Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Центр Післядипломної освіти
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Попереднє навчання генеративно-змагальних моделей перекладу зображень в умовах обмеженої кількості даних
(тема)

Виконав:
студент 2 курсу, групи СШІмзд-19-1
Лавриненко Р. М.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник к.т.н., доцент, Дейнеко А. О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Центр Післядипломної освіти
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Лавриненку Роману Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Попереднє навчання генеративно-змагальних моделей перекладу зображень в умовах обмеженої кількості даних
затверджена наказом університету від 29.03.2021 р. № 36Стз
2. Термін подання студентом роботи до екзаменаційної комісії _____ _____ 20 ___ р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет та відомих наукових проектів, електронні документації, набори даних платформи Kaggle для змагання «Use GANs to create art» https://www.kaggle.com/c/gan-getting-started/
4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної області, особливостей змагання Kaggle «Use GANs to create art» та постановка задачі дослідження, переклад зображень без учителя з використанням генеративно-змагальної мережі CycleGAN, односторонні моделі перекладу зображень без застосування реконструкції, перенос знань та попереднє навчання в генеративно-змагальних мережах, генерація зображень в умовах обмеженої кількості даних, експерименти з застосуванням найновіших генеративно-змагальних моделей в змаганні Kaggle «Use GANs to create art», аналіз отриманих результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) знімок таблиці лідерів змагання Kaggle «Use GANs to create art» з позицією студента

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. каф. ШІ Дейнеко А.О.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1.	Отримання завдання на кваліфікаційне проектування	01.04.2021	виконано
2.	Аналіз завдання та пошук літератури за темою	02.04.2021	виконано
3.	Опрацювання літератури та аналіз об'єкту	06.04.2021	виконано
4.	Вибір програмних засобів	07.04.2021	виконано
5.	Проведення експериментів	18.04.2021	виконано
6.	Аналіз отриманих результатів	19.04.2021	виконано
7.	Оформлювання пояснювальної записки	23.04.2021	виконано
8.	Оформлення презентаційних матеріалів	26.04.2021	виконано
9.	Представлення на рецензування	30.04.2021	виконано
10.	Представлення кваліфікаційної роботи	__.05.2021	

Дата видачі завдання _ _____ 20 21 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 115 с., 34 рис., 1 табл., 20 формул, 1 дод., 52 джерела.

АУГМЕНТАЦІЯ ДАНИХ, ГЕНЕРАТИВНО-ЗМАГАЛЬНА МЕРЕЖА, ГЛИБОКЕ НАВЧАННЯ, ПЕРЕНЕСЕННЯ ЗНАНЬ, ПОПЕРЕДНЄ НАВЧАННЯ.

У роботі порівняно ефективність способів аугментації даних та попереднього навчання в генеративно-змагальних моделях перекладу зображень в умовах обмеженої кількості даних. Розглянуто існуючі рішення та встановлено, що вони не повністю вирішують проблему малої кількості навчальних даних, яка загрожує перенавчанням дискримінатора та постійним погіршенням результату. Вдалі результати за таких умов досягаються ранньою зупинкою навчання. Запропоновано модель двоцільового дискримінатора, за допомогою якого очікується уникнення негативних ефектів перенавчання дискримінатора та потреби в ранній зупинці навчання.

Навчання генеративно-змагальних мереж ускладнене неоднорідністю характеристик мереж, що навчаються змагаючись – генераторів та дискримінаторів. Особливо складно організувати процес навчання, якщо дискримінатор та генератор були попередньо навчені (або знання було перенесено) для вирішення проблеми обмеженої кількості навчальних даних. У такому разі дискримінатор із самого початку має значні переваги над генератором. У цій роботі запропоновано адаптивний підхід до вирішення проблеми узгодження швидкості навчання генератора та дискримінатора генеративно-змагальних мереж. На кожному кроці навчання визнається штрафний коефіцієнт, на який буде помножено градієнти перед їх застосуванням до вагових коефіцієнтів дискримінатора. Штрафний коефіцієнт вираховується від значення втрат дискримінатора, що забезпечує постійну адаптацію швидкості навчання дискримінатора.

РЕФЕРАТ

Записка пояснительная: 115 с., 34 рис., 1 табл., 20 формул, 1 прил., 52 источника.

АУГМЕНТАЦИЯ ДАННЫХ, ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНАЯ СЕТЬ, ГЛУБОКОЕ ОБУЧЕНИЕ, ПЕРЕНОС ЗНАНИЙ, ПРЕДВАРИТЕЛЬНОЕ ОБУЧЕНИЕ.

В работе сравнивается эффективность способов аугментации данных и предварительного обучения в генеративно-состязательных моделях перевода изображений в условиях ограниченного количества данных. Рассмотрены существующие решения и установлено, что они не полностью решают проблему малого количества данных, что грозит переобучением дискриминатора и постоянным ухудшением результата. Удачные результаты в таких условиях достигаются ранней остановкой обучения. Предложена модель двухцелевого дискриминатора, с помощью которого ожидается избежание негативных эффектов переобучения дискриминатора и потребности в ранней остановке.

Обучение генеративно-состязательных сетей затруднено неоднородностью характеристик сетей, обучающихся соревнуясь – генераторов и дискриминаторов. Особенно сложно организовать процесс обучения, если дискриминатор и генератор были предварительно обучены (или знание было перенесено) для решения проблемы ограниченного количества учебных данных. В таком случае дискриминатор с самого начала имеет преимущество над генератором. В этой работе предложен адаптивный подход к решению проблемы согласования скорости обучения генератора и дискриминатора. На каждом шаге обучения определяется штрафной коэффициент, на который будут умножены градиенты перед их применением к весовым коэффициентам дискриминатора. Штрафной коэффициент вычисляется от значения потерь дискриминатора, что обеспечивает постоянную адаптацию скорости обучения дискриминатора.

ABSTRACT

Explanatory note: 115 p., 34 fig., 1 tabl., 20 formulas, 1 ann., 52 sources.

DATA AUGMENTATION, DEEP LEARNING, GENERATIVE ADVERSARIAL NETWORK, LEARNING TRANSFER, PRETRAINING.

The paper compares the effectiveness of data augmentation and pretraining methods in generative adversarial models of image translation with limited data. Existing solutions are considered but they do not completely solve the problem of small amount of training data, which threatens to overfit the discriminator and the continious deterioration of the result. Successful results under such conditions are achieved by early stopping. A two-objective discriminator is proposed, which is expected to avoid the negative effects of overfitting the discriminator and the need for early stopping.

Training of generative adversarial networks is complicated by the heterogeneity of the characteristics of competing networks – generators and discriminators. It is especially difficult to organize the training process if the discriminator and generator have been previously trained (or learning has been transferred) to solve the problem of limited amount of training data. In this case, the discriminator from the beginning has significant advantages over the generator. This paper proposes an adaptive approach to solving the problem of matching the learning speed of the generator and the discriminator of generative adversarial networks. At each step of the training, a penalty coefficient is calculated, by which the gradients will be multiplied before their application to the weights of the discriminator. The penalty coefficient is calculated from the value of the discriminator's loss, which ensures the continious adaptation of the discriminator`s learning rate.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Colab – Google Colab https://colab.research.google.com;

CycleGAN – циклічно узгоджена генеративна змагальна мережа;

GAN – генеративна змагальна мережа;

GPU – (англ. graphics proccesing unit, GPU) окремий пристрій персонального комп'ютера або ігрової приставки, виконує графічний рендеринг. Сучасні графічні процесори дуже ефективно обробляють і зображують комп'ютерну графіку, завдяки спеціалізованій конвеєрній архітектурі вони набагато ефективніші в обробці графічної інформації, ніж типовий центральний процесор;

Kaggle – онлайн-платформа http://www.kaggle.com/ для змагань з аналітики та передбачувального моделювання, у рамках якого учасники конкурують у створенні найкращих моделей для прогнозування та опису даних;

TPU – Тензорний блок обробки (англ. tensor processing unit) – це інтегральна схема специфічного застосування (ASIC), призначена для прискорення розрахунків штучного інтелекту, була розроблена компанією Google спеціально для машинного навчання нейронних мереж.

ВСТУП

Машинне навчання любить дані, а саме величезну кількість даних. Машинне навчання з учителем любить величезну кількість розмічених даних, у яких із кожним прикладом поєднана правильна відповідь. А це титанічна, дорога і нудна людська праця. Тому в процесі вивчення глибинного навчання мене вразили не успіхи в галузі природньомовних текстів (GPT-3) або навчання з підкріпленням, як я очікував спочатку. Найбільший ефект справила технологія CycleGAN [1]: з двох наборів необроблених фотографій (коні і зебри) без будь-яких розміток місць, де знаходиться тіло коня або зебри, скільки коней на фото, нейромережа навчається перетворювати фото з кіньми у фото із зебрами. Або навпаки. Розмітка даних не потрібна, пари зображень не використовуються. На мою думку, розвиток таких технологій призведе до появи нових видів машинного навчання, які ближче до того, як вчиться людина, і не вимагають величезної роботи по розмітці даних.

Скільки відкривається можливостей для застосування технології. Супутниковий знімок місцевості перетворити в топографічний план згідно стандартів або навпаки трансформувати топоплан в знімок місцевості. Фотографію перетворити в аніме малюнок або навпаки оживити аніме малюнок у фотографію. Сегментація фотографій або відео. В інших сферах (обробка аудіо, NLP) можливі застосування для перетворення з одного стилю в інший. Якщо в галузі комп’ютерного зору буде можливий переклад художнього твору з одного стилю до іншого зі збереженням семантичного змісту без застосування пар зображень, то згодом ці технології можна буде застосувати до художнього перекладу текстів або навіть віршів, тобто те, що зараз вважається неможливим. І результат буде залежати не від стилів перекладачів, а від стилів поетів, що писали на мові, на яку перекладають.

Сама мережа, яка генерує результати вражаючої якості, є змагальною, де дві її складові частини – генератор і дискримінатор, одночасно

навчаються, змагаючись між собою: генератор намагається створити зображення, яке дискримінатор не відрізнить від реального, у той час, як дискримінатор вдосконалюється в експертизі, – відрізняти фальшиві зображення генератора від реальних зображень навчального набору даних. Результатом змагання є такі зображення, які перевершують поріг людської експертизи: людина не зможе відрізнити фейкове зображення, видане генератором, від реальної фотографії або картини художника.

Але ще один вид змагання також використовувався в роботі. Через місяць після того, як я вирішив займатися CycleGAN, на платформі Kaggle було запущено змагання «I'm Something of a Painter Myself. Use GANs to create art – will you be the next Monet?», у якому з 300 картин Моне і 7 тисяч фотографій природи нейромережа повинна навчитися перетворювати фотографії в картини Моне. Отже, усі мої ідеї пройдуть перевірку не експериментами з моєю ж «лінійкою» для їх оцінки, але зрівняються з досягненнями учасників змагання з усього світу. Задача вже не просто вразити якісним згенерованим зображенням, а показати кращий показник метрики згідно умов змагання Kaggle. Отже, оцінка роботи буде цілком об'єктивною – результат у таблиці лідерів змагання з більш ніж 200 учасників. Наприклад, в [50] використано задачу та рішення інших учасників цього ж самого змагання, але модель не може показати гарних результатів у змаганні, тому для отримання висновку замість порівняння з іншими учасниками проведено опитування серед студентів Стенфорду. Також Радбоуд університет Неймегена використовує це змагання при підготовці студентів на курсі «Машинне навчання на практиці».

У змаганні даний для прикладу базовий блокнот. У ньому організовано введення/виведення даних з урахуванням того, що нейромережу можна запускати на TPU – Тензорний блок обробки (англ. tensor processing unit) – це інтегральна схема специфічного застосування (ASIC), призначена для прискорення розрахунків штучного інтелекту, що була розроблена компанією Google спеціально для машинного навчання

нейронних мереж. Навіть за половинного завантаження можливо отримати прискорення в 3–16 разів порівняно з GPU. За baseline взята реалізація CycleGAN саме та, яка перетворює коней у зебр і реалізована за допомогою Tensorflow 2 та знаходиться на сайті Tensorflow в якості навчального прикладу. Отже, вихідна базова модель змагання саме та, яка описана в статті [1] «Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks».

Мета даної роботи – створити модель, яка покаже найкращий результат у змаганні Kaggle «Use GANs to create art» в умовах обмеженої кількості зображень. При цьому ще до початку дослідження в мене вже була робоча ідея, як я можу покращити модель CycleGAN. Вважав, що дискримінатори та генератори моделей для перекладу зображень без учителя потрібно попередньо навчати окремо на задачах класифікації (для дискримінатора) та відновлення зображення (для генератора). Перевагою буде відсутність такого гіперпараметру, як вибір початкової випадкової ініціалізації, а також це може допомогти у випадку обмеженої кількості даних для навчання.

Перевірка цієї ідеї та бажання показати найкращий результат у змаганні були основними причинами дослідження.

У ході експериментів виявився небажаний сторонній ефект попереднього навчання для генеративно-змагальних мереж. Дискримінатор та генератор за своєю будовою значно відрізняються, а в змагальному навчанні потребують узгодження швидкості навчання. Попереднє навчання додає ускладнень, оскільки навчаючись на задачі класифікації (відноситься зображення до цільової області чи до іншої) дискримінатор досить добре готовий до відповіді, чи справжнє зображення цільової області йому представлено. У той же час настільки ефективне попереднє навчання генератора організувати складно. Після попереднього навчання узгодити спільне змагальне навчання генератора та дискримінатора вдається не завжди. Виникла ідея адаптивно узгодити швидкість навчання генератора та

дискримінатора, застосовуючи штрафний коефіцієнт до градієнтів дискримінатора, який зележить від ступеня їх розбалансованості.

При розрахунку змагальних втрат як бінарної кросентропії значення цих змагальних втрат для дискримінатора у випадку «рівності ефективності» генератора та дискримінатора дорівнює логарифму з 0.5 – тобто 0,693. За таких умов середня оцінка навчальної партії справжніх та згенерованих зображень співпадає – дискримінатор не може їх відрізнити, але в той же час дає значущий зворотній зв'язок генератору, як йому покращитись. Зменшення значення змагальних втрат показує на перевагу ефективності дискримінатора та небезпеку руйнування процесу навчання.

Зважаючи на наявність такого індикатора ефективності навчання, я пропоную адаптивний підхід для керування швидкістю навчання дискримінатора, на кожному кроці навчання визначаючи штрафний коефіцієнт, на який буде помножено градієнти перед їх застосуванням до вагових коефіцієнтів. Розраховувати його пропоную як подвоєні втрати дискримінатора в степені три – при значенні втрат 0,5 штрафний коефіцієнт буде рівний одиниці, а при менших значеннях – різко спадати, зменшуючи швидкість навчання дискримінатора та даючи можливість генератору виправити ситуацію. За результатами досліджень при значеннях втрат від 0,5 до 0,693 впливати на швидкість навчання дискримінатора недоцільно.

При дослідженні методів навчання в умовах обмеженої кількості даних виявлено, що їх застосування в більшості випадків лише відстрочує перенавчання, і для отримання гарного результату потрібно вчасно достроково припинити навчання. Оскільки процес навчання генеративно-змагальних мереж займає значний час (від годин до місяців), а точна метрика може не завжди бути доступна (як і в даному змаганні), потреба в ранній зупинці є надзвичайно болючою проблемою. Підбір гіперпараметрів за таких умов дуже ускладнений, адже вони можуть змінити строк найкращої зупинки, і часто неможливо правильно оцінити ефект зміни гіперпараметрів. Пропоную для вирішення даної проблеми

використовувати двоцільовий дискримінатор. По суті – це два дискримінатори зі спільними, окрім останнього, шарами, які навчаються за допомогою двох різних способів обчислення змагальних втрат – наприклад, бінарної кросентропії (BCE loss) та завісних втрат (hinge loss). Крім того, один із дискримінаторів має виставляти вищі оцінки згенерованим зображенням, а інший – справжнім. Таким чином, спільні шари дискримінаторів будуть одночасно використовуватися для різних цільових функцій і не зможуть так легко перенавчитися, як єдиний дискримінатор чи ансамбль дискримінаторів. А останніх шарів – голів дискримінаторів, що не є спільними, може не вистачити для перенавчання. В експериментах така модель дискримінатора дає постійно досить гарні результати. Хоча вони і поступаються вдалій ранній зупинці аналогічної моделі з одним дискримінатором, але на відміну від неї отримуємо повторювані результати, що дає змогу підбирати особливості моделі та гіперпараметри і робити достовірні висновки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Технічні умови дослідження моделей перекладу зображень

Переклад зображень з однієї предметної області в іншу без використання пар відповідних зображень вперше був представлений у 2017 році в трьох публікаціях [1], [2], [3]. Щоб навчити модель перекладати зображення, не маючи правильної відповіді для кожного прикладу, зробили припущення, що перекладене з області X до області Y зображення можна реконструювати, переклавши знову з області Y до області X, і початкове та реконструйоване зображення мають співпадати. Тобто пару зображень зробили штучно, не маючи її у вихідних даних. І тепер штучна «ground truth» для кожного навчального зображення – це його ж подвійний переклад.

Наприклад для вхідного зображення коня генератор «кінь–зебра» створює зображення «зебра», з якого генератор «зебра–кінь» відновлює зображення «кінь (реконструйований)». Середня абсолютна різниця значень пікселів «кінь» та «кінь (реконструйований)» становить реконструкційні втрати (рисунок 1.1).

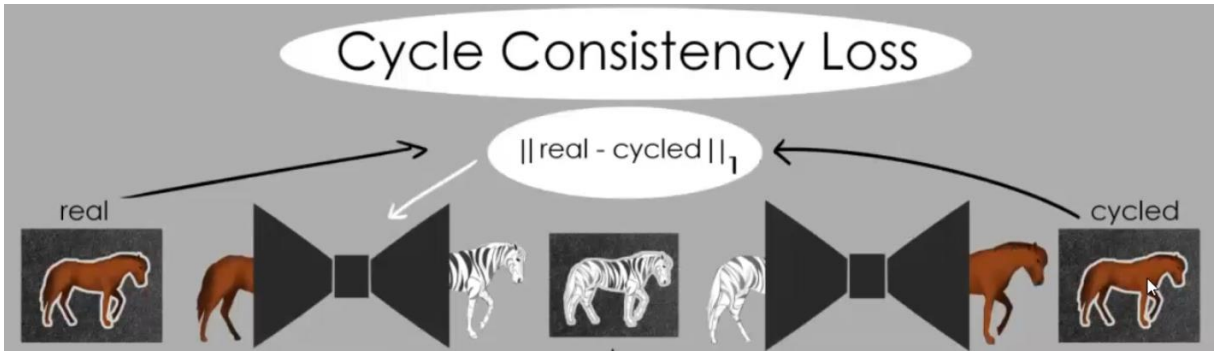

Рисунок 1.1 – Реконструкційні втрати

Оскільки генератори навчаються за допомогою дискримінаторів, а маємо вже два генератори, то потрібні й два дискримінатори. А оскільки

генератор Y–>X теж має навчатися за допомогою реконструкційної втрати, то її треба теж застосовувати двічі. У результаті отримуємо модель CycleGAN (рисунок 1.2). Два генератори F та G навчаються за допомогою змагальних втрат від дискримінаторів Dy та Dx і реконструкційних втрат. Змагальні втрати обчислюються на основі вхідних зображень та згенерованих, а реконструкційні – на основі вхідних та реконструйованих.

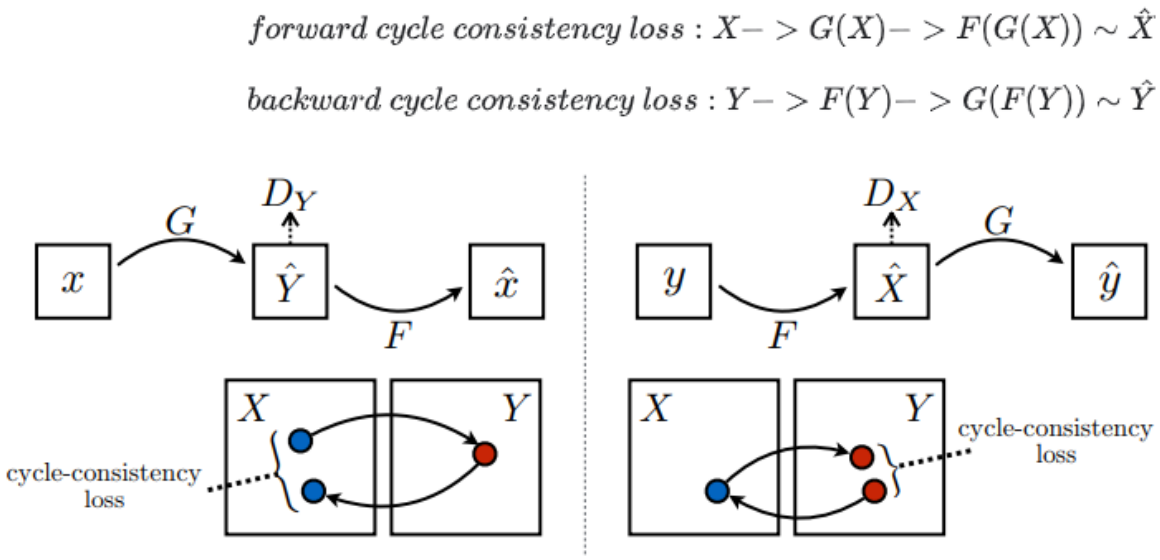

Рисунок 1.2 – Схема CycleGAN

Також можуть використовуватись ще одні втрати – втрати ідентичності (identity loss), суть їх полягає в тому, що коли генератору «кінь–зебра» на вхід подати зображення «зебра», то результат не повинен дуже відрізнятися від вхідного «зебра»:

$$\text{Identity loss} = |G(Y) - Y| + |G(X) - X| \quad (1.1)$$

Протягом наступних років після виходу публікацій [1], [2], [3] безпарний переклад зображень з однієї предметної області в іншу викликав надзвичайний інтерес науковців. Наукові статті публікуються у великій

кількості, при цьому в кожній із них успішність запропонованої моделі підтверджується порівнянням метрик із попередніми. Самі метрики для порівняння найчастіше взяті з задач генерації зображень за допомогою генеративно-змагальних мереж (GAN). Загальноприйнятого методу оцінки моделей перекладу даних на цей час спільнотою не прийнято, відповідно існує загроза, що дослідник підбирає для вимірювання своєї ідеї зручну «лінійку» (метрику), а відтворення результатів генеративно-змагальних моделей іншими дослідниками вимагає значних витрат часу та обчислювальних ресурсів. Іншим фактором, що ускладнює перевірку результатів досліджень, є вплив гіперпараметрів та початкової випадкової ініціалізації на результат роботи моделі. Тобто за однакових умов запущена кілька разів генеративно-змагальна модель може одного разу показати найкращий результат, на другий – гірший за порівнювані моделі, а третього разу градієнти можуть зникнути або вибухнути. Тому для проведення досліджень я скористався змагальною платформою Kaggle, на якій у серпні 2020 року було запущено постійне змагання «I’m Something of a Painter Myself. Use GANs to create art – will you be the next Monet?». Усі учасники в рівних умовах, і якщо вдасться показати кращий результат, ніж в інших учасників змагання, – це буде досить вагомим підтвердженням ефективності запропонованих ідей та правильності висновків.

Відповідно умови та правила змагання є важливим фактором дослідження і потребують опису. Роботи художників можна впізнати завдяки їх унікальному стилю, наприклад, вибору кольору або мазкам пензля. Таких митців, як Клод Моне, тепер можна імітувати за допомогою алгоритмів завдяки генеративним змагальним мережам (GAN). У цьому змаганні учасники додають його стиль до фотографій або створюють зображення в його стилі з нуля. За останні роки комп’ютерний зір надзвичайно просунувся, і GAN тепер здатні дуже переконливо імітувати об’єкти. Але створення шедеврів, гідних музеїв, вважається більше мистецтвом, ніж наукою. Тож чи може наука про дані у формі GAN

обдурити класифікатори, які повірять, що за допомогою GAN створили справжнього Моне? Це виклик, який приймають учасники змагання.

GAN складається щонайменше з двох нейронних мереж: моделі генератора та моделі дискримінатора. Генератор – це нейронна мережа, яка створює зображення. Для даного конкурсу генератор повинен створити зображення в стилі Моне. Цей генератор має бути навчений за допомогою дискримінатора. Дві моделі працюватимуть одна проти одної, генератор намагатиметься обдурити дискримінатор, а дискримінатор намагатиметься точно класифікувати реальні та генеровані зображення. Завдання учасників – створити GAN, яка генерує від 7000 до 10000 зображень у стилі Моне.

Зображення учасників оцінюються за метрикою Memorization-Informed Frechet Inception Distance (MiFID), що є модифікацією від Fréchet Inception Distance (FID). Memorization-Informed – означає перевірку, яка частка з поданих зображень великою мірою схожа з іншими справжніми картинами Моне – на випадок, коли учасник недобросовісно подав модифіковані картини Моне замість згенерованих, або ж сама модель допускає «протікання» зображень із навчального набору даних до генерованих зображень. Якщо частка перевищує якийсь ліміт (наприклад, 10%) – загальна метрика дуже зростає (штраф). Інакше результатом буде просто обчислення за метрикою FID. Чим менша MiFID, тим кращі зображення.

FID, як і Inception Score, часто використовуються в останніх публікаціях як стандарт для методів оцінки GAN. У FID використовується попередньо навчена мережа Inception для отримання активацій із проміжних шарів. Потім моделюється розподіл даних для цих ознак за допомогою багатовимірного розподілу Гауса з середнім значенням μ та коваріацією Σ. FID між реальними зображеннями та генерованими зображеннями обчислюється:

$$\mathrm{FID} = ||\mathrm{mu\_r} - \mathrm{mu\_g}||^2 + \mathrm{Trace}\,(\mathrm{Sigma\_r} + \mathrm{Sigma\_g} - 2 \times \sqrt{(\mathrm{Sigma\_r} \backslash \mathrm{Sigma\_g})}), \qquad (1.2)$$

де Trace підсумовує всі діагональні елементи, FID обчислює відстань Фреше між двома розподілами Гауса, отриманими з активацій проміжних шарів мережі Inception.

Учаснику потрібно створити 7 000–10 000 зображень у стилі Моне у форматі jpg. Їх розміри повинні бути 256x256x3. Потім потрібно заархівувати ці зображення, і результат повинен мати вихідний файл із назвою images.zip. Цей файл має бути результатом роботи скріпта або ноутбука в середовищі Kaggle з обмеженням в максимум 5 годин роботи на GPU або TPU.

Учасники мають дотримуватися кодексу честі: конкуренти повинні використовувати генеративні методи для створення зображень, а не подавати безпосередньо зображення картин Моне або змінені версії таких зображень. Це псує початкову мету змагання, якою є навчання генеративним моделям перекладу зображень та освоєння машинного навчання з використанням TPU.

Цей конкурс триває постійно, а таблиця лідерів рухома, тобто результати стають не дійсними і видаляються через два місяці. Призи за перемогу відсутні.

Вхідні дані складаються з:

– 300 картин Моне розміром 256x256 в TFRecord форматі

– 7028 фотографій ландшафту розміром 256x256 в TFRecord форматі.

Організатори змагання настійно рекомендують базовий код моделі (Monet CycleGAN Tutorial), який розповідає про основи завантаження даних із TFRecords, використання TPU та побудову CycleGAN. Реалізація CycleGAN за допомогою Tensorflow 2 в ньому саме та, яка перетворює коней у зебр, і опублікована на сайті Tensorflow в якості навчального прикладу. Скопіювавши даний ноутбук, можна негайно приступити до дослідження (підбирати гіперпараметри і т.п.) У той же час згідно правил зображення в стилі Моне можна створити з нуля, використовуючи інші

архітектури GAN, такі як DCGAN. Надіслані файли зображень не обов'язково повинні бути перетвореними фотографіями.

1.2 Попередні дослідження архітектури CycleGAN

Звичайно, найпершою ідеєю було «осучаснити» представлену базову модель, використавши нові функції активації, наприклад, адаптивну функцию активації AdPReLU [4], параметри якої налаштовуються в процесі навчання. Також замінити MSE/MAE на Huber(0.5), на мою думку, саме в такий спосіб більш адекватно оцінювати відстань між реальним та генерованим зображеннями, адже незначна різниця кольорів менше 0.5 буде пом’якшена квадратом без шкідливого впливу на більші різниці, як в MSE. Найбільш упевнений я був у тому, що необхідно видалити частину втрат генератора, пов’язану з необхідністю ідентичності результату у випадку подачі справжньої картини Моне генератору картин Моне (identity loss). Щодо цього питання я опублікував ноутбук на платформі Kaggle «CycleGAN without identity loss». Більшість спроб покращення моделі не принесли результату, висновок щодо них – Fail. Перечислю лише основні з них:

– проблема identity loss – складової функції втрат генераторів CycleGAN. Якщо на вхід подаємо фото коней разом із зебрами, чекаючи на виході зебр (або фото чоловіків і жінок, чекаючи фото тільки жінок), то логічним результатом є врахування ідентичності. І якщо подати на вхід фото одних зебр (одних жінок), – повинні отримати на виході фото бажано без змін – ідентичне вхідному. Що в базовій CycleGAN і реалізовано. Але якщо зміни істотні (наприклад, робимо топографічний план із супутникових знімків або мультфільм із фотографій), то подача картин Моне (або топографічного плану) на вхід замість фото (або супутникових знімків) не має сенсу. Дана складова функції втрат є опціональною – іноді вона працює, інколи позитивного ефекту від неї немає. Я припустив, що немає чого

«мучити» CycleGAN зайвою складовою функції втрат. Провів велику кількість експериментів типу ablation study, де в працюючої CycleGAN моделі видаляв identity loss, але виявилося, що для перетворення «фотографії–Моне» наявність identity loss все-таки робить змагальне навчання «більш гладким». І, незважаючи на некоректність самої ідеї, допомагає зберегти колірну палітру при перетворенні. У разі відсутності цієї складової кольору перетворювалися в дуже незвичайні;

– замінити середню абсолютну похибку на середньоквадратичну або функцію втрат Юбера (Huber loss). Я спробував використати середньоквадратичну помилку (MSE) для відновленого зображення (замість MAE у базовому блокноті). Якщо різниця кольорів між точкою реконструйованого малюнка не значна, це неможливо розрізнити для людського ока. Квадрат значення похибки покарає великі відхилення та зменшить значимість малих відхилень. У той же час недолік MSE – нетерпимість викидів у навчальних даних – не застосовується до нашої задачі, оскільки навчальні дані не містять правильних «ground truth» значень. Виявилось, що для CycleGAN це погіршує результат. Хороший переклад може означати деяку втрачену інформацію, яку неможливо відновити. Відповідно квадратичне покарання за це – помилка. Змінено на функцію Юбера. Але в попередніх дослідженнях неможливо було довести переваги використання функції Юбера в порівнянні з середньою абсолютною похибкою. Дане питання потребує додаткових досліджень;

– оскільки виявилося, що навчання нестабільне і при збільшенні кількості епох з якогось моменту результат починає погіршуватись замість покращення, вирішив спробувати технологію ранньої зупинки навчання (early stopping). Для цього потрібно мати метрику, яка співпадає з метрикою змагання. Хоча в описі змагання вказано, яка саме версія моделі Inception використовується для обрахування метрики FID, але, перебравши всі внутрішні шари та їх середні та максимальні pooling, не вдалося знайти відповідність фактичній метриці змагання. Також не знайдено відповідний

шар і серед інших варіантів Inception, Xception чи VGG. У той же час, якщо обрахувати метрику FID по виходу будь-якого внутрішнього шару InceptionV3, то це значення можна використати для грубої оцінки моделі. Однак по такій грубій оцінці неможливо успішно впровадити early stopping;

– реконструйоване зображення рахувати як ще одне фейкове, і оцінку його дискримінатором використати для навчання відповідного генератора та дискримінатора. У попередньому дослідженні позитивного ефекту не отримано;

– застосування Wasserstein loss [7] замість BCE loss. Це не призвело до покращення метрики змагання. Крім того, при навчанні CycleGAN mode collapse поки що не зустрічався, тому сумнівна потреба застосування Wasserstein loss;

– замість Wasserstein loss чи BCE loss використав Least square error – похибку найменших квадратів [38]. Ця зміна відразу призвела до отримання стабільних та гарних показників метрики змагання. Сумнівів у доцільності її подальшого застосування немає.

Отже, з усіх запропонованих змін позитивний ефект отримано лише від переходу до похибки найменших квадратів (LSE) [38] та розроблено свою оцінку метрики FID, яка, на жаль, відрізняється від метрики змагання. Жодні інші зміни, у тому числі до функцій активації чи гіперпараметрів навчання, не призвели до стабільного позитивного ефекту, але вдалося виявити причини цього. Про них буде йти мова в наступних двох підрозділах.

1.3 Виявлення проблеми перенавчання дискримінатора

Оскільки процес навчання виявився нестабільним, і збільшення кількості епох не допомагає, то така ситуація схожа на перенавчання. Для ідентифікації проблеми я виділив частину навчального набору фотографій

(100 штук) до валідаційного набору, з'єднавши їх з 1367 картинами Моне (навчальний набір складається лише з 300 картин Моне). Додав до моделі метод test_step, метою якого є лише відображення значень дискримінаторів на валідаційному наборі даних після кожної епохи навчання. У результаті після кожної епохи я можу отримати інформацію, як саме дискримінатори оцінюють справжні фотографії та картини Моне, яких вони не бачили при навчанні – справжніми чи фейками, та порівняти цю оцінку з відповідною на навчальному наборі даних.

При цьому слід врахувати, що для навчання можливо спробувати «безпечну» аугментацію даних – випадкове дзеркальне відображення праворуч-ліво, що вдвічі може збільшити кількість початкових даних. Інші види аугментації погіршують результати моделі, оскільки генератор починає видавати зображення не з початкового розподілу картин Моне, а з аугментованого розподілу, що віддаляє від цілі та погіршує результат метрики змагання. Таким чином можна спробувати збільшити кількість картин Моне вдвічі (до 600) та фотографій (з урахуванням 100 шт. для валідації залишається 6938 фотографій) 6938 х 2 = 13876.

При навчанні моделі з batch_size = 1 та подвоєною дзеркальною аугментацією кількістю зображень виявлено, що до одинадцятої епохи оцінки дискримінаторами справжніх валідаційних зображень близькі до оцінок справжніх зображень навчального набору і коливаються від 0,48 до 0,57 (у моделі при змаганні генератора та дискримінатора вже використовується LSE похибка найменших квадратів замість загальноприйнятої BCE бінарної кросентропії, тому значення коливаються навколо 0,5, а не навколо 0,69). Але, починаючи з одинадцятої епохи, дискримінатор картин Моне починає різко знижувати оцінку валідаційних картин Моне – з 0,5 до 0,35 за наступні 10 епох. У подальшому коливається від 0 до 0,35 (рисунок 1.3).

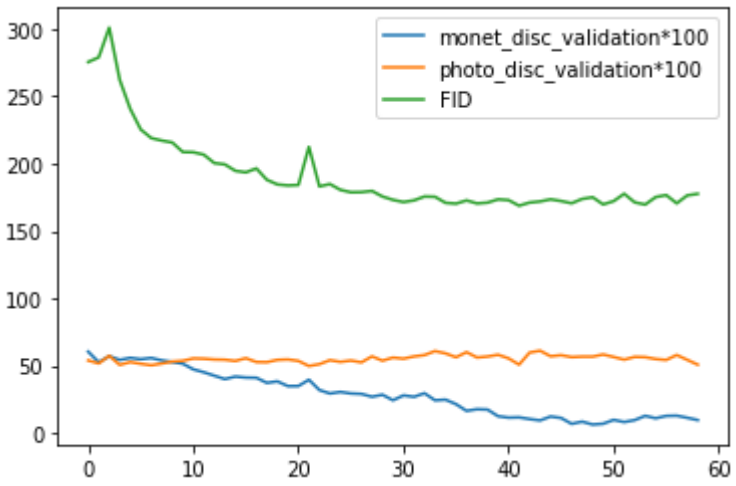

Рисунок 1.3 – Значення оцінки справжніх картин Моне валідаційного набору даних дискримінатором починає знижатися після десятої епохи

У той же час оцінка справжніх картин Моне навчального набору даних, починаючи з одинадцятої епохи, становить близько 0,65 – 0,7, коли згенеровані зображення оцінюються в 0,3 – 0,35. Це свідчить про перенавчання дискримінатора картин Моне на навчальному наборі – він уже бачив усі 600 картин 30 разів і точно їх запам’ятав. Непоказані картини Моне валідаційного набору він упевнено вважає за фейкові, як і переважну більшість згенерованих зображень. У такому разі цінність його оцінки для навчання генератора відсутня – замість експертної оцінки, як покращити зображення, дискримінатор надає інформацію, що такого зображення не запам’ятав серед справжніх. Метрика FID, починаючи з 40-ї епохи, замість зниження починає зростати, результати стають гіршими. Таким чином, експериментально встановлено причину, чому навчання не стійке, і збільшення кількості епох призводить до погіршення результату.

Коли замість подвоєної дзеркальною аугментацією кількості картин Моне для навчання, подати лише 300 оригінальних картин, ситуація погіршується ще швидше. Ознаки перенавчання бачимо вже після восьмої епохи (рисунок 1.4). Щодо результатів дискримінатора фотографій, то як у випадку подвоєної кількості, так і коли фотографій близько 6000, ознак

перенавчання немає. Справжні фотографії навчального та валідаційного набору оцінюються в 0,55 – 0,6 у той час, як згенеровані в 0,4 – 0,45.

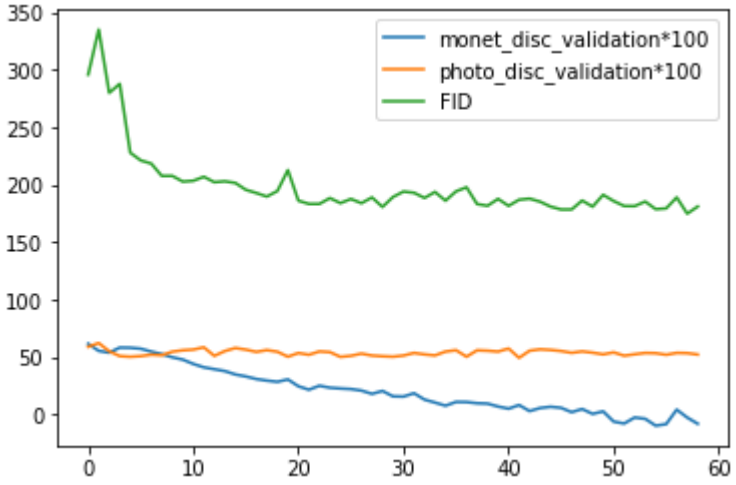

Рисунок 1.4 – Значення оцінки валідаційного набору даних дискримінатором для 300 справжніх картин Моне починає знижуватися приблизно з восьмої епохи

Для визначення меж перенавчання збільшено batch_size до 32, щоб у 32 рази збільшити «бачуваність» зображень при навчанні, та перевірено наявність перенавчання для різної кількості картин Моне. Встановлено, що навіть при 1024 х 2 картинах Моне та фотографіях протягом чотирьох годин навчання на TPU дискримінатор перенавчається (рисунок 1.5).

У ході даних експериментів було виявлено, як ще впливає мала кількість даних на результат моделі. Коли використовувалися 300 картин Моне, навчений генератор видавав фотографії значно нижчої якості, ніж коли 1250 картин Моне використовувалося для навчання. Хоча перенавчений дискримінатор картин Моне не впливає на генератор фотографій, але різноманіття вхідних даних впливає на якість генерації навіть у тій області зображень, у якій достатня кількість. Тобто в умовах, коли недостатня кількість зображень А, а достатня Б, при навчанні CycleGAN погіршуються генеровані зображення як А, так і Б. Крім того,

навіть за відсутності перенавчання дискримінатора А, якість генерованих зображень Б залежить від кількості зображень А: чим різноманітніші зображення А подаються генератору Б при навчанні, тим кращий результат.

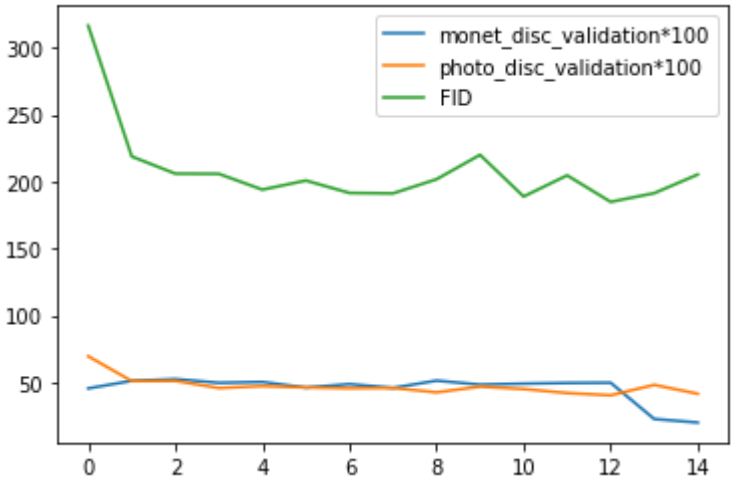

Рисунок 1.5 – При 1024 x 2 = 2048 зображень дискримінатор перенавчається після трьох годин навчання з TPU

Як підсумок, слід мати на увазі, що відсутність перенавчання дискримінатора не є достатнім фактором ефективної роботи моделі перекладу зображень. Різноманітність навчальних даних є умовою ефективного навчання генераторів. Обмежена кількість даних суттєво впливає на ефективність моделі (рисунок 1.6).

1.4 Виявлення проблеми кодування зображення при перекладі

При експериментах із функціями втрат випадково допустив помилку, що призвела до завдання генератору фотографій видавати зображення, мінімально схожі на фотографії. Сірі квадрати виявилися результатом. Але вражаючим відкриттям було те, як на це відреагувала CycleGAN із реконструкцією зображення (Cycle consistency loss). Виявилося, що з цих

сірих квадратів генератор картин Моне може надзвичайно точно відновити початкову картину Моне (рисунок 1.7).

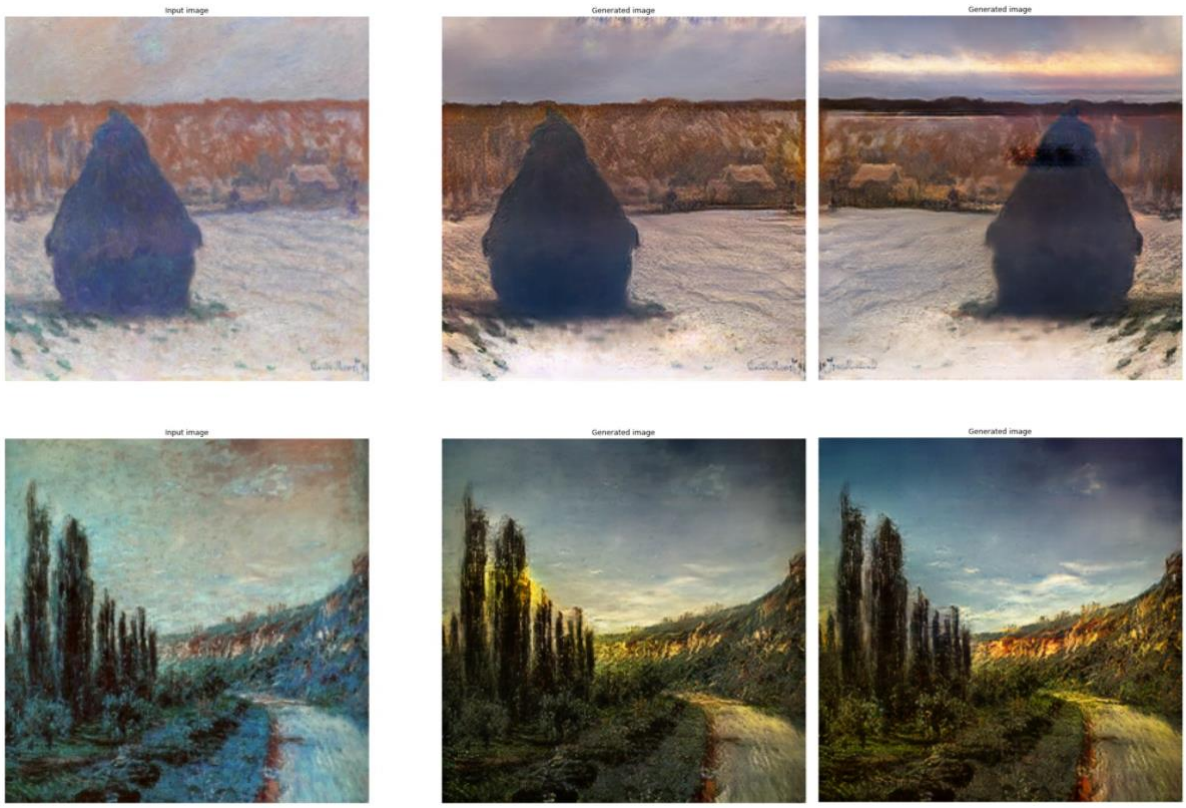

Рисунок 1.6 – Якість фотографій залежить від різноманітності картин Моне в навчальному наборі. Зліва – картини Моне, посередині – згенеровані фотографії при 1250 картинах Моне навчального набору, справа – згенеровані при 1350 x 2 картин Моне навчального набору

Цей факт кодування початкового зображення генератором із метою подальшої реконструкції зображення іншим генератором повністю змінює відношення до цінності Cycle consistency loss та моделі CycleGAN в цілому. Виявляється, замість того, щоб здобути знання щодо перетворення зображень однієї області в іншу, модель користується «шпаргалкою», закодовуючи початкове зображення і відновлюючи його детально для отримання гарних показників Cycle consistency loss.

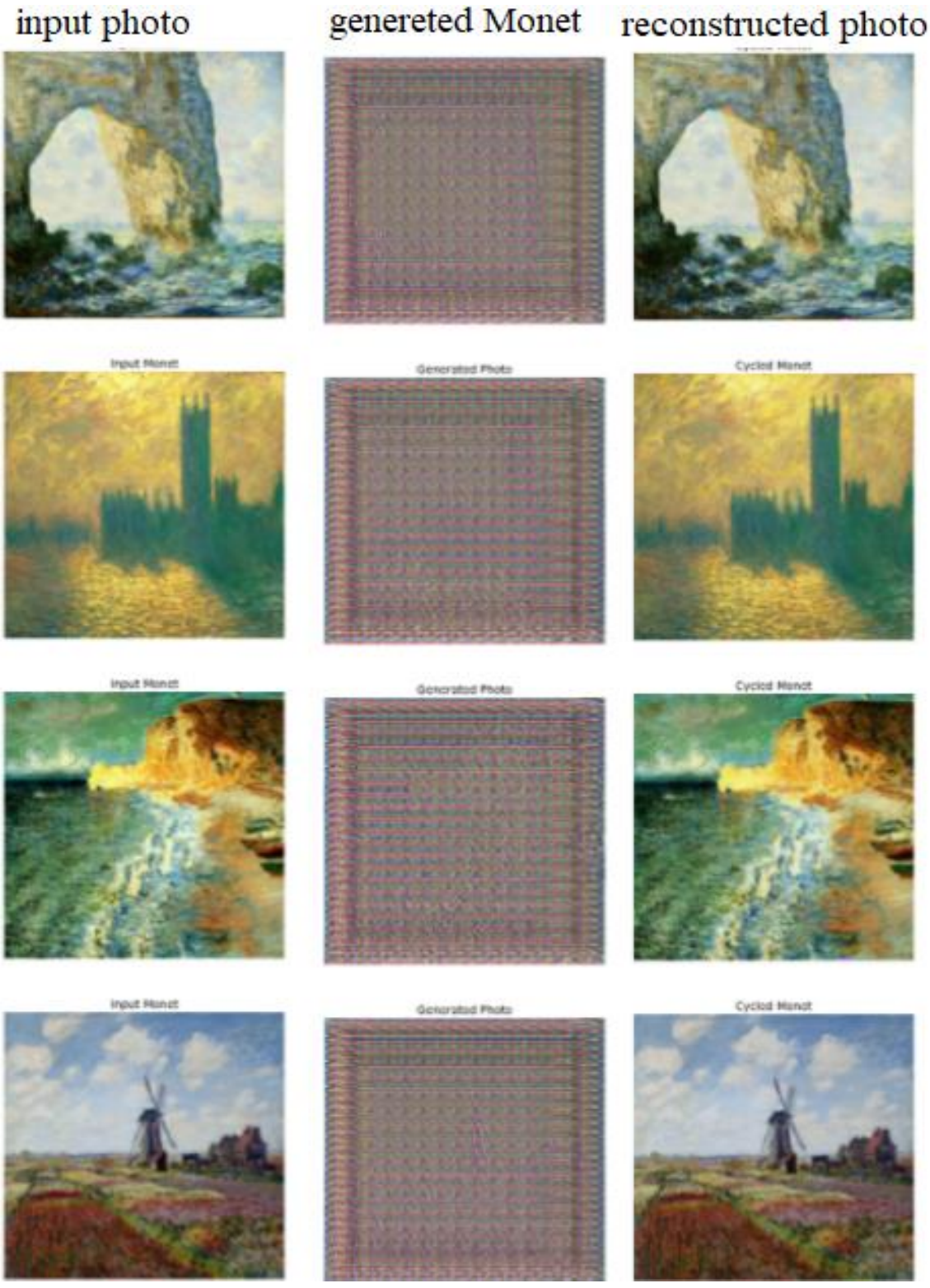

Рисунок 1.7 – Зліва – початкові картини Моне, посередині – згенеровані зіпсованим генератором фотографій зображення, справа – відновлені генератором картин Моне з «сірих квадратів», що демонструє факт повного кодування початкових зображень в згенерованих для їх наступної реконструкції

У дійсності, розбіжності з поставленою перед моделлю задачею немає. Відновлене зображення не перевіряється на приналежність до типу вхідного чи за семантичним змістом, а лише повинно мінімально відрізнятися від нього. Щоб навчити модель перекладати зображення, не маючи правильної відповіді для кожного прикладу, зробили припущення, що перекладене з області А до області Б зображення можна реконструювати, переклавши знову з області Б до області А, і початкове та реконструйоване зображення мають співпадати. Але це припущення не правильне. Деякі дані втрачено. Якщо перетворюємо зебр на коней, ми втрачаємо смуги, і модель не може відновити смуги в правильних місцях. Якщо генератор перетворить коней назад на зебр, він може розмістити смуги в абсолютно неправильних місцях. Як це оцінять тоді реконструкційні втрати? Дуже погана зебра! На місцях білих смуг маємо чорні і навпаки. Найгірший варіант оцінки. Що робитиме модель наступного разу? Вона буде кодувати смуги на зображенні «коней» для відновлення пізніше. Відповідно модель не спонукається вивчити, що таке кінь і зебра, оскільки все одно не вгадає, як правильно розмістити смуги на зебрі.

Але це лише одна частина проблеми Cycle consistency loss. Що стосується перекладу зображення в зображення, ми хочемо, щоб модель була точною. Дерево потрібно перекласти як дерево. Уся семантика повинна бути збережена. Але потреба в точній реконструкції не мотивує до цього. Єдиною мотивацією є кодування оригінального зображення в перекладене зображення. Таким чином, Cycle consistency loss не додає ні точності перекладу, ні якості зображення. Це означає, що інші моделі перекладу даних, які звичайно навчаються перекладу в одну сторону, заслуговують на особливу увагу, оскільки вони позбавлені проблеми з реконструкцією CycleGAN.

1.5 Постановка задачі дослідження

Об’єктом дослідження є генеративно-змагальні моделі перекладу зображень без учителя в змаганні на платформі Kaggle «Use GANs to create art» при генерації картин Моне за умови, що для дискримінатора моделі доступні лише триста справжніх картин Моне. Для порівняння ефективності моделі маємо результати інших учасників змагання Kaggle.

При попередніх дослідженнях встановлено, що внесення змін до базової моделі CycleGAN як в частині підбору гіперпараметрів, так і заміни функцій активації та цільових функцій, не призводить до позитивних ефектів, оскільки:

– навчання нестабільне, більша кількість епох не означає кращий показник метрики;

– вдалося довести факт перенавчання дискримінатора картин Моне: інші справжні картини Моне навчений дискримінатор впевнено вважає «фейками»;

– при використанні реконструкції модель CycleGAN закодовує в згенерованому зображенні початкове для подальшої точної реконструкції, що ставить під сумнів доцільність використання реконструкції.

Відповідно пріоритетною проблемою є мала кількість зображень предметної області – в даному випадку маємо лише 300 картин Моне. Другим питанням є доцільність та умови використання реконструкції (Cycle consistency loss). Лише після їх вирішення результати можна буде покращити, застосовуючи попереднє навчання чи нові методи навчання. І наостанку можна буде розглянути проблему великої кількості гіперпараметрів для забезпечення стійкості процесу навчання чи адаптації моделі перекладу зображень до особливостей даних без їх підбору.

Метою дослідження є представити модель, яка покаже найкращий результат в змаганні Kaggle «Use GANs to create art» в умовах обмеженої кількості зображень.

Для досягнення мети можна виділити такі задачі:

– дослідити вплив попереднього навчання та аугментації даних на результати генеративно-змагальних моделей в умовах обмеженої кількості зображень;

– отримані технології використати в найсучасніших моделях перекладу зображень, зробивши їх придатними до роботи в умовах обмеженої кількості зображень. Результат роботи модифікованих моделей перевірити за допомогою даного змагання Kaggle;

– запропонувати покращення існуючих способів попереднього навчання та аугментації даних.

2 АНАЛІЗ ІСНУЮЧИХ МОДЕЛЕЙ, ФОРМУЛЮВАННЯ РОБОЧИХ ГІПОТЕЗ ДОСЛІДЖЕНЬ

2.1 Супутні роботи

Переклад зображення в зображення без використання пар розмічених даних були запропоновані в 2017 році [1], [2], [3]. Я знайшов два різні підходи:

– UNIT [2] (мережі перекладу без учителя зображення однієї області в зображення іншої, англ. Unsupervised Image-to-Image Translation Networks) на основі варіаційних автоенкодерів (VAE) для вивчення спільного для областей розподілу зображень;

– CycleGAN [1] та DiscoGAN [3] зберігають ключові атрибути між вхідним та перекладеним зображенням, використовуючи реконструкційні втрати (cycle consistency loss).

Удосконалення, запропоновані в [5], (StarGAN перекладає зображення між кількома областями і використовує класифікатор області як частину повної цільової функції), [6] (додавання шуму для захисту від змагальної самоатаки), [7] стабільність навчання з використанням втрати Вассерштейна, [8] пропонує багатоцикловий перекладу із взаємними інформаційними обмеженнями – MCMI), [9] HarmonicGAN додає додаткове згладжування до CycleGAN, [10] Lipschitz-регуляризований CycleGAN для поліпшення семантичної стійкості в перекладі зображень без використання пар зображень.

Щоб уникнути проблем, пов'язаних із циклічними втратами, запропоновано DistanceGAN [11], геометричну генеративну змагальну мережу GcGAN [12] та контрастивне навчання для безпарного перекладу зображення на зображення CUT [13]. У [14] пропонують вивчити спільний латентний простір сіамською мережею додатково до змагальної мережі без застосування реконструкційної втрати. CerfGAN [15] потребує лише двох

мереж (декодер та багатокласовий дискримінатор, який працює також як енкодер) для вирішення проблем перекладу зображення в зображення, але він використовує реконструкційні втрати, як це робить CycleGAN.

Проблема випадкової ініціалізації була зазначена в [16]: «Стабільність навчання: підходи до адаптації областей, які покладаються на певну форму змагального навчання, чутливі до випадкової ініціалізації. Для вирішення цієї проблеми ми враховуємо специфічні для завдання втрати, підготовлені як для вихідних, так і для сформованих зображень, та регуляризацію схожості пікселів, що дозволяє нам уникнути mode collapse та стабілізувати навчання. Використовуючи ці інструменти, ми можемо зменшити дисперсію продуктивності для тих самих гіперпараметрів у різних випадкових ініціалізаціях нашої моделі».

Передача навчання (learning transfer) для перекладу зображення в зображення було відзначено лише в [17]. Вони представляють модель DeepI2I, але вона отримує гірші результати на обмежених наборах даних. Для пом'якшення цієї проблеми пропонують ініціалізувати deepI2I із попередньо навченого GAN. Тож вони першими вивчають передачу навчання для мереж I2I. Іншою важливою функцією deepI2I є те, що модель не використовує реконструкційні втрати на рівні пікселів. Натомість вона порівнює результати прихованих шарів дискримінатора. Передача навчання в GAN описана у [18]. Дослідники прийшли до висновку, що попередня підготовка лише дискримінаторів покращує результати як у показниках FID, так і в IW, тоді як попередня підготовка лише генераторів шкідлива. Тож найкращий результат отримують завдяки попередній підготовці як генераторів, так і дискримінаторів. Вони досліджували донавчання (finetuning) попередньо навчених GAN, що призвело до поліпшення продуктивності для областей з обмеженою кількістю даних. Однак цей метод страждає від mode collapse та перенавчання, оскільки він оновлює всі параметри генератора для адаптації до цільової області. [19] (MineGAN)

першими розглядають можливість передачі знань із декількох GAN в одну цільову область.

Навчання генеративних змагальних мереж з обмеженими даними: [20], [33], [34] говорить, що ключовою проблемою малих наборів даних є те, що дискримінатор перенавчаєтья на навчальних прикладах; його зворотний зв'язок із генератором стає безглуздим, і навчання починає розходитися. Аугментація не є гарною ідеєю для GAN. GAN, яка навчається за допомогою аугментації даних, навчається генерувати аугментований розподіл. Автори демонструють, як використовувати широкий спектр аугментації, щоб запобігти перенавчанню дискримінатора, одночасно гарантуючи, що жодна аугментація не перетече на створені зображення. [17], [18], [19] використовують попереднє навчання для боротьби з проблемою обмеженості даних. Роботи [19], [22] показали перші перспективні результати з високою роздільною здатністю на складних природних зображеннях, враховуючи недавній успіх у високоякісному навчанні GAN. Обидві починаються з тонкої настройки попередньо навченої моделі GAN і додають додаткові параметри в деякі частини оригінальної мережі для навчання. [17], [18], [22] пропонують донавчання на обмеженій кількості даних моделй, які були попередньо навчені на великому наборі даних, [23] пропонують навчання з нуля на обмежених даних, [19], [24] використовують передачу знань між різними, але пов’язаними між собою областями.

2.2 Сучасні варіанти архітектури CycleGAN

Одними з перших проблема реконструкційної втрати описана в статті [6] «Adversarial Self-Defense for Cycle-Consistent GANs». Незважаючи на успіх циклічних GAN, вони мають великий недолік. Реконструкційні втрати примушують генераторну мережу, щоб приховати (закодувати) інформацію, необхідну для достовірної реконструкції вхідного зображення

всередині крихітних збурень перекладеного зображення, які не виявляються людським оком або дискримінатором. Проблема особливо гостра при перекладі «багато до одного», наприклад, фотографії на семантичні сегменти, де модель повинна реконструювати текстури та кольори втрачені під час перекладу в цільову область. У цій роботі показують, як така самоатакуюча поведінка при навчанні без учителя впливає ефективність перекладу, і запропонували два прийоми захисту. Одним, більш ефективним із них, є додавання шуму до генерованого зображення перед реконструкцією. Такий метод захисту має істотно зменшити кількість самоатакуючого структурованого шуму і, таким чином, зробити відображення більш залежним від вхідного зображення, що призводить до більш зрозумілого перекладу й реконструкції та підвищення якості перекладу.

Даний спосіб покращення та захисту CycleGAN від кодування вхідного зображення в генеровані досить легко додати до моделі – всього одного рядка коду потрібно для генерації шуму в методі training_step:

```
noise = tf.random.normal(shape = (256,256,3), mean = 0.0,
stddev = 0.02, dtype = tf.float32)                    (2.1)
```

Далі noise додаємо до згенерованого зображення перед подачею до зворотнього генератора. У моделях, що попередньо досліджувалися, провів достатню кількість експериментів та отримав наступні результати. По-перше, у переважній більшості випадків метрика змагання дійсно покращувалася при додаванні шуму. По-друге, раніше невидиме кодування почало ставати видимим (риснок 2.1), тобто мотивація до кодування настільки сильна в CycleGAN, що генератору вдається сховати від дискримінатора в таких збуреннях, які «невидимі» дискримінатору, але вже видимі людському оку.

І при подальшому збільшенні амплітуди шуму на ці збурення вже реагує метрика FID, погіршуючи оцінку якості зображення.

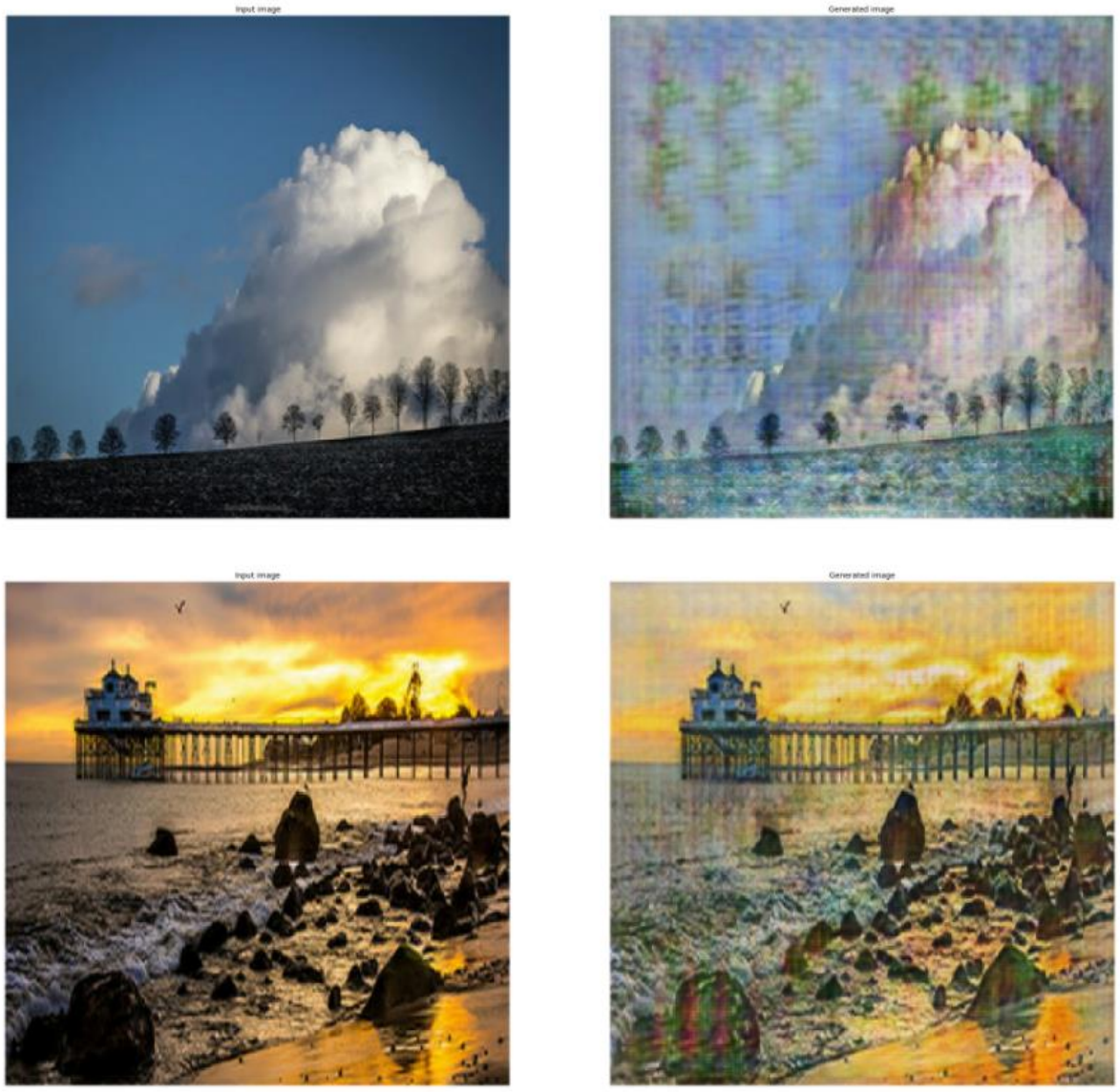

Рисунок 2.1 – При додаванні в згенероване зображення шуму перед подачею в зворотній генератор кодування вхідного зображення не припиняється, але іноді стає настільки грубим, що видно людському оку. Зліва – вхідні фотографії, праворуч – згенеровані картини Моне, у яких генератор для можливості точного відновлення фотографії по картині закодував особливості вхідної фотографії в регулярних штрихах

Причиною незначного покращення метрики FID при застосуванні шуму може бути просто вплив «м'якої» аугментації, що дещо відтерміновує перенавчання дискримінатора в умовах обмеженої кількості картин Моне. Тому від подальшого застосування шуму я вирішив відмовитися, оскільки він не вирішує проблеми реконструкційної втрати та може лише ускладнити дослідження з обмеженою кількістю даних. До того ж, протягом наступних півтора років після виходу статті [6] інші дослідники не застосували даний підхід у своїх моделях, пропонуючи інші шляхи вирішення проблеми.

У [7] запропонували алгоритм навчання GAN альтернативний традиційному – Wasserstein GAN. Він може покращити стабільність навчання, позбутися таких проблем, як collapse mode та надати значущі криві навчання, корисні для пошуку гіперпараметрів. Ця робота була великим прогресом у навчанні звичайних GAN, і я вирішив застосувати її версію зі штрафом градієнтів – WGAN-GP – для CycleGAN. Виявилося, що в умовах обмеженої кількості даних WGAN-GP не вирішує проблему реконструкційної втрати, покращення стабільності навчання теж не помічено. Але метрика змагання навіть гірша, ніж при застосуванні двійкової кросентропії. Основна перевага обрахування змагальних втрат з WGAN-GP – боротьба з collapse mode – виявилась непотрібною в CycleGAN, оскільки за базової архітектури collapse mode ніколи не спостерігався. Тому я припинив експерименти з WGAN-GP на користь випробуваного багатьма дослідниками обчислення змагальної втрати за найменшими квадратами – Least Square Error (LSE). У той же час у випадку застосування односторонніх моделей перекладу зображень без реконструкційних втрат або моделей генерації зображень без перекладу не виключаю потреби запровадження WGAN-GP.

Harmonic Unpaired Image-to-image Translation (HarmonicGAN) [9] – вводять плавну регуляризацію графу для безпарного перекладу зображення в зображення для досягнення гармонійних перекладів. Але основною метою цієї регуляризації є підвищення точності перекладу, а метрика нашого

змагання не перевіряє точність перекладу фотографій у картини Моне, а лише те, що згенеровані зображення мають стиль картин Моне. Тому змоги об’єктивно оцінити покращення немає, а проблему реконструкційної втрати, як і проблему малої кількості даних, HarmonicGAN не вирішує.

Так само в [10] Lipschitz-регуляризований CycleGAN для поліпшення семантичної стійкості в перекладі зображень без використання пар зображень, призначений для забезпечення семантиної точності перекладу. Його застосування можливе лише після вирішення питань щодо кодування даних у CycleGAN та малої кількості даних. І при цьому не можемо передбачити, чи вплине позитивно дане покращення на метрику змагання, але точно знаємо, що метрика змагання не оцінить покращення семантичної точності.

Для покращення якості генерації зображень за допомогою механізму уваги запропонована модель U-GAT-IT [41] (рисунок 2.2). Якщо x ∈ {Xs, Xt} – зразок із вхідної та цільової області, генератор моделі перекладу $G_{s \to t}$ складається з енкодера Es, декодера Gt та допоміжного класифікатора ηs, де ηs (x) являє собою ймовірність того, що x походить від Xs. $E^{\kappa}s(x)$ – k-та карта активації енкодера і $E^{kij}s$ (x) – значення в (i, j). Використовуючи class activation maps (CAM) [42], допоміжний класифікатор навчається вивчати ваги k-ї карти активацій для вхідної області, $w^{k}s$, використовуючи глобальний середній пулінг та максимальний пулінг, тобто ηs (x) = σ ($\Sigma_{k} w^{k}s \Sigma_{ij} E^{kij}s$ (x)). Експлуатуючи $w^{k}s$, можна для особливостей області обчислити увагу as(x)=ws ∗ Es (x) = {$w^{k}s$ ∗$E^{\kappa}s(x)$ | 1≤k≤n}, де n – кількість закодованих карт об'єктів. Тоді модель перекладу $G_{s \to t}$ дорівнює Gt (as (x)).

Дискримінатор моделі U-GAT-IT отримає на вхід справжнє або згенероване зображення: x ∈ {Xt, $G_{s \to t}$(Xs)}. Подібно до інших моделей перекладу, дискримінатор Dt є багатомасштабною моделлю і складається з енкодера E_{Dt}, класифікатора C_{Dt} і допоміжного класифікатора η_{Dt}. На відміну від інших моделей перекладу, обидва η_{Dt}(x) та Dt (x) навчаються розрізняти, походить x від Xt або $G_{s \to t}$ (Xs). Отримавши зразок x, Dt (x) використовує

карти уваги $a_{Dt}(x) = w_{Dt} * E_{Dt}(x)$, використовуючи w_{Dt} на закодованих картах ознак $E_{Dt}(x)$, які навчаються $\eta_{Dt}(x)$. Тоді значення дискримінатору Dt (x) стає рівним $C_{Dt}(a_{Dt}(x))$.

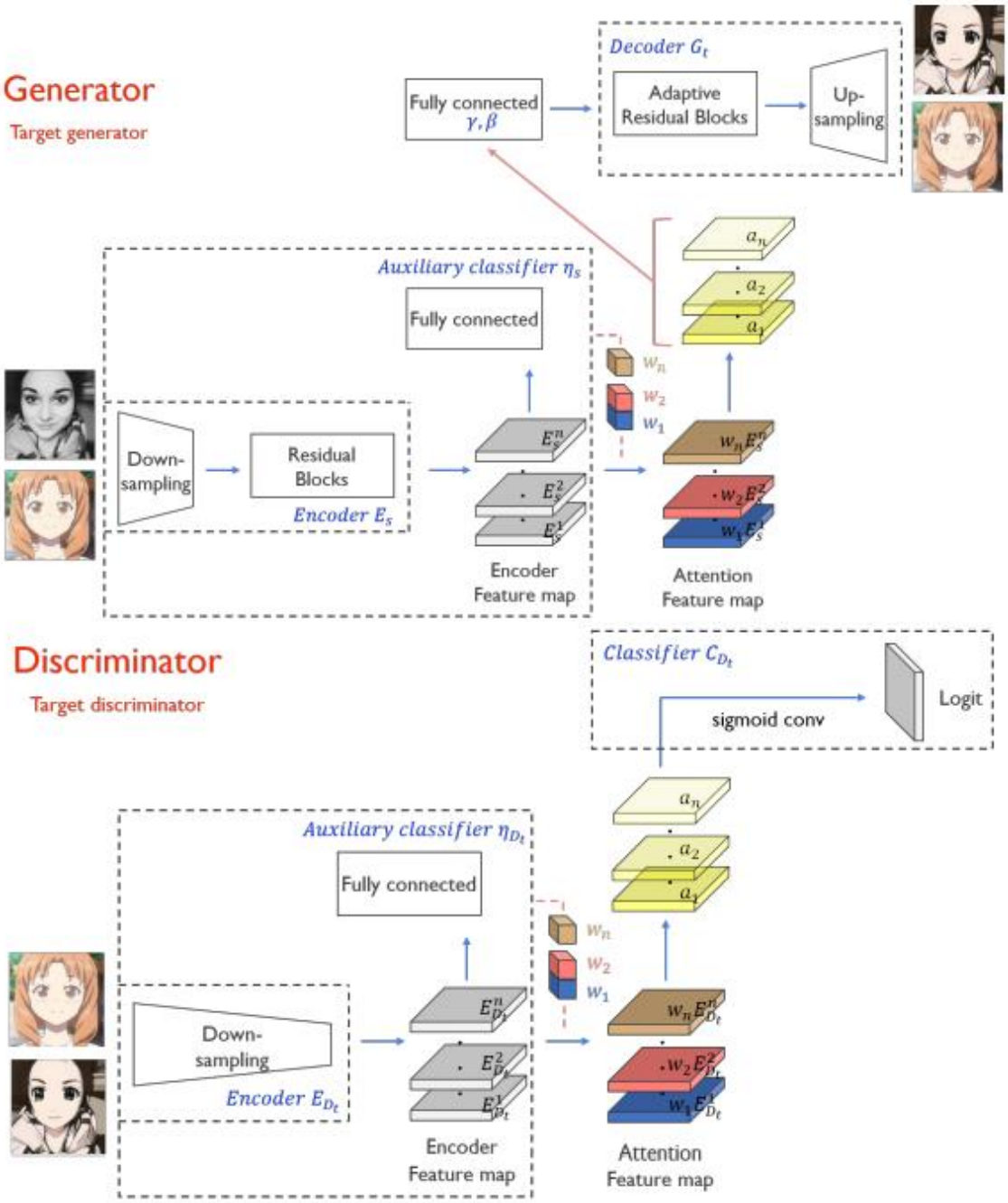

Рисунок 2.2 – Модель архітектури U-GAT-IT

Окрім функцій втрат, притаманних моделі CycleGAN – змагальних, реконструктивних та ідентичності – U-GAT-IT використовує також CAM (class activation maps) loss, використовуючи інформацію з допоміжних класифікаторів ηs та η_{Dt} для зображення x ∈ {Xs, Xt}. Генретор $G_{s \to t}$ та дискримінатор Dt дізнаються, у якому місці зображення їх найбільше потрібно вдосконалити:

$$L_{cam}^{s \to t} = -(\mathbb{E}_{x \sim X_s}[log(\eta_s(x))] + \mathbb{E}_{x \sim X_t}[log(1 - \eta_s(x))]), \tag{2.2}$$

$$L_{cam}^{D_t} = \mathbb{E}_{x \sim X_t}[(\eta_{D_t}(x))^2] + \mathbb{E}_{x \sim X_s}[(1 - \eta_{D_t}(G_{s \to t}(x))^2]$$

Боротьба з проблемою реконструкційних втрат в CycleGAN розглянута у статті [8] «MCMI: Multi-Cycle Image Translation with Mutual Information Constraints». Підхід MCMI розглядає одноциклові моделі перекладу зображень як модулі, які можна використовувати періодично в режимі багатоциклового перекладу, де процес перекладу обмежений взаємними інформаційними обмеженнями між вхідними та вихідними зображеннями. Запропоновані взаємні обмеження інформації можуть покращити зіставлення між областями, оптимізуючи функції перекладу, які не задоволняють властивості Маркова під час перекладу зображень. Вони показали, що навчені MCMI моделі створюють зображення вищої якості та переклад семантично точніший порівняно з найсучаснішими методами перекладу зображень. Фреймворк MCMI може бути застосований до існуючих моделей безпарного перекладу зображення в зображення з мінімальними модифікаціями.

Більшість непарних I2I моделей перекладу покладаються на реконструкційні втрати для досягнення генерації реалістичного зображення. Основне припущення полягає в тому, що семантична інформація зберігається між областями, коли модель змушена реконструювати зображення з перекладеного. Однак для завдання з переходом «один-до-багатьох» (наприклад, кольорові фотографії до чорно-

білих), від зворотнього перекладу очікується лише один можливий результат – початкове зображення, а інформації про початкові кольори в чорно-білому відсутня. Це робить генератори зображень сприйнятливими до стеганографії, де інформація, необхідна для реконструкції вхідних даних, прихована в невеликих збуреннях вихідного зображення. Отже, вихідне зображення не обмежується семантичним зв'язком із вхідним, щоб задовольнити реконструкційні втрати. Це призводить до менш надійних моделей I2I і погіршує функцію перетворення, яка не може зберегти важливу семантичну інформацію між областями. У [8] представлено теоретико-інформаційний погляд на цю проблему та пропонується фреймворк перекладу I2I для вирішення проблем із реконструкційними втратами. Їх розуміння полягає в тому, що при перетворенні з джерела до цільової області, ідеальна модель I2I повинна покладатися лише на семантичну інформацію з джерела, а не з будь-яких інших додаткових даних. Вони переформулювали модель I2I, яка відповідає такому обмеженню, як Ланцюг Маркова, і продемонстрували, що реконструкційні втрати порушують властивість Маркова: як вихідні, так і вхідні зображення змушені реконструювати одне одного. В ідеалі хотілося б отримати переваги щодо якості зображення, застосування реконструкції, одночасно задовольняючи властивість Маркова для збереження семантичної інформації.

Щоб узгодити властивість Маркова та циклічну реконструкцію, замінюється одноцикловий переклад на багатоцикловий переклад, і в цей багатоцикловий переклад додаються взаємні інформаційні обмеження (MCMI).

Зокрема, пропонується багатоцикловий переклад, де перший цикл перекладу використовує циклічні реконструкційні втрати і теорему про нерівність обробки даних (Data processing inequality), щоб забезпечити незростаючу взаємну інформацію (MI) для подальших циклів перекладу. Будь-яка навчена функція перетворення, яка не задовольняє нерівності

обробки даних, не буде відповідати ланцюгу Маркова і, отже, не зможе задовольнити властивість Маркова. Це дозволяє уникнути функціонального простору, де властивість Маркова не буде задоволено протягом навчання. Так, незважаючи на реконструкційні втрати в першому циклі, фреймворк MCMI призводить до якіснішого перекладу зображень та семантичної відповідності.

Ще один підхід представлено в [25] «SPA-GAN: Spatial Attention GAN for Image-to-Image Translation». Загалом, при перекладі зображення в зображення потрібно виявляти цікаві області у вхідному зображенні та дізнаватися, як перевести виявлені локації в цільову область. У безпарного перекладу зображень між двома областями слід звернути увагу на ділянки зображення, які підлягають перекладу. Завдання визначння таких місць є найбільш важливим у застосуванні перекладу зображення в зображення, де переклад повинен застосовуватися лише до певного типу об’єкта, а не цілого зображення. Наприклад, для передачі вхідного зображення «зебри» в цільову область «коні» спочатку потрібно знайти зебр на вхідному зображенні, а потім перетворити їх в коней.

Останні дослідження показують, що механізм уваги корисний для вдосконалення продуктивності генеративних змагальних мереж (GAN) при перекладі зображення в зображення. Зокрема, увага приділяється розкладанню генератора на дві окремі мережі: мережу уваги для прогнозування регіонів зацікавленості та трансформаційну мережу перекладу зображення з однієї області в іншу. Автори ж безпосередньо вводять механізм уваги до архітектури GAN та пропонують нову просторову увагу – модель SPA-GAN для перекладу зображення в зображення (рисунок 2.3). SPAGAN обчислює увагу у своєму дискримінаторі та використовує для допомоги генератору більше зосередитись на найбільш дискримінаційних регіонах між вхідною та цільовою областями. Зокрема, увага з боку дискримінатора визначається як просторова карта, що показує місця, на які фокусується дискримінатор для

класифікації вхідного зображення як справжнього чи підробленого. Потім ці видобуті карти просторової уваги повертаються назад до генератора, щоб більші вагові коефіцієнти надавалися дискримінаційним областям при обчисленні втрат генератора. Це має призводити до більш реалістичного зображення. Також вводять втрати карти ознак (Feature Map Loss) для збереження специфічних особливостей областей під час перекладу. Тобто в SPA-GAN обмежуємо карти ознак, отримані в першому шарі декодера, який повинен відповідати за визначення регіонів зацікавленості як реальних, так і генерованих зображень, щоб створені зображення були більш реалістичними.

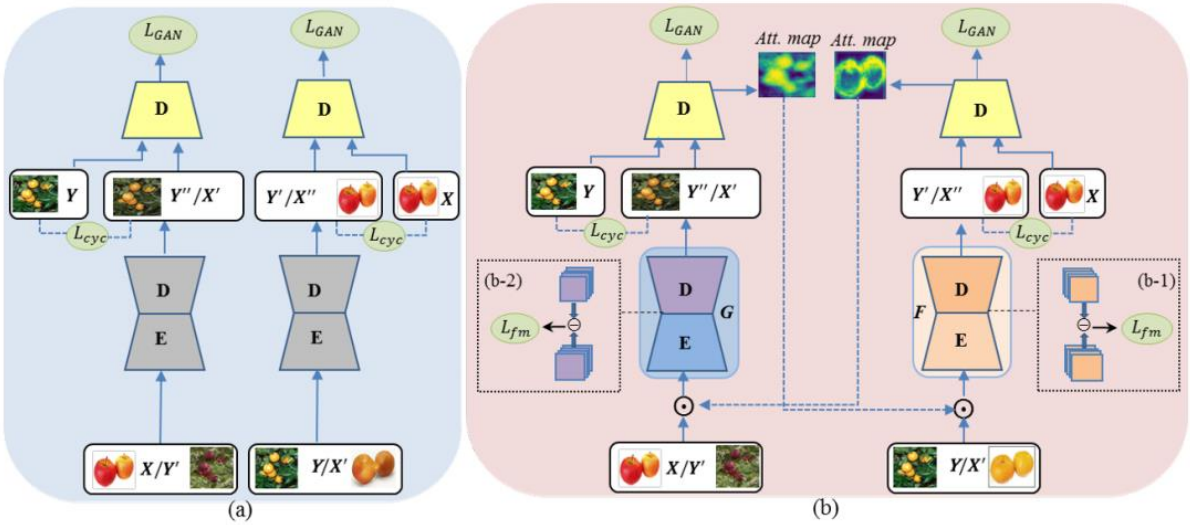

Рисунок 2.3 – Порівняння CycleGAN (a) та SPA-GAN (b)

Включення механізму уваги в переклад зображень може допомогти генеративній мережі приділити увагу точкам зацікавленості і зробити зображення бльш реалістичними. Запропонована модель SPA-GAN досягає цього шляхом явного передавання знань від дискримінатора до генератора, щоб змусити його зосередитись на дискримінаційних місцях вхідної та вихідної областей. На рис. 2.2 показані основні компоненти SPA-GAN і порівняння його з моделлю CycleGAN. Обидва фреймворки вивчають два

обернених відображення через один генератор і один дискримінатор для кожної області. Але у SPA-GAN дискримінатор породжує карти уваги додатково до класифікації вхідних даних як справжніх чи підроблених. Ці карти уваги повертаються назад на вхід генератора, тобто маємо ще один зворотній звязок. У той час як CycleGAN навчається за допомогою змагальної та реконструкційної втрат, SPA-GAN інтегрує змагальну, реконструкційні втрати та втрати карти ознак для отримання більш реалістичних результатів.

У GAN дискримінатор класифікує вхідні дані як підроблені або справжні. У SPA-GAN дискримінатор виділяє найбільш дискримінаційні регіони реальних та підроблених зображень додатково до класифікації. Ці дискримінаційні регіони ілюструють місця, на яких фокусується дискримінатор, щоб правильно класифікувати зображення. Формально, якщо вхідне зображення x, то просторова карта уваги $A_{DX}(x)$ має розмір, який збігається з розміром вхідного зображення x, і отримується шляхом подачі x дискримінатору. Визначається $A_{DX}(x)$ як сума абсолютних значень активацій для кожного пікселя по всіх кольорових каналах:

$$A_D = \sum_{i=1}^{C} |F_i| \ , \qquad (2.3)$$

де F_i – значення виходів і-го шару дискримінатора для конкретного вхідного зображення, а C – кількість кольорових каналів. A_D безпосередньо вказує на важливість нейронів шару для кожного пікселя зображення при класифікації вхідного зображення як підробленого чи справжнього.

Щоб застосувати просторову карту уваги до вхідного зображення x, вхідне зображення поелементно перемножується на просторову карту уваги (вони мають однаковий розмір) і подається генератору G, щоб допомогти йому зосередитися на найбільш дискримінаційних частинах при генеруванні x`:

$$x' = G(A_{D_X}(x) \odot x) \tag{2.4}$$

Щоб спонукати генератор вивчати специфічні особливості області, вводяться додаткові втрати карти ознак (Feature Map Loss). В ідеалі як справжнє вхідне, так і згенероване по ньому зображення повинні мати однакову абстракцію високого рівня для декодування. Втрати карти ознак карають за відмінності між картами ознак у першому шарі декодерів, які відповідають за розшифровку семантики високого рівня для справжніх та генерованих зображень відповідно. Зокрема, для генератора втрати карти ознак між відвідуваним зразком x_a у вхідній області X та відповідним згенерованим зразом $y`_a$ в зворотному відображенні Y` обчислюються за формулою:

$$\mathscr{L}_{fm}(G) = \frac{1}{C} \sum_{i=1}^{C} (||G^i(x_a) - G^i(y'_a)||_1) \ , \tag{2.5}$$

де G^i – i-та карта ознак, а C – кількість карт ознак в даному шарі генератора G. Втрати карти ознак вираховуються в першому шарі декодеру і додаються до загальних втрат генератора F (виконує зворотнє перетворення Y в X), і так же само втрати розраховуються для іншого генератора, і тоді загальні втрати карт ознак становлять:

$$\begin{aligned} \mathscr{L}_{fm}(G,F) = \frac{1}{C} \sum_{i=1}^{C} (||G^i(x_a) - G^i(y'_a)||_1) \\ + \frac{1}{C} \sum_{i=1}^{C} (||F^i(y_a) - F^i(x'_a)||_1) \end{aligned} \tag{2.6}$$

При цьому в дослідженні не вказано, чи застосування механізму просторової уваги допомагає в боротьбі проти звички генератора кодувати вхідне зображення у вихідному, щоб інший генератор його відновив.

Є ще один простий підхід до покращення – One2one CycleGAN [44]. Суть проста – два дискримінатори і тільки один генератор, якому доводиться інвертувати зображення з однієї області до іншої. Перевагами автори вважають зменшення кількості параметрів моделі (позбавляємося одного генератора), і оскільки він навчається обома наборами даних, то це ніби подвоює кількість даних, таким чином борються з перенавчанням. Але маю сумніви щодо ефективності на обмеженій кількості даних, оскільки від перенавчання страждають саме дискримінатори, а вони не змінюються по зрівнянню зі звичайною CycleGAN. У той же час, як раніше я встановив, будь-яка додаткова регуляризація покращує CycleGAN. Також, на мою думку, One2one CycleGAN зовсім не вирішує проблеми кодування вхідного зображення в генерованому – основної проблеми реконструкційної втрати.

У [47] (ACL-GAN) запропоновано в класичній моделі CycleGAN із втратами ідентичності замінити попіксельне порівняння реконструйованого та початкового зображень на ще одні змагальні втрати – Adversarial Consistency Loss, у якій додатковий дискримінатор намагається розрізнити зображення, згенеровані генератором з початкового зображення та з перетвореного до цільової області.

На рисунку 2.4 наведено схему ACL-GAN. Два генератори Gs, Gt навчаються змаганням з двома дискримінаторами Ds, Dt (зліва). Для стабільності навчання додано втрати ідентичності (справа). Замість реконструкційної втрати додатковий дискримінатор Dacl змагається з генератором Gs, розрізняючи походження згенерованого ним зображення – з початкового x_s чи з перетвореного до цільової області x_T

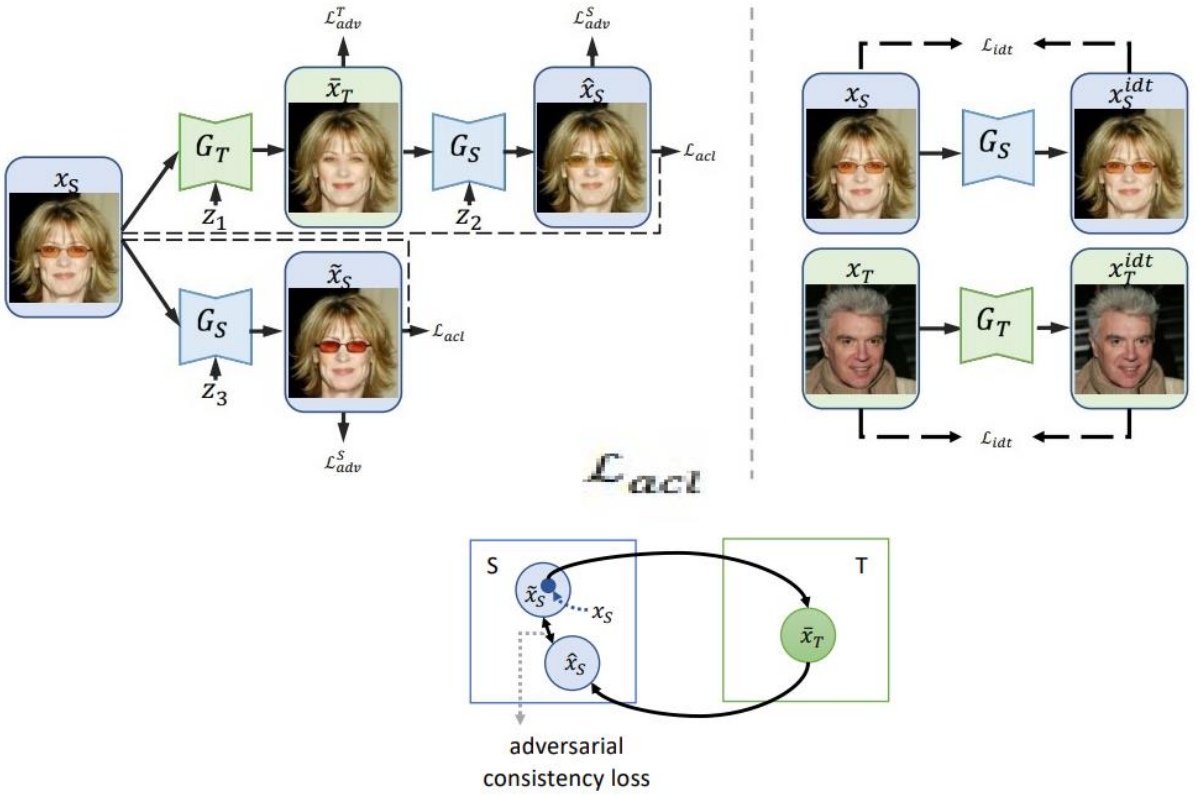

Рисунок 2.4 – Схема ACL-GAN

2.3 Сучасні варіанти односторонніх моделей перекладу зображень

Оскільки реконструкційні втрати мають указані в попередніх розділах недоліки, то відмова від архітектури CycleGAN із перекладами в обидві сторони і побудова моделі з перекладом лише в одну сторону є альтернативою для безпарного перекладу зображень. Звичайно одностороння модель складається з трьох функціональних частин: енкодер, декодер та дискримінатор. На вхід енкодера подається вхідне зображення, і результатом є інформація, яка подаються до декодеру, вихід якого є перекладеним зображенням. Разом енкодер та декодер становлять собою генератор, і для його навчання методом змагання використовується дискримінатор, який намагається визначити: справжнє чи згенероване генератором зображення подається йому на вхід. Так, змагаючись, вони покращуються, і результатом мають бути такі згенеровані зображення, які

людина не зможе відрізнити від справжніх. Відмінність від звичайної генеративно-змагальної мережі GAN полягає в тому, що замість випадкового вектору на вхід декодера подається інформація, яка є результатом обробки енкодаром вхідного зображення.

Якщо навчати таку модель лише за допомогою змагальних втрат, немає можливості спонукати її генерувати зображення, які є перекладом вхідних в іншу область. Змагальні втрати лише забезпечують приналежність згенерованих зображень до області, справжні зображення якої показуємо дискримінатору. Але відповідність перекладу не забезпечується.

Деякі архітектури, наприклад, U-Net [26] з використанням пропускаючих з'єднань (skip connections), де кожен шар енкодера додатково з'єднаний із шаром відповідної роздільної здатності декодера, генерують зображення досить близькі до вхідних. Це зумовлено «просочуванням» елементів вхідного зображення до вихідного.

Незважаючи на близькість елементів зображень, така генерація не може називатися перекладом, це лише генерація «за мотивами» вхідного зображення. Більш того навчання такої можелі може закінчитися mode collapse – ситуацією, коли генератор буде видавати один і той самий рузультат, незважаючи на вхідне зображення (рисунок 2.5).

Односторонні моделі перекладу зображень використовують додаткові втрати, метою яких є забезпечення точності перекладу, і також вони слугують регуляризатором, не допускаючи mode collapse (рисунок 2.6).

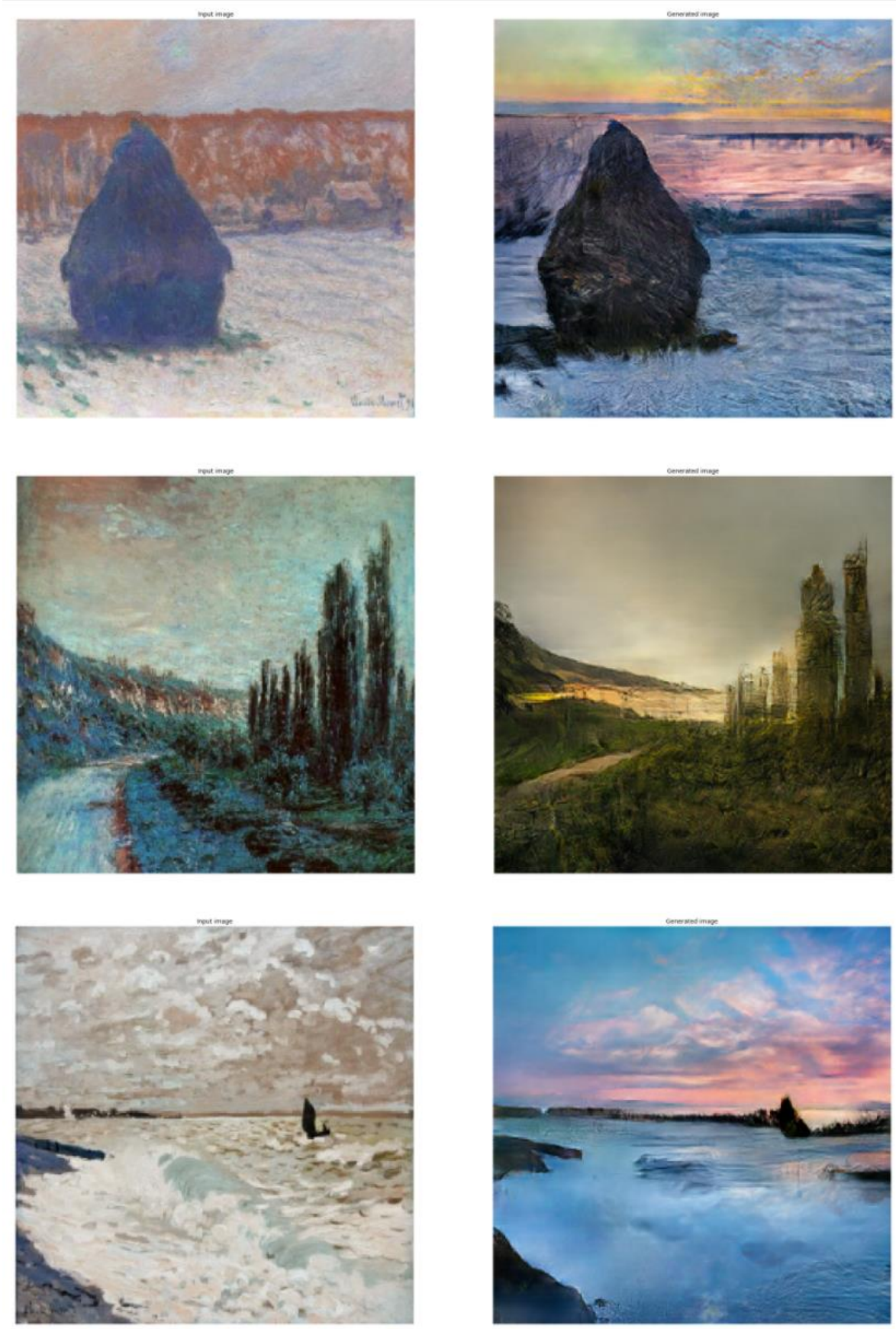

Рисунок 2.5 – U-Net генератор із використанням лише змагальних втрат генерує зображення, які не є перекладом, хоча мають схожі елементи

Рисунок 2.6 – Приклад mode collapse. Незважаючи на різноманітність вхідних (зліва) зображень, генератор видає схожі зображення (справа)

DistanceGAN. Однією з перших односторонніх моделей перекладу даних була представлена DistanceGAN [11]. Дослідники показали, що можна вивчити відображення між вхідною областю і цільовою областю у вигляді одностороннього безпарного перекладу, забезпечуючи високу міжобласну кореляцію між відповідними попарними відстанями, обчисленими в кожній області. Новий вид втрат дозволяє одностороннє відображення, яке показує кращі числові результати, ніж реконструкційні втрати. Він полягає в тому, що відстані між двома зображеннями однієї області $|| x_1 – x_2 ||$ та їх перекладами до іншої області $|| G(x_1) – G(x_2) ||$ залежні

між собою. Дистанції між зразками обчислюються попарно для кожної з пар всередниі навчальної партії (batch) та для пар згенерованої партії. Також обчислення дистанцій можна застосовувати навіть до кожного окремого зображення, обчислюючи відстань, наприклад, між верхньою та нижньою частинами зображення.

Автори надали обґрунтування та емпіричні свідчення того, що для багатьох семантичних відображень існує високий ступінь кореляції між попарними відстанями в двох областях. Іншими словами, нехай d_k є нормованим вектором попарних відстаней партії в одній області і нехай dg_k – вектор отриманих нормованих відстаней в іншій області, тоді $\sum d_k dg_k$ має бути великою. При тренуванні генератора G середнє значення та дисперсія попарних дистанцій всього навчального набору зображень кожної області попередньо обчислюється та використовується для нормалізації вектору дистанцій зразків кожної партії. Замість максимізації кореляції $\sum d_k dg_k$ автори пропонують мінімізувати суму абсолютних різниць $\sum |d_k - dg_k|$.

Загальний вид дистанційних втрат наступний:

$$\mathcal{L}_{\text{distance}}(G_{AB}, \hat{p}_A) = \mathbb{E}_{x_i, x_j \sim \hat{p}_A} \left| \frac{1}{\sigma_A}(\|x_i - x_j\|_1 - \mu_A) - \frac{1}{\sigma_B}(\|G_{AB}(x_i) - G_{AB}(x_j)\|_1 - \mu_B) \right| , \quad (2.7)$$

де μ_A, μ_B (σ_A, σ_B) – середні значення (середньо-квадратичні відхилення) попарних відстаней тренувальних наборів даних для області A та B і обчислюються попередньо до навчання моделі.

Автори також випробували ідею додати дистанційні втрати до моделі CycleGAN, використовуючи одночасно дистанційні, реконструкційні та змагальні втрати, отримавши найкращі результати саме для такої моделі. Зважаючи на простоту реалізації, односторонню DistanceGAN спробую дослідити в подальших експериментах.

GcGAN. У безпарному перекладі зображень, із точки зору імовірнісного моделювання, наша мета полягає в моделюванні спільного розподіл P_{XY} з урахуванням зразків, взятих із розподілів P_X і P_Y окремих

областей. Оскільки два розподіли можна зробити з нескінченного набору можливих з’єднань, важко гарантувати, що окремо взяті вхідний x і вихідний G_{XY} (x) є парою, у якій відносини між зображеннями мають зміст без додаткових припущень та обмежень. Для вирішення цієї проблеми останні підходи використовували припущення про можливість реконструкції, тобто відображення G_{XY} та його зворотне відображення G_{YX} мають бути бієкціями. Зокрема, якщо подаюти приклад x в мережу $G_{XY} \circ G_{YX}$ ($X \rightarrow Y \rightarrow X$), то вихід повинен бути реконструкцією x. Те ж саме стосується і y, тобто G_{YX} (G_{XY} (x)) ≈ x та G_{XY} (G_{YX} (y)) ≈ y. Далі DistanceGAN [11] показує, що дотримання відстані між зображеннями в межах областей дозволяє одностороннє безпарне відображення областей, а не одночасне вивчення як G_{XY}, так і G_{YX}. Але такі обсеження не враховують особливі властивості зображень, як прості геометричні перетворення (глобальні геометричні перетворення без деформації фігури), які не змінюють семантики зображення. Людина може легко сприйняти зображення після геометричного перетворення (наприклад, повороту на 90^0), тобто при таких перетвореннях зберігається інформація, яка відрізняє різні об'єкти між собою.

На цій основі розроблено обмеження геометричної узгодженості, яке допомагає зменшити простір пошуку можливих розв’язків, зберігаючи при цьому правильний набір розв’язків, і призводить до геометрично узгодженої генеративної змагальної мережі (GcGAN) для безпарного перекладу зображень. Це обмеження геометричної узгодженості мотивоване тим, що геометричне перетворення f (·) між вхідними зображеннями повинно зберігатися відповідними перекладачами G_{XY} та $G_{\tilde{X}\tilde{Y}}$, якщо $\tilde{X}$ та $\tilde{Y}$ – області, отримані шляхом застосування f (·) на прикладах X та Y відповідно. Якщо взяти випадковий приклад x із вхідного домену X та заздалегідь визначену геометричну функція перетворення f (·), геометрична узгодженість може бути виражена формулою:

$$f(G_{XY}(x)) \approx G_{\tilde{X}\tilde{Y}}(f(x)) \text{ та } f^{-1}(G_{\tilde{X}\tilde{Y}}(f(x))) \approx G_{XY}(x), \qquad (2.8)$$

де $f^{-1}(\cdot)$ – обернена функція $f(\cdot)$. Оскільки навряд чи G_{XY} та $G_{\tilde{X}\tilde{Y}}$ завжди зазнають збою в одному і тому ж місці, G_{XY} та $G_{\tilde{X}\tilde{Y}}$ спільно регулюють один одного за допомогою обмеження геометричної узгодженості і тим самим виправляють помилки один одного. Таке обмеження геометричної узгодженості дозволяє односторонній безпарний переклад зображень між областями, тобто G_{XY} можна навчити незалежно від G_{YX}.

У статті [12] використані два прості, але репрезентативні геометричні перетворення: вертикальне перевертання (vf) та обертання за годинниковою стрілкою на 90 градусів (rot), щоб продемонструвати геометричну узгодженість. Звичайно, як і належить авторам опублікованої статті, вони зрівняли свою модель формування реалістичних зображень з іншими існуючими на той час моделями перекладу зображень і виявили, що вона найкраща.

Але знову виявилося, що в разі додавання обмеження геометричної узгодженості до CycleGAN отримуються ще кращі результати. Тобто односторонні моделі перекладу зображень на основі DistanceGAN та GcGAN гірші за використання їх особливостей в складі CycleGAN, при чому вони не вирішують основних проблем CycleGAN, пов'язаних із реконструкційними втратами (рисунок 2.7).

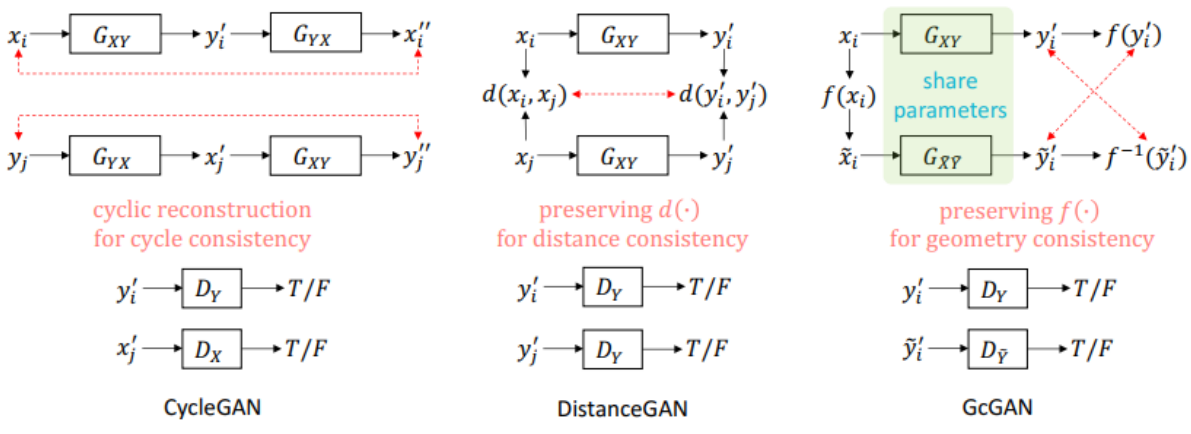

Рисунок 2.7 – Різниця між моделями CycleGAN, DistanceGAN та GcGAN

TraVeLGAN. Побічний ефект навчання word2vec в обробці природньомовних текстів – «семантична алгебра» – надихнув дослідників [14] «TraVeLGAN: Image-to-image Translation by Transformation Vector Learning». Окрім генератора та дискримінатора, TraVeLGAN використовує третю мережу – сіамську мережу, для створення латентного простору даних для кодування семантики високого рівня, що характеризує області, до яких належать зображення. Цей простір направляє генератор під час навчання, змушуючи генератор дотримуватись векторної арифметики між точками цього простору. Вектор, який перетворює одне зображення на інше у вхідній області, повинен бути тим самим вектором, який перетворює згенеровану версію цього зображення в згенеровану версію іншого зображення. Це схоже на кодування word2vec у природній мові, якщо перетворюємо одне зображення вхідної області в інше зображення вхідної області, перемістивши об’єкт переднього плану з верхнього лівого кута до нижнього правого кута, тоді генератор повинен генерувати відповідний об’єкт у двох точках в цільовій області, віддалених на такий самий вектор перетворення.

У word2vec семантичні векторні перетворення є властивістю вивчення латентного простору з контекстів слів. У TraVeLGAN модель навчається виробляти ці вектори під час вивчення латентного простору. Переклад зображень складається з двох частин: (a) перетворення зображення в іншу область і (b) – зробити перекладене зображення певним чином схоже на вхідне зображення. Звичайно, (a) досягається змаганням генератора з окремою мережею – дискримінатором, а для досягнення (б) просто обмежують клас функцій генератора. Автори пропонують натомість досягти (b) також із використанням окремої мережі.

Потрібно вивчити два відображення G_{XY}: X → Y та G_{YX}: Y → X, що роблять переклад між областями. Більше того, генератори мабть не просто імітувати області на загальному рівні, а потрібні значущі та ідентифікуючі відносини між двома зображеннями. Автори стверджують, що це завдання складається з двох компонентів: приналежність до області та

індивідуальність. Приналежність до області забезпечується за допомогою GAN – змаганням генератора з дискримінатором. А задача індивідуальності полягає в тому, щоб було поясненні взаємозв'язку, чому саме отримане зображення є перекладом вхідного зображення x_i, а не якогось іншого x_j. Інакше генератор може генерувати випадкових членів області Y, не зважаючи на вхідні зображення.

Автори TraVeLGAN вводять поняття трансформаційного вектору між двома точками. Найважливіша ідея полягає в тому, що незалежно від того, яка трансформація необхідна для отримання x_j з x_i, аналогічна трансформація має відокремити дві згенеровані версії цих зображень $G_{XY}(x_i)$ та $G_{XY}(x_j)$.

Тобто генератор має вивчити перетворення з такою умовою: якщо вектор між вхідними зображеннями ними ν (xi, xj) = xj – xi, то повинно бути ν (xi, xj) = ν (GXY (xi), GXY (xj)). Це більш потужна властивість, ніж навіть збереження відстані між точками, оскільки простір потрібно організувати таким чином, щоб збереглися напрямки векторів, а також величини. Ця властивість вимагає, щоб вектор, який перетворює xi до xj, був тим самим вектором, який перетворює G_{XY} (xi) до G_{XY} (xj).

Але такий фреймворк міг би визначити лише прості перетворення, оскільки він дивиться безпосередньо на вхідне зображення. За аналогією з word2vec там алгебра понять працює не на one-hot кодуванні слів, а у деякому зменшеному семантичному латентному просторі. Так і в TraVeLGAN натомість перевизначили вектор перетворення:

$$\nu (xi, xj) = S (xj) – S (xi), \qquad (2.9)$$

де S – функція, яка представляє кожну точку в деякому латентному просторі.

Якщо S вивчає семантику високого рівня для кожного зображення, то можна використовувати наше уявлення про збереження векторів перетворення для керівництва генерацією. Автори пропонують вивчити

такий простір з аналогічно змаганню з дискримінатором у традиційній системі GAN: за допомогою кооперативної сіамської мережі S. Ціль мережі S – перетворити зображення в такий простір, у якому взаємовідносини між вхідними зображеннями такі ж, як і між їх згенерованими версіями:

$$\begin{aligned} L_{TraVeL} &= \Sigma\Sigma_{i \neq j} Dist(\nu_{ij}, \nu'_{ij}) \\ \nu_{ij} &= S(x_i) - S(x_j) \\ \nu'_{ij} &= S(G_{XY}(x_i)) - S(G_{XY}(x_j)) \ , \end{aligned} \tag{2.10}$$

де Dist – дистанційна метрика, така, як косинусна подібність.

Ця формула включає параметри G, але сам G спочатку потребує цей простір, щоб навчитися генерувати. Як підсумок, ці дві мережі залежать одна від одної для досягнення своїх цілей. Але на відміну від випадків G та D, цілі G і S не протилежні, а кооперативні. Вони обидві хочуть, щоб L_{TraVeL} було мінімізовано, але G не навчиться тривіальній функції для задоволення цієї мети, оскільки також намагається обдурити дискримінатора. А мережа S все ще могла б вивчити тривіальну функцію (наприклад, завжди виводить нуль), тому, щоб уникнути цього, автори додають їй ще одну вимогу, щоб зробити її ціль багатозадачною. Це повинно відповідати стандартним контрастним сіамським цілям L_{Sc}, що кожна точка знаходиться принаймні на δ від будь-якої іншої точки в латентному просторі:

$$L_{S_c} = \Sigma\Sigma_{i \neq j} max(0, (\delta - ||\nu_{ij}||_2)) \tag{2.11}$$

Це стимулює S до вивчення латентного простору, який визначає деякі відмінності між зображеннями, тоді як L_{TraVeL} стимулює S до його організації. Отже, кінцеві цільові функції S і G:

$$L_S = L_{Sc} + L_{TraVeL} \tag{2.12}$$

$$L_G = L_{adv} + L_{TraVeL}$$

G і S співпрацюють у тому сенсі, що кожен намагається мінімізувати L_{TraVeL}, але кожен має додаткову конкретну мету до свого завдання. Ці мережі спільно навчаються, і в такий спосіб G вчиться генерувати зображення, які S може побачити і відобразити в якийсь простір, де зв’язок між оригінальними та створеними зображеннями зберігається.

Для експериментів автори статті TraVeLGAN використали U-net архітектуру з пропускаючими з'єднаннями [26] для генератора. Для дискримінатора – стандартна згорткова мережа класифікатора з stride-2, яка подвоює кількість фільтрів на кожному шарі, поки рівень не становитиме 4x4, і видає єдине число – сигмоїдальну ймовірність. Сіамська мережа ідентична, за винятком того, що виводить не єдине число як дискримінатор, а виводить дані розміром латентного простору, без будь-якої нелінійної активації.

Зважаючи на зрозумілість та простоту реалізації, поєднану з осмисленою ідеєю вивчення латентного простору, модель TraVeLGAN буде досліджена в моїй роботі.

CUT. На основі попередніх розробок NVIDIA в генеративно-змагальних мережах – StyleGAN та StyleGAN2 – їх автори запропонували свою односторонню модель перекладу зображень – CUT [13] (Contrastive Unpaired Translation). При перекладі ми бажаємо отримати на виході вигляд цільової області (зебри), зберігаючи структуру або зміст конкретного вхідного зображення коня. Це, по суті, задача розплутування: відокремлення вмісту, який потрібно зберегти між областями, від зовнішнього вигляду, який повинен змінитися. Як правило, цільовий вигляд забезпечується з використанням змагальних втрат, при цьому вміст зберігається з використанням реконструкційних обмежень. Але реконструкція передбачає, що відносини між двома областями є бієкцією, а найчастіше це не так.

Автори пропонують альтернативний, досить прямий спосіб підтримки відповідності за змістом, але не зовнішнім виглядом – шляхом максимізації

взаємної інформації між відповідними клаптиками (патчами) вхідного та вихідного зображень. У правильному результаті, якщо на виході в патчі, наприклад, отримали згенерований зебровий лоб, то ми очікуємо, що в тому місці оригінального зображення знаходився лоб коня, а не інші частини коня або об’єкти заднього фону.

Досягають цього, використовуючи різновид контрастних втрат, – функцію втрат InfoNCE [27] (рисунок 2.8), яка спрямована на навчання кодуванню, що асоціює відповідні патчі один з одним, одночасно роз'єднуючи (віддаляючи) їх від інших. Для цього енкодер вчиться звертати увагу на спільність між двома областями – частинами об'єктів та фігурами, ігноруючи те, що змінюється (фактура тварин). Дві мережі, генератор і енкодер, співпрацюють разом, щоб сформувати таке зображення, що може бути легко простежено походження його патчів у вхідному зображенні. Крім того, автори виявили наступне: якщо негативні приклади патчів брати всередині самого вхідного зображення, а не зовні з інших зображень у наборі даних, то це змушує патчі краще зберігати зміст.

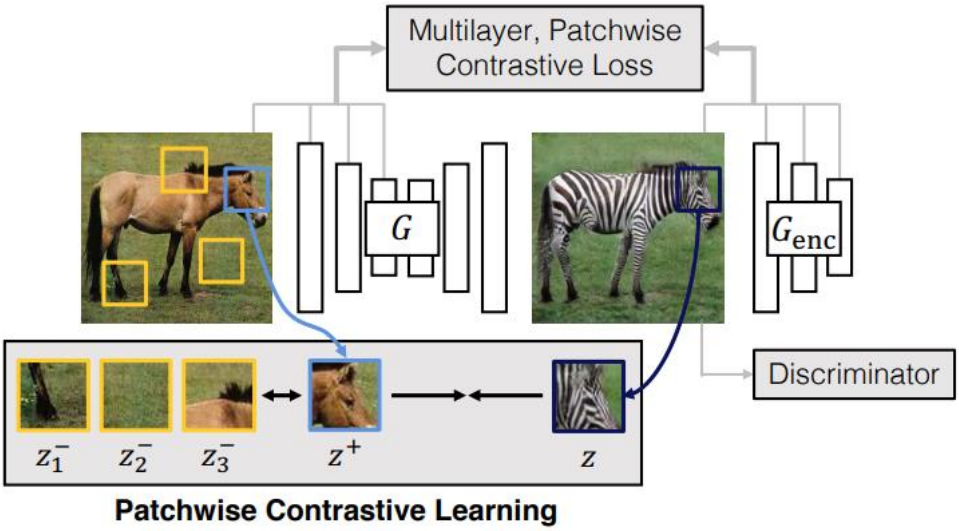

Рисунок 2.8 – Контрастивне навчання для одностороннього перекладу. Сформований вихідний патч повинен виглядати найбільш схоже до відповідного вхідного патчу в порівнянні до інших випадкових патчів

Автори використовують фреймворк оцінки контрасту з шумом (noise contrastive estimation framework) [27] для максимізації взаємної інформації між входом і виходом. Ідея контрастивного навчання полягає в поєднанні двох сигналів, «запиту» та його «позитивної відповіді» на відміну від інших точок у наборі даних, які називаються «негативними» (рисунок 2.9).

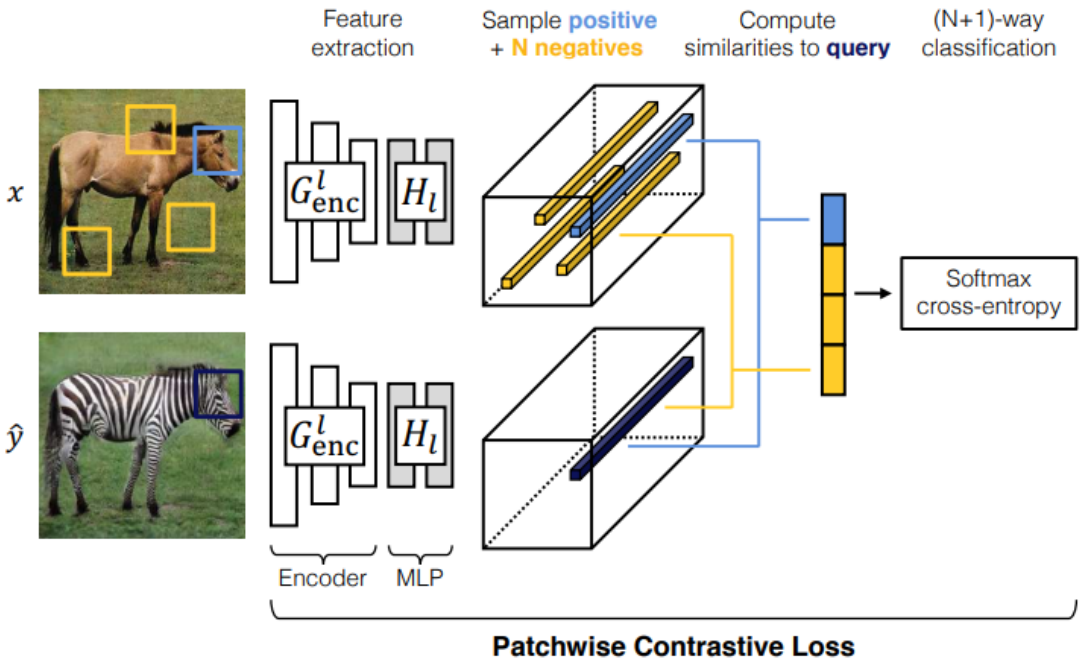

Рисунок 2.9 – Втрати контрасту патчів

Обидва зображення, x та yˆ, кодуються в особливий тензор. Із генерованого зображення yˆ вибирається патч і порівнюється з вхідним патчем у тому самому місці. Вирішується задача класифікації (N + 1), де N – кількість негативних прикладів патчів, які відбираються з того ж вхідного зображення в різних місцях. Повторно використовується енкодер генератора G_{enc} і додається двошарова мережа MLP. Ця мережа вчиться проектувати як вхідні, так і вихідні патчі на спільний простір, закодовуючи їх.

Запит позитивний та N негативних перетворюються у K-мірні вектори v, $v+ \in R^K$ і $v- \in R^{N\times K}$, відповідно $v_n- \in R^K$ позначає n-й негатив. Ці вектори нормалізуються на одиничну сферу, щоб запобігти руйнуванню або розширенню простору. Отримується задача класифікації (N + 1), де відстані між запитом та іншими прикладами масштабуються за температурою τ = 0,07 і передаються як логіти. Втрати обчислюються як перехресна ентропія і представляють собою ймовірність обрання позитивного прикладу над негативними:

$$\ell(\boldsymbol{v}, \boldsymbol{v}^+, \boldsymbol{v}^-) = -\log\left[\frac{\exp(\boldsymbol{v}\cdot\boldsymbol{v}^+/\tau)}{\exp(\boldsymbol{v}\cdot\boldsymbol{v}^+/\tau)+\sum_{n=1}^{N}\exp(\boldsymbol{v}\cdot\boldsymbol{v}_n^-/\tau)}\right] \quad (2.13)$$

Мета – пов’язати вхідні та вихідні дані. У цьому контексті запит – це генероване зображення. Позитивні та негативні (відповідні та невідповідні) – вхідне зображення. Автори застосовують багатошарове контрастне навчання. Оскільки навіть на рівні пікселів кольори тіла зебри (чорно-біле) може бути сильніше пов’язане з кольором тіла коня, ніж з фоновими відтінками трави, використовуються виходи багатьох шарів енкодера для навчання на основі контрасту патчів.

Оскільки енкодер Genc вже обчислив свої шари нейронів для вхідного зображення при створенні перекладу, вони легко доступні, і цю перевагу використують. Кожен шар і просторове розташування нейрона в ньому представляє патч вхідного зображення. Чим глибші шари, тим більший розмір патчу. Вибираються L шарів, що цікавлять, і пропускаються через невелику двошарову мережу MLP (H_L), отримується вектор ознак $\{z_l\}_L = \{H_l(G^l enc(x))\}_L$, де $G^l enc$ представляє вихід l-го вибраного шару. Шари пронумеровано $l \in \{1, 2, ..., L\}$, а $s \in \{1, ..., S_l\}$, де S_l – кількість просторових розташувань нейронів у кожному шарі. Тоді $z^s_l \in R^{Cl}$ та інші ознаки $z^{S\backslash s}_l \in R^{(Sl-1)\times Cl}$, де Cl – кількість каналів на кожному шарі. Так само

кодується вихідне зображення y^ в $\{z\hat{}_l\}_L = \{H_l (G^l_{enc} (G (x)))\}_L$. Тоді функція втрат PatchNCE вираховується:

$$\mathcal{L}_{\text{PatchNCE}}(G, H, X) = \mathbb{E}_{\boldsymbol{x} \sim X} \sum_{l=1}^{L} \sum_{s=1}^{S_l} \ell(\hat{\boldsymbol{z}}_l^s, \boldsymbol{z}_l^s, \boldsymbol{z}_l^{S \setminus s}) \qquad (2.14)$$

Для того, щоб генератор не робив непотрібних перетворень, додатково можна використати PatchNCE від справжніх зображень області Y – $L_{PatchNCE}(G, H, Y)$, тобто аналогічно втраті ідентичності, яку використовують у CycleGAN. Тоді загальна формула втрат моделі CUT буде мати вигляд:

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) + \lambda_X \mathcal{L}_{\text{PatchNCE}}(G, H, X) + \lambda_Y \mathcal{L}_{\text{PatchNCE}}(G, H, Y) \qquad (2.15)$$

Council. У статті «Breaking the cycle – colleagues are all you need» в односторонній моделі перекладу використали цілу «раду» з чотирьох генераторів, які навчаються за допомогою восьми дискримінаторів, по два на кожен генератор. При цьому з цих двох генераторів один звичайний – розпізнає справжнє і генероване зображення, а інший – спеціальний, розпізнає чи належить генероване зображення його генератору, чи якомусь іншому генератору. Цей спеціальний дискримінатор примушує генератор до «конформізму» з іншими «колегами»-генераторами, знаходячи спільні ознаки.

TSIT. Досить цікавий підхід TSIT (Two-Stream Image-to-image Translation) – двохпоточний переклад зображень пропонують в [43] (рисунок 2.10). Серед його особливостей завісні змагальні втрати (англ. hinge loss), втрати на основі VGG-19 для забезпечення семантичного змісту (perceptual loss), втрати відповідності ознак (feature matching loss) для забезпечення відповідності проміжних ознак на різних шарах багатомасштабного дискримінатора (рисунок 2.11).

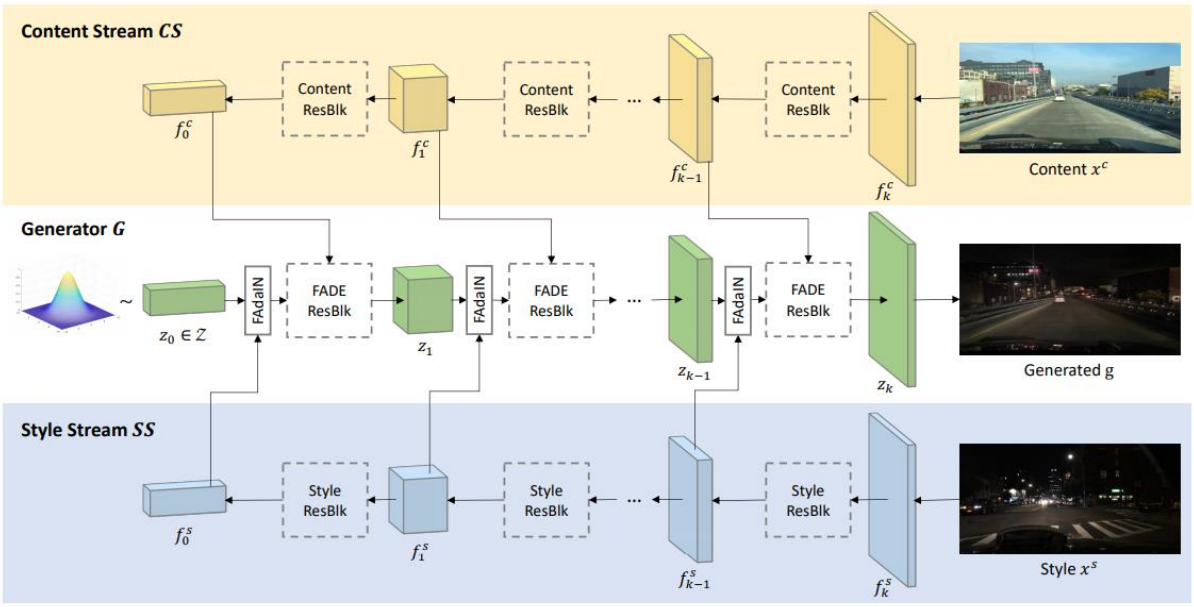

Рисунок 2.10 – Схема TSIT. Генератор отримує на вхід гаусівський шум. Багатомасштабний клаптиковий дискримінатор не показано на схемі

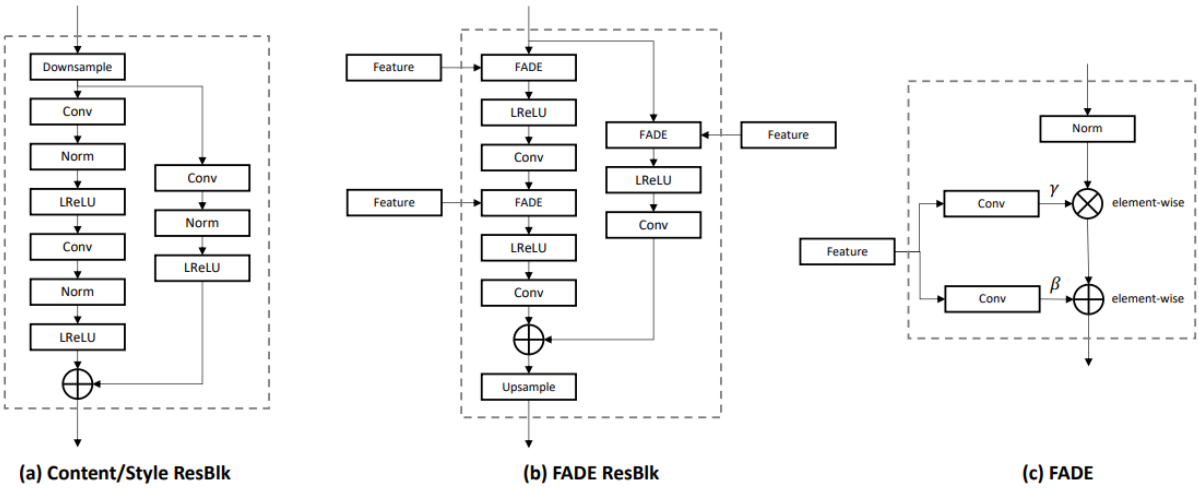

Рисунок 2.11 – Модулі TSIT: блок зміст/стиль у симетричних потоках змісту/стилю (a), блок FADE в генераторі (b), модуль FADE в блоці FADE (c) – він виконує поелементну денормалізацію шляхом модуляції нормалізованої активації з використанням вивченого афінного перетворення, визначеного параметрами модуляції γ та β

2.4 Методи вирішення проблеми обмеженої кількості даних

У жовтні 2020 року вийшла стаття «Training Generative Adversarial Networks with Limited Data» [20], присвячена проблемі обмеженої кількості даних при навчанні генеративно-змагальних мереж взагалі. Моделі безпарного перекладу зображень є одним із різновидів застосування GAN, тому можна розглянути загальні принципи вирішення проблеми в GAN та спробувати застосувати їх до моделей перекладу зображень, які будемо перевіряти на даних змагання Kaggle.

Посилаючись на [21], [28], у статті [20] відмічено, що ключова проблема малих наборів даних полягає в тому, що дискримінатор перенавчається на навчальних прикладах; його зворотний зв'язок із генератором стає безглуздим, і навчання починає розходитися. Саме до такого висновку я теж дійшов, попередньо досліджуючи модель CycleGAN. Як і майже у всіх областях глибокого навчання, аугментація даних є стандартним рішенням проти перенавчання. При цьому обертання, шум тощо робить модель класифікатора зображень стійкою до спотворень, що зберігають семантику – це дуже бажана якість для класифікатора. Але на відміну від класифікатора GAN, який навчається в рамках подібної аугментації даних, вчиться генерувати аугментований розподіл. І таке «витікання» аугментації до генерованих зображень вкрай не бажано. Наприклад, посилення шуму призводить до зашумлених генерованих зображень, навіть якщо таких немає у наборі даних. У попередніх дослідженнях мною було встановлено, що це стосується і моделі CycleGAN – аугментація вхідних даних не покращує метрики FID, а навпаки погіршує її.

Отже, методами, які застосовували для вирішення проблеми обмеженої кількості даних для генеративно-змагальних мереж, є особливі способи застосування аугментації та перенесення знань.

Лише в двох статтях у 2021 році застосовано принципово інший підхід – Self-Supervised Discriminator [37], самоконтрольований дискримінатор. Цей підхід забезпечити сильну регуляризацію для D – напрочуд простий (рисунок 2.12). Розглядають D як енкодер і тренують його за допомогою маленьких декодерів. Таке навчання автоматичному кодуванню змушує D витягувати з зображення ознаки, за якими декодери можуть зробити хороші реконструкції. Декодери навчаються разом із D через реконструкційні втрати лише на справжніх зображеннях:

$$\mathcal{L}_{recons} = \mathbb{E}_{\mathbf{f}\sim D_{encode}(x),\ x\sim I_{real}}[\|\mathcal{G}(\mathbf{f}) - \mathcal{T}(x)\|], \qquad (2.16)$$

де f – проміжна карта ознак D, функція G складається з обчислювача f та декодера, функція T представляє обробку зразків x зі справжніх зображень. I_{real}.

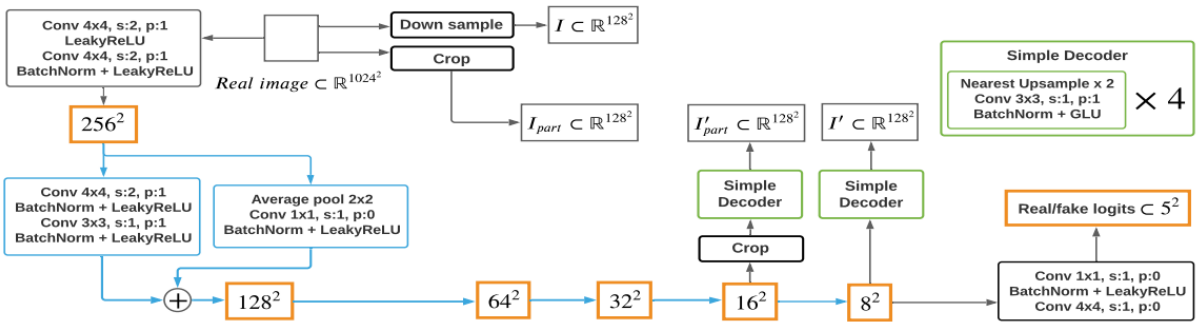

Рисунок 2.12 – Структура дискримінатора D. Сині лінії представляють собою однакові блоки пониження розмірності, а зелені блоки – однакові декодери

Самоконтрольований D використовує два декодери для карт ознак на двох масштабах: f1 на 16^2 і f2 на 8^2. Декодери мають лише чотири згорткові шари conv для отримання зображень розміром 128 × 128, спричиняючи незначні додаткові обчислення (набагато менші, ніж інші методи

регуляризації). З f1 вирізається 1/8 висоти та ширини, а потім обрізається справжнє зображення в такій же пропорції, щоб отримати Ipart. Щоб отримати I змінюється розмір справжнього зображення. Декодери видають I'part з обрізаного f1, та I' з f2. D, і декодери навчаються разом, щоб мінімізувати втрати (вираз 2.16) шляхом зіставлення Ipart з I'part та I з I'. Таке реконструктивне навчання гарантує, що D створює більш повне уявлення про вхідне зображення, що буде складатися з загальної композиції (з f2), так і докладних текстур (з f1). Підхід автокодування, який тут використовується, є типовим методом для самоконтрольованого навчання, яке підвищує надійність моделі та здатності до узагальнення. У контексті GAN автори [37] виявили, що регуляризований D за допомогою стратегії самоконтрольованого навчання значно покращує якість синтезу генератора.

2.5 Аугментація даних

Оскільки ще в [30] відмічено, що будь-яка аугментація, застосована до навчального набору даних, буде унаслідувана згенерованими зображеннями, пряме застосування аугментації до набору справжніх даних не приносить значного успіху в GAN. У 2017 році в [37] було спостереження, що додавання гауссового шуму до вхідних зображень і, лінійно зменшуючи його силу під час навчання, покращує сходження моделей GAN. У [29] вивели таку саму ідею самостійно з теоретичної точки зору. Вони показали: додавання гауссового шуму до обох справжніх та згенерованих зображень може допомогти проти нестабільності навчання при збереженні розподілу вхідних та генерованих зображень. А в [31] запропоновано balanced consistency regularization (bCR) як вирішення проблеми протікання аугментації до результату. За ідеєю bCR згенеровані зображення перед подачею на розгляд дискримінатору теж підлягають такій самій аугментації. Крім того, на вхід дискримінатора подаються також і неаугментовані версії справжнього та згенерованого зображень.

Дослідження продовжені в [35], але цей підхід не повністю припиняє протікання аугментації, оскільки генератор не штрафується, якщо згенерує зображення з елементами аугментації, і тому він більш схожий за результатом на звичайну аугментацію.

У 2020 році з'явилося одразу три статті [20], [33], [34] щодо аугментації даних при навчанні GAN. Усі три статті по колу посилаються одна на одну, тобто дослідження проводилися приблизно одночасно, і всі три демонструють результат кращий за конкурентів.

StyleGAN2-ADA. У статті [20] автори StyleGAN з NVIDIA демонструють, як використовувати широкий спектр аугментації для запобігання перенавчання дискримінатора, забезпечуючи при цьому, щоб жодна з аугментацій не протікала до сформованих зображень. Вони пропонують інший підхід – стохастичну аугментацію дискримінатора. Аугментація дискримінатора означає надівання спотворюючих, можливо навіть руйнівних окулярів на дискримінатор, і прохання до генератора виготовити зразки, які неможливо відрізнити від навчального набору при перегляді через ці окуляри (рисунок 2.13). Але навчання неявно скасовує спотворення і знаходить правильний розподіл, якщо спотворення є оборотніми перетвореннями розподілу ймовірностей у просторі даних. Такі оператори називають «невитікаючими».

Сила цих оборотніх перетворень полягає в тому, що вони дозволяють робити висновки про рівність або нерівність основних множин, які слід генерувати, спостерігаючи лише за аугментованими множинами. Це не означає, що аугментація, виконана на окремому зображенні, повинна не змінити зображення. Наприклад, настільки екстремальна аугментація, як встановлення 90% вхідного зображення на нуль, є оборотною в сенсі розподілу ймовірностей: навіть людина могла б міркувати про оригінальний розподіл, ігноруючи чорні зображення, орієнтуючись лише на 10% зображення. З іншого боку, рівновипадкові повороти на $\{0^o, 90^o, 180^o, 270^o\}$

не оборотні: неможливо розрізнити відмінності між орієнтаціями після аугментації.

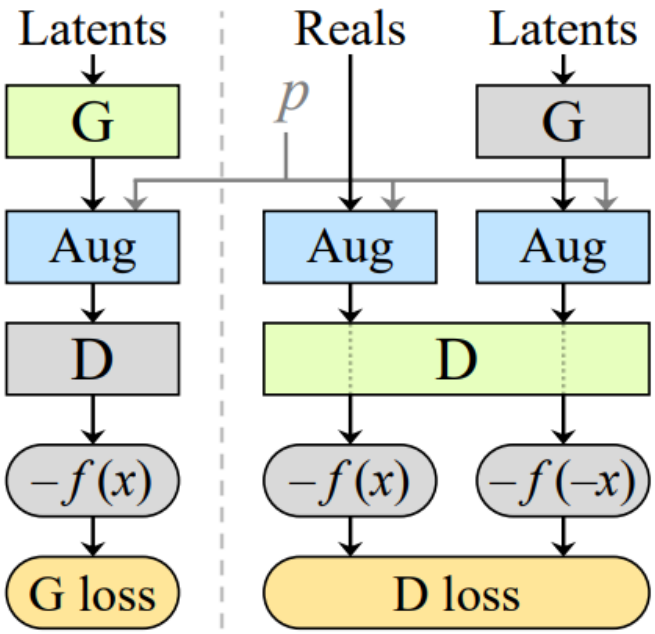

Рисунок 2.13 – Стохастична аугментація дискримінатора

Ситуація змінюється, якщо це обертання виконується лише з імовірністю p <1: це збільшує відносне входження 0^{o}, і тепер аугментовані розподіли можуть співпадати лише за умови, що генеровані зображення мають правильну орієнтацію. Подібним чином можна розробити багато інших стохастичних аугментацій, які будуть непротікаючими за умови, що не застосовуються з ненульовою ймовірністю. Це можна зробити для великого класу широко використовуваних аугментацій, включаючи детерміновані відображення (наприклад, базові перетворення), додавання шуму, перетворення кольору, повороти, перевертання, масштабування та отвори (наприклад, виріз частини зображення). Крім того, композиція непротікаючих аугментацій, застосована у фіксованому порядку, дає непротікаючу аугментацію.

Автори [20] також дослідили, за якого значення ймовірності застосування аугментації *p*, ці аугментації будуть непротікаючими. Експериментально встановлено, що при $p < 0.8$ протікання схоже не трапиться, тобто це є безпечне значення. Також дослідили, які перетворення найкорисніші в колекції, а додавання яких – малоефективне, оскільки чим більше аугментацій додано, тим повільніша модель. Найбільш корисними були pixel blitting (дзеркальні відображення, 90° повороти, цілочисельні перетворення) та інші геометричні перетворення. Середньої корисності виявилися кольорові перетворення. Щодо шуму, вирізання та фільтрування – користі від них не отримано. Ефект більш залежав від об'єму навчальних даних. При кількості зображень 2000 ефект від аугментації дуже значний, при 10 тисячах зображень великі значення *p* не потрібні, а при 140 тисячах зображень будь-яка аугментація шкідлива.

Також висловлена ідея, щоб саме значення *p* не задавати гіперпараметром, а вивчати під час навчання, тобто модель сама адаптується до особливостей вхідних даних. Дану модель автори реалізували як доповнення до своєї StyleGAN2, назвавши її StyleGAN2-ADA, де ADA – адаптивна аугментація дискримінатора (Adaptive discriminator augmentation).

DiffAugment. У [33] відмічають, що критично важливо усунути необхідність величезних наборів даних для навчання GAN. Але зменшення кількості навчальних даних призводить до різкого погіршення роботи. Наприклад, якщо взяти лише 10% або 20% даних CIFAR-10, точність навчання дискримінатора насичується швидко (майже до 100%); але точність його на валідаційному наборі постійно зменшується (до 30%). Припускають, що дискримінатор просто запам'ятовує весь навчальний набір. Це жорстке перенавчання порушує динаміку тренувань і призводить до погіршення якості зображення. Приблизно такі ж дані та висновки я отримав при попередньому дослідженні моделі CycleGAN із висновком, що дискримінатор перенавчається.

Дискримінатор має тенденцію запам’ятовувати спостереження в міру навчання. Надмірно перенавчений дискримінатор карає будь-які згенеровані зразки, які хоча б трохи відрізняються від навчальних даних, що спричиняє неінформативні градієнти, погане узагальнення і зазвичай призводить до нестабільності навчання.

Широко вживаною стратегією зменшення перенавчання в класифікації зображень є аугментація даних, що може збільшити різноманітність навчальних даних без збору нових зразків. Такі перетворення як обрізання, перевертання, масштабування, операції з кольорами та маскування (виріз) частин є загальновживаними аугментаціями для моделей комп’ютерного зору. Але принцип застосування аугментації даних до GAN – інший. Якщо перетворення буде додано лише до справжніх зображень, то генератор буде й генерувати аугментовані зображення. Як наслідок, результати страждають від зсувів розподілу та артефактів (наприклад, маскується область, неприродний колір). Згідно іншого підходу можна аугментувати як справжні, так і генеровані зображення під час навчання дискримінатора, та це порушить тонкий баланс між генератором і дискримінатором, що призводить до поганого сходження навчання, оскільки вони оптимізують абсолютно різні цілі.

Для боротьби з цим у [33] представляють простий, але ефективний метод, – диференційовану аугментацію DiffAugment (Differentiable Augmentation), який застосовує таку саму диференційовану аугментацію як справжніх, так і згенерованих зображень і для генератора, і для дискримінатора. Це дозволяє розповсюджувати градієнти і до генератора, регулюючи дискримінатор, не маніпулюючи цільовим розподілом і підтримуючи баланс динаміки навчання. Експерименти на різноманітних архітектурах та наборах даних GAN послідовно демонструють ефективність цього методу. За допомогою DiffAugment автори вдосконалили BigGAN і StyleGAN2 та отримали найкращі показники на CIFAR-10, CIFAR-100, використовуючи лише 20% навчальних даних.

У дослідах автори застосовують такі прості трансформації, як трансляція (у межах [-1/8, 1/8] від розміру зображення, доповненого нулями), виріз (маскування випадковим квадратом половини розміру зображення), а також колір (включаючи випадкову яскравість в межах [−0,5, 0,5], контраст в межах [0,5, 1,5] та насиченість у межах [0, 2]). Політика DiffAugment, як правило, підтримує вищу валідаційну точність дискримінатора ціною нижчої точності на навчальному наборі, полегшення проблеми перенавчання та досягнення кращого сходження (рисунок 2.14).

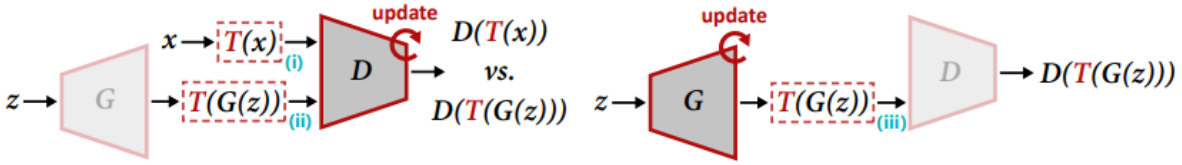

Рисунок 2.14 – Схема DiffAugment для навчання дискримінатора D (лівіше) та генератора G (правіше)

DiffAugment застосовує аугментацію T як до справжніх зображень, так і до згенерованих. Коли оновлюються параметри G, градієнти мають повертатися скрізь T, тому T має бути диференційованим по відношенню до входу.

DAG. У [34] принциповим питанням є те, що при аугментації даних (DA), розподіл набору навчальних даних стає Pd^T , який може відрізнятися від розподілу вхідних даних Pd. Тоді при навчанні генератора мінімізується $JS(Pd^T || Pg)$ замість JS (Pd || Pg). Автори провели всебічне дослідження, щоб зрозуміти проблему застосування DA для навчання GAN. Головне завдання полягає у використанні аугментованого набору даних з розподілом Pd^T для покращення вивчення Pd – оригінального розподілу набору даних. Вони пропонують аугментацію даних, оптимізовану для генеративно-змагальних мереж DAG (Data Augmentation optimized for GAN). Розглядають оборотні

перетворення та регуляризацію дискримінатора. Теоретично надають гарантії сходження навчання для оборотних перетворень; емпірично показують, що для досягнення покращення можна використовувати як оборотні, так і необоротні перетворення. І, звичайно, запропонована ними DAG при застосованні в деяких існуючих моделях GAN, може досягати найкращих показників.

Першу спробу використати аугментацію для навчання GAN вони назвали IDA – покращений DA (Improved DA). Аугментація {Tk} здійснюється над згенерованим та справжнім зразком, і суміш перетворених справжніх/згенерованих зразків подається як вхідні дані для навчання єдиного дискримінатора D. Навчання дискримінатора (розглядається як двійковий класифікатор) за допомогою аугментованої вибірки має тенденцію до поліпшення узагальнення дискримінатором. Хоча IDA може отримати вигоду від інвертування (оборотності) перетворення, єдиний дискримінатор не зберігає дивергенції JS оригінального GAN. І оскільки IDA не зберігає JS, то не гарантує сходження GAN.

У другій спробі – аугментація даних, оптимізована для GAN (DAG), запропоновано механізм подолання вищезазначеної проблеми (рисунок 2.15). DAG спрямована на використання розширеного набору даних X^T із зразками, перетвореними за допомогою $T = \{T_1, T_2, \ldots, T_K\}$ для покращення вивчення розподілу вхідних даних. DAG використовує переваги різних трансформацій, використовуючи різні дискримінатори D, $\{D_k\} = \{D_2, D_3, \ldots D_K\}$. Дискримінатор D_k тренується на зразках, перетворених T_k , щоб розрізняти трансформовані справжні зразки проти трансформованих генерованих зразків.

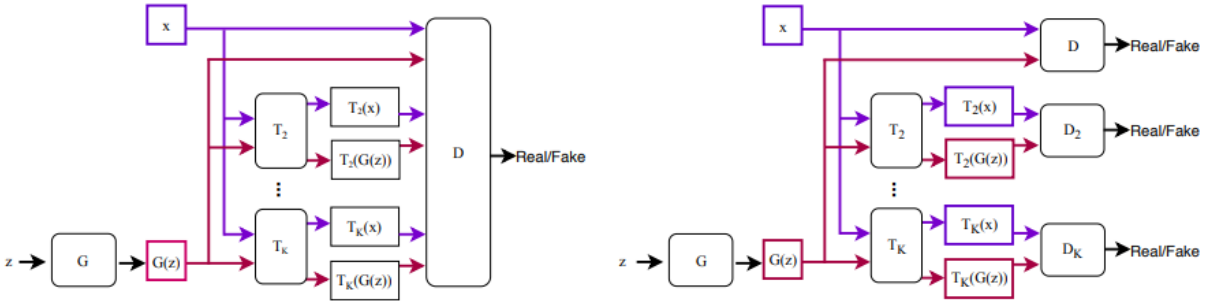

Рисунок 2.15 – Модель IDA (лівіше) з єдиним дискримінатором та модель DAG (правіше) з мультидискримінатором. У навчанні генератора беруть участь тільки згенеровані зразки (червоні шляхи)

Генератор створює зображення, щоб обманути одночасно і дискримінатор D, і $\{D_k\}$. І щоб вдосконалити себе генератор отримає зворотній зв'язок одразу від К дискримінаторів. Якщо генератор хоче, щоб створені зразки виглядали реалістично, то їх аугментовані варіанти повинні виглядати реалістично також. Відгуки обчислюються не тільки JS (Pd || Pg) оригінальних зразків, але також JS (Pd^{Tk}|| Pg^{Tk}) трансформованих зразків.

Щоб скористатися перевагами аугментації даних, автори пропонують застосувати регуляризацію дискримінаторів через спільні параметри для D, {Dk}. Як і IDA, дискримінатор отримує вигоду від аугментації даних для поліпшення репрезентативності при навчанні дискримінатора і, крім того, модель зберігає ту саму ціль JS для забезпечення сходження навчання. Кількість спільних шарів у дискримінаторів не впливає на збереження JS у DAG, тому вплив кількості спільних шарів встановлюється шляхом експериментів. З такою регуляризацією дискримінатора продуктивність значно покращена. У практичній реалізації D та {Dk} поділилися усіма шарами, крім найостанніших шарів, для реалізації різних голів для різних виходів.

У цій роботі автори зосередились на оборотньому перетворенні зображення. Перетворення є оборотнім, якщо його трансформований зразок

можна повернути до точного оригіналу зображення. Наприклад, це можуть бути популярні афінні перетворення зображення – обертання, перевертання або фліпрот (перевертання + обертання) тощо; але емпіричним шляхом автори виявили, що DAG-фреймворк добре працює з більшістю видів аугментації (навіть із необоротними перетвореннями), тобто обрізання та трансляція. Якщо ж перетворення є оборотним, властивість збіжності GAN теоретично гарантована.

Експериментально встановили: чим більше аугментацій (і відповідно дискримінаторів) застосовували, тим кращі показники метрики FID отримували.

ACCR-CycleGAN. Щодо моделей перекладу зображень аугментація була застосована в статті «Augmented Cyclic Consistency Regularization for Unpaired Image-to-Image Translation» [45]. На відміну від звичайного CycleGAN замість втрати ідентичності використовується особливий вид втрат, коли дискримінатор має навчитися видавати на аугментовані зображення такі ж самі оцінки справжнє/генероване, як і на звичайні зображення. Аугментація проводиться такими перетвореннями, які не змінюють семантику зображень (дзеркальні відображення, обрізання, зміщення, вирізи). Втрати вираховуються як квадрат різниці оцінки дискримінатора для звичайного і аугментованого зображень. Дискримінатору показують три пари зображень: справжнє та його аугментована версія, генероване та його аугментована версія, реконструйоване та його аугментована версія. При цьому дискримінатор не навчається правильно розрізнити: реконструйоване зображення є справжнім чи генерованим, йому лише потрібно давати дуже близьку оцінку реконструйованому зображенню та його аугментованій версії.

2.6 Попереднє навчання та перенесення знань

Перенесення навчання зменшує вимоги до об’єму навчальних даних, якщо починати навчання з моделі, навченої за допомогою якогось іншого набору даних, замість випадкової ініціалізації.

Автори DeepI2I [17] стверджують, що першими застосували перенесення знань для безпарного перекладу зображень. Тобто до листопада 2020 року не було опубліковано жодного використання попереднього навчання та перенесення знань для I2I моделей. Автори розглядають проблему, що сучасні архітектури I2I мають обмежену потужність перекладати між класами зі значними змінами форми (наприклад, з обличчя собаки на обличчя суріката). Для успішного перекладу між такими областями необхідне семантичне розуміння зображення. Ця інформація зазвичай витягується з глибоких шарів мережі з низькою роздільною здатністю. Втрата просторової роздільної здатності, яка виникає при додаванні декількох down-sampling шарів, є одним із факторів, що ускладнює використання глибоких мереж для I2I. Для високоякісного I2I як інформація низького рівня про стиль, так і семантична інформація високого рівня повинні обидві передаватися з енкодера в генератор системи I2I. Тому автори пропонують глибоку ієрархічну структуру перекладу I2I, де враховуються репрезентації інформації на різних рівнях абстракції (глибини). Використання ієрархічної структури дозволяє виконувати переклад I2I між класами зі значущими змінами форми. Запропонований метод DeepI2I, ґрунтується на найсучаснішій моделі BigGAN [28], поширюючи його на переклад I2I, додаючи енкодер з тією ж архітектурою, що і дискримінатор. Це вводить таку новину в область I2I, як ортогональна регуляризація для точного контролю над компромісом між якістю та різноманітністю генерованого зображення. Оскільки BigGAN використовує, крім самого зображення, інформацію про клас зображення

(conditional GAN), то DeepI2I теж отримує можливість перекладати між багатьма областями, але зобов'язана використовувати мітки класів.

Іншим фактором, який ускладнює застосування перекладу для зображень великого розміру, є те, що збільшується кількість параметрів, а значить для їх навчання потрібні великі набори даних. Для вирішення цієї проблеми автори пропонують новий метод передачі знань для перекладу I2I. Передача знань від попередньо навченої мережі в значній мірі приносить користь дискримінатору, що дозволяє повторно використовувати високоякісні мережі. Автори пропонують ініціалізувати вагові параметри моделі I2I з вагами з попередньо навченої GAN. Вони використовують дискримінатор попередньо навченого BigGAN для ініціалізації як енкодера, так і дискримінатора моделі перекладу DeepI2I, а також попередньо навчений генератор, щоб ініціалізувати декодер генератора моделі DeepI2I. Для усунення розбіжностей між різними шарами енкодера та декодера пропонують мережу адаптерів для вирівнювання розмірів даних попередньо ініціалізованих енкодера та декодера (рисунок 2.16).

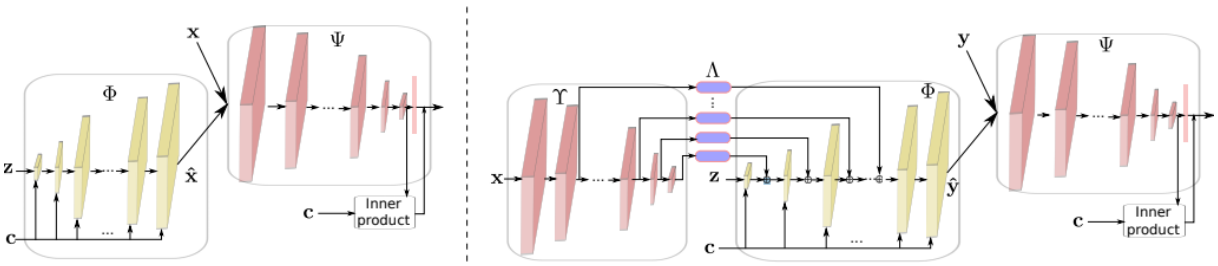

Рисунок 2.16 – Зліва: традиційна форма BigGAN, яка містить генератор Φ та дискримінатор Ψ. Справа: DeepI2I

DeepI2I складається з чотирьох частин: енкодер Υ, адаптер Λ, генератор Φ та дискримінатор Ψ. Енкодер Υ ініціалізується попередньо навченим дискримінатором Φ у складі BigGAN (ліворуч) , а також генератором Φ і дискримінатором Ψ попередньо навченої BigGAN

(ліворуч). Адаптер Λ спрямований на вирівнювання попередньо навченого енкодера ϒ та попередньо навченого генератора Ψ

Оцінюючи, яка з моделей DeepI2I працює краще – попередньо навчена в складі BigGAN чи навчена з нуля, – автори виявили, що попереднє навчання значною мірою прискорює збіжність і перевершує DeepI2I, яка навчалася з нуля.

Трансфер навчання сприяло широкому застосуванню мереж глибокої класифікації. Він дозволив використовувати знання з високоякісних попередньо навчених мереж на великих наборах даних для завдань із відносно невеликою кількістю маркованих даних. Враховуючи успіх передачі знань для класифікації, може бути дивно, що цей метод не застосовувався до проблем I2I. Не зрозуміло, який набір даних може функціонувати як універсальний набір даних I2I, (наприклад, imageNet використано для завдань класифікації). Навчання мережі I2I на imageNet є необґрунтованим, оскільки багато перекладів буде безглуздими (переклад «машини» на «будинки», або «червоний» на «натовп»). Натомість автори стверджують, що знання необхідні енкодеру, декодеру та дискримінатору системи I2I з високоякісних попередньо навчених GAN.

Вони пропонують ініціалізувати мережу DeepI2I із попередньо навченого високоякісного BigGAN. Попередньо навчений дискримінатор Ψ використовують для ініціалізації дискримінатора Ψ запропонованої моделі, і попередньо навчений генератор Φ для ініціалізації декодера Φ deepI2I. Це залишило б енкодер ϒ досі неініціалізованим, що призвело б до складнощів навчання генератора, який не може нарівні змагатися з висококваліфікованим дискримінатором. Тому пропонують також використовувати попередньо навчений дискримінатор Ψ для ініціалізації енкодеру ϒ. Це має сенс, оскільки дискримінатор BigGAN має можливість правильно класифікувати вхідні зображення в imageNet, що є ефективним видобувачем ознак.

Така архітектура встановлює зв'язки між різними рівнями енкодера та декодера. Це дозволяє передавати інформацію різних рівнів (і абстракції) між енкодером ϒ та генератор Ф. Поєднання здійснюється виділеною мережею адаптерів і суматорами. Автори тренують deepI2I у два етапи. На першому етапі навчають лише адаптери та дискримінатор, підтримуючи енкодер та декодер замороженими. На другому етапі тренують всю систему, за винятком енкодера, який фіксують незмінним. Виявляється, що заморожений енкодер (попередньо навчений як дискримінатор BigGAN) призводить до кращих показників.

Інші дослідники займалися проблемою попереднього навчання генеративно-змагальних мереж без застосування у сфері перекладу зображень. Уперше перенесення навчання до GAN було застосовано в 2018 році [18]. Для навчання GAN на відносно малому наборі даних замість початкової випадкової ініціалізації використали параметри GAN навченої на ImageNet, і далі донавчали на цільовому наборі даних. Також дослідили, як ефективність попереднього навчання на інших даних та перенесення знань залежить від наборів даних – для попереднього навчання та цільового. Виявилося, що великі різноманітні набори даних. як ImageNe, успішно конкурують з меншими, але більш близькими до цільової області. Недолік – таке донавчання покращує на відносно великих наборах даних, хоча б від 10 тисяч зображень. Для менших – дискримінатор однаково швидко перенавчається.

У [22] пропонують новий метод передачі знань для генератора, у якому при донавчанні основні шари параметрів заморожені, а оновлюються лише параметри масштабу і зсуву в шарах BatchNormalization генератора. За допомогою оновлення тільки цих параметрів та замороження всіх інших параметрів генератора, кількість потрібних зображень для навчання зменшується. Застосували цей метод до дуже невеликого набору даних, що складався менше ніж зі 100 зображень, і показали, що якість вища, ніж при застосуванні попередніх методів, і що при генерації зображень модель

розпізнає семантику даних. Недолік – якість результуючих зображень досить низька.

У [19] запропонували MineGAN – метод передачі знань для генеративних моделей, заснований на видобуванні знань, що є найбільш корисним для конкретної області. При цьому можна добувати знання як з однієї, так і з декількох попередньо навчених GAN. Це робиться за допомогою майнерської мережі, яка визначає, яка частина генеративного розподілу кожної попередньо навченої GAN найближча до цільової області.

У [23] завданням є генерація зображень, коли мається лише кілька зображень цільової області. Ключовим компонентом є здатність використовувати попереднє навчання. Наприклад, можна використовувати знання варіацій зовнішнього вигляду обличчя, щоб легко уявити варіації конкретного мультиплікаційного обличчя.

Основним припущенням цієї моделі є те, що вхідна (попередньо вивчена) та цільова області мають деякі спільні латентні фактори з деякими відмінностями, пов’язаними з їх чіткою різницею у зовнішності. Наприклад, при переході від справжніх природних облич до смайлів варіації пози та виразу можуть бути природньо поширені на цільову область.

Для досягнення цієї мети автори пропонують пряму та ефективну техніку адаптації. Вони адаптують ваги попередньо навченої моделі без введення додаткових параметрів. Зафіксована архітектура передбачає, що нудне ручне проєктування нових параметрів (наприклад, кількості параметрів, їхня позиція тощо) не є необхідними. Натомість проблема полягає в тому, як адаптувати ваги відповідно до обмежених даних цільової області, зберігаючи передані знання або різноманітність від джерела знань.

Ключовою властивістю, яку слід зазначити, є те, що ваги мають різний рівень важливості. Отже, до кожного параметру не слід ставитися однаково в процесі адаптації або донавчання. Автори пропонують кількісно визначити «важливість» кожного параметра, підкреслюючи збереження важливих параметрів під час процесу налаштування. Для цього

використовують еластичну консолідацію вагових коефіцієнтів EWC (Elastic Weight Consolidation), якою оцінюється важливість кожного параметра, оцінюючи його інформацію Фішера. При цьому для можливості навчання на екстремально малій кількості зображень (<10) заморожують дискримінатор.

По відношенню до GAN перенесення навчання досліджувалося в [32], де показано сильні результати, заморозивши шари дискримінатора з найвищою роздільною здатністю після переносу знань (Freeze-D).

У статті «Freeze the Discriminator: a Simple Baseline for Fine-Tuning GANs» [32] запропонували заморожувати низькорівневі шари дискримінатора і показали, що донавчання обох: генератора і дискримінатора – із замороженими нижніми шарами дискримінатора (з найвищою роздільною здатністю) виявляється на диво простим й ефективним методом перенесення навчання. Перенесення навчання з (Freeze-D) дає значно кращі результати, ніж навчання з нуля, і його успіх найчастіше залежить, насамперед, від різноманітності вхідного набору даних, а не від подібності між попередньо вивченим набором даних та цільовим набором.

У [24] одним із ключових спостережень, що мотивує метод, є добре навчена ГАН, що може генерувати реалістичні зображення, яких не спостерігається в навчальному наборі даних, що демонструє здатність GAN до узагальнення. Таке узагальнення GAN надзвичайно привабливе у випадку обмеженої кількості даних.

Якщо наївно тренувати ГАН при обмеженій доступній кількості даних, отримаємо перенавчання, оскільки потужні моделі GAN мають численні параметри, необхідні для реалістичної генерації. Щоб уникнути перенавчання, можна розглянути можливість передачі додаткової інформації з інших областей за допомогою перенесення знань.

Більшість робіт із перенесення знань зосереджені на дискримінаторі, зважаючи на те, що низькорівневі фільтри (ті, що близькі до вхідних

спостережень) мають багато параметрів і вимагають багато даних для навчання, і їх попереднє навчання часто призводить до кращих результатів.

Автори [24] пропонують перенесення/заморожування фільтрів низького рівня обох: генератора і дискримінатора – попередньо навченої моделі GAN для полегшення генерації в інших цільових областях з обмеженими даними. Крім того, представляють нову техніку adaptive filter modulation (AdaFM), для кращої адаптації переданих низькорівневих фільтрів до цільових областей. У цьому запропонованому підході не просто «заморожують» передані фільтри, а й розширюють їх залежно від цілі.

2.7 Робочі гіпотези досліджень

У 2021 році з'явилася перша стаття [39], у якій оглянуто всі роботи по перекладу зображень I2I, розроблених за останні роки. Проаналізовано ключові методи існуючих робіт I2I та досягнутий прогрес. Дану статтю я використав для пошуку ідей для дослідження, щоб сформулювати робочі гіпотези, за допомогою яких показати найкращий результат у змаганні Kaggle «Use GANs to create art».

Ключове твердження. Модель має бути простою і зрозумілою. Оскільки ідентифіковані основні проблеми – перенавчання дискримінатора на картинах Моне та кодування вхідного зображення в генероване для точної реконструкції, – заходи мають бути саме в цих напрямках.

Змагальні втрати. Як показали попередні дослідження, базовим варіантом краще взяти обчислення змагальних втрат методом найменших квадратів (як в LSGAN [38]). У дослідах треба буде порівняти ефективність із релятивістським дискримінатором [40] (вимірює ймовірність, що реальні дані є більш реалістичними, ніж створені дані, роблячи функцію втрат релятивістською) та самоконтрольованим дискримінатором [37].

Модель CycleGAN. За базову модель доцільно взяти одну з доступних публічних ноутбуків учасників змагання, яка показує високі результати.

Інша гіпотеза – спробувати застосувати згасаючий коефіцієнт для реконструкційної втрати. Для першої епохи він максимальний, а далі протягом навчання лінійно зменшується до нуля. В останню епоху діють лише змагальні втрати при оптимізації моделі, і, таким чином, як не допускається mode collapse, так і стає непотрібним кодування вхідного зображення в генероване для його відновлення.

Ще одна гіпотеза, що може бути застосована для змагання Kaggle – за прикладом ACL-GAN – додатковий дискримінатор намагається розпізнати походження генерованого знімка: його входом був справжній знімок чи перекладений іншим генератором (тобто це перше чи друге покоління генерації). Якщо це допоможе уникнути mode collapse та кодування вхідного зображення в генероване, то відсутність будь-яких інших зайвих втрат може стати перевагою.

Односторонні моделі перекладу зображень. Для експериментів за базову вибираю першу запропоновану модель – DistanceGAN, до якої можу спробувати запропонувати покращення. Для досліджень вибираю модель, яка проста і зрозуміла в реалізації, спрямована на вивчення спільного латентного простору як на рівні семантики, так і на рівні вигляду, – TraVeLGAN [14]. Модель TSIT [43] можливо буде дослідити у випадку, якщо вдасться успішно застосувати самоконтрольований дискримінатор для вирішення проблеми обмеженої кількості даних, оскільки не можу застосувати дієві способи аугментації даних або перенесення знань до цієї моделі.

Аугментація даних. До односторонніх моделей застосовую методи стохастичної аугментації даних [20] та DiffAugment [33] з використанням github.com/mit-han-lab/data-efficient-gans/blob/master/DiffAugment_tf.py.
Якщо ж такої аугментації буде недостатньо, DAG (Data Augmentation optimized for GAN) потрібно розглянути. Щодо CycleGAN, доцільно досліджувати ACCR-CycleGAN [45], замінивши змагальну функцію втрат

на метод найменших квадратів (як LSGAN [38]) або завісні змагальні втрати (hinge loss).

Крім того, у зв’язку з надзвичайною простотою реалізації, слід дослідити One2one CycleGAN [44], оскільки очікую, на відміну від авторів, повну відсутність ефекту по боротьбі з перенавчанням моделі на обмежених даних.

Попереднє навчання та передача знань. Початкова гіпотеза, що дискримінатори та генератори потрібно попередньо навчити окремо на завданнях класифікації та відновлення зображень, після розгляду сучасних застосувань перенесення знань у генеративно-змагальних мережах не видається успішною для вирішення проблеми обмеженої кількості даних. Отже, запропоновані в розглянутих статтях варіанти попереднього навчання в складі інших генеративно-змагальних мереж здаються кращим варіантом. При розгляді існуючих методів виникла ідея застосування навчання енкодера самоконтрольованого дискримінатора на великих наборах зображень. Далі навчений енкодер використовувати як частини генераторів та дискримінаторів, заморозивши його повністю.

3 ПОБУДОВА МОДЕЛЕЙ ПЕРЕКЛАДУ ЗОБРАЖЕНЬ ТА ОПИС ДОСЛІДЖЕНЬ

3.1 Середовище реалізації

Змагальна платформа Kaggle та проект Google Colab надають можливість користувачам виконувати хмарні обчислення на потужних графічних процесорах (GPU) та тензорних блоках обробки (TPU, tensor processing unit). TPU – це інтегральна схема специфічного застосування, призначена для прискорення розрахунків штучного інтелекту, що була розроблена компанією Google спеціально для машинного навчання нейронних мереж.

Набір даних – фотографій та картин Моне – зберігається як публічний Kaggle dataset у середовищі Google Cloud Storage, доступний як із ноутбуків Kaggle (Kaggle kernels), так і з ноутбуків Google Colab. Зважаючи на це, переважно моделі досліджуються в ноутбуках платформи Kaggle (безкоштовно надається кожному користувачу щотижня 30 годин обчислень TPU та близько 40 годин GPU), і додатково за потреби в середовищі Google Colab.

Мова програмування – Python, фреймоврк – TensorFlow v2.4.1.

Оцінка результату проводиться на платформі Kaggle здачею ноутбука (Kaggle kernel) у постійно триваючому змаганні «I’m Something of a Painter Myself. Use GANs to create art – will you be the next Monet?». Результат порівнюється в «живій» таблиці лідерів, у якій кожен результат тримається два місяці і видаляється.

Метрика оцінки результату – Memorization-Informed Frechet Inception Distance (MiFID), що є модифікацією від Fréchet Inception Distance (FID) із захистом від недобросовісних учасників або «недобросовісних» генеративно-змагальних моделей, що допускають «протікання» навчальних зображень до згенерованих.

На початок досліджень таблиця лідерів (182 учасники) має наступні показники метрики:

– 1 місце – 35,9;
– 2 місце – 37,6;
– 10 місце – 38,5;
– 50 місце – 41,0;
– 100 місце – 52,5.

Найкращий результат, досягнутий при попередніх дослідженнях, моделі CycleGAN – 38,8 (14 місце).

3.2 Базова модель дослідження CycleGAN

При попередніх дослідженнях було встановлено, що основною складністю отримання гарного результату є перенавчання дискримінатора на картинах Моне, і в той же час аугментація справжніх картин Моне може погіршити результат – генератор починає видавати «аугментовані» зображення. Розглядаючи доступні публічно успішні результати інших учасників змагання встановлено, що їхній підхід складається з наступних етапів:

– вибір нешкідливої аугментації (наприклад, горизонтальне дзеркальне відображення з випадковістю 0,5 та збільшення зображення до 286x286 з випадковим обрізанням знову до 256x256);

– вибір розміру навчальної партії даних, для декількох успішних спроб інших учасників встановлено batch_size = 4;

– вчасно припинити навчання на найкращому результаті, доки перенавчений дискримінатор не починає псувати генеровані зображення;

– кожна спроба навчання вдалої моделі залежить від випадкової початкової ініціалізації, тому результат FID, на якому закінчиться навчання,

може значно відрізнятися в межах [38], [42], тому потрібно декілька спроб навчання.

При цьому сама модель CycleGAN залишалася у більшості випадків базовою зі стартового кернела (Monet CycleGAN Tutorial), тому краще за все буде використати таку успішну модель за основу.

Результати експериментів з однією зі вдалих моделей CycleGAN:

– FID=39,3 для базового варіанту (lambda_cyc=10, batch_size=4, кількість епох навчання – 25 по 1407 кроків кожна, змагальні втрати обчислюються як бінарна кросентропія, регуляризаторами є реконструкційні втрати та втрати ідентичності, помірна аугментація навчальних даних, що мінімально впливає на семантику та стиль зображень – горизонтальне дзеркальне відображення та невелике обрізання країв). Згенеровані картини Моне мають кольори близькі до початкових фотографій.

– FID=46,8 – базова без втрати ідентичності. Згенеровані картини Моне за кольорами дуже відрізняються від фотографій, кольори зовсім не збереглися. Як і в попередніх дослідженнях, втрати ідентичності допомагають збереженню кольорів вхідного зображення.

– FID=43,5 базова з batch_size=16, кількість епох навчання – 7, кольори – не змінені, як у базовому варіанті.

– FID=41,6 змагальні втрати обчислюються методом найменших квадратів.

– FID=42,1 базова з завісними змагальними втратами (hinge loss).

– FID=39,6 – базова без реконструкційної втрати, одностороння, зі втратами ідентичності. Кольори вхідного зображення зберігаються в генерованому.

Висновок: внесення змін до вдалої моделі CycleGAN спричиняє погіршення результатів, але при цьому високі показники CycleGAN забезпечуються не реконструкційними втратами, а саме додатковими

втратами – у даному випадку втратами ідентичності. Тому відсутні підстави досліджувати далі модель CycleGAN, замість того отримали базову одноторонню модель із досить високою ефективністю.

3.3 Базова модель дослідження односторонніх моделей

Базова модель. При дослідженні CycleGAN вдалося отримати не описаний у статтях досить успішний варіант односторонньої моделі перекладу зображень, де єдиним регуляризатором генератора виступають втрати ідентичності – тобто якщо на вхід генератору картин Моне подати готову картину Моне – генероване зображення не має відрізнятися від вхідного.

При цьому видалення такого значного регуляризатора як реконструкційні втрати наштовхнуло на ідею, що потрібно спробувати компенсувати брак регуляризації збільшенням коефіцієнту лямбда для втрат ідентичності. Після збільшення в 10 разів (lambda_id = 100) отримано кращий результат змагання – FID = 38.5 (дев’яте місце таблиці лідерів).

DistanceGAN. Дана модель зі вхідних наборів зображень використовує по два наперед обчислених числа – середня попарна різниця зображень та средньоквадратичне відхилення попарної різниці зображень. Для регуляризації моделі статистика попарних різниць вхідних зображень навчальної партії (batch) має відповідати такій же статистиці в партії згенерованих зображень з урахуванням наперед обчислених даних по навчальних наборах. Тому якість роботи моделі залежить від розміру партій, але пам’яті GPU вистачає лише для партій розміру 16. Цього виявилося недостатньо для ефективної реалізації ідеї DistanceGAN. Було проведено досліди, де замість попарної різниці зображень регуляризація обчислювалася за середніми та середньоквадратичними відхиленнями активацій шарів Inception v3 при подачі на неї партії згенерованих зображень – з ідеєю наблизити важливу для обчислення метрики

характеристику до такої ж характеристики зображень цільової області, але успішними такі спроби не виявилися. Будь-яка статистика на партії невеликого розміру не відповідає загальному розподілу даних, і спроба штрафувати модель, щоб привести її до загальнообласної статистики, – хибна ідея.

Найкращий показник метрики змагання для DistanceGAN – 54.7 (тобто надзвичайно погано в порівнянні з базовою моделлю).

TraVeLGAN. Базова версія без застосування DiffAugmentation дала результат 47.2, при цьому явні ознаки перенавчання, і крива навчання почала коливатись, а результати погіршуватися (рисунок 3.1).

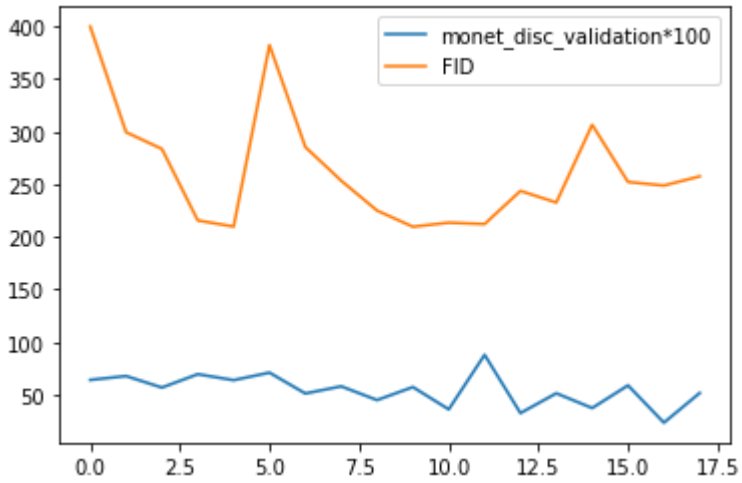

Рисунок 3.1 – Крива навчання базової версії TraVeLGAN без аугментації

3.4 Результати аугментації даних в моделі CycleGAN

ACCR-CycleGAN. Щоб розглянути вплив методів аугментації, до базової вдалої моделі CycleGAN додали втрати дискримінатора, які намагаються мінімізувати різницю значень дискримінатора для оригінального та аугментованого зображень, тобто переобладнали модель в ACCR-CycleGAN. При цьому аугментацію вхідних зображень не використовую. За основу для зрівняння взято модель CycleGAN з завісними

втратами з розміром навчальної партії 8, і з першої ж спроби («безпечна» аугментація як для вхідних зображень базової моделі) отримано результат близький до базової моделі, але без ознак перенавчання дискримінатора на таких же епохах навчання. При спробі навчання з розміром навчальної партії 4 виявилося, що кольори не зберігаються, і результат незадовільний. При розмірі навчальної партії 8 генеровані зображення краще зберігають кольори вхідних зображень.

Щодо параметрів навчання взятих зі статті [45] коефіцієнт лямбда для справжніх зображень ACCR-втрат однаковий протягом всього навчання, а лямбда для згенерованих та реконструйованих зображень має зростати від нуля до деякого значення протягом навчання. Дискримінатора має давати однакові показники для справжніх та аугментованих справжніх, а також для генерованих та аугментованих генерованих, реконструйованих та аугментованих реконструйованих зображень. З розвитком навчання і зростанням коефіцієнту лямбда значення, які дає дискримінатор, починають бути дуже схожими як для справіжніх, так і для генерованих та валідаційних справжніх зображень (рисунок 3.2).

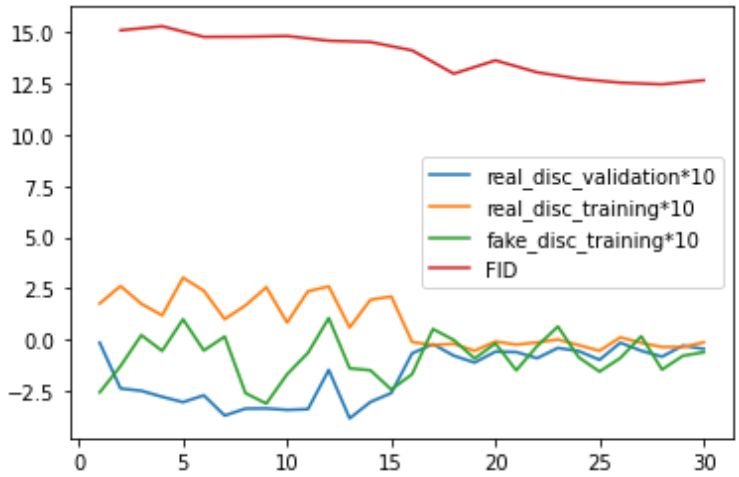

Рисунок 3.2 – При навчанні ACCR-CycleGAN значення, які видає дискримінатор стають дуже близькими для будь-якого зображення

Схоже ACCR-втрати спонукають до того, що на будь-яке зображення дискримінатор видає однакове значення. При цьому якість згенерованих зображень за метрикою змагання стає незадовільною (FID = 49.3), хоча при припиненні навчання до початку такого ефекту FID = 43.7.

ACCR-CycleGAN є особливою моделлю перекладу зображень, у якій при навчанні дискримінатора враховуються не тільки змагальні втрати, а додатковою умовою є однаковість оцінки для справжніх та аугментованих зображень.

One2one CycleGAN [44]. За основу побудови взято модель CycleGAN з завісними втратами, з результатом FID = 41.2. При видаленні одного з генераторів отримано результат FID = 45.5. Дана модель не зберігає кольорів при перетвореннях, при цьому втрати ідентичності застосувати неможливо. У той же час можливо застосувати втрати збереження середніх кольорів клаптиків, які вже було застосовано при попередніх дослідженнях CycleGAN із досить гарним результатом у порівнянні до втрат ідентичності. Після їх застосування отримано результат FID = 39.6, що краще за результат базової моделі з завісними втратами і досить близько до найкращих результатів. Але якість роботи одного генератора, незважаючи на гарну метрику змагання, є незадовільною – зворотнє перетворення картин в фотографій виконується незадовільно – чим більш реалістична картина Моне на вході – тим гірша згенерована фотографія (рисунок 3.3).

Щодо перенавчання дискримінатора, то воно спостерігається, починаючи з 25–28 епохи, так само, як і в базовій моделі.

Таким чином, встановлено, що підхід One2one CycleGAN при настільки обмеженій кількості даних, як у даному змаганні, не допоможе усунути перенавчання. У той же час перспективним залишається використання єдиного енкодера для генераторів, можливо, і дискримінаторів, який може бути наперед навчений на задачах класифікації та заморожений при навчанні генеративно-змагальної мережі. Або навіть

навчатися разом з GAN, використовуючи втрати і генераторів, і дискримінаторів.

Рисунок 3.3 – Зліва – справжні картини Моне, справа – згенеровані One2one CycleGAN фотографії не є реалістичними. При цьому зі справжніх фотографій даний генератор видає картини Моне з дуже гарною метрикою змагання FID = 39.6

DiffAugment [33]. Щоб розглянути вплив методів аугментації, до базової вдалої моделі CycleGAN додаю DiffAugment. Замість «безпечної»

аугментації вхідних зображень (випадкова дзеркальне горизонтальне відображення та випадкове обрізання країв) застосовую аугментацію «трансляція» з реалізації DiffAugment доступної на GitHub. При цьому перед подачею до функції аугментації навчальну партію вхідних зображень об'єдную з партією генерованих зображень, щоб така ж сама аугментація була виконана над ними. Після цього отримане аугментоване об'єднання знову розбираю на справжні та генеровані партії. Результат 45 епох навчання з розміром навчальної партії 16 – FID = 36.4 – третє місце таблиці лідерів з першої спроби використання моделі.

При цьому виявилася помилка в припущенні щодо можливості оцінки перенавчання за допомогою інших справжніх картин Моне, яких немає в навчальному наборі. Виявляється, що 300 картин Моне навчального набору вибрані не випадково з усіх доступних картин Моне. Схоже, вони відрізняються за якимись критеріями, і ці критерії дискримінатор розпізнає і не вважає їх за зображення навчального набору. Дискримінатор ставить їм оцінки гірші, ніж згенерованим зображенням (рисунок 3.4). Тому оцінки зображенням валідаційного набору в дійсності нічого не говорять про факт перенавчання дискримінатора на навчальних даних.

Уже з десятої епохи навчання валідаційні зображення сигналять, що їх дискримінатор не визнає за картини Моне, при цьому модель успішно навчається ще багато епох і показує один із найкращих результатів. Тому валідаційні картини Моне не можуть бути критерієм оцінювання перенавчання. Лише занадто високі оцінки справжніх картин та занадто низькі оцінки для згенерованих можуть вказувати на проблему перенавчання.

Підхід до проблеми обмеженої кількості даних DiffAugment спрацював успішно. Щоб покращити результат спробував використовувати реконструкційні втрати тільки в першій половині навчання. Але виявилося, мабуть, моделі не вистачає регуляризації в цьому випадку – результат 39,3. Залишилася ідея повторити навчання, підвищивши коефіцієнт lambda_id

втрат ідентичності, або використовувати інші втрати – збереження кольору замість ідентичності чи збереження оцінки VGG16 реконструйованого зображення замість прямих реконструкційних втрат.

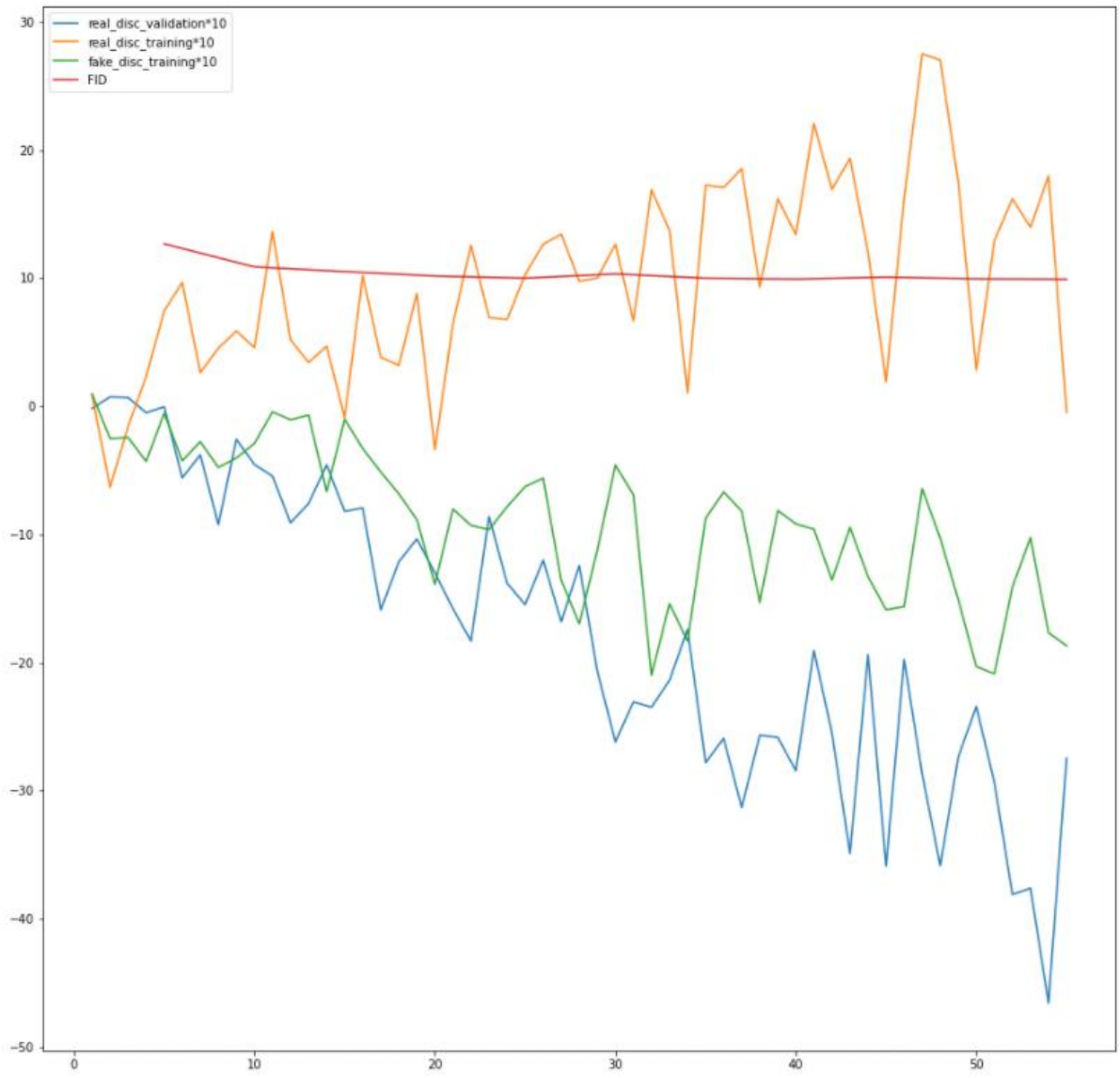

Рисунок 3.4 – Оцінка зображень дискримінатором. Оранжева лінія – оцінка справжніх картин навчального набору. Зелена лінія – оцінка згенерованих зображень. Синя лінія – оцінка справжніх картин Моне валідаційного набору даних є гіршою за оцінку згенерованих зображень. Чим далі модель навчається, тим гірше оцінює валідаційні дані

3.5 Результати попереднього навчання в моделі CycleGAN

Початкова ідея. Ідею попереднього навчання генераторів та дискримінаторів моделі CycleGAN реалізував на моделі, у якій прямі реконструкційні втрати замінив на порівняння активацій шару mixed4 моделі inceptionV3. Також попіксельне порівняння втрат ідентичності замінив на поклапиткове порівняння кольорів генерованого та вхідного зображень. Аугментацію зображень не застосовував взагалі. Дискримінатори попередньо навчав на задачі класифікації зображень (фотографія чи картина Моне подана на вхід). Для навчання генераторів на вхід подавав збільшене обрізане зображення, а на виході очікував, що генератор відновить початкове зображення.

Результат моделі в змаганні – FID = 38.6, але при повторних запусках навчання повторити високий результат не вдалося, всі спроби покращення були в межах [40 .. 44].

Також при оцінці кривої навчання було помічено, що навчання CycleGAN після попереднього навчання генераторів та дискримінаторів – більш плавне. Але коливання кривої навчання порівнюваного процесу навчання було спричинене проблемами з обмеженою кількістю зображень, яку попереднє навчання суттєво не вирішує. Хоч у випадку без попереднього навчання спостерігаються значні коливання в другій половині навчання, залежність плавності навчання від попереднього навчання встановити неможливо, оскільки друга половина навчання мало залежить від початкової ініціалізації, крім того, незважаючи на ці коливання, результати навчання в кінці приблизно однакові (рисунок 3.5).

При спробі застосувати такий же підхід до односторонньої моделі перекладу зображень виявився негативний ефект попереднього навчання. Добре розвинутий дискримінатор не давав корисної інформації до початкового генератора, і навчання взагалі не стартувало – постійний mode collapse, хоча модель без попереднього навчання не мала таких проблем.

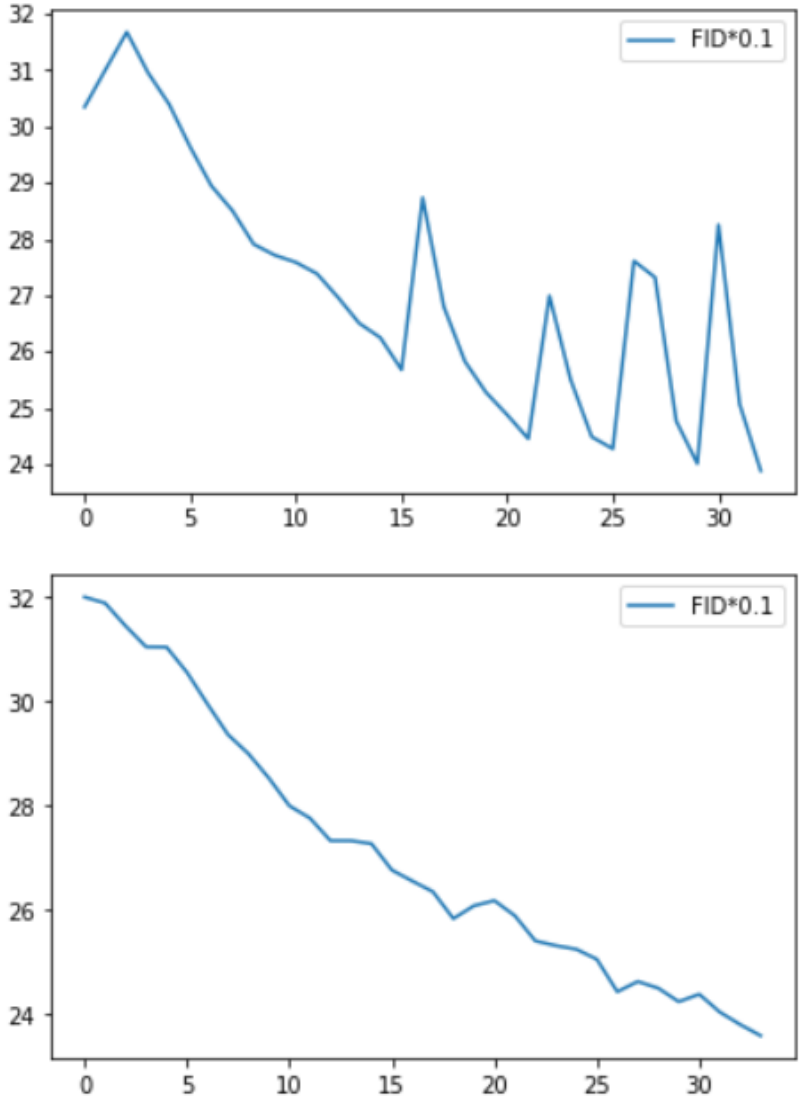

Рисунок 3.5 – Крива навчання моделі CycleGAN з попереднім навчанням генераторів та дискримінаторів (внизу), та навчання з такими ж параметрами з початкової випадкової ініціалізації (вгорі)

Інакше кажучи, якщо початкове навчання дискримінатора і може принести користь (а згідно інших досліджень після навчання можна заморозити частину шарів або всі шари, окрім нормалізації, а також навчати в складі звичайної GAN), то ефективне навчання генераторів є проблемою. Навчати в складі GAN – не вистачить часу (умови змагання Kaggle). Навчання в складі автоенкодера – не виявилося ефективним, без

заморожування шарів скоріше за все не відрізняється від випадкової ініціалізації.

Висновок. Попереднє навчання генераторів та дискримінаторів у складі автоенкодера та класифікатора має досить слабкий ефект, не вирішує проблеми малої кількості зображень однієї з областей, може мати негативний ефект в односторонніх моделях перекладу зображень. Запропоновані іншими дослідниками ідеї попереднього навчання (у складі GAN із заморожуванням частини шарів) є набагато практичнішими, але вимагають багато додаткового часу навчання.

Самоконтрольований дискримінатор SLE-GAN [37] у складі CycleGAN. Використовуючи код https://github.com/gaborvecsei/SLE-GAN, досліджую ефективність самоконтрольованого дискримінатора у випадку обмеженої кількості даних. У найефективнішу модель CycleGAN із DiffAugment було додано самоконтрольовані дискримінатори, наперед навчені на задачах відновлення вхідного зображення з глибоких шарів енкодера. Результат – навчання не почалося – дискримінатор дуже швидко набрав ефективність і повністю пропав градієнт для навчання генератора.

Клаптиковий дискримінатор з заморожуванням останнього шару. Спробував попередньо навчити CycleGAN із DiffAugment на більшому схожому наборі картин, пілся чого заморозити в дискримінаторі всі шари окрім останнього, і продовжити навчання на цільовому обмеженому наборі картин Моне. Усі досліди закінчилися невдало – донавчання дуже погіршувало результат замість покращення, спроба відкоригувати знання про розподіл даних лише останнім шаром повністю руйнувала процес навчання. Звичайно, можливо спробувати залишати незамороженими більше шарів, і інші дослідники досягали успіху, але дане дослідження не є ціллю роботи. Висновок можливо зробити відразу. Попереднє навчання на іншому наборі даних з заморожуванням шарів може бути і шкідливим, якщо недостатню кількісь шарів залишити доступною для донавчання, або ж

дискримінатор буде перенавчатися, якщо залишити незамороженою велику кількість шарів.

3.6 Результати аугментації даних в односторонніх моделях

StyleGAN2-ADA. Зважаючи на те, що архітектура StyleGAN2 може бути основою для моделі перекладу зображень, а умовами змагання Kaggle не передбачено обов'язкового використання саме моделей перекладу зображень, а підійде генерація за допомогою будь-якої генеративно-змагальної мережі, перш за все я спробував генерацію зображень за допомогою StyleGAN2.

Оскільки до StyleGAN2 опубліковано доповнення, яке дозволяє навчання на обмеженій кількості даних (StyleGAN2-ADA), використав реалізацію StyleGAN2 з https://github.com/manicman1999/StyleGAN2-Tensorflow-2.0

Виявлені наступні особливості:

а) у третині експериментів навчання закінчилося вибухом градієнтів;

б) очікується, що StyleGAN2 навчається приблизно за мільйон кроків. Для конфігурації GPU доступному на Kaggle це означає близько одного місяця часу, що зовсім не підходить для змагання з 5 годинами максимального часу навчання навіть із використанням TPU.

Після 9 годин навчання StyleGAN2 на 7038 фотографіях природи отримано результати далекі від реалістичних (рисунок 3.6).

На цьому експерименти з StyleGAN2 було припинено, але ідею адаптивної аугментації дискримінатора ADA можна буде використати в інших моделях. При цьому адаптивність не є обов'язковою умовою, оскільки час навчання наших моделей буде порівняно невеликим і підбір гіперпараметру (ймовірності застосування аугментації) не буде значною проблемою.

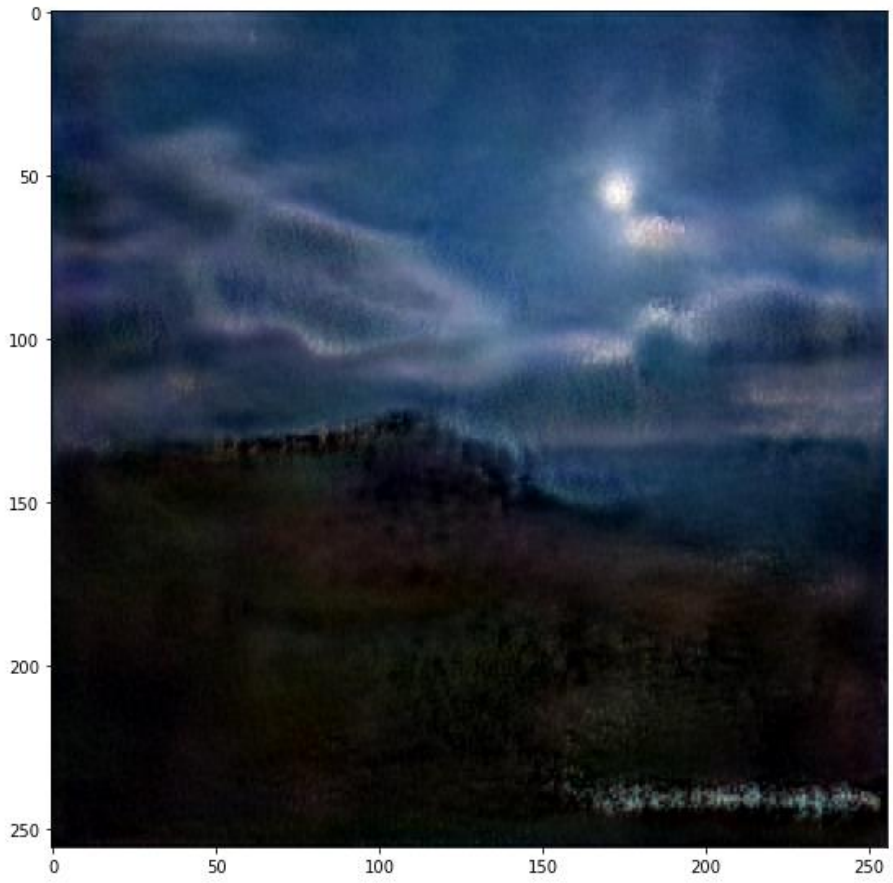

Рисунок 3.6 – Результат 9 годин навчання StyleGAN2 на 7038 фотографіях природи за допомогою GPU на платформі Kaggle

DiffAugment [33]. Цей підхід показав найкращий результат у змаганні, коли був застосований до моделі CycleGAN. При використанні до базової односторонньої моделі (IdentGAN) результат – FID = 41.8 (гірше за базовий варіант), для покращення треба підбирати гіперпараметри, але це не є метою дослідження.

DAG [34]. Особливість DAG – наявність «багатоголового» дискримінатора, тобто для кожної окремої аугментації свій окремий дискримінатор, але початкові шари в них спільні. Виявилося, що запропоновані аугментації досить прості – більшість із них є відображеннями та поворотами (повністю відтворювані), а також невелике обрізання (невідтворюване, але в експериментах добре працює). Реалізація

на задачі картин Моне показала, що модель з DAG швидко перенавчається – отримано результати близькі до базової моделі, яка використовує такі ж аугментації, але лише до навчального набору даних. Результати набагато гірші за DiffAugment, але в рази більше потребує обчислень – перспектив у змаганні не має. У той же час відмічено, що наявність мультидискримінатора створює деякий регуляризаційний ефект. Розглянувши інші статті [51], [52], де використовується подвоєння дискримінатора, вирішив дослідити модель, у якій роздвоєний дискримінатор буде навчатися за допомогою одразу двох функцій втрат: бінарної кросентропії та завісних втрат.

Двоцільовий дискримінатор з DiffAugment. Для досліджень запропонував модель (рисунок 3.7), у якій два дискримінатора (зі спільними, окрім останнього, шарами, як в [52]), навчаються за допомогою різних функцій обчислення втрат: перший – за допомогою бінарної кросентропії, а інший – за допомогою завісних втрат. Окрім цього , як в і [51], перший дискримінатор буде виставляти високі оцінки генерованим зображенням, а другий – справжнім (але в [51] дискримінатори не мають спільних шарів).

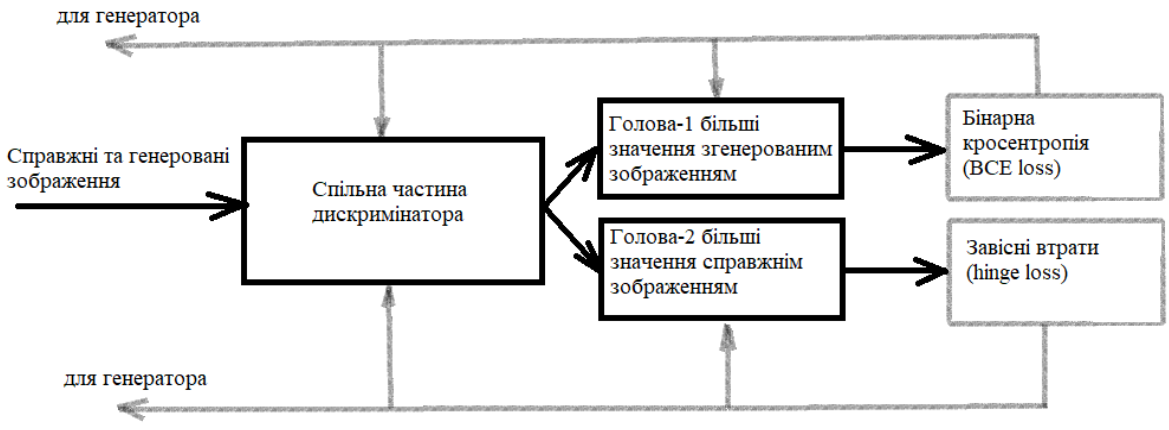

Рисунок 3.7 – Схема запропонованого двоцільового дискримінатора

Очікую, що спільне різноцільове використання більшості шарів дискримінатора завадить перенавчанню у випадку обмеженої кількості даних. В односторонній моделі перекладу зі втратами ідентичності та DiffAugment не було отримано результатів, кращих за базову односторонню модель FID = 38.6, а при використанні двоцільового дискримінатора з DiffAugment постійно отримувались результати FID = 37.6 – 37.9, і однією з особливостей навчання є те, що не потрібно підшуковувати час для ранньої зупинки – при продовженні навчання результат не починає погіршуватися, як це спостерігається у випадку неадекватного зворотнього зв'язку від перенавченого дискримінатора. Найкращий отриманий результат – 36.4 (третє місце в змаганні).

Застосування двоцільового дискримінатора в складі CycleGAN з DiffAugment показало приблизно такий же результат – FID = 35,8

Спільні шари дискримінаторів одночасно використовуються для різних цільових функцій і не можуть так легко перенавчитися, як єдиний дискримінатор чи ансамбль дискримінаторів. А останніх шарів – голів дискримінаторів, що не є спільними, не вистачає для перенавчання. В експериментах така модель дискримінатора дає постійно досить гарні результати. Хоча вони і поступаються вдалій ранній зупинці аналогічної моделі з одним дискримінатором, але на відміну від неї отримуємо повторювані результати незалежно від часу зупинки, що дає змогу підбирати особливості моделі та гіперпараметри і робити достовірні висновки.

SLE-GAN [37] з DiffAugment [33]. У вигляді генеративно-змагальної мережі використав код https://github.com/gaborvecsei/SLE-GAN, досліджуючи можливість генерації картин Моне з випадкового вектору. Дискримінатору показував справжні картини Моне, спотворені DiffAugment. Оскільки на відміну від GPU з максимальним розміром навчальної партії 32 навчання на TPU дозволяє використовувати більший, ніж в десять разів об'єм пам'яті та розмір партії 128, встановлено, що

найкращі та найбільш реалістичні результати досягаються з найбільшим розміром навчальної партії, але в таких умовах за одну епоху навчання з 1407 кроків дискримінатор 60 разів бачить навчальний набір з 300 картин, і за 10–15 епох вже перенавчається, незважаючи на DiffAugment. При спробі навчання з малим розміром навчальної партії 1–4 перенавчання не досягається, але самоконтрольований дискримінатор навчається набагато ефективніше від SLE-генератора, і в результаті за 10 епох повністю зникає градієнт для змагального навчання – дискримінатор з великою впевненістю відрізняє справжні картини від генерованих, і генератор не може покращитись.

Застосування такої ж моделі (як з простою аугментацією, так і з DiffAugment) на 7000 фотографій природи теж не дало гарних результатів: спочатку навчання йде досить ефективно, не зважаючи на те, що дискримінатор навчається набагато швидше за генератор. Але в подальшому така ж сама проблема, як і на картинах Моне – градієнт зникає від занадто ефективного дискримінатора.

У таких випадках звичайним заходом є оновлення вагових коефіцієнтів дискримінатора один раз на кілька оновлень для генератора. При розрахунку змагальних втрат як бінарної кросентропії, значення цих змагальних втрат для дискримінатора у випадку «рівності ефективності» генератора та дискримінатора дорівнює логарифму з 0.5 – тобто 0,693. За таких умов середня оцінка навчальної партії справжніх та згенерованих зображень співпадає – дискримінатор не може їх відрізнити, але в той же час дає значущий зворотній зв’язок генератору щодо його покращення. Коли ж дискримінатор починає легко відсортовувати справжні зображення від генерованих, значення втрат дискримінатора знижується, і коли воно стає меншим за 0,3 – як генератор не отримує цінного зворотнього зв’язку від дискримінатора, так і дискримінатор не розвивається від слабкого генератора.

Зважаючи на наявність такого індикатора ефективності навчання, я пропоную адаптивний підхід для керування швидкістю навчання дискримінатора, на кожному кроці навчання визначаючи штрафний коефіцієнт, на який буде помножено градієнти перед їх застосуванням до вагових коефіцієнтів. Розраховувати його пропоную за формулою:

$$\text{Штрафний\_коефіцієнт} = (\text{ втрати\_дискримінатора x } 1{,}45\ )^3. \qquad (3.1)$$

У цьому випадку залежність штрафного коефіцієнту від значення втрат зображено на рисунку 3.8.

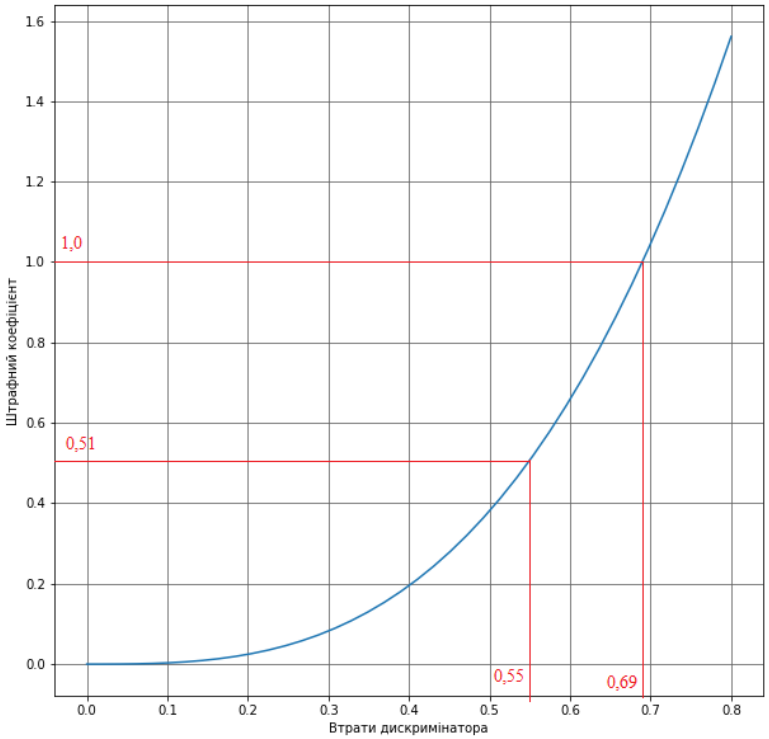

Рисунок 3.8 – Залежність штрафного коефіцієнту від значення втрат дискримінатора

Коли значення втрат дискримінатора близьке до 0,69, коефіцієнт дорівнює одиниці. Коли втрати становлять 0,55, коефіцієнт 0,51 означає, що крок навчання дискримінатора вдвічі менший за крок генератора. Це

приблизно буде означати два навчання генератора на одне дискримінатора, але при цьому очікується більша точність навчання дискримінатора, а головне – модель сама адаптується до ситуації, підбираючи відповідний крок. Коли втрати становлять 0,4, коефіцієнт – 0,2, тобто відповідає навчанню генератора 5 разів на одне навчання дискримінатора.

Але таке постійне втручання в змагальний процес навчання, коли значення втрат постійно наближуємо до 0.69, у дослідах не показує ефективності. Тому для подальших експеримнтів застосовую пом'якшений варіант (рисунок 3.9), у якому штрафний коефіцієнт вираховується лише при значеннях втрат менше 0.5 за формулою:

$$\text{Штрафний\_коефіцієнт} = \text{Min}\,[1\ ,\ (\ 2 \times \text{втрати\_дискримінатора}\ )^3]. \qquad (3.2)$$

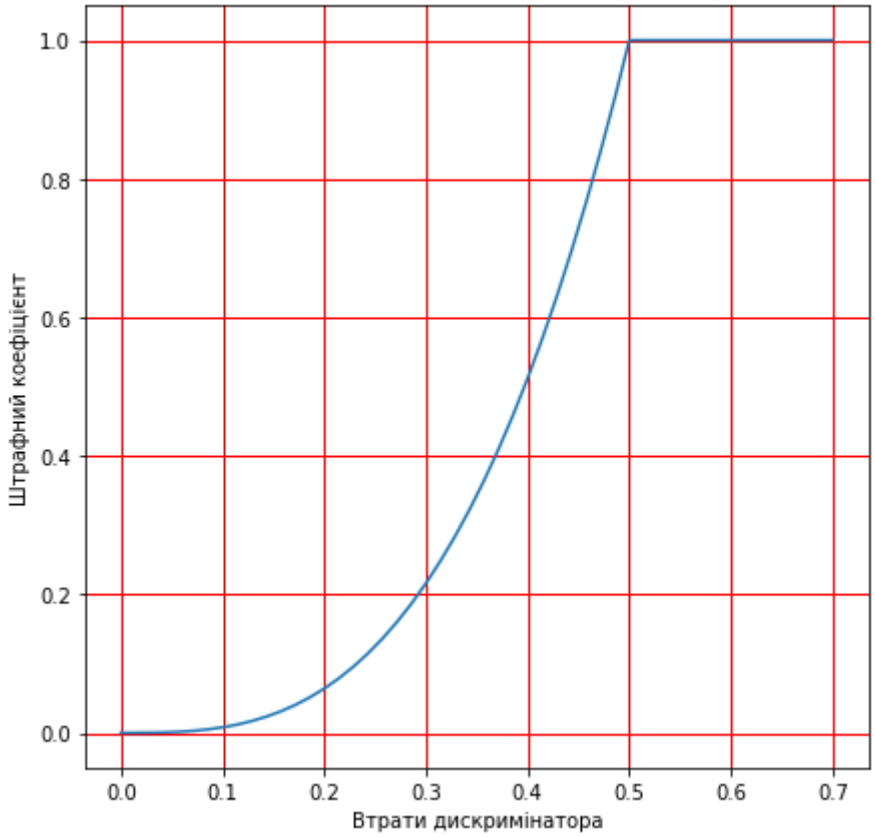

Рисунок 3.9 – Залежність штрафного коефіцієнта від значення втрат дискримінатора при застосуванні нижче значення 0.5

3.7 Результати попереднього навчання в односторонніх моделях

SLE-GAN [37]. Реалізація самоконтрольованого енкодера, з виходу якого можливо відновити форму та текстури елементів зображення, підштовхнула до ідеї використовувати єдиний наперед навчений та заморожений енкодер для генераторів та дискримінаторів. Ще більш привабливо було б і декодер генератора одночасно попередньо навчити в складі автоенкодера та заморозити, а все навчання моделі перекладу зображень проходить в операціях над виходом енкодера в латентному просторі (рисунок 3.10).

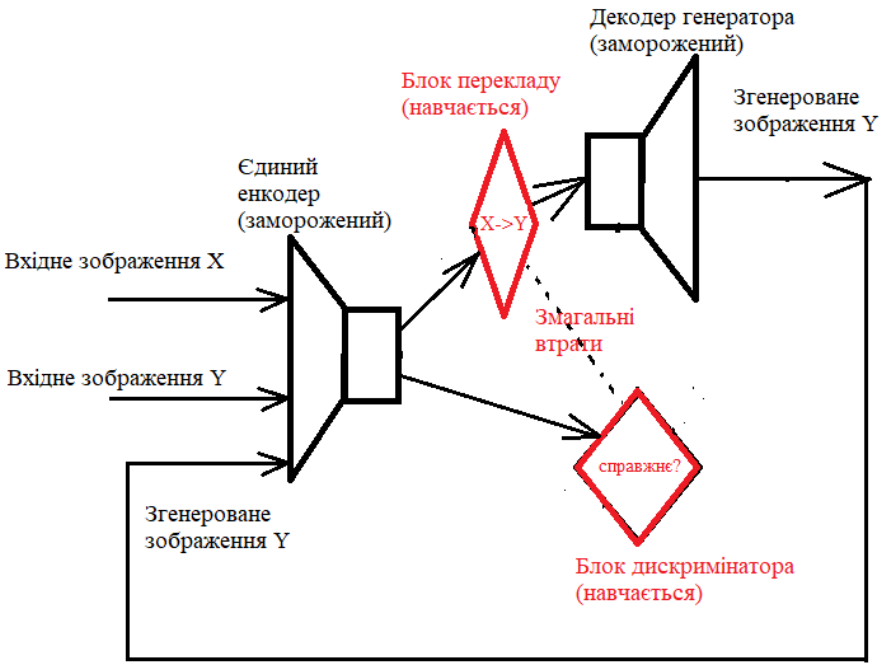

Рисунок 3.10 – Фантастична модель перекладу зображень, у якій навчаються лише блоки перекладу та дискримінації, які проводять операції над вектором представлення зображення в латентному просторі

Функція дискримінатора навчається по виходу енкодера визначати справжнє чи згенероване зображення, а функція перекладу змінює вектор,

отриманий від енкодера по зображенню в одній області, і посилає його на заморожений декодер, який видасть зображення в іншій області.

Для реалізації задуму побудований автоенкодер з самоконтрольованого дискримінатора, який видає вектор на вхід SLE-генератора. Проведено експерименти з різними налаштуваннями, але у всіх зображення на тестовому наборі даних дуже розмите – генератор не може відновити початкове зображення з вектору за час, передбачений змаганням Kaggle (рисунок 3.11).

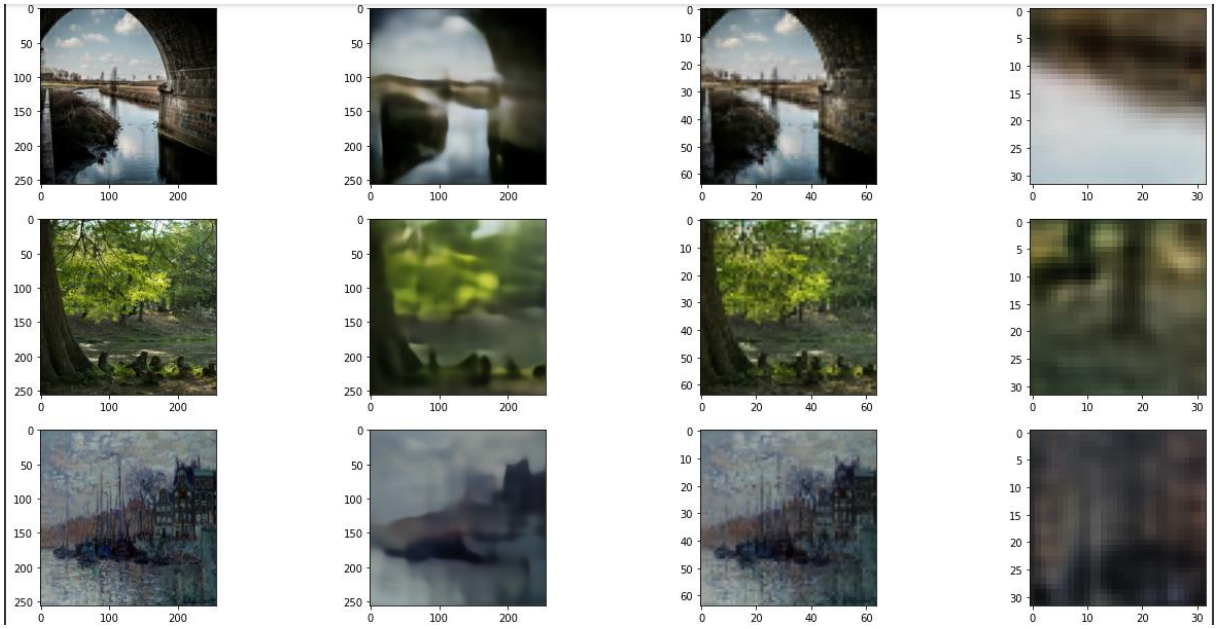

Рисунок 3.11 – Навчання самоконтрольованого енкодера та декодера генератора як автоенкодера. Перший стовпчик – вхідні зображення, другий – відновлені декодером, третій та четвертий – результати самоконтрольованого навчання (зменшене зображення та центральний клаптик). Вихідне зображення розмите

Реверсивне навчання. Якщо половину часу виділити для навчання односторонньої моделі генерувати фотографії з картин Моне, а другу половину – навпаки генерувати картини Моне з фотографій природи, можливий ефект використання попереднього навчання і відтермінування

перенавчання дискримінатора на картинах Моне. Але експерименти не показали покращення результатів. Крім того, проблема потреби в ранній зупинці навчання залишається, і таке поліпшення не є вирішенням проблеми малої кількості даних.

3.8 Результати побудованих моделей у змаганні

В таблиці 3.1 наведено порівняння результатів моделей генерації картин Моне в таблиці лідерів змагання Kaggle.

Таблиця 3.1 – Результати моделей генерації картин Моне

Модель	MIFID (Kaggle score)	Орієнтовне місце в таблиці лідерів
Кращий результат попередніх досліджень (CycleGAN з попереднім навчанням)	38,8	14
Базова CycleGAN	39,3	23
Базова одностороння	38,5	9
ACCR-CycleGAN	43.7	90
CycleGAN з DiffAugment	35,3	2 (1)
One2one CycleGAN	39,6	34
DistanceGAN	54.7	135
TraVeLGAN без аугментації	47,2	110
Базова одностороння з DiffAugment	41,8	75
Одностороння з двоцільовим дискримінатором та DiffAugment	36,4	3
CycleGAN з двоцільовим дискримінатором та DiffAugment	35,6	2
Попереднє навчання CycleGAN при донавчання з замороженим SS-енкодером лише з змагальними втратами	Mode collapse	–
Попереднє навчання з заморожуванням останнього шару дискримінатора	62,7	180

Оцінка найкращого результату відображена в таблиці лідерів змагання – друге місце, досягнуте варіантом CycleGAN з DiffAugment (рисунок 3.12). Але результати першого місця досягнуті генерацією картин Моне з інших картин Моне (згідно твердженню авторів в обговоренні), і це не відповідає умовам змагання. Тому фактично результат моделі CycleGAN з DiffAugment – тимчасове перше місце в змаганні Kaggle.

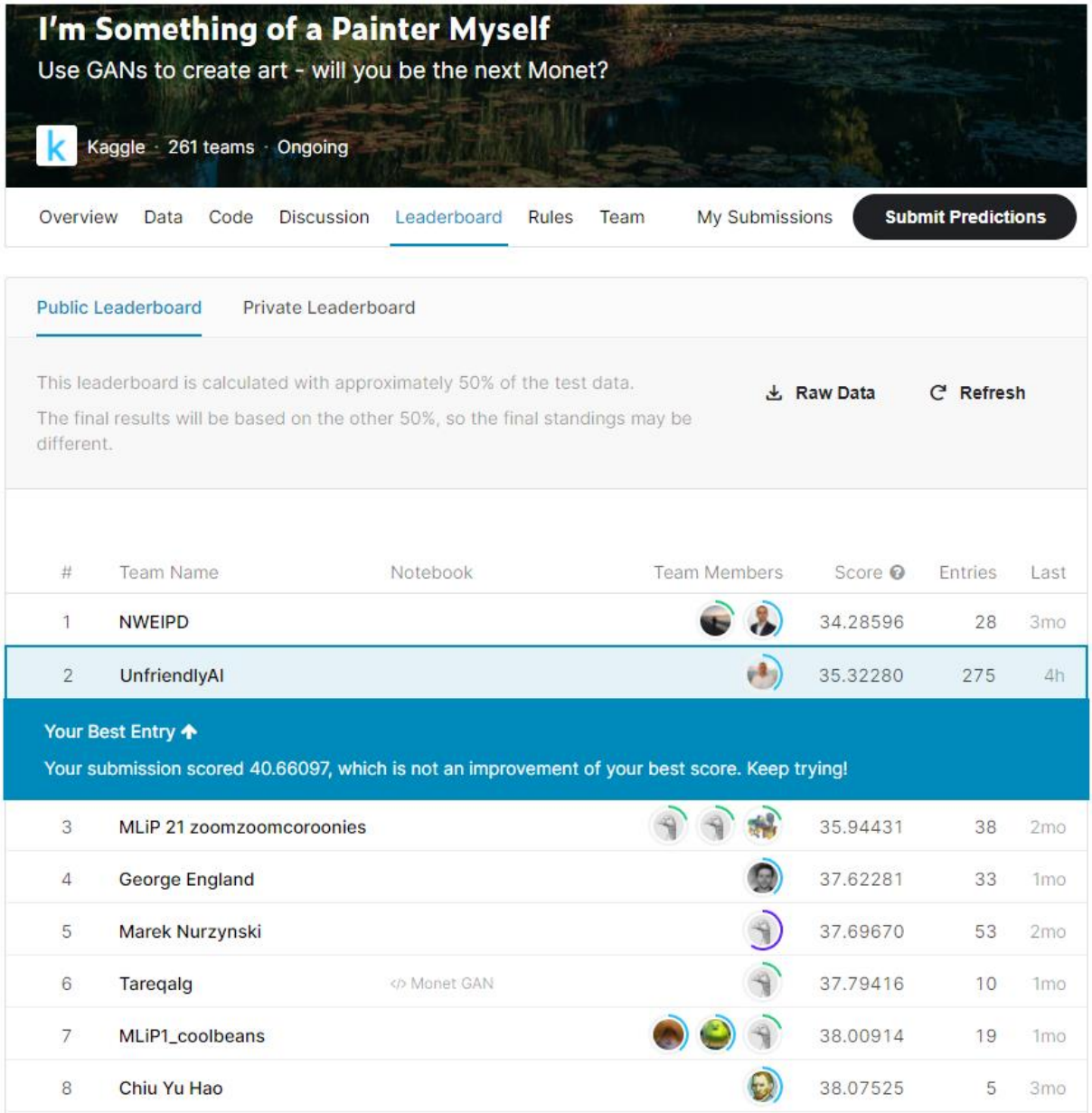

Рисунок 3.12 – Таблиця лідерів змагання Kaggle «I'm Something of a Painter Myself. Use GANs to create art – will you be the next Monet?»

3.9 Результат перевірки гіпотез досліджень

One2one CycleGAN [44]. Як очікував, вплив на перенавчання дискримінатора – відсутній. Виявилася негативна сторона єдиного генератора, що обертає зображення: у разі дуже схожих областей можливе запутування генератора – дуже реалістичну картину Моне спотворює замість перетворення в фотографію.

ACCR-CycleGAN. Відсутній захист від перетікання аугментації до генерованих зображень, додаткові втрати негативно впливають на якість оцінки дискримінатора і відповідно на навчання.

StyleGAN2-ADA. Час навчання моделі – близько місяця замість 5 годин, неможливо застосувати.

DiffAugment [33]. Дійсно виявився успішним підходом боротьби з проблемою перенавчання дискримінатора при обмеженій кількості зображень. При цьому ефективність DiffAugment обмежена, перенавчання спостерігається після деякої кількості епох.

Попереднє навчання генератора та дискримінатора. Перші значні успіхи в попередніх дослідженнях були з використанням попереднього навчання, але за відсутності сучасних технік аугментації, розроблених саме для генеративно-змагальних моделей. При використанні DiffAugment досягається настільки значне покращення, що ефект від попереднього навчання довести неможливо. Натомість виявлено можливість негативного впливу попереднього навчання, коли дискримінатор стає набагато ефективнішим за генератор, і змагальне навчання завершується невдачею або не починається зовсім. При спробі донавчити дискримінатор без заморожування шарів виявляється, що попередні знання швидко втрачаються, як і ефект від попереднього навчання.

Самоконтрольований (self-supervised) дискримінатор. Спроба самоконтрольованого навчання початкових шарів дискримінатора з їх подальшим заморожуванням не призвела до позитивного ефекту. Якщо

решта глибоких шарів, які донавчаються, буде з малою кількістю параметрів – дискримінатор не може в достатній мірі вивчити розподіл даних, а якщо ж збільшити кількість параметрів – знову стає можливим перенавчання. Жоден з експериментів із самоконтрольованим дискримінатором не дав кращих результатів за базові варіанти з клаптиковим дискримінатором.

Попереднє навчання моделі перекладу зображень на схожому більшому наборі даних. Без заморожування шарів – при переході до обмеженого набору картин Моне – «знання» швидко втрачаються, позитивного ефекту не спостерігалося. Із заморожуванням – дискримінатор може мати межу ефективності, яка пов’язана зі зменшеною кількістю параметрів, що залишилися для навчання. Підхід, застосування якого призводить до позитивного ефекту та не має шкідливих наслідків, не був знайдений.

Односторонні моделі перекладу зображень покажуть результат кращий, ніж CycleGAN. Ця гіпотеза теж не підтвердилася – найкращий результат та тимчасове перше місце в змаганні отримано через CycleGAN з DiffAugment. Отримувати стабільні результати з CycleGAN набагато простіше, ніж з односторонніми моделями.

ВИСНОВКИ

В кваліфікаційній роботі магістра розглядається задача побудови моделі перекладу зображень в умовах обмеженої кількості даних для участі в постійно діючому змаганні Kaggle.

Розглянуто сучасні моделі на основі архітектури CycleGAN та односторонні моделі перекладу зображень. Порівняно ефективність підходів до проблеми обмеженої кількості зображень: аугментації даних та перенесення навчання.

Висунуто гіпотезу, що односторонні моделі матимуть перевагу над варіантами архітектури CycleGAN, що використовують реконструкцію зображень. Але найкращий результат в змаганні показала модель на основі CycleGAN.

Запропоновано та досліджено ідею навчання генераторів та дискримінаторів моделі перекладу зображень на задачах класифікації та відновлення зображень. У результаті досліджень встановлено, що такий підхід не надає переваг у порівнянні з навчанням із випадкової ініціалізації, а в деяких випадках є шкідливим – ефективний початковий дискримінатор не дає змоги розвиватися невдалому генератору.

Запропоновано адаптивний підхід до вирішення проблеми узгодження швидкості навчання генератора та дискримінатора генеративно-змагальних мереж. На кожному кроці навчання визнається штрафний коефіцієнт, на який буде помножено градієнти перед їх застосуванням до вагових коефіцієнтів дискримінатора. Штрафний коефіцієнт вираховується від значення втрат дискримінатора, що забезпечує адаптацію швидкості процесу змагального навчання.

Запропоновано двоцільовий дискримінатор, який дозволяє уникнути негативних ефектів при перенавчанні на обмеженій кількості даних та потреби в ранній зупинці навчання для отримання кращого результату.

Об’єм проведеної роботи та додаткові результати:

– більше 1800 годин навчання на TPU та GPU наданих платформою Kaggle, окрім використання TPU платформи Google Colab;

– зроблено близько 450 спроб здачі на оцінку згенерованих картин Моне;

– опубліковано на платформі Kaggle чотири статті з кодом (kernels), які переглянуті більше 2500 разів та 41 раз використані за основу своїх робіт іншими учасниками змагання: CycleGAN without identity loss, CycleGAN with pretraining of generators and discriminators, CycleGAN is obsolete, DiffAugment is all you need.

Досягнута основна мотивуюча мета досліджень: зайняти якомога краще місце серед учасників змагання Kaggle. При цьому основна гіпотеза про те, що попереднє навчання генераторів та дискримінаторів моделей перекладу даних у складі автоенкодера та класифікатора підвищує ефективність в умовах обмеженої кількості даних – не підтверджена, оскільки використання найсучасніших технік аугментації даних у генеративно-змагальних моделях виявилося ефективнішим за запропоноване попереднє навчання. Також попереднє навчання генеративно-змагальних мереж може призводити і до негативних ефектів на результат.

Код реалізації двоцільового дискримінатора опублікований на платформі Kaggle під назвою «Two-objective discriminator» https://www.kaggle.com/unfriendlyai/two-objective-discriminator

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

2. M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. arXiv preprint arXiv:1703.00848, 2017.

3. T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), pages 1857–1865, 2017.

4. Bodyanskiy, Y.V., Deineko, A., Pliss, I., & Slepanska, V. (2019). Formal Neuron Based on Adaptive Parametric Rectified Linear Activation Function and its Learning. DCSMart.

5. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Conference on Computer Vision and Pattern Recognition. pp. 8789–8797 (2018).

6. D. Bashkirova, B. Usman, and K. Saenko. Adversarial self-defense for cycle-consistent gans. In Advances in Neural Information Processing Systems, pages 635–645, 2019.

7. Arjovsky, Martín et al. «Wasserstein GAN.» ArXiv abs/1701.07875 (2017).

8. Xu, X. et al. «MCMI: Multi-Cycle Image Translation with Mutual Information Constraints.» ArXiv abs/2007.02919 (2020).

9. Rui Zhang, Tomas Pfister, and Jia Li. Harmonic unpaired image-to-image translation. arXiv preprint arXiv:1902.09727, 2019.

10. Jia, Z., Yuan, B., Wang, K., Wu, H., Clifford, D., Yuan, Z., & Su, H. (2020). Lipschitz Regularized CycleGAN for Improving Semantic Robustness in Unpaired Image-to-image Translation. ArXiv, abs/2012.04932.

11. S. Benaim and L. Wolf. One-sided unsupervised domain mapping. In NIPS, 2017.

12. Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2427–2436, 2019.

13. Taesung Park, Alexei A Efros, Richard Zhang, and JunYan Zhu. Contrastive learning for unpaired image-to-image translation. In European Conference on Computer Vision, pages 319–345. Springer, 2020.

14. Amodio, M., Krishnaswamy, S.: Travelgan: Image-to-image translation by transformation vector learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8983–8992 (2019).

15. X. Liu, S. Zhang, H. Liu, X. Liu, and R. Ji. Cerfgan: A compact, effective, robust, and fast model for unsupervised multi-domain image-to-image translation. arXiv preprint arXiv:1805.10871v2, 2018.

16. Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017b.

17. Wang, Y., Yu, L., & Weijer, J.V. (2020). DeepI2I: Enabling Deep Hierarchical Image-to-Image Translation by Transferring from GANs. *ArXiv, abs/2011.05867*.

18. Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In ECCV, pages 218–234, 2018.

19. Y. Wang, A. Gonzalez-Garcia, D. Berga, L. Herranz, F. S. Khan, and J. van de Weijer. Minegan: effective knowledge transfer from gans to target domains with few images. In CVPR, 2020.

20. Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. arXiv preprint arXiv:2006.06676, 2020a.

21. D. Zhang and A. Khoreva. PA-GAN: Improving GAN training by progressive augmentation. In Proc.NeurIPS, 2019.

22. A. Noguchi and T. Harada. Image generation from small datasets via batch statistics adaptation. In ICCV, 2019.

23. Yijun Li, Richard Zhang, Jingwan Cynthia Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. Advances in Neural Information Processing Systems, 33, 2020.

24. Miaoyun Zhao, Yulai Cong, and Lawrence Carin. On leveraging pretrained gans for generation with limited data. In International Conference on Machine Learning, 2020.

25. Hajar Emami, Majid Moradi Aliabadi, Ming Dong, and Ratna Babu Chinnam. 2019. SPA-GAN: Spatial Attention GAN for Image-to-Image Translation. ArXiv:1908.06616.

26. O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.

27. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018).

28. Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In ICLR, 2019.

29. M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In Proc. ICLR, 2017.

30. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In Proc. NIPS, 2014.

31. Z. Zhao, S. Singh, H. Lee, Z. Zhang, A. Odena, and H. Zhang. Improved consistency regularization for GANs. CoRR, abs/2002.04724, 2020.

32. S. Mo, M. Cho, and J. Shin. Freeze the discriminator: a simple baseline for fine-tuning GANs. CoRR, abs/2002.10964, 2020.

33. S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, «Differentiable augmentation for data-efficient gan training,» in NeurIPS, 2020.

34. N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung. On data augmentation for GAN training. CoRR, abs/2006.05338, 2020.

35. Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang, «Image augmentations for gan training,» arXiv preprint arXiv:2006.02595, 2020.

36. Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In International Conference on Learning Representations (ICLR), 2018.

37. Liu, B., Zhu, Y., Song, K., & Elgammal, A. (2021). Towards Faster and Stabilized GAN Training for High-fidelity Few-shot Image Synthesis. ArXiv, abs/2101.04775.

38. X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, «Least squares generative adversarial networks,» in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2794–2802.

39. Pang, Y., Lin, J., Qin, T., & Chen, Z. (2021). Image-to-Image Translation: Methods and Applications. *ArXiv, abs/2101.08629*.

40. A. Jolicoeur-Martineau, «The relativistic discriminator: a key element missing from standard gan,» arXiv preprint arXiv:1807.00734, 2018.

41. J. Kim, M. Kim, H. Kang, and K. Lee, «U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation,» arXiv preprint arXiv:1907.10830, 2019.

42. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2921-2929.

43. L. Jiang, C. Zhang, M. Huang, C. Liu, J. Shi, and C. C. Loy, «Tsit: A simple and versatile framework for image-to-image translation,» arXiv preprint arXiv:2007.12072, 2020.

44. Z. Shen, S. K. Zhou, Y. Chen, B. Georgescu, X. Liu, and T. S. Huang, «One-to-one mapping for unpaired image-to-image translation,» The IEEE Winter Conference on Applications of Computer Vision, pp. 1170– 1179, 2019.

45. Ohkawa, T., Inoue, N., Kataoka, H., & Inoue, N. (2020). Augmented Cyclic Consistency Regularization for Unpaired Image-to-Image Translation. *ArXiv, abs/2003.00187*.

46. Nizan, O., Tal, A.: Breaking the cycle – colleagues are all you need. In: arXiv preprint arXiv 1911.10538 (2019).

47. Zhao, Y.; Wu, R.; and Dong, H. 2020. Unpaired Imageto-Image Translation using Adversarial Consistency Loss. arXiv preprint arXiv:2003.04858.

48. Zuo, Z., Xu, Q., Zhang, H., Wang, Z., Chen, H., Li, A., Zhao, L., Xing, W., & Lu, D. (2020). Multimodal Image-to-Image Translation via Mutual Information Estimation and Maximization. ArXiv, abs/2008.03529.

49. Mario Lucic, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. Highfidelity image generation with fewer labels. arXiv preprint arXiv:1903.02271, 2019.

50. Pan, Junkun. «Paint like Vincent Van Gogh: Artistic Style Generator Using CycleGAN and VGG19.» (2020).

51. Nguyen, T. et al. «Dual Discriminator Generative Adversarial Nets.» *NIPS* (2017).

52. Hoang, Q., Nguyen, T., Le, T., & Phung, D.Q. (2017). Multi-Generator Generative Adversarial Nets. *ArXiv, abs/1708.02556*.