Daisy Flow [Tola et al. 2010] or DeepFlow [Weinzaepfel et al. 2013]. In our work, we adopt the latest advanced technique called Deep Image Analogy [Liao et al. 2017] for dense matching, since it is capable of matching images that are visually different but semantically related.

Our work is inspired by recent observations that CNNs trained on image recognition tasks are capable of encoding a full spectrum of features, from low-level textures to high-level semantics. It provides a robust and reliable similarity metric to variant image appearances (caused by variant lightings, times, viewpoints, and even slightly different categories), which may be challenging for low-level feature metrics (*e.g.*, intensity, SIFT, Gabor wavelet) used in many works [Welsh et al. 2002; Liu et al. 2008; Charpiat et al. 2008; Tai et al. 2005].

We take the intermediate output of VGG-19 as our feature representation. Certainly, other recognition networks, such as GoogleNet [Szegedy et al. 2015] or ResNet [He et al. 2015] can also be used. The original VGG-19 is trained on color images and has a degraded accuracy of recognizing grayscale images, as shown in Table 1. To reduce the performance gap between color images and their gray versions, we train a gray-VGG-19 only using the luminance channel of an image. It increases the top-5 accuracy from 83.63% to 89.39%, and approaches that of the original VGG-19 (91.24%) evaluated on color images.

**Table 1:** Classification accuracies of original and our fine-tuned VGG-19 calculated on ImageNet validation dataset.

	Top-5 Class Acc(%)	Top-1 Class Acc(%)
Ori VGG-19 tested on color image	91.24	73.10
Ori VGG-19 tested on gray image	83.63	61.14
Our VGG-19 tested on gray image	<b>89.39</b>	<b>70.05</b>



**Figure 4:** Two branches training of Colorization sub-net. Both branches take nearly the same Input 2 except for the concatenated chrominance channel. The aligned ground truth chrominance  $T'_{ab}$  is used for the Chrominance branch to compute the chrominance loss  $L_{chrom}$ , while the aligned reference chrominance  $R'_{ab}$  is used in the Perceptual branch to compute the perceptual loss  $L_{perc}$ .

We then feed the two luminance channels  $T_L$  and  $R_L$  into our gray-VGG-19 respectively, and obtain their five-level feature map pyramids ( $i = 1 \dots 5$ ). The feature map of each level  $i$  is extracted from the  $relu\{i\}_1$  layer. Note that the features have progressively coarser spatial resolution with increasing levels. We upsample all feature maps to the same spatial resolution of the input images and denote the upsampled feature maps of  $T_L$  and  $R_L$  as  $\{F_{T_L}^i\}_{i=1,\dots,5}$  and  $\{F_{R_L}^i\}_{i=1,\dots,5}$  respectively. Bidirectional similarity maps  $sim_{T \rightarrow R}^i$  and  $sim_{R \rightarrow T}^i$  are computed between  $F_T^i$  and  $F_R^i$  at each pixel  $p$ :

$$\begin{aligned} sim_{T \rightarrow R}^i(p) &= d(F_T^i(p), F_R^i(\phi_{T \rightarrow R}(p))), \\ sim_{R \rightarrow T}^i(p) &= d(F_T^i(\phi_{R \rightarrow T}(\phi_{T \rightarrow R}(p))), F_R^i(\phi_{T \rightarrow R}(p))). \end{aligned} \quad (1)$$

As mentioned in Liao et al. [2017], cosine distance performs better in measuring feature similarity since it is more robust to appearance variances between image pairs. Thus, our similarity metric  $d(x, y)$  between two deep features is defined as their cosine similarity:

$$d(x, y) = \frac{x^T y}{|x||y|}. \quad (2)$$

The forward similarity map  $sim_{T \rightarrow R}$  reflects the matching confidence from  $T_L$  to  $R_L$  while the backward similarity map  $sim_{R \rightarrow T}$  measures the matching accuracy in the reverse direction. We use  $sim_{T \leftrightarrow R}$  to denote both.

## 3.2 Colorization Sub-Network

We design an end-to-end CNN  $C$  to learn selection, propagation and prediction of colors simultaneously. As shown on the right of Fig. 3,  $C$  takes a thirteen-channel map as the input, which concatenates the gray target  $T_L$ , aligned reference with chrominance channels only  $R'_{ab}(p) = R_{ab}(\phi_{T \rightarrow R}(p))$ , and bidirectional similarity maps  $sim_{T \leftrightarrow R}$ . It also predicts  $ab$  channels of the target image  $P_{ab}$ . Next, we describe the loss function, network architecture and training strategy of the network.

### 3.2.1 Loss

Usually, the objective of colorization is to encourage the output  $P_{ab}$  to be as close as possible to the ground truth  $T_{ab}$ , the original  $ab$  channels of a color image  $T_{Lab}$  in the training dataset. However this is not true in exemplar-based colorization, because the colorization  $P_{ab}$  should allow customization with  $R'_{ab}$  (e.g., a flower can be colorized in either red, yellow, purple depending on the reference). Thus, it is not accurate to directly penalize a measure of the difference between  $P_{ab}$  and  $T_{ab}$ , as in other colorization networks (e.g., using  $L_2$  loss [Cheng et al. 2015; Iizuka et al. 2016],  $L_1$  loss [Isola et al. 2017; Zhang et al. 2017], or classification loss [Larsson et al. 2016; Zhang et al. 2016]).

Instead, our objective function is designed to consider two desiderata. First, we prefer reliable reference colors to be applied in the output, thus making it faithful to the reference. Second, we encourage the colorization to be natural, even when no reliable reference color is available.

To achieve both goals, we propose a multi-task network, which involves two branches, *Chrominance branch* and *Perceptual branch*. Both branches share the same network  $C$  and weight  $\theta$  but are associated with their own input and loss functions, as shown in Fig. 4. A parameter  $\alpha$  is used to dictate the relative weight between the two branches.

In the *Chrominance branch*, the network learns to selectively propagate the correct reference colors, which depends on how well the target  $T_L$  and the reference  $R_L$  are matched. However, training such a network is not easy: 1) on the one hand, the network cannot be trained directly with  $R'_{ab}$ , the reference chrominance warped on the target, because the corresponding ground truth colorization is unknown; 2) while on the other hand, the network cannot be trained using the ground truth target chrominance  $T_{ab}$  as a reference, because that would essentially be providing the network with the answer it is supposed to predict. Thus, we leverage the bidirectional mapping functions to reconstruct a “fake” reference  $T'_{ab}$  from the ground truth chrominance, i.e.,  $T'_{ab}(p) = T_{ab}(\phi_{R \rightarrow T}(\phi_{T \rightarrow R}(p)))$ .  $T'_{ab}$  replaces  $R'_{ab}$  in the training stage with the underlying hypothesis that correct color samples in  $T'_{ab}$  are very likely to lie in the same positions as correct color samples in  $R'_{ab}$ , since both are warped with  $\phi_{T \rightarrow R}$ .

To train the chrominance branch, both  $T_L$  and  $T'_{ab}$  are fed to the network, yielding the result  $P_{ab}^T$ :

$$P_{ab}^T = C(T_L, sim_{T \leftrightarrow R}, T'_{ab}; \theta). \quad (3)$$

Here,  $P_{ab}^T$  is colorized with the guidance of  $T'_{ab}$ , and should recover the ground truth  $T_{ab}$  if the network selects the correct color samples and propagates them properly. The smooth  $L_1$  distance is evaluated at each pixel  $p$  and integrated over the entire image to evaluate the *Chrominance loss*:

$$\mathcal{L}_{chrom}(P_{ab}^T) = \sum_p \text{smooth\_}L_1(P_{ab}^T(p), T_{ab}(p)) \quad (4)$$

where  $\text{smooth\_}L_1(x, y) = \frac{1}{2}(x - y)^2$ , if  $|x - y| < 1$ ,  $\text{smooth\_}L_1(x, y) = |x - y| - \frac{1}{2}$ , otherwise. We take the smooth  $L_1$  loss as the distance metric to avoid the averaging solution in the ambiguous colorization problem [Zhang et al. 2017].

Using the *Chrominance branch* only works for reliable color samples in  $R'_{ab}$  but may fail when the reference is dissimilar to parts of the image. To allow the network to predict perceptually plausible colors even without a proper reference, we add a *Perceptual branch*. In this branch, we take the reference  $R'_{ab}$  and the target



**Figure 5:** Visualization of Perceptual loss. Both colorized results have the same  $L_2$  chrominance ( $ab$  channels) distance to the ground truth, but the unnatural green face (right) has a much larger Perceptual loss than a more plausible skin color (left). Input image: Zhang et al. [2017].

$T_L$  as the network input during training. Then, we generate the predicted chrominance  $P_{ab}$ :

$$P_{ab} = \mathcal{C}(T_L, sim_{T \leftrightarrow R}, R'_{ab}; \theta). \quad (5)$$

In this branch, we minimize *Perceptual loss* [Johnson et al. 2016] instead. Formally:

$$\mathcal{L}_{perc}(P_{ab}) = \sum_p ||F_P(p) - F_T(p)||^2 \quad (6)$$

where  $F_P$  represents the feature maps extracted from the original VGG19 *relu5\_1* layer for  $P_{Lab}$ , and  $F_T$  is the same for  $T_{Lab}$ . *Perceptual loss* measures the semantic differences caused by unnatural colorization and is robust to appearance differences caused by two plausible colors, as shown in Fig. 5. We also did some exploration using cosine distance but found  $L_2$  distance generated superior results. A similar loss is widely used in other tasks, like style transfer [Chen et al. 2017a; Chen et al. 2018], photo-realistic image synthesis [Chen and Koltun 2017], and super resolution [Sajjadi et al. 2017].

Our network  $\mathcal{C}$ , parameterized by  $\theta$ , learns to minimize both loss functions (Equation (4) and (6)) across a large dataset:

$$\theta^* = \arg \min_{\theta} (\mathcal{L}_{chrom}(P_{ab}^T) + \alpha \mathcal{L}_{perc}(P_{ab})), \quad (7)$$

where  $\alpha$  is empirically set to 0.005 to balance both branches.

### 3.2.2 Architecture

The sub-network adopts a U-net encoder-decoder structure with some skip connections between the lower layers and symmetric higher layers. We empirically chose the U-net architecture because of its effectiveness, as evidenced in many image generation tasks [Badrinarayanan et al. 2015; Yu and Koltun 2015; Zhang et al. 2017]. Specifically, our network consists of 10 convolutional blocks. Each convolutional block contains  $2 \sim 3$  *conv-relu* pairs, followed by a batch normalization layer [Ioffe and Szegedy 2015] with the exception of the last block. The feature maps in the first 4 convolutional blocks are progressively halved spatially while doubling the feature channel number. To aggregate multi-scale contextual information without losing resolution (as in Yu et al. [2015], Zhang et al. [2017] and Fan et al. [2018]), dilated convolution layers with a factor of 2 are used in the 5th and 6th convolutional blocks. In the last 4 convolutional blocks, feature maps are progressively doubled spatially while halving the feature channel number. All down-sampling layers use convolution with stride 2, while all up-sampling layers use deconvolution with stride 2. Symmetric skip connections are added between the outputs of 1st and 10th, 2nd and 9th, and 3rd and 8th blocks, respectively. Finally, a convolution layer with a kernel size  $1 \times 1$  is added after the 10th block to predict the output  $P_{ab}$ . The final layer is a *tanh* layer (also used in Radford et al. [2015] and Chen et al. chen2017stylebank), which makes  $P_{ab}$  within a meaningful bound.

### 3.2.3 Dataset

We generate a training dataset based on ImageNet dataset [Russakovsky et al. 2015] by sampling approximately 700,000 image pairs from 7 popular categories: animals (15%), plants (15%), people (20%), scenery (25%), food (5%), transportation (15%) and artifacts (5%), involving 700 classes out of the total 1,000 classes due to the cost of generating training data. To let the network be robust to any reference, we sample image pairs with different extents of similarity. Specifically, 45% of image pairs belong to Top-5 similarity (selected by our recommendation algorithm described in Section 4) in the same class. Another 45% are randomly sampled within the same class. The remaining 10% have less similarity as they are randomly sampled from different classes but within the same category. In the training stage, we randomly switch the role of the two images for each pair to augment data. In other words, the target and the reference can be switched as two variant pairs during training. All images are scaled with the shortest edge of 256 pixels.

### 3.2.4 Training

Our network is trained using the Adam optimizer [Kingma and Ba 2014] with a batch size of 256. For every iteration, within the batch, the first 50% of data (128) go through the *Chrominance branch* using  $T'_{ab}$  as a reference and the remaining 50% (128) go through the *Perceptual branch* using  $R'_{ab}$ . The two branches respectively use corresponding losses. When updating the *Chrominance branch*, only *Chrominance loss* is used for gradient back propagation. When updating the *Perceptual branch*, only *Perceptual loss* is used for gradient back propagation. The initial learning rate is set to 0.0001 and decays by 0.1 every 3 epochs. By default, we train the whole network with 10 epochs. The whole training procedure takes around 2 days on 8 x Titan XP GPUs.

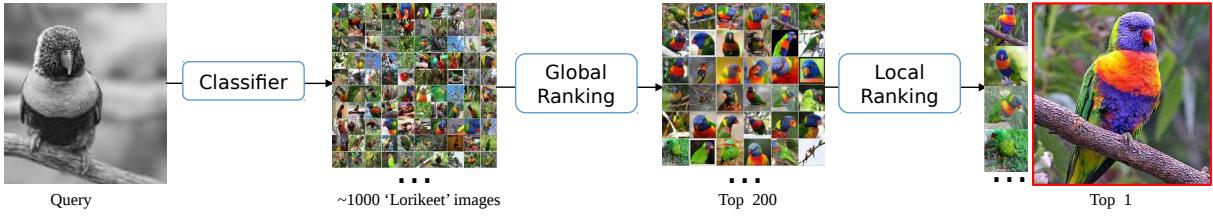
## 4 Color Reference Recommendation

As discussed earlier, our network is robust to reference selection, and provides user control for the colorization. To aid users in finding good references, we propose a novel image retrieval algorithm that automatically recommends good references to the user. Alternatively, the approach yields a fully automatic system by directly using the Top-1 candidate.

The ideal reference is expected to match the target image in both semantic content and photometric luminance. The purpose of incorporating the luminance term is to avoid any unnatural composition of luminance and chrominance. In other words, combining the reference chrominance with the target luminance may produce visually unfaithful colors to the reference. Therefore, we desire the reference's luminance to be as close as possible to the target's.

To measure semantic similarity, we adopt the intermediate features of a pre-trained image classification network as descriptors, which have been widely used in recent image retrieval works [Krizhevsky et al. 2012; Babenko et al. 2014; Gong et al. 2014; Babenko and Lempitsky 2015; Razavian et al. 2016; Tolias et al. 2015].

We propose an effective and efficient image retrieval algorithm. The system overview is shown in Fig. 6. We feed the luminance channel of each image from our training dataset (see Section 3.2.3) to our pre-trained gray-VGG-19 (in Section 3.1), and get its feature  $F^5$  from the last convolutional layer *relu5\_4* and  $F^6$  from the first fully-connected layer *fcb*. These features are pre-computed and stored in the database for the latter query. We also feed the query image (*i.e.*, the target gray image) to the gray-VGG-19 network, and get its corresponding features  $F_T^5, F_T^6$ . We then proceed with two ranking steps described next.



**Figure 6:** Color reference recommendation pipeline. Input images: ImageNet dataset.

#### 4.0.1 Global Ranking

Through gray-VGG-19, we can also get the recognized Top-1 class ID for the query image  $T_L$ . According to the class ID, we narrow down the search domain to all the images ( $\sim 1,000$  images) within the same class. Here, we want to further filter out dissimilar candidates by comparing  $fc$  features between the query and all candidates. Even within the same class, the candidate could have a context that is irrelevant to the query. For example, the query could be "a cat running on grass", but the candidate could be "a cat sitting inside the house". We would like the semantic content in the two images to be as similar as possible however. To achieve this, for each candidate image  $R_i (i = 1, 2, 3, \dots)$  in this class, we directly compute the cosine similarity (in Equation (2)) between  $F_T^b$  and  $F_{R_i}^b$  as the global score and rank all candidates by their scores.

#### 4.0.2 Local Ranking

The global ranking provides us the top- $N$  (we set  $N = 200$ ) candidates  $R_i$ . As we know,  $fc$  features fail to provide more accurate information about the object since it ignores the spatial information. For this purpose, we further prune these candidates by conducting a local ranking on the remaining  $N$  images. The local similarity score consists of both semantic and luminance terms.

For each image pair  $\{T_L, R_i\}$ , at each point  $p$  in  $F_T^5$ , we find its nearest neighbor  $q$  in  $F_{R_i}^5$  by minimizing the cosine distance between  $F_T^5(p)$  and  $F_{R_i}^5(q)$ , namely  $q = NN(p)$ . Then, the semantic term is defined as the cosine similarity  $d(\cdot)$  (see Equation (2)) between two feature vectors  $F_T^5(p)$  and  $F_{R_i}^5(q)$ .

The luminance term measures the similarity of luminance statistics between two local windows corresponding to  $p$  and  $q$  respectively. We evenly split image  $T_L$  into a 2D grid with each grid having  $16 \times 16$  resolution. Each grid in the image  $T_L$  indeed corresponds to a point in its feature map  $F_T^5$ , since it undergoes 4 down-sampling layers.  $C_T(p)$  is denoted as the grid cell in the image corresponding to the point  $p$  in  $F_T^5$ . Likewise,  $C_{R_i}(q)$  from  $R_L$  corresponds to the point  $q$  in  $F_{R_i}^5$ . The function  $d_H(\cdot)$  measures the correlation coefficient between luminance histograms of  $C_T(p)$  and  $C_{R_i}(q)$ .

The local similarity score is summarized as:

$$score(T, R_i) = \sum_p (d(F_T^5(p), F_{R_i}^5(q)) + \beta d_H(C_T(p), C_{R_i}(q))), \quad (8)$$

where  $\beta$  determines the relative importance between the two terms (empirically set to 0.25). This similarity score is computed for each pair  $\{T_L, R_i\} (i = 1, 2, 3, \dots)$ . According to all local scores, we re-rank all retained candidates and retrieve the top selections.

We compress neural features with the common PCA-based compression [Babenko et al. 2014] to accelerate the search. The channels of feature  $fc6$  are compressed from 4,096 to 128 and the channels of features  $relu5\_4$  are reduced from 512 to 64 with practically negligible loss. After these dimensionality reductions, our reference retrieval can run in real-time.

## 5 Discussion

In this section, we analyze and demonstrate the capabilities of our colorization network through ablation studies.

### 5.1 What does the Colorization sub-net learn?

The *Colorization sub-net C* learns how to select, propagate, and predict colors based on the target and the reference. As discussed earlier, it is an end-to-end network that involves two branches, each playing a distinct role. At first, we want to understand the behavior of the network using just the *Chrominance branch* during learning. For this purpose, we only train the *Chrominance branch* of  $C$  by minimizing the *Chrominance loss* (in Equation (4)), and evaluate it on one example to intuitively understand its operation (Fig. 7). By comparing the chrominance of the predicted result (4th column) with the chrominance of the aligned reference (3rd column), we notice that they have consistent colors in most regions (e.g., "blue" sky, "white" plane and "green" lawn). That indicates that our *Chrominance branch* picks color samples from the reference and propagates them to the entire image to achieve a smooth colorization.

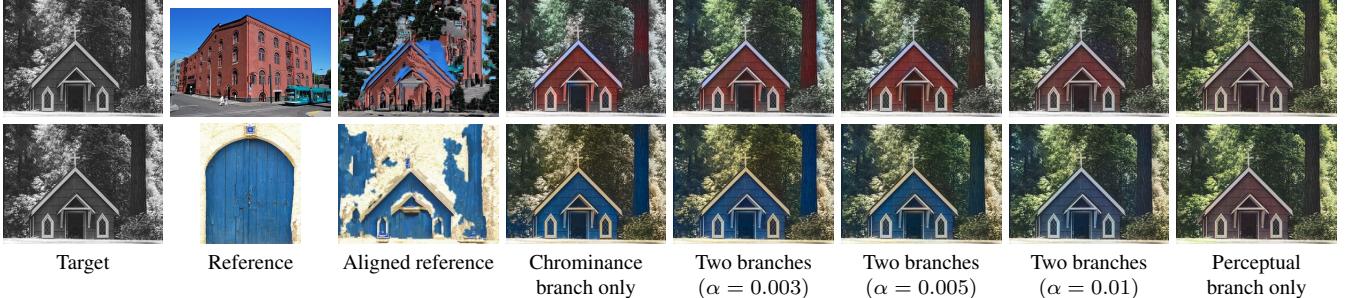
To learn which color samples are selected by the network, we compute the chrominance difference between the predicted result and the aligned reference in the 5th column ("blue" denotes nearly no difference while "red" denotes a noticeable difference). Colors of the points with smaller errors are more likely to be selected by the network and then retained in the final result.

"How does the network infer good samples?" or "Can it be directly inferred from the matching between images?" To answer these questions, we compare the difference map (6th column) with the averaged five-levels matching errors  $1 - sim_{T \rightarrow R}$  (7th column) and  $1 - sim_{R \rightarrow T}$  (8th columns). On the one hand, we can see that the matching errors are essentially consistent with the difference. This demonstrates that our network can learn a good sampling based on the matching quality, which serves as a key "hint" to determine appropriate locations. On the other hand, we find that the network does not always select points with smaller matching errors, as evidenced by a significant number of inconsistent samples. Without similarity maps, the Colorization sub-net can hardly infer the matching accuracy between the aligned reference and the input. It will also increase ambiguity of the color prediction. Thus, adaptive selection according to similarities may be infeasible through an intuitive heuristic. However, by using the large-scale data, our network can more robustly learn this mechanism directly.

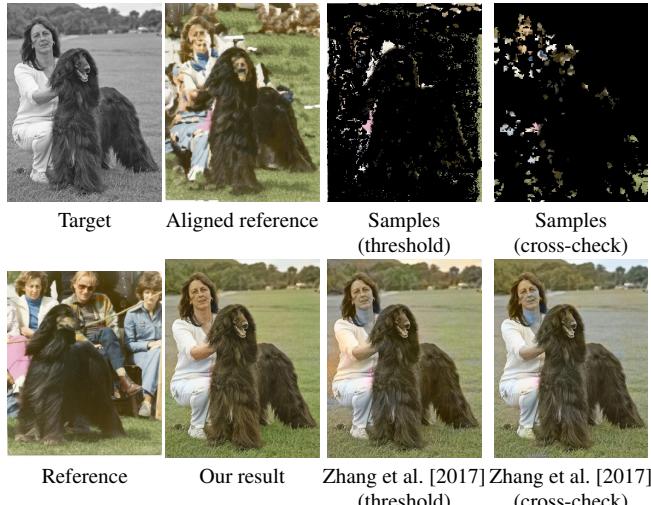
To understand the role of the *Perceptual branch*, we train it by solely minimizing the *Perceptual loss* (in Equation (6)). We show an example in Fig. 8. For this case, some regions do not have a good match to the reference (i.e., the right "trunk" object). By using the *Chrominance branch* only, we attain results with incorrect colors for trunk objects (4th column). However, the *Perceptual branch* is capable of addressing this problem (8th column). It predicts the single and natural brown color for the trunk, since the majority of trunks in the training data are brown. Thus, the prediction of the *Perceptual branch* is purely based on the dominant color of objects



**Figure 7:** Visualization of color selection in the Chrominance branch. The points with smaller difference between the predicted colorization  $P_{ab}$  and aligned reference color  $R'_{ab}$  are most likely to be selected by the network and maintained in the final results. Note how inconsistencies between the similarity maps and the true color difference make it difficult to determine good points by the hand-crafted rules. Input images: ImageNet dataset.



**Figure 8:** Comparison of results from the training with different branch configurations. Input images (from left to right, top to bottom): Tabitha Mort/pexels, Steve Morgan/wikimedia and Anonymous/pxhere.



**Figure 9:** Comparison of our end-to-end network with the alternative of selecting color samples with manual thresholds or cross-check matching, and then colorizing with Zhang et al. [2017]. Input images: ImageNet dataset.

from the large-scale data, and independent of the reference. As we can see in the 8th column, it predicts the same colors even for different references.

To enjoy the advantages of both branches, we adopt a multi-task training strategy to train both branches simultaneously. The term  $\alpha$  is used as their relative weight. The double-branch results in 5th – 7th columns of Fig. 8 explicitly indicate that our network learns to adaptively fuse the predictions of both branches: selecting and propagating the reference color at well-matched regions, but generalizing to the natural color learnt from large-scale data for mismatched or unrelated regions. The relative weight  $\alpha$  tunes the preference towards each branch. Evaluated on the ImageNet validation data, we set  $\alpha = 0.005$  as the default in our experiments.

## 5.2 Why is end-to-end learning crucial?

Our *Colorization sub-net* learns three key components in colorization: color sample selection, color propagation, and dominant color prediction. To our knowledge, there is no other work that learns three steps simultaneously through a neural network.

An alternative is to simply sequentially process the three steps. In our study, we adopt the state-of-the-art color propagation and prediction method [Zhang et al. 2017]. Such a learning-based method significantly advances previous optimization methods [Levin et al. 2004], especially when given few user points. We try two color selection strategies: 1) Threshold: select color points with the top 10% averaged bidirectional similarity score; 2) Cross-check in matching: select color points where the bidirectional mapping satisfies  $\phi_{T \rightarrow R}(\phi_{R \rightarrow T})(p) = p$ . Once the points are obtained, we directly feed them to the pre-trained color propagation network [Zhang et al. 2017]. We show the two predicted colorization results in 3rd and 4th columns of Fig. 9 respectively.

As we can see, the colorization does not work well and introduces many noticeable color artifacts. One possible reason is that the network [Zhang et al. 2017] is not trained on the type of input samples, but rather on user-guided points instead. Therefore, such a sequential learning would always result in a sub-optimal solution.

Moreover, the study also shows the difficulty in determining hand-crafted rules for point selections, as mentioned in Section 5.1. It is hard to eliminate all improper color samples through heuristics. The pre-trained network will also propagate wrong samples, thus causing such artifacts. On the contrary, our end-to-end learning approach avoids these pitfalls by jointly learning selection, propagation and prediction, resulting in a single network that directly optimizes for the quality of the final colorization.

## 5.3 Robustness

A significant advantage of our network is the robustness to reference selection when compared with traditional exemplar-based colorization. It can provide plausible colors whether the reference



**Figure 10:** Our method predicts plausible colorization with different references: manually selected, automatically recommended, randomly selected in the same class of the target, randomly selected in the same category, and randomly selected out of the category. Input images: ImageNet dataset except the two manual reference photos by Andreas Mortonus/flickr and Indi Samarajiva/flickr.



**Figure 11:** Our method works with different dense matching algorithms. The first row shows the target and the aligned references by different matching algorithms: SIFTFlow ([Liu et al. 2011]), DaisyFlow ([Tola et al. 2010]), DeepFlow ([Weinzaepfel et al. 2013]), and Deep Image Analogy ([Liao et al. 2017]). The second row shows the reference and final colorized results using different aligned references. Input images: ImageNet dataset.

is related or unrelated to the target. Fig. 10 shows how well our method works on varying references with different levels of similarity to the target image. As we can see, the colorization result is naturally more faithful to the reference when the reference is more similar to the target in their semantic content. In other situations, the result will be degenerated to a conservative colorization. This is due to the *Perceptual branch*, which predicts the dominant colors from large-scale data. This behavior is similar to the existing learning-based approaches (*e.g.*, [Iizuka et al. 2016; Larsson et al. 2016; Zhang et al. 2016]).

In addition, our network is also robust to different types of dense matching algorithms, as shown in Fig. 11. Note that our network is only trained using Deep Image Analogy [Liao et al. 2017] as the default matching approach, and the network is tested with various matching algorithms. We can also observe that the result is more faithful to the reference color at well-aligned regions; while the result is degenerated to the dominant colors at misaligned regions.

Note that better alignment can improve the results of objects which can find semantic correspondences in the reference, but cannot help the colorization of objects which do not exist in the reference.

## 5.4 Transferability

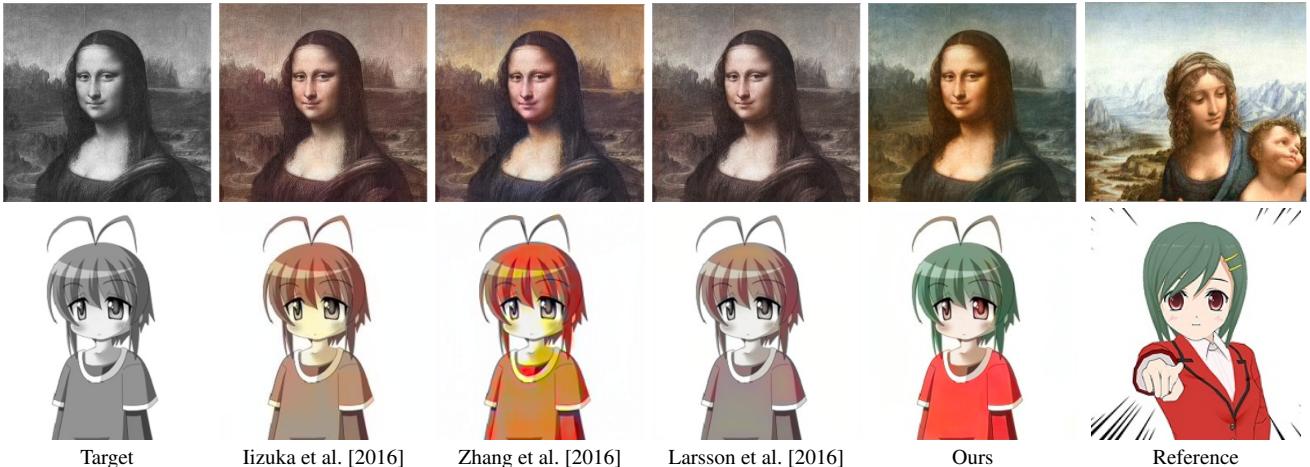
Previous learning-based methods are data-driven and thus only able to colorize images that share common properties with those in the training set. Since their networks are trained on natural images, like the ImageNet dataset, they would fail to provide satisfactory colors for unseen images, for example, human-created images (*e.g.*, paintings or cartoons). Their results may degrade to no colorization (1st, 3rd columns in Fig. 12) or introduce notable color artifacts (2nd column). By contrast, our method benefits from the reference and successfully works in both cases. Although our network does not see such types of images in training, with the *Chrominance branch* it learns to predict colors based on correlations of image pairs. The learnt ability is common to unseen objects.

## 6 Comparison and Results

In this section, we first report our performance and user study results. Then we qualitatively and quantitatively compare our method to previous techniques, including learning-based, exemplar-based, and interactive-based methods. Finally, we validate our method on legacy grayscale images and videos.

### 6.1 Performance

Our core algorithm is developed in CUDA. All of our experiments are conducted on a PC with an Intel E5 2.6GHz CPU and an NVIDIA Titan XP GPU. The total runtime for a  $256 \times 256$  image is approximately 0.166s, including 0.016s for reference recommendation, 0.1s for similarity measurement and 0.05s for colorization.



**Figure 12:** Transferability comparison of colorization networks trained on ImageNet. Input images (from left to right, top to bottom): Charpiat et al. [2008], Snow64/wikimedia and Ryo Taka/pixabay.

**Table 2:** Colorization results compared with learning-based methods on 10,000 images from the ImageNet validation set. The second and third columns are the Top-5 and Top-1 classification accuracies after colorization using the VGG19-BN and VGG16 network. The last column is the PSNR between the colored result and the ground truth.

	VGG19-BN/ VGG16	VGG19-BN/ VGG16	PSNR(dB)
Top-5 Class Acc(%)	Top-1 Class Acc(%)		
Ground truth (color)	90.35/89.99	71.12 /71.25	NA
Ground truth (gray)	84.2/81.35	61.5/57.39	23.28
Iizuka et al. [2016]	85.53/84.12	63.42/61.61	24.92
Zhang et al. [2016]	84.28/83.12	60.97/60.25	22.43
Larsson et al. [2016]	85.42/83.93	63.56/61.36	<b>25.50</b>
Ours	<b>85.94/84.79</b>	<b>65.1/63.73</b>	22.92

## 6.2 Comparison with Exemplar-based methods

To compare with existing exemplar-based methods [Welsh et al. 2002; Irony et al. 2005; Bugeau et al. 2014; Gupta et al. 2012], we run our algorithm on 35 pairs collected from their papers. Fig. 13 shows several representatives and the complete set can be found in the supplemental material. To provide a fair comparison, we directly borrow their results from their publications or run their publicly available code.

In these examples, the content and object layouts of the reference are very similar to the target (*i.e.*, no irrelevant objects or great intensity disparities). This is a strict requirement of existing exemplar-based methods, whose colorization relies solely on low-level features and is not learned from large-scale data. On the contrary, our algorithm is more general and has no such restrictive requirement. Even on these very related image pairs, our method shows better visual quality than previous techniques. The success comes from the sophisticated mechanism of color sample selection and propagation that are jointly learned from data rather than through heuristics.

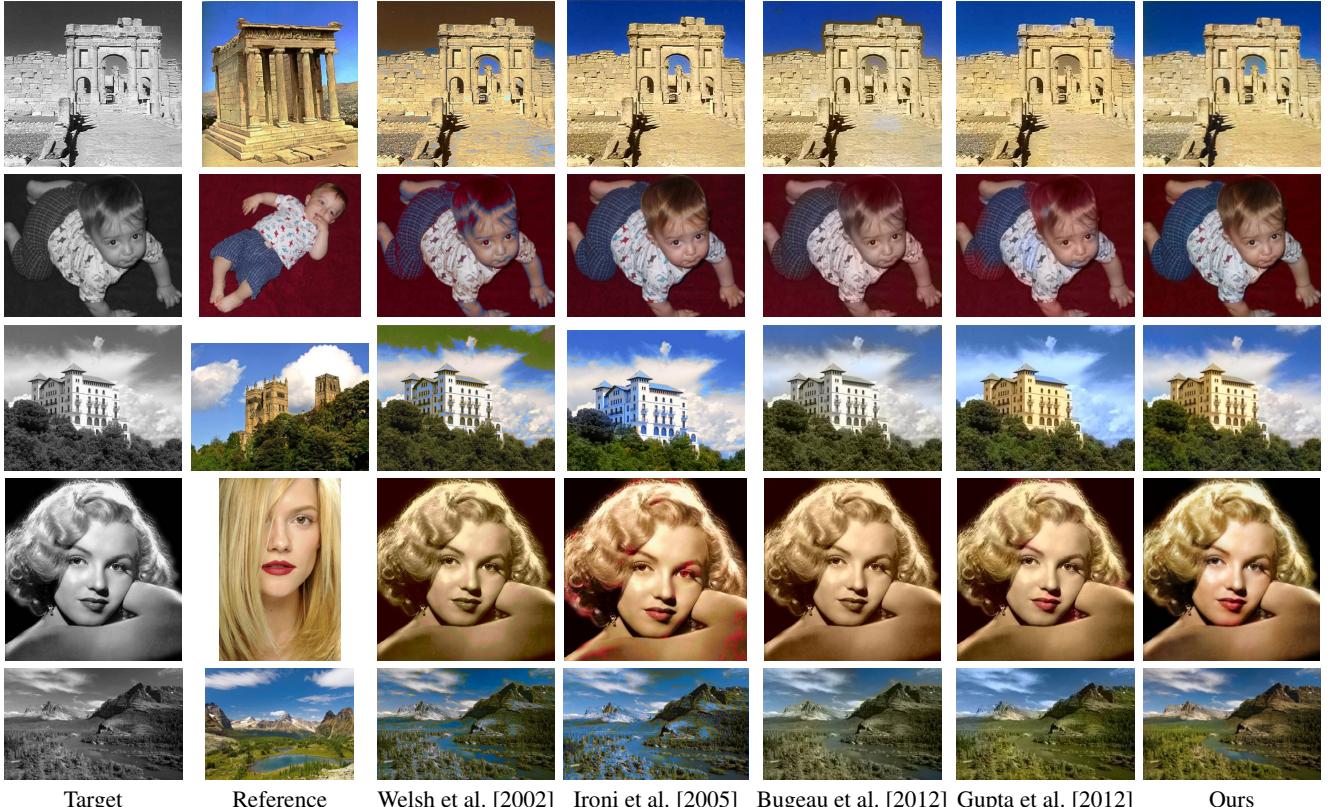
## 6.3 Comparison with learning-based methods

We compare our method with the-state-of-the-art learning-based colorization networks [Larsson et al. 2016; Zhang et al. 2016; Iizuka et al. 2016] by evaluating on 10,000 images in the validation set of ImageNet (same as Larsson et al. [2016]). Our method is trained on a subset of the ImageNet training set, as described in Section 3.2. We tested our automatic solution by taking the Top-1 recommendation as the reference (Sec. 4). To be fair, we use author-released models trained on the ImageNet dataset as well to run their methods.

We show a quantitative comparison of colored results in Table 2 on two metrics: PSNR relative to the ground truth and classification accuracy. Our results have a lower PSNR score (22.9178dB) than Larsson et al. [2016] and Iizuka et al. [2016], because PSNR overly penalizes a plausible but different colorization result. A correct colorization faithful to the reference may even achieve a lower PSNR than a conservative colorization, such as predicting gray for every pixel (24.9214dB). On the contrary, our method outperforms all other methods on image recognition accuracy rates when sending the colored results into VGG19 or VGG16 pre-trained on image recognition task. It indicates that our colored results seem to be more natural than others, which can be recognizable as well as the true color image.

A qualitative comparison for selected representative cases is shown in Fig. 14. For a full set of 200 images randomly drawn from 1,000 cases, please refer to our supplemental material. From this comparison, an apparent difference is that our results are more saturated and colorful when compared to Iizuka et al. [2016] and Larsson et al. [2016], with the help of sampling colorful points from the reference. Zhang et al. [2016] uses a class-rebalancing step to oversample more colorful portions of the gamut during training, but such a solution sometimes results in overly aggressive colorization and causes artifacts (*e.g.* the blue and orange colors in the 4th row of Fig. 14). Our approach can control colorization and achieve desired colors by simply giving different references, thus our results are visually faithful to the reference colors.

In addition to quantitative and qualitative comparisons, we use a perceptual metric to evaluate how compelling our colorization looks to a human observer. We ran a *real vs. fake* two-alternative forced-choice user study on Amazon Mechanical Turk (AMT) across different learning-based methods. This is similar to the approach taken by Zhang et al. [2016]. Participants in the study were



**Figure 13:** Comparison results with example-based methods. Input images: Ironi et al. [2005] and Gupta et al. [2012].

shown a series of pairs of images. Each pair consisted of a ground-truth color photo and a re-colorized version produced by either our algorithm (randomly selected reference or Top-1 recommended reference) or a baseline [Iizuka et al. 2016; Larsson et al. 2016; Zhang et al. 2016]. The two images were shown side-by-side in randomized order. For every pair, participants were asked to observe the image pair for no more than 5 seconds and click on the photo they believed was the most realistic as early as possible. All images were shown with the resolution of 256 pixels on the short edge.

To guarantee all algorithms can be compared by the same "turker" populations, we included results from different algorithms in one experimental session for each participant. Each session consisted of 5 practice trials (excluded from subsequent analysis), followed by 50 randomly selected test pairs (each algorithm contributed 10 pairs). During the practice trials, participants were given feedback as to whether their answers were correct. No feedback was given during the 50 test pairs. We conducted 5 different sessions to make sure every algorithm covered all image pairs. The participants were only allowed to complete at most one session. All experiment sessions were posted simultaneously and a total of 125 participants were involved in the user study (25 $\pm$ 2 participants per session).

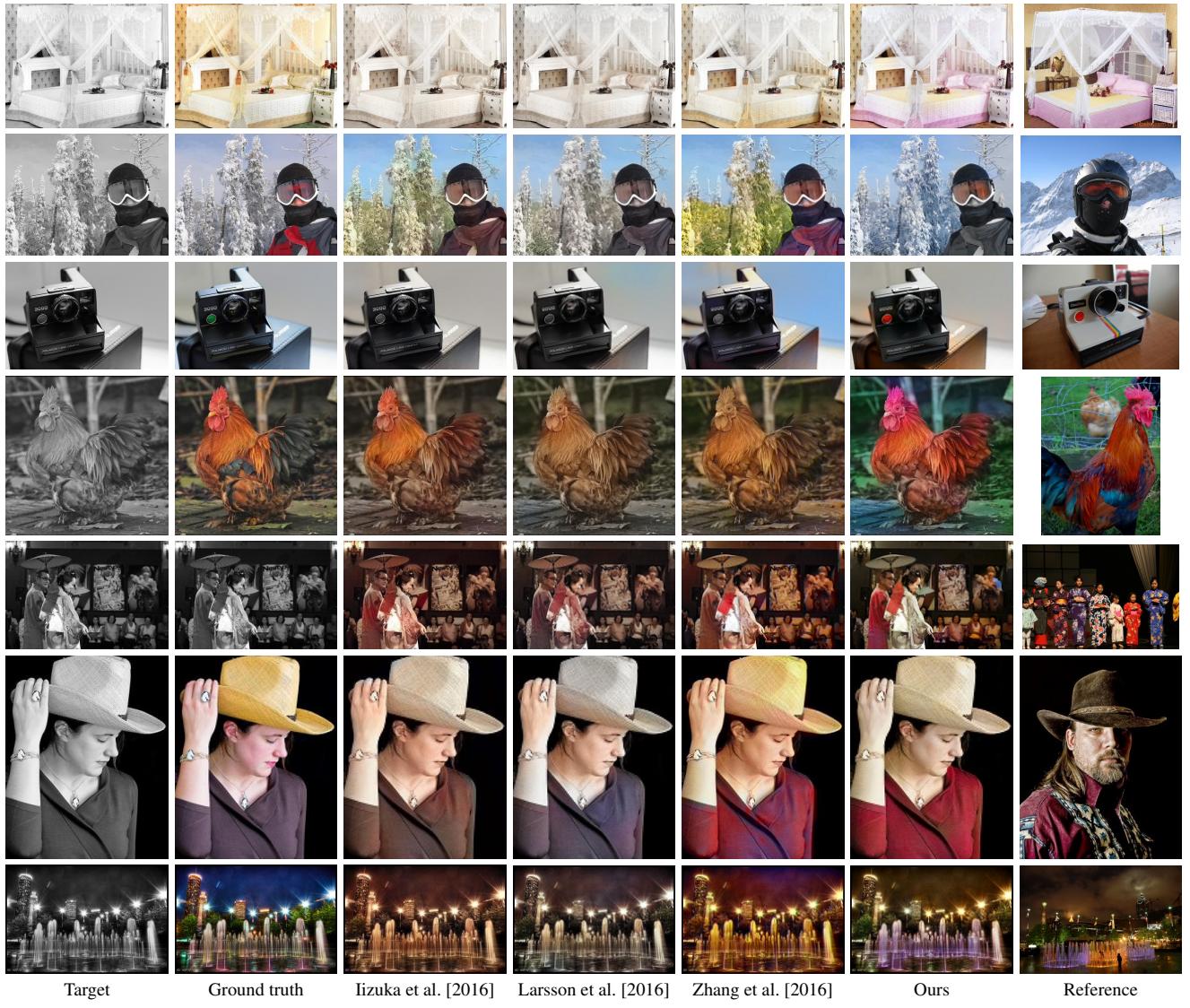
As shown in Table 3, our method with the Top-1 reference (38.08%) and Zhang et al. [2016] (35.36%) respectively ranked 1st and 2nd in the fooling rate. We felt that this may be partly because participants preferred more colorful results to less saturated results as shown in Fig. 15. Zhang et al. [2016] uses a class-rebalancing step to encourage rare colors but at the expense of images which are overly-aggressively colorized; while our method produces more vibrant colorization by utilizing correct color samples from the reference. Our method with random reference also degenerates to conservative color prediction since few reliable color samples can be

**Table 3:** Amazon Mechanical Turk real v.s. fake fooling rate. We compared our method using an automatically recommended reference or a random intra-class reference with other learning-based methods. Note that the best expectation of fooling rate should be around 50%, which occurs when the user cannot distinguish real from fake images and is forced to choose between two equally believable images. Input images: ImageNet dataset.

Method	Fooling Rate (%)
Iizuka et al. [2016]	24.56 $\pm$ 1.76
Larsson et al. [2016]	24.64 $\pm$ 1.71
Zhang et al. [2016]	35.36 $\pm$ 1.52
Ours with random reference	21.92 $\pm$ 1.56
Ours with Top-1 reference	<b>38.08 <math>\pm</math> 1.72</b>

used from the unrelated reference. This verifies that a good reference is important to high-quality colorization.

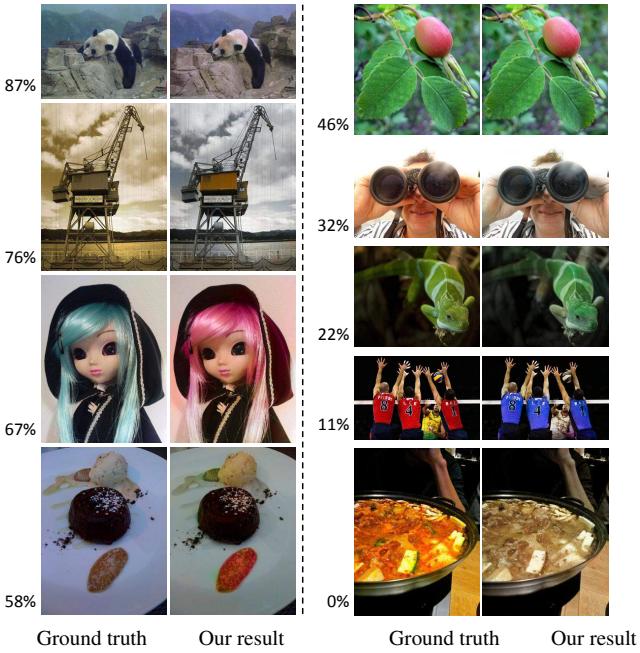
Fig. 16 provides a better sense of the participants' competency at detecting subtle errors made by our algorithm. The percentage on the left shows how often participants think our colorized result is more realistic than the ground truth. Some issues may come from lack of colorization in some local regions (e.g., 0%, 11%), or poor white balancing in the ground truth image (e.g., 22%, 32%). Surprisingly, our results are considered more natural to human observers than the ground truth image in some cases (e.g. 87%, 76%).



**Figure 14:** Comparison results with learning-based methods. Input images: ImageNet dataset.



**Figure 15:** An example to show users preference on vibrant colorization. The numbers in brackets represent its fooling rates. Our colorized results (3rd and last columns) are guided by the top-right references. Input images: ImageNet dataset.



**Figure 16:** Examples from the user study. Results are generated with our method with the Top-1 reference and are sorted by how often the users chose our algorithm’s colorization over the ground truth. Input images: ImageNet dataset.

#### 6.4 Comparison with interactive-based methods

We compare our hybrid method with a different hybrid solution [Zhang et al. 2017] which combines user-guided scribbles (*i.e.*, points) and deep learning. As shown in Fig. 17, by giving a proper reference selected by the user, our method can achieve comparable quality to theirs with dozens of user-given color points. Thus, our method proposes a simple way to control the appearance of colorization generated with the help of deep neural networks.

Zhang et al. [2017] also present a variant of their method which uses a global color histogram of a reference image as input to control colorization results. In Fig. 18, we show a comparison with results by Zhang et al. [2017] using the global color histogram either from the reference image (2nd column) or the aligned reference (3rd column). Their method provides a global control to alter color distribution and average saturation but fails to achieve locally variant colorization effects. Our method can preserve semantic correspondence and locally map the reference color to the target (*e.g.*, the plant colorized green and the flowerpot colorized in blue).

#### 6.5 Colorization of legacy photographs and movies

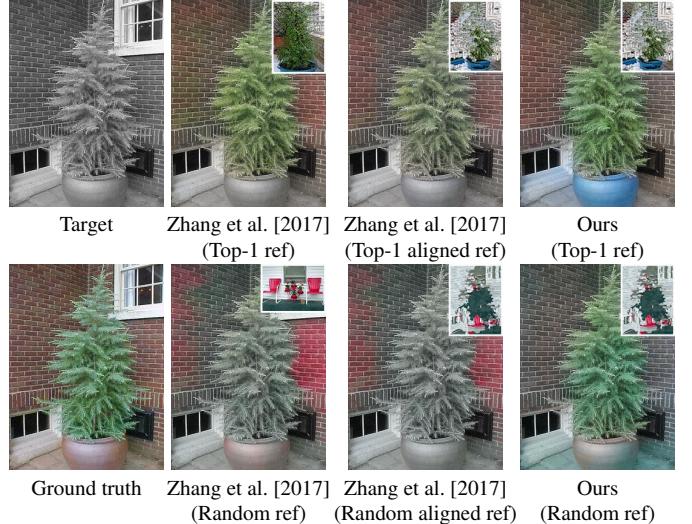
Our system was trained on “synthetic” grayscale images by removing the chrominance channels from color images. We tested our system on legacy grayscale images, and show some selected results in Fig. 19. Moreover, our method can be extended to colorize legacy movies by independently colorizing each frame and then temporally smoothing the colorized results with the method of Bonneel et al. [2015]. Some selected frames of a movie example are shown in Fig. 20. Please refer to our supplemental material for a video demo.

### 7 Limitations and Conclusions

We have presented a novel colorization approach that employs a deep learning architecture and a reference color image. Our ap-



**Figure 17:** Comparison results with the interactive-based method. The points overlaid on the target are manually given and used in Zhang et al. [2016], while the reference in the last column is manually selected and used by our approach. Input images (from left to right, top to bottom): Ansel Adams/wikipedia, Carina Chen/pixabay, Dorothea Lange and Bess Hamiti/pixabay.



**Figure 18:** Comparison to Zhang et al. [2016] using global histogram hints from references overlaid on the top-right corner. The histogram used in Zhang et al. [2016] is either from the reference (2nd column) or from its aligned version generated by Liao et al. [2017] (3rd column). Input images: ImageNet dataset.

proach is a general solution for exemplar-based colorization since it yields plausible results even in cases where the target image does not have clear correspondences in the reference. In such cases, it is still capable of producing plausible and natural colors for the target image. Unlike most deep-learning colorization frameworks, our approach allows us to control colorized results. Furthermore, with the reference recommendation algorithm, the system also provides the user with an automatic tool for re-coloring black-and-white photographs and movies. Our approach also suffers from some limitations that can be addressed in future work. First, our network cannot colorize objects with unusual or artistic colors, since it is constrained by the learning from the proposed *Perceptual branch*, as shown in the top row of Fig. 21.

Second, the perceptual loss based on the classification network (VGG) cannot penalize incorrect colors in regions with less semantic importance, such as the wall in the second row of Fig. 21, or fails to distinguish less semantic regions with similar local texture, such as the similar sand and grass textures in the third row of Fig. 21. In addition, our result is less faithful to the reference when there are dramatic luminance disparities between images, as shown in the bottom row of Fig. 21. To mitigate this limitation, our reference



**Figure 19:** Colorization of legacy pictures. In each set, the target grayscale photo is the upper-left, the reference is the lower-left and our result lies on the right. Input images (from left to right, top to bottom, target to reference): George L. Andrews/wikipedia, Official White House Photographer/wikimedia, Vandamm/wikimedia, Anonymous/wikimedia, Esther Bubley/wikimedia, Anonymous/wikimedia, Nick Macneill/geograph, Bernd/pixabay, Oberholster Venita/pixabay, EU2017EE Estonian Presidency/wikimedia, Audrey Coey/flickr, EU2017EE Estonian Presidency/wikimedia, Patrick Feller/wikimedia and Anonymous/pixabay.

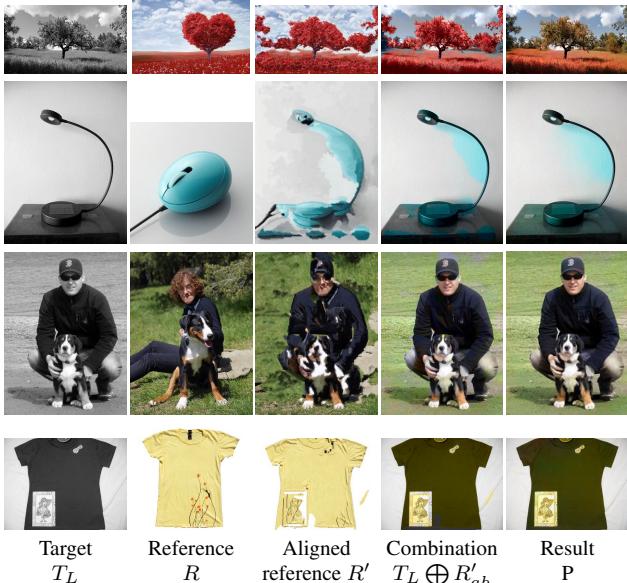


**Figure 20:** Extending our method to video colorization. All black and white frames (top row) are independently colorized with the same reference (leftmost column of bottom row) to generate colorized results (right 4 columns of bottom row). The input clip is from the film *Anna Lucasta* (public domain) and the reference photo is by Heather Harvey/flickr.

recommendation algorithm enforces luminance similarity in the local ranking. Occasionally, our method fails to predict colors for some local regions, as shown in Fig. 16. It would be worthwhile to explore how to better balance the two branches of our network.

## References

- BABENKO, A., AND LEMPITSKY, V. 2015. Aggregating local deep features for image retrieval. In *Proc. ICCV*, 1269–1277.
- BABENKO, A., SLESAREV, A., CHIGORIN, A., AND LEMPITSKY, V. 2014. Neural codes for image retrieval. In *Proc. ECCV*, Springer, 584–599.
- BADRINARAYANAN, V., KENDALL, A., AND CIPOLLA, R. 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*.
- BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 28, 3, 24–1.
- BAY, H., TUYTELAARS, T., AND VAN GOOL, L. 2006. Surf: Speeded up robust features. 404–417.
- BONNEEL, N., TOMPKIN, J., SUNKAVALLI, K., SUN, D., PARIS, S., AND PFISTER, H. 2015. Blind video temporal consistency. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)* 34, 6, 196.
- BUGEAU, A., AND TA, V.-T. 2012. Patch-based image colorization. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, IEEE, 3058–3061.
- BUGEAU, A., TA, V.-T., AND PAPADAKIS, N. 2014. Variational exemplar-based image colorization. *IEEE Trans. on Image Processing* 23, 1, 298–307.
- CHARPIAT, G., HOFMANN, M., AND SCHÖLKOPF, B. 2008. Automatic image colorization via multimodal predictions. 126–139.
- CHEN, Q., AND KOLTUN, V. 2017. Photographic image synthesis with cascaded refinement networks. In *Proc. ICCV*, vol. 1.
- CHEN, D., LIAO, J., YUAN, L., YU, N., AND HUA, G. 2017. Coherent online video style transfer. In *Proc. ICCV*.
- CHEN, D., YUAN, L., LIAO, J., YU, N., AND HUA, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *Proc. CVPR*.
- CHEN, D., YUAN, L., LIAO, J., YU, N., AND HUA, G. 2018. Stereoscopic neural style transfer. In *Proc. CVPR*.
- CHENG, Z., YANG, Q., AND SHENG, B. 2015. Deep colorization. In *Proc. ICCV*, 415–423.
- CHIA, A. Y.-S., ZHUO, S., GUPTA, R. K., TAI, Y.-W., CHO, S.-Y., TAN, P., AND LIN, S. 2011. Semantic colorization with



**Figure 21:** Limitations of our work. Top row: our network cannot colorize objects with unusual or artistic colors constrained by the perceptual loss. Second row: the perceptual loss does not sufficiently penalize incorrect reference colors on regions with less semantic importance, e.g. a smooth background. Third row: the classification network fails to distinguish regions with similar local textures, e.g. sand and grass. Forth row: the result is visually less faithful to the reference if their luminance gaps are too large. Input images: ImageNet dataset except the images on the first row by Anonymous/pxhere and Anonymous/pixabay.

internet images. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)* 30, 6, 156.

ÇİÇEK, Ö., ABDULKADIR, A., LIENKAMP, S. S., BROX, T., AND RONNEBERGER, O. 2016. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 424–432.

DESHPANDE, A., ROCK, J., AND FORSYTH, D. 2015. Learning large-scale automatic image colorization. In *Proc. ICCV*, 567–575.

FAN, Q., CHEN, D., YUAN, L., HUA, G., YU, N., AND CHEN, B. 2018. Decouple learning for parameterized image operators. In *ECCV 2018, European Conference on Computer Vision*.

GONG, Y., WANG, L., GUO, R., AND LAZEBNIK, S. 2014. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. ECCV*, Springer, 392–407.

GUPTA, R. K., CHIA, A. Y.-S., RAJAN, D., NG, E. S., AND ZHIYONG, H. 2012. Image colorization using similar images. In *Proc. of the 20th ACM international conference on Multimedia*, ACM, 369–378.

HE, K., ZHANG, X., REN, S., AND SUN, J. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

HE, M., LIAO, J., YUAN, L., AND SANDER, P. V. 2017. Neural color transfer between images. *arXiv preprint arXiv:1710.00756*.

HUANG, Y.-C., TUNG, Y.-S., CHEN, J.-C., WANG, S.-W., AND WU, J.-L. 2005. An adaptive edge detection based colorization algorithm and its applications. In *Proc. of the 13th annual ACM international conference on Multimedia*, ACM, 351–354.

IIZUKA, S., SIMO-SERRA, E., AND ISHIKAWA, H. 2016. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM, vol. 35, 110.

IOFFE, S., AND SZEGEDY, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 448–456.

IRONY, R., COHEN-OR, D., AND LISCHINSKI, D. 2005. Colorization by example. In *Rendering Techniques*, 201–210.

ISOLA, P., ZHU, J.-Y., ZHOU, T., AND EFROS, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*.

JOHNSON, J., ALAHI, A., AND FEI-FEI, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, Springer, 694–711.

KINGMA, D., AND BA, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

LARSSON, G., MAIRE, M., AND SHAKHNAROVICH, G. 2016. Learning representations for automatic colorization. In *Proc. ECCV*, 577–593.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 23, 3, 689–694.

LIAO, J., YAO, Y., YUAN, L., HUA, G., AND KANG, S. B. 2017. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088* 36, 4, 120.

LIU, X., WAN, L., QU, Y., WONG, T.-T., LIN, S., LEUNG, C.-S., AND HENG, P.-A. 2008. Intrinsic colorization. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)* 27, 5, 152.

LIU, C., YUEN, J., AND TORRALBA, A. 2011. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 5, 978–994.

LOWE, D. G. 1999. Object recognition from local scale-invariant features. In *Proc. ICCV*, vol. 2, IEEE, 1150–1157.

LUAN, Q., WEN, F., COHEN-OR, D., LIANG, L., XU, Y.-Q., AND SHUM, H.-Y. 2007. Natural image colorization. In *Proc. of the 18th Eurographics conference on Rendering Techniques*, Eurographics Association, 309–320.

PITIE, F., KOKARAM, A. C., AND DAHYOT, R. 2005. N-dimensional probability density function transfer and its application to color transfer. In *Proc. ICCV*, vol. 2, 1434–1439.

QU, Y., WONG, T.-T., AND HENG, P.-A. 2006. Manga colorization. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)* 25, 3, 1214–1220.

RADFORD, A., METZ, L., AND CHINTALA, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

RAZAVIAN, A. S., SULLIVAN, J., CARLSSON, S., AND MAKI, A. 2016. A baseline for visual instance retrieval with deep convolutional networks. *arXiv preprint arXiv:1412.6574*.

- REINHARD, E., ADHIKMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Computer graphics and applications* 21, 5, 34–41.
- RONNEBERGER, O., FISCHER, P., AND BROX, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 234–241.
- RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., ET AL. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3, 211–252.
- SAJJADI, M. S., SCHOLKOPF, B., AND HIRSCH, M. 2017. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proc. CVPR*, 4491–4500.
- SANGKLOY, P., LU, J., FANG, C., YU, F., AND HAYS, J. 2016. Scribbler: Controlling deep image synthesis with sketch and color. In *Proc. CVPR*.
- SIMONYAN, K., AND ZISSERMAN, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. 2015. Going deeper with convolutions. In *Proc. CVPR*.
- TAI, Y.-W., JIA, J., AND TANG, C.-K. 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In *Proc. CVPR*, vol. 1, IEEE, 747–754.
- TOLA, E., LEPESTIT, V., AND FUÀ, P. 2010. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 5, 815–830.
- TOLIAS, G., SICRE, R., AND JÉGOU, H. 2015. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*.
- WEINZAEPFEL, P., REVAUD, J., HARCHAOUI, Z., AND SCHMID, C. 2013. Deepflow: Large displacement optical flow with deep matching. In *Proc. ICCV*, 1385–1392.
- WELSH, T., ASHIKMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)* 21, 3, 277–280.
- YANG, H., LIN, W.-Y., AND LU, J. 2014. Daisy filter flow: A generalized discrete approach to dense correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3406–3413.
- YATZIV, L., AND SAPIRO, G. 2006. Fast image and video colorization using chrominance blending. *IEEE Trans. on Image Processing* 15, 5, 1120–1129.
- YOSINSKI, J., CLUNE, J., NGUYEN, A., FUCHS, T., AND LIPSON, H. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.
- YU, F., AND KOLTUN, V. 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.
- ZHANG, R., ISOLA, P., AND EFROS, A. A. 2016. Colorful image colorization. In *Proc. ECCV*, 649–666.
- ZHANG, R., ZHU, J.-Y., ISOLA, P., GENG, X., LIN, A. S., YU, T., AND EFROS, A. A. 2017. Real-time user-guided image colorization with learned deep priors. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 36, 4, 119.
- ZHOU, T., KRÄHENBÜHL, P., AUBRY, M., HUANG, Q., AND EFROS, A. A. 2016. Learning dense correspondence via 3d-guided cycle consistency. *arXiv preprint arXiv:1604.05383*.