IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 16, 2020, accepted November 16, 2020, date of publication November 26, 2020,
date of current version December 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3040737

Image Colorization Using the Global Scene-Context Style and Pixel-Wise Semantic Segmentation

TRAM-TRAN NGUYEN-QUYNH, SOO-HYUNG KIM, (Member, IEEE), AND NHU-TAI DO, (Graduate Student Member, IEEE)

Department of Artificial Intelligence Convergence, Chonnam National University, Gwangju 61186, South Korea

Corresponding author: Soo-Hyung Kim (shkim@jnu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2018R1D1A3A03000947 and Grant NRF-2020R1A4A1019191.

ABSTRACT In this paper, we present an encoder-decoder architecture that exploits global and local semantics for the automatic image colorization problem. For the global semantics, the low-level encoding features are fine-tuned by the scene-context classification to integrate the global image style. Moreover, the architecture deals with the uncertainty and relations among the scene styles based on the label smoothing and pre-trained weights from Places365. For local semantics, three branches learn the mutual benefits at the pixel-level, in which average and multi-modal distributions are respectively created from regression and soft-encoding branches, while the segmentation branch determines to which object the pixel belongs. Our experiments, which involve training with the Coco-Stuff dataset and validation on DIV2K, Places365, and ImageNet, show that our results are very encouraging.

INDEX TERMS Image colorization, soft-encoding, u-net, scene-context classification, semantic segmentation.

I. INTRODUCTION

Colorizing a gray-scale image not only brings a lot of special semantics into that image, but also helps the image become more vivid and emotional [1]. An image often contains many objects, and a human can easily decide which color should belong to each object in the image using their knowledge about object meaning. However, most objects do not merely have one color; for example, a shirt can be red, blue, yellow, or many other colors. Humans will also predict object colors based on a certain amount of subjective emotion, which is one of the biggest challenges for a machine that is tasked with mimicking both the knowledge and feeling of humans.

Because of this, previous colorization systems have often involved a joint partnership with users, such as providing some prior knowledge by inputting color points [2] or transferring colors from reference the gray-scale images. Choosing an image that contains the correct object details is too time-consuming, as is drawing color points. Therefore, automatic colorization based on the deep convolutional neural network without human involvement has been introduced [3]–[6]. Given the vast benefits of the auto colorization system, researchers have found many ways to make computers understand the semantic information of images during process to achieve effective auto-coloring results. One typical method, [7] used a scene classification from the Places365 database to train a model about the global semantics of images. Moreover, [8] focused on both the semantic composition and the local objects of a scene to color arbitrary images using the VGG-16 network architecture. In the method presented by Zhang *et al.* [9], they added to their model the ability to compile the object-level semantics with cross-channel encoding. In the above works, the semantics were only discovered at the image level. By contrast, current studies on semantic segmentation have reached the pixel level by assigning labels to each pixel [10]. In the same way, colorization is also a task that involves assigning colors to each pixel based on a color probability distribution [11]. To improve the colorization method by combining these concepts, aside from scene meaning, we recognize that the

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Sharif.

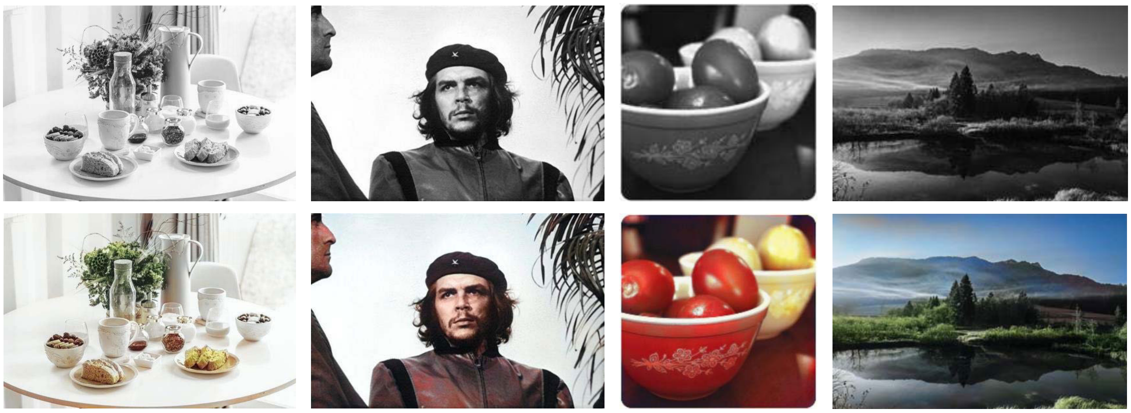


FIGURE 1. Some image results were colored by our proposed colorization method.

semantic segmentation task also plays a vital role in providing semantic in pixel-level.

Our life experiences play an essential role in developing our information cognition model. Information, which includes global and local types, is saved in our subconscious and gradually form knowledge over time. When seeing an image, we can analyze the scene type, which objects are likely to be present, their material types, and more. Color knowledge is one of the types of information that we store in our brains throughout our lives. Therefore, it is easy for human to fill a gray-scale picture with colors. However, there are still many problems with how a machine automatically chooses the correct colors in painting black-and-white images. The computer only calculates the prediction based on light and color correlation without regard to the object semantics leading to color confusion. The inputted pictures have many different sizes, so it is impossible to avoid scale variation, which leads to color bleeding among boundaries and color noises.

An example of the process is shown in Fig. 2 viewers can observe a gray-scale input and easily guess some overview information, such as the fact that it is outdoor scene having the sky and the field, this is called scene semantic or global information. In addition, the objects, such as the people, the kite, the grass, and the sky, are called local information with semantic segmentation. The combination of global and local knowledge in scene semantic and semantic segmentation will help the observers determine what the objects are in the image and reference object models about the colors of the objects.

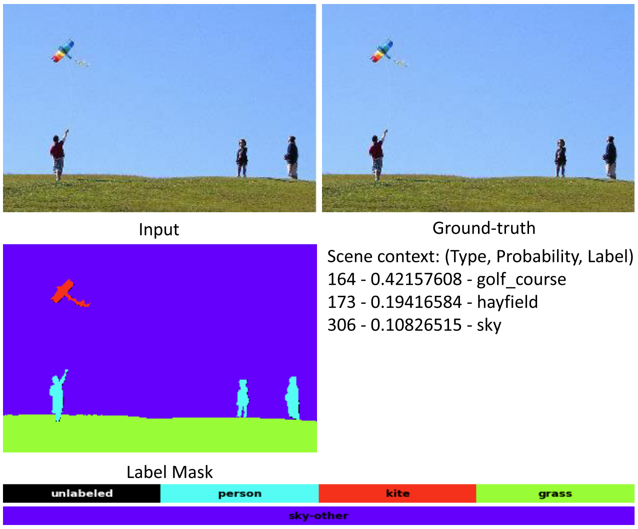

Input
Ground-truth
Scene context: (Type, Probability, Label)
164 - 0.42157608 - golf_course
173 - 0.19416584 - hayfield
306 - 0.10826515 - sky
Label Mask
unlabeled
person
kite
grass
sky-other

FIGURE 2. Semantic segmentation and scene-context classification in the colorization problem. Given a gray-scale image, we colorize that image to produce a color image (ground-truth) using semantic segmentation at pixel-level (where pixels belong to items such as the kite, a person, grass, etc.) and scene-context in the global style (meanings of the image based on probability such as golf-course 42%, hay-field 19%, sky 11% and others 28%).

For this reason, we suggest a solution using both segmentation and scene semantics to color images automatically, as shown in Fig. 1. We are also interested in an appropriate network that encourages these elements. In our paper, we use the encoder-decoder architecture, as shown in Fig. 3, due to the flexibility of the network in multi-task learning. The low-level encoding features will be fine-tuned by the scene-context classification branch in the middle of the encoder and decoder branches. This means they contain not only the encoding data of the input image, but also the global style of the image, to help the decoder branch colorize images more precisely based on the global semantics of the scene-context. For the pixel-level semantics, the output of the decoder branch, which is in the form of 2D pixel mapping features will be fine-tuned by the segmentation, regression, and soft-encoding branches. Three branches will learn the mutual benefits for the pixel-level meaning in segmentation, the average color model in regression, and the bias color model in soft-encoding to build robust 2D pixel mapping features. In the implementation details, we use the U-net [12] with the skip connection to help our proposed model converge better and to avoid the vanishing problem caused by the dead activation and deep network problems.

The main contributions in this study are:

- Propose an encoder-decoder architecture that uses the scene-context classification and pixel-wise segmentation for image colorization.

- For the scene-context semantics, we deal with the uncertainty and the relations among scenes by label smoothing using the prior probabilities from the pretrained-weight in Places365. For the pixel-level semantics, we integrate the mutual benefit learning from the pixel-wise segmentation model, the average color model, and the bias color model.
- We train the architecture on the Coco-Stuff dataset [13] and validate it on DIV2K [14], Places365 [15], ImageNet [16], and Internet collection images to ensure it achieves the good results.

The rest of our paper includes five sections. In section 2, we mention prior works related to the image colorization problem. Next, in section 3, we describe our network architecture, the colorization processing data used for input, output, and semantics, and the multi-loss function. Section 4 describes the implementation details of our proposed method. The details of the experiments are given in Section 5 along with a discussion of the results. Finally, section 6 concludes our paper.

II. RELATED WORKS

Coloring old black and white photos to recreate historical scenes can rescue a lot of cultural and memorial values from the ravages of time and bring modern people closer to the past. Colorization remains still one of the more exciting topics among users of image editing software, as demonstrated by the many coloring guides on the Internet. However, colorization using image editing software is costly and time-consuming. A digital artist must spend a lot of time to colorize old black and white photos, as he or she must start by making many layers from the different regions of the image, and then assign and adjust the colors for these regions to be more suitable based on his or her own knowledge and imagination. Computer researchers have launched many coloring applications in attempts to solve the various difficulties associated with this type of colorization.

Previous studies have often referred to three conventional approaches of colorization. The first kind is scribble-based, which uses some color annotations on the image; the second one obtains colors from a reference image to apply to a target image; and the last one is automatic colorization based on deep learning.

A. COLORIZATION BASED ON SCRIBBLES

This is a colorization method that involves automatically propagating the colors of the initial scribbles to the same color neighbor pixels. Levin *et al.* [2] initiated this method in 2004 based on the hypothesis that surrounding pixels with similar luminance will also have similar colors. Still, their results showed color spilling status between objects. After that, the method described in [17] alleviated this situation by exploring edge information at the local level to prevent colors from bleeding over boundaries. Some methods, such as [18] and [19] promoted efficiency by propagating colors, not only to neighboring pixels but also pixels having similar patterns. Although images colorized using these methods will be attractive to observers with natural and intensive detailed colors, they require substantial amounts of time and labor.

B. COLORIZATION BASED ON EXAMPLES

Rather than drawing a lot of scribbles, the colorization way using examples involves transferring colors from similar images to the target images. Some of these methods measure similarity between the reference and target images at pixel level. Specifically, Charpiat *et al.* [3] chose some related color images as inputs, then built the distribution probability of reasonable colors for each pixel at local level, then finally optimized the probability of the colored image by graph-cut. With the same goal, Welsh *et al.* [20] transferred colors from the reference images to both the target images and videos. In contrast to the two methods described above, Liu *et al.* [4] prepared reference images by automatically searching the internet for adaptive photos. At the segment level, Irony *et al.* [21] computed the color on the segment of the example image that could be transferred to a target pixel. Moreover, the method in [22] added a manual caption to segmentation to filter appropriate reference samples. Although these methods produce nice and natural outcomes, they require a lot of time to find suitable references even when using automatic web retrieval.

C. COLORIZATION BASED ON DEEP LEARNING

Recently, with the evolution of computer vision applications, colorization has played an increasingly important role, and it is continuously improved in attempts to meet the needs of users. It has been applied to support advanced tasks, such as [23], which uses color attributes to enhance object detection performance, and [24], which builds a color correction application to produce more optimal results. Most recently, in the comic industry, colorization methods have been actively developed to substantially reduce costs and labor [9], [25]–[27].

To reduce the manual effort required of previous methods, the colorization process has leveraged machine deep learning to learn color prediction. For example, [6] suggested an image clustering technique for a large dataset and built a deep neural architecture to be trained with image feature descriptors. In recent times, semantics training for the machine has been exploited to achieve better performance. For instance, Iizuka *et al.* [7] constructed two branches based on deep convolutional neural networks to incorporate global and local features. To support semantics for the network, Zhang *et al.* [11] manipulated a cross-channel encoding scheme while Larsson *et al.* [8] designed a system that can predict a color histogram for each pixel and that is pre-trained for classification work. These methods achieved great results when coloring complex images.

In summary, there are three main approaches to colorize images. The first is colorization based on color scribbles of users on the image based on similar luminance to map similar

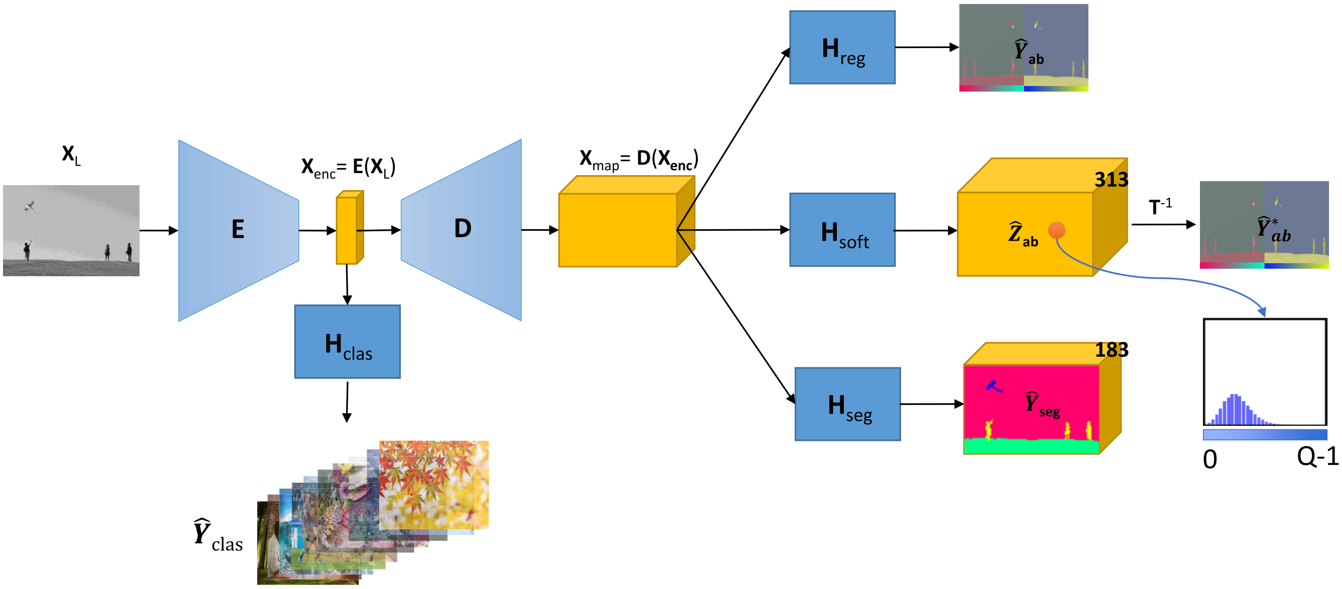


FIGURE 3. Semantic Image Colorization Model. It has the global scene-context classification $\mathbf{H}_{clas}$ to enhance the encoding feature $\mathbf{X}_{enc}$ from the encoding **E** in the uncertainty by transferring the learning of the scene-context probability vector $\hat{\mathbf{Y}}_{clas}$ from the pre-trained weight on the large-scale scene dataset. The 2D feature map $\mathbf{X}_{map}$ is enhanced by learning the mutual benefits at the pixel level in the color regression model $\mathbf{H}_{reg}$ for the mean color result $\hat{\mathbf{Y}}_{ab}$, the index color distribution model $\mathbf{H}_{soft}$ for the bias color result under pixel-wise probability vector $\hat{\mathbf{Z}}_{ab}$, and the semantic segmentation $\mathbf{H}_{seg}$ for the semantic matters with label mask prediction $\hat{\mathbf{Y}}_{seg}$.

colors [2], exploring edge information [17], and propagating colors at neighboring pixels or similar patterns [18] and [19]. The second is transferring colors from a reference image to a target image by measuring the similarity between reference and target images such as [3] using graph-cut optimization from the distribution probability of each local pixel. The last is an automatic colorization based on deep learning such as hyper-column VGG network with the un-rebalanced loss [8], two-stream architecture fusing global and local features with scene-context classification [7], and the cross-channel encoding scheme [11].

III. PROPOSED METHOD

In this section, we describe our proposed model, illustrated in Fig.3 to tackle the multi-model using color distribution for bias selection and color regression for the mean color result, as well as semantic matters using the global scene-context classification and pixel-level semantic segmentation. Then, we describe the semantic colorization in detail such as the index color distribution vector with the soft-encoding and decoding scheme, the uncertainty scene-context data, and the pixel-level semantic segmentation. Finally, we will mention the multi-loss function for our proposed model.

A. SEMANTIC IMAGE COLORIZATION MODEL

Let $p \triangleq (x, y)$ be the pixel location of the given grayscale image $\boldsymbol{I}_{gray} \in \mathbb{R}^{H\times W\times 1}$; then, our problem needs to find a plausible color version $\boldsymbol{I}_{RGB}(p) \in \mathbb{R}^3$ in the CIE RGB color space at pixel p. To simplify the color coherency problem involving the association between the best-unsaturated color $\boldsymbol{I}_{RGB}(p)$ and corresponding gray level $\boldsymbol{I}_{gray}(p)$, the CIE L*ab* color space [28] is chosen based on the similarity in the approximation about the color distance of the human perception and the distance of color coordinates in the color space. It comprises three components: the L channel presenting the lightness or grayscale axis and the a and b channels expressing the color axes with the orthogonal property. Therefore, our problem becomes that of building a mapping function $\mathbf{F}_{reg}$ based on the convolutional neural network to predict the *ab* values $\boldsymbol{Y}_{ab}(p) \in \mathbb{R}^2$ at pixel p with a given grayscale level $\boldsymbol{X}_L(p) \in \mathbb{R}$ expressed as below:

$$\hat{\boldsymbol{Y}}_{ab} = \mathbf{F}_{reg}(\boldsymbol{X}_L) \tag{1}$$

$$\hat{\boldsymbol{I}_{Lab}} = \left(\boldsymbol{X}_L, \hat{\boldsymbol{Y}}_{ab}\right) \tag{2}$$

Because of the multi-modal property of color distributions, learning models often fall into the average effect in the colorization process. The colorized results are grayish and desaturated. Larsson *et al.* [8] address this problem by replacing directly predicted a and b values in the per-pixel color distribution. Due to the correlation between a and b channels, Zhang *et al.* [11] use the 2D color distribution on the ab channel as the prediction output. By soft-encoding scheme $\mathbf{T}$, they represent $\boldsymbol{Y}_{ab}(p)$ under the 2D color distribution on the ab channels. Then, using grid-size 10, it is quantized to 1D vector representation $\mathbf{Z}_{ab}(p) \in \mathbb{R}^Q$ with Q = 313 in-gamut values:

$$\mathbf{Z}_{ab} = \mathbf{T}(\boldsymbol{Y}_{ab}) \tag{3}$$

Therefore, our image colorization model in the color distribution approach becomes:

$$\hat{\mathbf{Z}}_{ab} = \mathbf{F}_{soft}\left(X_L\right) \tag{4}$$

$$\hat{\mathbf{Y}}^{*}_{ab} = \mathbf{T}^{-1}\left(\hat{\mathbf{Z}}_{ab}\right) \tag{5}$$

where $\mathbf{T}^{-1}$ is the inverse function of the soft-encoding function $\mathbf{T}$.

From intuitive observations, the colorized image at local p is not only affected by spatial coherency but also by semantic matters in the local and global scopes. This means that the model must know the objects to which pixels belong, such as trees, sky, ocean, houses, etc. (**object localization**), as well as the place where the image has been captured such as indoors, outdoors, a shopping mall, a kitchen, etc. (**scene semantics**).

We need to incorporate the semantic segmentation $\mathbf{Y}_{seg} \in \mathbb{R}^{H\times W\times S}$ and the uncertainty scene semantic $\mathbf{Y}_{clas} \in \mathbb{R}^{C}$ from the grayscale image $\mathbf{X}_L$ for the image colorization problem, where S and C are the numbers of segmentation and scene classes, respectively. By adopting the encoder-decoder architecture, an image colorization method integrated into the semantic matters comprises the encoder $\mathbf{E}$, the decoder $\mathbf{D}$, the classification block $\mathbf{H}_{clas}$ and the convolution blocks for segmentation $\mathbf{H}_{seg}$, regression colorization mapping $\mathbf{H}_{reg}$, and color distribution mapping $\mathbf{H}_{soft}$. Our joint model with semantic matters can be expressed as follows:

$$\hat{\mathbf{Z}}_{ab} = \mathbf{H}_{soft}\left(\mathbf{D}\left(\mathbf{E}\left(\mathbf{X}_L\right)\right)\right) = \mathbf{F}_{soft}\left(\mathbf{X}_L\right) \tag{6}$$

$$\hat{\mathbf{Y}}_{ab} = \mathbf{H}_{reg}\left(\mathbf{D}\left(\mathbf{E}\left(\mathbf{X}_L\right)\right)\right) = \mathbf{F}_{reg}\left(\mathbf{X}_L\right) \tag{7}$$

$$\hat{\mathbf{Y}}_{seg} = \mathbf{H}_{seg}\left(\mathbf{D}\left(\mathbf{E}\left(\mathbf{X}_L\right)\right)\right) = \mathbf{F}_{seg}\left(\mathbf{X}_L\right) \tag{8}$$

$$\hat{\mathbf{Y}}_{clas} = \mathbf{H}_{clas}\left(\mathbf{E}\left(\mathbf{X}_L\right)\right) = \mathbf{F}_{clas}\left(\mathbf{X}_L\right) \tag{9}$$

where $\mathbf{X}_{enc} = \mathbf{E}\left(\mathbf{X}_L\right)$ is an encoding representation of the grayscale image and $\mathbf{X}_{map} = \mathbf{D}\left(\mathbf{E}\left(\mathbf{X}_L\right)\right)$ is the 2D feature map of every pixel used to decode the image into ab values, color distribution, and segmentation.

Our semantic image colorization model, illustrated in Fig. 3 uses the shared convolution neural networks at the encoder model $\mathbf{E}$ and the per-pixel feature map model $\mathbf{D}$. Therefore, a joint conditional distribution for colorization output is modelled as follows:

$$p\left(\hat{\mathbf{Z}}_{ab}|\mathbf{X}_L\right) = p\left(\hat{\mathbf{Z}}_{ab}|\mathbf{X}_L, \mathbf{X}_{enc}, \mathbf{X}_{map}\right) \tag{10}$$

As in the equation above, the color probability at pixel p is calculated based on the gray-scale input $\mathbf{X}_L$, the 2D map feature $\mathbf{X}_{map}$ for what and where the pixel belongs to, and the uncertainty scene-context feature $\mathbf{X}_{enc}$. In addition, the role of $\mathbf{F}_{reg}$ is the regularization factor for convergence to the ground-truth value, and the $\mathbf{F}_{soft}$ model plays the role of making a the suitable selection of the color distribution that satisfies the multi-modal property and the semantic matters.

B. SEMANTIC COLORIZATION DATA

1) COLOR DISTRIBUTION DATA WITH THE SOFT-ENCODING SCHEME

Soft-encoding Scheme: Because of the multi-modal property in the colorization problem, the regression color values $\hat{\mathbf{Y}}_{ab}$ could not exactly express the different color versions of the same things depending on the global and local scope in the scene context. Larsson *et al.* [8] and Zhang *et al.* [11] suggest using the color distribution on the a, b, or ab channels to deal with this problem. Similar to in Zhang *et al.* [11], we also quantize the 2D color space in the a and b channels by the grid-size 10 into the 1D vector quantization with Q = 313 in-gamut values as shown in Fig.4.

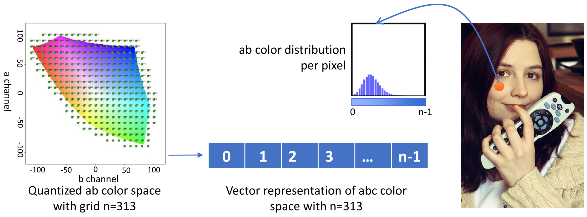


FIGURE 4. Soft-encoding Process.

For further explanation, the left image in Fig. 4 is the color space in the ab channel divided by a grid with a size of 10. In total, there are 313 points in the grid expressing the two coordinates a and b values when projecting onto the a and b axes, respectively. Every point in the grid will be numbered from 1 to Q=313. From there, we can express the coordinate in the 2D ab space based on the ab color index of every point in the grid. With the label smoothing technique, the probability values in $\mathbf{Z}_{ab}\left(p\right)$ depend on the distance between the *ab* coordinate with the nearest points on the grid. We only take the top 5 ab color indices nearest the ab coordinate of the pixel p, fill the probability under a Gaussian distribution, and normalize it to 1.

Weighted color index values: To encourage the rare colors in the image, we need to calculate the weighted term of every color index q in the training dataset. Zhang *et al.* [11] and Tram *et al.* [29] suggest to use the smoothness prior distribution $\mathbf{P}_{smooth}$ to calculate the balance terms as follows:

$$\begin{aligned} \mathbf{P}_{smooth} &= Interp\left(\mathbf{P}\right) * \mathbf{G}_{\sigma}, \\ \sum \mathbf{P}_{smooth} &= 1, \\ \mathbf{P}\left(q\right) &= \frac{\mathbf{N}_{train,q}}{\mathbf{N}_{train}} \end{aligned} \tag{11}$$

where $\mathbf{N}_{train,q}$ and $\mathbf{N}_{train}$ are respectively the number of occurrences of the color index q and the number of pixels in the training dataset. $\mathbf{P}_{smooth}$ is calculated by applying the interpolation operator *interp* (such as bicubic), smoothing with convolution operator $*$ using Gaussian kernel $\mathbf{G}_{\sigma}$, and normalizing the value to 1.

The weighted term $\mathbf{w}(q)$ at the color index q is derived from $\mathbf{P}_{smooth}$ such that:

$$\mathbf{w}(q) = \left((1-\lambda)\,\mathbf{P}_{smooth} + \frac{\lambda}{Q}\right)^{-1},$$
$$\mathbb{E}[\mathbf{w}] = \sum_{q=1}^{Q} \mathbf{P}_{smooth}(q)\,\mathbf{w}(q) = 1 \tag{12}$$

where $\lambda \in [0, 1]$ is the tuning term for mixing the smoothness probability $\mathbf{P}_{smooth}$ with the uniform distribution $\frac{\lambda}{Q}$, Q is the number of color index values, and the expectation of the weighted term $\mathbf{w}(q)$ is normalized to 1.

Decoding Scheme: **Mode Scheme** - $\mathbf{Z}_{ab}(p)$ at pixel location p in the image is a quantized color distribution vector for expressing the occurrence probability of the color index q. Each such vector has a total of 313 probabilities corresponding to the 313 bins of color index. Therefore, we can obtain the prediction color index q^* by the highest probability at pixel location p as follows:

$$q^* = \underset{q}{arg\,max}\ \mathbf{Z}_{ab}(p, q) \tag{13}$$

This method often achieves inconsistent results in obtaining the mode of color distribution. By contrast, the colorization result obtained using the mean value in the color distribution is a grayish, desaturated color.

Annealed-Mean Scheme - To avoid this problem, the annealed-mean distribution $\mathbf{Z}^*_{ab}(p)$ at pixel location p is used to obtain the color value in the color distribution:

$$\mathbf{Z}^*_{ab}(p, q) = \frac{exp\left(log((\mathbf{Z}^*_{ab}(p, q))/T\right)}{\sum_q exp\left(log((\mathbf{Z}^*_{ab}(p, q))/T\right)},$$
$$\sum_q \mathbf{Z}^*_{ab}(p, q) = 1 \tag{14}$$

where T is the temperature term for tuning the color distribution for a stronger peak when $T \to 0$ and for no change when $T = 1$.

Therefore, the ab color value $\mathbf{Y}^*_{ab}(p)$ at pixel location p is the expectation value of $\mathbf{Z}^*_{ab}(p)$:

$$\mathbf{Y}^*_{ab}(p) = \mathbb{E}_{ab}\left(\mathbf{Z}^*_{ab}(p)\right) \tag{15}$$

In the Annealed-Mean Scheme, by choosing $T \to 1$, the colorization result becomes grayish, and desaturated. When selecting $T \to 0$, the result is similar to that obtained using the mode scheme. In this paper, we choose $T = 0.38$, the same value used by as Zhang *et al.* [11]. This maintains the spatial constraint more than the mode scheme.

2) THE UNCERTAINTY SCENE-CONTEXT CLASSIFICATION

The main purpose of the scene-context classification is to transfer the style of the global prior knowledge of the image into the colorization image process. The global prior knowledge is transferred from a large-scale dataset containing the diversity scenecontext characteristics given by the scene type such as indoors (kitchen, bedroom), outdoors (farm, pasture), and objects as shown in Fig. 5.

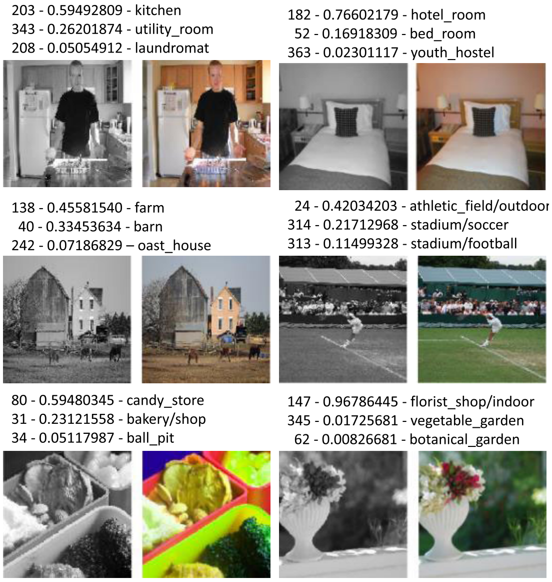
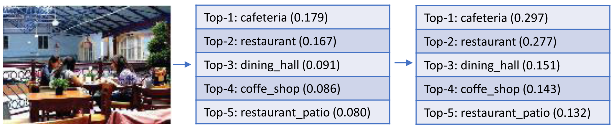

203 - 0.59492809 - kitchen
343 - 0.26201874 - utility_room
208 - 0.05054912 - laundromat
182 - 0.76602179 - hotel_room
52 - 0.16918309 - bed_room
363 - 0.02301117 - youth_hostel
138 - 0.45581540 - farm
40 - 0.33453634 - barn
242 - 0.07186829 – oast_house
24 - 0.42034203 - athletic_field/outdoor
314 - 0.21712968 - stadium/soccer
313 - 0.11499328 - stadium/football
80 - 0.59480345 - candy_store
31 - 0.23121558 - bakery/shop
34 - 0.05117987 - ball_pit
147 - 0.96786445 - florist_shop/indoor
345 - 0.01725681 - vegetable_garden
62 - 0.00826681 - botanical_garden

FIGURE 5. **The diversity scene-context characteristics in the COCO-Stuff dataset.**

In addition, the scene context also has the uncertainty property, as each a scene brings to mind more than unique meaning. For instance, for the image in Fig. 6, one person may think of it as a cafeteria, while person may imagine it like a restaurant. In this way, the detailed contextual meaning of an image will depend on the person to some extent. This is also caused by the error stemming from the transfer learning from the pre-train model, as is the case in Fig. 5 with the images about the object description shown in the middle column. Therefore, the uncertainty scene-context helps exploit the relationship among the main scene-contexts in the large-scale dataset dealing with the multi-modal property of image colorization under global coherent constraints. It also helps the model learn incorrect scene-context, which decreases the number of mistakes.



Top-1: cafeteria (0.179)	Top-1: cafeteria (0.297)
Top-2: restaurant (0.167)	Top-2: restaurant (0.277)
Top-3: dining_hall (0.091)	Top-3: dining_hall (0.151)
Top-4: coffe_shop (0.086)	Top-4: coffe_shop (0.143)
Top-5: restaurant_patio (0.080)	Top-5: restaurant_patio (0.132)

FIGURE 6. **The uncertainty scene-context classification.**

To train a machine to mimic the processes of the human mind, we apply a technology called label smoothing from Müller *et al.* [30] to improve the generalization and learning speed of the network. This prevents the network from being over-confident in the classification problem. According to the traditional cross-entropy classification, the one-hot vector consists of two values in which "1" presents the correct class and "0" indicates the incorrect class. However, with the label smoothing technique, the label y_{hot} of the one-hot encoded is replaced with a mixture of y_{hot} and the uniform distribution.

In this paper, instead of using the mixture of one-hot encoding and the uniform distribution as described above, we fill the top five highest probabilities from the pre-trained VGG16 model [31] on the Places [15] and normalize the values to 1 as shown on Fig. 6. We may transfer knowledge from the scene domain network to our colorization model under uncertainty, and therefore improve the model's generalizability.

3) THE PIXEL-LEVEL SEMANTIC SEGMENTATION

Aside from the global semantic features for style transferring in the uncertainty scene-context, the colorization process is affected by the semantic matters in the local scope at the pixel level. It means that the pixel belongs to the object and stuff types. In human vision, one uses their experiences with the objects in an image to make a colorized image suitable for the scene global coherent constraint.

In Fig. 7, with the scene-context classification, the uncertainty scene-context is calculated from the pre-trained weight on Places365 [15] for an image on Coco-Stuff [13]. It shows the three possible contexts for the image corresponding to with the top three highest probabilities. Using the label smoothing, we prevent the occurrence of errors when transferring learning, and set up the relationship between the scene types of soccer field, park, and golf course.

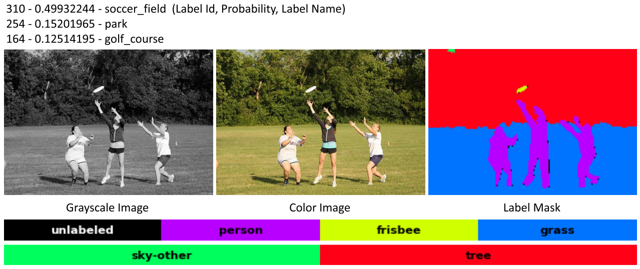


FIGURE 7. **Per-pixel Semantic-Segmentation and uncertainty classification on Coco-stuff dataset.**

Moreover, in the image details, a multi-model is shown with three colorization styles in the "person" label. For the stuff label, there are "tree", "grass" and "sky-other" labels affecting the scene-context. There is also small object label with a "frisbee" label. The content label mask helps the model to learn the local description and enhance the decoding feature map $\mathbf{X}_{map}$ to boost the accuracy in the complex scenes.

In practice, a color image could be segmented better than a gray-scale image [32]. Although our input system involves a gray-scale image, the semantic segmentation model takes advantage of the colorization model to improve the segmentation results. On the other hand, it helps the other models exploit the semantic pixels in terms of where they belong to and what the objects and stuff are.

C. MULTI-LOSS FUNCTION

Our joint colorization model has the goal of image colorization in the multi-modal property integrating the semantic matters. For this reason, our joint loss $\mathbb{L}_{joint}$ in Eq. 16 is the weighted sum of the category cross-entropy loss for the color distribution $\mathbb{L}_{soft}$, the mean-square error loss for color regression $\mathbb{L}_{reg}$, the soft-dice loss for semantic segmentation $\mathbb{L}_{seg}$ and the category cross-entropy loss for the uncertainty scene classification $\mathbb{L}_{clas}$ with the label smoothing technique [30].

$$\mathbb{L}_{joint} = \alpha_1\mathbb{L}_{soft} + \alpha_2\mathbb{L}_{reg} + \alpha_3\mathbb{L}_{clas} + \alpha_4\mathbb{L}_{seg} \tag{16}$$

where α_i is the weighted term used to adjust the importance among colorization regression, distribution, semantic segmentation, and scene classification losses.

To minimize the errors of the per-pixel 2D color distributions caused by the prediction $\hat{\mathbf{Z}}_{ab}$ and ground-truth $\mathbf{Z}_{ab}$ on *ab* channels, we use the weighted category cross-entropy loss for $\mathbb{L}_{soft}$ as in [11]:

$$\begin{aligned}\mathbb{L}_{soft}\left(\mathbf{Z}_{ab}, \hat{\mathbf{Z}}_{ab}\right) &= -\sum_{p} w\left(q^*\right)\sum_{q=1}^{Q}\mathbf{Z}_{ab}\left(p, q\right)\log\hat{\mathbf{Z}}_{ab}\left(p, q\right),\\ \text{where } q^* &= \underset{q}{arg\,max}\ \mathbf{Z}_{ab}\left(p, q\right)\end{aligned} \tag{17}$$

where, $\mathbf{Z}_{ab}(p)/\hat{\mathbf{Z}}_{ab}(p)$ is the ground-truth/prediction of the color distribution vector at pixel location p, $\mathbf{Z}_{ab}(p, q)/\hat{\mathbf{Z}}_{ab}(p, q)$ is the color probability at bin q of $\mathbf{Z}_{ab}(p)/\hat{\mathbf{Z}}_{ab}(p)$, q^* is the bin in the color distribution vector $\mathbf{Z}_{ab}$ with the highest probability, and $w(q^*)$ is the weighted term to encourage the rare color.

With the regularization role in the regression model $\mathbf{F}_{reg}$, $\mathbb{L}_{reg}$ is the mean-square-error loss used to measure the convergence between the predicted and ground-truth colors:

$$\mathbb{L}_{reg}\left(\mathbf{Y}_{ab}, \hat{\mathbf{Y}}_{ab}\right) = \frac{1}{H \times W}\sum_{p}\left\|\mathbf{Y}_{ab}(p) - \hat{\mathbf{Y}}_{ab}(p)\right\|_2^2 \tag{18}$$

where H and W are the height, and width of the input image, respectively.

Here, the model $\mathbf{F}_{reg}$ has the role of approximating the common colorization result to be suitable in terms of semantic matters by using the scene-context model and the per-pixel semantic segmentation model, while the model $\mathbf{F}_{soft}$ addressed the multi-model properties in the colorization process. It can produce a vibrant colorized image that can fool human vision. However, it can also make the mistake of over-aggressive colorization, requiring a fix by an annealed-mean from the color distribution based on the weighted average over the color probability bins.

For the per-pixel semantic segmentation model $\mathbf{F}_{seg}$, we use softmax dice loss [33] to calculate each class mask and the average yield as final score $\mathbb{L}_{seg}$:

$$\mathbb{L}_{seg}\left(\mathbf{Y}_{seg}, \hat{\mathbf{Y}}_{seg}\right) = 1 - \frac{2\sum_p \mathbf{Y}_{seg}(p)\,\hat{\mathbf{Y}}_{seg}(p)}{\sum_p \mathbf{Y}_{seg}(p)^2 + \sum_p \hat{\mathbf{Y}}_{seg}(p)^2} \tag{19}$$

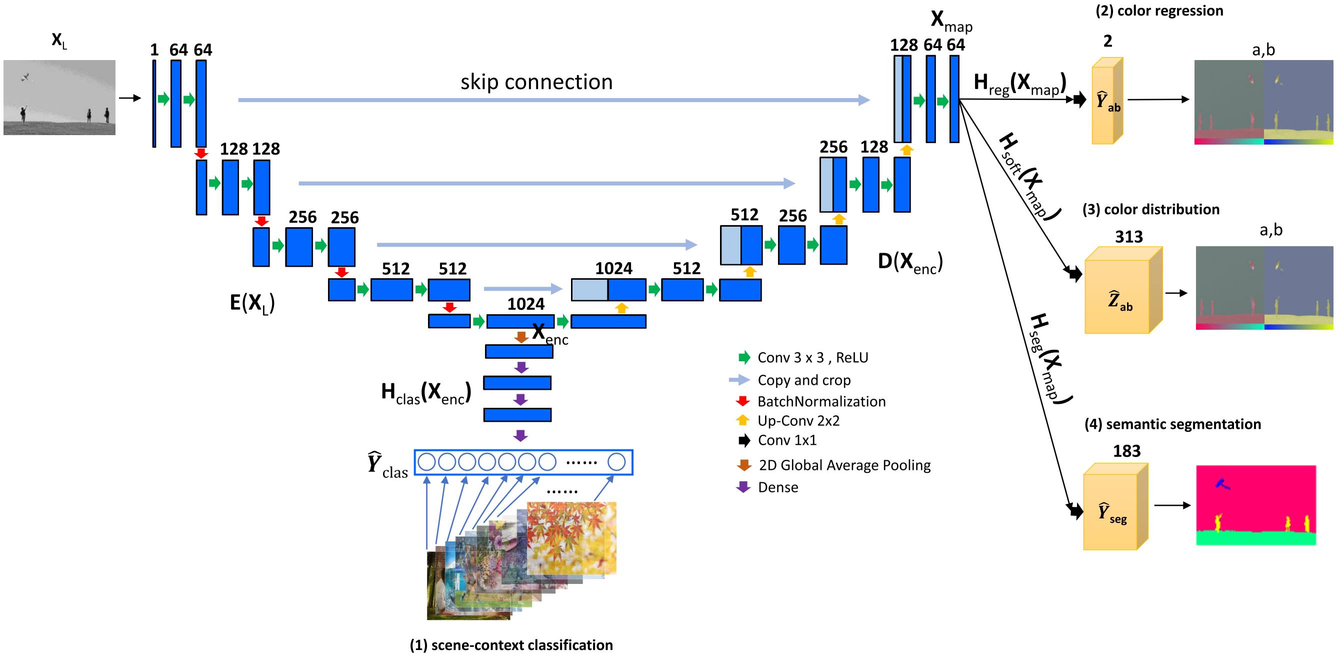


FIGURE 8. Implementation Details of Semantic Image Colorization Model using U-Net.

It is difficult for the $\mathbf{F}_{seg}$ model to predict segmentation under a gray-scale image instead of the RGB domain, as is common. However, using per-pixel fusing among regression colorization and distribution helps the segmentation model learn the color information better, which strengthens the accuracy in segmentation. On the other hand, the segmentation model also supplies the per-pixel semantic for the colorization process.

Finally, for the uncertainty scene-context classification model $\mathbf{F}_{clas}$, the category cross-entropy loss is used to calculate $\mathbb{L}_{clas}$:

$$\mathbb{L}_{clas}\left(\mathbf{Y}_{clas}, \hat{\mathbf{Y}}_{clas}\right) = -\sum_{i=1}^{C} \mathbf{Y}_{clas,i} log \hat{\mathbf{Y}}_{clas,i} \tag{20}$$

where C is the number of scene types and $\mathbf{Y}_{clas,i}/\hat{\mathbf{Y}}_{clas,i}$ is the ground-truth/prediction of scene probability on class i.

The model $\mathbf{F}_{clas}$ transfers the global meaning in the scene (indoor/outdoor, kitchen room, etc.) into the colorization process through the low-level encoding feature $\mathbf{X}_{enc}$. Based on the pre-trained weight on Places365 [15], the ground-truth $\mathbf{Y}_{clas}$ is used to calculate the scene-context probability under the uncertainty condition with the label smoothing technique [30]. This means that instead of giving the correct class with probability 1, we assign the top 5 highest probabilities by Gaussian distribution and normalize the values to 1. The benefit of label smoothing is not only in enhancing the accuracy of the correct label, but also in learning the relationship with nearest scene-contexts under the uncertainty in labeling the scene type.

In summary, by multi-loss $\mathbb{L}_{joint}$, the overall model will learn the mutual benefits among the specific models $\mathbf{F}_{soft}$, $\mathbf{F}_{reg}$, $\mathbf{F}_{seg}$, and $\mathbf{F}_{clas}$. The colorization process will address the semantic matters in a per-pixel manner by $\mathbf{F}_{seg}$ as well as a global scene-context by $\mathbf{F}_{clas}$. Moreover, the colorization process produces the common frequency pattern by $\mathbf{F}_{reg}$ integrated in the multi-modal colorization distribution by $\mathbf{F}_{soft}$.

IV. IMPLEMENTATION DETAILS

In this section, we describe our network architecture in further details. Our implementation for automatic image colorization based on learning from the per-pixel scene-semantic and global scene-context is described in Fig. 8. The input of our model is the grayscale image X_L with size $H \times W \times 1$. The outputs of model are scene-context classification $\hat{Y}_{clas}$ with size $H \times W \times C$ where $C = 365$ the number of scene classification classes, the color regression $\hat{Y}_{ab}$ on ab channel with size $H \times W \times 2$, the color distribution $\hat{Z}_{ab}$ with size $H \times W \times Q$ where $Q = 313$ the number of bins in the soft-encoding scheme, and the semantic segmentation mask $\hat{Y}_{seg}$ with size $H \times W \times S$ where $S = 183$ the number of segmentation classes.

The colorization problem could be considered as a semi-supervised learning process that involves learning the colorization features by making the gray-scale version of a color image as the input data and then learning the color version of that same image. In the testing phase, we use the learning model to predict the gray-scale image without the color version.

Therefore, in the first step, we construct the colorization data by transforming the color image $\mathbf{I}_{RGB}$ into $\mathbf{I}_{Lab}$ in the CIE Lab color space. This space has three coordinates: L represents the gray-scale axis, the a axis represents the green–red component, and the b axis indicates the blue–yellow component. We choose this color space because its metric is convenient for measuring the correlation between the L lightness channel and the ab color channel. With this context, the input image $\mathbf{X}_L \in \mathbb{R}^{H \times W \times 1}$ is the lightness channel $\mathbf{I}_L$. The colorization task only needs learn the regression color values $\mathbf{Y}_{ab}$ in the a and b channels $\mathbf{I}_{ab}$. Based on the global scene-context style and the pixel-level semantic matters, we suggest the index color distribution vector $\mathbf{Z}_{ab}$ for multi-model solving, the uncertainty scene-context vector $\mathbf{Y}_{clas}$ by transferring learning for the global style, and semantic segmentation $\mathbf{Y}_{seg}$ for the pixel-level semantic matters. The details are described in Section III-B.

We choose the U-net structure [12] as the platform for building our network, as it has many advantages. For one, this network is suitable for a completely automated colorization task because it can be trained end-to-end. Further, we also leverage the architecture including the contracting path named $\mathbf{E}(\mathbf{X}_L)$ that can capture context to encode the lightness channel to the low-level encoding features $\mathbf{X}_{enc}$, and the expanding path named $\mathbf{D}(\mathbf{X}_{enc})$, of which precise localization decodes the encoding features into the 2D feature map $\mathbf{X}_{map}$.

For details, $\mathbf{E}_{enc}(\mathbf{X}_L)$ uses the back-bone VGG16 [31], comprised of four convolution blocks with two convolution layers using kernel 3×3 and one batch normalization layer as well as one convolution block in the middle for the encoding feature. After every encoding block, the resolution of the image is reduced to half its value by the striding 2 in the last convolution in the block and the number of filters double. $\mathbf{D}(\mathbf{X}_{enc})$ consists of four decoding blocks containing one up-sampling layer, and three convolution layers. After every decoding, the up-sampling layer doubles the resolution and the number of filters decreases by half. There are also **skip connections** between the last convolution in the encoding block and the first convolution layer in the decoding block at the same depth level. It helps our model avoid the broken connection when the features from the below-decoding block are transferred feature to the upper-decoding block by vanishing or the dying-ReLU problem.

For the uncertainty scene-context classification $\mathbf{H}_{clas}(\mathbf{X}_{enc})$, it puts it in the middle of the U-Net model to enhance the $\mathbf{X}_{enc}$. The classification branch includes the 2D global average pooling layer, two dense layers, and the soft-max layer outputting the one-hot vector containing the probabilities of scene-context occurrence.

The color distribution branch $\mathbf{H}_{soft}(\mathbf{X}_{dec})$, the color regression branch $\mathbf{H}_{reg}(\mathbf{X}_{dec})$, and the semantic segmentation branch $\mathbf{H}_{seg}(\mathbf{X}_{dec})$ are placed after the decoding model $\mathbf{D}(\mathbf{X}_{enc})$, and they have the input set to be the 2D feature map $\mathbf{X}_{dec}$. By using the convolution layer 1×1 with the same number of filters as the output values (two for color regression, 313 for color distribution, and 183 for semantic segmentation), they produce the output at the pixel level with same size as the gray-scale input. They are only different from the activation layer with the *tanh* function for color regression and the *softmax* function for color distribution and segmentation.

V. EXPERIMENTS AND DISCUSSION

A. DATASETS

Our experiments were performed on the COCO-Stuff dataset [13] in Fig.9. The COCO-Stuff is a subset of the COCO dataset for large-scale object detection, segmentation, and captions with more than 118,000 images for training and 5,000 images for the validation set. It consists of 172 classes involving 80 things, 91 stuff, and one class unlabeled. We also convert each image to the size of 224×224 before the training process.

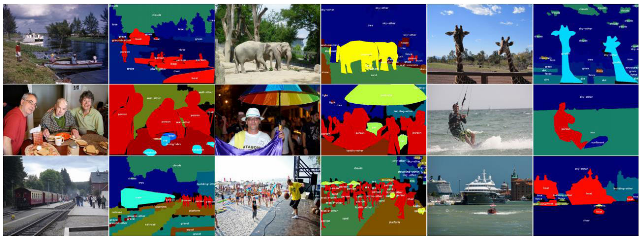


FIGURE 9. Semantic-Segmentation in COCO-Stuff Dataset [13].

For scene ground-truth, we used the pre-trained weight on the VGG16 model of the Places365-Standard dataset [15]. We predicted the scene probabilities of 5.000 images from theCOCO-Stuff validation set, obtained the top five probabilities and normalized the value to 1. Fig. 10 shows some images and classes of Places365-Standard with 1.8 million images having a train set, as well as 365 different classes of scene/location. Each class has from 3,068 to 5,000.

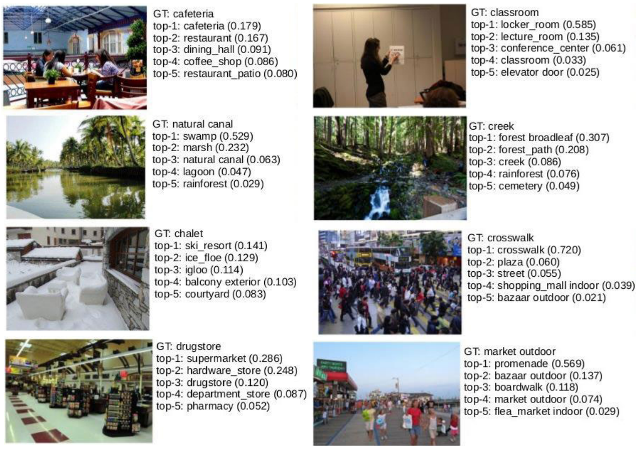

GT: cafeteria
top-1: cafeteria (0.179)
top-2: restaurant (0.167)
top-3: dining_hall (0.091)
top-4: coffee_shop (0.086)
top-5: restaurant_patio (0.080)
GT: classroom
top-1: locker_room (0.585)
top-2: lecture_room (0.135)
top-3: conference_center (0.061)
top-4: classroom (0.033)
top-5: elevator door (0.025)
GT: natural canal
top-1: swamp (0.529)
top-2: marsh (0.232)
top-3: natural canal (0.063)
top-4: lagoon (0.047)
top-5: rainforest (0.029)
GT: creek
top-1: forest broadleaf (0.307)
top-2: forest_path (0.208)
top-3: creek (0.086)
top-4: rainforest (0.076)
top-5: cemetery (0.049)
GT: chalet
top-1: ski_resort (0.141)
top-2: ice_floe (0.129)
top-3: igloo (0.114)
top-4: balcony exterior (0.103)
top-5: courtyard (0.083)
GT: crosswalk
top-1: crosswalk (0.720)
top-2: plaza (0.060)
top-3: street (0.055)
top-4: shopping_mall indoor (0.039)
top-5: bazaar outdoor (0.021)
GT: drugstore
top-1: supermarket (0.286)
top-2: hardware_store (0.248)
top-3: drugstore (0.120)
top-4: department_store (0.087)
top-5: pharmacy (0.052)
GT: market outdoor
top-1: promenade (0.569)
top-2: bazaar outdoor (0.137)
top-3: boardwalk (0.118)
top-4: market outdoor (0.074)
top-5: flea_market indoor (0.029)

FIGURE 10. Scene-Context Classification in Places365 dataset [15].

For colorization testing, we built the first 1000 images from the validation sets of four datasets: the ImageNet [16], Places365 [15], and Coco-Stuff [13], as well as 100 images of the high-resolution validation set in the DIV2K dataset [14] shown in Fig.11. In addition, we obtained some images from the Internet for validating the colorization results. The images

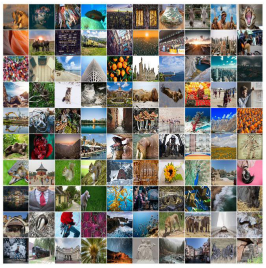


FIGURE 11. DIV2K Dataset [14].

that were published on the Internet were a wide range of life and legacy images.

Table 1 lists the numbers of images used for training, validation and testing in this study.

TABLE 1. Number of images in training, validation and testing datasets.

	Training	Validation	Testing
COCO-Stuff [13]	118000	5000	1000 (ctest1k)
Place365 [15]			1000 (ctest1k)
DIV2K [14]			100 (hight-resolution)
ImageNet [16]			1000 (ctest1k)

B. ENVIRONMENTAL SETUP, TRAINING DETAILS

Environment: Our development environment was Python 3.7 using Tensorflow Keras 2. We used a desktop PC with Intel Corei5 8400 with 32 GB of RAM and a GeForce GTX 1080 8GB RAM graphics card.

Training Scheme: In this study, we did not use cross-validation due to training on the large dataset Coco-Stuff and testing on the different test sets from DIV2K, Place365, ImageNet, Coco-Stuff dataset.

Training Data: We trained the model on 118,000 images of the COCO-stuff training set, and validated the model on 5,000 images. Each input was resized to at most 256 pixels. Some of the techniques used for data augmentation are random contrast, brightness, random horizontal flip, rotate, scale and translate.

Training Details: To overcome over fitting when learning multi-task, the model training consisted of two stages for: in the first stage, training began with an initial learning rate of 0.0004 using a reduce learning rate on plateau at every five epochs with 0.95 decay, then the model was fine-turned again by using a Cycle Learning Rate with an initial learning rate of 0.0008 and next values of the learning rate cyclically varying within a range of [0.0008, 0.00001] in the period of eight epochs for optimizing colorization, semantic scene, and semantic segmentation. Every configuration required three days off training time. Some optimization functions were applied for learning such as Adam, SGD and RMSProp; Adam and RMSProp provided better results than SGD.

C. EVALUATION METRICS AND COMPARISON METHODS

Evaluation Metrics: Following previous papers, we used two kinds of evaluation metrics: quantitative and quality metrics.

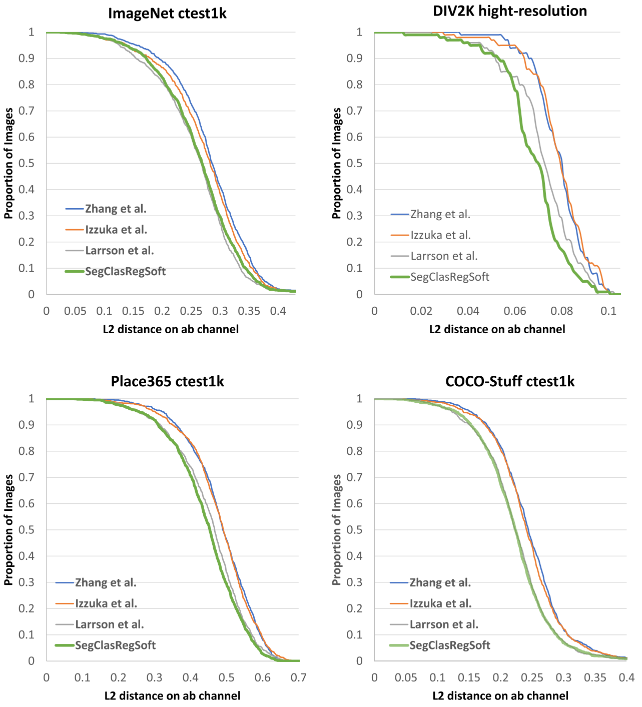


FIGURE 12. Cumulative histograms of $L2_{ab}$ distance (lower = fewer pixels with errors). The results for our methods are better on ImageNet, DIV2K, Places365, and COCO-Stuff.

For the quantitative metrics, we divided them into two small groups: similarity metrics and the perceptual approach. There are two similarity metrics used in this study: Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity Index (SSIM) [34], and the L_2 distance of ab channel, as shown below:

$$PSNR\left(\mathbf{I},\hat{\mathbf{I}}\right) = 10\log_{10}\frac{255^2}{MSE\left(\mathbf{I},\hat{\mathbf{I}}\right)} \tag{21}$$

$$MSE\left(\mathbf{I},\hat{\mathbf{I}}\right) = \frac{1}{P}\sum_{p}\left(\mathbf{I}(p)-\hat{\mathbf{I}}(p)\right)^2 \tag{22}$$

$$L_2\left(\mathbf{I},\hat{\mathbf{I}}\right) = \frac{\sqrt{\sum_p\left(\mathbf{I}(p)-\hat{\mathbf{I}}(p)\right)^2}}{2P} \tag{23}$$

$$SSIM(\mathbf{I},\hat{\mathbf{I}}) = \frac{(2\mu_{\mathbf{I}}\mu_{\hat{\mathbf{I}}}+C_1)+(2\sigma_{\mathbf{I}\hat{\mathbf{I}}}+C_2)}{(\mu_{\mathbf{I}}^2+\mu_{\hat{\mathbf{I}}}^2+C_1)(\sigma_{\mathbf{I}}^2+\sigma_{\hat{\mathbf{I}}}^2+C_2)} \tag{24}$$

where $\mathbf{I}$ and $\hat{\mathbf{I}}$ are the ground-truth and prediction images, respectively; P is the number of image pixels; C_1 and C_2 are the numerical stabilizing constants; and μ and σ are the mean and standard variance of an image, respectively. For a color image, we computed these metrics on every channel and obtained the average result. PSNR and SSIM evaluations were performed on RGB color images. L_2 was evaluated based on the ab color channel.

These similarity metrics quantify the reconstruction quality and structural similarity of ground truth and images filled

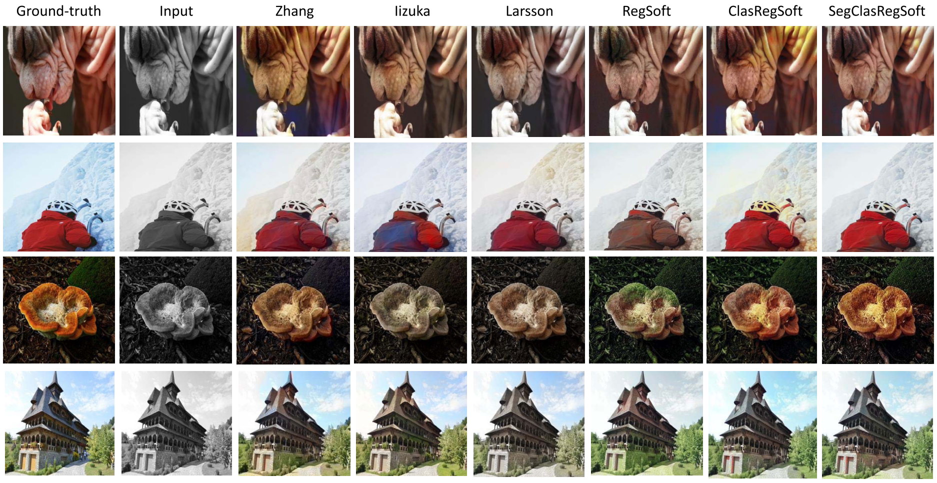


FIGURE 13. Successful cases of the DIV2K high-resolution validation. Our results were more vibrant and had more precise edges than the other methods. Moreover, the yellow color noise also was reduced in our ClasRegSoft versions comparison on RegSoft version.

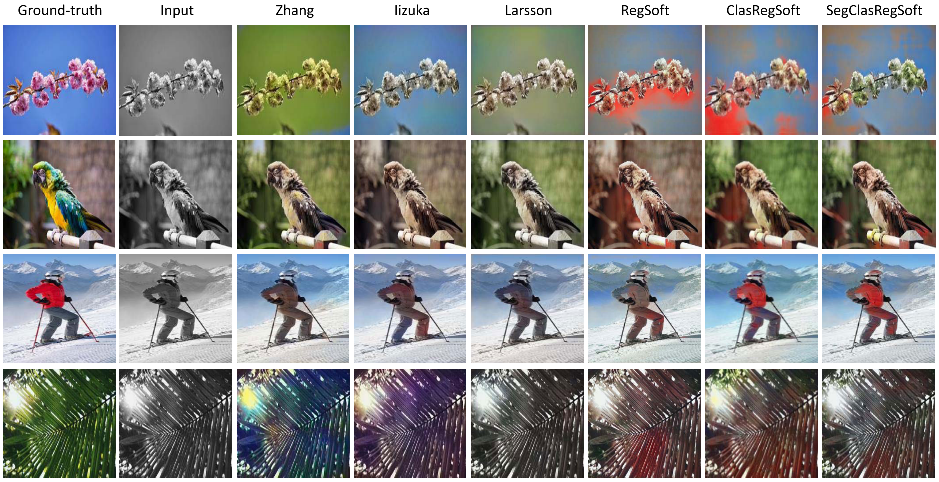


FIGURE 14. Fail cases of the DIV2K high-resolution validation. The fail cases often fell into images with rare colors, so the predicted colors were either pale or wrong. In addition, red noises also occurred.

with predicted colors. However, according to our observations, they do not sufficiently reflect the visual quality.

Therefore, we added another form of evaluation involving the perceptual approach in the quantitative metrics as follows:

First, we used the semantic interpretability [7], [11] by classification accuracies in the ImageNet and Places365 datasets for the colorized images. We used Top1-Acc with VGG16 pretrained weight on ImageNet and Places365.

Secondly, we also used Fréchet Inception Distance [35] to measure the semantic distance between the colorized output and the realistic natural images.

Finally, our method was also evaluated by the perceptual metric of **L**earned **P**erceptual **I**mage **P**atch **S**imilarity (LPIPS) described by Zhang *et al.* [36] (with AlexNet backbone, version 0.1). This metric assesses how well metrics correspond with human perceptual judgments under traditional distortions: noise, photometric, blur, warping and compression by computing the cosine distance from the normalized vector features.

For the quality metrics, we used the visual performance to show the success cases as well as the failure cases for comparison. We evaluated the visual performance on four public datasets DIV2K, ImageNet, COCO-Stuff, and Places365. We also showed the colorization results on legacy and life images collected from the Internet.

In terms of comparison methods, our method was compared to three robust colorization methods Iizuka *et al.* [7], Larsson *et al.* [8], and Zhang *et al.* [11], as presented in Table 2.

We also took the pre-trained weights issued by that these authors predict colors for images and compared them to our predictions, as listed in Table 2. Despite differences in the

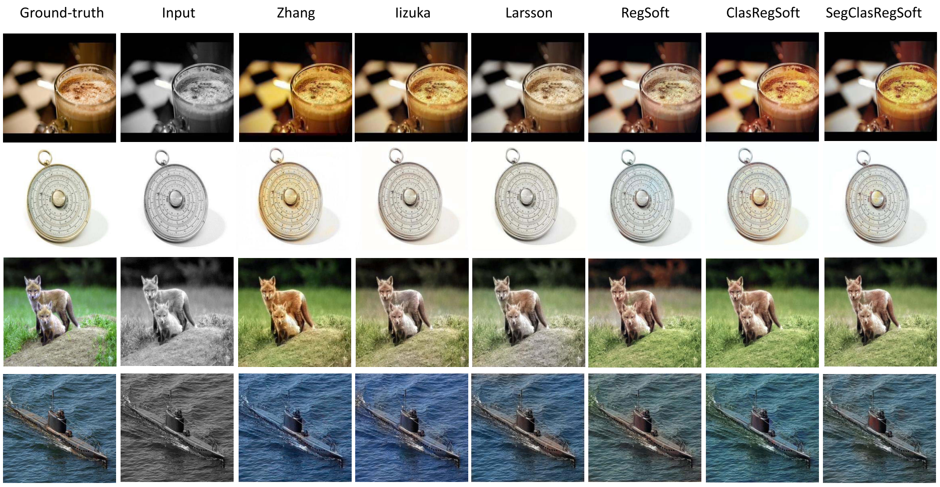


FIGURE 15. Successful cases of the ImageNet ctest1k. Our results were closer to the original photos than the others.

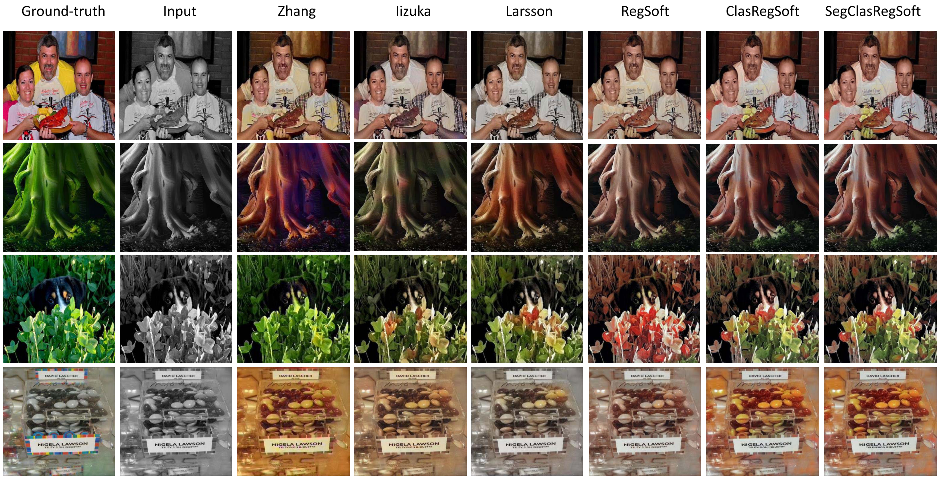


FIGURE 16. Fail cases of the ImageNet ctest1k. For the images with more details, our results met difficulties for colorization with incorrect colors and noise occurrences. These defects are similar to the results of Iizuka *et al.* [7] and Larsson *et al.* [8].

TABLE 2. Comparison Methods.

Method	Name	Training Data
1	Iizuka et al. [7]	Place365
2	Larsson et al. [8]	ImageNet
3	Zhang et al. [11]	ImageNet
4	Ours with RegSoft	COCO-Stuff
5	Ours with ClassRegSoft	COCO-Stuff
6	Ours with SegClassRegSoft	COCO-Stuff

training dataset, all of the images are collected from the Internet and show a variety of activities, things, and scenes in nature. In this study, we also conducted an ablation study to evaluate several model designs in our proposed method, as listed in Table 2.

We explored the effect of every branch on the colorization result. For the *RegSoft* design, we only kept the regression and the color distribution branch with branches (2) + (3), as shown in Fig. 8. For *ClassRegSoft*, we added one more classification branch to our model with branches (1) + (2) + (3). Finally, the *SegClassRegSoft* design had all branches (1) + (2) + (3) + (4), including the regression, color distribution, classification, and segmentation branches, respectively.

We have four kinds of outputs in our model. The semantic segmentation provided the label mask for determining the semantic of pixels in the image, while the classification scene-context determined the global context of an image. These branches helped our model learn the semantic colorization.

For the colorization results, our model returned from branch 2 with colorization regression. The output of this branch provided an average result with desaturation and a grayish effect. However, with using the classification and segmentation, this branch gave a better result, as shown in the success cases of ImageNet at Fig.15 with at least noise and

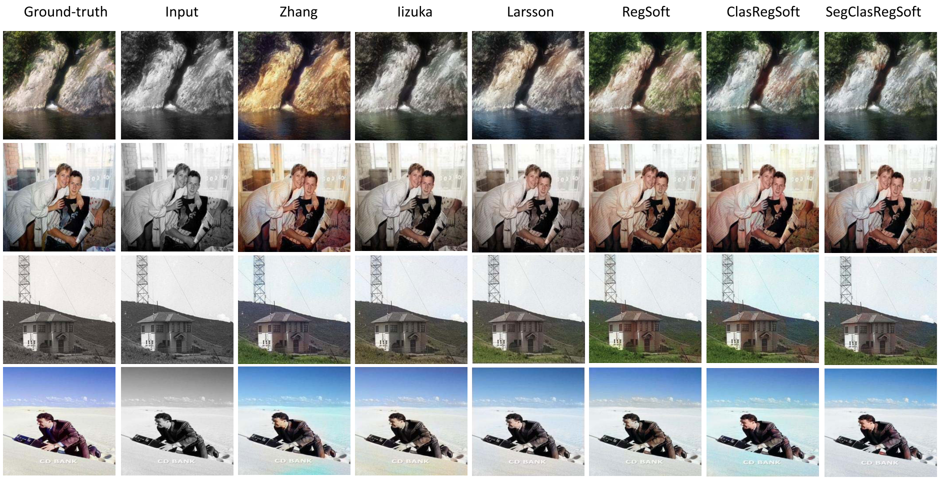


FIGURE 17. Successful cases of the Places365 ctest1k. The colors in our results were good enough to produce some vivid photos.

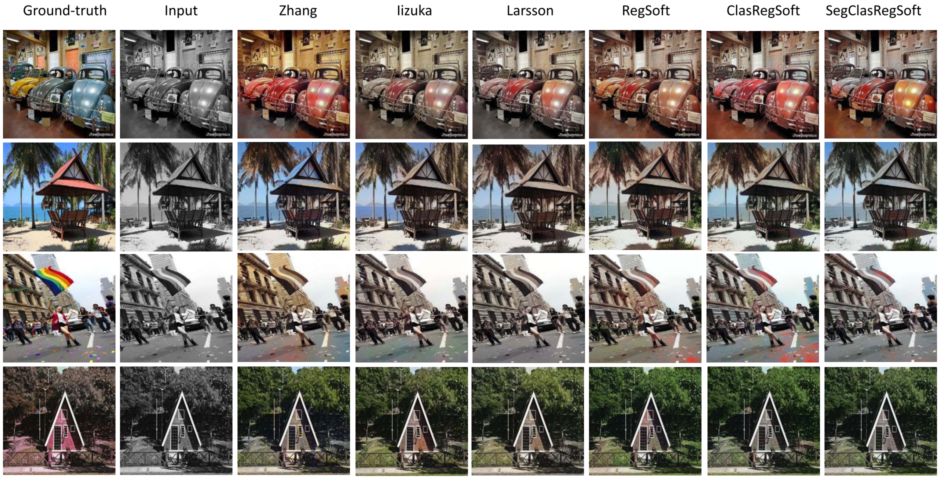


FIGURE 18. Fail cases of the Places365 ctest1k. Our results had some red noise in RegSoft, and ClasRegSoft version. However, the results were enhanced in SegClasRegSoft version.

more colors in *ClassRegSoft* and *SegClassRegSoft* comparing to *RegSoft* design.

Finally, the soft-encoding result in branch 3 returned the color distribution of the ab channels. The colorization result was calculated based on the weighted average between the probability values and the corresponding intensity in every a and b channel by Eq. 15. Using the annealing-mean scheme, we adjusted the probability values in the role of the weighted term by the exponent function controlled by temperature term T by Eq. 14. The benefits of this scheme are that it keeps the spatial constraints and removes the noise by the weighted average effects. When T reaches 1, the color probabilities almost stay the same, leading to average effects. When T reaches 0, the higher probability values are increased substantially more than the lower values by the exponent function effect. The colorization result gave a more vibrant result with few artifacts.

D. EXPERIMENT 1: QUANTITATIVE COMPARISONS ON SIMILARITY METRICS

Tables 3 and 4 list our results of the quantitative comparisons on Similarity Metrics. For PSNR and SSIM, higher values are the better. For $L2_{ab}$ lower values are better. The method of Larsson *et al.* gave almost better PSNR results on ImageNet, DIV2K, and COCO-Stuff (23.335, 23.49, and 23.773 respectively) and SSIM results on ImageNet and DIV2K (0.869, and 0.929, respectively). However, our methods provided the best results on the $L2_{ab}$ metric for DIV2K, Places365, and COCO-Stuff with values of 0.068, 0.442, and 0.223, respectively. The result shows that semantic segmentation played an important role in enhancing the colorization results, and it helped our method improve the accuracy of the ab channels.

Fig. 12 shows comparisons of the error distributions of $L2_{ab}$ on comparison datasets. Our fully automatic method

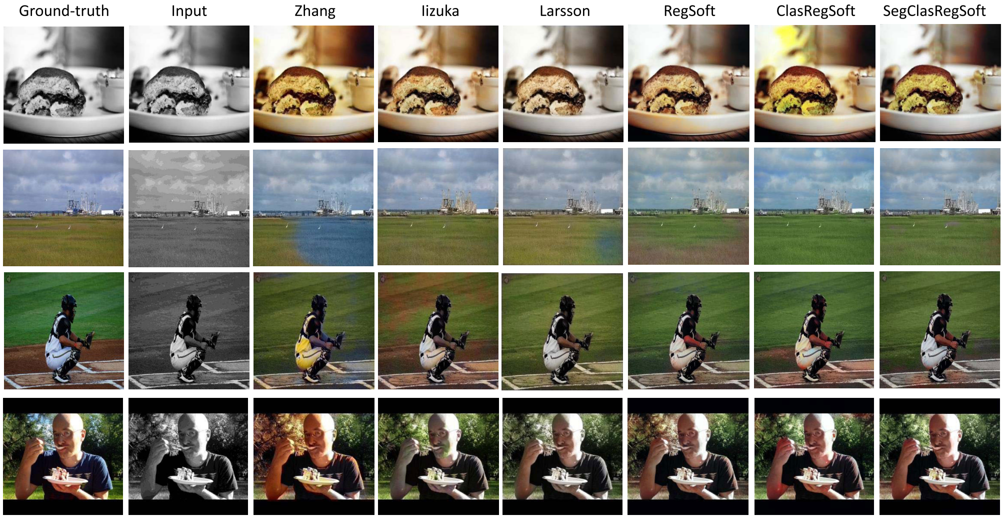


FIGURE 19. Successful cases of the COCO-Stuff ctest1k. Our SegClasRegSoft could distinguish food and others, so instances of color bleeding slightly decreased.

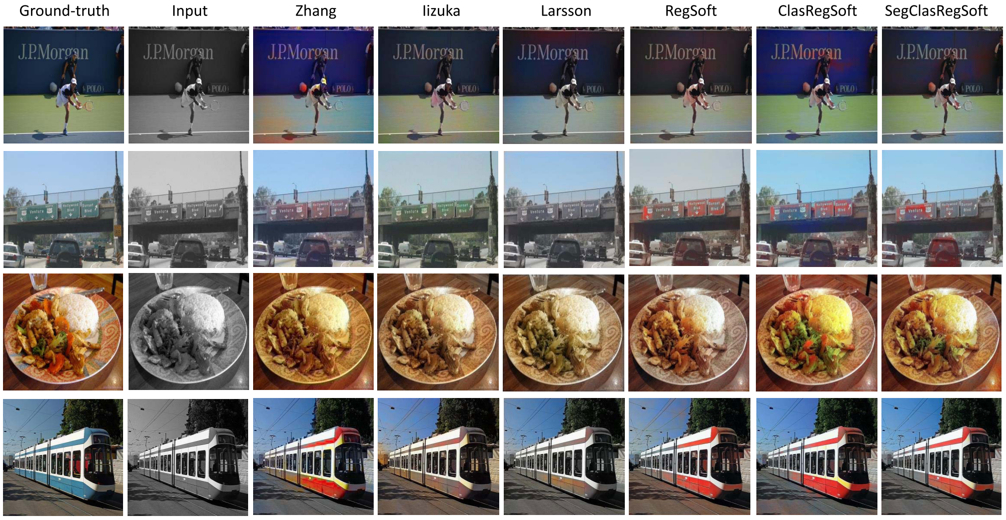


FIGURE 20. Fail cases of the COCO-Stuff ctest1k. In a few images with low quality and many small objects, our methods were not successful.

dominates all competing approaches by version *SegClasRegSoft*, except for the ImageNet dataset.

E. EXPERIMENT 2: QUALITY COMPARISONS

1) DIV2K HIGH-RESOLUTION VALIDATION

We showed some successful and fail cases on DIV2K high-resolution validation in Figs.13 and 14, respectively. The results obtained with our version *SegClassRegSoft* were improved more than our previous methods. The colors in our pictures were more vibrant and had more precise edges than the other methods. We also reduced the yellow color noise that occurred in our *RegSoft* and *ClasRegSoft* versions.

2) ImageNet ctest1k

We first used 1,000 images from the ImageNet validation set to make the ImageNet ctest1k. Although Zhang *et al.* [11] and Larsson *et al.* [8] performed training on ImageNet, we obtained some successful cases that were closer to the original photos than their results.

The pictures in ground-truth shown in Fig. 16 with many more details also cause many difficulties for automatic colorization through our methods or others. In Fig. 16, we observed incorrect colors and noise again in the fail cases. These defects are similar to the results of Iizuka *et al.* [7] and Larsson *et al.* [8] on the ImageNet validation set.

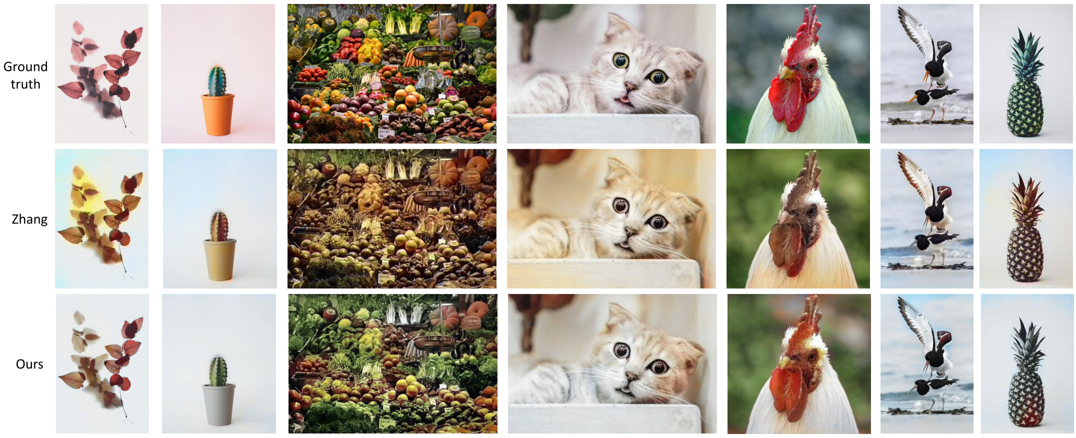


FIGURE 21. Colorization Images on Internet. In the chicken image at column 5, our result showed the more detail at the edge comparison on Zhang's method. By pixel-wise semantic segmentation, we achieved the better result in enhancing the semantic objects in the image.

TABLE 3. Similarity results on ImageNet ctest1k and DIV2K high-resolution validation.

Method	ImageNet ctest1k			DIVK2K		
	PSNR ↑	SSIM ↑	$L2_{ab}$ ↓	PSNR ↑	SSIM ↑	$L2_{ab}$ ↓
Iizuka et al. [7]	22.841	0.865	0.277	22.981	0.919	0.079
Larsson et al. [8]	**23.335**	**0.869**	**0.26**	**23.490**	**0.929**	0.072
Zhang et al. [11]	21.297	0.848	0.286	20.929	0.896	0.079
Ours with RegSoft	22.102	0.896	0.269	22.026	0.914	0.071
Ours with ClassRegSoft	21.068	0.886	0.274	21.694	0.912	0.071
Ours with SegClassRegSoft	21.900	0.893	0.264	22.330	0.917	**0.068**

TABLE 4. Similarity results on Place365 and COCO-Stuff ctest1k.

Method	Place365 ctest1k			COCO-Stuff ctest1k		
	PSNR ↑	SSIM ↑	$L2_{ab}$ ↓	PSNR ↑	SSIM ↑	$L2_{ab}$ ↓
Iizuka et al. [7]	**25.572**	**0.948**	0.481	23.541	0.871	0.242
Larsson et al. [8]	25.096	0.945	0.452	**23.773**	0.873	**0.223**
Zhang et al. [11]	23.076	0.928	0.484	21.502	0.851	0.245
Ours with RegSoft	23.599	0.932	0.474	22.872	0.912	0.23
Ours with ClassRegSoft	22.916	0.924	0.466	22.134	0.907	0.23
Ours with SegClassRegSoft	23.858	0.931	**0.442**	22.985	**0.913**	**0.223**

3) Places365 ctest1k

The reason we conducted experiments on the Places365 ctest1k is that it was leveraged for machine learning by Iizuka *et al.* [7]. Therefore, we also attempted coloring on the first 1,000 images of the Places365 validation set. In general, the colors in our results were good enough to produce some vivid photos, as shown below.

4) COCO-STUFF ctest1k

Finally, there can be no shortage of experiments with the COCO-Stuff validation set. In the first image from the top in Fig. 19, which we evaluated compared to an original grayscale image, our SegClasRegSoft could distinguish food and others, so instances of color bleeding slightly decreased.

To have evaluations from different aspects, we selected a few images having low quality and small objects, but our methods were not successful, as shown on Fig. 20.

F. EXPERIMENT 3: INTERNET AND LEGACY RESULTS

We had a feeling that high-quality images would be advantageous. We randomly collected high-quality images from the Internet to make the ground-truth in the first line in Fig. 21. It can not be denied that the results are slightly better than those from Zhang *et al.* Our results in the first, fifth, sixth, and seventh columns had less noises colors. In the third column, our method predicted various colors for vegetables and fruits while the predictive colors by Zhang *et al.* had an excess of reddish brown. We compared with only Zang *et al.* because their predictions regarding colors look better than the other methods from the above experiments.

We also challenged ourselves by trying to color old historical images without colors. The results in Fig. 22 were not vivid due to brightness and quality, but our methods made these old images a bit more pleasing.

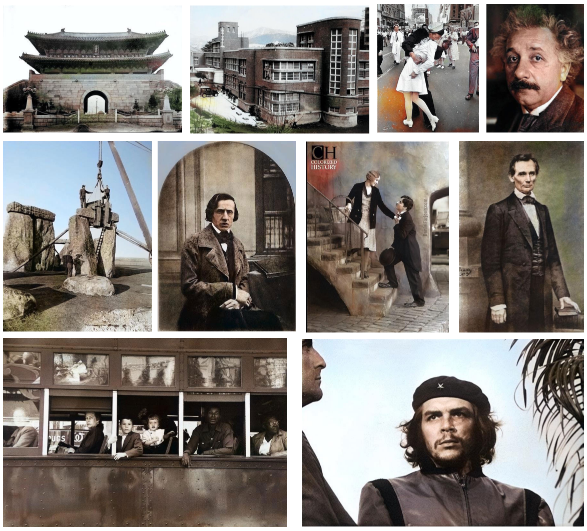


FIGURE 22. Legacy Colorization Images.

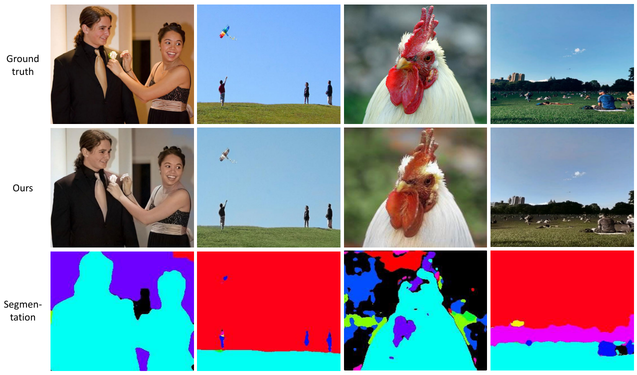


FIGURE 23. Segmentation Visualization.

G. EFFECTS OF SEGMENTATION ON COLORIZATION RESULTS

Fig. 23 shows visualizations of the segmentation results when our model colorized the image, they show that our model could capture the semantic segmentation to enhance the colorization results.

VI. CONCLUSION

In this paper, we proposed the encoder-decoder architecture to deal with the global and local semantics in the colorization problem. The encoding features are affected by the scene-context classification to bring the global image style to the image colorization in the global semantics. Moreover, it used the pre-trained weight from Places365 to exploit the uncertainty and relations among the scene labels by using the label smoothing from the top five probabilities in its prediction.

The image colorization at the pixel-level was fine-tuned by the semantic segmentation of both objects and the kinds of scenes from the Coco-Stuff dataset. This helped our model exploit the meanings of what objects the pixels belong to, such as a car or kite (objects) and a field or sky (scene). The final color version is the result of the mutual benefit learning of the semantic segmentation model with the average colorization model in the regression branch and the color distribution model in a soft-encoding branch where our model addresses the multi-model problem in colorization. Our validation experiments in ImageNet, DIV2K, and Places365 show good results.

In this study, we met the difficulties for image colorization in the cases such as many complex patterns, rare colors, many objects. It leaded to noise occurrences, incorrect colors and some regions without colorization. In the future, we will enhance the bias in the color selection from the color distribution model by the optimization process from multi-scale outputs. Moreover, we will use the generative model to colorize images better.

REFERENCES

[1] A. Jahanian, S. Keshvari, S. V. N. Vishwanathan, and J. P. Allebach, "Colors—Messengers of concepts: Visual design mining for learning color semantics," *ACM Trans. Comput.-Hum. Interact.*, vol. 24, no. 1, pp. 1–39, Mar. 2017.

[2] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH Papers SIGGRAPH*, 2004, pp. 689–694. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1186562.1015780

[3] G. Charpiat, M. Hofmann, and B. Schölkopf, "Automatic image colorization via multimodal predictions," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 5304, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Germany: Springer, 2008, doi: 10.1007/978-3-540-88690-7_10.

[4] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng, "Intrinsic colorization," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 1–9, Dec. 2008.

[5] Y. Morimoto, Y. Taguchi, and T. Naemura, "Automatic colorization of grayscale images using multiple images on the Web," in *SProc. IGGRAPH Posters SIGGRAPH*, 2009, p. 1.

[6] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 415–423.

[7] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, Jul. 2016, doi: 10.1145/2897824.2925974. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2897824.2925974 https://dl.acm.org/doi/

[8] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9908, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-46493-0_35.

[9] L. Zhang, Y. Ji, X. Lin, and C. Liu, "Style transfer for anime sketches with enhanced residual U-net and auxiliary classifier GAN," in *Proc. 4th IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2017, pp. 512–517.

[10] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015, May 2015, pp. 1520–1528. [Online]. Available: http://arxiv.org/abs/1505.04366

[11] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9907, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-46487-9_40.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI* (Lecture Notes in Computer Science), vol. 9351, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. Cham, Switzerland: Springer, 2015, doi: 10.1007/978-3-319-24574-4_28.

[13] H. Caesar, J. Uijlings, and V. Ferrari, "COCO-stuff: Thing and stuff classes in context," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1209–1218.

[14] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.

[15] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[17] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proc. 13th Annu. ACM Int. Conf. Multimedia - Multimedia*, 2005, pp. 351–354.

[18] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1214–1220, Jul. 2006.

[19] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Natural image colorization," in *Proc. 18th Eurograph. Conf. Rendering Technique*, 2007, pp. 309–320.

[20] T. Welsh, M. Ashikhmin, and K. Mueller, *Transferring Color to Greyscale Images*. New York, NY, USA: ACM Press, Jul. 2002, p. 277. [Online]. Available: http://portal.acm.org/citation.cfm?doid=566570.566576

[21] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Proc. Symp. Quart. J. Modern Foreign Literatures*, 2005, pp. 201–210. [Online]. Available: http://www.mendeley.com/research/colorization-by-example/

[22] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin, "Semantic colorization with Internet images," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 1–8, Dec. 2011.

[23] F. Shahbaz Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3306–3313.

[24] J. M. Sanchez and X. Binefa, "Improving visual recognition using color normalization in digital video applications," in *Proc. IEEE Int. Conf. Multimedia Expo. ICME. Latest Adv. Fast Changing World Multimedia*, Jul. 2000, pp. 1187–1190.

[25] S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, and J. Choo, "Coloring with limited data: Few-shot colorization via memory-augmented networks," 2019, *arXiv:1906.11888*. [Online]. Available: http://arxiv.org/abs/1906.11888

[26] H. Kim, H. Young Jhoo, E. Park, and S. Yoo, "Tag2Pix: Line art colorization using text tag with SECat and changing loss," 2019, *arXiv:1908.05840*. [Online]. Available: http://arxiv.org/abs/1908.05840

[27] T.-H. Sun, C.-H. Lai, S.-K. Wong, and Y.-S. Wang, "Adversarial colorization of icons based on structure and color conditions," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 683–691. [Online]. Available: http://arxiv.org/abs/1910.05253

[28] A. R. Robertson, "The CIE 1976 color-difference formulae," *Color Res. Appl.*, vol. 2, no. 1, pp. 7–11, Mar. 1977.

[29] T.-T. Nguyen-Quynh, N.-T. Do, and S.-H. Kim, "MLEU: Multi-level embedding U-Net for fully automatic image colorization," in *Proc. 4th Int. Conf. Mach. Learn. Soft Comput.*, Jan. 2020, pp. 119–123.

[30] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" 2019, *arXiv:1906.02629*. [Online]. Available: http://arxiv.org/abs/1906.02629

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. ICLR Conf. Track*, 2015, pp. 1–14.

[32] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognit.*, vol. 34, no. 12, pp. 2259–2281, Dec. 2001.

[33] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA ML-CDS* (Lecture Notes in Computer Science), vol. 10553, M. Cardoso *et al.*, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-67558-9_28.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[35] H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 6629–6640.

[36] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595. [Online]. Available: https://ieeexplore.ieee.org/document/8578166/

TRAM-TRAN NGUYEN-QUYNH received the B.S. degree in information system from the HCM City University of Foreign Language and Information Technology, Vietnam, in 2005, and the M.S. degree in information system management from International University, Vietnam National University at HCMC, Vietnam, in 2017. She is currently pursuing the M.S. degree with the Department of Computer Science, Chonnam National University, South Korea. From 2005 to 2009, she was a Web Programmer Freelancer. Then, she was a Lecturer with the Faculty of Information Technology, Vinatex Economic–Technical College of HCM, Vietnam, for a period of seven years. Next, she moved to the Faculty of Information Technology, HCM University of Foreign Languages and Information Technology, with a Lecturer Role. Her research interests include pattern recognition, deep learning, and computer vision.

SOO-HYUNG KIM (Member, IEEE) received the B.S. degree in computer engineering from Seoul National University, in 1986, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, in 1988 and 1993, respectively. Since 1997, he has been a Professor with the School of Electronics and Computer Engineering, Chonnam National University, South Korea. His research interests include pattern recognition, document image processing, medical image processing, and deep learning.

NHU-TAI DO (Graduate Student Member, IEEE) received the B.S. degree in information system from the HCM City University of Foreign Language and Information Technology, Vietnam, in 2005, and the M.S. degree in information system management from International University, Vietnam National University at HCMC, Vietnam, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Chonnam National University, South Korea. From 2005 to 2017, he was a Lecturer with the Faculty of Information Technology, HCM University of Foreign Languages and Information Technology, Vietnam. His research interests include pattern recognition, deep learning, computer vision, and parallel programming.