Lab 5-8

Diana-Elena Stancu

November 2023, Github

Contents

1 Abstract

Current state-of-the-art colorization models often demand substantial computational resources, limiting their viability in resource-constrained environments. This research addresses the demand for practical image colorization solutions in this kind of environments, proposing a new model designed for deployment on mobile and web platforms.

2 Introduction

The technique of adding colour to a black and white or grayscale image is known as image colorization. To create a full-color image, it entails transferring the grayscale image's intensity values to a colour system, like RGB, and then adding colour to the channels that are lacking. While there are many approaches to image colorization, the most of them entail image processing, such as texture

generation, machine learning, or image segmentation. One common strategy is to colourize photos using a deep learning-based technique like a convolutional neural network (CNN). Usually, this method entails utilising a sizable dataset of colour images to train a CNN, which is then used to forecast the grayscale image's missing colour channels.

3 Related work

It has been shown in recent years that GAN [1] can produce vibrant visuals from some random noise. The generation network and discriminative network are the two fundamental network components that make up GAN. There are still many unsolved questions with GAN, despite the fact that it solves some of the problems associated with generative models. Theoretically, it can be shown that when the generator gradually learns the true distribution, the discriminator will eventually lose its ability to distinguish between real and false samples if the model is optimised in the real sample space.

Colorful Image Colorization [10] approached the problem as a classification task and they also considered the uncertainty of this problem (i.e. a car can be red, but also blue or yellow), while others had a regression-specific approach. There are also various type of approaches regarding the input and automation level of it. Lee et al. [6] tackles the automatic colorization task of a sketch image given an already-colored reference image. Years ago, this activity required a great deal of human input and hardcoding; but, with the power of AI and deep learning, the entire process may now be completed end-to-end.

4 Methodology

Image-to-Image Translation with CAN [3] proposed a general solution to numerous image-to-image tasks in Deep Learning, including image colorization. In the first stage, the approached architecture will be reproduced and adapted to the problem needs. New improvements can be done for the U-Net [7] used as the generator of the GAN, and even the main architecture can be enhanced using Conditional Generative Adversial Network. The last goal is to combine it with a MobileNetV2 [8] and measure the trade-offs and compare with the performant models.

4.1 Datasets

The training datasets employed in this research comprise the Common Objects in Context COCO 2017 dataset, renowned for its extensive collection of diverse images. Additionally, a proprietary dataset sourced from Pik-Fix [9] is incorporated, characterized by its distinctive feature of being meticulously edited by professionals.

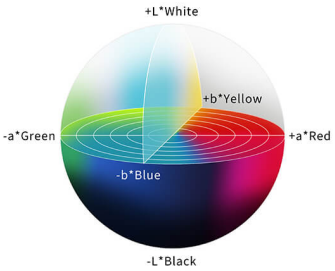

Figure 1: Enter Caption

4.2 Color space

The adoption of the Lab color space **??** in image colorization models is rooted in its distinct representation of color information, providing an advantageous framework for training. In the traditional RGB color space, each pixel is characterized by three values denoting the intensity of Red, Green, and Blue, posing a challenge for model training due to the inherent complexity arising from the multitude of possible combinations. In contrast, the Lab color space introduces a novel tripartite encoding. The L channel captures the Lightness, visually manifesting as a grayscale image, while the a and b channels delineate the green-red and yellow-blue components, respectively. Leveraging Lab facilitates an intuitive model training approach where the grayscale image (L channel) serves as the input, and the model is tasked with predicting the remaining two channels (a, b) for subsequent concatenation, yielding the full-color image. This strategy streamlines the colorization task by simplifying the prediction process, rendering it more tractable and less prone to instability compared to the RGB counterpart, where predicting three numbers introduces a substantially greater computational and training challenge.

4.3 Loss function optimization

In the optimization of the loss function within the context of our conditional Generative Adversarial Network (GAN) framework, the generator model takes a grayscale image (1-channel image, denoted as x) as input, generating a 2-channel image (y), comprising *a and *b color channels. The discriminator, in turn, assesses the authenticity of this generated image by concatenating the two produced channels with the original grayscale image and discerning whether the resulting 3-channel image is real or fake. Both the generator and discriminator are conditioned on this grayscale image. Mathematically, denoting the generator as G, the discriminator as D, and the input noise as z, the loss for the conditional GAN is expressed as:

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] \tag{1}$$

This GAN loss is augmented with an L1 Loss, representing the mean absolute

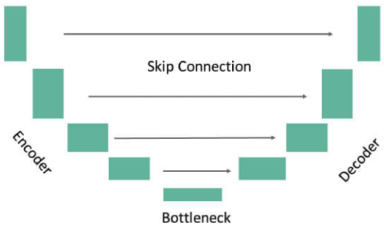

Figure 2: U-Net Architecture

error, measuring the disparity between the predicted colors and the actual colors. This supplementary loss function addresses the tendency of the model, when exclusively subjected to L1 loss, to exhibit conservatism in colorization, favoring neutral colours such as "gray" or "brown" when confronted with other colours. The L1 Loss is chosen over L2 loss to mitigate the propensity for generating grayish images. The combined loss function, incorporating both GAN and L1 losses, is formulated as:

$$\mathcal{L}_{\mathrm{L1}}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|] \tag{2}$$

$$\mathcal{L}_G = \arg\min_G \max_D \mathcal{L}_{\mathrm{cGAN}}(G,D) + \lambda \mathcal{L}_{\mathrm{L1}}(G) \tag{3}$$

where lambda serves as a coefficient regulating the relative contributions of the GAN and L1 losses to the overall loss. Notably, the discriminator loss does not incorporate the L1 loss, delineating the distinct roles of these components in the composite optimization framework.

4.4 GAN implementation

As proposed in the paper, the generator is implemented based on a U-Net architecture 7 [7]. This architecture derives its name from its U-shaped configuration, characterized by a contracting path on the left and an expansive path on the right, with a bottleneck-like structure in the middle. The contracting path involves the repeated application of down-sampling operations, typically convolutional layers. The necessity of employing a U-Net in the context of a GAN lies in its inherent ability to preserve fine details during the transformation process. The discriminator is based on stacked Convolutional-Batch Normalization-Leaky ReLU blocks.

4.5 Evaluation metrics

One fundamental approach to assess the performance of a Generative Adversarial Network (GAN) involves human scoring, where both real and generated images are randomly presented, and human evaluators assign labels distinguishing between real and fake images. For our method we will use two methods:

$$\mathrm{PSNR} = 10 \times \lg\left(\frac{255^2}{\mathrm{MSE}}\right)$$
$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[I(i,j) - I'(i,j)\right]^2$$

Figure 3: PSNR equation

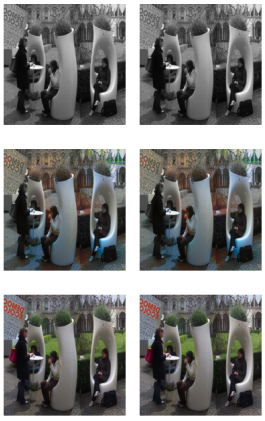

Figure 4: Example success 1 after 20 epochs

Peak Signal-to-Noise Ratio (PSNR) which reflects the distortion degree of the colorized image.

5 Results

For the first experiment, the model was tested on a subset of approximately 6000 photos from the extensive COCO dataset, which comprises over 1 million images. The GAN was trained for 20 epochs, revealing reasonable success in colorizing a significant portion of grayscale images. However, challenges surfaced, indicating limitations in accurately colorizing certain complex images (examples: 4 5).

After training on approximately 90 epochs the results got better with a PSNR value of 16.21. See 1.

6 Conclusion

The baseline colorization model, while demonstrating a rudimentary understanding of common objects in images, falls short in producing visually appealing results and struggles with rare objects, exhibiting issues such as color spillovers and undesired circular patterns. This limitation becomes particularly

Table 1: PSNR Values for Image Colorization Models on COCO dataset

Method	**PSNR**
Our method	16.21
Isola's et al. [4]	21.83
Zhang's et al. [11]	25.06
Lasson's et al. [5]	23.86
Lizuka's et al. [2]	23.69

Figure 5: Example fail 1 after 20 epchos

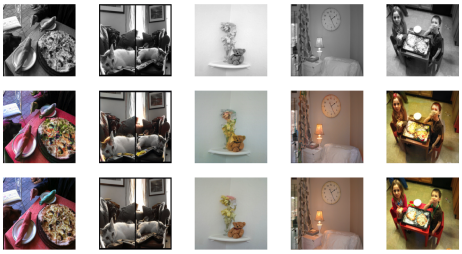

Figure 6: Example of colorization after aprox. 90 epochs

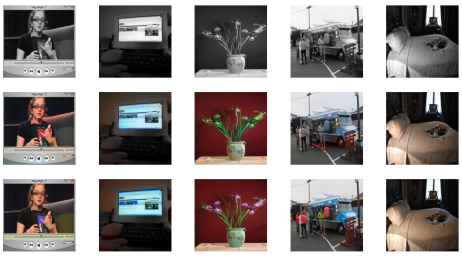

Figure 7: Example of colorization after aprox. 90 epochs

evident when working with a small dataset. Recognizing the need for a more robust strategy, this research diverged from the conventional approach.

To address limitations in the baseline colorization model, another (not tried yet) strategy can be—training the generator separately in a supervised, deterministic manner. This aims to overcome challenges in Generative Adversarial Network (GAN) training, where both the generator and discriminator lack initial task knowledge. This shift enhances the model's ability to capture intricate details, particularly with limited data. This approach not only tackles identified shortcomings but also paves the way for future work, including optimizing pretraining.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[2] 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. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016.

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[5] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 577–593. Springer, 2016.

[6] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5801–5810, 2020.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image*

Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pages 234–241. Springer, 2015.

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[9] Runsheng Xu, Zhengzhong Tu, Yuanqi Du, Xiaoyu Dong, Jinlong Li, Zibo Meng, Jiaqi Ma, Alan Bovik, and Hongkai Yu. Pik-fix: Restoring and colorizing old photos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1724–1734, 2023.

[10] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.

[11] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.