Department of Computer Science and Engineering (Data Science)

Subject: Image Processing and Computer Vision - II Laboratory (DJ19DSL702)

AY: 2024-25

Experiment 3

(GAN for Image Colorization)

Aim: Convert Black and white image into Colored image using GAN

Theory:

1. Introduction

Generative Adversarial Networks (GANs) are a powerful class of deep learning models used for generating realistic data samples, such as images. In this lab, you will learn how to implement a GAN to transform black and white images into colored images. The GAN framework involves training two neural networks, a Generator, and a Discriminator, in a competitive manner to produce high-quality colorized images.

2. Generative Adversarial Networks (GANs)

GANs consist of two main components:

2.1 Generator

The Generator is responsible for generating new samples from random noise. In the context of black-and-white to color image translation, the generator takes a grayscale image as input and tries to generate a corresponding colored image. The generator typically uses transposed convolutions (also known as deconvolutions) to upsample the input noise into a full-sized image.

2.2 Discriminator

The Discriminator acts as a binary classifier, distinguishing between real colored images and generated (fake) colored images. It takes both real color images and generated color images as input and assigns a probability of being real to each sample. The goal of the discriminator is to distinguish real images from fake ones correctly.

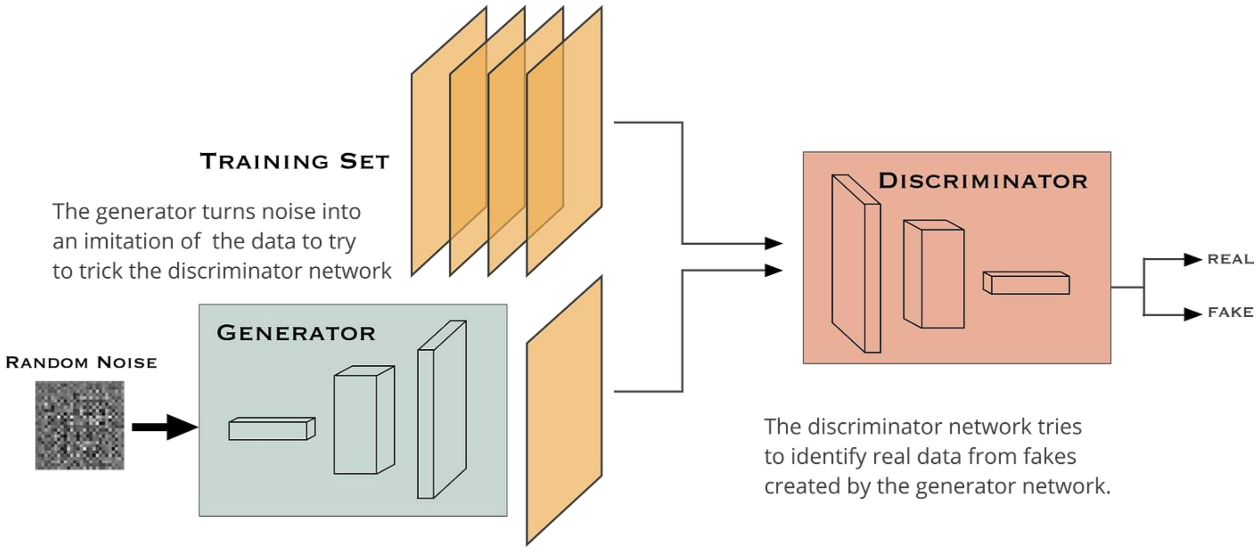
TRAINING SET
The generator turns noise into an imitation of the data to try to trick the discriminator network
RANDOM NOISE
GENERATOR
DISCRIMINATOR
REAL
FAKE
The discriminator network tries to identify real data from fakes created by the generator network.

Figure. Concept of Generator and Discriminator

3. Dataset Preparation

To perform black-and-white image-to-color image translation, you will need a dataset containing pairs of black and white images along with their corresponding colored images. Choose an appropriate dataset that suits your experiment's requirements, or you can create a custom dataset with black and white images and their colored versions.

4. GAN Architecture for Image Colorization

In this lab, you will implement a GAN architecture specifically designed for image-to-image translation tasks like black and white to color image conversion. The GAN architecture consists of the following steps:

4.1 Generator Architecture

Design the architecture of the generator network. It should take a black-and-white image as input and produce a colored image of the same size. The generator should be designed to upscale the input image while preserving and enhancing the details.

The model used as a Generator for the demonstration is shown at : (link)

4.2 Discriminator Architecture

Design the architecture of the discriminator network. It should take both real colored images and generated colored images as input and predict the probability of each being real. The discriminator should be able to differentiate between real and fake images effectively.

The model used as a Discriminator for the demonstration is shown at : (link)

5. GAN Training

Training a GAN involves a two-step process:

5.1 Generator Training

During generator training, you pass black and white images through the generator to obtain generated color images. These generated color images are then fed into the discriminator. The generator aims to generate color images that are realistic enough to fool the discriminator into classifying them as real.

5.2 Discriminator Training

In discriminator training, you use both real colored images and generated colored images to train the discriminator. The discriminator's goal is to correctly classify real images as real and generated images as fake.

6. Loss Functions

The training of a GAN involves minimizing two different loss functions:

6.1 Adversarial Loss

The adversarial loss measures how well the generator can generate images that resemble real colored images. It is computed based on the discriminator's classification of the generated images.

6.2 Perceptual Loss

The perceptual loss ensures that the generated images retain the details and content of the original color images. It is computed based on a feature representation extracted from a pre-trained image classification network (e.g., VGG-16).

7. Hyperparameter Tuning

To achieve the best results, experiment with various hyperparameters such as learning rate, batch size, number of epochs, and network architectures for both the generator and discriminator. Hyperparameter tuning is essential to strike the right balance between stability and convergence speed.

8. Results and Evaluation

After training the GAN, evaluate the quality of the generated colored images. Use both qualitative and quantitative measures to assess the performance. Visualize the colorized images and compare them with the ground truth colored images.

You can use metrics like Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) to quantify the quality of the generated images.

9. Conclusion

In this lab, you have learned how to implement a Generative Adversarial Network for black and white image to colored image translation. GANs are versatile tools for image processing tasks and have shown impressive results in generating realistic and high-quality images. By completing this lab, you have gained valuable insights into the GAN architecture and its application in image colorization.

Lab Assignments to complete after this session:

Use the given dataset and perform the following tasks:

Dataset: https://drive.google.com/drive/folders/1R0iAMQ-sqZXHjPgnlNc99H_X7S4P0524?usp=sharing

This dataset contains a zip file of RGB images to train the GAN.

The steps involved in the experiment are as follows:

a. Preparing the dataset: Download and preprocess the dataset. Download the zip file on colab (using " !unzip <file path>" command).

b. Parse only a part of the dataset to save computation and convert each image to a grayscale image.

c. The Generator: Prepare a Generator model that would take in a grayscale image (x) and produce a RGB image(G(x)). Note, x will be a tensor of shape (batch size , 120 , 120 , 1) and the output G(x) will have a shape (batch size , 120 , 120 , 3)

d. The Discriminator: Prepare a Discriminator model, represented as D , that will take in the real image y (from the training data) and the generated image G(x) (from the generator) to output two probabilities.

e. Loss Functions: Define Loss Functions for the Generator and Discriminator.

f. Training: Train the GAN on the giving dataset.

g. Results: Display the Grayscale input, Colourized output and the Groundtruth side-by-side.