Colorization Based on Semantic Segmentation

Zhou Haipeng

Abstract

Colorization is an important area in image processing and computer vision. Given a grayscale image, the task is to transfer it to be colorful. Currently the *state of the art* models display impressive performance with using advanced neural network. Though some note that semantic information plays a significant role in colorization, almost none of such a model combined with semantic segmentation. The leading cause of our proposed is that semantic information shares coarse color clues in the same class, like people's hair is mainly black, etc. Our method is composed of 2 parts: the semantic segmentation and colorization networks, which are enabled to leverage the semantic features and chrominance information. According to our experimental results, we find that with combined the semantic segmentation the color region will be centralized while the normal colorization subnet shared sparse color. The ideal output results demonstrate the innovation and feasibility of our works, though our model encounters some failed examples on account of lack trainable data.

Keywords: colorization, semantic segmentation, deep learning

1. Introduction

Colorization is a multi-solvable and ill-posed problem in Computer Vision. It is of great significance since the color image is visually plausible and perceptually meaningful and has enormous potential in practical applications such as image restoration, black and white movie rendering, 3D modeling, etc. There is no fixed mapping from grayscale to colorful image due to the ambiguity and diversity when processing it. This multimodal task could be challenging, noting that the sky is blue on a sunny day, but it can also become darker when raining. At the same time, the one-channel image indicated none of the information about it. However, the color clues are hidden in the gray photographs. We are still able to handle powerful feature extraction tools, especially from retaining semantic information.

Traditional Colorization is user-guided work, which requires abundant human interactions like choosing the segmentation, placing numerous color scribbles and referring to related images, etc. These early attempts highly depend on the user scribbles to guide the image processing [1] [2] [3]. With the development of deep learning, some of the *state of the art* models combined the interaction with advanced neural networks [4]. Colorization could also be regarded as a style transfer work, which means we can refer to other colorful images to extract color for the grayscale image as close as possible. Moreover, it enlightens other researchers to directly extract the color

information from the reference colorful images and fill it to the grayscales [5] [6] with referring the huge dataset like ImageNet [7] and COCO-Stuff [8]. But such kinds of methods rely on the quality of interactions or exemplar image. Other groups [9] [10] proposed data-driven models, which means End to End transfer the images without any other guidelines but ground truth, and reach the *state of the art* performance. Recently some works like [11] noticed that the semantic information makes huge differences in image colorization, and it helps to constrain the chrominance region, which means there is less vague artifact existing in the object or stuff boundary.

In this paper, I propose a colorization neural network combined with semantic segmentation. The proposed model contains two parts: semantic segmentation and colorization. For the semantic segmentation part, we finetune the Deeplab v3+ [19] with parameter-transfer learning. And the subnet of colorization adopts an off-the-shelf model from [4]. The source codes are referred from these two official projects in Github and are fused individually. The pipeline of our model is as shown in Fig1. The rest contents of this paper are composed as follows. In Section2 we briefly introduce the related works on colorization and semantic segmentation. Then the Section3 provides an overview of our proposed model and explain the details in Section4. Experiments will be discussed in Section5. And we summary our works in Section5.

Our contributions are as follows:

- A fully data-driven based method to automatically generate the colorful image.
- A proposed network of innovation with implementing semantic segmentation to leverage and guide the colorization.

2. Related Work

Scribble-based colorization

Initially the colorization highly depends on the user guidelines. The proposed methods formulate the colorization as an optimization problem that the user interactions and scribbles based on chrominance similarity. For instance, Luan *et al.* [1] assume that the chrominance has similarity intensity in the neighboring pixels then gives weights to multiply global region so that it enables to minimize the specific loss and iterate the weight in similarity. Several follow-up approaches reduce color bleeding via edge detection in [3]. These methods can generate convincing results with detailed and careful guidance hints provided by the users.

Example-based colorization

The reference images allow the model to extract the color clues adaptively and reduce the intensive and expensive user interactions. These methods [5] [6] [12] [13] just need to provide a colorful image whose content or chrominance is related to the grayscale image, then the color information will be transplanted to the target. Deshpande *et al.* [5] utilize Encoder-Decoder architecture and GMM (Gaussian Mixture Model). The Encoder implements VAE (Variational Autoencoder[14]) to sample diverse colors and allocates the embedding produced by GMM. Xu et al. [13] adopt the Encoder-Decoder

structure as well and use AdaIN [15] to better generate chrominance in the gray image with reference photo, then follow a fusion sub-net to integrate.

Data-Driven colorization

The learning-based colorization normally generate in the pattern of end-to-end. There are a variety of models based on CNN, which helps to extract the features. Iizuka et al. [10] build up a plain network with two branches: one is responsible for object classification and another for spatial feature extraction. The colorization is a classification task using the Cross-Entropy loss function. Zhang [9] *et al.* improved the training and proposed that colorization is not a hard target problem. His group is based on the thought of soft-target [16] and fuses the top-5 predicted results.

Semantic segmentation

Semantic segmentation is one of the four basic tasks in computer vision. Recently there are a variety of methods [17][18] [19] with proposed new theories to achieve the *state of the art* performance. [17] proposed a pyramid structure when upsampling the feature maps. The pyramid structure is beneficial for the network to connect different level features. Chen *et al.* [18][19]develop a series version of models called DeepLab, and in the final version v3+ it is combined all of the advantages of previous model like using dilated convolution, adopting pyramid structure and etc.

3. Overview

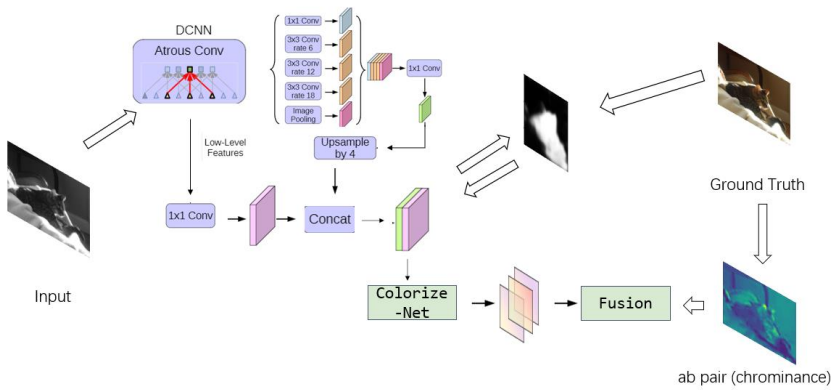

Fig1. **Model overview**. Two parts: one is semantic segmentation and another is colorization net

Given a grayscale image $X \in R^{H\times W\times 1}$ as input and the task of the model is to predict the missing chrominance $Y \in R^{H\times W\times 2}$, where Y is the value of *ab* channel in CIE Lab color space and H, W denote the height and width of the image respectively. Unlike directly transfer the X into I_{pred} which has RGB -- three channels to present color information, we adopt this transformation to avoid bigger predict error because we only need to output two channels. The luminance channel is as same as X and then concatenate the output to form final result. The formulation is illustrated in Fig2.

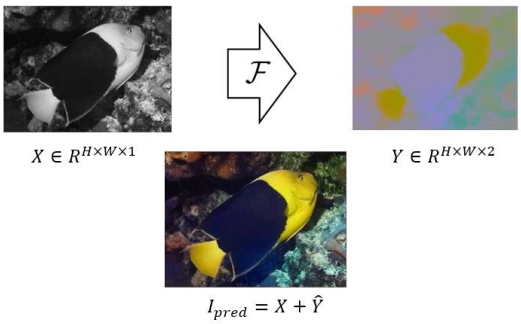

Fig2. Demonstration of the formulation of result

The *F* in Fig2., is our proposed model in Fig1. The part of semantic segmentation adopts DeepLab v3+ [19] as backbone. The high level and low level features are extracted by the depthwise convolutional backbone, and then send to the ASPP. The ASPP is organized by atrous convolution in the pyramid structure mentioned before, and the details will be specified in Section4. Next the output of ASPP is concatenated with the low level features to form the input of decoder. With three times of convolution in decoder to fuse different features, we upsample and obtain the semantic segmentation result. Then we freeze the weight to employ the colorization subnet. The input of the colorization part is the segmentation result $E \in R^{\frac{H}{4}\times\frac{W}{4}\times classes}$, where *classes* denotes the number of objects and stuffs in dataset. The colorization network utilizes the similar structure of U-Net [20] and refer the improved version in [4]. After that, we fuse the chrominance result produced by colorization subnet and luminance channel to obtain final result.

4. Method

The model is composed of two backbones: the semantic segmentation and colorization. In this section, we display more details and theories about our proposed method.

4.1 Semantic segmentation

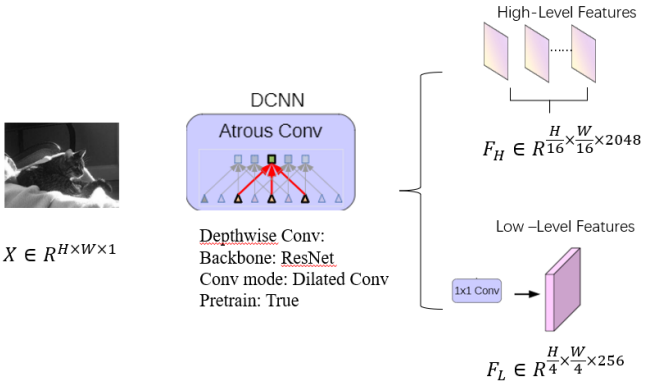

Fig3. The backbone of our semantic segmentation

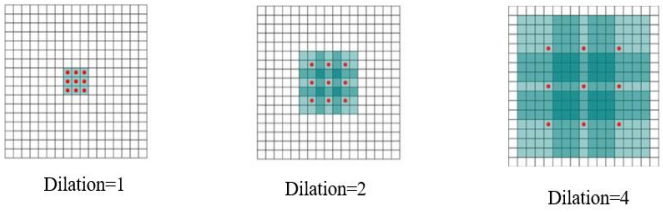

Fig4. Dilated convolutional kernel: the red points are trainable and the value in green is 0

The semantic segmentation part borrows the off-the-shelf model—DeepLab v3+ [19] . The Depthwise CNN implements ResNet 101 [21] as backbone with changed mode in computation of convolution. The ResNet allows network extract features in a very deep layers but suffer little from gradient vanishing problem. The atrous (or called dilated) convolution applied in the backbone enlarge the ability of ResNet to capture spatial features. The atrous convolution proposed by Yu *et al.* [22], is proved that it reduces the computation complexity while is enable to obtain receptive field.

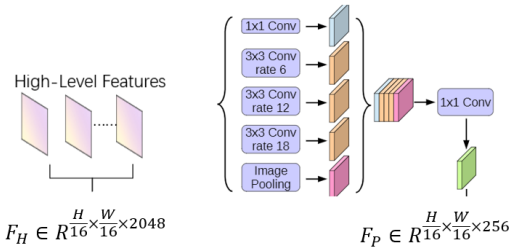

Fig5. The structure of ASPP in pyramid pooling

The ASPP pooling adopt different scales of atrous convolution to extract features of different sizes and dimensions, i.e., the smaller the ratio the smaller the object is perceived (like tiny box), and vice versa the larger the spatial features are extracted more abstract (the main spatial feature of the object).

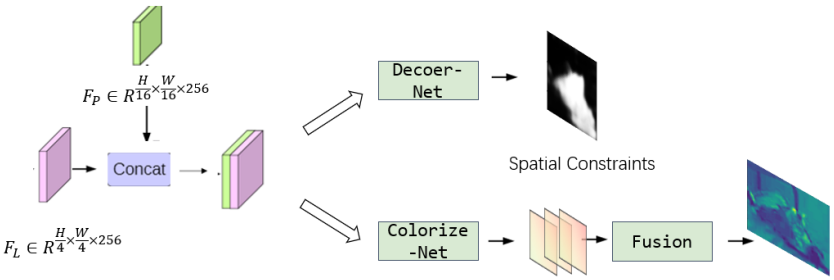

Fig6. The connection of two subnets

Then the output of ASPP F_P is concatenated with the low feature F_P. There are two branches: one is for final predication of segmentation and another is colorization. Firstly, the concatenated features are sent to the decoder which is organized with several convolution layers to classify the label of each pixel. After this part finished training, we freeze the weight to start another part.

4.2 Colorization backbone

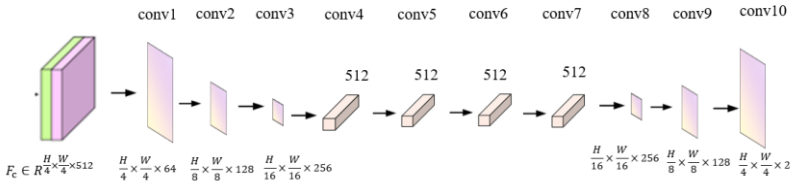

Fig7. The subnet of colorization

The colorization is illustrated as Fig7 shown, and the main structure is similar to U-Net [20] . For Zhang *et al.* [4] also deploy this structure in their *state of the art* model and provide the pre-trained weight, we adaptively match the weight to our model. The structure of colorization subnet is a plain and symmetry CNN which is also powerful ability to downsample and upsample.

4.2 Details in loss function

There are two different loss function applied in our model. For the semantic segmentation, the network needs to predict the label in pixel level. It is a typical classification problem and the ideal loss function is Cross Entropy as the equation bellow.

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^\top, \quad l_n = -w_{y_n} \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})}$$

where x is the input, y is the target, w is the weight, C is the number of classes, and N spans the minibatch dimension as well as d_1 …… d_k for the K-dimensional case.

For colorize-net, following the loss function mentioned in [4] we use the smooth l_1 loss with $\delta = 1$ as the following equation.

$$\ell_\delta(x, y) = \tfrac{1}{2}(x - y)^2 \mathbb{1}_{\{|x-y|<\delta\}} + \delta(|x - y| - \tfrac{1}{2}\delta) \mathbb{1}_{\{|x-y|\geq\delta\}}$$

The smooth l_1 help to convergence quicker and is not sensitive to outliers. In this condition, the chrominance predication will be regard as a regression problem to avoid colorizing the image in rigid and lack of diversity compare to colorization in a classified way.

5. Experiments

5.1 Preparation

Dataset

We use COCO-Stuff [8] for training and validation. This dataset provides a variety of colorful images, which contain natural scene and stuff with multiple objects. Compared with the previous version, it has more specific labels to denote the images. Currently version includes 118k images for training and 5k images for testing. Though we our model refers the off-the-shelf which provides the pre-trained weight, but we still have to finetune due the difference of input, fusion part and etc. Such kind of huge dataset is expensive to train, so we take part of the dataset (about 10k for training and 2k for testing from the corresponding dataset) to accelerate train. Specifically, the inputs are

processed with following equation which is a common transformation from colorful image to grayscale.

$$Gray = R*0.299 + G*0.587 + B*0.114$$

And the ground truth of the network is divided in two sections: the segmentation label for the training of semantic segmentation and the RGB image for final colorization assessment. Fig4 display the two kinds of ground truth.

Fig8. Demo of COCO-Stuff dataset from official website [8]

Support

Our model is support by *JiuTian* (https://jtedu.cmri.cn/), which is an advanced cloud service platform providing GPU for AI training. All of the training utilizes NVIDIA GPU V100, only one graph card but powerful. The server also provides 200G to storage relevant data in Linux system.

Training Details

There are different strategies for training the model. As mentioned before, the proposed network is composed of two main subnets: semantic segmentation and colorization. For the semantic segmentation, we initialize the model with the pre-trained weight provided by [19] and published in the Github. The only difference is that our input is grayscale while in DeepLab v3+ is RGB image. To finetune the model we deploy the same hyper parameters proposed in [19] except the number of epochs. The initial setting of learning rate is equal to 1e-5 and with *POLY* scheduler and momentum 0.9 to adjust it when training. The loss will quickly convergence benefited from the pre-trained model so we only train this part with 15 epochs. In the part of colorization, because the official model in [4] do not share the pre-train model on COCO-Stuff but ImageNet, we train this subnet for 30 epochs with 5e-5 learning rate. The complete model adopts same optimizer using SGD.

5.2 Demo of result

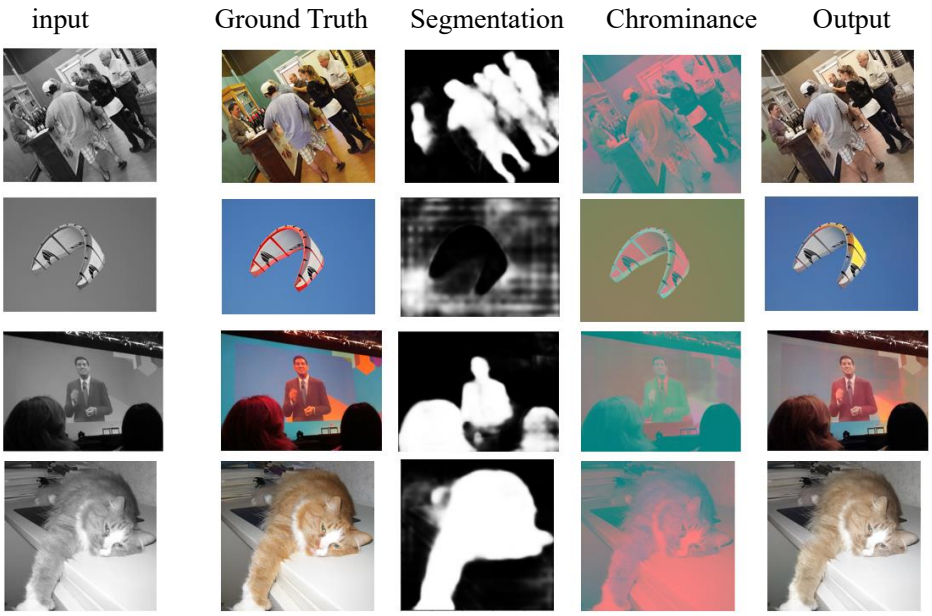

Fig9. Demo of our result: from left to right it displays the input (grayscale image), ground truth (original RGB image), semantic segmentation, chrominance and our results respectively

In this section, we display some of our successful colorized images. The results of semantic segmentation are taken from the label of main content of the image. E.g., in the last row the cat is the object, which corresponded to the label *"37"* in COCO-Stuff.

Because the segmentation backbone output $O \in R^{\frac{H}{4}\times\frac{W}{4}\times classes}$, where classes are 183 in the dataset. We only extract the 37th index of channels to display the result of *cat*. The other labels from the first to the third row are *people(0), sky(155) and people* respectively. And in order to better show the chrominance, we merge the predicted *ab* channels and luminance (input) in 4th column. The last column illustrates the result from Lab transfer to RGB. And Fig10 is a proof of our training.

```
        ## Info for epoch 15 ##
        val_loss       : 1.80574
        Pixel_Accuracy : 0.573
        Mean_IoU       : 0.23700000034570694
        Class_IoU      : {0: 0.737, 1: 0.507, 2: 0.521, 3: 0.484, 4: 0.481, 5: 0.825, 6: 0.429, 7: 0.537, 8: 0.314, 9: 0.238, 1
0: 0.851, 11: 0.0, 12: 0.766, 13: 0.301, 14: 0.01, 15: 0.41, 16: 0.761, 17: 0.48, 18: 0.632, 19: 0.784, 20: 0.336, 21: 0.768, 22:
0.837, 23: 0.86, 24: 0.779, 25: 0.0, 26: 0.245, 27: 0.384, 28: 0.0, 29: 0.0, 30: 0.059, 31: 0.0, 32: 0.498, 33: 0.298, 34: 0.0, 3
5: 0.01, 36: 0.0, 37: 0.588, 38: 0.0, 39: 0.0, 40: 0.167, 41: 0.274, 42: 0.383, 43: 0.143, 44: 0.0, 45: 0.162, 46: 0.293, 47: 0.
0, 48: 0.0, 49: 0.0, 50: 0.213, 51: 0.599, 52: 0.277, 53: 0.43, 54: 0.138, 55: 0.422, 56: 0.269, 57: 0.658, 58: 0.686, 59: 0.193,
60: 0.28, 61: 0.241, 62: 0.284, 63: 0.504, 64: 0.323, 65: 0.0, 66: 0.325, 67: 0.0, 68: 0.0, 69: 0.521, 70: 0.0, 71: 0.313, 72: 0.
403, 73: 0.007, 74: 0.037, 75: 0.424, 76: 0.316, 77: 0.0, 78: 0.221, 79: 0.0, 80: 0.535, 81: 0.392, 82: 0.0, 83: 0.102, 84: 0.42
4, 85: 0.08, 86: 0.0, 87: 0.777, 88: 0.0, 89: 0.0, 90: 0.0, 91: 0.064, 92: 0.0, 93: 0.098, 94: 0.47, 95: 0.417, 96: 0.151, 97: 0.
256, 98: 0.233, 99: 0.184, 100: 0.377, 101: 0.401, 102: 0.0, 103: 0.0, 104: 0.0, 105: 0.365, 106: 0.16, 107: 0.0, 108: 0.322, 10
9: 0.257, 110: 0.295, 111: 0.146, 112: 0.301, 113: 0.0, 114: 0.123, 115: 0.0, 116: 0.172, 117: 0.295, 118: 0.153, 119: 0.003, 12
0: 0.124, 121: 0.042, 122: 0.03, 123: 0.566, 124: 0.111, 125: 0.003, 126: 0.067, 127: 0.179, 128: 0.088, 129: 0.248, 130: 0.0, 13
1: 0.306, 132: 0.041, 133: 0.0, 134: 0.588, 135: 0.0, 136: 0.0, 137: 0.469, 138: 0.059, 139: 0.384, 140: 0.0, 141: 0.074, 142: 0.
027, 143: 0.0, 144: 0.452, 145: 0.0, 146: 0.281, 147: 0.246, 148: 0.438, 149: 0.258, 150: 0.072, 151: 0.183, 152: 0.0, 153: 0.43
6, 154: 0.826, 155: 0.125, 156: 0.554, 157: 0.0, 158: 0.7, 159: 0.0, 160: 0.0, 161: 0.0, 162: 0.029, 163: 0.0, 164: 0.033, 165:
0.0, 166: 0.012, 167: 0.0, 168: 0.664, 169: 0.335, 170: 0.065, 171: 0.35, 172: 0.102, 173: 0.0, 174: 0.0, 175: 0.262, 176: 0.017,
177: 0.097, 178: 0.0, 179: 0.476, 180: 0.172, 181: 0.0, 182: 0.0}
```

Fig10. A screen capture of our training

5.3 Analysis

For the assessment of our result, there is metrics for semantic segmentation but none of specific standard for colorization. In this case, we analysis our final result directly in the visual way.

For Semantic segmentation, as displayed in Fig10, the IOU (Intersection over Union) performs not very well compared to the proposed version in the paper [19]. Besides,

the accuracy of classification in the pixel is fair as well, even some of the accuracy is zero. The main reason is that we are training our model with limited data (10k, about one tenth of the COCO-Stuff dataset). In this condition, the segmentation result performs badly when processing limited images whose content is trained in our model, and other content is not included in our dataset. For instance, the second row in Fig9 demonstrates the details of the *image: a kite fly in the blue sky*. However, the sky was segmented bad because half of the sky region is black. That is to say, after using *softmax* to normalize and quantification, the possibility of the prediction in the sky region is very low. The class IOU (only 0.125) in Fig10 prove it as well.

With respect to the colorization, it shows powerful generalization capability. Still refer the same demo in Fig9, though the semantic segmentation is not quite nice what we would expect, the colorization subnet fixes this problem well. But noting that other chrominance in Fig9, we can obviously observe that the predicted result is lack of diversity compare to the ground truth. The ground truths possess more vibrant and bright colors while the colors of outputs are monotonous, e.g., the hair of the cat is darker than the predict one.

More specifically, we explore the function of the semantic segmentation. The demo in Fig11 illustrate the difference between the output results. We use a simple plain CNN to relace the semantic subnet and send to the same colorization network. We can observe that the chrominance in the second row is discrete and the proposed model output the color in a centralized region. Meanwhile the boundaries between different objects are clearer in the last row. Visually the quality of colorization in the last row is better than the second row.

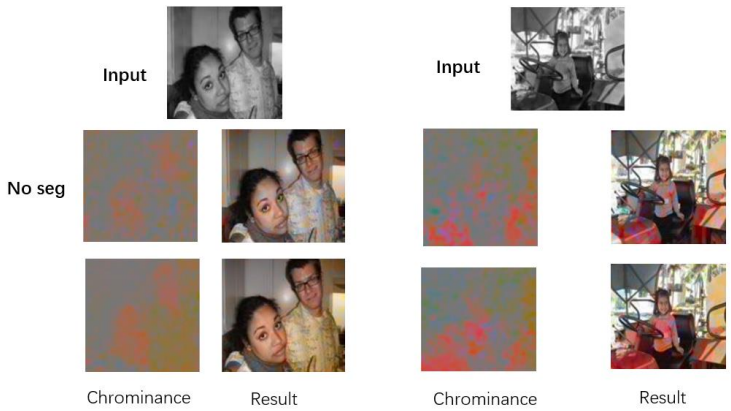

Fig11. Comparison: the second row without semantic segmentation and the last row is our proposed result

5.4 Some failures

Fig12. Some failures in colorization

Here we display some of our failed examples. As shown in Fig12, we can see that only part of the contents in image were colorized. For instance, both of the trees in the first and the third row of image were colored well in green, but other objects like the cat, traffic sign etc. almost changed little or have not changed. We conceive that the leading reason still relies on the dataset for we only trained few of dataset. Some objects cannot be semantically separated effectively in some scenarios, and the generalization capability of colorization network is insufficient as well.

6. Conclusion

In this paper, we proposed an architecture for colorization based on semantic segmentation. The semantic segmentation achieves the state of the art performance in pixel level classification with implementing DeepLab v3+. The off-the-shelf colorization also produce plausible color region in *ab* channels. Based on the semantic information, the evident experimental result illustrate that the chrominance distribution is guided by it. The analysis of the semantic segmentation prove that it is beneficial to constrain the chrominance region. But our model still exists some problems as demonstrated in the failed examples, our model still suffers from the drawbacks like lack of diversity of color, perform well in limited classes, weak Robustness and so on. One of the main reasons is that our training processing have to improve since we only utilize 1 / 10 data of COCO-Stuff dataset.

Now days, there are other advanced paradigms proposed like Transformer [23], *Pix to Pix* [24]. With the development of deep learning, hope that more advanced frame will be deploy in the colorization area. In the future, we conceive that the research of colorization will bring the existence of specific application and actual benefits.

Supplement

Notice: all of the project adopts open-source code and make few revisions to conform our topic. Our proposed thoughts and methods are individual, as for more details about the code please refer to the public published projects in Github mentioned in [4] [19] .
(Number of Words except References: 3218)

References

[1] Huang, Y. C., Tung, Y. S., Chen, J. C., Wang, S. W., & Wu, J. L. (2005, November). An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia* (pp. 351-354).

[2] Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y. Q., & Shum, H. Y. (2007, June). Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques* (pp. 309-320).

[3] Yatziv, L., & Sapiro, G. (2006). Fast image and video colorization using chrominance blending. *IEEE transactions on image processing*, *15*(5), 1120-1129.

[4] Zhang, R. Y., Zhu, J. Y., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017). Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics*, *36*(4), 119.

[5] Deshpande, A., Lu, J., Yeh, M. C., Jin Chong, M., & Forsyth, D. (2017). Learning diverse image colorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6837-6845).

[6] Li, H., Sheng, B., Li, P., Ali, R., & Chen, C. P. (2021). Globally and Locally Semantic Colorization via Exemplar-Based Broad-GAN. *IEEE Transactions on Image Processing*.

[7] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), 211-252.

[8] Caesar, H., Uijlings, J., & Ferrari, V. (2018). Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1209-1218).

[9] Zhang, R., Isola, P., & Efros, A. A. (2016, October). Colorful image colorization. In *European conference on computer vision* (pp. 649-666). Springer, Cham.

[10] Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2016). Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, *35*(4), 1-11.

[11] Zhao, J., Han, J., Shao, L., & Snoek, C. G. (2020). Pixelated semantic colorization. *International Journal of Computer Vision*, *128*(4), 818-834.

[12] He, M., Chen, D., Liao, J., Sander, P. V., & Yuan, L. (2018). Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)*, *37*(4), 1-16.

[13] Xu, Z., Wang, T., Fang, F., Sheng, Y., & Zhang, G. (2020). Stylization-based architecture for fast deep exemplar colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9363-9372).

[14] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[15] Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

[16] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[17] Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2881-2890).

[18] Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

[19] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV) (pp. 801-818).

[20] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[21] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[22] Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

[23] Kumar, M., Weissenborn, D., & Kalchbrenner, N. (2021). Colorization transformer. arXiv preprint arXiv:2102.04432.

[24] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).