

Pixel-level Semantics Guided Image Colorization

Jiaojiao Zhao¹

j.zhao3@uva.nl

Li Liu²

liuli1213@gmail.com

Cees G.M. Snoek¹

cgmsnoek@uva.nl

Jungong Han³

jungonghan77@gmail.com

Ling Shao²

ling.shao@ieee.org

¹ University of Amsterdam,
The Netherlands

² Inception Institute of Artificial
Intelligence,
UAE

³ Lancaster University,
UK

Abstract

While many image colorization algorithms have recently shown the capability of producing plausible color versions from gray-scale photographs, they still suffer from the problems of context confusion and edge color bleeding. To address context confusion, we propose to incorporate the pixel-level object semantics to guide the image colorization. The rationale is that human beings perceive and distinguish colors based on the object's semantic categories. We propose a hierarchical neural network with two branches. One branch learns what the object is while the other branch learns the object's colors. The network jointly optimizes a semantic segmentation loss and a colorization loss. To attack edge color bleeding we generate more continuous color maps with sharp edges by adopting a joint bilateral upsampling layer at inference. Our network is trained on PASCAL VOC2012 and COCO-stuff with semantic segmentation labels and it produces more realistic and finer results compared to the colorization state-of-the-art.

1 Introduction

Colorizing a gray-scale image [3, 4, 6, 10, 17, 25] has wide applications in a variety of computer vision tasks, such as image compression [10], outline and cartoon creations [10, 17], and infrared images [20] and remote sensing images colorizations [10]. Human beings excel in assigning colors to gray-scale images as they can easily recognize the objects and have gained knowledge about their colors. No one doubts the sea is typically blue and a dog is naturally never green. Certainly, lots of objects have diverse colors which makes the prediction quite subjective. However, it remains a big challenge for machines to acquire both the world knowledge and “imagination” that humans possess. Previous works require reference images [13, 23] or color scribbles [20] as guidance. Recently, several automatic approaches [15, 19, 28, 30, 31] were proposed based on deep convolutional neural networks.

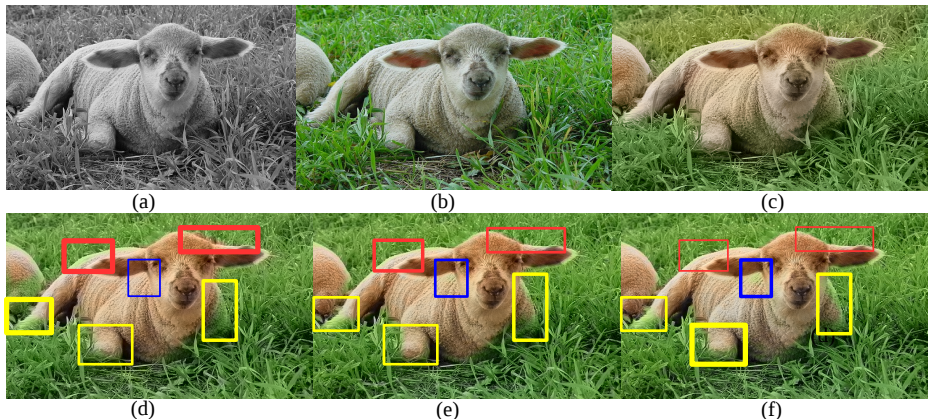


Figure 1: Colorization results generated using the same input image but at different scales. (a) gray-scale image (300x500); (b) ground-truth (300x500); (c) proposed colorized version of the input image at size 224x224; colored versions of [60] with the input images at sizes: (d) 176x176; (e) 224x224; (f) 300x300. We highlight some parts for edge bleeding (red boxes) and context confusion (blue boxes) to compare the results with various scales. For all boxes, the thicker the linewidth, the worse the result. Furthermore, even for the difficult parts (yellow boxes), the proposed method has better colorization abilities. Best viewed in color on zoomed-in screen.

Despite the improved colorization, there are still common pitfalls that make the colorized images appear less realistic. For example, color confusion in objects caused by incorrect semantic understanding and boundary color bleeding by scale variation. Our objective is to effectively address both problems to generate better colorized images with high quality.

Both traditional [2, 16] and recent colorization solutions [15, 19, 30] have highlighted the importance of semantics. In [30, 31], Zhang *et al.* apply cross-channel encoding as self-supervised feature learning with semantic interpretability. In [19], Larsson *et al.* claim that interpreting the semantic composition of the scene and localizing objects are key to colorizing arbitrary images. Larsson *et al.* pre-train a network on ImageNet for a classification task which provides the global semantic supervision. Iizuka *et al.* [15] leverage a large-scale scene classification database to train a model, exploiting the class-labels of the dataset to learn the global priors. These works only explore the image-level classification semantics. As it is stated in [8], the image-level classification task favors translation invariance. Obviously, colorization task needs representations that are translation-variant to an extent. From this perspective, semantic segmentation task which also needs translation-variant representations is more reasonable to provide pixel-level semantics for colorization. Deep CNNs have shown great successes on semantic segmentation [6, 29], especially with deconvolutional layers [26]. It gives a class label to each pixel. Similarly, referring to [19, 30], colorization assigns each pixel a color distribution. Both challenges can be viewed as an image-to-image prediction problem and formulated as a pixel-wise classification task. Our proposed network is able to harmoniously train with two loss functions of semantic segmentation and colorization.

Edge color persistence is another common problem for existing colorization methods [14, 19, 22, 30]. To speed up training and reduce memory consumption, deep convolutional

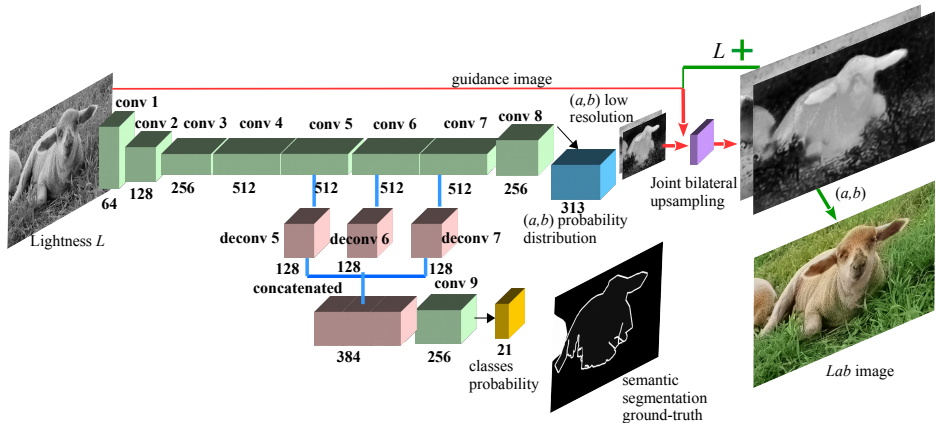


Figure 2: Our hierarchical network structure includes semantic segmentation and colorization. The semantic branch learns the pixel-wise object classes for the gray-scale images, which acts as a coarse classification. The colorization branch performs a finer classification according to the learned semantics. We apply multipath deconvolutional layers to improve semantic segmentation. At inference, a joint bilateral upsampling layer is added for predicting the colors.

neural networks prefer to take small fixed-sized images as inputs. However, test images may be at any scale compared to the resized training images. In this case, the well-trained model imposes a conflict between the semantics and edge color persistence on test images. In Figure 1, we present some results produced by [10] when the input image is at different scales. The smaller the scale of the input image, the better understanding of the object colors but the worse the edge colors. Moreover, the downsampling and upsampling operations in the networks also cause edge color blurring. Inspired by the idea of joint bilateral filtering [13] which keeps edges clear and sharp, we propose a joint bilateral upsampling (JBU) layer for producing more continuous color maps of the same size with the original gray-scale image.

Our contributions include: (1) We propose a multi-task convolutional neural network to learn what the object is and the colors of the object should be. (2) We propose a joint bilateral upsampling layer to generate a color inference from a color distribution. The method produces realistic color images with sharp edges. (3) The two strategies can be embedded in many existing colorization networks.

2 Methodology

We propose a hierarchical architecture to jointly optimize semantic segmentation and colorization. In order to estimate a specific color for each pixel from a color distribution, we propose a joint bilateral upsampling layer at the test phase. The architecture is illustrated in Figure 2 and detailed next.

2.1 Loss Function with Semantic Priors

We consider the CIE *Lab* color space to perform the colorization task as only two channels *a* and *b* need to be learned. The lightness channel *L* with a height *H* and a width *W* is defined

as an input $\mathbf{X} \in \mathbf{R}^{H \times W \times 1}$ and the output $\hat{\mathbf{Y}} \in \mathbf{R}^{H \times W \times 2}$ represents the two color channels a, b . The colorization problem is to learn a mapping function $f: \mathbf{X} \rightarrow \hat{\mathbf{Y}}$. Following the work in [50], we divide the color ab space into $Q = 313$ bins where Q is the number of discrete ab values. The deep neural network shown in Figure 2 is constructed to encode $\hat{\mathbf{Z}} = G(\mathbf{X})$ to a probability distribution over possible colors $\hat{\mathbf{Z}} \in [0, 1]^{H \times W \times Q}$. Given a ground-truth \mathbf{Z} , a multinomial cross entropy loss function with class-rebalance for colorization L_c is formulated as:

$$L_c(\hat{\mathbf{Z}}, \mathbf{Z}) = - \sum_{h,w} v_c(\mathbf{Z}_{h,w}) \sum_q \mathbf{Z}_{h,w,q} \log(\hat{\mathbf{Z}}_{h,w,q}), \quad (1)$$

where $v_c(\cdot)$ indicates the weights for rebalancing the loss based on color-class rarity.

We jointly learn the other loss function L_s specifically for semantic segmentation. Generally, semantic segmentation should be performed in the RGB image domain due to that colors are important for semantic understanding. However, the input of our network is a gray-scale image which is more difficult to segment. Fortunately, the network incorporating colorization learning supplies color information which in turn strengthens the semantic segmentation for gray-scale images. The mutual benefit between the two learning parts is the core of our network. Actually, semantic segmentation, as a supplementary means for colorization, is not required to be very precise. We define a weighted cross entropy loss with the standard softmax function \mathbf{E} for semantic segmentation as:

$$L_s(\mathbf{X}) = - \sum_{h,w} v_s(\mathbf{X}_{h,w}) \log(\mathbf{E}(\mathbf{X}_{h,w}; \theta)), \quad (2)$$

where $v_s(\cdot)$ is the weighting terms to rebalance the loss based on object-category rarity.

Finally, our loss function L is a combination of L_c and L_s and can be jointly optimized:

$$L = \lambda_c L_c + \lambda_s L_s, \quad (3)$$

where λ_i is the weights to balance the losses for colorization and semantic segmentation.

2.2 Inference by Joint Bilateral Upsampling Layer

There are some strategies for a point estimation from a color distribution according to [19, 30]. Usually, taking the mode of the prediction distribution for each pixel will provide a vibrant result but with splotches. Alternatively, applying the mean of the distribution will produce desaturated results. In order to find a balance between the two factors, Zhang *et al.* in [30] propose to use the annealed-mean of the distribution. The trick helps to achieve more acceptable results but cannot predict the edge colors well. We draw inspiration from the joint bilateral filter [18] to address the issue. The joint bilateral filter uses both a spatial filter kernel on the initially generated color maps and a range filter kernel on a second guidance image (the gray-scale image here) to estimate the color values. More formally, for one position p , the filtered result on the color channel c ($= a, b$) is:

$$J^c_p = \frac{1}{k_p} \sum_{q \in \Omega} I^c_q f(\|p - q\|) g(\|\tilde{I}_p - \tilde{I}_q\|), \quad (4)$$

where f is the spatial filter kernel, e.g., a Gaussian filter, and g is the range filter kernel, centered at the gray-scale image (\tilde{I}) intensity value at p . Ω is the spatial support of the kernel f , and k_p is a normalizing factor. Edges are preserved since the bilateral filter $f \cdot g$ takes on

smaller values as the range distance and/or the spatial distances increase. Thus, the strategy hits three birds with one stone. That is, it decreases the splotches, keeps the colors saturated and makes the edges sharp and clear.

The input size of the network is usually small to speed-up training and reduce memory consumption so the outputs are low-resolution color maps. In order to get a colored version at any test image resolution but with fine edges, we further adopt the joint bilateral upsampling (JBU) method. Let p and q denote (integer) coordinates of pixels in the gray-scale image \tilde{I} , and p_{\downarrow} and q_{\downarrow} denote the corresponding (possibly fractional) coordinates in the low resolution output S^c , the upsampled solution \tilde{S}^c is:

$$\tilde{S}_p^c = \frac{1}{k_p} \sum_{q_{\downarrow} \in \Omega} S_{q_{\downarrow}}^c f(\|p_{\downarrow} - q_{\downarrow}\|) g(\|\tilde{I}_p - \tilde{I}_q\|), \quad (5)$$

We implement the joint bilateral upsampling using a neural network layer resulting in an end-to-end solution at inference.

2.3 Network Architecture

Our hierarchical network structure is specifically shown in Figure 2. The bottom layers conv1-conv4 are shared by the two tasks for learning the low level features. The high-level features contain more semantic information. We add three deconvolutional layers respectively after the top layers conv5, conv6 and conv7. Then the feature maps from the deconvolutional layers are concatenated for semantic segmentation, which is appropriate to capturing the fine-details of an object. Intuitively, the network will firstly recognize the object and then assign colors to the object. At the training phase we jointly learn the two tasks and at the test phase a joint bilateral upsampling layer is added to produce the final results.

3 Experiments

3.1 Experimental Settings

Datasets: Two datasets including the PASCAL VOC2012 [9] and the COCO-stuff [10] are used. The former one is a common semantic segmentation dataset with 20 object classes and a background class. Our experiments are performed on the 10582 images for training and the 1449 images in validation set for testing. The COCO-stuff is a subset of the COCO dataset [11] generated for scene parsing, containing 182 object classes and a background class on 9000 training images and 1000 test images. Each input image is rescaled to 224x224.

Implementation details: Commonly available pixel-level annotations intended for semantic segmentation are sufficient for our method to improve colorization. We don't need new pixel-level annotations for colorization. We train our network with joint semantic segmentation and colorization losses with the weights $\lambda_s : \lambda_c = 100 : 1$ so that the two losses are similar in magnitude. Our multi-task learning for simultaneously optimizing colorization and semantic segmentation effectively avoids overfitting. 40 epochs are trained for the PASCAL VOC2012 and 20 epochs for the COCO-stuff. A single epoch takes approximately 20 minutes on a GTX Titan X GPU. The run time for the network is about 25 ms per image and the model size is 147.5 Mb, which are a little worse than those of [6] (22 ms and 128.9 Mb), as we have a few more layers for semantic segmentation. They are much better than [19] with a run time of 225 ms and a model size of 561 Mb and [13] with a run time of 80 ms and a model size of

694.7 Mb. When performing JBU, the domain parameter σ_s for the spatial Gaussian kernel is set to 3 and the range parameter σ_r for the intensity kernel is set to 15.

3.2 Illustration of Reasonability of the Strategies

A simple experiment is performed for stressing that colors are critical for semantic segmentation. We apply the Deeplab-ResNet101 model [15] trained on the PASCAL VOC2012 training set for semantic segmentation and test on three versions of the validation images, including gray-scale images, original color images and our colorized images. The mean intersection over union (IoU) is adopted to evaluate the segmentation results. As seen in Figure 3, with the original color information, the performance 86% is much better than that of the gray images 79%. The performance of our proposed colorized images is 5% lower than that of the original RGB images. One reason is the imperfection of the generated colors. We believe another reason is that the model was trained on the original color images. However, as we state above our generated color images are not learned to be the ground-truth. So the 5% difference can be acceptable. More importantly, the proposed colorized images outperform the gray images by 2%, which well supports the importance of colors on semantic understanding.

As for the joint bilateral upsampling, one may be concerned about the resolution problem. Color maps are not dominant on the resolution of the color images but the lightness channel is. A rough comparison of the peak signal-to-noise ratio (PSNR) on the images under three conditions is shown in Table 1. In the table, we list the means of the PSNR over the generated images without semantic information or JBU, only with semantic information, and with semantic information and JBU. Obviously, the three different settings have very similar PSNRs. The joint bilateral upsampling does not affect the quality so much but helps to preserve edge colors.

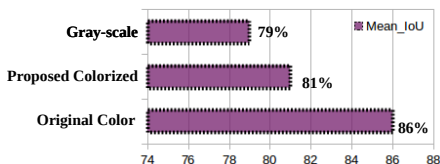


Figure 3: Segmentation results in terms of Mean IoU of gray, proposed colorized and original color images on PASCAL VOC2012 validation dataset. Color aids semantic segmentation.

Method	PSNR
Without Semantics or JBU	22.7
Only With Semantics	22.3
With Semantics & JBU	22.0

Table 1: Similar Mean of PSNR under three different settings on PASCAL VOC2012 validation dataset. Joint bilateral upsampling does not affect the image quality so much.

3.3 Ablation Study

We compare our proposed methods in two settings: (1) only using semantics and (2) using semantics and JBU, with the state-of-the-art. Some successful cases on the two datasets are shown in Figure 4. The first three rows are from the PASCAL VOC2012 validation dataset and next two rows are from the COCO-stuff. As shown in the figure, the results from [15, 19] look grayish because colorization is treated as a regression problem in the two pipelines. In the first row, the maple leaves are not assigned correct colors by [15, 19]. The sky and the tail are polluted in [15]. In the fourth row, the edges of the skirt are not sharp in the first

three columns. The results from [80] are more saturated but suffer from edge bleeding and color pollution. However, by injecting semantics our methods result in better perception of the object colors. To emphasize the effect of JBU, we zoom in on local areas of the results before and after JBU. One can clearly observe the details of the edge colors. JBU helps to achieve the finer results such as the sharp edges of the maple leaf in the first row and the clear edge of the skirt in the fourth row.

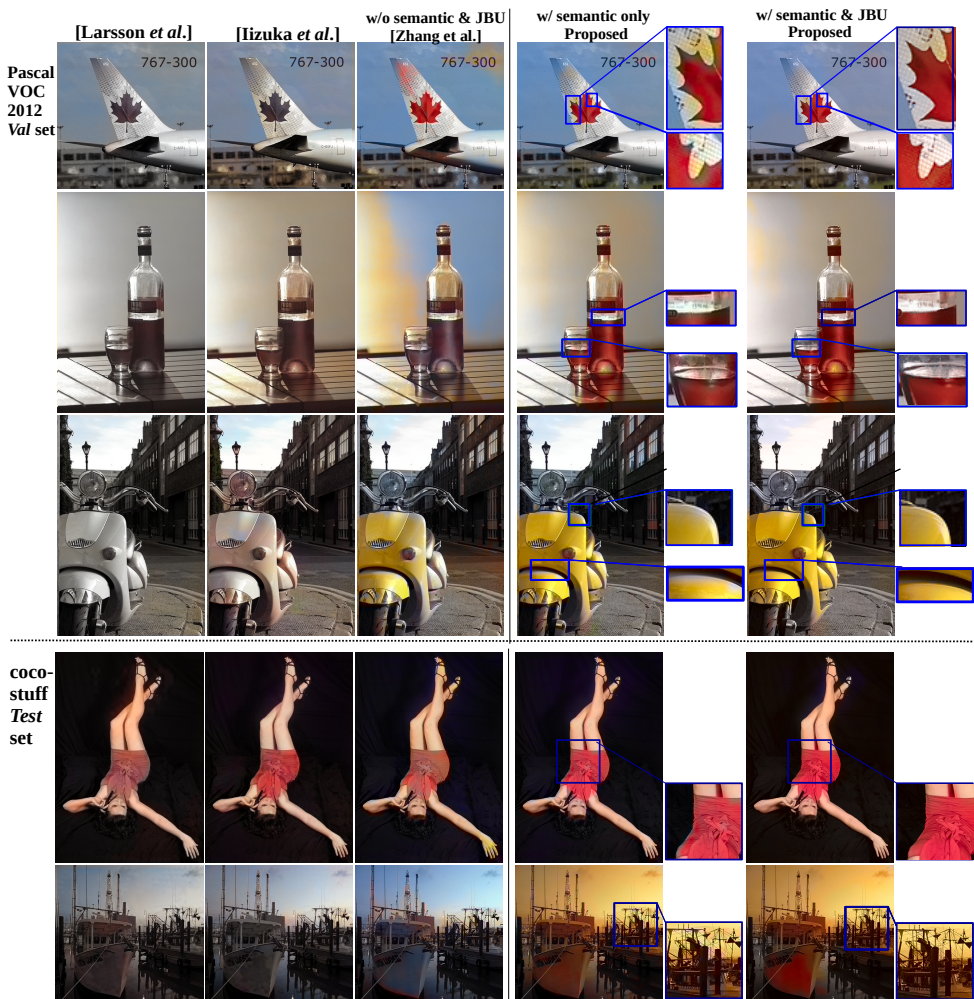


Figure 4: Example colorization results comparing the proposed methods with the state-of-the-art on Pascal VOC 2012 validation and COCO-stuff test datasets. Where the state-of-the-art suffers from desaturation, color pollution and edge bleeding, our proposed methods, with semantic priors, have better content consistency. Furthermore, with JBU, the edge colors are preserved well. We also show some local parts for detailed comparison between the edges. More results generated by our model are shown in supplementary material.

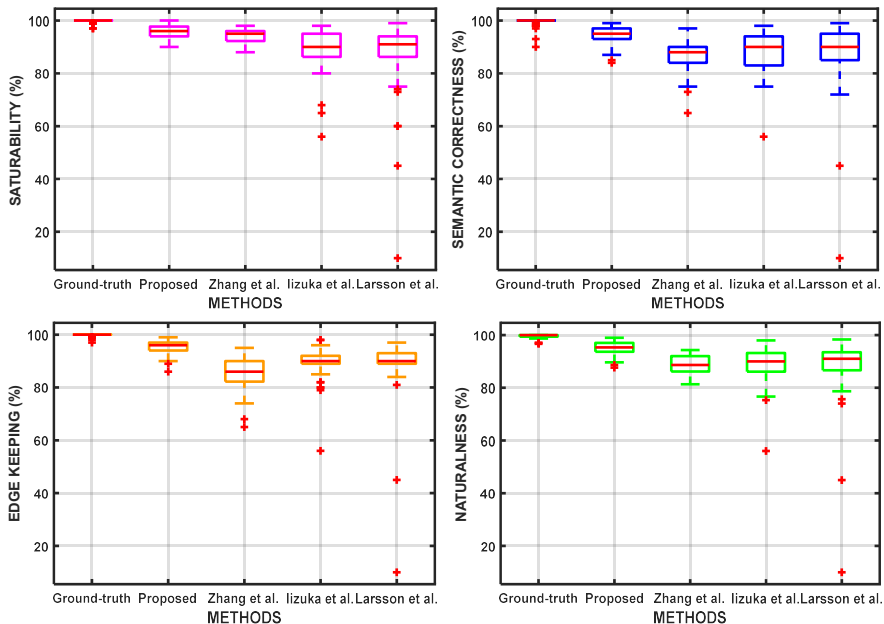


Figure 5: Comparison of (a) Saturability (b) Semantic Correctness (c) Edge Keeping and (d) Overall Naturalness. Our proposed method gets better performance in all criteria.

3.4 Comparisons with State-of-the-art

Generally, we want to produce visually compelling results which can fool a human observer, rather than recover the ground-truth. Quantitative colorization metrics may penalize reasonable, but different with ground-truth colors, especially for some artifacts (*e.g.* red air balloon or green air balloon). As discussed above, colorization is a subjective issue. So qualitative results are even more important than quantitative. Similar with most papers including [15, 30], we ask 20 human observers to do a real test on a combined dataset including the PASCAL VOC2012 validation and the COCO-stuff subset. Given a color image produced by our method or the three compared methods [15, 19, 30] or the real ground-truth image, the observers should decide whether it looks natural or not. We propose three metrics including saturability, semantic correctness and edge keeping for evaluating the naturalness. The overall naturalness is the equally weighted sum of the three values. Images are randomly selected and shown one-by-one in a few seconds to each observer. Finally, we calculate the percentage of the four criteria for each image and draw the error bar figures for comparing the images generated by different approaches (shown in Figure 5). The method in [30] can produce rich colors but always with bad edges. The method in [15] keeps clear edges. Our proposed method performs better and is closer to the ground-truth. Table 2 shows the means of the four criterion percentages of each approach. Our automatic colorization method outperforms the others considerably. We present some examples in Figure 6 and label the four criterion values. For a fair comparison, the images in the last three rows are generated by the three references. It means they are taken as successful cases respectively in the three references. However, the results from [15, 19] look grayish in the first row. In the second row, the colors of the sky and the river from the state-of-the-art seem abnormal. In the third row,

the building from [30] is polluted and the buildings from [15, 19] are desaturated. Overall, our results look more realistic and saturated.

Method	Saturability (%)	Semantic Correctness (%)	Edge Keeping (%)	Naturalness (%)
Iizuka <i>et al.</i>	89.00	87.90	88.90	88.61
Larsson <i>et al.</i>	86.00	86.80	88.00	86.99
Zhang <i>et al.</i>	94.00	86.50	85.40	88.66
This Paper	95.70	94.10	94.80	94.89
Ground-truth	99.69	99.27	99.76	99.58

Table 2: Comparison of Naturalness between the state-of-the-art and the proposed method on the combined dataset. The mean value of each criterion is shown. Our proposed method has better performance.

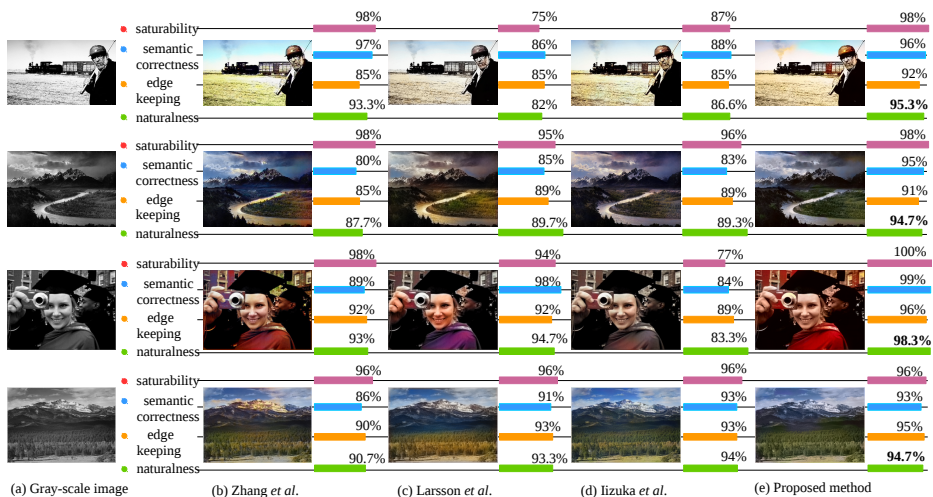


Figure 6: Exemplar comparison of Naturalness (includes Saturability, Semantic Correctness and Edge Keeping) with the state-of-the-art automatic colorization methods. (a) gray-scale image; (b) [30]; (c) [19]; (d) [15]; (e) proposed method. Our method can produce more plausible and finer results.

3.5 Failure Cases

Our method can output plausible colorized images but it is not perfect. There are still some common issues encountered by the proposed approach and also other automatic systems. We provide a few failure cases in Figure 7. It is believed that incorrect semantic understanding results in unreasonable colors. Though we incorporate semantics for improving colorization,

there are not enough categories. We assume a finer semantic segmentation with more class labels will further enhance the results.

4 Conclusion

In this paper, we address two general problems of current automatic colorization approaches: context confusion and edge color bleeding. Our hierarchical structure with semantic segmentation and colorization was designed to strengthen the ability of semantic understanding so that content confusion will be reduced. And our joint bilateral upsampling layer successfully preserves edge colors at inference. We achieved satisfying results in most cases. Our code will be released to foster further improvements in the future.



Figure 7: Failure cases. *Top row, left-to-right*: not enough colors; incorrect semantic understanding; *bottom row, left-to-right*: background inconsistency; small object confusion (leaves and apples); lack of object categories (peacock, jewelry).

References

- [1] Mohammad Haris Baig and Lorenzo Torresani. Multiple hypothesis colorization and its application to image compression. *Computer Vision and Image Understanding*, 164: 111–123, 2017.
- [2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. *arXiv:1612.03716*, 2016.
- [3] Yun Cao, Zhiming Zhou, Weinan Zhang, and Yong Yu. Unsupervised diverse colorization via generative adversarial networks. *arXiv:1702.06674*, 2017.
- [4] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. In *European Conference on Computer Vision*, pages 126–139, 2008.
- [5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous

- convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
- [6] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *IEEE International Conference on Computer Vision*, 2015.
- [7] Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, and Yu-Wing Tai. Semantic colorization with internet images. *ACM Transactions on Graphics (TOG)*, 30(6), 2011.
- [8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Neural Information Processing Systems(NIPS)*, 2016.
- [9] Mark Everingham, Luc Van Gool, Christopher K.L. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge 2012 (voc2012) results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>, 2012.
- [10] Kevin Frans. Outline colorization through tandem adversarial networks. *arXiv:1704.08834*, 2017.
- [11] Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens, and Kevin Murphy. Pixcolor: Pixel recursive colorization. *arXiv:1705.07208*, 2017.
- [12] Jiayi Guo, Zongxu Pan, Bin Lei, and Chibiao Ding. Automatic color correction for multisource remote sensing images with wasserstein cnn. *Remote Sensing*, 9:483, 2017.
- [13] Raj Kumar Gupta, Alex Yong-Sang Chia, Deepu Rajan, Ee Sin Ng, and Zhiyong Huang. Image colorization using similar images. In *ACM international conference on Multimedia*, pages 369–378, 2012.
- [14] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. An adaptive edge detection based colorization algorithm and its applications. In *ACM international conference on Multimedia*, pages 351–354, 2005.
- [15] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. In *Conference on Special Interest Group on Computer GRAPHics and Interactive Techniques*, 2016.
- [16] Revital Irony, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. In *EGSR '05 Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*, pages 201–210, 2005.
- [17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Patten Recognition*, 2017.
- [18] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3), 2007.
- [19] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, 2016.

- [20] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Transactions on Graphics (TOG) (Proceedings of ACM SIGGRAPH 2004)*, 23:689–694, 2004.
- [21] Matthias Limmer and Hendrik P.A. Lensch. Infrared colorization using deep convolutional neural networks. In *IEEE International Conference on Machine Learning and Applications*, 2016.
- [22] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *arXiv:1405.0312*, 2014.
- [23] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. Intrinsic colorization. *ACM Transactions on Graphics (TOG) (Proceedings of ACM SIGGRAPH Asia 2008)*, 2008.
- [24] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *EGSR'07 Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320, 2007.
- [25] Yuji Morimoto and Yuichi Taguchi, and Takeshi Naemura. Automatic colorization of grayscale images using multiple images on the web. In *ACM Conference on Special Interest Group on Computer GRAPHics and Interactive Techniques*, 2009.
- [26] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision*, 2015.
- [27] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. Manga colorization. In *ACM Conference on Special Interest Group on Computer GRAPHics and Interactive Techniques*, 2006.
- [28] Amelie Royer, Alexander Kolesnikov, and Christoph H. Lampert. Probabilistic image colorization. In *British Machine Vision Conference*, 2017.
- [29] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:640–651, 2016.
- [30] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, 2016.
- [31] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S. Lin, and Alexei A. Efros. Real-time user-guided image colorization with learned deep priors. In *Conference on Special Interest Group on Computer GRAPHics and Interactive Techniques*, 2017.