Towards End-to-End Joint Image Colorization and Super-Resolution with Neural Networks: Report

Berat Biçer
Computer Engineering
Bilkent University, Ankara, Turkey
berat.bicer@bilkent.edu.tr

Ergün Batuhan Kaynak
Computer Engineering
Bilkent University, Ankara, Turkey
batuhan.kaynak@bilkent.edu.tr

I. Introduction

Contemporary research focuses on image colorization and super-resolution as two separate tasks. In this project, we to train a single neural network to jointly solve both of these problems in an end-to-end fashion. We define the notation x as input to the end-to-end network N where x is any grayscale, low-resolution image. The network N learns a function $f : x, x_{ref} \rightarrow x'$ where x_{ref} is the reference color image and x' is the network output which is a higher-resolution, colored image. Traditionally, this can simply expressed in a pipeline architecture of networks as $f(x, x_{ref}) = f_r(f_c(x, x_{ref}))$ where functions f_c, f_r are learned mappings such that $f_c : x, x_{ref} \rightarrow x_c$ and $f_r : x_c \rightarrow x_r$. Here, x_c is the output low-resolution color image and x_r is the output high-resolution color image, or vice versa. In this setting, functions f_r, f_c are individually learned by unique networks, hence doing so separately takes a considerable amount of time and is quite expensive in terms of implementation and computational resources. This process also ignores possible interdependence between these two tasks, which has not been explored before in the literature. To this end, we make the following contributions:

1) To our knowledge, the only precursor study in this context [1] attempted to reconstruct high resolution color images with cascaded subnetworks. We attempt to solve this problem with a single set of features and obtain desired results with a single network, trained end-to-end fashion.
2) We designed and trained a neural network which jointly learns super-resolution and colorization. We also share implementation details, qualitative and quantitative results of the network, comparison with the state-of-the-art disjoint super resolution and colorization networks, and related discussions.

II. Building the Joint Network

Since building a joint image colorization (IC) and super resolution (SR) network was not attempted before, we wanted to build these networks separately and observe how the respective changes to the architectures change their results.

To this aim, we first implemented an only IC network similar to Iizuka et. al. [2] and we see that:

- CIE LAB color space is advantageous: a* and b* channels produce colors while L* channel contains shapes/boundaries.
- L* channel of reference image is fed into the network, output is learned a* and b* channels.
- Final output is obtained by merging L* of reference image and learned a* & b* are merged into a single image.
- Pixelwise MSE is a good loss function.
- Iizuka et al used 2.3M train images; training this network takes a long time.

Next, we implement our SR network, using the structure of SRResNet [3] and see that:

- Residual connections are very important, but memory consuming.
- If we build deeper networks with residual blocks, our results got better and better (although this is mostly true for most deep learning applications, it was more true for SR than it was for IC).
- Using small resolutions for low resolution input causes the network to converge fast with bad results, mostly since the input image does not contain a lot of information to begin with.

Both standalone IC and SR network benefited from using more data and running for more epochs.

In our project proposal, we had proposed that we would be using generative adverserial networks (GANs). To test the usability of GANs, we implement SRGAN [] for image super resolution task. Although we have spend quite a time trying to make GANs work, they proved to be infeasible. We believe the main reason behind this to be the amount of data we use. As we increased our data, the point at which the mode collapse occurs would be delayed and our loss would get slightly better. Since training of GANs took a lot, it was very hard to pinpoint problems and make observations. In the end, we opt for not using GANs because our already hard GAN training would go worse when we configured our network for joint training.

III. Implementation Details

A. Dataset

MSCOCO [4] is a multi-purpose dataset with 118K train images. We employed full training set for training where each

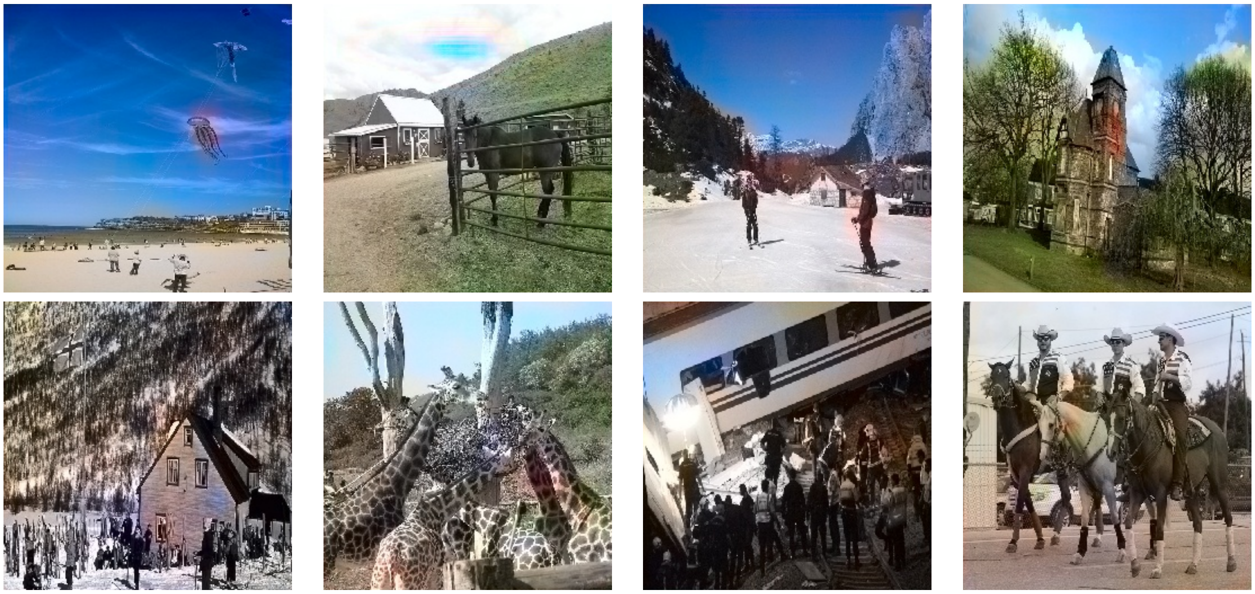

Fig. 1: Good(top) and bad(bottom) results obtained with the process described in Section III-C.

sample is transformed as follows: a sample from training set is downscaled to 256x256 and converted to CIE LAB colorspace, named reference image. L* channel of each reference image is extracted and further downscaled to 96x96, which are fed into the network as input. A* and b* channels of the reference images are also extracted and used as reference for training.

B. Architecture

Network is designed in encoder-decoder fashion with additional upsampling layers in the decoder to increase spatial dimension. A summary of the network is given in Table I. and II. Each convolution in encoder is followed by batch normalization and leaky ReLU activation. Each convolution in decoder network is followed by batch normalization, leaky ReLU activation, and nearest-neighbour interpolation upsampling respectively. Decoder network has two more layers than encoder networks, which allows outputs with larger spatial dimension than inputs, of shape $2 \times 256 \times 256$.

TABLE I: Details of encoder network. Each layer represents a convolution operation executed with specified parameters.

	Filters	Kernel Size	Stride	Padding
Conv1	64	3	2	1
Conv2	128	3	2	1
Conv3	128	3	1	1
Conv4	256	3	2	1
Conv5	256	3	1	1
Conv6	512	3	2	1

C. Training

Network described in Section III-B is trained on a Tesla K80 for 10+ days on remote settings. Network is trained using element-wise mean-squared loss (MSE) for 148 epochs with batch size of 160, Adam optimizer with learning rate 0.0001 and regulation on loss plateau with a factor of 0.75.

TABLE II: Details of decoder network. Each layer represents a convolution operation as in Table I.

	Filters	Kernel Size	Stride	Padding
Conv1	512	3	1	1
Conv2	256	3	1	1
Conv3	128	3	1	1
Conv4	64	3	1	1
Conv5	32	3	1	1
Conv6	16	3	1	1
Conv7	8	3	1	1
Conv8	2	3	1	1

IV. Additional Joint Network Attempts

At this point, we have managed to train a joint network and obtain quantitative and qualitative results. We now construct some more networks to improve our results, using more of the information we got from our observations. Here, we list the challenges that we need to address:

- We need a network with high number of layers, and these layers should consist of residuals blocks (connections).
- Large feature map sizes (e.g. 512) for intermediate representation of colorization.
- Since we input R and output $2R$, end-to-end architecture is ill-defined for non-examplar based method, we need to obtain the grayscale L* channel from the ground truth (or interpolate the L* channel of the input but that beats the point of having a joint network in the first place).
- Need to input relatively high R to an already large network to learn image super resolution.

V. Results

Network output is learned a* and b* channels of the reference image which is of shape $2 \times 256 \times 256$. Final results are obtained in CIE LAB colorspace by merging L* channel of the reference image with network output. In Figure 1 and 2, we share qualitative results from the network. We make the following observations: 'Good' results are simplistic: images containing clouds, sea, grass where background is mostly uniform and object segmentation is easier wheras complex scenes - those containing composite object boundaries, many distinct colors, humans, etc. cannot be learned well. During training, we also realized validation loss stopped improving after the first few epochs. In light of these observations and considering some studies [2] employed training sets significantly larger than ours, we believe the network overfits; which requires extensive investigation in the future.

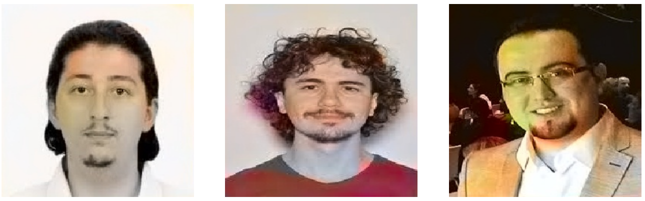

Fig. 2: Qualitative results for human portraits.

Quantitative results are given in Table III. Note that since we do not have access to dataset on other results are calculated, our PSNR value is computed on MSCOCO validation set.

TABLE III: Reported PSNR values for colorization and super resolution tasks.

Method	PSNR
Ours	30.04
Global + global hist [5]	27.85 ± 0.13
Global + global sat [5]	25.78 ± 0.15
Local + gt colors [5]	37.70 ± 0.14
SRGAN [6] (DIV2K)	28.92
ESRGAN [3] (Mangal109)	33.66

References

[1] L. Zhao, H. Bai, J. Liang, B. Zeng, A. Wang, and Y. Zhao, "Simultaneous color-depth super-resolution with conditional generative adversarial networks," *Pattern Recognition*, vol. 88, pp. 356–369, 2019.

[2] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "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)*, vol. 35, no. 4, pp. 1–11, 2016.

[3] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[5] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *arXiv preprint arXiv:1705.02999*, 2017.

[6] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.