Depth-Aware Image Colorization Network

Wei-Ta Chu
National Chung Cheng University
Chiayi, Taiwan
wtchu@ccu.edu.tw

Yu-Ting Hsu
National Chung Cheng University
Chiayi, Taiwan
thounote@gmail.com

ABSTRACT

The color bleeding problem remains a challenging issue in image colorization. That is, different objects share the same color when they are nearby, leading to the boundary between objects looks unnatural. In this paper, we study how to combine depth information into a neural network and achieve better image colorization. The reasons to integrate depth information are twofold: (1) Depth information clearly provides boundary information between objects, and (2) depth information is commonly available as the development of RGB-D cameras. To the best of our knowledge, depth information was not considered in image colorization before. We evaluate the proposed method from both objective and subjective perspectives, and demonstrate that better colorization results can be obtained when depth information is further considered.

CCS CONCEPTS

• **Computing methodologies** → *Scene understanding*; *Neural networks*;

KEYWORDS

Image colorization, depth map, deep neural networks

ACM Reference Format:
Wei-Ta Chu and Yu-Ting Hsu. 2018. Depth-Aware Image Colorization Network. In *Understanding Subjective Attributes of Data, with the Focus on Evoked Emotions 2018 Workshop (EE-USAD'18), October 22, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/3267799.3267800

1 INTRODUCTION

An image colorization system assigns a suitable color to each pixel of the input grayscale image. This is a challenging task since it is under-constrained with very limited information available. Typically, image colorization methods can be roughly divided into three categories: scribble-based methods, example-based methods, and learning-based methods.

Scribble-based methods [11][13][6][15] request users to give some scribbles on the input grayscale image as the hints. This burdens users, and is not an effective way. On the other hand, example-based methods [2][4][8][12] predict colors of a grayscale image based on hints derived from a given reference image. However, such reference images are usually not readily available, and the performance of example-based methods highly depends on the reference image.

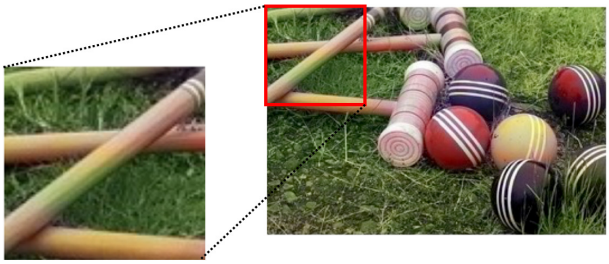

Figure 1: An example showing a colorization result with the color bleeding problem.

Recently, learning-based colorization methods [3][5][7][16][10] have achieved great success. Given a grayscale image, an automatic learning-based colorization system outputs a colorful image without any user input. However, the color bleeding issue remains a challenging and unsolved problem. As shown in Figure 1, from the colorization results including croquet clubs and balls on grass, the color of the shaft is not natural. The colors of mallet and shaft seem to be tan, but the color at the middle of the shaft is green. This makes the colorization result unnatural.

One possible reason for this unnatural result may be the limited description of object boundaries. Currently, depth information can be easily captured and is widely used in object detection, semantic segmentation, and many other applications. To diminish the color bleeding problem, we propose a colorization method considering depth information using a neural network and study how depth information benefits neural network-based colorization methods.

Figure 2 shows the framework of the proposed depth-aware method, which consists of the intensity-based prediction net, the depth-based prediction net, the fusion function, and the color mapping function. We mainly use convolutional neural networks (CNN) to construct two prediction nets. The input of the intensity-based prediction net is the grayscale image, and the input of the depth-based prediction net is the depth map. These two networks output the predicted distributions of colors for the given input image. We then fuse them and map the fused distribution into real colors by a mapping function.

The major contributions of this work are described as follows:

- We propose to consider depth information in a neural network-based colorization method. This model includes two sequences of convolutional blocks in parallel, and a fusion function is

EE-USAD'18, October 22, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5978-8/18/10…$15.00
https://doi.org/10.1145/3267799.3267800

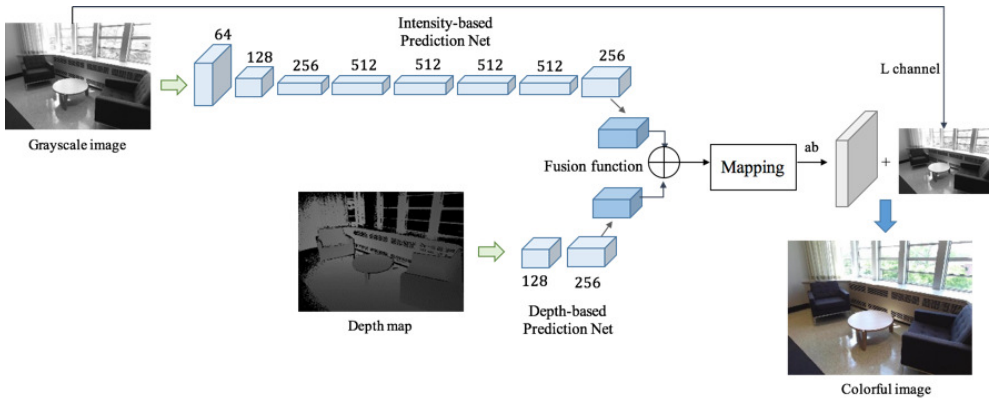

Figure 2: Illustration of the proposed depth-aware colorization network.

designed to combine two types of information in order to facilitate colorization.

- We have confirmed that depth information really helps colorization from both the objective and the subjective perspectives. To the best of our knowledge, this would be the first work verifying the effectiveness of depth in colorization.

2 RELATED WORKS

2.1 Scribble-based Methods

In scribble-based methods, users are requested to assign colors for some areas on a grayscale image. Levin et al. [11] proposed one of the first scribble-based colorization methods. Based on the input scribbles, colors are automatically propagated to the entire image with the formulated optimization algorithm. If the texture of the input image is complex, many scribbles should be given to obtain satisfactory. Luan et al. [13] developed an approach utilizing the texture feature to reduce user's efforts. Although the number of required scribbles is reduced, drawing scribbles is still a tedious process for users, not to say that colorization results highly rely on the input scribbles.

2.2 Example-based Methods

Scribble-based methods are not fully automatic since the user has to manually draw scribbles on a grayscale image. Although some methods were proposed to release the burden, it is still not efficient enough. Example-based methods use a reference image as the hint instead of scribbles. Users prepare reference images beforehand and then input the grayscale image and the prepared reference image to the system without manually drawing scribbles. However, finding reference images is a time-consuming task. Chia et al. [4] proposed a system where users can only input some semantic text labels to find reference images from the internet. After retrieving candidate images, the system automatically segments the input grayscale image and candidate images in order to get the salient foreground object. The candidate image having the most similar foreground object to the one in the grayscale image is chosen as the reference image. With this process, the internet serves a huge candidate pool to find reference images. Recently, Li et al. [12] proposed an approach using locality consistent sparse representation to do example-based colorization. A grayscale image can be colorized into the result with the color distribution very similar to the reference image. Again, the colorization results are highly influenced by the reference image.

2.3 Learning-based Methods

Learning-based methods learn how to perform colorization based on a large data collection. Cheng et al. [3] utilized a single neural network to do fully-automatic colorization. They designed a deep neural network with three fully-connected layers to extract features from the grayscale image. After colorizing the grayscale image, a joint bilateral filter was used to do post-processing for noise reduction. Iizuka et al. [7] developed deep neural networks with convolutional layers to extract features. They trained a classification network to judge whether this image is indoor or outdoor. The result of the classification network provides semantic information, and then they treated this semantic information as the global feature of an image. They fused this global feature with the local feature extracted from the colorization network to achieve better colorization. Zhang et al. [16] also proposed an approach based on CNN. They built a model to predict the distribution of feasible colors. In addition, they used a rebalance weight to further adjust the distribution of rare colors. In this paper, we implement this approach and further consider depth information to achieve better colorization results.

3 DEPTH-AWARE IMAGE COLORIZATION

We describe how to build the depth-aware colorization network (DACNet) that considers depth information to do colorization. Conceptually, given a grayscale image $\boldsymbol{X} \in \mathbb{R}^{H \times W \times 1}$, we would like to find a function $\mathcal{F}$ that outputs the distribution of colors $\hat{\boldsymbol{Y}}$ that is as close as to the true color distribution $\boldsymbol{Y} \in \mathbb{R}^{H \times W \times 2}$ (in terms of a and b values) of $\boldsymbol{X}$. The H and W denote height and width of the image, respectively. Several works have been proposed to find the function $\mathcal{F}$ based on neural networks [3]. Usually the L2 distance between $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$ is calculated as the loss to guide network training. However, this loss is not robust to the multimodal nature of the

image colorization problem, often yielding desaturated colorization results.

In order to handle the desaturation issue, Zhang et al. [16] viewed colorization as a multimodal classification task, and proposed a novel loss function to guide network training. They quantized real colors in the Lab color space into Q bins. Instead of directly predicting the most probable color for each pixel, Zhang et al. estimated a probability distribution of possible colors for each pixel. That is, given a grayscale image $\boldsymbol{X}$, a function $\mathcal{G}$ is to be determined to output $\hat{\boldsymbol{Z}} = \mathcal{G}(\boldsymbol{X}) \in [0,1]^{H\times W\times Q}$ indicating a distribution of possible colors for each pixel. The built network outputs the distribution $\hat{\boldsymbol{Z}}$ in terms of these Q color bins. To guide network training, the distribution $\boldsymbol{Z} = \mathcal{H}^{-1}(\boldsymbol{Y})$ transformed from the ground truth $\boldsymbol{Y}$ is compared with the estimated distribution $\hat{\boldsymbol{Z}}$.

Our work mainly follows the idea proposed in [16], with further consideration of depth information. Figure 2 shows the proposed DACNet, which consists of the intensity-based prediction network, the depth-based prediction network, the fusion function, and the mapping function. The intensity-based prediction net takes a grayscale image $\boldsymbol{X}_1$ as the input, and outputs a distribution of possible colors $\hat{\boldsymbol{Z}}_1 = \mathcal{G}_1(\boldsymbol{X}_1)$. The depth-based prediction net takes a depth map $\boldsymbol{X}_2$ as the input, and outputs another distribution of possible colors $\hat{\boldsymbol{Z}}_2 = \mathcal{G}_2(\boldsymbol{X}_2)$ according to depth information. These two distributions $\hat{\boldsymbol{Z}}_1$ and $\hat{\boldsymbol{Z}}_2$ are then fused by the fusion function. We compute the loss between the fused distribution and the ground truth, in order to guide the network to find best parameters. We describe details of these four components in the following.

3.1 Intensity-Based Prediction Network

From the given grayscale image $\boldsymbol{X}_1$, we would like to find a function $\mathcal{G}_1$ that maps $\boldsymbol{X}_1$ into the estimated distribution of colors $\hat{\boldsymbol{Z}}_1$. With this concept, we consider all possible colors for a pixel, and formulate image colorization as a multimodal classification problem.

To determine the function $\mathcal{G}_1$, we build the intensity-based prediction network consisting of several convolutional blocks to extract features from the input, and then generate a distribution based on these features. Eight convolutional blocks are included in this network, and the detailed configurations of these blocks are described in Table 1. In this table, "conv3-128", for example, denotes that the convolutional kernel is 3×3 pixels, and there are totally 128 feature maps after convolution. Each block consists of several convolutional layers and ReLU layers. Both Conv1 and Conv2 consist of two sets of convolutional layers and ReLU layers. Conv3 to Conv8 consist of three sets of convolutional layers and ReLU layers. Each block is followed by a BatchNorm layer, except for Conv8. The "dconv4-256" term in the last column denotes a 4×4 de-convolutional kernel.

3.2 Depth-Based Prediction Network

The input of the depth-based prediction network is a depth map $\boldsymbol{X}_2$. We would like to find a function $\mathcal{G}_2$ that maps $\boldsymbol{X}_2$ into the estimated distribution of colors $\hat{\boldsymbol{Z}}_2$. A depth map presents rich contour information, and we think it provides important clues to achieve better colorization. We build a CNN model consisting of three convolutional blocks to extract features.

Detailed settings of the depth-based prediction network are shown in Table 2. The first two convolutional blocks extract features from the depth map. There are two convolutional layers in a block. Each convolutional layer in a block is followed by a ReLU layer. The first convolutional block outputs 128 feature maps, and the second convolution block outputs 256 feature maps. We add a BatchNorm layer between two blocks to do batch normalization. The last block consists of one convolutional layer with kernel size set to 1×1. This layer aims to transform the feature into a distribution of Q bins. In this work, Q is set to 313 according to the setting in [16].

Based on various experimental results, the structure of the depth-based prediction net should not be too deep. If we use more convolutional layers, the performance will drop. According to the settings shown in Table 2, we can effectively extract depth information and predict the distribution of colors $\hat{\boldsymbol{Z}}_2$ well.

3.3 Fusion

After we obtain two types of distributions $\hat{\boldsymbol{Z}}_1 \in [0,1]^{H\times W\times Q}$ and $\hat{\boldsymbol{Z}}_2 \in [0,1]^{H\times W\times Q}$ respectively from the intensity-based prediction network and the depth-based prediction network, we fuse them in order to jointly consider two types of information. In this work, we combine $\hat{\boldsymbol{Z}}_1$ and $\hat{\boldsymbol{Z}}_2$ by element-wise addition, i.e., $\hat{\boldsymbol{Z}} = \hat{\boldsymbol{Z}}_1 \oplus \hat{\boldsymbol{Z}}_2$. The fused distribution $\hat{\boldsymbol{Z}} \in \mathbb{R}^{H\times W\times Q}$ represents the distribution of Q different colors for each pixel in the image of $H\times W$ pixels. Other fusion methods like concatenation can also be employed to obtain the fused distribution. However, in our experiments, we found element-wise addition performs better.

3.4 Loss Function and Rebalance Weight

To guide network training, previous works like [3] calculated the L2 distance between the ground truth colors $\boldsymbol{Y} \in \mathbb{R}^{H\times W\times 2}$ and the estimated colors $\hat{\boldsymbol{Y}} \in \mathbb{R}^{H\times W\times 2}$ as the loss function. Minimizing the distance makes the colorization network generate desaturated results, because colorizing pixels with average colors usually cause smaller L2 distance.

To overcome the aforementioned problem, Zhang et al. [16] viewed colorization as a multimodal classification problem, and proposed a multinomial cross entropy loss $L_c(\cdot,\cdot)$ as follows:

$$L_c\hat{\boldsymbol{Z}}, \boldsymbol{Z} = -\sum_{h,w} v(\boldsymbol{Z}_{h,w}) \sum_q \boldsymbol{Z}_{h,w,q} \log(\hat{\boldsymbol{Z}}_{h,w,q}). \tag{1}$$

The second summation denotes cross entropy between the estimated distribution $\hat{\boldsymbol{Z}}$ and the truth distribution $\boldsymbol{Z}$. The term $v(\cdot)$ denotes a weighting used to rebalance the loss based on color-class rarity, which will be described later. Therefore, Equation (1) is the multimodal cross entropy loss with a rebalanced weight term.

The weighting term is the key of the loss function. The pixels in the background strongly influence the colorization result since these they account for a large part of an image, such as wall and clouds. The term v is thus designed to weight the cross entropy in the training stage to address this class-imbalance problem. The weighting term is designed as follows:

$$v(\boldsymbol{Z}_{h,w}) = \boldsymbol{w}_{q^*}, \quad \text{where } q^* = \arg\max_q \boldsymbol{Z}_{h,w,q}. \tag{2}$$

$$w \propto \left((1-\lambda)\tilde{\boldsymbol{p}} + \lambda\frac{1}{Q}\right)^{-1}, \quad \mathbb{E}[\boldsymbol{w}] = \sum_q \tilde{\boldsymbol{p}_q}\boldsymbol{w}_q = 1. \tag{3}$$

Table 1: Detailed configuration of the intensity-based prediction network.

Conv1	Conv2	Conv3	Conv4	Conv5	Conv6	Conv7	Conv8
conv3-64 ReLU conv3-64 ReLU BatchNorm	conv3-128 ReLU conv3-128 ReLU BatchNorm	conv3-256 ReLU conv3-256 ReLU conv3-256 ReLU BatchNorm	conv3-512 ReLU conv3-512 ReLU conv3-512 ReLU BatchNorm	conv3-512 ReLU conv3-512 ReLU conv3-512 ReLU BatchNorm	conv3-512 ReLU conv3-512 ReLU conv3-512 ReLU BatchNorm	conv3-512 ReLU conv3-512 ReLU conv3-512 ReLU BatchNorm	dconv4-256 ReLU conv3-256 ReLU conv3-256 ReLU

Table 2: Detailed configuration of the depth-based prediction network.

Conv1	Conv2	Conv3
conv3-128 ReLU conv3-128 ReLU BatchNorm	conv3-256 ReLU conv3-256 ReLU	conv1-313

Equation (2) shows that each pixel is weighted by a factor $\boldsymbol{w} \in \mathbb{R}^Q$, based on its closest *ab* bin. The weighting factor determined by Equation (3) is proportional to the reciprocal of the mixture of the distribution $\tilde{\boldsymbol{p}}$ and the uniform distribution $\frac{1}{Q}$. The distribution $\tilde{\boldsymbol{p}}$ denotes how likely a color value appears in the world, acting like *a priori* distribution. According to [16], the distribution $\tilde{\boldsymbol{p}}$ is obtained from the statistics of the ImageNet training set, smoothed by a Gaussian kernel $\boldsymbol{G}_\sigma$. The expectation of $\boldsymbol{w}$ is set as 1 in order to do normalization. When the value of one color in a distribution is high, it means the color may be the background color, and thus we lower the weight. The rebalanced weight $v(\cdot)$ is thus detemined by the pixel color rarity.

This loss mentioned above is calculated to train the model. When testing, we directly use the fused distribution to do color mapping and obtain the colorization result. More details of this mapping function is described in the following section.

3.5 Color Mapping

After obtaining the estimated distribution of colors $\hat{\boldsymbol{Z}}$, we need to map it to point estimate $\hat{\boldsymbol{Y}}$ in the *ab* space. By combining the estimated *a* value and *b* value with the given *L* value from the input grayscale image, we finally determine the color for each pixel.

Intuitively, we can take the mode of $\hat{\boldsymbol{Z}}$, or the mean value of $\hat{\boldsymbol{Z}}$, to get the estimated *ab* values. However, the mode of $\hat{\boldsymbol{Z}}$ causes vibrant and sometimes spatially inconsistent results, while the mean of $\hat{\boldsymbol{Z}}$ causes desaturated results. To overcome this issue, Zhang et al. [16] proposed to take the annealed-mean of the distribution $\hat{\boldsymbol{Z}}$:

$$\mathcal{H}(\hat{\boldsymbol{Z}}_{h,w}) = \mathbb{E}[f_T(\hat{\boldsymbol{Z}}_{h,w})], \tag{4}$$

where the function f_T is a adjusted softmax function:

$$f_T(\boldsymbol{z}) = \frac{\exp(\log(\boldsymbol{z}/T))}{\sum_q \exp(\log(\boldsymbol{z}_q/T))}. \tag{5}$$

When the parameter T is set as 1, the function f_T acts as a common softmax function and leaves the input distribution unchanged. When the parameter T approaches 0, the function f_T tends to map the input distribution into a one-hot encoding at the distribution mode. In [16], they found that the parameter $T = 0.38$ captures the vibrancy of the mode and maintain spatial coherence well. We follow the same setting.

When testing, we run forward propagation of the depth-aware colorization network to generate two distributions $\hat{\boldsymbol{Z}}_1 = \mathcal{G}_1(\boldsymbol{X}_1)$ and $\hat{\boldsymbol{Z}}_2 = \mathcal{G}_2(\boldsymbol{X}_2)$ by the intensity-based prediction network and the depth-based prediction network, respectively. They are then fused to form the distribution $\hat{\boldsymbol{Z}}$. With the mapping function $\mathcal{H}$ mentioned in Equation (4), we obtain the point estimate $\hat{\boldsymbol{Y}} = \mathcal{H}(\hat{\boldsymbol{Z}})$ in the *ab* space.

4 EXPERIMENTAL RESULTS

4.1 Experimental Settings

We evaluate the proposed method based on two datasets, SUNRGBD [14] and Stanford 2D-3D-Semantics [1]. These two datasets have both color images and the associated depth information. For each dataset, we use 80% of the data for training, and use the rest 20% of data for testing. We convert color images into grayscale images, and then the grayscale images and the associated depth maps are input to the proposed DACNet.

We resize training images into 176×176 and develop DACNet based on the model in [16] by Caffe [9]. We train this network using the Adam optimizer and set the weight decay as 0.0005. The momentum and momentum2 used by the Adam optimizer are set as 0.9 and 0.99, respectively. The number of iterations is 12,000. The initial learning rate lr_{base} is set to 3.16e-05. Since we use the "step" learning policy in Caffe, the learning rate lr is dynamically changed according to the following equation:

$$lr = lr_{base} \times \gamma^{\lfloor iteration/stepsize \rfloor} \tag{6}$$

where γ and $stepsize$ are set as 0.316 and 4000, respectively.

We use some metrics to evaluate the distance between colorization results and the ground truth, including L1 distance, root-mean-square error (RMSE), and peak signal to noise ratio (PSNR). We also utilize the edge detection rate as the measure, because we think that good colorization results may improve results of edge detection. We apply the Canny edge detector to find edge pixels, and then compute the rate as the ratio of "the number of detected edge pixels that are also in the ground truth" to "the number of edge pixels in the ground truth". If this ratio is larger, the result of edge detection on the colorization result is more similar to the ground truth.

4.2 Experiments on the SUNRGBD Dataset

The SUNRGBD dataset contains 10,335 pairs of color images and their corresponding depth maps in several indoor scenes captured

Figure 3: Sample images and their associated depth maps in the the SUNRGBD dataset.

Table 3: Experimental results in terms of different metrics on the SUNRGBD dataset.

Metric	Zhang et al. [16]	Ours
L1 distance	7.92	6.69
RMSE	11.89	10.91
PSNR	62.12	64.44
Edge detection rate	0.8623	0.8706

by four cameras. We delete blur images and those not suitable in the colorization task. After deletion, 5,687 pairs of images are retained, where 4,600 pairs of images are used for training and 1,087 pairs are used for testing. Figure 3 shows some sample pairs in the SUNRGBD dataset.

Table 3 shows the experimental results in terms of different metrics. As can be seen, our model jointly considers features from the grayscale image and the depth map and outperforms [16] in terms of all metrics. This shows that depth information really aids in image colorization. Particularly from the improvement on edge detection rate, the depth information provides object contour information, and yields colorization results with clearer object boundaries.

Figure 4 shows sample colorization results obtained by [16] and our approach on the SUNRGBD dataset. We especially enlarge a region containing the boundary between two objects. The color bleeding problem can be clearly seen in the result of [16], while our approach provides largely improved colorization result. We also can see that our result is much similar to the ground truth.

4.3 Experiments on the Stanford 2D-3D-Semantics Dataset

The Stanford 2D-3D-Semantics dataset (2D-3D-S) provides 25,434 pairs of RGB images and corresponding depth maps. Images in this dataset are captured from three buildings of official and educational use. Figure 5 shows some sample pairs in the Stanford 2D-3D-S dataset.

Table 4 shows the experimental results in terms of different metrics. Again, our model outperforms [16] in terms of all metrics. Comparing the values in Table 4 with that in Table 3, we found the proposed DACNet performs even better in the Stanford 2D-3D-S dataset. The Stanford 2D-3D-S dataset mainly consists of

Table 4: Experimental results in terms of different metrics on the Stanford 2D-3D-S dataset.

Metric	Zhang et al. [16]	Ours
L1 distance	8.16	5.15
RMSE	12.69	8.02
PSNR	60.86	74.85
Edge detection rate	0.7820	0.7941

indoor scenes of office and education facilities. The same room is captured by a camera several times. Views of images in this dataset are sometimes close. This maybe the reason that the performance of DACNet on the Stanford 2D-3D-S dataset is better than the SUNRGBD dataset.

Figure 6 shows sample colorization results obtained by [16] and our approach on the Stanford 2D-3D-S dataset. As can be seen, our approach generates more natural colorization results.

We especially show details of a colorization result in Figure 7. We examine a specific 3×3 image patch in the original image and in the colorization results. The second row shows colors of the nine pixels in the patch, and the third row shows the estimated color distribution of the nine pixels. As can be seen, the colors of the marked region estimated by our method is more similar to ground truth. Based on this patch, the average L1 distance between our result and the ground truth is 4.30, while the average L1 distance between [16] and the ground truth is 13.40.

4.4 Subjective Evaluation

We try to describe colorization results in terms of quantitative values in Table 3 and Table 4. However, we know that evaluating colorization results is very subjective. Therefore, we design a subjective evaluation as follows.

We randomly juxtapose the original colorful image, the colorization result of [16], and the colorization result of our approach. We call these three images as a test triple. For each test triple, two questions were asked to 23 subjects:

- Question 1: Please sort these three images according to your preference, from favorite to least like.
- Question 2: Which image looks more natural in the real world? Sort these three images, from most likely to least likely.

For each question, the top-ranked image gets 3 points, and second-ranked image gets 2 points, and the last image gets 1 point. For each question, we randomly show five triples to each subject, and collect the average ranking results.

Figure 8 shows the average points of three different results. For Question 1, the left subfigure shows that the ground truth colorful images averagely get 2.5 points, and our approach averagely gets 2 points, which is unsurprising. More importantly, our approach outperforms [16] since only 1.5 points can be obtained by [16]. For Question 2, the right subfigure shows a very similar trend. These trends clearly verify that our approach yields better colorization results from the subjective perspective.

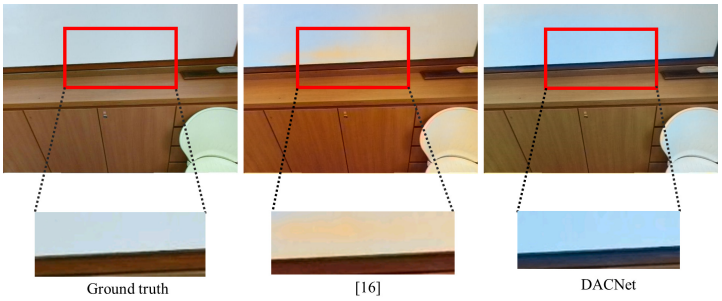

Figure 4: Sample colorization results on the SUNRGBD dataset.

Figure 5: Sample images and their associated depth maps in the Stanford 2D-3D-S dataset.

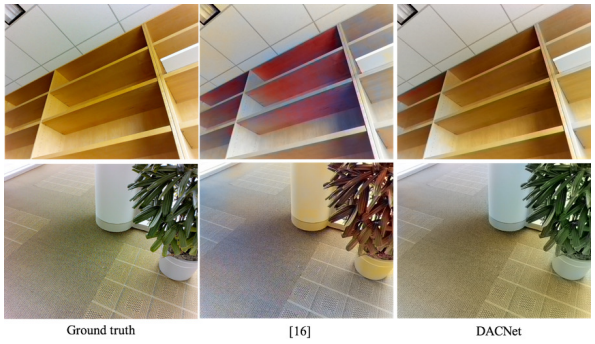

Figure 6: Sample results on the Stanford 2D-3D-S dataset.

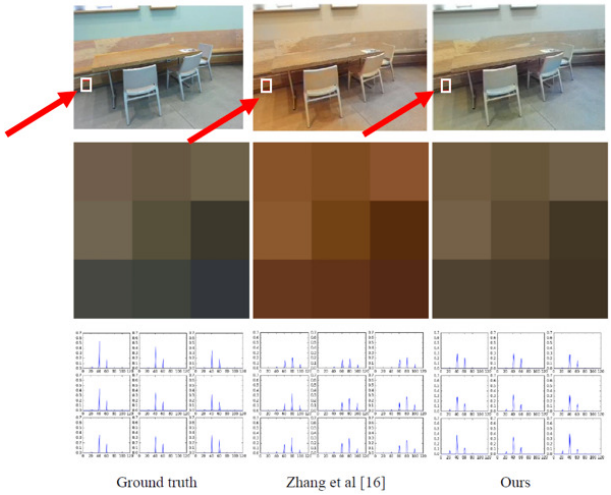

Figure 7: Sample details of a colorization result.

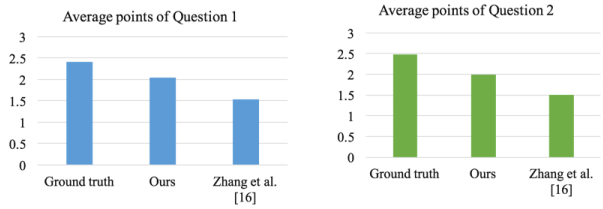

Figure 8: Average points of three different results. Left: results of Question 1; right: results of Question 2.

5 CONCLUSION

We propose a depth-aware image colorization neural network (DACNet) to do image colorization. The proposed method considers depth information to generate colorization images. We fuse the estimated color distribution from a grayscale image and the estimated color distribution from the corresponding depth map into one distribution. We then map this distribution to real colors and achieve the final colorization result. We comprehensively evaluate our colorization results through various metrics. The experimental results show that our proposed method outperforms another method from both quantitative and qualitative perspectives.

To further improve performance, we can utilize more semantic features extracted from grayscale images and depth maps. Currently, we assume that depth maps are given in advance. In the future, we would combine depth estimation in the proposed network, so that only the grayscale image should be input to the network.

Acknowledgement. This work was partially supported by the Ministry of Science and Technology of Taiwan under the grant MOST 107-2221-E-194-038-MY2 and MOST 107-2218-E-002-054.

REFERENCES

[1] I. Armeni, S. Sax, A.R. Zamir, and S. Savarese. 2017. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. In *arXiv preprint arXiv:1702.01105*.

[2] G. Charpiat, M. Hofmann, and B. Scholkopf. 2008. Automatic image colorization via multimodal predictions. In *Proceedings of European Conference on Computer Vision*. 126–139.

[3] Z. Cheng, Q. Yang, and B. Sheng. 2015. Deep colorization. In *Proceedings of IEEE International Conference on Computer Vision*. 415–423.

[4] A. Y.-S. Chia, S. Zhuo, R.K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin. 2011. Semantic colorization with internet images. *ACM Transactions on Graphics* 30, 6, Article 156 (2011).

[5] A. Deshpande, J. Rock, and D. Forsyth. 2015. Learning large-scale automatic image colorization. In *Proceedings of IEEE International Conference on Computer Vision*. 567–575.

[6] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. 2005. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of ACM International Conference on Multimedia*. 351–354.

[7] S. Iizuka, E. Simo-Serra, and H. Ishikawa. 2016. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics* 35, 4, Article 110 (2016).

[8] R. Ironi, D. Cohen-Or, and D. Lischinski. 2005. Colorization by example. In *Proceedings of Eurographics Conference on Rendering Techniques*. 201–210.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of ACM International Conference on Multimedia*. 675–678.

[10] G. Larsson, M. Maire, and G. Shakhnarovich. 2016. Learning representations for automatic colorization. In *Proceedings of European Conference on Computer Vision*. 577–593.

[11] A. Levin, D. Lischinski, and Y. Weiss. 2004. Colorization using optimization. *ACM Transactions on Graphics* 23, 3 (2004), 689–694.

[12] B. Li, F. Zhao, Z. Su, X. Liang, Y.-K. Lai, and P.L. Rosin. 2017. Example-based image colorization using locality consistent sparse representation. *IEEE Transactions on Image Processing* 26, 11 (2017), 5188–5202.

[13] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. 2007. Natural image colorization. In *Proceedings of Eurographics Conference on Rendering Techniques*. 309–320.

[14] S. Song, S.P. Lichtenberg, and J. Xiao. 2015. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

[15] L. Yatziv and G. Sapiro. 2006. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing* 15, 5 (2006), 1120–1129.

[16] R. Zhang, P. Isola, and A.A. Efros. 2016. Colorful image colorization. In *Proceedings of European Conference on Computer Vision*. 649–666.