arXiv:1804.04128v1 [cs.CV] 11 Apr 2018

Text2Colors: Guiding Image Colorization through Text-Driven Palette Generation

Wonwoong Cho*[1], Hyojin Bahng*[1], David K. Park*[1], Seungjoo Yoo*[1]
*Equal contribution, random order
Ziming Wu[2], Xiaojuan Ma[2], and Jaegul Choo[1]

[1]Korea University
{tyflehd21,hjj552,heykeetae,seungjooyoo,jchoo}@korea.ac.kr
[2]Hong Kong University of Science and Technology
zwual@connect.ust.hk, mxj@cse.ust.hk

Abstract. In this paper, we propose a novel approach to generate multiple color palettes that reflect the semantics of input text and then colorize a given grayscale image according to the generated color palette. In contrast to existing approaches, our model can understand rich text, whether it is a single word, a phrase, or a sentence, and generate multiple possible palettes from it. To achieve this task, we introduce our manually curated dataset called Palette-and-Text (PAT), which consists of 10,183 pairs of text and its corresponding color palette. Our proposed model consists of two conditional generative adversarial networks: the text-to-palette generation networks and the palette-based colorization networks. The former employs a sequence-to-sequence model with an attention module to capture the semantics of the text input and produce relevant color palettes. The latter utilizes a U-Net architecture to colorize a grayscale image using the generated color palette. Our evaluation results show that people preferred our generated palettes over ground truth palettes and that our model can effectively reflect the given palette when colorizing an image.

Keywords: Color Palette Generation, Image Colorization, Conditional Generative Adversarial Networks

1 Introduction

Humans can associate certain words with certain colors. The real question is, can machines effectively learn the relationship between colors and texts? Previous approaches have attempted to match a single word with a single color [1,2,3,4]. Since multi-color palettes are highly expressive in conveying semantics [5], other studies have further tried to link single words with multi-color palettes [6,7]. However, these methods only take single words as input and can only recommend color palettes in pre-existing datasets. Compared to previous works, our model can generate a whole new palette of diverse colors when given rich text input,

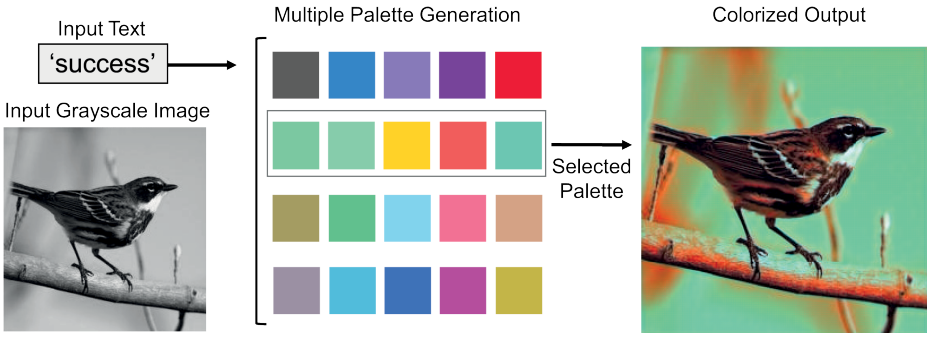


Fig. 1. Our model can produce a diverse selection of palettes for text inputs such as 'success.' Users can choose which palette will be used for the final colorization output.

including both single-word and multiple-word descriptions. A diversity of a color palette refers to how different the five colors included in the palette are.

In this paper, we propose a novel method to generate multiple color palettes that convey the semantics of rich text and then colorize a given grayscale image according to the generated color palette. Perception of color is inherently multimodal [8], meaning that a particular text input can be mapped to multiple possible color palettes. To incorporate such multimodality into our model, our palette generation networks are designed to generate multiple palettes from a single text input. We further apply our generated color palette to the colorization task. Motivated from previous user-guided colorizations that utilize color hints given by users [9,10], we design our colorization networks to utilize the palette during the colorization process. Our evaluation demonstrates that the colorized outputs do not only reflect the colors in the palette but also convey the semantics of the text input.

The contribution of this paper includes:
(1) We propose a novel deep neural network architecture that can generate multiple color palettes based on natural-language text input.
(2) Our model is able to use the generated palette to produce plausible colorizations of a grayscale image.
(3) We introduce our manually curated dataset called Palette-and-Text (PAT), which includes 10,183 pairs of a multi-word text and a multi-color palette. [1]

2 Related Work

Color Semantics Meanings associated with a color are both innate and learned [11]. For instance, red can make us instinctively feel alert, and white is sometimes culturally learned to be associated with purity [11]. Since color has a strong association with high-level semantic concepts [12], producing palettes from text input is

[1] Dataset and codes are publicly available at https://github.com/awesome-davian/Text2Colors/

useful in aiding artists and designers [5] and allows automatic colorization from palettes [9,13]. A downside to using text to choose a filter is that filter names do not usually convey the filters colors [6], thus making it difficult for users to find the filter that matches their taste just by looking at filter names. To bridge this discrepancy between color palettes and their names, color-word association has long been studied. Query-based and learning-based approaches are two typical methods for palette recommendation based on user text input. The former methods use text inputs to query an image from an image dictionary [6,7]. Then, colors are extracted from the queried image to make an associated palette. This method faces a big problem; the text input is sometimes mapped to the image content of the queried image rather than the color that the text implies. Instead of looking for a target directly, learning-based approaches match color palettes to their linguistic descriptions by learning their semantic association from large-scale data. Machine learning techniques [14,15,16] have been used to achieve promising results. However, they can only support word-level input and also cannot generate new color palettes.

Conditional GANs Conditional generative adversarial networks (cGAN) are GAN models that use conditional information for the discriminator and generator [17]. cGANs have drawn promising results for image generation from text [18,19,20] and image-to-image translation [21,22,23]. StackGAN [20] is the first model to use conditional loss for text to image synthesis. It achieved much better results compared to previous models by breaking down image synthesis into a sketch and refinement task. StackGAN also introduced conditioning augmentation technique which allows the model to output diverse results even when given the same input text. Conditioning augmentation will further be elaborated in Section 4.1.

Interactive Colorization Colorization is a multimodal task and desired colorization results for the same object may vary from person to person [8]. A number of studies introduce interactive methods that allows users to control the final colorization output [10,24]. In these models, users directly interact with the model by pinpointing where to color. Even though these methods achieve satisfactory results, a limitation is that users need to have a certain level of artistic skill. Thus instead of making the user directly color an image, other studies make users take a more indirect approach by utilizing color palettes to recolor an image [25,13]. Palette-based filters are an effective way for non-experts to recolor an image [25].

Sequence-to-Sequence with Attention Recurrent Neural Networks (RNNs) are a popular tool due to their superior ability to learn from sequential data. RNNs are utilized in various tasks including sentence classification [26], text generation [27] and sequence-to-sequence prediction [28]. Neural machine translation systems model the probability that a source sentence $\{x_1, \cdots, x_n\}$ is translated to a target sentence $\{y_1, \cdots, y_n\}$ as $p(y \mid x)$. Incorporating attention into

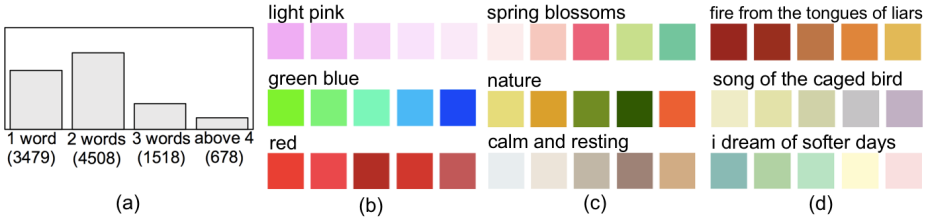


Fig. 2. About our Palette-and-Text (PAT) dataset: **(a)** shows the number of data items with respect to their text lengths. On the right are examples that show diverse text-palette pairs in PAT. Those text descriptions matching with their palettes include **(b)** direct color names, **(c)** texts with a relatively low level of semantic relations to colors, **(d)** those with a high-level semantic context.

a sequence-to-sequence model is known to help to improve the model performance [29]. Attention modules enable networks to selectively focus on parts of a source sentence. This allows a model to learn relations between different modalities (e.g., text - colors, text - action [30], English - French [31]). ub

3 Palette-and-Text (PAT) Dataset

This section introduces our manually curated dataset named Palette-and-Text (PAT). PAT contains 10,183 text and five-color palette pairs, where the set of five colors in a palette is associated with its corresponding text description as shown in Figs. 2(b)-(d). The text description is made up of 4,312 unique words. The words vary with respect to their relationships with colors; some words are direct color words (e.g., pink, blue, etc.) while others evoke a particular set of colors (e.g., autumn or vibrant). To the best of our knowledge, there has been no dataset that matches a multi-word text and its corresponding 5-color palette. This dataset allows us to train our models for predicting semantically consistent color palettes with textual inputs.

Other Color Datasets Munroe's color survey [32] is a widely used large-scale color corpus. Based on crowd-sourced user judgment, it matches a text to a single color. While Munroe's color survey deals with one color, Kobayashi's Color Image Scale [5] is a well-established multi-color dataset. Kobayashi uses 180 adjectives to express 1170 three-color combination palettes. While both datasets are actively used in many applications, our task requires a different dataset that matches a text to multiple colors and is large enough for a deep learning model to learn from.

Data Collection We generated our PAT dataset by refining user-named palette data crawled from a community website called color-hex.com, where users upload custom-made color palettes for others to use. We crawled 47,665 palette-text

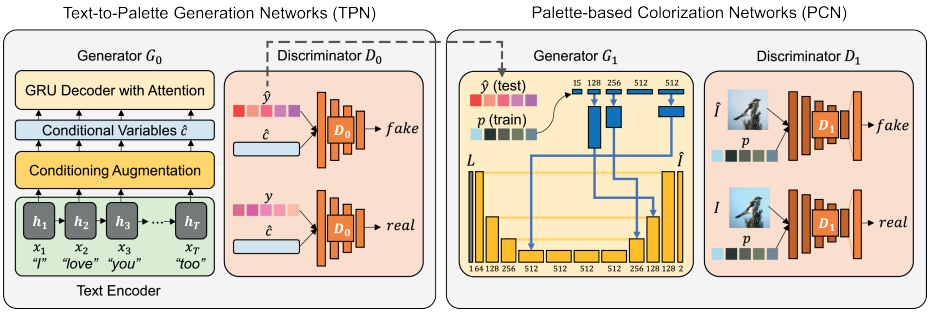


Fig. 3. Overview of our Text2Colors architecture. During training, generator G_0 learns to produce a color palette $\hat{y}$ given a set of conditional variables $\hat{c}$ processed from input text $x = \{x_1, \cdots, x_T\}$. Generator G_1 learns to predict a colorized output of a grayscale image L given a palette p extracted from the ground truth image. At test time, the trained generators G_0 and G_1 are used to produce a color palette from given text and then colorize a grayscale image reflecting the generated palette.

pairs and removed non-alphanumerical and non-English words. Among them, we found that users sometimes assign palette names in an arbitrary manner, missing their semantic consistency with their corresponding color palettes. Some names are a collection of random words (e.g., 'mehmeh' and 'i spilled tea all over my laptop rip'), or are riddled with typos (e.g., 'cause iiiiii see right through you boyyyyy' and 'greene gardn'). Thus, using unrefined raw palette names would hinder model performances significantly. To refine the noisy raw dataset, four annotators voted whether the text paired with the color palette properly matches its semantic meanings. We then used only the text-palette pairs in which at least three annotators out of four agreed that semantic matching exists. Mis-spelling and punctuation errors were manually corrected after the annotators finished sorting out the data.

The PAT dataset has several characteristics. First, it was refined according to annotators perceptions, which are inherently subjective. A text-palette pair perfectly plausible to one person may not be agreeable to another. We wanted to incorporate such subjectivity by allowing a diverse selection of text-palette pairs. Including text-palette pairs in the dataset only when all four annotators agree was found to be unnecessarily strict, leaving not much room for personal subjectivity. Thus, the text-palette pair was included in PAT when at least three annotators decided the text reflects the palette. Second, our dataset consists of short texts. As shown in Fig. 2, about 15% of the text descriptions in the dataset contain more than four words. During the sorting process, the annotators found it harder to reach a consensus for longer texts. This may be because as the text becomes longer, its scope of possible matching colors becomes broader and ambiguous. As a result, longer texts were more likely to be removed from the final dataset.

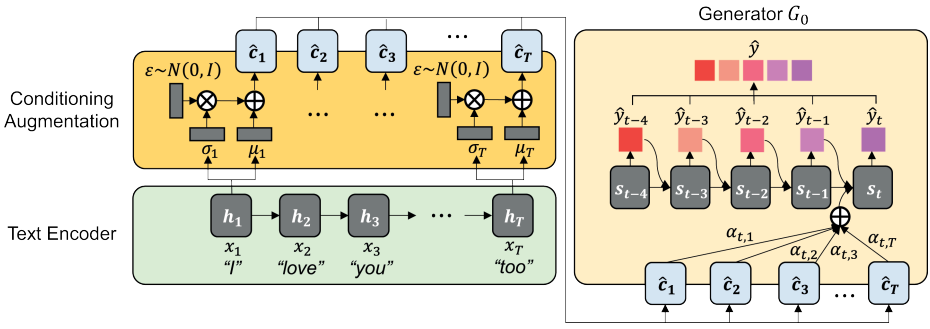


Fig. 4. Model architecture of a generator G_0 that produces the t-th color in the palette given an input text $x = \{x_1, \cdots, x_T\}$. Note that randomness is added to each hidden state vector h in the sequence before it is passed to the generator.

4 Text2Colors: Text-Driven Colorization

Our model consists of two networks: Text-to-Palette Generation Networks (TPN) and Palette-based Colorization Networks (PCN). We train the first networks to generate color palettes given a single text and then train the second networks to predict reasonable colorizations given a grayscale image and the generated palettes. We utilize conditional GANs (cGAN) for both networks.

4.1 Text-to-Palette Generation Networks (TPN)

Objective Function In this section, we illustrate the Text-to-Palette Generation Networks shown in Figs. 3 and 4, one of the first networks that produce reasonable color palettes associated with the text input. Let $x_i \in \mathbb{R}^{300}$ be word vectors initialized by 300-dimensional pre-trained vectors from GloVe [33]. Those words not included in the pre-trained set are initialized randomly. Using the CIE *Lab* space for our task, $y \in \mathbb{R}^{15}$ represents a 15-dimensional color palette consisting of five colors with *Lab* values. After a GRU text encoder encodes x into hidden states $h = \{h_1, \cdots, h_T\}$, we add variation to the encoded representation of text by sampling latent variables $\hat{c}$ from a Gaussian distribution $\mathcal{N}(\mu(h), \Sigma(h))$. Using the sequence of conditioning variables $\hat{c} = \{\hat{c}_1, \cdots, \hat{c}_T\}$ as *condition*, the objective function of the first cGAN can be expressed as

$$L_{D_0} = \mathbb{E}_{y \sim P_{data}}[\log D_0(\hat{c}, y)] + \mathbb{E}_{\hat{y} \sim P_{G_0}}[\log(1 - D_0(\hat{c}, \hat{y}))], \tag{1}$$

$$\begin{aligned} L_{G_0} = \mathbb{E}_{\hat{y} \sim P_{G_0}}[\log(1 - D_0(\hat{c}, \hat{y}))] + \lambda_H L_H(\hat{y}, y) \\ + \lambda_{KL} D_{KL}(\mathcal{N}(\mu(h), \Sigma(h)) \parallel \mathcal{N}(0, I)), \end{aligned} \tag{2}$$

where discriminator D_0 tries to maximize L_{D_0} against generator G_0 that tries to minimize L_{G_0}. y is sampled from the true color palette distribution P_{data} while $\hat{y}$ is sampled from the model distribution P_{G_0}. Previous approaches have benefited from mixing the GAN objective with L_2 distance [34] or L_1 distance [22]. We

have explored previous loss options and found the Huber (or smooth L_1) loss to be the most effective in increasing diversity among colors in generated palettes. The Huber loss is given by

$$L_H(\hat{y}, y) = \begin{cases} \frac{1}{2}(\hat{y} - y)^2 & \text{for } |\hat{y} - y| \leq \delta \\ \delta\,|\hat{y} - y| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases} \tag{3}$$

This loss term is added to the generator's objective function to be close to the ground truth palette while fooling the discriminator. λ_H and λ_{KL} are the hyperparameters to balance the three terms in Eq. 2. We set $\delta = 1, \lambda_H = 100, \lambda_{KL} = 0.5$ in our model.

Networks Architecture

Conditioning Augmentation. Learning a mapping from text to color is inherently multimodal. For instance, a text 'autumn' can be mapped to a variety of plausible color palettes. As the text becomes longer, such as 'midsummer to autumn' or 'autumn breeze and falling leaves', the scope of possible matching palettes becomes more broad and diverse. To appropriately model the multimodality of our problem, we utilize the conditioning augmentation (CA) [20] technique. Rather than using the fixed sequence of encoded representation $h = \{h_1, \cdots, h_T\}$ as input to our generator, we randomly sample latent variables $\hat{c}$ from a Gaussian distribution $\mathcal{N}(\mu(h), \Sigma(h))$ as shown in Fig. 4. This randomness allows our model to generate multiple plausible palettes given the same text input. We adopted the Kullback-Leibler (KL) divergence regularization term [20], i.e.,

$$D_{KL}(\mathcal{N}(\mu(h), \Sigma(h)) \parallel \mathcal{N}(0, I)),$$

which is then added to the objective function of the generator during training.

Generator. To obtain the set of conditioning vectors $\hat{c} = \{\hat{c_1}, \cdots, \hat{c_T}\}$, the pre-trained word vectors $x = \{x_1, \cdots, x_T\}$ is first fed into a GRU text encoder to compute hidden states $h = \{h_1, \cdots, h_T\}$. This text representation is fed into a fully-connected layer to generate μ and σ (the values in the diagonal of Σ) for the Gaussian distribution $\mathcal{N}(\mu(h), \Sigma(h))$. $\hat{c}$ is computed by $\hat{c} = \mu + \sigma \odot \epsilon$, where $\odot$ is the element-wise multiplication and $\epsilon \sim \mathcal{N}(0, I)$. The resulting set of vectors $\hat{c} = \{\hat{c_1}, \cdots, \hat{c_T}\}$ will be used as *condition* for our cGAN.

We design our color palette generator as a variant of a RNN decoder with attention mechanism [29,35]. The decoder is trained to predict the next color $\hat{y}_t$, given the conditioning vectors $\hat{c} = \{\hat{c_1}, \cdots, \hat{c_T}\}$ and all the previously predicted colors $\{\hat{y}_1, \cdots, \hat{y}_{t-1}\}$ of the palette $\hat{y}$. In other words, the decoder defines a probability over the color palette $\hat{y}$ by decomposing the joint probability into the ordered conditionals:

$$p(\hat{y}) = \prod_{t=1}^{T} p(\hat{y}_t \mid \{\hat{y}_1, \cdots, \hat{y}_{t-1}\}, \hat{c}), \tag{4}$$

where $\hat{y} = (\hat{y}_1, \cdots, \hat{y}_5)$. We define each conditional probability in Eq. (4) as

$$p(\hat{y}_i \mid \hat{y}_1, \cdots, \hat{y}_{i-1}, \hat{c}) = g(\hat{y}_{i-1}, s_i, c_i) \tag{5}$$

where s_i is a GRU hidden state vector for time i, computed by

$$s_i = f(s_{i-1}, \hat{y}_{i-1}, c_i).$$

The context vector c_i depends on a sequence of conditioning vectors $\hat{c} = \{\hat{c_1}, \cdots, \hat{c_T}\}$ to which the text encoder maps the text input. Each condition $\hat{c}_i$ contains the information about the entire text input with a strong focus on the i-th word with a slight variation. The context vector c_i is computed as the weighted sum of these conditions $\hat{c}_i$'s, i.e.,

$$c_i = \sum_{j=1}^{T} \alpha_{ij} \hat{c}_j. \tag{6}$$

The weight α_{ij} of each conditional variable $\hat{c}_j$ is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})} \quad \text{where} \quad e_{ij} = a\left(s_{i-1}, \hat{c}_j\right). \tag{7}$$

$$a\left(s_{i-1}, \hat{c}_j\right) = w^T \sigma(W^{(1)} s_{i-1} + W^{(2)} \hat{c}_j), \tag{8}$$

where $\sigma(\cdot)$ is a sigmoid activation function and w is a weight vector. The additive attention [35] $a\left(s_{i-1}, \hat{c}_j\right)$ computes how well the j-th word of the text input matches the i-th color of the palette output. The score α_{ij} is computed based on the GRU hidden state s_{i-1} and the j-th condition $\hat{c}_j$. This attention mechanism enables the model to effectively map complex text input to the color palette output.

Discriminator. For the discriminator D_0, the conditioning variable $\hat{c}$ and the color palette are concatenated and fed into a series of fully-connected layers. By jointly learning features across the encoded text and palette, the discriminator classifies whether the palettes are real or fake.

4.2 Palette-based Colorization Networks (PCN)

Objective Function The goal of the second networks is to automatically produce colorizations of a grayscale image guided by the color palette as a conditioning variable. The inputs are a grayscale image $L \in \mathbb{R}^{H \times W \times 1}$ representing the lightness in CIE *Lab* space and a color palette $p \in \mathbb{R}^{15}$ consisting of five colors in *Lab* values. The output $\hat{I} \in \mathbb{R}^{H \times W \times 2}$ corresponds to the predicted *ab* color channels of the image. The objective function of the second model can be expressed as

$$L_{D_1} = \mathbb{E}_{I \sim P_{data}}[\log D_1(p, I)] + \mathbb{E}_{\hat{I} \sim P_{G_1}}[\log(1 - D_1(p, \hat{I}))], \tag{9}$$

$$L_{G_1} = \mathbb{E}_{\hat{I} \sim P_{G_1}}[\log(1 - D_1(p, \hat{I}))] + L_H(\hat{I}, I). \tag{10}$$

D1 and G1 included in the equation are shown in Fig.3. We have also added the Huber loss to the generator's objective function. In other words, the generator learns to be close to the ground truth image with *plausible* colorizations, while incorporating palette colors to the output image to fool the discriminator.

Networks Architecture

Generator. The generator consists of two sub-networks: the main colorization networks and the conditioning networks. Our main colorization networks adopts the U-Net architecture [36], which has shown promising results in colorization tasks [22,10]. The skip connections help the networks recover spatial information [36], as the input and output images share the location of prominent edges [22].

The role of the conditioning networks is to apply the colors of the palette to the generated image. During training, the networks are given a palette $p \in \mathbb{R}^{15}$ extracted from the ground truth image I. We utilize the Color Thief [2] function to extract a palette consisting of five dominant colors of the ground truth image. Similar to the previous work [10], the conditioning palette p is fed into a series of 1×1 *conv-relu* layers as shown in Fig. 3. The feature maps in layers 1, 2, and 4 are duplicated spatially to match the spatial dimension of the *conv*9, *conv*8, and *conv*4 features in the main colorization networks and merged by element-wise addition. The conditioning variable is fed into upsampling layers with the skip connection as well as the middle of the main networks. This allows the generator to detect prominent edges and apply palette colors to suitable locations of the image. During test time, we use the generated palette $\hat{y}$ from the first networks (TPN) as the conditioning variable, colorizing the grayscale image with the predicted palette colors.

Discriminator. As our discriminator D_1, we use a variant of the DCGAN architecture [37]. The image and conditioning variable p are concatenated and fed into a series of *conv-leaky relu* layers to jointly learn features across the image and the palette. Then, it is fed into a fully-connected layer to classify whether the image is real or fake.

4.3 Implementation Details

We first train D_0 and G_0 of TPN for 500 epochs using the PAT dataset. We then train D_1 and G_1 of the PCN for 100 epochs, using the extracted palette from a ground truth image. Finally, we use the trained generators G_0 and G_1 during test time to colorize a grayscale image with generated palette $\hat{y}$ from a text input x. All networks are trained using the Adam optimizer [38] with a learning rate of 0.0002. Weights were initialized from a Gaussian distribution with zero mean and standard deviation of 0.05. We set other hyper parameters as $\delta = 1, \lambda_H = 100$, and $\lambda_{KL} = 0.5$.

[2] http://lokeshdhakar.com/projects/color-thief/

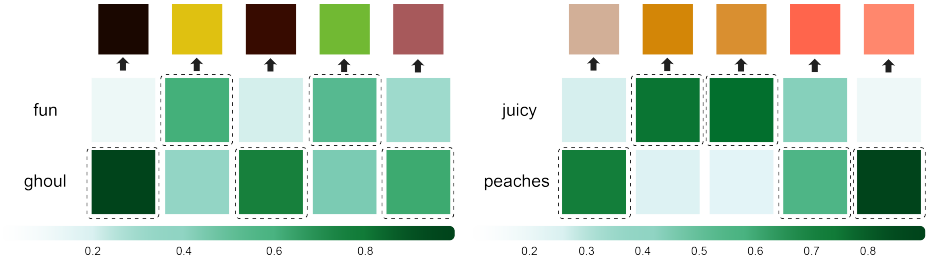


Fig. 5. Attention analysis: Attention scores measured by the TPN for two text input samples. Each box color (in green) denotes the attention score computed in producing the corresponding color shown on top. The dashed-line boxes indicate the word that each color output attended.

5 Experimental Results

This section presents both quantitative and qualitative analyses of our proposed model. We evaluate the TPN (Section 4.1) based on our PAT dataset. For the training of the PCN (Section 4.2), we use two different datasets, CUB-200-2011 (CUB) [39] and ImageNet ILSVRC Object Detection (ImageNet dataset) [40].

5.1 Analysis on Attention Outputs

The attention module (Section 4.1) plays a role of attending to particular words in text input to predict the most proper colors that match the text input. Fig. 5 illustrates how the predicted colors are influenced by attention scores. The green-colored boxes show attention scores computed for each word token when predicting each corresponding color in the palette. Higher scores are indicated by dashed-line boxes. We observe that three colors generated by attending to *ghoul* are all dark and gloomy, while the other two colors attending to *fun* are bright. This attention mechanism enables our model to thoroughly reflect the semantics included in text inputs of various lengths.

5.2 Analysis on Multimodality and Diversity of Generated Palettes

In this section, we will evaluate multimodality and diversity of our generated palettes. Multimodality refers to how many different color palettes a single text input can be mapped to. In other words, if a single text can be expressed with more color palettes, the more multimodal it is. Diversity within a palette refers to how diverse the colors included in a single palette are. Following the current standard for perceptual color distance measurement, we use the CIEDE2000 [41] on CIE *Lab* space to compute a model's multimodality and diversity. To measure multimodality, we compute average pairwise distances between colors of two different palettes for every generated palette combination. To measure diversity of a color palette, we measure the average pairwise distance between the five colors within a palette. All measurements are computed based on the test dataset.

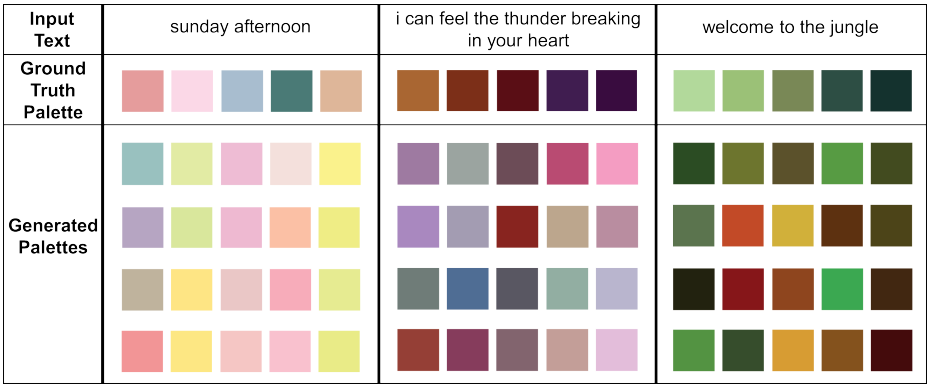


Fig. 6. Qualitative analysis on multimodality: Our TPN can generate diverse palettes when given the same text input.

Table 1. Quantitative analysis results

		Palette Evaluation				User Study: Part I			
Model Variations		**Diversity**		**Multimodality**		**Fooling Rate (%)**			
Objective Function	CA	Mean	Std	Mean	Std	Mean	Std	Max	Min
cGAN+Huber($\lambda_H = 100$)	X	19.36	8.74	0.0	0.0	-	-	-	-
cGAN+No Huber($\lambda_H = 0$)	O	19.54	8.52	19.06	8.06	30.7	13.3	57.5	17.4
No cGAN+Huber($\lambda_H = 100$)	O	**22.16**	9.53	**19.39**	7.81	52.7	13.9	61.6	**42.1**
cGAN+Huber($\lambda_H = 100$)	O	20.82	10.11	18.18	8.11	**56.2**	14.4	**76.7**	37.1

Results. Table 1 shows the diversity and multimodality measurement among the variants of our model. The CA module (see Section 4.1 for details) enables our networks to suggest multiple color palettes when given the same text input. The model variant without CA (first row in Table 1) results in zero multimodality, indicating that the networks generate identical palettes for the same text input. Among the variants with CA, the highest score on diversity and multimodality is achieved without cGAN, but in the expense of a lower fooling rate. Results on the fooling rate will be further illustrated in Section 5.3.

5.3 User Study

We conduct a user study to reflect universal user opinions on the outputs of our model. Our user study is composed of two parts. The first part measures how the generated palettes match the text inputs. The second part is a survey that compares the performance of our palette-based colorization model to another state-of-the-art colorization model. 53 participants took part in our study.

Part I: Matching between text and the generated palettes Our goal is to generate a palette with a strong semantic connection with the given text input.

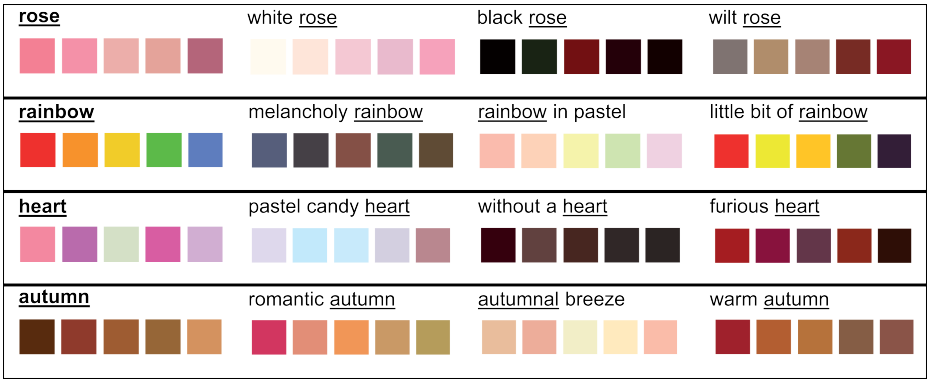


Fig. 7. Qualitative analysis on semantic contexts: Our model reflects the subtle nuance differences in semantic context of a given text input through color palettes. Except for the first column, all the text combinations included in this figure are unseen data.

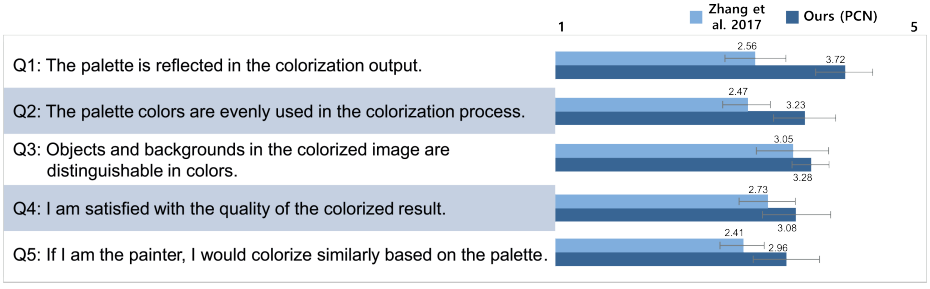


Fig. 8. Colorization performance comparisons: Mean and standard deviation values for each question are reported for the baseline [10] and our PCN.

A natural way to evaluate it is to quantify the degree of connection between the text input and the generated palette, in comparison to the same text input and its ground truth palette. Given a text input, its generated palette, and the ground truth palette, we ask human observers to select the palette that best suits the text input. A fooling rate (FR) in this study indicates the relative number of generated palettes chosen over ground truth palettes. More people choosing the generated palette results in a higher FR. This measure has often been used to assess the quality of colorization results [10,42]. We will use this metric to measure how much a text input matches its generated palette.

Study Procedure. Users participate in the user study over three variants of our TPN. Each of the three variants consists of 30 evaluations. We randomly choose a single data item out of 992 test data and show the text input along with the generated palette and the ground truth palette.

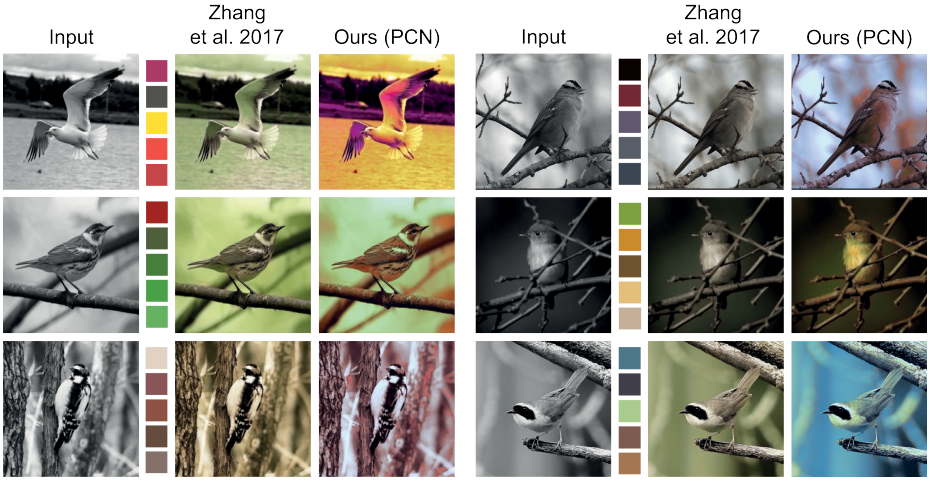


Fig. 9. We compare the colorization results guided by palettes. The five-color palette used for colorization is shown next to the input grayscale image. Note that our PCN performs better at applying various colors included in the palette.

Results. In Table 1, we measure the FR score for each person and compute the mean and the standard deviation (std) of all of the scores from participants. Max and min scores represent the highest and the lowest FR scores, respectively, recorded by a single person. Users record the highest FR when the model contains all the cGAN, the Huber loss, and the CA module, while maintaining a high level of diversity and multimodality. The FR of 56.2% indicates that the generated palettes are indistinguishable to human eyes and sometimes even match the input text better than the ground truth palettes. Note that the standard deviation of 14% implies diverse responses to the same data pairs.

Part II: Colorization Comparisons In this part of the user study, we conduct a survey on the performance of the PCN given palette inputs. Users are asked to answer five questions based on the given grayscale image, the color palette, and the colored image. For quantitative comparison, we set a state-of-the-art colorization model [10] as our baseline. This model is originally composed of local and global hint networks. In our implementation of the baseline model, we utilize the global hint networks to infuse our generated palette to the main colorization networks. Note that we modified the baseline model to fit our task. For the palette, we take the five most used colors out of the quantized 313 bins of the ground truth image rather than the entire global color distribution. The major difference between our PCN and the baseline is that we infuse the palette hint in several different layers and employ an adversarial training strategy.

Study Procedure. We show colorization results of our PCN and the baseline model one-by-one in a random order. Then, we ask each participant to answer

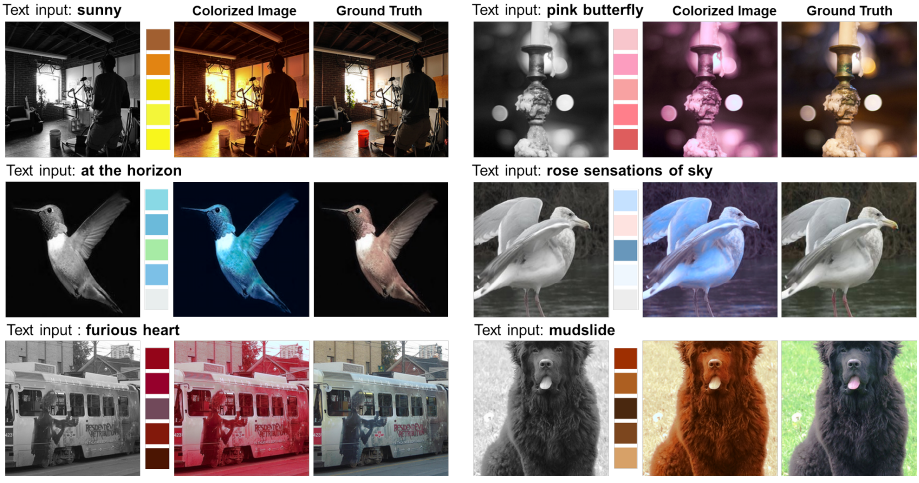


Fig. 10. Colorization results from text inputs to our model. The text input is shown above the input grayscale image, and the generated palettes are on the right of the grayscale image. We can see that the color palette is well-reflected in the colorized image when compared to the ground truth image.

five different questions (shown in Fig. 8) based on a five-point Likert scale. The focus of our questions is to evaluate how well the palette was used in colorizing the given grayscale image. The total number of data samples per test is 15.

Results. The resulting statistics are reported in Fig. 8. Our PCN achieves higher scores than the baseline model across all of the questions. We can infer that the palettes generated by our model are preferred over palettes created by a human hand. Since our model learns consistent patterns from a large number of human-generated palette-text pairs, our model may have generated color palettes that more users could relate to.

6 Conclusion

We proposed a generative model that can produce multiple palettes from rich text input and colorize grayscale images accordingly. Evaluation results confirm that our TPN can generate plausible color palettes from text input and can incorporate the multimodal nature of colors. Qualitative results on our PCN also show that the diverse colors in a palette are effectively reflected in the colorization results. Future work includes extending our model to a broader range of tasks requiring color recommendation and conducting a more detailed analysis of our dataset.

References

1. Heer, J., Stone, M.: Color naming models for color selection, image editing and palette design. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI), ACM (2012) 1007–1016
2. Chuang, J., Stone, M., Hanrahan, P.: A probabilistic model of the categorical association between colors. In: Proceedings of the IS & T Color and Imaging Conference (CIC). Volume 2008., Society for Imaging Science and Technology (2008) 6–11
3. Kawakami, K., Dyer, C., Routledge, B.R., Smith, N.A.: Character sequence models for colorful words. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2016)
4. Monroe, W., Hawkins, R.X., Goodman, N.D., Potts, C.: Colors in context: A pragmatic neural model for grounded language understanding. Transactions of the Association of Computational Linguistics (ACL) (2017)
5. Kobayashi, S.: Color image scale. http://www. ncd-ri. co. jp/english/main_0104. html (2009)
6. Liu, Y., Cohen, M., Uyttendaele, M., Rusinkiewicz, S.: Autostyle: Automatic style transfer from image collections to users' images. In: Computer Graphics Forum. Volume 33., Wiley Online Library (2014) 21–31
7. Solli, M., Lenz, R.: Color semantics for image indexing. In: Proceedings of Conference on Colour in Graphics Imaging and Vision (CGIV). Volume 2010., Society for Imaging Science and Technology (2010) 353–358
8. Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: Proceedings of the IEEE European Conference on Computer Vision(ECCV), Springer (2008) 126–139
9. Xiao, Y., Zhou, P., Zheng, Y.: Interactive deep colorization with simultaneous global and local inputs. arXiv preprint arXiv:1801.09083 (2018)
10. Zhang, R., Zhu, J.Y., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A.: Real-time user-guided image colorization with learned deep priors. ACM Transactions on Graphics (TOG) (2017)
11. Crozier, W.: The psychology of colour preferences. Coloration Technology **26**(1) (1996) 63–72
12. De Bortoli, M., Maroto, J.: Colours across cultures: Translating colours in interactive marketing communications. In: Proceedings of the European Languages and the Implementation of Communication and Information Technologies (ELICIT). (2001) 3–4
13. Cho, J., Yun, S., Lee, K., Choi, J.Y.: Palettenet: Image recolorization with given color palette. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2017) 62–70
14. Jahanian, A., Keshvari, S., Vishwanathan, S., Allebach, J.P.: Colors–messengers of concepts: Visual design mining for learning color semantics. ACM Transactions on Computer-Human Interaction (TOCHI) **24**(1) (2017) 2
15. Murray, N., Skaff, S., Marchesotti, L., Perronnin, F.: Toward automatic and flexible concept transfer. Computers & Graphics **36**(6) (2012) 622–634
16. McMahan, B., Stone, M.: A bayesian model of grounded color semantics. Transactions of the Association of Computational Linguistics (ACL) **3**(1) (2015) 103–115
17. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)

18. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. Proceedings of the International Conference on Machine Learning (ICML) (2016)
19. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 49–58
20. Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., Metaxas, D.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). (2017) 5907–5915
21. Kim, T., Cha, M., Kim, H., Lee, J., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. Proceedings of the International Conference on Machine Learning (ICML) (2017)
22. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
24. Li, X., Zhao, H., Nie, G., Huang, H.: Image recoloring using geodesic distance based color harmonization. Computational Visual Media **1**(2) (2015) 143–155
25. Chang, H., Fried, O., Liu, Y., DiVerdi, S., Finkelstein, A.: Palette-based photo recoloring. ACM Transactions on Graphics (TOG) **34**(4) (2015) 139
26. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). (2015) 1422–1432
27. Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks. In: Proceedings of the International Conference on Machine Learning (ICML). (2011) 1017–1024
28. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems (NIPS). (2014) 3104–3112
29. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2015)
30. Ahn, H., Ha, T., Choi, Y., Yoo, H., Oh, S.: Text2action: Generative adversarial synthesis from language to action. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2017)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS). (2017) 6000–6010
32. Munroe, R.: Color survey results. Online at http://blog.xkcd.com/2010/05/03/color-surveyresults (2010)
33. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). (2014) 1532–1543
34. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 2536–2544

35. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. Proceedings of the International Conference on Learning Representations (ICLR) (2014)
36. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Springer (2015) 234–241
37. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. Proceedings of the International Conference on Learning Representations (ICLR) (2015)
38. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. Proceedings of the International Conference on Learning Representations (ICLR) (2014)
39. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Technical report (2011)
40. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision (IJCV) **115**(3) (2015) 211–252
41. Sharma, G., Wu, W., Dalal, E.N.: The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. Color Research & Application **30**(1) (2005) 21–30
42. Guadarrama, S., Dahl, R., Bieber, D., Norouzi, M., Shlens, J., Murphy, K.: Pixcolor: Pixel recursive colorization. Proceedings of the British Machine Vision Conference (BMVC) (2017)
43. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Proceedings of the IEEE European Conference on Computer Vision (ECCV), Springer (2016) 649–666

Appendix

7 User Study Samples

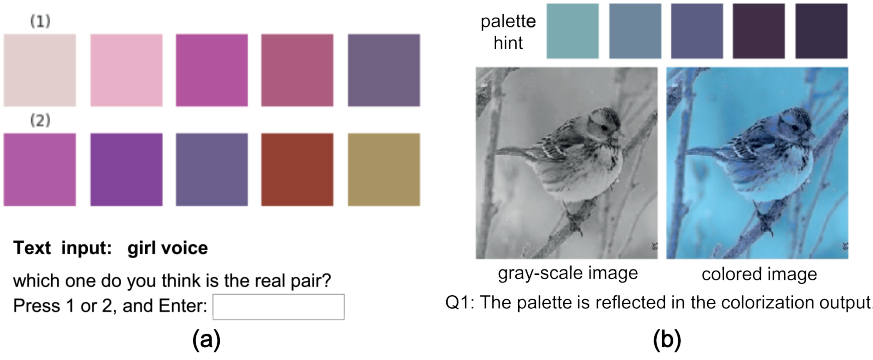


Fig. 11. UI design of our user study.

Our user study consists of two parts, one for evaluation of Text-to-palette Generation Networks (TPN) and the other for evaluation of Palette-based Colorization Networks (PCN). Fig. 11**(a)-(b)** illustrates how our data tuples were shown to the participants in **Part I** and **Part II**, respectively. Fig. 11(a) shows how we evaluate TPN networks. After about five seconds, the sample disappears, and users are asked to select the palette that better matches the text input. Before the test begins, participants go through a short practice session to properly understand the task. Among the practice session samples, we put one highly distinguishable palette pair on purpose to check if the users understood the task. 92% of users chose the ground truth palette over the intentionally mismatching palette. This confirms that a majority of participants understood the test rules. The ground truth palettes are shown to the user shortly after every iteration in the practice session, but during the real test, we do not disclose the ground truth palette.

8 Text-to-Palette Generation Networks (TPN)

8.1 Model Comparisons for Learning Global Color Distributions

Fig. 12 shows comparisons of color distributions between ground truth palettes of the training data and generated palettes from our test data. For each color distribution, we quantize the ab values of every palette color into 313 color bins [43] and visualize the probability distribution of ab values. We compare three model variants of different objective functions: cGAN+Huber (λ_H=100), Huber (λ_H=100), and cGAN (λ_H=0).

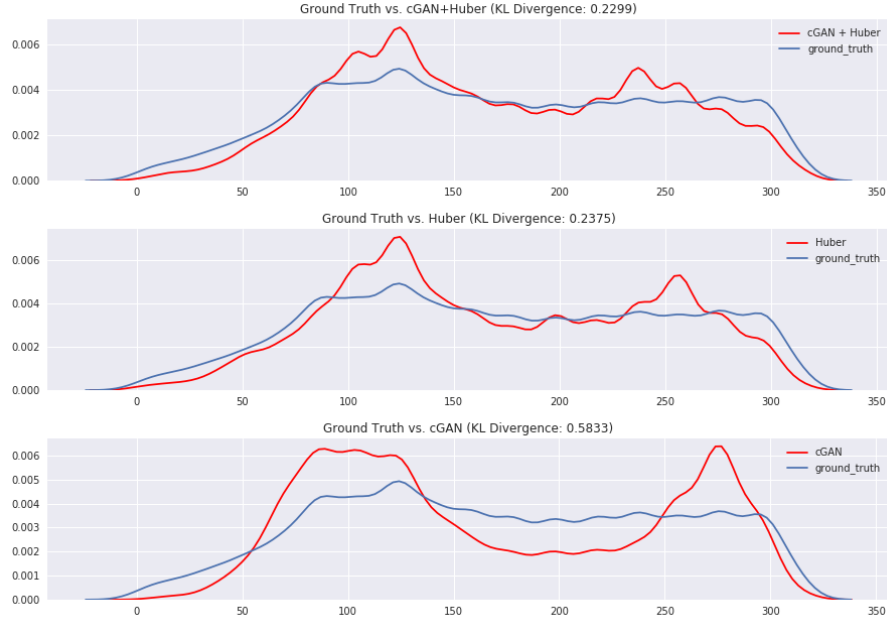


Fig. 12. Color Distribution Comparisons: Red lines correspond to color distributions of generated palettes from three model variants. The blue lines denote the ground truth color distribution of the training data. The KL divergence of the three distribution pairs are computed as 0.2299, 0.2375, and 0.5833 in order.

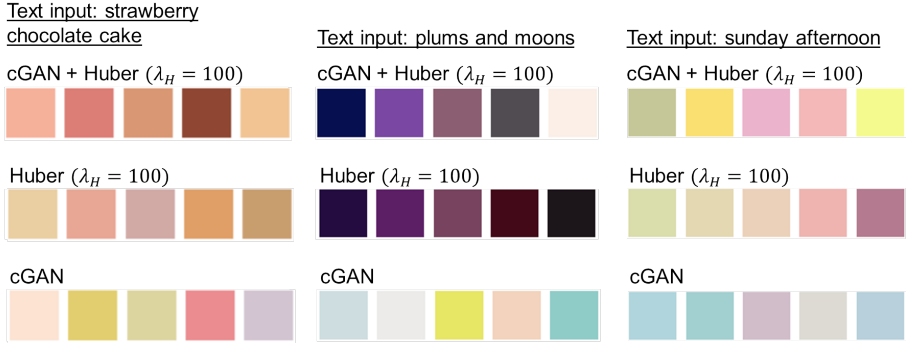


Fig. 13. Comparison of palette prediction results from different model variations.

We also compute the Kullback-Leibler (KL) divergence between the ground truth palette distribution of the training data and that of our model variants. As shown in the bottommost plot of Fig. 12, the Huber loss plays a critical role in producing proper colors close to the ground truth image. Without the Huber loss, the model does not only fail to recover the color distribution similar to the ground truth data but also records the lowest fooling rate of 30.7% in user

study results. On the other hand, the model with cGAN+Huber loss (λ_H=100) records the lowest KL divergence of 0.2299 as well as the best fooling rate of 56.2%, while the model with only the Huber loss (λ_H=100) records the second best. This is due to the fact that only using the Huber loss leads to blindly averaging over multiple ground truth palettes, resulting in slightly desaturated palette results as shown in the second row of Fig. 13. In contrast, the model with both cGAN+Huber loss learns and preserves various ground truth colors rather than simply averaging them, resulting in more bright, highly saturated results as shown in the first row of Fig. 13.

8.2 Additional Color Palette Generation Examples

This section shows additional, diverse and detailed results from the TPN.



Fig. 14. Our TPN can handle phrase-level inputs. This figure shows TPN outputs on diverse phrases about 'love.'

In Fig. 14, we show how our model handles phrase-level inputs. To make comparison easier, all the input phrases are about 'love.' It is interesting to see how our model chooses to express the subtle nuance differences included in the input text. Notice how the output color palettes tend to be darker for text inputs that are negative towards 'love' (e.g., 'i thought i loved you' and 'where did our love go'). All input phrases included in this figure are unseen data except for 'i love you.'

The results in Fig. 15 show outputs of our model in comparison to ground truth palettes. If an input word is seen at least once in the training data, our model is able to output a color palette related to the input word. For instance, let us take a look at the color palette named 'mango and grapefruit' on the top left. The word 'grapefruit' is included only once in the training set. Yet, the model successfully outputs a color palette that matches the text input. Also, ground truth palettes are included for a direct comparison with generated palettes. Even if the predicted palette is not exactly identical to the ground truth palette, both can be perceived as reasonable colors, which shows the generalization capability of our model.



Fig. 15. Palette predictions from the test set.

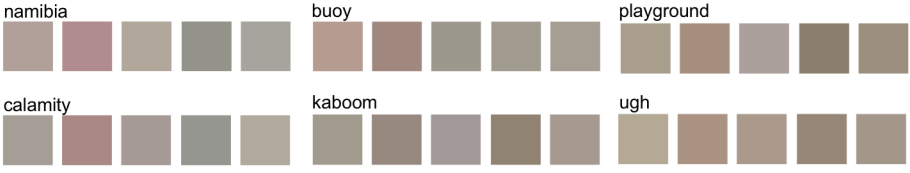


Fig. 16. Failed results from TPN: our model fails and outputs the same washed-out grayish-brown color palettes when given unknown tokens.

Even though our model can effectively produce semantically meaningful colorizations, it struggles when unknown tokens are given as input. Unknown tokens refer to words not included in the training set at all. It is not surprising that our model fails and outputs the same washed-out grayish-brown palettes as seen in Fig. 16. On the other hand, our model can still produce reasonable palettes in the case of unseen, new combinations of words found in the training set. For example, 'bright life' in Fig. 15 was seen separately as 'bright' and 'life' in the training set but not together. Thus, 'bright life' is classified as unseen data, which our model has no problem in predicting color palettes from.

9 Palette-based Colorization Networks (PCN)

We present additional colorization results on datasets including CUB-200-2011 (CUB dataset) [39], ImageNet ILSVRC Object Detection (ImageNet dataset) [40], and Graphical Pattern images (Pattern images) in Figs. 17-22. In these figures, the leftmost columns represent input grayscale images. Text inputs are given

above the grayscale image. The vertical color palettes next to the grayscale images are those palettes generated from the text input. The output has been colorized with the generated color palette. These results show that our model effectively utilizes the generated color palettes during the colorization process. The colorized image may be different from its natural colors because our networks incorporate additional color hints. We display the original ground truth image on the right to compare how different an image becomes after applying the palettes.

9.1 CUB-200-2011

Figs. 17 and 18 show additional colorization results on the CUB dataset.

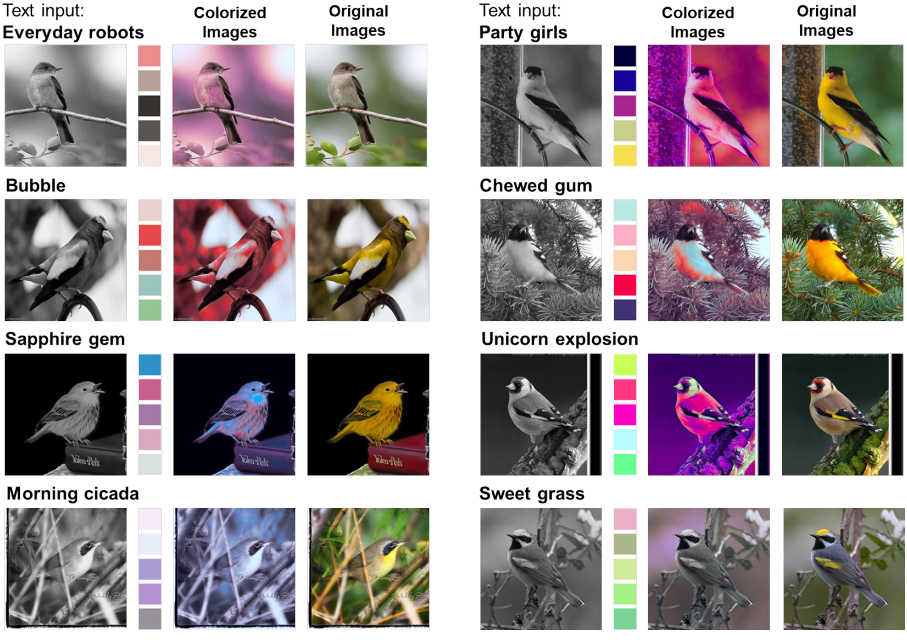


Fig. 17. Results on CUB dataset (1).

9.2 ImageNet ILSVRC Object Detection

Figs. 19 and 20 show additional colorization results on the ImageNet dataset.

9.3 Graphical Pattern Images

Our PCN model generalizes surprisingly well on other types of images. Our model is trained on ImageNet dataset, which is mostly made up of natural images.

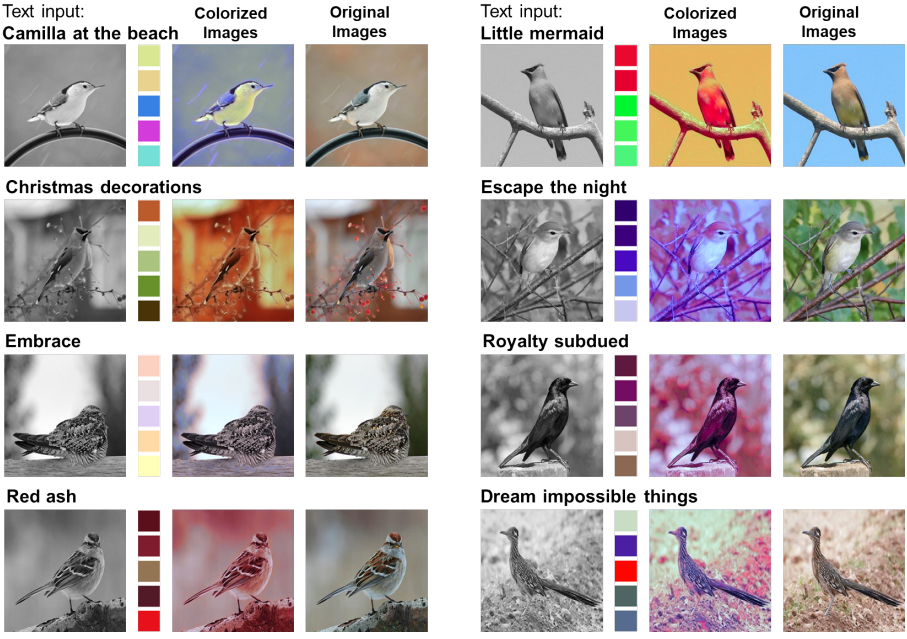


Fig. 18. Results on CUB dataset (2).

Instead of natural images, we used our colorization model to colorize graphical pattern images. The graphical pattern images are crawled from Google search with keywords such as 'pattern,' 'fabric pattern.' and 'beautiful patterns.' As seen in Figs. 21 and 22, graphical pattern images are quite different from natural images. The colorized outputs show that our model can apply our generated color palettes to images of diverse shapes and textures. The results qualitatively show that our palette-based colorization model is transferable to other image domains.

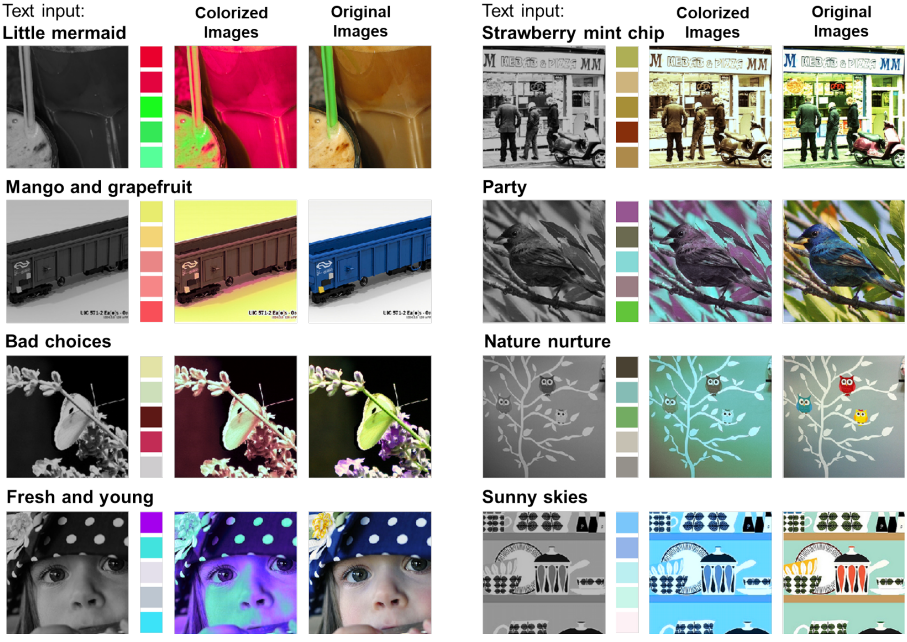


Fig. 19. Results on ImageNet dataset (1).

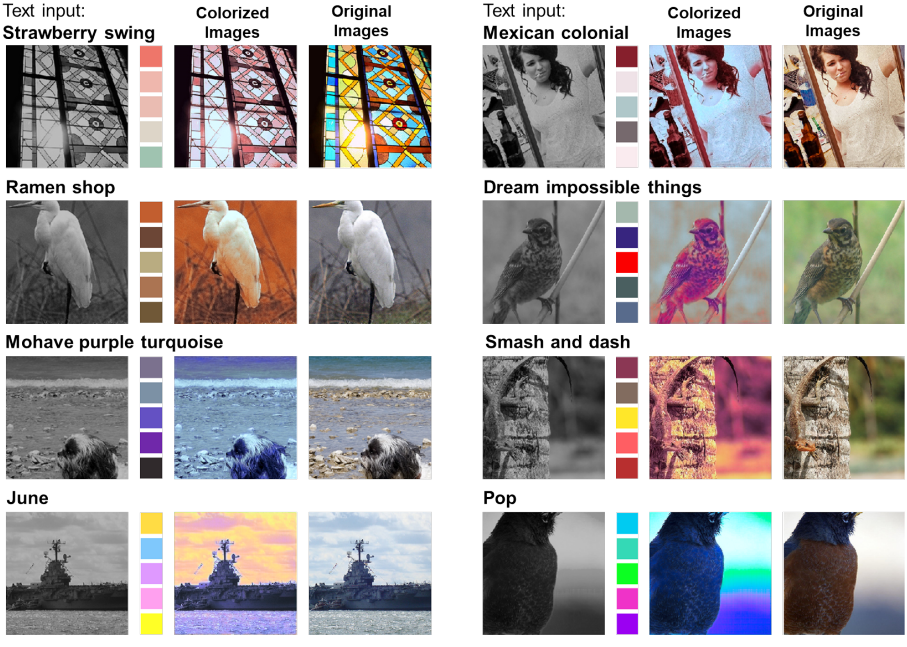


Fig. 20. Results on ImageNet dataset (2).

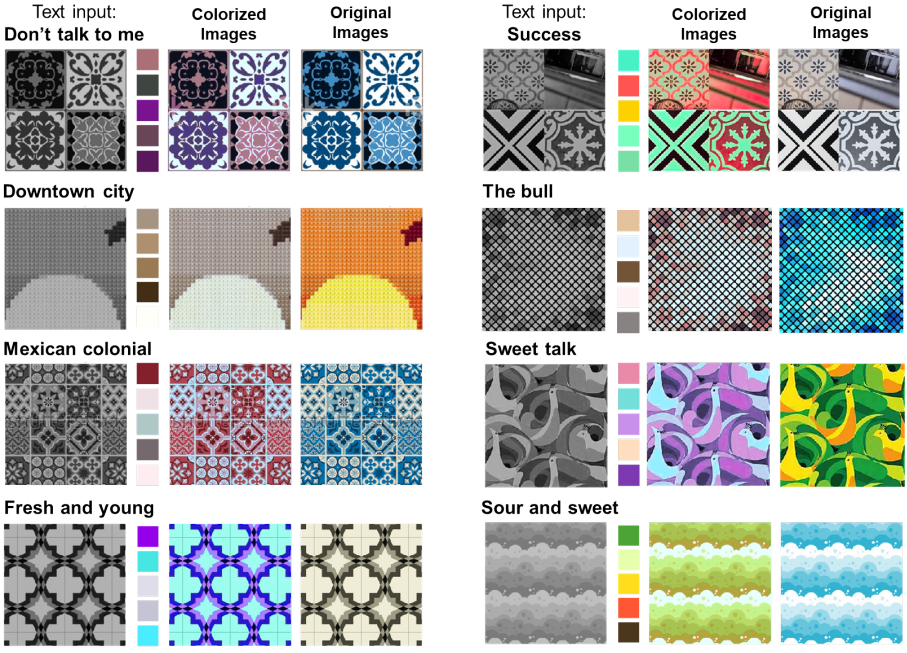


Fig. 21. Results on graphical pattern images (1).

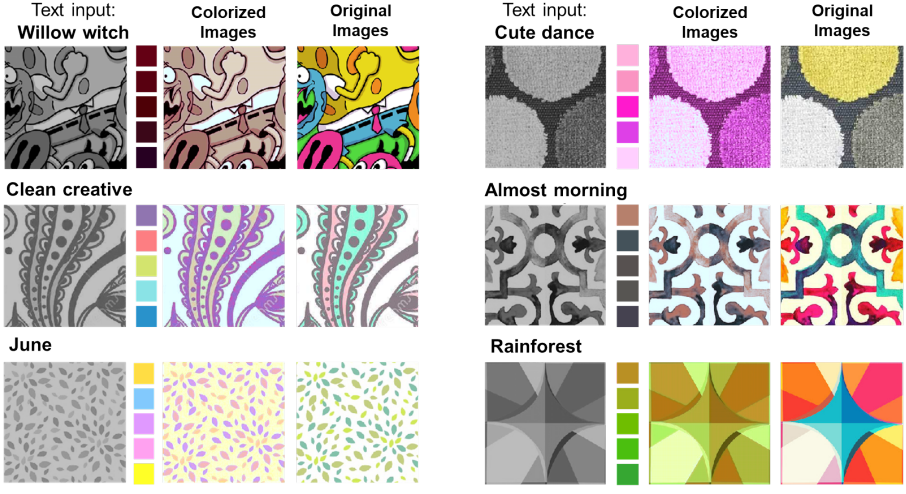


Fig. 22. Results on graphical pattern images (2).