## Project Understanding

## A. Introduction

📘 Pix2pix is a powerful model for image-to-image translation tasks, but it can be further improved for specific applications such as colorization. One way to improve the performance of pix2pix for colorization is to use a Wasserstein GAN (WGAN) instead of the traditional GAN architecture. WGANs use the Wasserstein distance metric to train the generator and discriminator, which can help stabilize the training process and produce more realistic results.

Another way to improve the performance of pix2pix for colorization is to use a U-Net architecture based on residual blocks. U-Net is a type of convolutional neural network (CNN) that is well-suited for image segmentation tasks. It consists of a series of convolutional layers and max pooling layers, with skip connections between layers of the same resolution. This allows the network to learn fine details of the input image, which can be particularly useful for colorization tasks.

Residual blocks are a type of building block for neural networks, which consist of multiple layers with skip connections. The skip connections allow the gradient to pass through the layers more easily, which can help the network converge faster and produce better results.

Using WGAN with a U-Net architecture based on residual blocks can help improve the performance of pix2pix for colorization by providing better stability and improved ability to learn fine details of the input image.

📌 The goal of this paper is :

- The goal of the "Image-to-Image Translation with Conditional Adversarial Networks" (pix2pix) paper is to propose a method for image-to-image translation using a conditional GAN architecture.
- The method uses a generator network that is trained to convert images from one domain (e.g. sketches) to another domain (e.g. photographs).
- The generator is trained using a combination of adversarial loss and L1 loss.
- The generator takes an image from the input domain and a random noise as input, and generates an image in the target domain.
- The discriminator network is trained to classify the generated image as real or fake, based on whether it is similar to a target image from the target domain.
- The paper uses the patch GAN architecture to discriminator with 70x70 patches.
- The authors showed the effectiveness of the proposed method on a variety of image-to-image translation tasks, such as converting edges to photographs, day to night, and labels to street scenes.

arrow_upward

arrow_downward

link

edit

delete

more_vert

❓ What is image colorization ?

Image colorization is the process of adding color to a grayscale image or a black and white image. It involves mapping the intensity values of the grayscale image to a color space, such as RGB, and then filling in the missing color channels to produce a full-color image. There are different ways to approach image colorization, but most methods involve some form of image processing, such as image segmentation, texture synthesis, or machine learning.

One popular approach is to use a deep learning-based method, such as a convolutional neural network (CNN) to colorize images. This approach typically involves training a CNN on a large dataset of color images, and then using this network to predict the missing color channels of a grayscale image.

Another approach is to use a Generative Adversarial Network (GAN) model, where a generator network generates the color version of the grayscale image and a discriminator network is trained to distinguish between the generated color version and the real color image.

In recent years, there have been some impressive results in image colorization using deep learning-based methods, which can produce high-quality colorization results on a wide range of images.

## B. Theory

## 1. Generative adversarial networks (GAN)

arrow_upward

arrow_downward

link

edit

delete

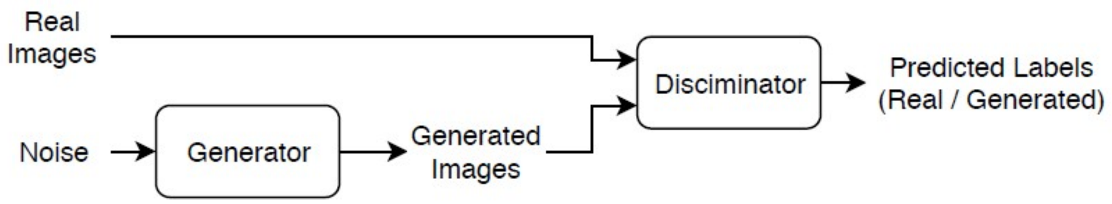
more_vert

A generative adversarial network (GAN) is a type of deep learning network that can generate data with similar characteristics as the input training data.

A GAN consists of two networks that train together:

- Generator — Given a vector of random values as input, this network generates data with the same structure as the training data.

- Discriminator — Given batches of data containing observations from both the training data, and generated data from the generator, this network attempts to classify the observations as "real" or "generated".



**2. Conditional Generative adversarial networks (cGAN)**

arrow_upward

arrow_downward

link

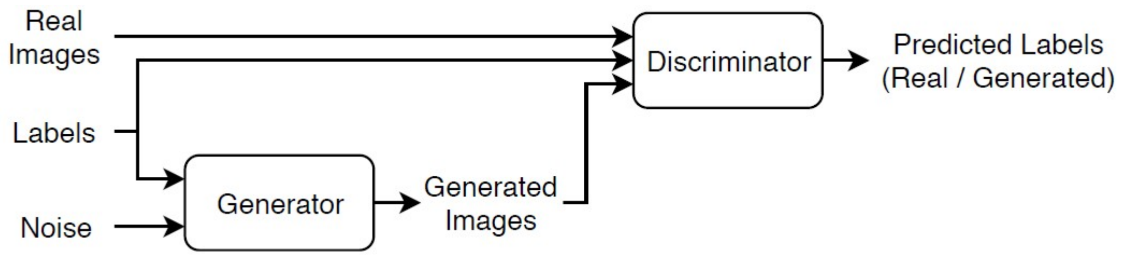
edit

delete

more_vert

A conditional generative adversarial network (CGAN) is a type of GAN that also takes advantage of labels during the training process.

- Generator — Given a label and random array as input, this network generates data with the same structure as the training data observations corresponding to the same label.
- Discriminator — Given batches of labeled data containing observations from both the training data and generated data from the generator, this network attempts to classify the observations as "real" or "generated".



To train a conditional GAN, train both networks simultaneously to maximize the performance of both:

- Train the generator to generate data that "fools" the discriminator.
- Train the discriminator to distinguish between real and generated data.

To maximize the performance of the generator, maximize the loss of the discriminator when given generated labeled data. That is, the objective of the generator is to generate labeled data that the discriminator classifies as "real".

To maximize the performance of the discriminator, minimize the loss of the discriminator when given batches of both real and generated labeled data. That is, the objective of the discriminator is to not be "fooled" by the generator.

Ideally, these strategies result in a generator that generates convincingly realistic data that corresponds to the input labels and a discriminator that has learned strong feature representations that are characteristic of the training data for each label.

source : https://it.mathworks.com/help/deeplearning/ug/train-conditional-generative-adversarial-network.html

## 3. Why choosing cGAN over GAN

Conditional Generative Adversarial Networks (CGANs) are an extension of standard Generative Adversarial Networks (GANs) that are designed to handle conditional data. A CGAN consists of a generator network and a discriminator network, just like a standard GAN. However, in a CGAN, the generator and discriminator are both conditioned on some additional input data. This additional input data can be used to control the output of the generator, allowing it to produce more specific or customized results.

There are several reasons why a CGAN can be better than a standard GAN:

1. Control over the generated data: In a CGAN, the generator's output is conditioned on the input data, which allows the model to be more specific and controlled in its output. For example, if the input is a grayscale image, the model can colorize it to a specific color scheme.

2. Improved stability and training: Because the generator is conditioned on additional input data, it can be easier to train and more stable than a standard GAN. This is because the generator is able to focus on a specific subset of the data, rather than trying to generate all possible outputs.

3. Handling missing data: CGANs are well suited for handling missing data or data with missing modalities. The additional input data can be used to condition the generator to produce plausible outputs for the missing data.

4. Handling multiple classes: CGANs can be used to generate data for multiple classes in a one-to-many mapping, where the generator is conditioned on the class label and produces an image from that class.

5. Handling conditional data: In some tasks, the data is conditional, such as in image-to-image translation, where the output is conditioned on the input. CGANs can handle this kind of conditional data very well.

It's important to note that in some tasks a GAN might be enough or even better than a CGAN, it depends on the task and the data.