Automatic Image Colorization with Adversarial and Classification Losses

Lorenzo Saccaro

22 September 2023

Università degli Studi di Padova

Outline

Introduction

- Colorization is the process of adding color to a grayscale image.
- It is a **non-trivial** task: for each pixel at least two values must be predicted from the single input.
- It is a **multimodal** problem: multiple **plausible** colorizations are possible.
- Multiple approaches have been proposed in the literature: simple CNNs, user-guided networks etc.

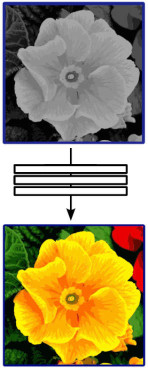


Figure: Colorization task

Related Works: pix2pix

- Philip Isola et. al. propose a general framework for **image-to-image translation** based on conditional GANs.

- **U-Net**-based generator and a **PatchGAN** discriminator.

- The grayscale image is fed to both the generator and the discriminator as side information

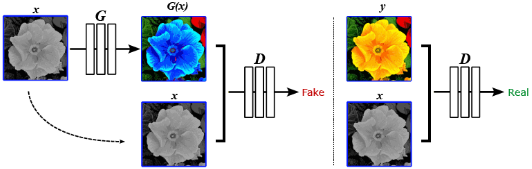


Figure: pix2pix framework

Related Works: ChromaGAN

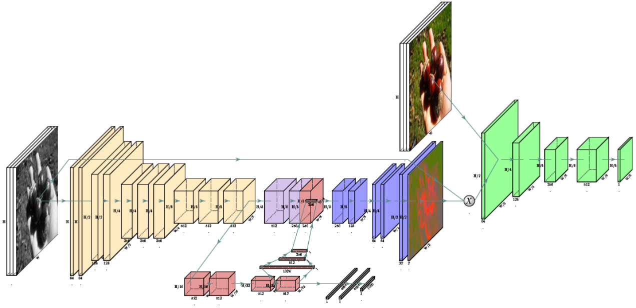


Figure: ChromaGAN architecture

- Patricia Vitoria et. al. propose a novel architecture called **ChromaGAN**
- **Pretrained VGG-16** on ImageNet as backbone of the generator
- The model is **jointly** trained to extract **semantic interpretation** of the scene by minimizing the **KL divergence** between the predicted and the ground truth class distributions.

Dataset: Places205

Università degli Studi di Padova

- **Places205**: 2.5 million images belonging to 205 scene categories
- Great variety and detail-rich images
- Only a **10%** subset has been used due to constraints on time and computational resources
- Starting from 238136 images, 162 are removed for being in grayscale. The following split is performed:
  - Training set: 202278 images
  - Validation set: 11899 images
  - Test set: 23797 images

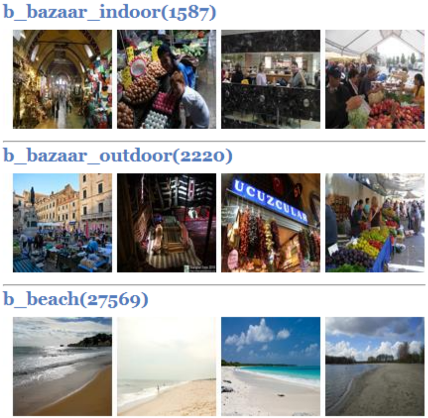


Figure: Places205 dataset samples

Preprocessing

The following preprocessing steps are performed:

- Conversion to the **CIE L*a*b** color space:
  - Drastically reduce the **complexity** of the problem
  - It is **perceptually uniform**: numerical differences between values are proportional to perceived differences between colors
- Image resizing to 256×256
- Input normalization in the $[-1, 1]$ range
- Random horizontal flipping with probability 0.5 during training

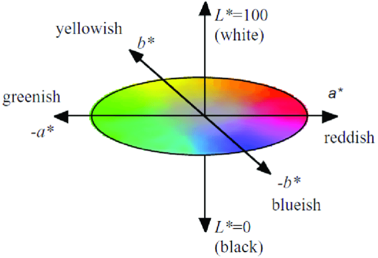


Figure: CIE L*a*b* color space visualization

Architecture: Generator

- **Encoder blocks**: Conv with stride 2, BatchNorm, LeakyReLU with slope 0.2
- **Decoder blocks**: ConvTranspose with stride 2, BatchNorm, ReLU (+50% Dropout first 3)
- **Skip connections**
- **Tanh** final activation (outputs in $[-1, 1]$)
- Parallel classification branch:
  - 3 encoder blocks + Flatten layer
  - **Feature fusion**: 2 Dense layers (1024, 512 units) + ReLU
  - **Classification head**: 2 Dense layers (2048, 1024 units) + ReLU + 50% Dropout + Dense layer (205 units)
- 54 M and 97 M parameters respectively.

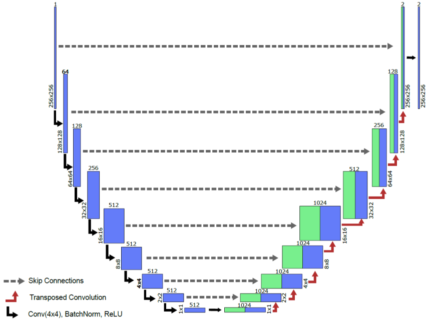


Figure: Generator U-Net architecture

Architecture: Discriminator

Università degli Studi di Padova

- Sequence of **encoder blocks** with increasing number of filters (64, 128, 256, 512)
- The last layer is a Conv layer that returns a 30×30 matrix
- Each entries corresponds to a 70×70 **patch** of the input image: it is its corresponding **receptive field**.
- The **PatchGAN** discriminator predicts whether each of these patches is real or fake.
- 2.5 M parameters.

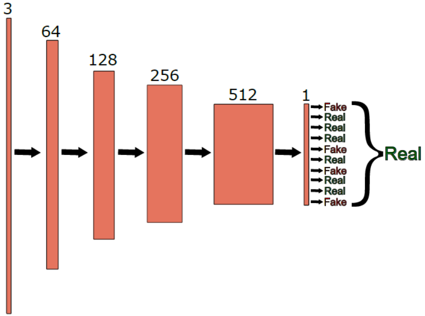


Figure: Discriminator PatchGAN architecture

cGAN Objective

- In the **adversarial** setting the generator (G) tries to fool the discriminator (D).
- $\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$
- $G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D)$
- $G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$
- **One-sided label smoothing** is applied to the real labels: if the discriminator becomes too confident, it could lead to **training instability** and **mode collapse**.

Wasserstein GAN

- $W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[||x, y||\right]$
- $\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})] - \mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})]$
- The discriminator (critic) must be 1-Lipschitz: one solution is **weight clipping**.
- $\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})] + \lambda \mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]$
- Improve **training stability** and make optimization of the generator easier, but longer.
- A couple of changes are needed:
  - The BatchNorm layers in the discriminator are disabled
  - The discriminator is trained more than the generator (5 times)

Training configurations

Three different configurations have been tested:
$\mathcal{L} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{class}\mathcal{L}_{class}$

① $\mathcal{L} = \mathcal{L}_{cGAN} + 100\,\mathcal{L}_{L1}$ (GAN + L1)

② $\mathcal{L} = \mathcal{L}_{cGAN} + 100\,\mathcal{L}_{L1} + 0.003\,\mathcal{L}_{cross-entropy}$ (GAN + L1 + Class)

③ $\mathcal{L} = 0.1\,\mathcal{L}_{WGAN-GP} + \mathcal{L}_{L2} + 0.003\,\mathcal{L}_{cross-entropy}$ (WGAN-GP + L2 + Class)

Adam optimizer with $\beta_1 = 0.5$ and initial learning rate of 2×10^{-4} + **cosine annealing scheduler** with final learning rate of 2×10^{-6} for ① and ②. Constant learning rate of 2×10^{-5} for ③. 30 epochs for all the configurations.

Evaluation metrics

The following metrics are used to **quantitatively** evaluate the quality of the generated images:

- **Fréchet Inception Distance** (FID): measures the distance between the real and the generated distributions of the Inception-v3 features.
- **Structural Similarity Index Measure** (SSIM): measures the similarity between two images.
- **Peak Signal-to-Noise Ratio** (PSNR): measures the ratio between the maximum possible power of a signal and the power of corrupting noise.
- **Learned Perceptual Image Patch Similarity** (LPIPS): measures the distance between two images in the VGG-16 feature space.

Results: quantitative analysis

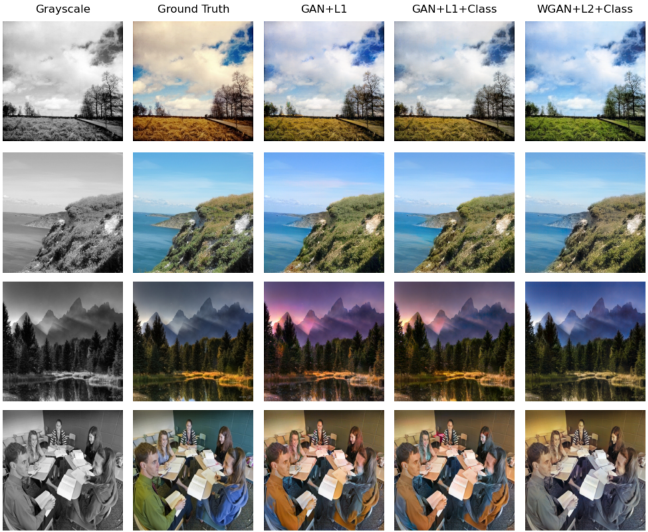
	FID ↓	SSIM ↑	PSNR ↑	LPIPS ↓
GAN + L1	1.506	0.893	26.360	0.098
GAN + L1 + Class	**1.217**	0.911	27.324	0.076
WGAN-GP + L2 + Class (30 epochs)	2.593	**0.934**	**27.734**	**0.073**
WGAN-GP + L2 + Class (50 epochs)	2.357	**0.934**	27.666	**0.073**

Table: Metrics evaluated on the test set. The best value for each metric is highlighted in bold.

Results: visual inspection (1)

Università degli Studi di Padova

Results: visual inspection (2)



Final remarks

- Overall, satisfying results, especially when dealing with **outdoor** scenes.
- Struggles with **indoor** scenes, and in particular with **people** and **objects**.
- The added classification loss improves the results both **quantitatively** and **qualitatively**.
- The adoption of the **WGAN-GP** objective helps with training stability, but further investigations are needed w.r.t. the obtained results.

Future works

- **More data**: see how the model performs on the whole Places205 dataset.
- **Hyperparameter tuning**: both for the architecture and the relative weights of the losses.
- Improve performance on people and objects:
  - Use an off-the-shelf **object detector** to extract object instances from the input images
  - During training, with a given probability, feed the model the extracted instances while following the standard training procedure.
  - Leverage this **data augmentation** technique to see if it improves the results in the aforementioned cases.

References

pix2pix paper: Phillip Isola et. Al. “Image to Image Translation with Conditional Adversarial Networks”, CVPR 2017

ChromaGAN paper: Patricia Vitoria et. Al., ”ChromaGAN: An Adversarial Approach for Picture Colorization”, 2020

WGAN paper: Arjovsky et. Al., ”Wasserstein GAN”, 2017

WGAN-GP paper: Gulrajani et. Al., ”Improved Training of Wasserstein GANs”, 2017

Instance-Aware colorization paper: Su et. Al., ”Instance-aware Image Colorization”, 2020

code available at https://github.com/lorenzo-saccaro/image-colorization-classification