Automatic Image Colorization using Generative Adversarial Networks

Project Paper of team: Yet Another Layer [YAL]
CSci 5561 - Computer Vision

Cameron Fabbri, Md Jahidul Islam

Abstract

Abstract need to be added Abstract need to be added

1 Introduction

Image colorization [1, 2, 3] refers to colorizing a given gray-scale image so that it appears real. A large amount of photographs, videos and movies, mainly antique, lack color; image colorization can provide a modern and vivid view to these images. In addition, surveillance cameras often capture (or store) gray-scale images for convenience. Several underwater inspection and surveillance applications [4, 5] often have to deal with color-less images due to lack of visible light in deep-water. Robust and efficient image colorization techniques can be used in these applications with substantial benefits.

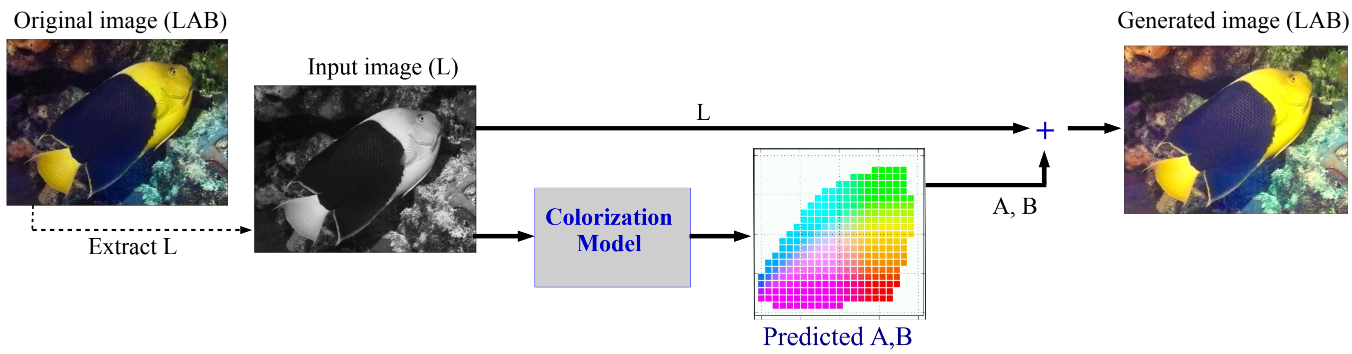


Figure 1: Basic image colorization procedure is shown. LAB color-space is generally used for convenience (*i.e.*, one less unknown dimension); given the lightness channel L, task for the colorization model is to predict A and B channels so that the colorized image appears natural.

Colorizing a gray-scale image (*i.e.*, only intensity values are known) is a difficult and ill-posed problem. Computer vision community have approached this problem in different ways over the last few decades [1, 2, 3, 6, 7, 8]. Before the advent of deep-learning [9], researchers have tried many classical techniques [6, 7, 8, 10, 11] to capture relationships between color components (RGB or LAB) and image level features. Due to multi-modality and ill-posed nature of the problem, optimization based techniques [10, 6] and probabilistic models [11] were the only ones that achieved decent colorization performance in few specific applications. However, overall performance of these techniques, in general, were still poor due to the high non-linearity and abstract nature of color-feature relationship.

Recently, deep-learning based image colorization techniques [1, 2, 12, 13], trained over millions of images, have shown significantly better performance over the earlier classical methods. For instance, the current state-of-the-art, 'colorful colorization' [1], can fool a human observer 32% of the time in a *colorization Turing-test* scenario. **Additionally, how use of GANs are likely to improve the performance which why we are going to try?**

In this project, we *// need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added.*

2 Background and Related Work

As mentioned in the previous Section, image colorization is an ill-posed problem due to multi-modality and ambiguity. While some natural objects commonly hold the same color (e.g grass is *usually* green), many are left up for interpretation. For example, given a gray-scale image of someone wearing a dark colored shirt, there is no way of figuring out the true color. Instead, the objective is to come up with a colorization that appears real, *i.e.*, natural.

User-based approaches [10, 8, 14, 15] were popular for being fast and relatively accurate as user can provide a good prior for the inherent color distribution. However, these methods are not applicable for large scale automatic colorization, which led researchers to adopt optimization and probabilistic approaches [6, 3, 11]. These approaches model a likelihood based color approximation for each pixel given the neighborhood information. Few methods introduce additional step for spatial coherency through image based segmentation as well. However, overall colorization performance of these approaches are not very appealing [16] for general usage in a large scale. This is because the prior distribution of color-space is domain-dependant; for instance, face images, underwater images, outdoor and satellite images, all have different color distributions. Besides, it is difficult to capture the highly non-linear and abstract color-feature relationships without large-scale training.

In recent times, deep-learning based approaches [1, 2, 12, 13] have produced significantly better colorization performance as they can extract highly non-linear spatial relationships if trained over large datasets. The convolutional layers learn appropriate filters to produce good feature-space representations from raw images. These feature extraction and filtering is performed over multiple layers to capture complex spatial relationships within the image-space, which is useful for image-to-image translation tasks. **Additionally, Prospects of GANs GANs ....**

In this project, we *// need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added. need to be added.*

3 Generative Approaches

In our project, we tried two classical methods: *colorization using optimization* [10], *colorization via multimodal predictions* [6]. The former is not an automatic approach (user provides color distribution prior), whereas, the later uses a set of reference images to formulate color distribution. Their operations are discussed in our term report; this paper focuses on the third generative approach, *colorful image colorization* [1], which is adopted in our project.

3.1 Colorful Image Colorization

Colorful image colorization [1] is considered as a major breakthrough on this problem. Published on 2016, it has managed to set a new benchmark for performance. It is an automatic approach that produces realistic colorizations based on a CNN-based model. It poses the problem as a multi-modal classification problem. The objective function is carefully designed to map the image-to-image translation problem to a classification problem. First, it takes advantage of the fact that A, B color components of LAB colorspace for natural images are concentrated in a small region, which can be discretized into finite number of bins (Q), as illustrated in the top row of Fig 2. Given the lightness channel (L_p) of a pixel p, its A, B pair corresponds to a particular bin (out of 313 bins in total), which is mapped to a 1-hot vector (Z_p). Consequently, task of the classification model, is to predict which bin each pixel corresponds to. That is, the output is a 313-mode probability distribution ($\hat{Z_p}$) for each pixel p. The objective function is modelled as a cross entropy loss between Z and $\hat{Z}$, expressed as follows:

$$L_{col}(Z, \hat{Z}) = -\sum_{p} Z_p \sum_{q \in Q} Z_p[q] log(\hat{Z}_p[q])$$

This cross-entropy loss can be further augmented with class rebalancing, to encourage rare colors. The detailed model specification, as shown in the bottom-row of Fig. 2, is an 8-layer CNN architecture where each conv layer refers to a block of 2 or 3 repeated conv and ReLU layers [17], followed by a BatchNorm layer [18]. The network has no pool layers; all changes in resolution are achieved through spatial down-sampling or up-sampling between conv blocks.

While working on designing a generator for our GAN-based model, we investigated this model with different objective functions (L_1 loss, L_2 loss, least-squared loss, etc.) instead of their cross-entropy based loss function. This is due to the fact that in a GAN-based model, *discriminator* expects an image from the *generator*, and tries to discriminate it as real or fake in order to force the generator to get better. Therefore, rather than adopting their classification model directly, I implemented their architecture using objective functions based on L_1, L_2, least-squared loss (so that it outputs an image, not classification probabilities). Additionally, it made our model end-to-end trainable, which can be easily incorporated in a GAN-based architecture.

Given an input image I, the network is fed with its L channel (I_L); the output layer of the network is adjusted to predict $\hat{I}_{AB}$. We have found L_1 and L_2 loss functions perform quite well with this model. These loss functions between true I_{AB} and $\hat{I}_{AB}$ can be expressed as follows:

$$L_1(I_{AB}, \hat{I}_{AB}) = \lambda \sum_{p} |I_{AB}[p] - \hat{I}_{AB}[p]|$$

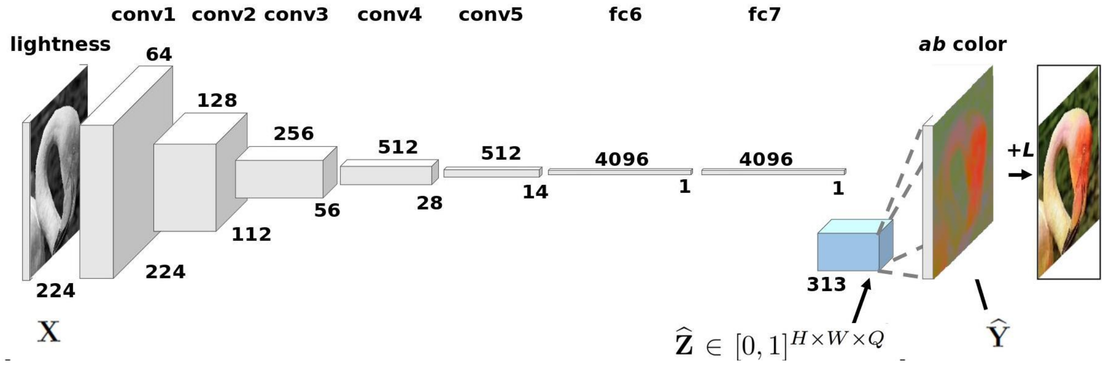


Figure 2: Architecture of colorful image colorization model [1] that is adopted in our evaluation

$$L_2(I_{AB}, \hat{I}_{AB}) = \lambda \sum_p (I_{AB}[p] - \hat{I}_{AB}[p])^2$$

Here, λ is a normalization constant. CelebA datasets [19] is used for training primarily, that has $195K$ training examples (a total of $202K$ images); In a separate training, a set of $200K$ images from Places2 dataset [20] were also used for training[1]. Training was performed using two 1080 gpus (in a core-i7 machine having 64GB RAM); training time for $100K$ iterations with a batch size 32 was about 2-3 days for each trial. The implementation is done using tensoflow [22] in python.

We found that this model performs better with L_1 loss function compared to L_2 loss. Additionally, training is much smoother for L_1 loss as well, as evident from Fig **??**. This might be because of the averaging effect of 2-norm which causes a blurry colorization (similar phenomena is discussed in [1]). The results on few test cases are shown in Fig. 3.

It is to be noted that these results were found while designing it as a generator only, we will discuss results for our GAN-based models in the project report. Currently, the loss function (L_{col}) for classification model of the original paper is being tried out, to investigate how it performs over these datasets.

4 Adversarial Approaches

5 Conclusion

Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added Abstract need to be added

References

[1] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[2] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[3] Aurelie Bugeau, Vinh-Thong Ta, and Nicolas Papadakis. Variational exemplar-based image colorization. *IEEE Transactions on Image Processing*, 23(1):298–307, 2014.

[4] Huimin Lu, Yujie Li, and Seiichi Serikawa. Underwater image enhancement using guided trigonometric bilateral filter and fast automatic color correction. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 3412–3416. IEEE, 2013.

[5] Luz A Torres-Méndez and Gregory Dudek. Color correction of underwater images for aquatic robot inspection. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 60–73. Springer, 2005.

[6] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. *Computer Vision–ECCV 2008*, pages 126–139, 2008.

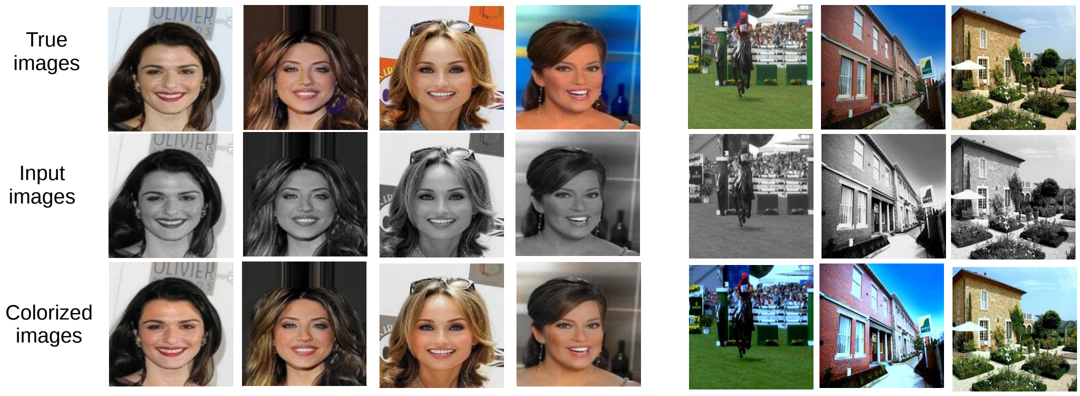
[7] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320. Eurographics Association, 2007.

[8] Vadim Konushin and Vladimir Vezhnevets. Interactive image colorization and recoloring based on coupled map lattices. In *Graphicon2006 conference proceedings, Novosibirsk Akademgorodok, Russia*, pages 231–234, 2006.

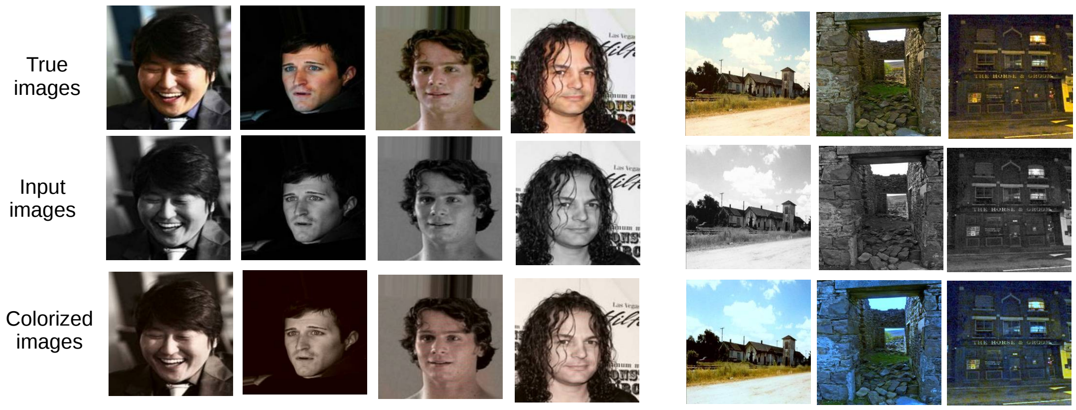
[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[10] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 689–694. ACM, 2004.

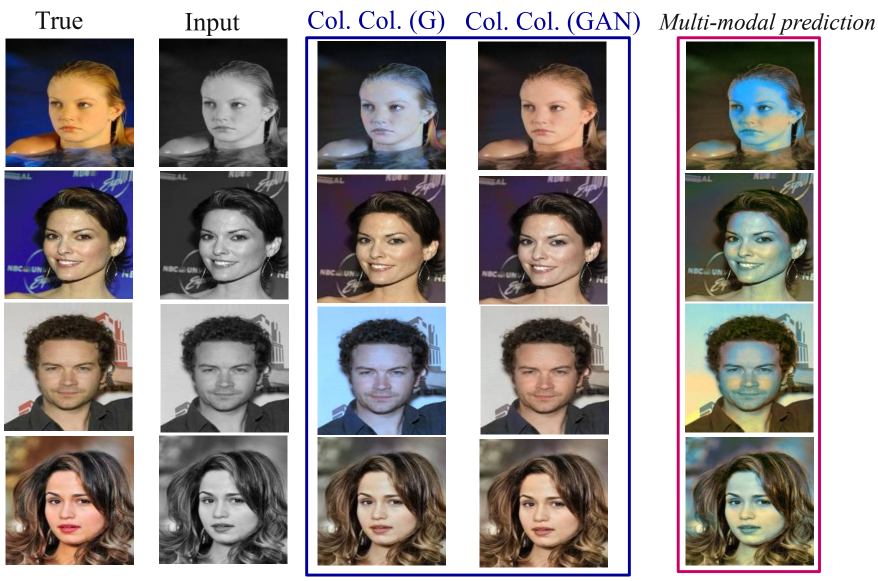
[1] Larger datasets like ImageNet [21] and full Places2 challenge dataset were not used for the project due to time constraint; however, we are planning to train our final GAN-based model over these datasets during the summer.



(a) Colorful Colorization [1] model with L1 loss



(b) Colorful Colorization [1] model with L2 loss



(c) Results in comparison with different models

Figure 3: Results for the colorful colorization model [1] with (a) L1 and (b) L2 loss is shown; First 4 columns show few examples from the test set of CelebA [19] dataset, while the rest (3) columns correspond to Places2 [20] dataset. Comparison for colorful colorization model used as a generator only (Col. Col. (G)), and as a generator in a GAN (Col. Col. (GAN)), is shown in (c); also, results obtained by colorization via multi-modal prediction [6] is provided in the last column to demonstrate the degree of improvements using colorful colorization model.

[11] Przemyslaw Lagodzinski and Bogdan Smolka. Digital image colorization based on probabilistic distance transformation. In *ELMAR, 2008. 50th International Symposium*, volume 2, pages 495–498. IEEE, 2008.

[12] Domonkos VARGA and Tamás Szirányi. Fully automatic image colorization based on convolutional neural network. *4th Winter School of PhD Students in Informatics and Mathematics*, page 36, 2016.

[13] Jie Li, Katherine A Skinner, Ryan M Eustice, and Matthew Johnson-Roberson. Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images. *arXiv preprint arXiv:1702.07392*, 2017.

[14] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[15] Michael J Vrhel and HJ Trussell. Color correction using principal components. *Color Research & Application*, 17(5):328–338, 1992.

[16] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 567–575, 2015.

[17] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] Celeba dataset. `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`. Accessed: 04-20-2017.

[20] Places2 dataset. `http://places2.csail.mit.edu/download.html`. Accessed: 04-20-2017.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[22] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.