Neural Network and Deep Learning Course Final Project

Image Colorization using Conditional Generative Adversarial Networks X-ray

Prof. Jacopo Pegoraro

Asal Rangrasiasl - 2046832
Roya Joulaei- 2050954
Mojtaba Roshana – 2050955

20/02/2023

CONTENTS

Image Colorization

image colorization can be a powerful tool for creating visually striking and emotionally engaging images that can be used in a variety of different contexts.

Related Works

- ❑ Fully convolutional neural netowrk
- ❑ CNN and inception-ResNet-V2
- ❑ Cluster-based-method (K-means)
- ❑ cGAN

Perceptual Losses for Rea
and Super-R

Justin Johnson(✉), Alexandr

Department of Computer Science, Sta
{jcjohns,alahi,feifeili

Abstract. We consider image transfor
image is transformed into an output
problems typically train feed-forward
using a *per-pixel* loss between the outp
allel work has shown that high-quality i
ing and optimizing *perceptual* loss func
extracted from pretrained networks. V
approaches, and propose the use of per
feed-forward networks for image transf
on image style transfer, where a feed-fo
the optimization problem proposed by
pared to the optimization-based metho
tative results but is three orders of mag
with single-image super-resolution, whe
a perceptual loss gives visually pleasing

Keywords: Style transfer · Super-res

1 Introduction

Many classic problems can be framed as *im*
tem receives some input image and transf
ples from image processing include denoisin
where the input is a degraded image (noi

Image-to-Image Translation with Conditional Adversarial Networks

Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros

Berkeley AI Research (BAIR) Laboratory, UC Berkeley

Figure 1: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

Abstract

We investigate conditional adversarial networks as a general-purpose solution to image-to-image translation problems. These networks not only learn the mapping from input image to output image, but also learn a loss function to train this mapping. This makes it possible to apply the same generic approach to problems that traditionally would require very different loss formulations. We demonstrate that this approach is effective at synthesizing photos from label maps, reconstructing objects from edge maps, and colorizing images, among other tasks. Moreover, since the release of the pix2pix *software associated with this paper, hundreds of twitter users have posted their own artistic experiments using our system. As a community, we no longer hand-engineer our mapping functions, and this work suggests we can achieve reasonable results without hand-engineering our loss functions either.*

cept may be expressed in either English or French, a scene may be rendered as an RGB image, a gradient field, an edge map, a semantic label map, etc. In analogy to automatic language translation, we define automatic *image-to-image translation* as the problem of translating one possible representation of a scene into another, given sufficient training data (see Figure 1). Traditionally, each of these tasks has been tackled with separate, special-purpose machinery (e.g., [14, 23, 18, 8, 10, 50, 30, 36, 16, 55, 58]), despite the fact that the setting is always the same: predict pixels from pixels. Our goal in this paper is to develop a common framework for all these problems.

The community has already taken significant steps in this direction, with convolutional neural nets (CNNs) becoming the common workhorse behind a wide variety of image prediction problems. CNNs learn to minimize a loss function –

n
Efros
du
paper attacks
photograph.
proaches have
saturated col-
duces vibrant
ertainty of the
ebalancing at
sult. The sys-
st time and is
rithm using a
oose between
cessfully fools
evious meth-
l pretext task
nnel encoder.
everal feature
elf-supervised
hallucinating their
vo out of the three
one notices that in
provide ample cues

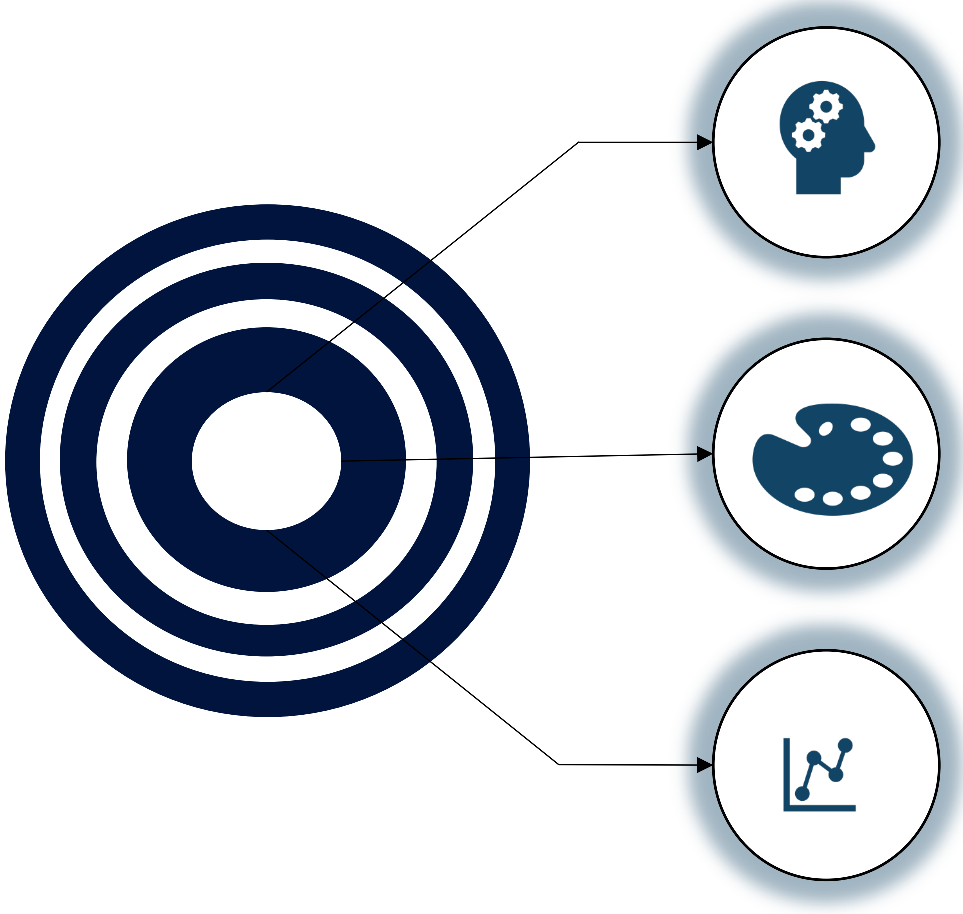
INTRODUCTION

we will explore the method of colorization using Conditional GAN (cGANs) proposed by Goodfellow et al.In deep generative modeling, deep neural networks learn a probability distribution over a given set of data points and generate similar data points

we use Lab color space instead of RGB to train the models because To train a model for colorization, we should give it a grayscale image and hope that it will make it colorful.

we use a UNet-based architecture for the generator, and using a convolutional PatchGAN for the discriminator.

AIM OF THE PROJECT



Proposed Method :
the cGAN takes a grayscale image as input and generates corresponding colorized image

Training Model, UNET & PathGan

Evaluate:
The Discriminator and GAN loss, The Generator loss

MATERIAL AND METHODS

Understanding The Context	Retrieving The Data	Dataset Preparation	Building Models and Training	Evaluation of Models and Reporting
study about different methods for colorization a grayscale image	ImageNetmini-1000 which is an image database organized by WordNet heirachy	Lab color space instead of RGB to train the models	We built a **cGAN and implement UNET as generator and patch GAN as Discriminator**	As a final result, we produced The Discriminator and GAN loss and the Generator loss

Dataset Preparation

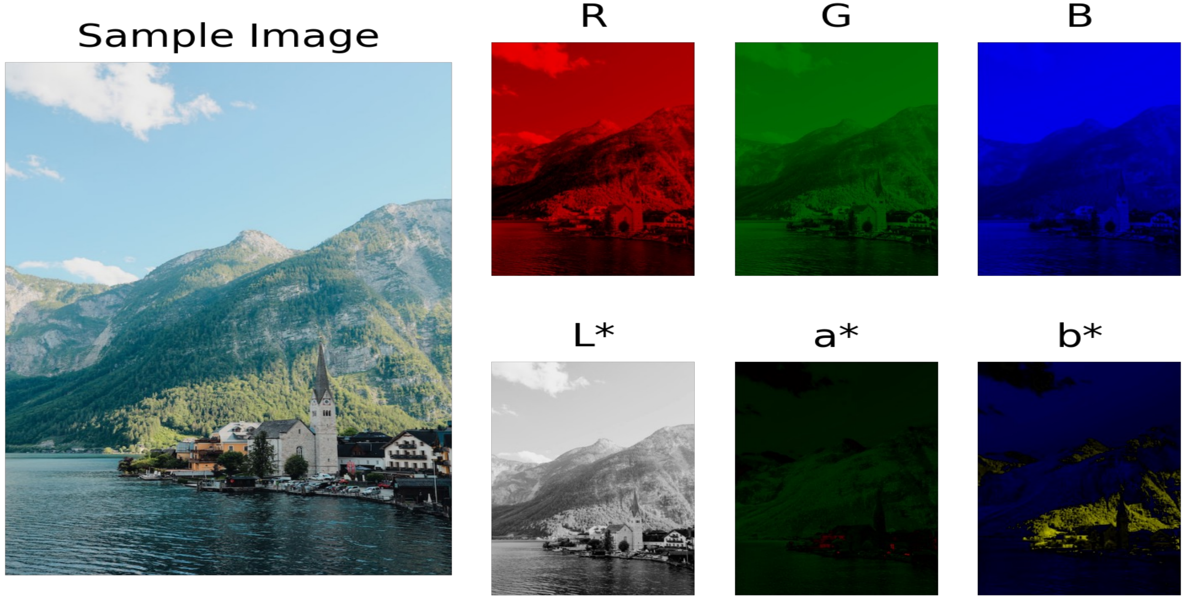
- ImageNet
  - 5000 Random sample
- Augmentation
  - Random Horizontal Flip
- Dataset
  - 80% Training
  - 20% Validation

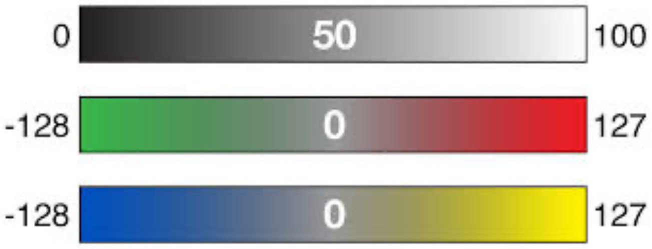
Data Set sample

Augmented sample

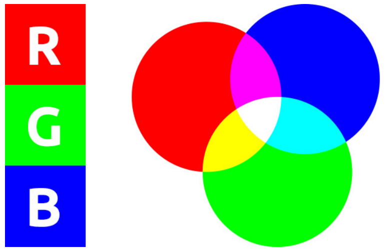
Dataset Preparation

- ❑ RGB
  - ❑ 3 different Channels for red, green, and blue.
- ❑ LAB
  - ❑ L: Lightening,
  - ❑ a: Green to Red,
  - ❑ b: Blue to Yellow.



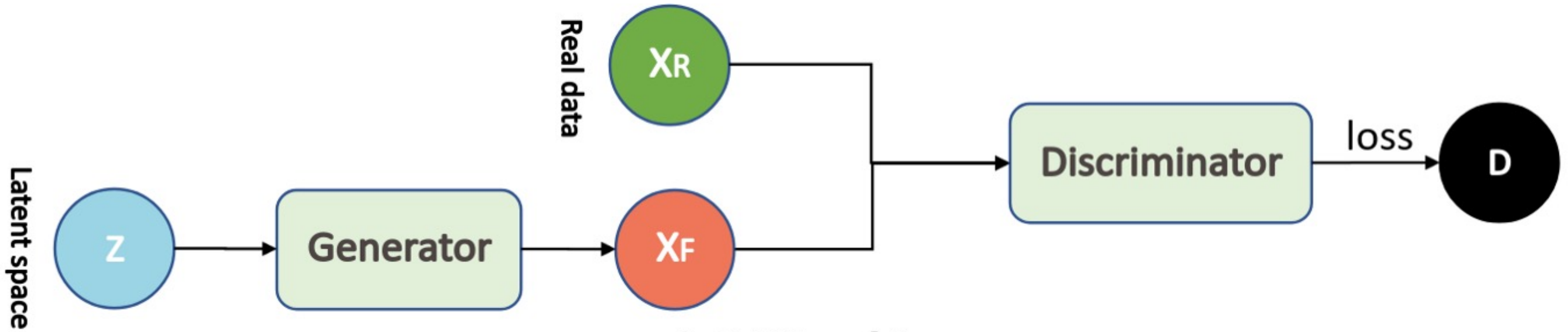


Lab

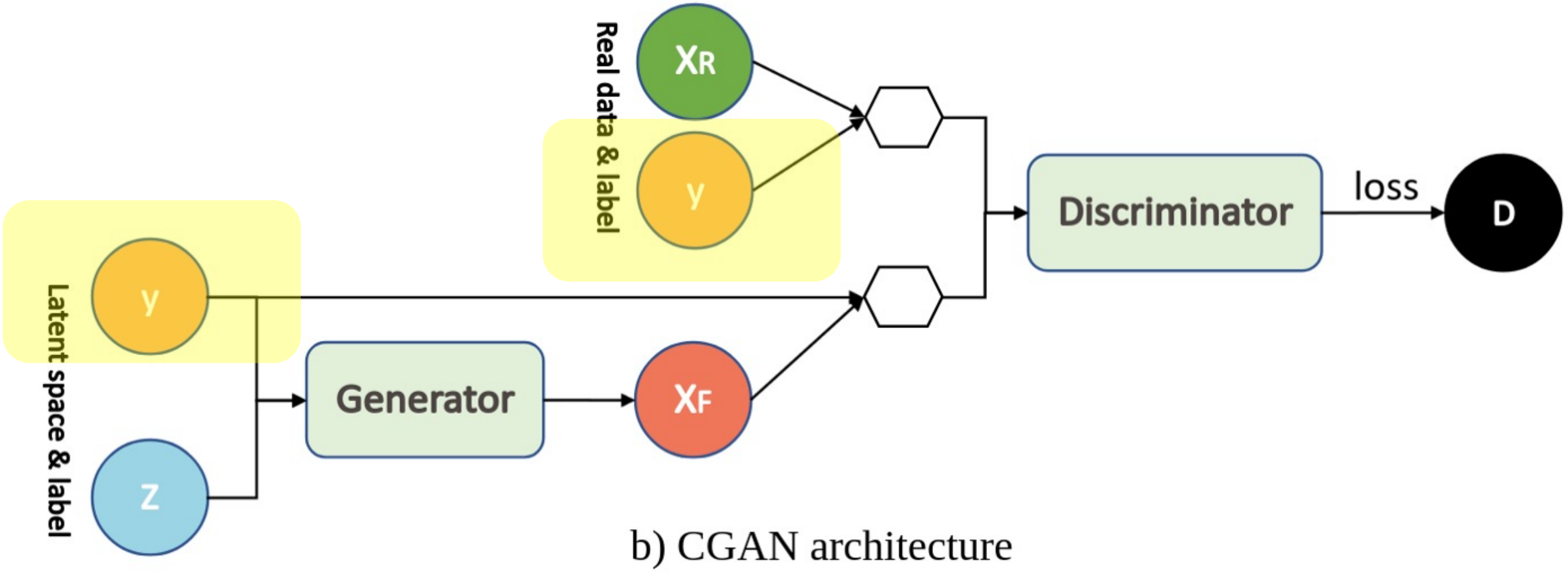


RGB

cGAN vs. GAN



a) GAN architecture



b) CGAN architecture

conditional GAN architecture

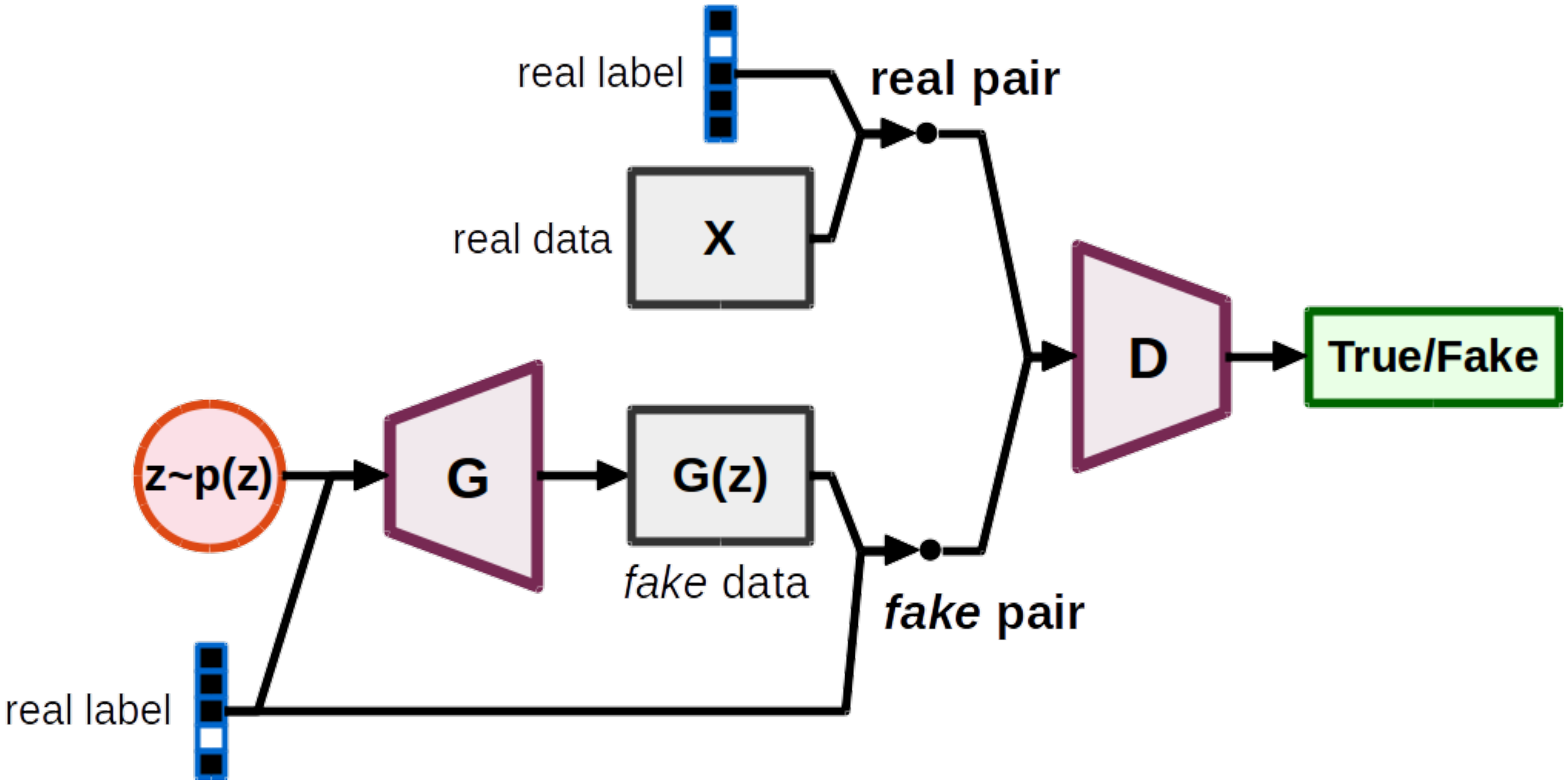
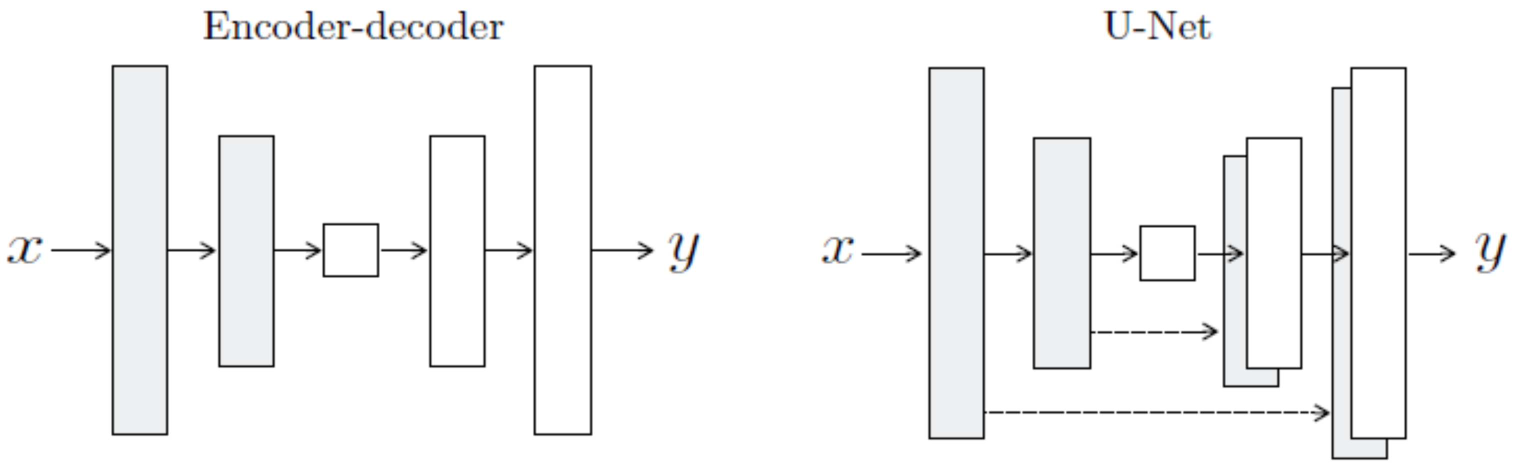


Illustration of UNET and EncoderDecoder architecture

encoder block of the generator network contains the architecture as follows:
C64-C128-C256-C512-C512-C512-C512-C512
In terms of decoder:
 CD512-CD512-CD512-C512-C256-C128-C64
The skip connections concatenate activations from layer i to layer n-i. This changes the number of channels in the decoder: UNet decoder:
CD512-CD1024-CD1024-C1024-C1024-C512 -C256-C128



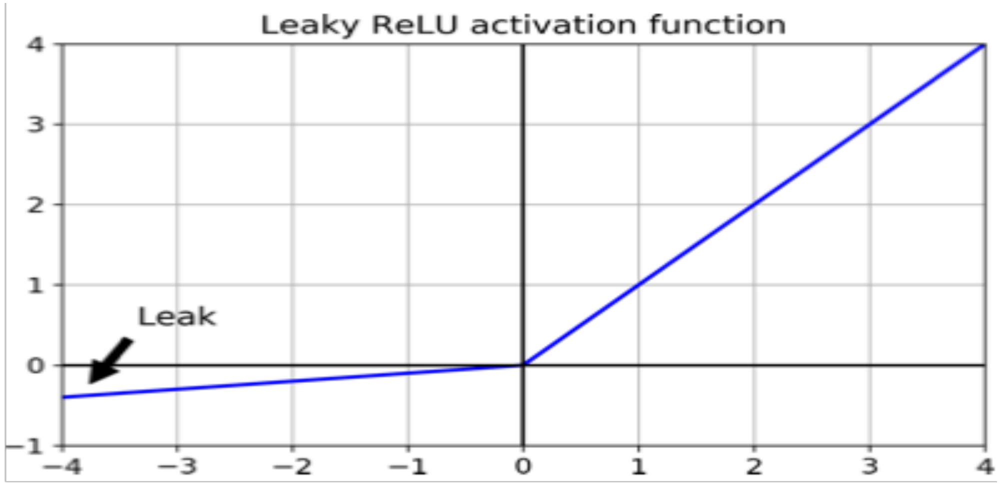
Most of the convolutional layers will also be followed by the batch normalization layers(except for the first C64 layer).

Hyperparameters of UNET

Activation function of Generator

Encoder: Leaky ReLU, slope = 0.2
Decoder: Relu, slope = 0.2
We use the hyperbolic tangent (tanh) activation function in the last layer of a U-Net architecture

Dropout → Rate = 0.5



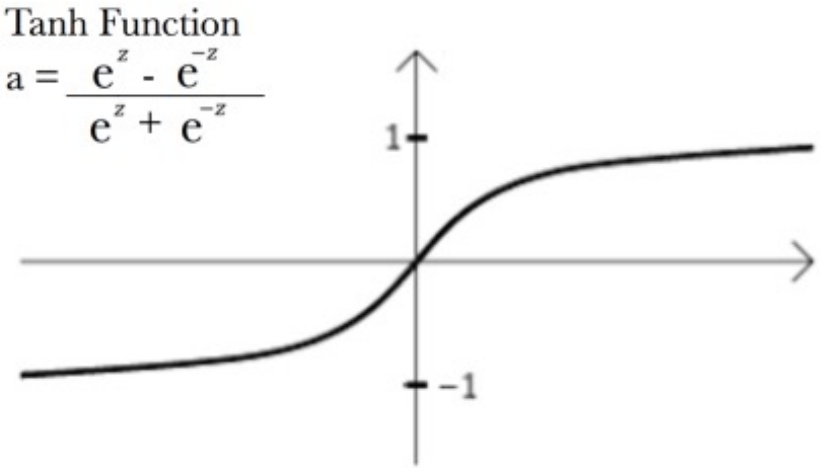


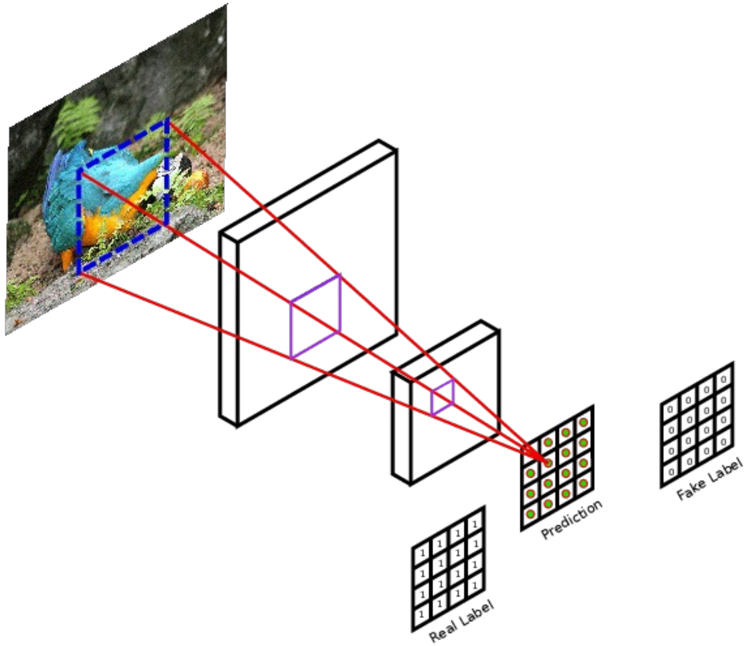
Illustration of PatchGAN

- ❑ Patch Size
  - ❑ 1 × 1
  - ❑ 16 × 16
  - ❑ 70 × 70 ✓
  - ❑ 286×286

The discriminator structure will follow the pattern
build of C64-C128-C256-C512

We implement a model by stacking blocks of Conv-BatchNorm-LeackyReLU

Loss | Binary Cross entropy



Training process & loss

Optimizer

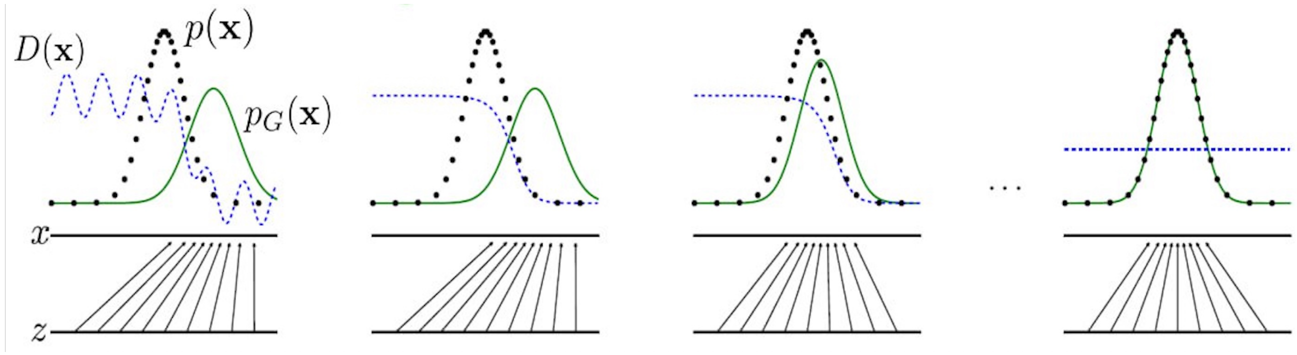
minibatch SGD , ADAM
λ = 100, β1 = 0.5, β2 = 0.999

Batch size

Size= 16

epoch

Number of epoch= 45



Loss formula for GAN :

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

Loss formula for cGAN :

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

loss of L1:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}\left[\|y - G(x,z)\|_1\right]$$

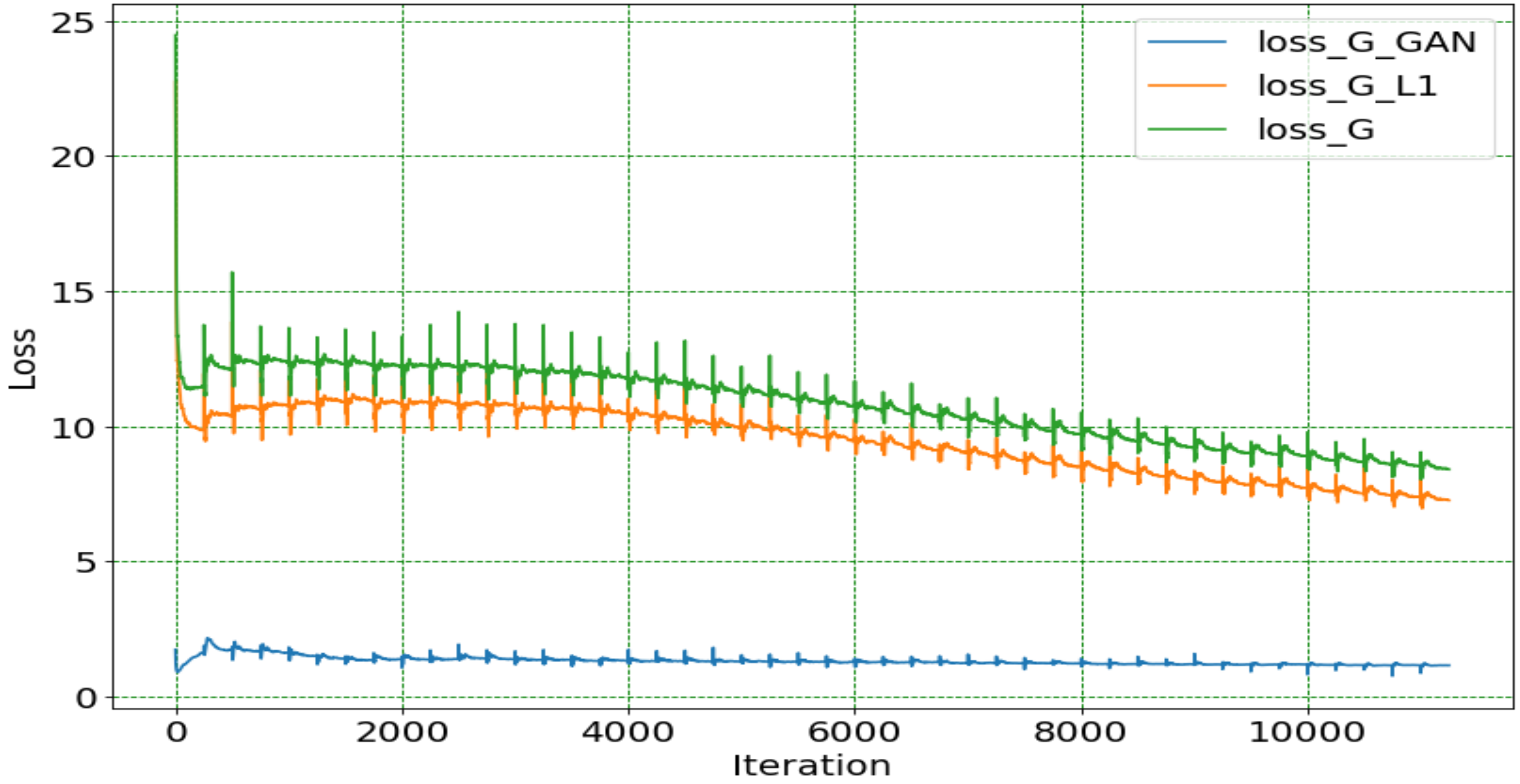
Our final objective is :

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G)$$

RESULTS

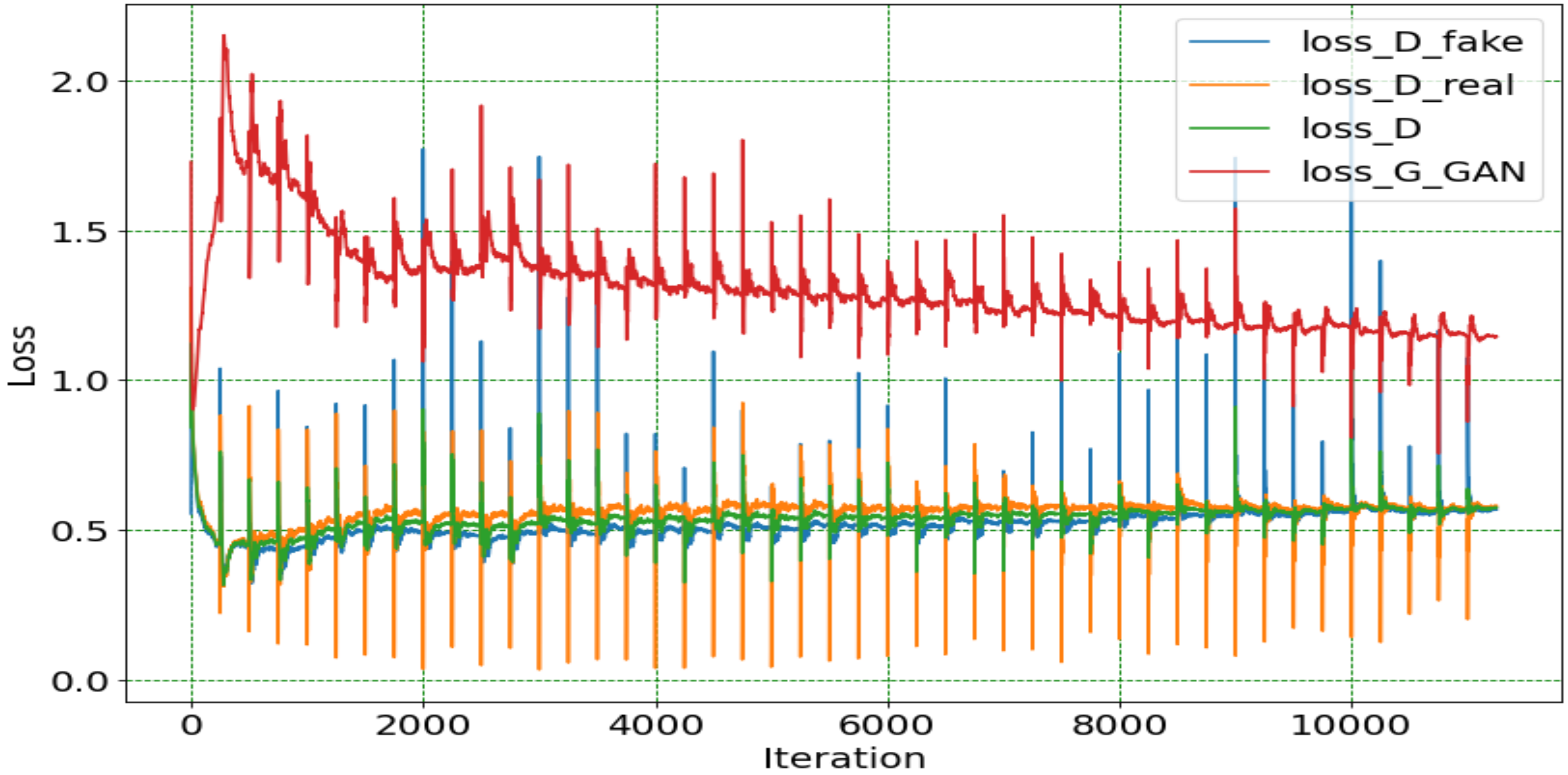
The Generator loss for 45 epochs

- **loss_G_L1 :** $\lambda \mathcal{L}_{L1}(G)$
- **loss_G_GAN:** $\mathcal{L}_{cGAN}(G, D)$
- **loss_G:** $G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$



RESULTS

The Discriminator and G GAN loss for 45 epochs



Model Performance of cGAN

Evaluation for 45 epochs.

- By increasing the number of epochs can allow the network to learn more complex relationships and generate more realistic colorization.

Ground truth | Epoch 5 | Epoch 15 | Epoch 30 | Epoch 45

Model Performance of cGAN

Results after 45 epochs

Grayscale

cGan

Ground truth

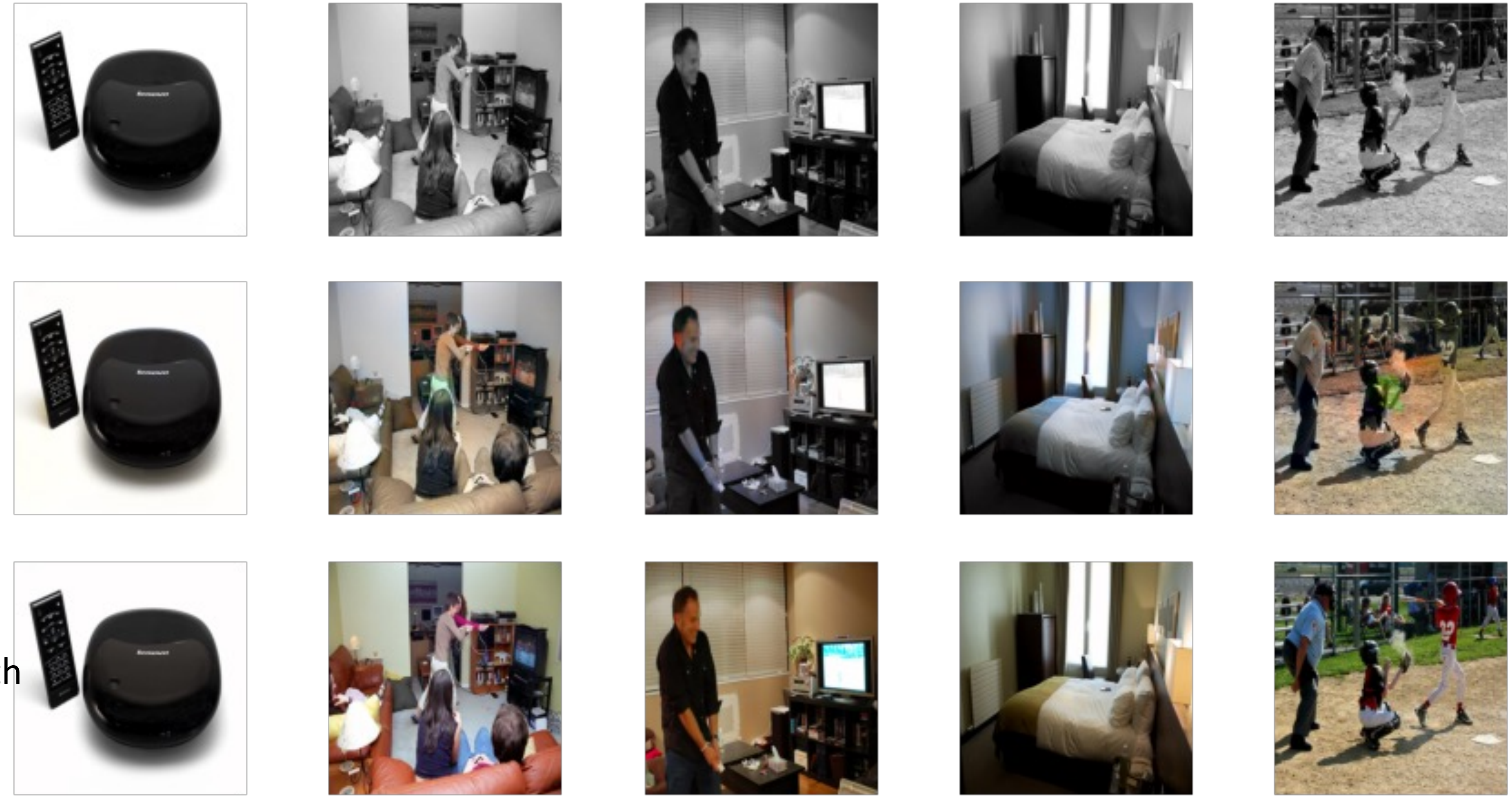
Model Performance of cGAN

Results after 45 epochs for COCO dataset

Grayscale

cGan

Ground truth



Fine tuning ResNet-18

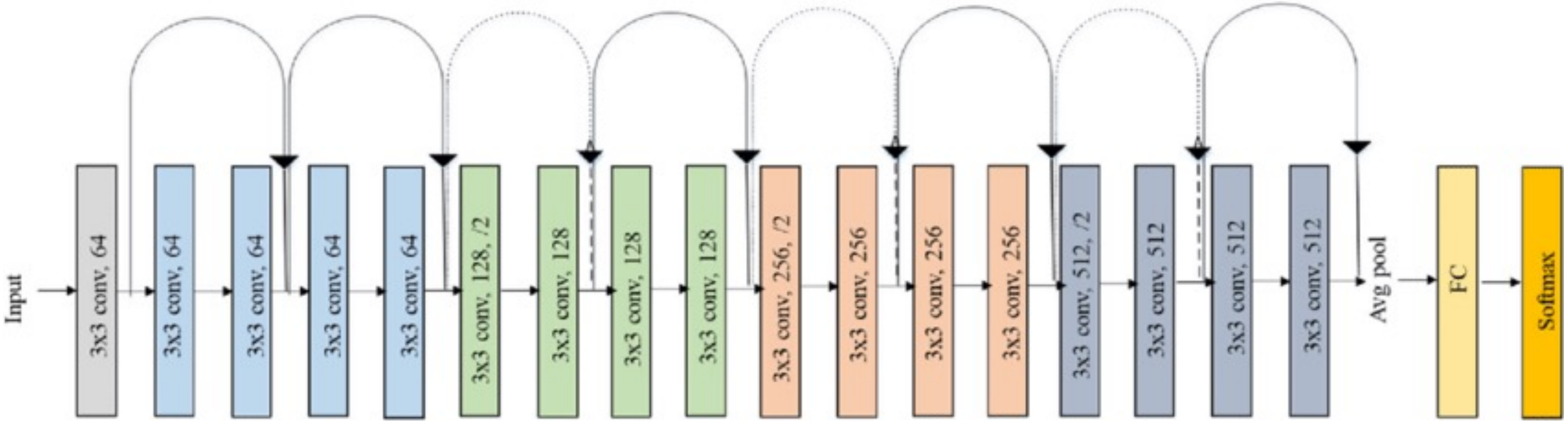
ResNet-18 is a deep convolutional neural network architecture that was introduced by Microsoft Research in 2015.
The network trained on more than a million images from the ImageNet database for classification task.

- **Removing the last two layers :**

  GlobalAveragePooling and a Fully Connected Layer

- **Use this backbone to build a U-Net as the generator:**

  2 output channels and 1 input size of 256x256
- **Train the generator only with L1 loss in 20 epochs (not depends on Discriminator)**
- **Do the same training process with new generator**

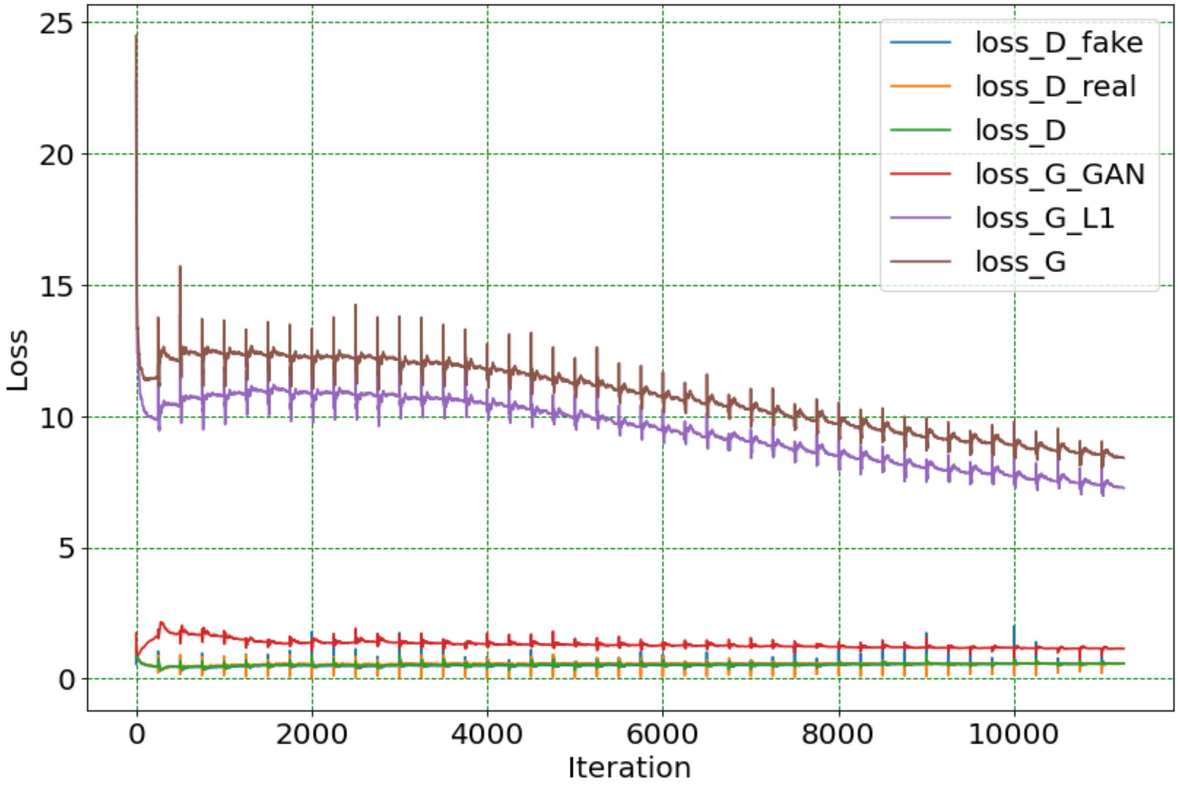


COMPARISION:

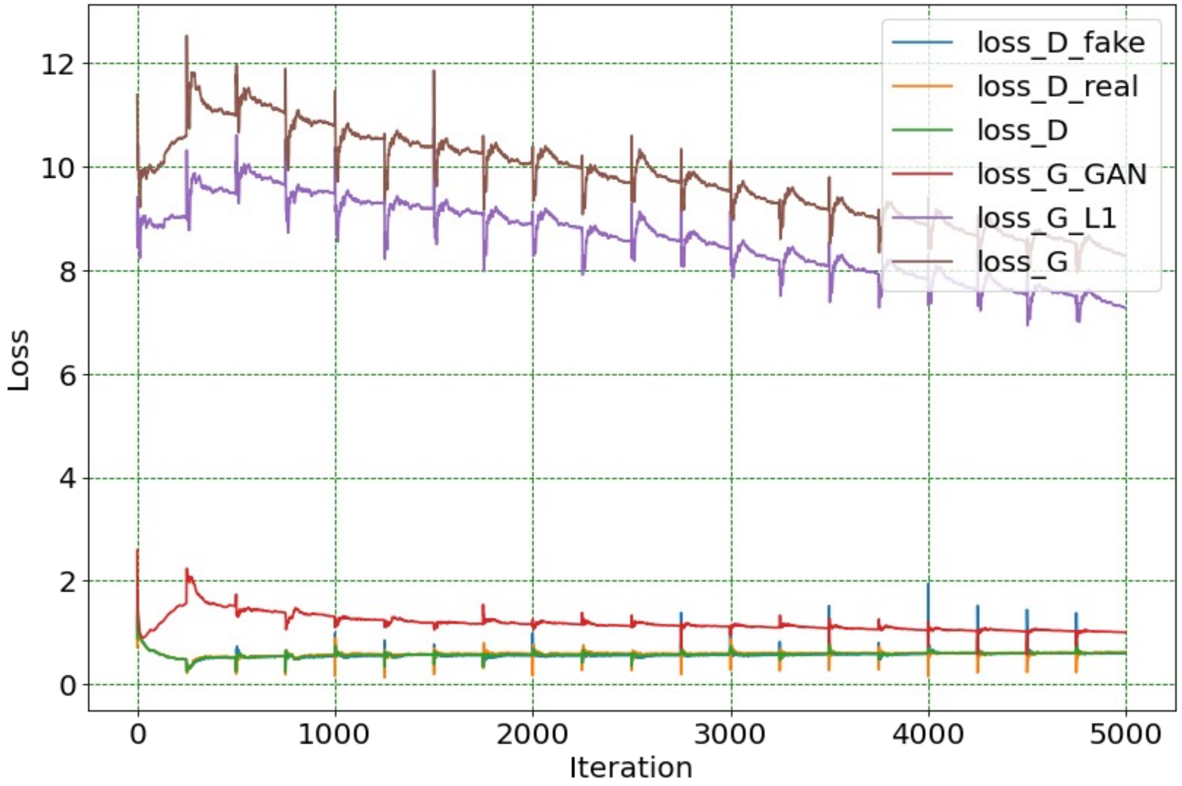
Main Model

ResNet18 as generator

Losses in 45 epochs



Losses in 20 epochs



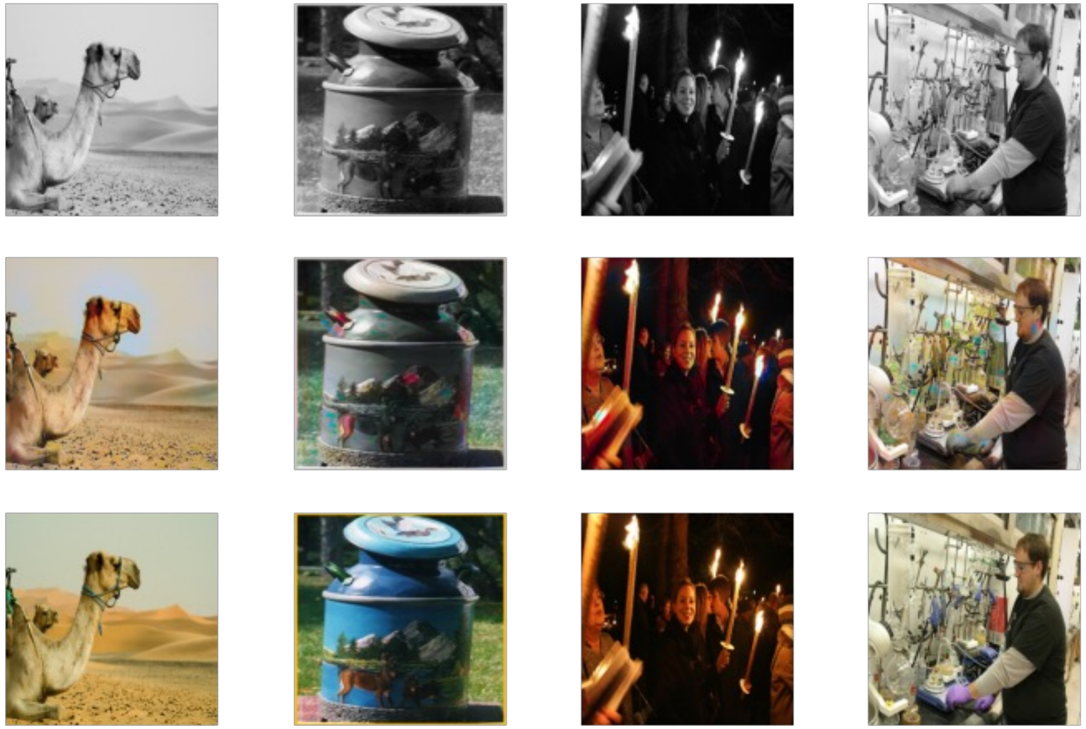
COMPARISION:

Main Model

20 epochs

ResNet18 as generator

20 epochs



Conclusion

- cGan has a good performance, even with small dataset and limited recourses we can get reasonable results
- Use Pretrained model as generator in cGan can improve the performance and we can rich better results
- With bigger dataset and a greater number of training epochs the results will be more realistic (which needs more recourses)

THANK YOU

QUESTIONS ARE WELCOME