CS424 – Group Project Report

Akeela Darryl Fattha
akeelaf.2022@scis.smu.edu.sg

Fadhel Erlangga Wibawanto
fadhelew.2022@scis.smu.edu.sg

Tan Zhi Rong
zhirong.tan.2022@scis.smu.edu.sg

Grace Angel Bisawan
gbisawan.2022@scis.smu.edu.sg

Lee Jia Heng
jiaheng.lee.2023@scis.smu.edu.sg

Abstract

This report provides an overview of Cycle-Consistent Adversarial Networks (CycleGAN), a technique for unpaired image-to-image translation. We explore the architecture, key innovations, applications, and limitations of CycleGAN models in computer vision tasks.

Contents

1. Task 1

1.1 Introduction

Image-to-image translation is the task of converting an image from one domain to another. CycleGAN, introduced by Zhu et al. in 2017, addresses the challenge of learning such translations without paired training data. This makes it particularly useful for applications where paired examples are difficult or impossible to obtain.

1.1.1 Configuration

1.2 Architecture

1.2.1 Discriminator

GANs consist of two networks: a generator that creates images and a discriminator that evaluates them. The two networks are trained adversarially, with the generator trying to fool the discriminator.

1.2.2 Generator

CycleGAN extends the GAN framework by using two generator-discriminator pairs, allowing translation between domains X and Y. The key innovation is the cycle-consistency loss, which ensures that translating an image to the target domain and back produces the original image.

1.3 Data Preparation

1.3.1 Augmentations

Add info that we didn't use any augmentations Size used was 256x256

1.3.2 Training/Validation

A simple 80/20 split on training data with validation

1.4 Loss Functions

Our overall loss functions for CycleGAN are as follows:

$$L_{total} = L_{GAN} + \lambda_{cyc} L_{cyc} + \lambda_{id} L_{id} + \lambda_{edge} L_{edge} + \lambda_{color} L_{color} \tag{1.1}$$

1.4.1 Discriminator

Patch

Least Squares

1.4.2 Generator

Adversarial

Use basic

Cycle Consistency

Identity

Edge Consistency

Color Consistency

1.4.3 Adaptive Loss Weighting

2. Task 2

2.1 Introduction

2.1.1 Configuration

2.2 Architecture

2.2.1 Discriminator

GANs consist of two networks: a generator that creates images and a discriminator that evaluates them. The two networks are trained adversarially, with the generator trying to fool the discriminator.

2.2.2 Generator

CycleGAN extends the GAN framework by using two generator-discriminator pairs, allowing translation between domains X and Y. The key innovation is the cycle-consistency loss, which ensures that translating an image to the target domain and back produces the original image.

2.3 Data Preparation

2.3.1 Pre-Processing

2.3.2 Training/Validation

A simple 80/20 split on training data with validation

2.4 Loss Functions

Our overall loss functions for CycleGAN are as follows:

$$L_{total} = L_{GAN} + \lambda_{cyc} L_{cyc} + \lambda_{id} L_{id} + \lambda_{edge} L_{edge} + \lambda_{color} L_{color} \tag{2.1}$$

2.4.1 Discriminator

Patch

Least Squares

2.4.2 Generator

Adversarial

Use basic

Cycle Consistency

Identity

Edge Consistency

Color Consistency

2.4.3 Adaptive Loss Weighting