

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»  
УДК 004.855

«До захисту допущено»

Завідувач кафедри  
\_\_\_\_\_ І.Р. Пархомей  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2019 р.

## **Магістерська дисертація**

**на здобуття ступеня магістра**

зі спеціальності 126 «Інформаційні системи та технології»

на тему: Дослідження генеративних можливостей інтелектуальних  
робототехнічних систем

Виконав: студент другого курсу, групи ІК-82мп  
(шифр групи)

\_\_\_\_\_ Гарькавий Сергій Вікторович \_\_\_\_\_

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник доцент, к.т.н., Олійник В.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант НК к.т.н., доцент, Пасько В.П.

(назва розділу)

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент \_\_\_\_\_

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ І.Р. Пархомей

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

Гарькавому Сергію Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Дослідження генеративних можливостей інтелектуальних робототехнічних систем» \_\_\_\_\_

науковий керівник дисертації к.т.н., Олійник В.В. \_\_\_\_\_ ,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «28» Жовтня 2019 р. № 3770-с

2. Термін подання студентом дисертації \_\_\_\_\_ 18.11.2019

3. Об'єкт дослідження – інтелектуальна робототехнічна система.

4. Предмет дослідження – генеративні можливості інтелектуальної робототехнічної системи.

5. Перелік завдань, які потрібно розробити – аналіз існуючих підходів до вирішення проблем генеративних можливостей інтелектуальних робототехнічних систем; аналіз підходів до оцінки генеративних моделей; формулювання моделі інтелектуальної системи; дослідження результатів роботи генеративної системи

6. Орієнтовний перелік ілюстративного матеріалу: шість плакатів

7. Орієнтовний перелік публікацій: дві публікації

## 8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння.	Лісовиченко О.І.		
НМ	Пасько В.П.		

9. Дата видачі завдання \_\_\_\_\_

### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	12.09.2019 р.	
2	Постановка задачі	15.09.2019 р.	
3	Аналіз інформаційного забезпечення	22.09.2019 р.	
4	Аналіз алгоритмічного забезпечення	26.09.2019 р.	
5	Розробка алгоритмічного забезпечення	15.10.2019 р.	
6	Розробка програмного забезпечення	10.11.2019 р.	
7	Маркетинговий аналіз стартап-проекту	12.11.2019 р.	
8	Висновки	16.11.2019 р.	

Студент

\_\_\_\_\_

(підпис)

Гарькавий С.В.

(ініціали, прізвище)

Науковий керівник дисертації

\_\_\_\_\_

(підпис)

Олійник В.В.

(ініціали, прізвище)

## АНОТАЦІЯ

У роботі проведено дослідження генеративних можливостей інтелектуальної робототехнічної системи, розглянуто генерацію новизни нейронною мережею даної системи, розглянуто особливості існуючих підходів до генерації новизни та зроблено огляд підходів до підвищення ефективності створення новизни.

Об'єктом дослідження є інтелектуальна робототехнічна система.

Було досліджено підходи до генерування новизни, зокрема зображень. Також досліджено підходи до оцінки генерації новизни створеної генеративними моделями і створено рішення для генерації зображень логотипів. Така система може бути використана для підвищення ефективності створення логотипів брендів.

Ключові слова: інтелектуальна робототехнічна система, методи оцінки генеративних моделей, нейронна мережа, генерація новизни.

Розмір пояснювальної записки – 84 аркушів, містить 23 ілюстрації, 28 таблиць, 2 додатки.

## ABSTRACT

In this work, the research on the generative abilities of the intelligent robotic system was conducted.

The object of the study is the intelligent robotic system.

The approaches to the novelty generation and particularly images was studied. Also the approaches to novelty generated by a generative models estimation. In addition, the system for logo images generation was developed. This system can increase the effectiveness of a logo for brands creation process.

Keywords: intelligent robotic system, generative models estimation, neural network, novelty generation.

Explanatory note size - 84 pages, contains 22 illustrations, 28 tables, 2 applications.

Пояснювальна записка  
до магістерської дисертації

на тему: *Дослідження генеративних можливостей  
інтелектуальних робототехнічних систем*

Київ – 2019 року

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	5
1. АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ.....	6
1.1. Мета дослідження .....	6
1.2. Аналіз задачі генерації нових об'єктів .....	8
1.3. Особливості, проблеми, складнощі дослідження генерації.....	11
1.4. Оцінка генеративних моделей.....	19
1.4.1. Непрямі контрольовані показники.....	19
1.4.2. Об'єктивність .....	20
1.4.3. Оцінювання новизни поза розподілом за допомогою позакласової оцінки .....	21
1.4.4. Позакласова оцінка .....	21
1.4.5. Людське судження або візуальний тест Тьюрінга .....	23
1.5. Аналіз підходів до рішення задачі генерації .....	23
1.5.1. Прямі методи.....	25
1.5.2. Ієрархічні методи .....	25
1.5.3. Ітераційні методи .....	26
1.5.4. Інші методи .....	28
1.6. Аналіз існуючих рішень .....	28
1.6.1. Машинне навчання .....	28
1.6.2. Глибоке навчання і генеративні моделі .....	30
1.6.3. Генеративні змагальні мережі .....	32
1.6.4. Обчислювальна творчість та генетичні алгоритми.....	37
1.6.5. Теорія дизайну .....	38
Висновки до розділу .....	38
2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ГЕНЕРАТИВНИХ МОДЕЛЕЙ ІНТЕЛЕКТУАЛЬНИХ РОБОТОТЕХНІЧНИХ СИСТЕМ .....	40
2.1. Генеративна змагальна мережа .....	40
2.1.1. Дискримінатор генеративної змагальної мережі .....	42
2.1.2. Генератор генеративної змагальної мережі.....	43
2.1.3. Навчання генеративної змагальної мережі.....	46

2.2. Проектування моделі генеративної мережі .....	47
Висновки до розділу .....	49
<b>3. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>50</b>
3.1. Вибір середовища та засобів розробки .....	50
3.2. Огляд навчальних даних.....	52
3.3. Навчання моделі.....	55
3.4. Результати роботи генеративної мережі.....	56
3.5. Експериментальне дослідження генеративної мережі.....	57
3.5.1. Налаштування параметрів мережі.....	59
3.5.2. Дискримінатор генеративної мережі .....	59
3.6. Результати експерименту та їх аналіз .....	60
3.7. Перспективи використання мережі та методів оцінки .....	62
Висновки до розділу .....	62
<b>4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТ .....</b>	<b>64</b>
4.1. Опис ідеї проекту.....	66
4.2. Технологічний аудит ідеї проекту .....	67
4.3. Аналіз ринкових можливостей запуску стартап-проєкту.....	68
4.4. Розроблення ринкової стратегії проекту .....	74
4.5. Розроблення маркетингової програми стартап-проєкту.....	76
Висновки до розділу .....	78
<b>ВИСНОВОК .....</b>	<b>79</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ .....</b>	<b>81</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GAN (Generative Adversarial Network) – генеративна змагальна мережа.

LLD (Large Logo Dataset) – великий набір даних логотипів

SGD (Stochastic Gradient Descent) – стохастичний градієнтний спуск

CC (Computational Creativity) – обчислювана креативність

CNN (Convolutional Neural Network) – згорткова нейронна мережа

## ВСТУП

В останні роки значні досягнення в глибоких нейронних мережах дозволили створити новаторські технології, такі як автомобілі з автопілотом та голосові особисті помічники. Майже всі успіхи глибоких нейронних мереж стосуються прогнозування, тоді як початкові прориви виходили з генеративних моделей. Сьогодні, хоча ми маємо дуже потужні методи глибокого генеративного моделювання, ці методи істотно використовуються лише для прогнозування або для генерації відомих об'єктів (тобто, коли є якісні зображення відомих класів): будь-який згенерований об'єкт, який невідомий апріорі, розглядається як некоректний або як помилковий. Іншими словами, коли прогнозування здається єдиною можливою метою, новизна розглядається як помилка, тому дослідники докладають багато зусиль щоб її усунути. Ця дисертація захищає точку зору, що замість того, щоб намагатися усунути цю новизну, ми повинні вивчити її а також дослідити генеративний потенціал глибоких мереж щодо створення корисної новизни. Дисертація присвячена вивченню генерації новизни пов'язаної з моделями знань, орієнтованих на дані, завдяки глибоким генеративним нейронним мережам.

Ключовим результатом є підтвердження того, що існуючі моделі, що мають традиційну функцію оптимізації, дійсно можуть генерувати невідомі об'єкти. Це також показує, що, хоча оптимізація, така як максимальна правдоподібність, призначена для усунення новизни, на практиці створює новизну. Через ряд експериментів вивчається поведінка цих моделей і новизна, яку вони створюють. Зокрема, пропонується нове налаштування завдань і показників для вибору хороших генеративних моделей. Нарешті, дисертація завершується серією експериментів, що висвітлюють характеристики моделей, які можуть демонструвати новизну. Експерименти показують, що розрідженість, рівень шуму, і обмеження ємності мережі виключає новизну і що моделі, які краще підходять для розпізнавання новизни також підходять і для генерації новизни.

# 1. АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ

## 1.1. Мета дослідження

Системи штучного інтелекту (AI) розглядаються як інструменти, які глибоко змінюють людське суспільство через автоматизацію, зниження витрат і можливість виконання завдань, які добре виконуються людьми (наприклад, розпізнавання образів, контроль, абстракція), а також завдання, які виходять за межі людських можливостей (наприклад, прогноз погоди). AI є широким полем для досліджень і включає кілька підрозділів, серед яких планування, навчання, сприйняття, контроль (наприклад, рух), комунікація (наприклад, через природну мову).

В даний час, завдяки емпіричним успіхам, машинне навчання розглядається як флагман штучного інтелекту, де майже всі успішні програми стосуються прогнозування. Для декількох відомих практичних цілей і в рамках інженерної парадигми завдання прогнозування надзвичайно цінне, оскільки парадигма прогнозування дає доступ до новаторських технологій, таких як перше покоління промислових автономних транспортних засобів або особистих помічників з голосом (наприклад, Siri, Amazon Echo). З точки зору досліджень, можна працювати над методами прогнозування, оскільки вони засновані на чітко визначеній і добре зрозумілій системі статистичної теорії навчання, що дає як розуміння, так і засоби для порівняння методів і вимірювання прогресу.

Проте, існуючі моделі штучного інтелекту, засновані на прогнозуванні, ігнорують і не відтворюють кілька інших істотних рис людського інтелекту, таких як емоції, інтуїтивна фізика, інтуїтивна психологія, вміння вчитися на малій кількості прикладів. Нездатність врахувати ці істотні людські пізнавальні риси машинним навчанням робить його мало цікавим для експериментального вивчення системи штучного інтелекту, адже це може виходити за рамки простої (однак потужної) парадигми навчання. Важливим є те, що досягнення більш просунутих рівнів штучного інтелекту, здатних

відтворювати ці характеристики, здається малоімовірним, просто тому, що спільнота з вивчення машинного навчання не враховує ці проблеми і залишається зосередженою на методах прогнозування.

Однією з таких характеристик людського інтелекту, що лежить в основі цього дослідження, є здатність людини створювати новизну. Можливість людини створювати новизну проявляється в багатьох областях, таких як розробка нових продуктів (наприклад, в машинобудуванні), мистецтві (наприклад, картини, музика) і інтелектуальних творах (наприклад, наукові теорії, романи). Хоча вплив і актуальність генерованої новизни констатовано в багатьох книгах з наукової літератури (когнітивна психологія, інноваційний менеджмент, теорія дизайну), штучний інтелект в загальному, і машинне навчання зокрема, залишилося досить нечутливим до теми.

Можливо, головною причиною такої відсутності інтересу є відсутність теоретичної основи, що чітко визначає, що є новизною і яким є завдання де було б корисним генерування новизни за допомогою методів машинного навчання. Такий фреймворк забезпечив би подальшу продуктивність досліджень з даної теми.

Головною метою дослідження цієї області є сприяння створенню такого фреймворку шляхом закладання підґрунтя, на основі якого генерування новизни може систематично вивчатися в області машинного навчання. Відправною точкою і початковою мотивацією цього дослідницького проекту є генеративні моделі в машинному навчанні. Хоча в більшості літератури з машинного навчання йдеться про прогнозування (і дискримінативні моделі), в тій же літературі продукуються генеративні моделі, які можуть генерувати дані. Проте в даний час більшість робіт із застосуванням генеративних моделей використовують їх для поліпшення моделей прогнозування. Більше того, навіть коли вони використовуються для генерації, поточні генеративні моделі в основному, навчаються генерувати відомі об'єкти, роблячи їх практично обмеженими для створення справжньої новизни. Вони, таким

чином, марні для розуміння механізму, за допомогою якого відбувається генерування новизни.

Отже, метою роботи є підвищення ефективності застосування інтелектуальних робототехнічних систем для розв'язання задач генерації нових об'єктів, зокрема зображень. Об'єктом дослідження є інтелектуальна робототехнічна система. Предметом дослідження є генеративні можливості інтелектуальної робототехнічної системи.

Під інтелектуальною робототехнічною системою, або ІРС, у даній роботі розуміється така технічна або програмна система, яка має змогу вирішувати задачі, які традиційно вважають творчими. ІРС це інформаційно-обчислювальна система з інтелектуальною підтримкою, що вирішує задачі без втручання людини. Це автоматизовані технічні систем з використанням напрацювань штучного інтелекту. У сфері штучного інтелекту таких роботів ще називають інтелектуальними агентами. Під цим терміном розуміють розумні сутності, що спостерігають за навколишнім середовищем і діють у ньому, при цьому їхня поведінка раціональна в тому розумінні, що вони здатні до розуміння і їхні дії завжди спрямовані на досягнення якої-небудь мети.

## 1.2. Аналіз задачі генерації нових об'єктів

Останні досягнення в машинному навчанні відновили інтерес до штучної творчості. Такі дослідження, як «глибокий сон» та «передача стилю» викликали загальний публічний інтерес і дали сильний поштовх для використання моделей глибокого навчання в дослідженнях обчислювальної творчості. Креативність в машинному навчанні поступово стає законним піддоменом із появою виділених дослідницьких груп, таких як Google Маджента.

Існує велика кількість робіт з вивчення творчості обчислювальними методами. Велика різноманітність методів від систем, заснованих на правилах, до еволюційних обчислень була використана для безлічі дослідницьких питань. Порівняно з цими методами, методи машинного навчання дають

важливу перевагу: вони дають можливість вивчати творчість по відношенню до знань (тобто творчість, керована знаннями). Тим не менш, для того щоб краще висвітлити цікаві точки дослідження обчислювальної творчості для спільноти з машинного навчання та дозволити дослідникам машинного навчання отримувати систематичні відповіді на проблеми обчислювальної творчості, важливо точно відповісти на три питання:

1. Що розуміється під генеруванням новизни.
2. Як можна генерувати новизну.
3. Як можна оцінити модель, що генерує новизну.

У межах машинного навчання шукати відповіді до цих питань буде закономірно в генеративному моделюванні. Основний підхід до генеративного моделювання передбачає, що є явища, що генерують спостережувані дані і прагнуть побудувати модель цих явищ, яка, наприклад, дозволила б генерувати подальші спостереження. Традиційне генеративне моделювання розглядає лише генерацію в розподілі, де метою є генерування об'єктів із категорії або категорій об'єктів, що вже спостерігаються. Щодо генерації новизни, це може розглядатися як породження відомих типів об'єктів. Хоча генерація в розподілі має свою цінність, ця перспектива суворо обмежена з точки зору творчості: навряд чи можна придумати літаючий корабель, створивши зразки з розподілу кораблів та літаючих об'єктів.

Дослідники творчості стверджують, що суть творчого процесу – це можливість будувати нові категорії на основі вже відомих категорій. Однак творчість виходить за рамки простого комбінованого дослідження: мова йде про генерування раніше невідомих, але значущих (або цінних) нових типів об'єктів, що використовують раніше набуті знання. У цій перспективі генерування новизни має на меті продемонструвати приклад з нового типу. Ця мета називається генерація поза розподілом і вона виходить за рамки того, що можна формалізувати в рамках традиційної теорії навчання, навіть якщо вивчення існуючих типів є найважливішою частиною процесу.

З точки зору машинного навчання, генерування предмета невідомого типу не є чітко визначеною проблемою, і дослідження в генеративному моделюванні зазвичай має на меті усунути цю можливість взагалі, оскільки це розглядається як джерело нестабільності, що призводить до помилкових зразків. У певному розумінні процедура відбору зразків призначена для усунення будь-якої можливості отримати зразок поза розподілом, що є проблемою для вивчення генерування новизни методами машинного навчання.

Можливо, найважливішою проблемою є оцінка того, що є хорошою моделлю для генерування поза розподілом. З одного боку, є прагнення створювати змістовну новизну, а не тривіальний шум. З іншого боку, це являється генеруванням невідомих об'єктів, тому традиційні метрики, що базуються на концепції ймовірності не підходять для даної задачі, оскільки новизна в сенсі генерування поза розподілом є мало ймовірною за визначенням. Ця відсутність показників перешкоджає відповіді на перші два питання.

У цій роботі стверджується, що для ширшого прийняття генерації новизни як теми для наукового вивчення в рамках машинного навчання, необхідний новий принцип, що дав би змогу здійснити таку оцінку, а отже, і експериментальні дослідження. У традиційному контексті навчання з наглядом головний принцип проектування – це мінімізація помилки на тестовому наборі «затримки». У роботі пропонується набір методів, де генеративний потенціал моделей можна оцінити, «затримуючи цілі класи», імітуючи таким чином невідому, але значущу новинку. Тоді мета генератора новизни – використання навчальних класів для побудови моделі, яка може генерувати об'єкти з майбутніх (затриманих) класів, невідомих під час навчання.

### 1.3. Особливості, проблеми, складнощі дослідження генерації

Зазвичай, коли мова йде про нейронні мережі, які застосовують до зображень, мають на увазі нейронні мережі, що розпізнають зображення, тобто про нейронні мережі, які беруть зображення, аналізують значення пікселів цього зображення, застосовують до зображення набір перетворень і подають на вихід якусь інформацію, наприклад, клас об'єкта, який на ньому зображений.

В останні роки все більше уваги присвячено нейронним мережам, які вирішують зворотні завдання, тобто завдання синтезу зображень. Таке завдання полягає в наступному: мережа повинна переглянути велику кількість зображень, що містять певні класи об'єктів, і, вивчивши, що означає певний клас об'єктів, вона повинна навчитися створювати нові зображення. Важливо, що ці нові зображення повинні бути, по-перше, реалістичними з точки зору людини. А по-друге, повинні істотно відрізнятися від тих об'єктів, які нейронна мережа бачила у процесі навчання.

Це завдання є цікавим з декількох причин. По-перше, це завдання є складнішим ніж завдання розпізнавання. Для того щоб генерувати нові зображення об'єктів певних класів, мережа повинна розуміти, що собою являють ці класи об'єктів, краще, ніж для того, щоб розпізнавати ці класи.

Нейронні мережі уже зараз вміють розпізнавати класи об'єктів в багатьох умовах не гірше, ніж людина. Найкраща система, яка синтезує зображення певного класу, просто подивившись перед цим набір інших зображень цього ж класу, не досягає особливо хороших результатів. Зображення виходять або розмиті, або на них є істотні артефакти, спотворення, або ці зображення виходять занадто схожими на ті, які нейронна мережа аналізувала в процесі навчання. Тобто дана задача являє собою новий виклик в області штучного інтелекту.

На даний момент часу майже всі підходи вирішують цю задачу наступним чином: береться багатовимірний розподіл (традиційний,

нормальний гаусовий розподіл в багатовимірному просторі, або рівномірний розподіл на сфери). Далі нейронна мережа навчається, викривляє багатовимірний простір так, щоб простий багатовимірний розподіл ставав складним, і в ідеалі описував необхідне зображення. Якщо створити таку нейронну мережу, яка правильним чином викривляє простір, тоді можна синтезувати зображення дуже просто: взяти вихідний рівномірний розподіл, витягнути з нього точку, пропустити її через нейронну мережу і на виході отримати синтезоване зображення з об'єктом певного класу.

Концептуально задача виглядає досить простою, але проблема полягає в тому, як саме вимірювати успішність нейронної мережі, як саме виміряти, що розподіл, який мережа видає, збігається з розподілом, який нам потрібно отримати. Проблема в тому, що розподіл справжніх, істинних зображень задається у вигляді 10 тисяч, 100 тисяч або 1 мільйон точок. Це означає, що розподіл, який мережа видає, ми можемо оцінювати по досить невеликим наборам даних, скажімо сотні або тисячі точок.

У підсумку ключове завдання, яке з'являється в області генерації зображень за допомогою нейронної мережі, полягає в тому, щоб порівнювати імовірнісні розподіли, задані невеликою кількістю зразків. Ця задача підлягає компетенції області математичної статистики, і в останні роки дослідники в області нейронних мереж повернулися до розробки або, навіть, до перевідкриття класичних методів математичної статистики. Робота з такими уявленнями високих порядків вимагає залучення теорії ймовірності, диференціальної геометрії, і функціонального аналізу. Значна частина сучасної математики в цій області стає дуже популярною.

Один з підходів, який в останні місяці став особливо популярним і яким зацікавилось багато дослідників – це підхід, пов'язаний зі змагальними мережами. Він полягає в тому, що до генеруючої мережі, яка викривляє простий розподіл в потрібний розподіл, навчається ще одна мережа – дискримінуюча (мережа, що оцінює). Процес навчання полягає в грі цих мереж, тобто залучається теорія ігор.

Процес спільного навчання генеруючої мережі і оцінюючої мережі полягає в змаганні, яке влаштовано таким чином. Метою генеруючої мережі, як і раніше, є синтез, генерація реалістичних зображень. А мета оцінюючої мережі – навчитися відрізнити зображення, які створює генеруюча мережа, від реальних зображень. Правила гри такі, що в процесі змагання оцінююча мережа передає в генеруючу мережу інформацію про те, чому вона оцінила те чи інше зображення як несправжнє. Мережа передає деякий вектор градієнта, який говорить, в який бік генеруюча мережа повинна змінити зображення-підробку так, щоб це зображення виглядало для оцінюючої мережі менш фальшивим. В результаті генеруюча мережа може підлаштувати свої параметри так, щоб на виході зображення-підробки виглядали більш реалістичними.

Якщо процес навчання йде успішно, генеруюча мережа створює зображення, які виглядають все краще і краще, які все складніше і складніше відрізнити від реальних зображень, і вони сходяться до розподілу, який потрібно, до розподілу реальних зображень.

Навчання змагальних мереж зараз найпопулярніша тема в машинному навчанні, принаймні, якщо судити по ажіотажу навколо цієї теми, який спостерігається на провідних конференціях по машинному навчанні. Нейронні мережі можуть генерувати маленькі зображення, які виглядають дуже реалістичними. На перший погляд вони виглядають як справжні зображення. Але, якщо перейти до зображень більшого розміру, то їх нереалістичність стає помітною. Помітно, що текстури мережа може вивчити добре, вони виглядають як справжні, а ось форми вдаються куди гірше.

Якщо навчити мережу на зображеннях кішок, то ті зображення, які вона буде синтезувати, матимуть кішок з реалістичною шерстю, і вони будуть стояти на реалістичній траві. Але якщо придивитися, то відразу стане помітно, що у кішки буде не чотири ноги, а, наприклад, дві з половиною. Щоб вирішити цю проблему потрібен синтез методів комп'ютерного тривимірного зору і поточних методів генерації зображень.

Якщо поки що повнорозмірні зображення нейронним мережам не вдаються, то яка на даний момент від цієї задачі користь?

По-перше, нейронні мережі вже вміють перетворювати існуюче зображення нетривіальними способами. Наприклад, вони можуть збільшувати роздільну здатність зображення або стилізувати зображення під різні стилі живопису, видаляти шуми з зображень. Виходить, що будь-який прогрес в області синтезу і генерації зображень призводить до прогресу в області нейронних мереж, які обробляють зображення.

По-друге, для того щоб генерувати зображення, нейронній мережі потрібно навчитися розуміти, як влаштований той чи інший клас об'єктів. В результаті поточні системи генерації вже виявляються дуже корисними для систем розпізнавання.

Якщо взяти змагальні мережі, навчити їх для синтезу зображень осіб, в якийсь момент часу зупинити їх навчання і розглянути тільки мережу-дискримінатор (мережа, яка оцінює, відрізняє підробки від справжніх зображень), то виявиться, що ознаки, які дана оцінююча мережа витягує з зображень обличь, дуже корисні для розпізнавання осіб. Особливо цінним є те, що така мережа може разом з генеруючою мережею навчатися на наборі зображень без розмітки, тобто, без учителя. Це означає, що можна взяти дуже великий набір фотографій осіб без будь-якої розмітки, навчити на них оцінюючу і генеруючу мережу і використовувати оцінюючу мережу для поліпшення якості розпізнавання.

Синтез зображень за допомогою нейронних мереж на даний момент унікальна область. У ній є завдання і для математиків, і для людей, яким цікаві алгоритми та програмування. Ще є відчуття швидкого прогресу, швидкого поліпшення якості результатів.

Розпочати аналіз задачі генерації нових об'єктів варто з питання розкриття "повного потенціалу" генеративних моделей. На рис.1.1. та на рис.1.2. показано зразки, створені з поточних генеративних моделей, що використовуються в літературі з машинного навчання. Які образи будуть

більш належним чином кваліфіковані як "новизна"? Під діючою парадигмою зображення на лівій стороні (цифри і собака) вважаються цінними, оскільки вони відтворюються з деталями відомих об'єктів, в той час як інші будуть розглядатися як «помилкові». Іншими словами, коли прогнозування, здається, єдиною можливою метою, новизна розглядається як помилка, яку дослідники намагаються усунути.

Це мислення спонукає до зменшення або недостатнього використання генеративного потенціалу генеративних моделей. Якщо змінити мету від прогнозування до генерування новизни, оскільки генерація новизни є важливим у своєму власному праві, то це стає цікавим питанням для дослідження, щоб зрозуміти повний генеративний потенціал цих моделей.



Рисунок 1.1. Згенеровані зображення тварин

Обидва рисунки зображені вище генеруються з моделі, що навчається на зображеннях тварин. Враховуючи, що модель була підготовлена для створення зображень тварин (наприклад, собак), чи можна сказати, що ліве зображення є «новизною»? Чи можна зробити висновки, що зображення є правильним шумом або новою собакою чи новим видом?

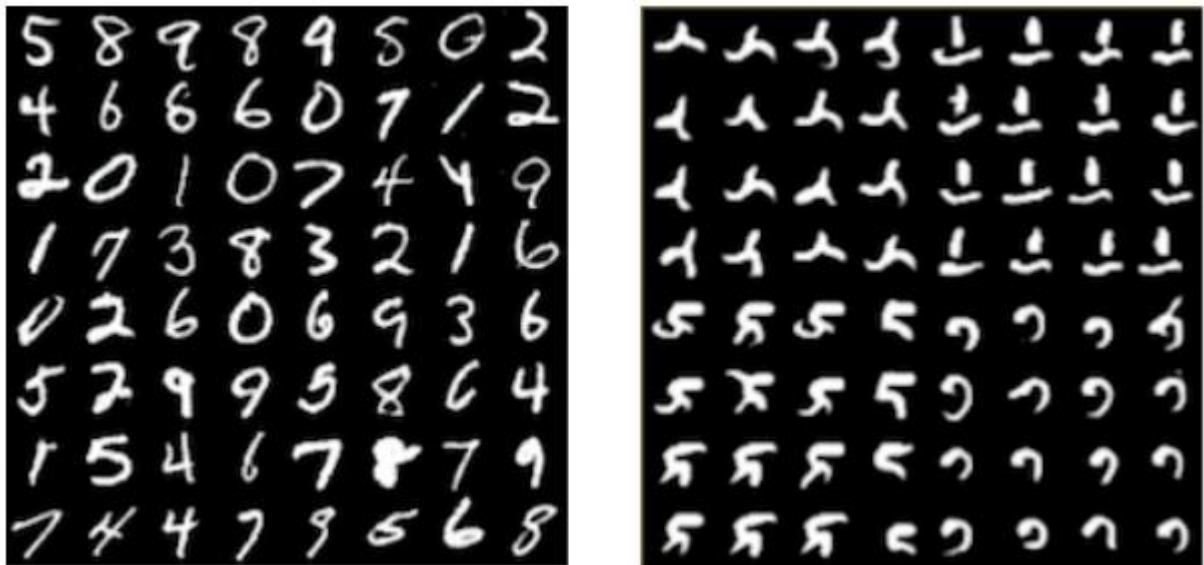


Рисунок 1.2. Згенеровані зображення цифр

Обидва набори зображень генеруються з моделі, що навчається на наборі цифр. Враховуючи, що модель була підготовлена для створення цифр, чи можна вважати зображення зліва «новизною»? Чи є зображення праворуч шумом або новими цифрами чи новими типами символів?

Важливим питанням, яке можна задати, є наступне: "Чи дійсно потрібен новий фундамент для вивчення генерування новизни, крім статистичної теорії навчання? Адже це є основою, яка виявилася надзвичайно продуктивною. Концептуально, однак, легко помітити протиріччя між цією теорією і завданням створення справжньої новизни. У вузьких рамках статистичної теорії навчання, загальною метою генеративного моделювання є навчання моделі, щоб вона змогла відтворювати набір відомих об'єктів з високою ймовірністю, використовуючи такі цільові функції, як максимальна правдоподібність. За такими критеріями, справжня новизна (яку ніколи раніше не бачили) за визначенням мало ймовірна. Що більш дивно, незважаючи на обмежувальні теоретичні рамки теорії статистичного навчання, сучасні реалізації генеративних моделей генерують новизну, як видно на рис.1.1. і рис.1.2. і більш загально в літературі [1] та [2]. В даний час помітна помилка: алгоритми розроблені з метою дотримання та реалізації максимальної правдоподібності та подібних рамок не виконують його.

Ці елементи дають достатньо підстав щоб серйозно сприймати це питання і досліджувати як нові теоретичні основи, так і повну генеративну здатність генеративних моделей. Однак, розробка повноцінної теорії генерування новизни в рамках машинного навчання не є легким завданням.

Хоча важко відповісти на питання цінності генерованих об'єктів в рамках машинного навчання і генеративних моделей, інші дослідники шукали відповіді в межах іншої літератури і парадигми. Наприклад, поле обчислювальної творчості (CC) – це міждисциплінарна сфера, метою якої є вивчення засобів генерування нових об'єктів за допомогою обчислювання значень. Доки існувала деяка робота з використанням машинного навчання, основна частина роботи в CC використовувала еволюційні моделі для генерації новизни. На відміну від генеративних мереж, в CC акцент робиться набагато менше на розумінні основних елементів генерування новизни. CC спільнота, здається, більш зацікавлена у кінцевому результаті генеративного процесу; тобто художня або естетична цінність згенерованих об'єктів або як обчислювальні засоби можуть посилити творчість людини. Іншим принциповим обмеженням робіт в CC, що стосуються цілей цієї тези, є функція значення (називається функція пристосованості в еволюційних моделях), що нав'язується творчому агенту дизайнером системи. Оскільки ці функції значення заздалегідь зафіксовані програмістом системи, генеровані об'єкти відображають його вибір, а не бачення машини. Хоча це не повинно бути проблемою, якщо функція значення відома заздалегідь (це рідко буває у творчому процесі [3]), ці системи мало цікавляться вивченням того, як штучний агент може розробити власне бачення під час свого дослідження.

Генерування новизни є легким завданням. Генератор псевдовипадкових чисел генерує новинку весь час. Аналогічно, простий генератор зображень, який визначає інтенсивність пікселів повністю випадково буде генерувати нові зображення. В принципі, такий простий генератор може згенерувати всі картини Ван Гога і Пікассо, а також нові стилі картин, які будуть винайдені в майбутньому. Але ймовірність цього настільки крихітна, що на практиці це

ніколи не відбудеться. Переважна більшість "нових" образів, які будуть сформовані таким чином, взагалі не матимуть структури. У машинному навчанні такі образи звично називають "шумом".

У чому полягає різниця між створенням картин Ван Гога і створенням шуму з точки зору генерування новизни? Спочатку, шум – це термін, що позначає незрозумілу дисперсію, посилаючись неявно на нездатність деяких спостерігачів знайти розпізнавані структури або закономірності в даних (наприклад, зображення буде розглядатися як шум у тому сенсі, що він не буде розпізнаний людиною). Інтуїтивно, поділ цих двох випадків потребує механізму відбору, поняттям вартості, щоб бути здатним розрізнати корисну новизну і некорисну новизну. Хоча функції цінності (наприклад, корисність) над набором відомих об'єктів є загальним питанням в економіці і науковій літературі, що є цінність для об'єктів що ще не сформовані і невідомі – це питання без відповіді [4].

Крім дилеми, пов'язаної з новизною та шумом, ще одним фундаментальним питанням для генерування новизни – це зв'язок між новизною і знанням про те, що вже існує. Більшість досліджень з обчислювальної творчості (наприклад, на основі еволюційних алгоритмів) зазвичай не мають моделі знань. Все кращі представлення з'являються у кожному циклі еволюційного обчислення (відповідно до заданої функції пристосованості), але не отримано жодної моделі домену. Що ще важливіше, це не загальна модель домену, що керує генеруванням. Ці моделі генерування і тестування піддавалися критиці в теорії дизайну на тій підставі, що генерування та тестування не обов'язково є окремими кроками, і на них впливає знання агента.

У цій роботі вважається, що знання дійсно є фундаментальним аспектом процесу формування новизни. Це робить моделі машинного навчання більш цікавими для дослідження. Алгоритми машинного навчання створюють керовані даними уявлення про світ. Це, по суті, моделі знань. Складність полягає в тому, що представлення не є нейтральними і незалежними

суб'єктами, що існують самостійно. Це дуже добре відомо в машинному навчанні, що, в залежності від використовуваного алгоритму навчання та зразкові, на якому вони навчаються, можна отримати дуже різні представлення.

Насправді, навчання представлення є важким і є одним з основних підрозділів машинного навчання. Менш відомим є ефект, що різні представлення можуть мати на те, що моделі можуть генерувати як новинку. Дійсно, якщо ми переглянемо на прикладі генерації зображень, що ми можемо генерувати, значно залежить від того чи ми працюємо над сирими інтенсивностями пікселів або на патчах зображень (наприклад,  $3 \times 3$  піксельні зображення). Доки знаходження хорошого представлення для прогнозування було центральною темою в машинному навчанні, питання що було б хорошим представленням для генерування новизни ще не досліджено.

#### 1.4. Оцінка генеративних моделей

Для оцінки генеративних моделей нижче приведено п'ять методів. Потім буде проведено їх аналіз та обрано два методи, які будуть використані в експериментальній частині роботи.

##### 1.4.1. Непрямі контрольовані показники

Коли генеративні моделі використовуються як частина мережі з контрольованою метою, оцінка ґрунтується на оцінці повної мережі. Приклади включають непідконтрольне попереднє навчання [5], [2]. Оригінальна мета, що активізувала дослідження нейронних мереж), напівконтрольоване навчання [6], [7], [8], [1], або супер роздільна здатність [9], [10], [11]. Мета дизайну стає зрозумілою, але налаштування обмежується для покращення конкретної мережі, і немає гарантії, що ці цілі можуть бути передані на інші завдання. В даному випадку, мета контрольованої мережі може фактично придушити новизну. В певному сенсі, GAN також підпадають під цю категорію: проектна мета генератора – обдурити якісний дискримінатор, тому

генератору пропонується не генерувати нові об'єкти, які можна легко відрізнити від відомих об'єктів. У даних експериментах виявлено, що GAN можна все-таки налаштувати на генерування новизну поза розподілом, ймовірно, через недоліки як генератора, так і дискримінатора.

Оцінка щільності Парзена регулярно використовується для оцінки вірогідності моделі [12]. Оцінювач щільності ядра підходить для генерування балів, і модель оцінюється за ймовірністю проведення тестового випробування, встановленого під щільністю ядра. Метрики можна легко обдурити [13], проте її прийнято в цій роботі для вимірювання як якості в розподілі, так і поза розподілом генераторів.

#### 1.4.2. Об'єктивність

У роботі [1] запропоновано нову метрику на основі ентропії для вимірювання "об'єктивності" (або початкової оцінки, Inception score) згенерованого набору об'єктів. Як і GAN, метрики використовують навчені дискримінатори, але на відміну від GAN, вони не навчаються для розділення реальних об'єктів та створених об'єктів, а для класифікації реальних об'єктів на існуючі категорії. Мета генератора – створити об'єкти, які точно належать до малої кількості класів. Щоб штрафувати генератори, що фіксуються на окремих об'єктах або категоріях, вони також вимагають, щоб набір об'єктів мав високу ентропію. Показники лише опосередковано пов'язані з класичною вірогідністю імовірності: у певному сенсі вимірюється, наскільки вірогідними є об'єкти, що проходять через дискримінатора. Формально об'єктивність визначається формулою (1.1).

$$\frac{1}{n} \sum_{i=1}^n \sum_{l=1}^K p_{i,l} \log \frac{p_{i,l}}{p_l}, \quad (1.1)$$

де  $K$  – кількість класів,

$$p_{i,l} = p(l|x_i), \quad (1.2)$$

де  $p_{i,l}$  – апостеріорна імовірність категорії даної згенерованому об'єкту з дискримінатором навченим на наборі відомих міток, та маргінальність (1.3).

$$p_l = \frac{1}{n} \sum_{i=1}^n p_{i,l}. \quad (1.3)$$

У роботі [1] запропоновано об'єктивність як один із способів для стабілізації GAN, але, що цікаво, подібний захід застосовувався і в контексті еволюційної генерації новизни [14].

#### 1.4.3. Оцінювання новизни поза розподілом за допомогою позакласової оцінки

Оскільки класична система перевірки під наглядом моделює минуле (навчання) та майбутнє (тест) шляхом випадкового розподілу існуючого набору даних, то можна імітувати наявні знання та новизну шляхом розподілу існуючих наборів даних, що містять цілі класи. Метою генератора новизни є використання навчальних класів для побудови такої моделі, що може генерувати об'єкти з майбутніх класів, невідомих на етапі навчання. В експерименті була спроба випустити окремі класи MNIST, але шум мітки пропустив заняття, які зробили оцінку складним. Щоб цього уникнути, вирішено кинути виклик генератору, навченому на MNIST, для генерації літер. Перевірено декілька різних дискримінаторів, використовуючи різні настройки, лише на цифрах (MNIST) на літерах (шрифти Google, загалом 1268 шрифтів, що дало 32968 прикладів), або на суміші цифр і букв, і використовували ці дискримінатори для оцінки генераторів новизни.

#### 1.4.4. Позакласова оцінка

Природно, дискримінатори літер бачать букви усюди. Оскільки букви – це все що вони знають, вони класифікують все в один з класів букв. Об'єктивність букви в генераторі цифр в розподілі може іноді бути високим. Наприклад, багато цифр «6» були віднесені до категорії букви «б». Щоб цього

уникнути підготовлено дискримінатор на об'єднанні цифр і букв, що дозволяє вибирати цифри, коли відчувалося, що згенерований об'єкт більше нагадує цифру.

Розроблено два показники за допомогою цього дискримінатора: позакласовий показник вимірює значення частоти впевнено класифікованих літер у створеному наборі та позакласовий максимізатор – середнє значення (над набором) ймовірності найбільш вірогідної літери. Ніякі з цих метрик не карають фіксовані генератори, виводячи однакові кілька літер весь час, тому було поєднано обидва показники з ентропією літери.

Формально, припустимо  $p_{i,1}, \dots, p_{i,K_{in}}$  внутрішньо класові нащадки,  $p_{i,K_{in}+1}, \dots, p_{i,K_{in}+K_{out}}$  позакласові нащадки, де  $K_{in} = 10$  це кількість внутрішніх класів (цифр) та  $K_{out} = 26$  це кількість позакласових класів (літер).

Нехай (1.4) та (1.5) будуть найбільш можливими позакласовими категоріями.

$$l_i^* = \arg \max_l p_{i,l}, \quad (1.4)$$

$$l_{out_i}^* = \arg \max_{K_{in} < l \leq K_{in} + K_{out}} p_{i,l}. \quad (1.5)$$

Нехай (1.6) – нормалізована емпірична частота позакласової категорії  $l$ .

$$\tilde{p}_l = \frac{\sum_{i=1}^n \mathbb{1}\{l = l_{out_i}^*\}}{\sum_{i=1}^n \mathbb{1}\{l_{out_i}^* > K_{in}\}}. \quad (1.6)$$

Різноманітність згенерованих прикладів (1.7) вимірюється нормалізованою ентропією емпіричних частот.

$$\text{різноманітність} = -\frac{1}{\log K_{out}} \sum_{l=K_{in}}^{K_{in}+K_{out}} \tilde{p}_l \log \tilde{p}_l, \quad (1.7)$$

$$\text{позакласова кіськість} = (1 - \lambda) \times \frac{1}{n} \sum_{i=1}^n \|\{l_i^* > K_{in} \wedge p_{i,l_i^*} > \theta\} + \lambda \times \text{різноманітність}, \quad (1.8)$$

$$\text{позакласовий максимум} = (1 - \lambda) \times \frac{1}{n} \sum_{i=1}^n p_{i,l_{out_i}^*} + \lambda \times \text{різноманітність}. \quad (1.9)$$

У даному експерименті  $\theta = 0.95$ ,  $\lambda = 0.5$ .

#### 1.4.5. Людське судження або візуальний тест Тьюрінга

Кінцевим тестом для генеративних моделей є те, чи подобаються людині згенеровані об'єкти. Візуальний огляд часто використовується як принцип оцінки в роботах [15], [16], [17]. По-перше, це, безумовно, робить розвиток (наприклад, вибір моделі та налаштування гіперпараметрів) повільним. По-друге, результати багато в чому залежать від того, які питання задаються і на чому відповіді ґрунтуються. Для тестування генеративних моделей звичайним запитанням типу GAN є те, чи об'єкти породжені природою (або людиною) або створені машиною (візуальний тест Тьюрінга). Навіть ті, хто іде далі в розробці мереж для створення новизни [18] просять людського судження для розмежування результатів між людиною і машиною.

#### 1.5. Аналіз підходів до рішення задачі генерації

На основі огляду генеративних мереж та їх аналізу, зроблено висновок про те, що для задачі генерації зображень найкраще підходять генеративні змагальні мережі. На рис.1.3 показано генеративний модуль інтелектуальної робототехнічної системи, показано генеративні моделі та їх місце відносно інших моделей. Також на рис.1.3 показано відмінність між генеративними моделями та дискримінативними моделями.

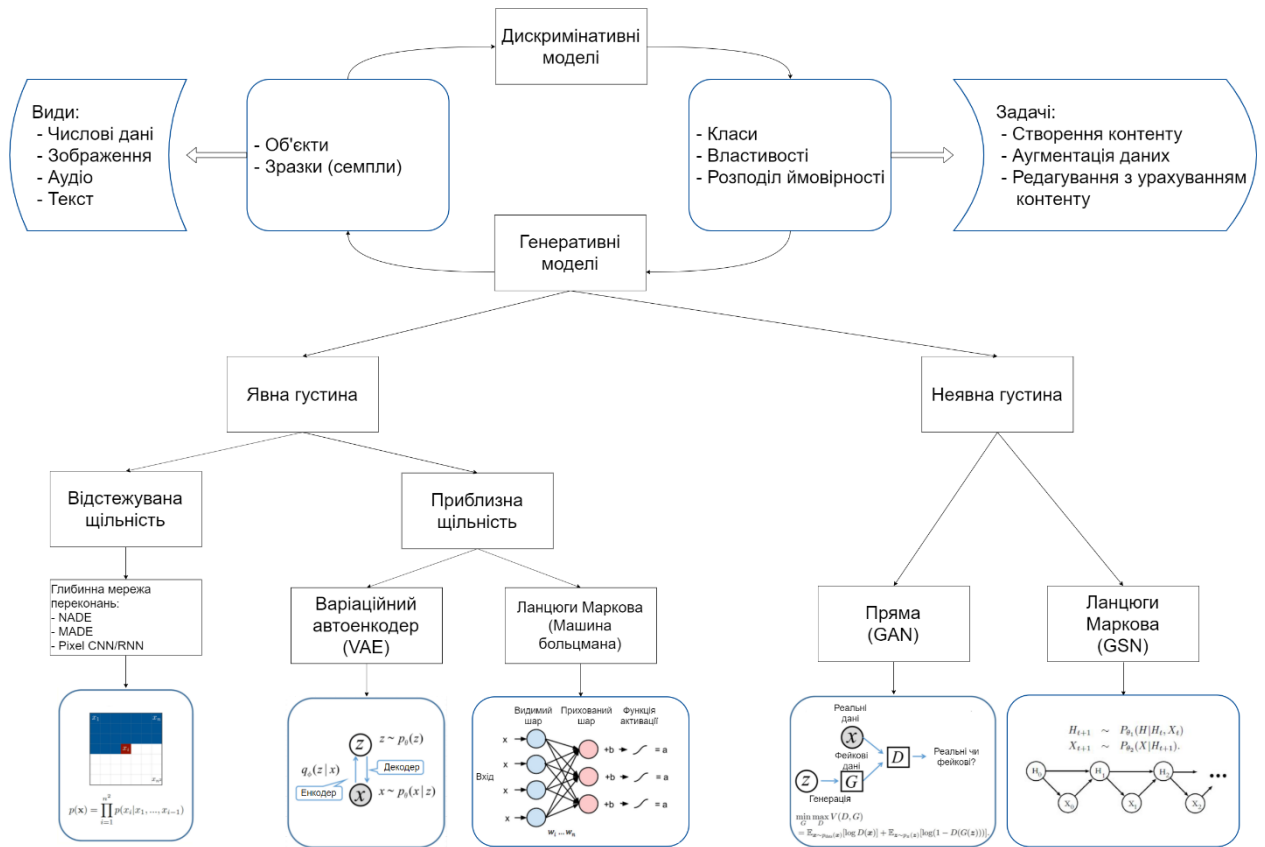


Рисунок 1.3. Генеративний модуль інтелектуальної робототехнічної системи

Також у цьому розділі підсумовано три основні підходи, що використовуються для генерування зображень генеративними змагальними мережами, тобто прямі методи, ітераційні методи та ієрархічні методи відповідно. Загальна структура цих трьох методів показані на рис. 1.4.

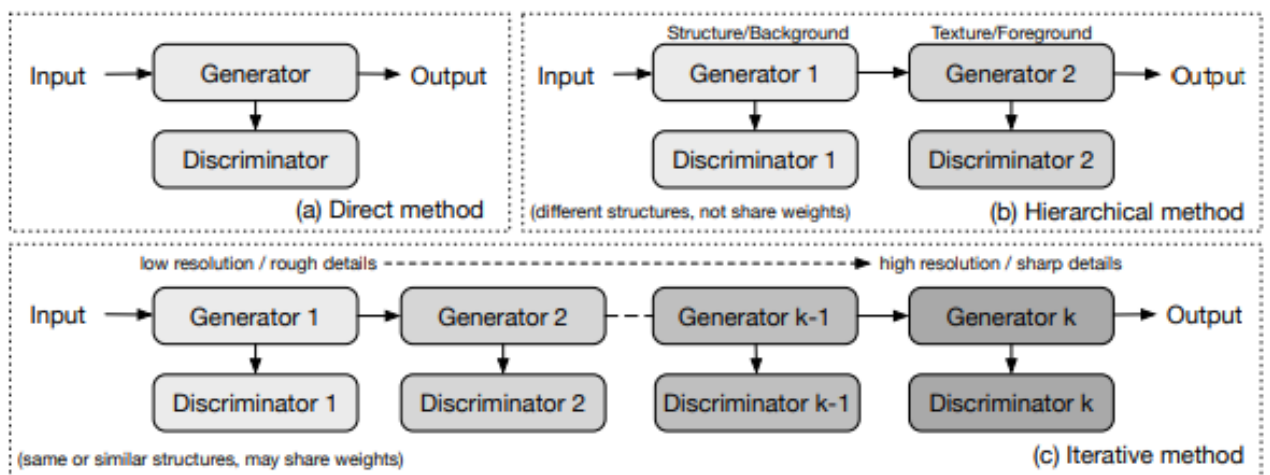


Рисунок 1.4. Підходи до синтезу зображень

### 1.5.1. Прямі методи

Всі методи цієї категорії слідують філософію використання одного генератора і одного дискримінатора у своїх моделях, а також структури генератора і дискримінатора прямі і без гілок. Багато з перших GAN моделей підпадають під цю категорію, такі як GAN, DCGAN, ImprovedGAN, InfoGAN, f-GAN і GANINT-CLS. Серед них DCGAN є однією з найбільш класичних, чия структура використовується багатьма моделями створеними пізніше. Загальні будівельні блоки, що використовуються у DCGAN показані на малюнку 6, де генератор використовує транспоновану згортку, нормалізацію пакетів і активацію ReLU, в той час як дискримінатор використовує згортку, нормалізацію пакетів і активацію LeakyReLU.

Такий метод є відносно більш прямим для розробки і реалізації в порівнянні з ієрархічними та ітеративними методами, і в ньому зазвичай досягаються хороші результати.

### 1.5.2. Ієрархічні методи

На відміну від прямого методу, алгоритми ієрархічного методу використовують два генератори і два дискримінатори у своїх моделях, де різні генератори мають різні цілі. Ідея цих методів полягає у розділенні зображення на дві частини, такі як "стиль і структура" і "передній план і фон". Відношення між двома генераторами може бути паралельним або послідовним.

SS-GAN пропонує використовувати дві GAN, StructureGAN для генерації поверхневої нормальної карти з випадкового шуму  $z^{\wedge}$ , та інший StyleGAN, який приймає як згенеровану поверхню нормальної карти, так і шум  $z^{\sim}$  на вхід і виводить зображення. Структура GAN використовує ті ж самі будівельні блоки, що і DCGAN, в той час як Style-GAN трохи відрізняється. Для генератора стилів згенерована поверхнева нормальна карта і вектор шуму проходять через кілька згорткових і транспонованих згорткових шарів відповідно, а потім результати об'єднуються в єдиний тензор, який пройде

через інші шари у генераторі стилів. Як для дискримінатора стилю, кожна поверхня нормальної карти і відповідне зображення об'єднуються на каналі розмірності для формування єдиного входу до дискримінатору. Крім того, SS-GAN припускає, що хороше синтетичне зображення повинне також бути використаним для реконструкції гарної поверхні нормальної карти. Згідно з цим припущенням, SS-GAN розробляє повністю зв'язану мережу, яка перетворює зображення на нормальну поверхню карти, і використовує піксельну втрату, яка примушує реконструйовану нормальну поверхню наблизитися до істинної. Головне обмеження SS-GAN полягає в тому, що потрібно використовувати Kinect для отримання правди у поверхневих нормальних картах.

Як особливий приклад, LR-GAN вибирає генерувати на передньому плані та на фоні, використовуючи різні генератори, але тільки один дискримінатор використовується для судження зображення, поки процес генерації рекурентного зображення пов'язаний з ітераційним методом. Проте експерименти з LR-GAN демонструють, що можна відокремити генерацію змісту переднього плану та фону і отримати більш чіткі зображення.

### 1.5.3. Ітераційні методи

Цей метод відрізняє себе від ієрархічних методів двома способами. По-перше, замість використання двох різних генераторів, що виконують різні ролі, моделі цієї категорії використовують кілька генераторів, які мають подібні або навіть однакові структури, і вони генерують зображення від грубого до нормального, де кожен генератор уточнює деталі результатів попереднього генератора. По-друге, при використанні однакових структур у генераторах, ітераційні методи можуть використовувати розподіл ваги серед генераторів, тоді як ієрархічні методи зазвичай не можуть.

LAPGAN є першим GAN, який використовує ітеративний метод генерування зображень від грубого до нормального з використанням піраміди Лапласа. Кілька генераторів в LAPGAN виконують одне і те ж завдання:

приймає зображення з попереднього генератора і вектор шуму в якості вхідних даних, а потім виводить деталі (залишкове зображення), які можуть зробити зображення більш чітким при додаванні до вхідного зображення. Єдина відмінність в структурі цих генераторів є розмірність вводу або виводу, а виключення полягає в тому, що генератор на найнижчому рівні приймає тільки вектор шуму на вхід і подає на вихід зображення. LAPGAN перевершує оригінальний GAN і показує, що ітераційний метод може генерувати більш чіткі зображення ніж прямий метод.

StackGAN, як ітераційний метод, має лише два шари генераторів. Перший генератор приймає вхід  $(z, c)$  і потім виводить нечітке зображення, яке може показувати грубу форму і розмиті деталі об'єктів, при цьому другий генератор приймає  $(z, c)$  і зображення, згенерований попереднім генератором, а потім виводить велике зображення з більшою реалістичністю деталей.

Іншим прикладом ітераційних методів є SGAN, який об'єднує генератори, що приймають особливості нижнього рівня на вхід, і виводить особливості більш високого рівня, а нижній – генератор приймає вектор шуму на вхід, а верхній генератор виводить зображення. Необхідність використання окремих генераторів для різних рівнів особливостей полягають у тому, що SGAN асоціює енкодер, дискримінатор і Q-мережу (яка використовується, щоб прогнозувати постеріорну ймовірність  $P(z_i | h_i)$  для максимізації ентропії, де  $h_i$  є вихідною особливістю  $i$ -го шару) для кожного генератора, щоб обмежувати і поліпшувати якість цих функцій.

Прикладом використання розподілу ваги є GRAN модель, яка є продовженням моделі DRAW яка заснована на варіаційному автоенкодері. Як в DRAW, GRAN генерує зображення рекурентним чином, яке подає вихідні дані попереднього кроку в модель і вихід поточного кроку буде переданий назад як вхід на наступний крок. Всі кроки використовують один і той же генератор, тому ваги розподіляються серед них, так само, як і в класичних рекурентних нейронних мережах (RNN).

#### 1.5.4. Інші методи

PPGN створює вражаючі зображення в декількох завданнях, таких як синтез класових зображень, синтез тексту в зображення і зображення малюнків. Відмінний від інших методів, згаданих раніше, PPGN використовує активацію максимізації для створення зображень, і вона базується на відборі зразків з попереднім навчанням з автоматичним енкодером, що прибирає шум (DAE). Створювати зображення, обумовлене певною маркувальна класовою міткою  $y$ , замість того, щоб використовувати спосіб прямого поширення (напр. рекурентні методи можна розглядати як методи прямого поширення, якщо вони розгорнуті у часі), PPGN запускає процес оптимізації, який знаходить вхід  $z$  до генератора, який змушує вихідне зображення сильно активувати певний нейрон в іншому, попередньо підібраному класифікаторі (в даному випадку нейрон у вихідному шарі, що відповідає його мітці класу  $y$ ).

Для того, щоб генерувати кращі зображення з більш високою роздільною здатністю, ProgressiveGAN пропонує почати з навчання генератора і дискримінатор  $4 \times 4$  пікселя, після чого він поступово додає додаткові шари, що подвоює вихідну роздільну здатність до  $1024 \times 1024$ . Такий підхід дозволяє вивчати модель спочатку грубій структурі, а зосередитися на уточненні деталей пізніше, замість того, щоб мати справу з усіма деталями в різному масштабі одночасно.

### 1.6. Аналіз існуючих рішень

У даному розділі будуть наведені приклади існуючих рішень. На основі цих рішень і прикладів результатів цих рішень можна буде зробити висновок про сучасний стан розвитку генеративних мереж у створенні нового контенту.

#### 1.6.1. Машинне навчання

Є три важливі роботи, які розглядають генерацію новизни у контексті машинного навчання. У роботі [14] пропонується інноваційний підхід, результати якого зображені на рис.1.5.



Рисунок 1.5. Синтетичні зображення

Вони генерують зображення за допомогою нейронної мережі, яка містить синтетичні особливості. На генератор подається оцінка на основі ентропії (аналогічно об'єктивності), що надходить від класифікатора Imagenet, і зворотній зв'язок використовується в циклі еволюційної оптимізації для керування генерацією. Важливий внесок роботи – продемонструвати важливість оцінки об'єктивності. Вони показують, що цікаві об'єкти не створюються, коли машину просять генерувати з єдиного заданого класу. Шляхи генерування часто проходять через об'єкти різних класів, які, здавалося б, не пов'язані з кінцевим об'єктом. Основна концептуальна відмінність з підходом в цій роботі полягає в тому, що у роботі [14] не ґрунтують свою генеративну модель на засвоєних знаннях: процес їх генерації є не вивчена модель, а скоріше стохастичний комбінаторний двигун. З одного боку, це робить генерацію досить повільною, а з іншого, отримані об'єкти відображають скоріше стиль заданих синтетичних особливостей, ніж риси, витягнуті з існуючих об'єктів.

Основна мета у роботі [18] та [19] – навчання та генерація «з першого разу». Тобто навчити класифікувати предмети, задані невеликою кількістю (часто одним) із прикладів, що надходять у заданій категорії, і вчити генерувати нові об'єкти за єдиним прикладом рис.1.6.



Рисунок 1.6. Необмежена набором даних генерація символів

Такий підхід безумовно, є проміжним кроком до позакласової генерації. Вкрай низька кількість прикладів концептуально обмежує навчання, генерацію, що ґрунтується на вірогідності. У роботі [18] обходять цю проблему, навчаючи сильні баєсівські моделі (програми) зверху вниз, які фіксують структурні властивості відомих об'єктів, які є узагальненими по класах. Вони також розглядають генерацію без обмежень «з першого разу» як розширення свого підходу та показують, що модель може генерувати нові символи з нуля.

У роботі [4] вони показують, що символи нових типів можна генерувати ретельно налаштованими автоенкодерами, вивченими повністю знизу вгору, не нав'язуючи баєсівську архітектуру зверху вниз рис.1.7.



Рисунок 1.7. Генерування нових об'єктів

У роботі також роблять перший крок у визначенні концептуальних рамок генерації новизни, аргументуючи мету генерування об'єктів з нових типів, невідомих на час навчання. Вони проектують методику пошуку цих нових типів напів-автоматично (поєднання кластеризації та маркування людиною). Вони стверджують важливість визначення цінності цих нових типів (і генерації поза розподілом загалом), але вони не роблять спроб розробити оціночні метрики.

### 1.6.2. Глибоке навчання і генеративні моделі

Важливе сімейство алгоритмів навчання представлення називається глибоким навчанням. Глибоке навчання у роботі [19] це парадигма машинного навчання, яка мала великий вплив у кількох областях, таких як комп'ютерне бачення [20], розпізнавання мови [21] і обробка природної мови [22]. Однією

з основних характеристик глибокого навчання є те, що воно може автоматично вивчати функції з необроблених даних (наприклад, з пікселів або звукових сигналів), уникаючи необхідність в інженерних розробках особливостей, при цьому платою є налаштування його архітектури і численні параметри.

Термін глибоке навчання в основному відноситься до ідеї використання декількох рівнів представлення зі зростаючою абстракцією. Ця ідея реалізується через нейронні мережі, які є сімейством функцій, що відображають вхідний простір на вихідний простір через набір нелінійних перетворень, який називають "шарами", і який може бути розглянутий як проміжна репрезентація даних.

Існує багато видів шарів, таких як, наприклад, повністю пов'язані та згорткові шари. Шари відрізняються тим, як вони перетворюють свій вхід на вихід. Це різні типи шарів складаються разом, у спрямованому ациклічному графіку (DAG), який називається обчислювальним графіком.

Кожен вид шару має параметри, що потенційно можуть навчатися, які називаються вагами, які коригуються на етапі навчання для оптимізації функції втрати. У шарі є також гіперпараметри, які є параметрами, що повинні бути зафіксовані до етапу навчання (наприклад, кількість юнітів). Спосіб вибору цих шарів (або практикуючим, або автоматичним способом), щоб скласти їх разом і вибір їх гіперпараметрів визначають архітектуру нейронної мережі.

Одна з особливостей нейронних мереж полягає в тому, що вони можуть бути навчені від початку до кінця (або наскрізне). Поняття наскрізного навчання означає, що якщо ми маємо конвеєр з множинними кроки, ми маємо можливість налаштувати параметри всіх кроків разом, щоб оптимізувати кінцеву мету, а не оптимізувати кожен крок окремо.

У нейронних мережах шари можна розглядати як проміжні етапи трубопроводу, що ідуть від входу до виходу. Наскрізний підхід до навчання в нейронних мережах означає, що всі параметри всіх шарів, починаючи з входу і до виходу, можуть регулюватися для оптимізації функції втрат, яка зазвичай

працює на вихідному просторі (наприклад, ми хочемо, щоб прогнозований висновок закрит бажаний вихід).

Наскрізне навчання можливе, оскільки шари вибираються як диференційовані функції. Це означає, що можна використовувати стохастичний градієнтний спуск (SGD) або його варіанти для оптимізації функції втрат, обчислюючи градієнти параметрів всіх шарів разом із зворотним поширенням [23]. Зворотне поширення – це алгоритм динамічного програмування, який полягає в рекурсивному застосуванні правила ланцюжка, починаючи з виходів і назад через шари, щоб обчислити градієнт усіх ваг. Потім градієнт використовують для оновлення ваг, щоб оптимізувати функцію втрат.

### 1.6.3. Генеративні змагальні мережі

У даному параграфі представлені деякі останні роботи з використанням генеративних змагальних мереж (GAN) для побудови генеративних моделей зображень. Робота за посиланням[24] була першою статтею про GAN, вони ввели теорію та процедуру навчання, та перерахували переваги та недоліки GAN порівняно з іншими видами генеративних моделей. Наприклад, однією з переваг GAN є те, що вибірка є простою, тоді як в інших генеративних моделях, подібно до машин Больцмана, вибірці потрібні Марківські ланцюги, які вимагають декількох кроків і таким чином можуть бути набагато повільнішими.

Також, в той час як GAN підтримують довільні види архітектури для генератора і дискримінатора (за умови їх диференційованості), їх навчання часто складне і нестійке, і тому вони можуть потерпіти невдачу. Одним із способів, яким вони можуть провалитися, є досягнення сценарія Helvetica, де генератор виводить одне і те ж зображення для різних вхідних даних.

Ідея GAN – навчити одночасно дві моделі: дискримінатор і генератор. Генератор приймає вектор випадкового шуму (зазвичай генерується за допомогою простого розподілу, як рівномірний або гауссовий) на вхід і на

вихід отримуємо згенерований зразок (наприклад, зображення). Дискримінатор приймає за вхідний сигнал або генерований зразок або реальний (вибірка з навчальних даних) і передбачає, чи він згенерований чи він реальний. Дві моделі (генератор і дискримінатор) грають у гру. Генератор намагається генерувати зразки, які обманюють дискримінатора, тобто намагається зробити так, щоб дискримінатор передбачив, що згенеровані дані є "реальними". З іншого боку, дискримінатор намагається розрізнити реальні і згенеровані дані так точно, як це можливо. Навчання здійснюється шляхом чергування оптимізації дискримінатора та оптимізації генератора.

У роботі [25] запроваджено умовні генеративні змагальні мережі, просту модифікацію оригінального GAN для підтримки умовності. В умовних GAN, генератор отримує вектор шуму, а також вектор умов як вхідні дані, наприклад, один гарячий вектор, який кодує клас, в який ми хочемо передати умови та вивести зображення. Дискримінатор отримує зображення, але також вектор умов як вхідний сигнал і повертає скаляр між 0 і 1.

У роботі [15] використовували багатомасштабну структуру природних образів щоб використовувати GAN, щоб навчитися поступово підвищувати вибір зображень з низькою роздільною здатністю та трансформувати їх до більш високої роздільної здатності. Для вибору моделі вони використовували оцінку правдивості Парзена. Для оцінки своєї моделі вони провели експеримент з опитуванням людей, чи зображення є реальним або підробленим, порівнюючи стандартний GAN і їх метод.

Перший документ, у якому автори намагаються підвищити стабільність і продуктивність генеративної змагальної мережі є робота [16]. У роботі представили архітектуру з оптимізацією гіперпараметрів і деякими налаштуваннями для ефективного навчання генератора глибоких згорткових нейронних мереж. Вони показують, що їх генеративна змагальна мережа може генерувати реалістичні зображення розміром  $64 \times 64$  і  $32 \times 32$  на трьох наборах даних, наборі даних обличч, набір даних спалень [26] і [27], зображених на рис.1.8.



Рисунок 1.8. Згенеровані зображення спалень

Вони оцінювали свою модель шляхом візуального огляду та за допомогою навченого представлення дискримінатора для перевірених завдань. Вони також візуалізують те, що ці моделі навчені і показують, що деякі згорткові фільтри навчені малювати конкретні об'єкти.

Вони також показують, що модель не запам'ятовує зображення, а зазначають, що інтерполяція між зображеннями в прихованому просторі (вхід генератора) є гладкою. Цікаво, що вони також показують, що це можливо перетворити зображення, виконуючи аналогічні міркування, використовуючи приховане представлення зображення. Вони спочатку усереднили приховані представлення, які призводять до зображень усміхненої жінки, які можна розглядати як представлення "усміхнена жінка". Тоді, аналогічно, вони обчислюють "нейтральне представлення жінки" і "нейтральне представлення чоловіка".

Нарешті, вони віднімають "нейтральне представлення жінки" від "усміхненої жінки" і додають представлення "нейтрального чоловіка" і генерують зображення з цього представлення, щоб отримати усміхнений образ людини. Для прикладу приведено рис.1.9.

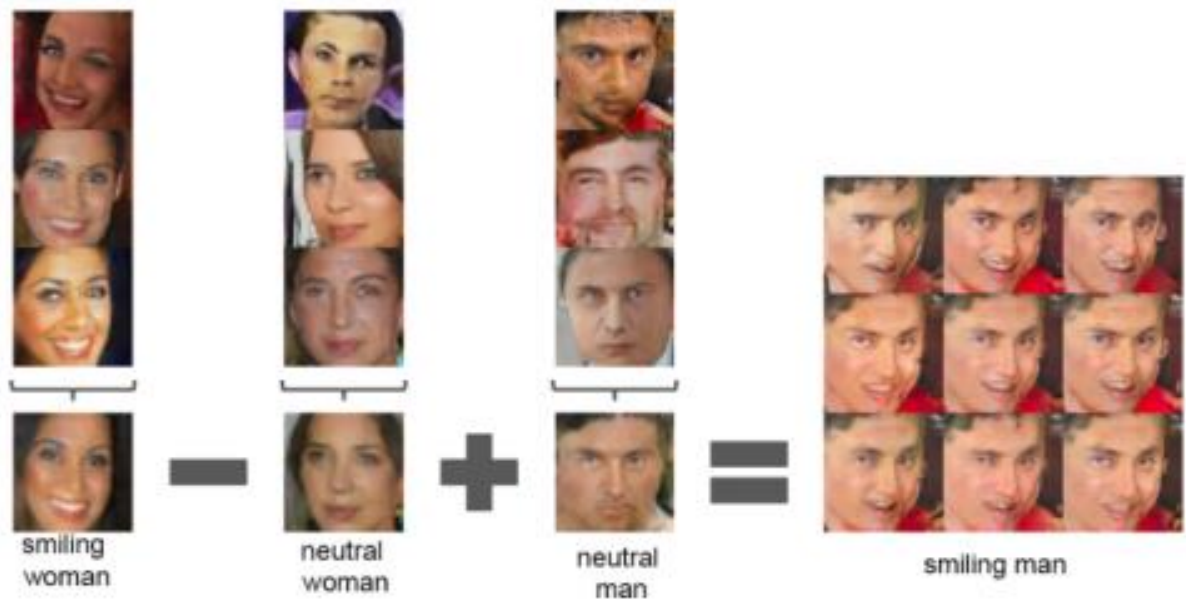


Рисунок 1.9. Згенеровані зображення облич

У роботі [28] запропоновано проксі для людської персептивної метрики для зображення, які використовують GAN. Вони пропонують використовувати відстань на абстрактному функціональному просторі, яке вивчається класифікатором, так що воно містить семантичну інформацію про зображення. Вони пропонують використовувати GAN разом з втратами, заснованими на особливостях, щоб згенеровані зображення мали різноманітні природні образи.

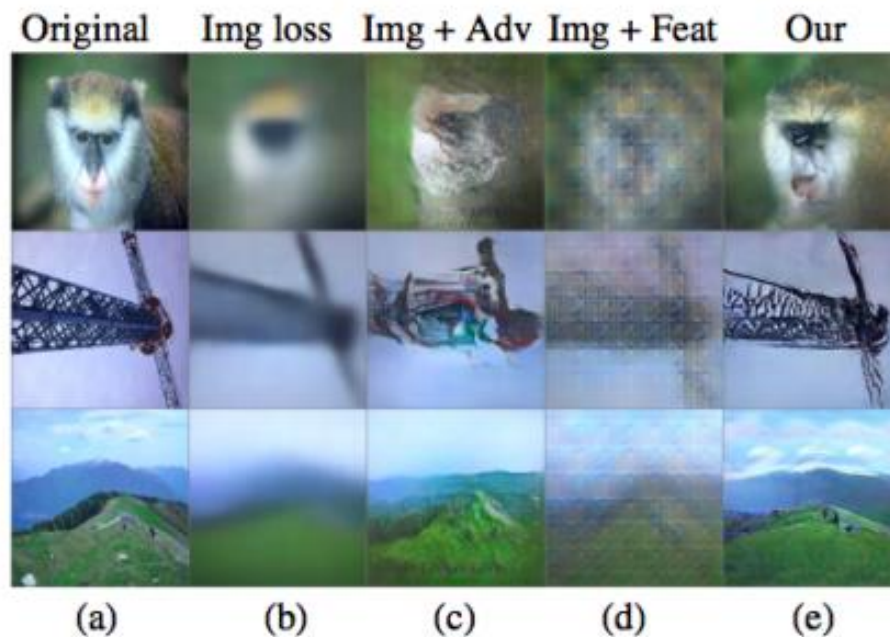


Рисунок 1.10. Ефект втрати

Вони використовують цю установку в автоенкодерах, які традиційно використовують евклідову відстань на піксельному просторі, і замінюють його на втрати на основі GAN у поєднанні з евклідовою відстанню в просторі ознак. Вони отримують зображення, які є більш реалістичними, ніж отримані з евклідовою відстанню на піксельному просторі, які часто бувають розмитими. Для ілюстрації ефекту запропоновані втрати приведено рис.1.10.

У роботі [1] запропоновано набір евристик для підвищення стабільності GAN, і вперше пропонують використовувати GAN в напів-контрольованому навчанні і одержав результати, що були витвором мистецтва. За допомогою підбору особливостей, замість дискримінатора, який виводить скаляр, вони мають дискримінатор, який виводить вектор особливостей, і мета перетворюється на підбір статистики набору особливостей з реальних даних і згенерованих даних, статистика, яка обрана, є середньою функцій над міні-партіями прикладів. Також, що важливо, що вони ввели початкову оцінку, яка стала стандартним інструментом для оцінки моделей для генерування зображень. Вони виявили, що початкова оцінка корелює з людськими оцінками чи є зображення реалістичними. Об'єктивність висока, коли класифікатор впевнений, що образи належать до одного з класів (низька ентропія), а також висока коли класи цих зображень різноманітні (висока ентропія), так що не всі зображення належать деяким домінуючим категоріям об'єктів.

У роботі [29] вивчають роз'єднані фактори зміни даних без вчителя за допомогою GAN. Ідея вивчення роз'єднаних факторів варіації датується, принаймні роботою [30]. Припускається, що мінливість даних надходить від незалежного поєднання численних факторів, таких як освітлення, точка зору, а також особливості обличчя.

У роботі [31] пропонують фреймворк для генерування зображень керованим шляхом, таким чином, користувач може забезпечити розташування і зміст різних частин зображення.

Робота [32] показала ефективність фреймворку GAN для трансляції зображення на зображення. Наприклад, його можна використовувати для перетворення зображення будинку в план будинку, або від ескізу до детального зображення, або від стилю живопису до іншого стилю.

Робота [8] розширює рамки умовних GAN і показує, що надання інформації про мітку генератору може допомогти створити більш реалістичні зображення, особливо в наборах даних з високою мінливістю. Способом розширення умовних GAN є змусити дискримінатор не тільки передбачити, чи є зображення реальним чи фальшивим, але й передбачити його клас.

Робота [33] узагальнила фреймворк трансляції зображення-зображення до ситуацій, коли пари зображень з вихідного домену до цільового домену недоступні. У цьому завданні ми отримуємо зразки з вихідного домену і вибірки з цільового домену, але без сполучень між ними і мета полягає в тому, щоб навчитися перетворювати зображення з вихідного домену в цільовий домен. Для прикладу приведено рис.1.11.

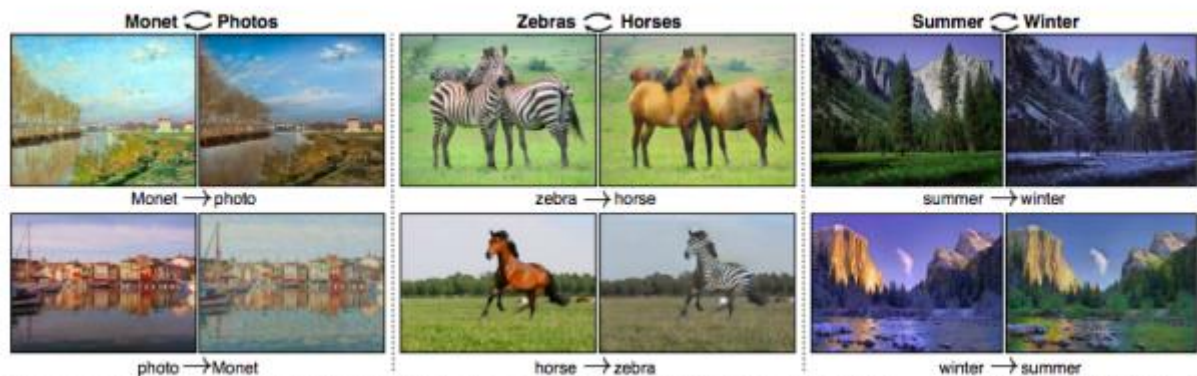


Рисунок 1.11. Зміна домену зображень

#### 1.6.4. Обчислювальна творчість та генетичні алгоритми

Обчислювальна творчість – це міждисциплінарна сфера, метою якої є вивчення засобів генерування новизни обчислювально. Основними питаннями, на які звертається увага є алгоритми генерації, а також метрики оцінки. Незважаючи на те, що СС дуже різноманітна, основні напрямки, які вивчаються, це мистецтво (головним чином зображення та музика [34]), хоча

нещодавно деякі автори заявили, що творчість не повинна обмежуватися лише мистецтвом, а СС має охоплювати і науковий напрямок. Сфера СС відноситься як до генерації, тобто алгоритмам та інструментам, які можна використовувати для створення нових об'єктів, так і до оцінки, показників та метрик, які використовуються для оцінки цінності та новизни кожного генерованого об'єкту. Більшість систем СС розділяють фази генерації та оцінки, що робить їх своєрідними системами генерування і тестування [35].

#### 1.6.5. Теорія дизайну

Існує велика кількість літератури про дизайн і теорії дизайну, мета яких полягає в тому, щоб пояснити, як працює процес проектування нових об'єктів (наприклад, новий тип автомобіля, новий лікарський засіб або новий твір мистецтва). Вивчення процесу проектування взагалі є, незалежно від суб'єкта або домену, важливим напрямком з урахуванням економічного і соціального впливу дизайну в сучасних суспільствах. У цьому абзаці описано особливу теорію дизайну, що називається С-К теорією. Ця теорія має особливість підкреслення важливості знань у створенні нових концепцій, які можуть призвести до нових типів об'єктів. Основні поняття теорії є ілюстрацією таких концепцій, які можна шукати у творчих агентів, які прагнуть генерувати новизну. Більш того, хоча теорія С-К пропонує математичні теорії, такі як логіка висловлювань або теорія множин для моделювання та представлення знань, в принципі, вона не має обмежень у тому, який формалізм використовувати для представлення знань. Це означає, що використання глибоких генеративних моделей як моделі знань також може розглядатися як внесок у дизайн.

#### Висновки до розділу

У даному розділі була описана мета роботи, предметна область та особливості, що присутні в рішенні задач в цій області.

Була детально розглянута задача генерації нових об'єктів. А також складнощі та проблеми що виникають при рішенні цього класу задач. Продемонстровані реальні спроби вирішення задач генерації нових зображень.

Також були розглянуті методи оцінки генеративних моделей та представлений їх опис. В наступних розділах деякі методи оцінки будуть використані для експериментів. Були розглянута група методів, підходів до рішення подібних задач генерування зображень. А також приведено деякі існуючі рішення.

У даному розділі була розглянута література для генеративного моделювання, що може бути хорошою відправною точкою для вивчення систем, які можуть генерувати нові об'єкти. Однак мета, яка описана в більшості робіт недостатньо узгоджена з генеруванням нових об'єктів, оскільки кінцевою метою, прийнятою в роботах є можливість генерувати дані, що виходять з розподілу даних. Тобто, більшість робіт зосереджуються на створенні відомих об'єктів реалістичного вигляду.

## 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ГЕНЕРАТИВНИХ МОДЕЛЕЙ ІНТЕЛЕКТУАЛЬНИХ РОБОТОТЕХНІЧНИХ СИСТЕМ

На основі аналізу сучасного стану проблеми генерації та огляду існуючих рішень та методів виявлено, що найліпшою архітектурою генеративної мережі у задачах створення візуального контенту на даний момент є моделі генеративних змагальних мереж.

У даній роботі буде описана архітектура, складові та конфігурація нейронної мережі для генерування зображень (логотипів).

Розглянемо архітектурні та структурні особливості генеративних змагальних нейронних [1] мереж на яких базується розроблений у даній роботі експеримент.

### 2.1. Генеративна змагальна мережа

Генеральна змагальна мережа (GAN), зображена на рис.2.1. складається з двох частин:

1. Генератор вчиться генерувати правдоподібні дані. Утворені екземпляри стають негативними прикладами навчання для дискримінатора.
2. Дискримінатор вчиться відрізняти фальшиві дані генератора від реальних даних. Дискримінатор штрафує генератор за отримання неправдоподібних результатів.

Коли навчання починається, генератор видає явно підроблені дані, і дискримінатор швидко вчиться говорити, що це підробка. По мірі прогресу навчання генератор наближається до отримання виходу, який може обдурити дискримінатор. Нарешті, якщо навчання генератора проходить добре, дискримінатору стає складно відчувати різницю між реальним та підробленим виходом. Він починає класифікувати підроблені дані як реальні, а його точність знижується.

І генератор, і дискримінатор є нейронними мережами. Вихід генератора підключається безпосередньо до входу дискримінатора. Завдяки зворотному



повинен робити це, контролюючи лише  $\theta^{(D)}$ . Генератор хоче мінімізувати  $J^{(G)}(\theta^{(D)}, \theta^{(G)})$  і повинен робити це, контролюючи лише  $\theta^{(G)}$ . Тому що функція вартості кожного гравця залежить від параметрів іншого гравця, але кожен гравець не може контролювати параметри іншого гравця.

Рішення для проблеми оптимізації можна знайти в локальному мінімумі, що являє собою точку в просторі параметрів, де всі сусідні пункти мають більшу або рівну вартість. Рішення гри представлено рівновагою Неша. У цьому контексті рівновага Неша є кортежем  $(\theta^{(D)}, \theta^{(G)})$  що є локальним мінімумом  $J^{(D)}$  відносно  $\theta^{(D)}$  та локального мінімуму  $J^{(G)}$  відносно  $\theta^{(G)}$ .

### 2.1.1. Дискримінатор генеративної змагальної мережі

Дискримінатор у генеративній змагальній мережі представляє собою просто класифікатор. Він намагається відрізнити реальні дані від даних, створених генератором. Він може використовувати будь-яку мережеву архітектуру, відповідно до типу даних, які класифікує.

#### 2.1.1.1. Навчальні дані дискримінатора

Дані для навчання дискримінатора генеративної змагальної мережі надходять з двох джерел:

1. Реальні екземпляри даних, такі як реальні фотографії людей. Дискримінатор використовує ці екземпляри як позитивні приклади під час навчання.
2. Підроблені екземпляри даних, створені генератором. Дискримінатор використовує ці випадки як негативні приклади під час навчання.

На рис.2.2. два поля "Зразок" представляють ці два джерела даних, що надходять у дискримінатор. Під час навчання дискримінатора генератор не навчається. Його вага залишається незмінною, хоча він дає приклади для того, щоб дискримінатор навчався.

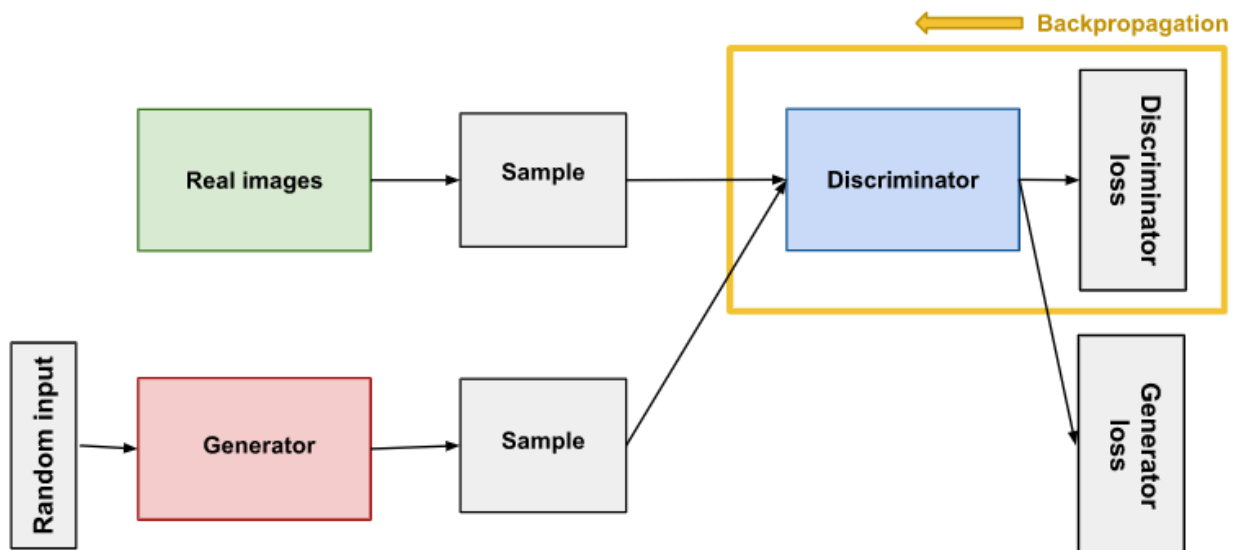


Рисунок 2.2. Зворотне розповсюдження у навчанні дискримінатора

### 2.1.1.2. Навчання дискримінатора

Дискримінатор має дві функції втрат. Під час навчання дискримінатор ігнорує втрати генератора і просто використовує втрати дискримінатора:

1. Дискримінатор класифікує як реальні, так і підроблені дані генератора.
2. Втрата дискримінатора карає дискримінатора за неправильну класифікацію реального екземпляра як фальшивого або фальшивого екземпляра як реального.
3. Дискримінатор оновлює свої ваги шляхом зворотного розповсюдження від втрати дискримінатора через мережу дискримінатора.

### 2.1.2. Генератор генеративної змагальної мережі

Частина генеративної змагальної мережі, що зветься генератор і зображена на рис.2.3., вчиться створювати підроблені дані, враховуючи зворотний зв'язок від дискримінатора. Він вчиться щоб змусити дискримінатор класифікувати його результати як реальні дані.

Навчання генератора вимагає більш тісної інтеграції між генератором і дискримінатором, ніж вимагає навчання дискримінатора. Частина генеративної змагальної мережі, яка навчає генератор, також включає наступні пункти:

1. випадковий вхід;
2. мережа генератора, яка перетворює випадковий вхід у екземпляр даних;
3. мережа дискримінатора, яка класифікує генеровані дані;
4. вихід дискримінатора;
5. втрата генератора, яка штрафує генератор за те, що йому не вдалося обдурити дискримінатора.

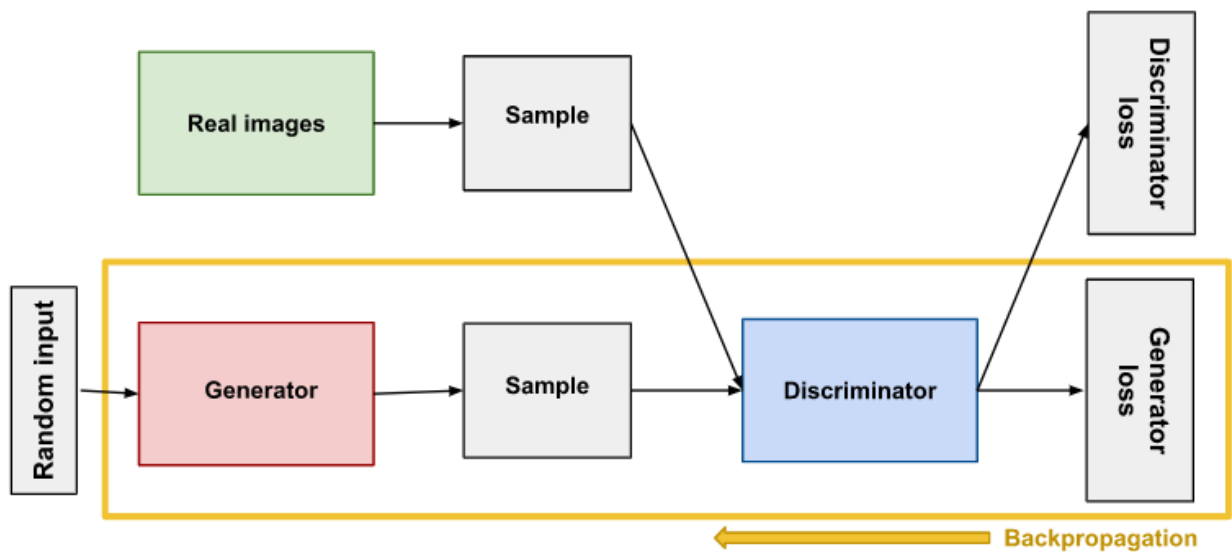


Рисунок 2.3. Зворотне розповсюдження у навчанні генератора

### 2.1.2.1. Випадковий вхід генеративної змагальної мережі

Нейронні мережі потребують вхідні дані певної форми. Зазвичай вводяться дані, з якими необхідно щось робити, як, наприклад, екземпляр, який необхідно класифікувати. Але виникає питання що використовується як вхід для мережі, яка видає абсолютно нові екземпляри даних.

У своїй найпростішій формі GAN приймає випадковий шум як свій вхід. Потім генератор перетворює цей шум у значущий вихід. Вводячи шум, можна змусити GAN виробляти найрізноманітніші дані, відбираючи вибірки з різних місць цільового розподілу.

Експерименти дозволяють припустити, що розподіл шуму не має великого значення, тому можна вибрати те, з чого легко взяти вибірку, як

рівномірний розподіл. Для зручності простір, з якого відбирають шум, зазвичай має менші розміри, ніж розмірність вихідного простору.

#### 2.1.2.2. Використання дискримінатора для навчання генератора

Для тренування нейронної мережі змінюються ваги мережі, щоб зменшити похибку або втрату її виходу. Однак генератор не пов'язаний безпосередньо із втратами, на які необхідно вплинути. Генератор подає дані в дискримінаційну мережу, а дискримінатор видає результат, на який необхідно вплинути. Втрата генератора штрафує генератор за отримання зразка, який дискримінаційна мережа класифікує як підроблений.

Цей додатковий фрагмент мережі повинен бути включений у зворотне розповсюдження. Зворотне розповсюдження регулює кожен вагу в потрібному напрямку, обчислюючи вплив ваги на вихід, як змінився вихід, якби змінився вагу. Але вплив ваги генератора залежить від впливу ваги дискримінатора, на яку він подається. Отже, зворотне розповсюдження починається на виході і повертається через дискримінатор в генератор.

У той же час, дискримінатор не повинен змінюватись під час підготовки генераторів. Тож генератор навчається за наступною процедурою:

1. Подається зразок випадкового шуму.
2. Генерується вихід генератора з випадкового шуму.
3. Отримується "Справжній" або "Підроблений" класифікація для генераторного виходу.
4. Розраховується втрата від класифікації дискримінатора.
5. Вираховується зворотне розповсюдження через дискримінатор і генератор, щоб отримати градієнти.
6. Використовуються градієнти для зміни лише ваги генератора.

Це одна ітерація підготовки генератора.

### 2.1.3. Навчання генеративної змагальної мережі

Оскільки генеративні змагальні мережі містить дві окремо підготовлені мережі, його алгоритм навчання повинен вирішувати наступну проблему. GAN повинен жонглювати двома різними видами навчання (генератор та дискриміратор).

#### 2.1.3.1. Почергове навчання

Генератор та дискриміратор мають різні навчальні процеси. Тоді як тренувати генеративні змагальні мережі в цілому? Навчання генеративних змагальних триває в почергові періоди:

1. Дискриміратор тренує одну або кілька епох.
2. Генератор тренує одну або кілька епох.
3. Кроки 1 і 2 повторюються, щоб продовжити навчання мережі генератора та дискриміатора.

Генератор підтримується постійним під час фази навчання дискриміатора. Оскільки навчання дискриміатора намагається розібратися, як відрізнити реальні дані від фальшивих, воно має навчитися розпізнавати вади генератора. Це зовсім інша проблема для ретельно навченого генератора, ніж для непідготовленого генератора, який виробляє випадковий вихід.

Так само підтримується постійний дискриміратор під час фази навчання генератора. Інакше генератор може ніколи не зійтись.

Саме цей рух вперед і назад дозволяє генеративним змагальним мережам вирішувати інакше нерозв'язні генеративні проблеми. Отримується опора у складній генеративній задачі, починаючи зі значно простішої проблеми класифікації. І навпаки, якщо не можна навчити класифікатор визначати різницю між реальними та згенерованими даними навіть для початкового вихідного генератора, не можна розпочати навчання генеративних змагальних мереж.

## 2.2. Проектування моделі генеративної мережі

Для даної роботи використано модифікацію GAN мережі під назвою DCGAN, яка зображена на рис.2.4. Основним для даного підходу є прийняття та модифікація трьох нещодавно продемонстрованих змін в архітектурі CNN.

Перша – це повністю згорткова мережа, яка замінює детерміновані просторові об'єднання функцій з різкими згортками, що дозволяє мережі навчатися на власному просторовому спаді. Цей підхід використовується у генераторі, що дозволяє йому дізнатися власну просторову оптимізацію та дискримінатор.

Друга – тенденція до усунення повністю зв'язаних шарів, окрім згорткових особливостей. Перший шар GAN, який приймає рівномірний розподіл шуму  $Z$  на вхід, можна назвати повністю зв'язним, оскільки це просто матричне множення, але результат перетворюється на 4-мірний тензор і використовується як початок стека згортки. Для дискримінатора останній шар згортки сплющується і потім подається в єдиний сигмоподібний вихід.

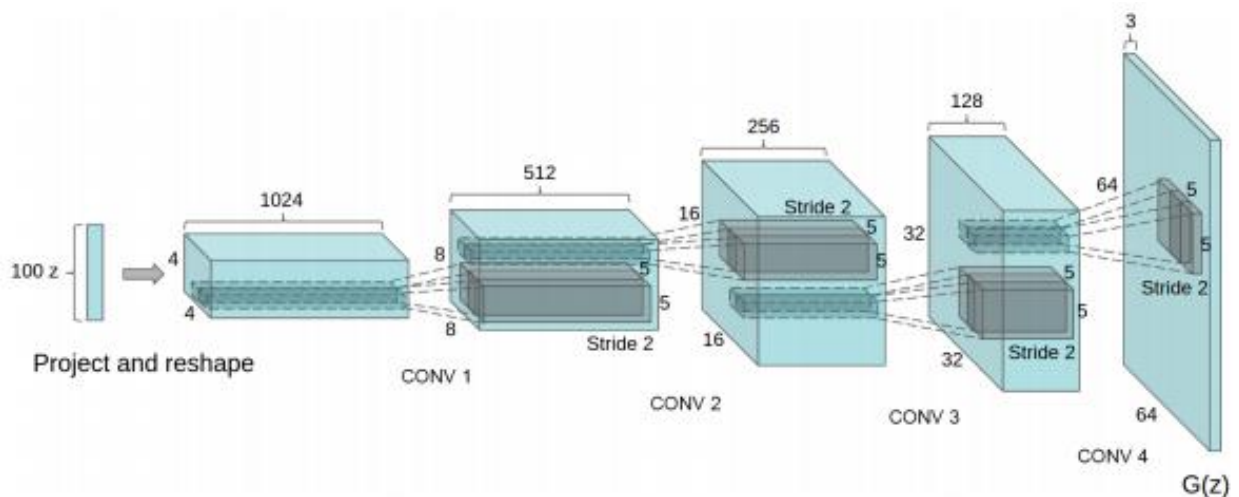


Рисунок 2.4. DCGAN генератор

Третя – це нормалізація партії (Batch Normalization), яка стабілізує навчання шляхом нормалізації входу до кожної одиниці, щоб мати нульову середню і дисперсію. Це допомагає вирішити проблеми з навчанням, які виникають через погану. Безпосереднє використання нормалізації на всі шари

призвело до коливання вибірки та нестабільності моделі. Цього можна уникнути не застосовуючи нормалізацію до вихідного шару генератора та вхідного рівня дискримінатора.

Активація ReLU [40] використовується в генераторі за винятком вихідного шару, який використовує функцію тангенса. Використання обмеженої активації дозволяє моделі навчитися швидше.

Для стабільних глибоких згорткових GAN використано таку архітектуру:

1. Замінено будь-які шари об'єднання суворими згортками (дискримінаторами) та дробовими шарами згортки (генератор).
2. Використано batchnorm і в генераторі, і в дискримінаторі.
3. Видалено повністю зв'язані приховані шари для глибоких архітектур.
4. Використано активацію ReLU в генераторі для всіх шарів, крім виходу, в якому використовується Tanh (тангенс).
5. Використано активацію LeakyReLU в дискримінаторі для всіх шарів.

LeakyReLU (протікаючий ReLU, LReLU) представляє собою одну зі спроб, вирішити проблему виходу із строю звичайних ReLU. Звичайний ReLU рис.2.5. справа в інтервалі  $x < 0$  дає на виході нуль, у той час як LeakyReLU рис.2.5. злів має на цьому інтервалі невелике негативне значення (кутовий коефіцієнт близько 0,01).

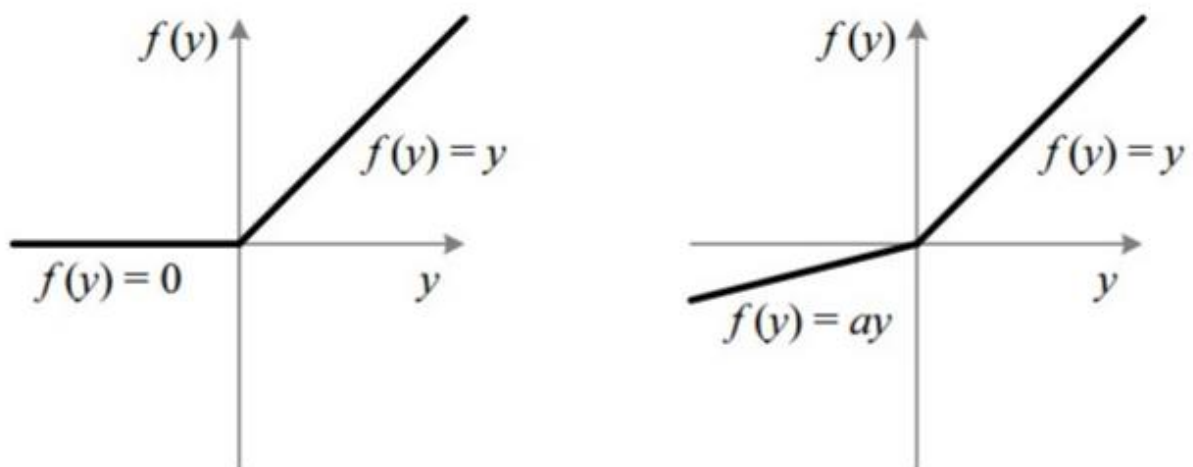


Рисунок 2.5. Функції активації

Тобто для LeakyReLU існує функція  $f(x) = \alpha x$  при  $x < 0$  та  $f(x) = x$  при  $x \geq 0$ , де  $\alpha$  – мала константа. Деякі дослідники повідомляють про успішний приклад застосування даної функції активації, але результати не завжди стабільні.

При навчанні DCGAN попередня обробка не застосовувалася до тренувальних зображень, крім масштабування до діапазону активації функції тангенса  $[-1, 1]$ . Усі моделі пройшли навчання з міні-серійним стохастичним градієнтним спуском (SGD, більш детально в пункті 3.2) з розміром міні-партії 128. Усі ваги були ініціалізовані з нормованого розподілу, орієнтованого на нуль при стандартному відхиленні 0,02. У LeakyReLU у всіх моделях ухил витоку встановлювався на 0,2. Використано оптимізатор Adam з налаштованими гіперпараметрами. Швидкість навчання 0,0002. Крім того терміну імпульсу  $\beta_1$  був встановлений 0,5.

#### Висновки до розділу

У даному розділі описано компоненти інформаційного забезпечення, що були обрані для проведення експерименту над генеративними можливостями інтелектуальних робототехнічних систем. Наведений загальний опис компонентів, описані принципи роботи деяких компонентів. Наведений перелік допоміжних елементів, таких як бібліотеки та методи. Також описана модель мережі.

### 3. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1. Вибір середовища та засобів розробки

Нейронна мережа, описана в даній роботі була реалізована за допомогою мови програмування *Python 2.7* [37]. Також використано бібліотеки *SciPy*, *pillow*, *Tensorflow 0.12.1* фреймворк [38].

Мова програмування *Python* є потужним інструментом для створення програм найрізноманітнішого призначення, доступний навіть для новачків. З його допомогою можна вирішувати завдання різних типів. Мова *Python* має деякі примітні особливості, які обумовлюють його широке розповсюдження. Далі буде наведено перелік переваг, на які зверталася увага та завдяки яким обрано дану мову програмування.

1. *Python* – інтерпретована мова програмування. З одного боку, це дозволяє значно спростити налагодження програм, з іншого – зумовлює порівняно низьку швидкість виконання.
2. Динамічна типізація, тобто не треба заздалегідь оголошувати тип змінної, що дуже зручно при розробці.
3. Гарна підтримка модульності. Можна легко написати свій модуль і використовувати його в інших програмах.
4. Підтримка об'єктно-орієнтованого програмування. При цьому його реалізація в *Python* є однією з найбільш зрозумілих.
5. Автоматичне прибирання сміття, відсутність витоків пам'яті.
6. Інтеграція з C / C ++, якщо можливостей *Python* недостатньо.
7. Зрозумілий та лаконічний синтаксис, що сприяє ясному відображенню коду. Зручна система функцій дозволяє при грамотному підході створювати код, в якому буде легко розібратися іншій людині в разі необхідності. Також можна читати програми і модулі, написані іншими людьми.
8. Величезна кількість модулів, як входять в поставку *Python*. У деяких випадках для написання програми досить лише знайти підходящі модулі

і правильно їх скомбінувати. Таким чином, можна думати про складання програми на більш високому рівні, працюючи з уже готовими елементами, які виконують різні дії.

9. Кросплатформеність. Програма, написана на *Python*, буде функціонувати абсолютно однаково незалежно від того, в якій операційній системі вона запущена. Відмінності виникають лише в рідкісних випадках, і їх легко заздалегідь передбачити завдяки наявності докладної документації.

*TensorFlow* – бібліотека програмного забезпечення з відкритим кодом для чисельних обчислень з використанням графів потоку даних. Вузли на графу представляють математичні операції, в той час як краї графіки представляють багатовимірні масиви даних (тензори), що протікають між ними. Ця гнучка архітектура дозволяє розгортати обчислення на одному чи декількох процесорах чи графічних процесорах на робочому столі, сервері чи мобільному пристрої без перезапису коду.

*TensorFlow* – найпопулярніша бібліотека з усіх у сфері машинного навчання, оскільки вона створена для того, щоб бути доступною для всіх. Бібліотека *Tensorflow* містить різні API, побудовані на масштабній архітектурі глибокого навчання, такі як CNN або RNN. *TensorFlow* заснований на обчислених графах, це дозволяє розробнику візуалізувати побудову нейронної мережі за допомогою Tensorboard. Цей інструмент корисний для налагодження програми. Нарешті, *Tensorflow* побудований для розміщення в масштабі. Він працює на процесорі та GPU.

Для роботи з кодом обрано IDE *PyCharm* [39]. Це інтелектуальний редактор коду, наданий *PyCharm*, що дозволяє програмістам писати високоякісний код на *Python*. Редактор дає можливість програмістам легко читати код за допомогою кольорових схем, автоматично вставляти відступи в нових рядках, вибирати відповідний стиль кодування та користуватися контекстними пропозиціями щодо заповнення коду. Параметри навігації з

розумним кодом, які надаються *PyCharm*, допомагають програмістам редагувати та вдосконалювати код, не витрачаючи зайвих витрат часу та сил.

*PyCharm* полегшує розробникам швидке та ефективно впровадження як локальних, так і глобальних змін, допомагає писати різні веб-додатки в *Python*, підтримуючи широко використовувані веб-технології, такі як HTML, CSS, *JavaScript*, *TypeScript* та *CoffeeScript*. Окрім підтримки широко використовуваних веб-технологій, *PyCharm* також надає першокласну підтримку для міцної веб-бази *Python*, як *Django*. Також крім підтримки різних бібліотек та фреймворків *Python*, *PyCharm* дозволяє розробникам працювати з низкою реляційних баз даних, включаючи *Oracle*, *SQL Server*, *MySQL* та *PostgreSQL*. Візуальний налагоджувач, наданий IDE, допомагає налагоджувати код *Python*, *JavaScript* та *Django*.

### 3.2. Огляд навчальних даних

Для навчання нейронної мережі і отримання значущих результатів необхідні великі набори даних. Це є загальновідомою істиною у сфері навчання нейронних мереж. Адже маючи більший набір даних (більше спостережень, більші часові рамки), можна отримати більше підходів до аналізу, зокрема, дозволяє використати прогностичну аналітику. Також на великих наборах даних краще можна визначити статистичні дані та залежності. Взагалі, релевантні залежності можна отримати лише на великих наборах даних, про що свідчить закон великих чисел. Також більший набір даних покриває більше інформаційних варіантів.

Для навчання нейронної мережі, архітектуру якої описано в даному розділі необхідний набір зображень логотипів для навчання. На даний момент в інтернет мережі можна вільно мати доступ до наступних наборів даних, представлених у табл.3.1.

Найбільш репрезентативні із представлених великих наборів даних із загальнодоступними логотипами представлені в табл.3.1.

Таблиця 3.1. Список наборів даних

Набір даних	Логотипів	Зображень
FlickLogos-27	27	1080
FlickLogos-32	32	8240
BelgaLogos	37	10000
LOGO-Net	160	73414
WebLogo-2M	194	1867177
LLD	486377+	609297

Через малу різноманітність логотипів, ці набори даних не підходять для вивчення та перевірки автоматичних генераторів логотипів. У той же час ряд веб сторінки дозволяють платно отримувати доступ до великої кількості логотипів, таких як [iconsdb.com](http://iconsdb.com) (4135+ іконок), [icons8.com](http://icons8.com) (59900+), [iconfinder.com](http://iconfinder.com) (7473+), [iconarchive.com](http://iconarchive.com) (450k +) та [thenounproject.com](http://thenounproject.com) (1м +). Однак різноманітність цих логотипів є обмежена кількістю джерел, а саме дизайнерів / художників, тем, категорій та візерунків, дизайну (багато з них чорно-білі).

На відміну від популярних наборів даних природних зображень, таких як ImageNet, CIFAR-10 та LSUN, набори даних обличч, як CelebA, рукописні цифри MNIST, логотипи є:

1. Штучні але сильно мультимодальні і, таким чином, складні для генеративної моделі.
2. Прикладні, оскільки існує очевидний попит у реальному світі на синтетично створені унікальні логотипи з тих пір, як вони дорогі у виготовленні.
3. Важкі у маркуванні, оскільки існує дуже мало категоричних властивостей, які проявляються у візуальному вигляді логотипу.



представлений сотнями і тисячами зображень. На даний момент існує в середньому понад п'ятсот зображень на вузол. На серпень 2017 року в ImageNet 14 197 122 зображень, розбитих на 21 841 категорій. Приклад зображень, що входять в набір даних представлено на рис.3.2.



Рисунок 3.2. Набір даних ImageNet

### 3.3. Навчання моделі

Для навчання моделі використовувався стохастичний градієнтний спуск. Це ітераційний метод оптимізації градієнтного спуску за допомогою стохастичного наближення. Його використовують для прискорення знаходження цільової функції, використовуючи обмежений тренувальний набір.

Слово "стохастичний" означає систему або процес, пов'язаний з випадковою ймовірністю. Отже, у стохастичному градієнтному спуску кілька зразків вибираються випадковим чином, а не весь набір даних для кожної ітерації. У Gradient Descent існує термін під назвою "batch", який позначає загальну кількість зразків із набору даних, який використовується для обчислення градієнта для кожної ітерації. Хоча використання всього набору даних є дуже корисним для отримання мінімумів менш зашумленим або менш випадковим чином, але проблема виникає, коли набори даних стають дійсно великими.

Припустимо, у наборі даних є мільйон зразків, тому якщо використовувати типову техніку оптимізації градієнтного спуску, доведеться використовувати всі мільйон зразків для виконання однієї ітерації під час

виконання градієнтного спуску, і це потрібно зробити для кожно ітерацію до досягнення мінімумів. Отже, виконувати обчислювально дуже затратно.

Цю проблему вирішує Stochastic Gradient Descent [43]. У SGD для виконання кожної ітерації використовується лише один зразок, тобто розмір партії одиниці. Зразок довільно перемішують і відбирають для проведення ітерації.

### 3.4. Результати роботи генеративної мережі

Мета описаної генеративної мережі полягає у створенні основи для системи з можливістю генерувати нескінченно багато варіантів логотипів для полегшення та прискорення процесу створення логотипів. На рис.3.3. зображено приклад отриманих зображень. Верхні три рядки логотипів є оригінальними зображеннями з набору даних а нижні три рядки є логотипами згенерованими мережею.



Рисунок 3.3. Приклад роботи мережі

Крім того, для полегшення роботи з мережею, передбачено інтерфейс користувача, що зображено на рис.4.4. Потенційний користувач має можливість модифікувати прототип логотипу відповідно до конкретних параметрів, таких як форма та колір, або налаштовувати його характеристики відносно іншого прототипу. Це може допомогти, наприклад дизайнеру та клієнту, які співпрацюють для створення логотипу бренду, щоб отримати уявлення про потенційний логотип, який дизайнер міг би тоді доопрацювати,

навіть якщо сама система ще не здатна створити високоякісну конструкцію. Даний інструмент значно полегшить роботу людей, які працюють з логотипами, створюють їх. Доцільність розробки даного функціоналу також описано у розділі стартапу.

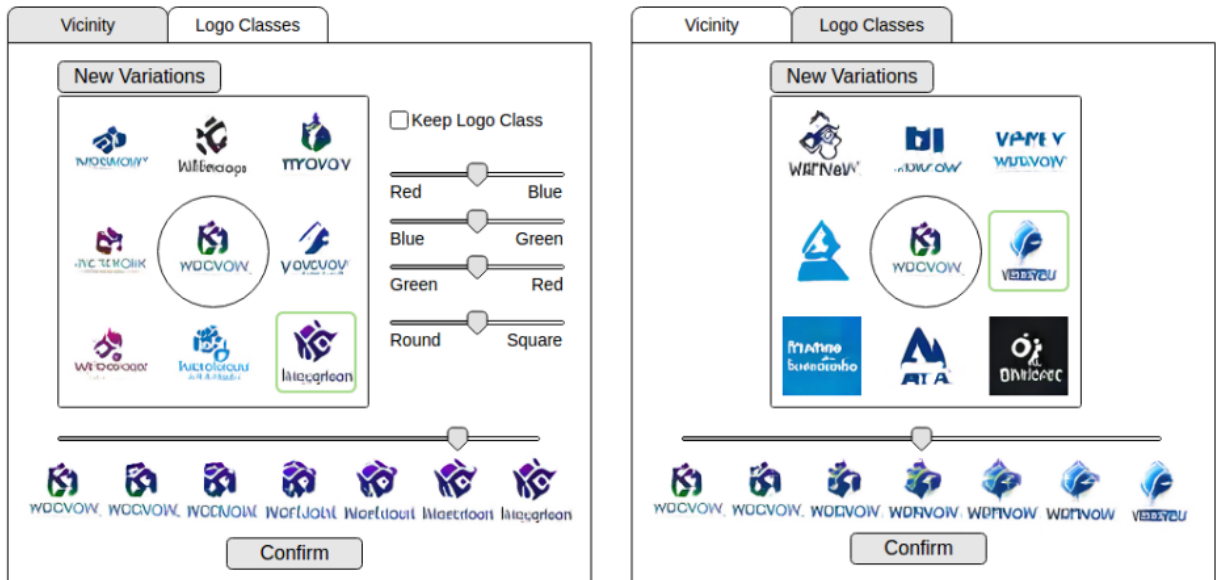


Рисунок 3.4. Інтерфейс користувача

### 3.5. Експериментальне дослідження генеративної мережі

Мета експерименту полягає в аналізі існуючих генеративних архітектур з точки зору генерації новизни. Більшість генеративних моделей походять з великого класу архітектур, іноді цілеспрямовано розроблених для того, щоб не «поводитись неправильно». Тобто відсікати будь-який прояв справжніх генеративних можливостей. По можливості ці моменти, розроблені для уникнення генерації "хибних" об'єктів були перетворені у опціональні гіперпараметри.

У розділі експерименту піднімається питання про аспект дослідження генеративних моделей в цілому. Як уже було розглянуто в першому розділі, при розробці генеративних мереж зазвичай моделі налаштовують так, що будь-який прояв генеративних можливостей, відхилення від заданого шаблону

генерації карається методами оцінки при навчанні такої мережі. У даному експерименті навпаки, мережі дозволяється генерувати нові зображення.

У експерименті проводиться аналіз генеративної мережі GAN за допомогою метода оцінки «Початкова оцінка» та візуального тесту Тьюрінга. Після проведення експерименту буде проведено аналіз того, наскільки можливо на даному етапі розвитку алгоритмів оцінки та архітектур генеративних моделей досліджувати генеративні можливості інтелектуальних робототехнічних систем.

Для аналізу результатів роботи проведено експеримент з трьома різними налаштуваннями мережі та її гіперпараметрів. Потім відібрано 100 згенерованих зображень і перевірено двома методами оцінки.

Конфігурації експериментів були наступні:

1. Для мережі були встановлені вказані нижче в табл.4.1. налаштування та гіперпараметри. Тобто, мережа була налаштована для нормальної роботи з оптимальними, підібраними параметрами. Так, як цю мережу можна використовувати для рішення прикладних задач з генерації зображень.
2. Мережа навмисно була налаштована так, щоб відбулося перенавчання. Тому на виході було отримано зображення із навчального набору. З'ясовано, що налаштування (вибір) генеративних моделей для генерації розподілу змусить їх «запам'ятати» класи об'єктів, на яких вони навчаються. Це є нормальним явищем але важливо зазначити, що це означає, що генерація абсолютно нових об'єктів є нетривіальною задачею, і переважна більшість архітектур, розроблені та налаштовані так, що не створюють по справжньому нових об'єктів.
3. Для мережі були встановлені гіперпараметри, що не є оптимальними, щоб зображення генерувалися якомога гірші.

### 3.5.1. Налаштування параметрів мережі

У даній роботі експерименти базувались на роботі [26]. Також використано їх архітектуру як основу для пошуку гіперпараметрів. У табл.3.2. наведені гіперпараметри та їхні припущення.

Таблиця 3.2. Припущення гіперпараметра GAN

Назва	Припущення	Тип
nb. discr. updates	1, 2, 3	Choice
l2 coedicient	$[10^{-6}, 10^{-1}]$	Logspace
Gen. input dim.	10...200	Choice
nb. fc. gen. units	$2^i, i=3...11$	Choice
nb. fc. discr. units	$2^i, i=3...11$	Choice
nb. filters gen	$2^i, i=3...9$	Choice
nb. filters discr.	$2^i, i=3...9$	Choice
Learning rate	$[10^{-6}, 10^{-1}]$ або 0.0002	Logspace
Weight initialization	Std $\in [10^{-6}, 10^{-1}]$	Logspace

### 3.5.2. Дискримінатор генеративної мережі

Для обчислення початкової оцінки підготовлено 3 дискримінатори. Перший був навчений класифікувати зображення LLD, другий – класифікувати зображення ImageNet, а третій – класифікувати між сумішшю цими двома наборами даних.

Архітектура, що використовується у всіх дискримінаторах, являла собою згорткову нейронну мережу з 3-ма згортковими шарами, на кожному з яких слідувало максимальне об'єднання (max pooling). Кількість фільтрів для 3-х згорткових шарів відповідно 32, 64 та 128. Використано фільтри розміром 3. Останній згортковий шар супроводжувався трьома повністю з'єднаними шарами з 512, 256 та 128 прихованими юнітами. Використано PReLU функцію активації та відкидання (dropout) з  $p = 0,5$  у кожному згортковому і повністю зв'язаному шарі.

### 3.6. Результати експерименту та їх аналіз

Після запуску навченої генеративної моделі отримано результати у вигляді зображень логотипів. Зображення очевидно мають різну якість як у візуальному плані, так і у смисловому. Нижче приведено приклади деяких згенерованих зображень для кожного експерименту:

#### 1. Результати першого експерименту.

Як видно з результатів експерименту на рис.3.5., результати роботи мережі є досить вдалимими. Логотипи досить якісні, тобто мають чіткі форми, яскраві кольори, лінії не змішуються між собою. Суб'єктивно можна оцінити, що ці логотипи схожі на логотипи що створюють люди.



Рисунок 3.5. Результати першого експерименту

#### 2. Результати другого експерименту.

Як видно з результатів експерименту на рис.3.6., результати роботи мережі є кращими ніж у першому експерименті. Це очікуваний результат, адже відбулося запам'ятовування вхідних даних мережею. Тобто зображення, згенеровані мережею відповідають зображенням набору тренувальних даних.



Рисунок 3.6. Результати другого експерименту

### 3. Результати третього експерименту.

Як видно з результатів експерименту на рис.3.7. результати роботи мережі є гіршими ніж у попередніх експериментах. Цей результат також є очікуваним, адже гіперпараметри були налаштовані для гіршої генерації. Проте, зображення логотипів все ж таки залишаються досить вдалими.



Рисунок 3.7. Результати третього експерименту

Після отримання результатів у вигляді згенерованих зображень був проведений їх аналіз. Для кожного експерименту застосовано два методи оцінки, їх результати описані нижче в табл.3.3.

Таблиця 3.3. Оцінка експериментів

Експеримент	№1	№2	№3
Початкова оцінка	8,11	11,2	6,24
Тест Тюрінга	70%	100%	50%

На основі цих результатів можна сказати, що початкова оцінка добре корелюється із суб'єктивною людською оцінкою якості згенерованих зображень, яка була представлена у вигляді тесту Тьюрінга. Зображення, що відображали зображення із тренувального набору отримали максимальну оцінку. Мережа, що генерує нові зображення і має налаштовані гіперпараметри отримала гіршу оцінку за обома методами оцінки ніж перший випадок мережі, але вищу оцінку ніж мережа з налаштованими в гіршу сторону параметрами.

### 3.7. Перспективи використання мережі та методів оцінки

Експеримент проведений у цій роботі, а конкретніше, його результати показують, що можна використовувати метрики для оцінки генеративних мереж, які усувають необхідність участі людини. Як видно з табл.4.2. порівняння оцінок мережі, існує кореляція між людською оцінкою. Відповідно чим вище було оцінено якість зображень логотипів людиною, тим вища була «початкова оцінка». Даний результат говорить про те, що можливо автоматизувати процес оцінки генеративних мереж, що в свою чергу підвищує ефективність роботи з такими мережами на порядки. Також це дає можливість усунути людський фактор як суб'єктивний, такий що не формалізується. Тому є сенс покращувати використану в даній роботі метрику оцінки мережі для досягнення кращих результатів, більшої точності. А також є сенс досліджувати інші методи оцінки та порівнювати їх між собою.

#### Висновки до розділу

У даному розділі проаналізовано необхідні для дослідження набори даних. Проведено їх порівняння та обрано найкращий варіант для проведення експерименту. Також описано підхід до навчання генеративної моделі. У даному розділі описано процес дослідження генеративних можливостей інтелектуальних робототехнічних систем.

Поставлена мета, задача дослідження. Була запропонована конфігурація з трьох експериментів над генеративною мережею з різними налаштуваннями. Потім показано налаштування стандартного вигляду мережі. Після проведення серії експериментів представлено результати у вигляді згенерованих зображень для кожної експериментальної конфігурації. На основі отриманих результатів була проведена оцінка за допомогою двох методів, а саме початкова оцінка та візуального тесту Тьюрінга.

Оцінки представлені в табл.4.2. показали, що існує кореляція між людським судженням та розробленою оцінкою. На основі цих результатів

зроблено висновок, що є сенс продовжувати дослідження в цій області, в області оцінок генеративних мереж.

В загальному результати отримані в розділі кажуть про те, що дослідження генеративних мереж є актуальною темою. Адже є підхід до кількісної оцінки генеративних можливостей цих мереж. Скоріше за все, людська оцінка ще довгий час буде однією з основних оцінок, одним з головних критеріїв судження про генеративні можливості.

#### 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

Процес розробки та виведення стартап-проекту на ринок має містити виконання деяких кроків, в яких визначаються ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів.

Метою розділу є формування інноваційного мислення, підприємницького духу та формування здатностей щодо оцінювання ринкових перспектив і можливостей комерціалізації основних науково-технічних розробок, сформованих у попередній частині магістерської дисертації у вигляді розроблення концепції стартап-проекту в умовах висококонкурентної ринкової економіки глобалізаційних процесів.

В загальному, етапи розробки стартап-проекту можна представити наступним чином:

##### 1) Маркетинговий аналіз стартап-проекту.

В межах цього етапу:

- відбувається опис ідеї проекту та визначаються загальні напрямки використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- аналізуються можливості ринку щодо реалізації проекту;
- на основі аналізу середовища ринку розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

##### 2) Організація стартап-проекту.

В межах даного етапу:

- розробляється календарний план-графік реалізації проекту;
- розраховується необхідні основні засоби та нематеріальні активи;
- розраховується запланований обсяг виробництва потенційного товару, на основі чого розраховується потреба у матеріальних ресурсах та персоналі;

- вираховуються загальні початкові витрати для запуску стартап-проекту та заплановані загальногосподарські витрати, необхідні для реалізації стартап-проекту.

### 3) Фінансово-економічний аналіз та оцінка ризиків проекту.

В межах цього етапу:

- вираховується обсяг інвестиційних затрат;
- вираховуються основні фінансово-економічні показники стартап-проекту (обсяг виробництва, собівартість виробництва продукції, ціна реалізації, податкове навантаження, чистий прибуток) та визначаються показники інвестиційної привабливості проекту (запас фінансової міцності, рентабельність продажів та інвестицій, період окупності стартап-проекту);
- вираховується рівень ризикованості стартап-проекту, визначаються головні ризики стартап-проекту та шляхи їх уникнення (реагування на ризики).

### 4) Заходи з комерціалізації проекту

Цей етап направлений на пошук інвесторів та просування інвестиційної пропозиції (оферти) проекту.

Він передбачає:

- визначення цільових груп інвесторів та опис їх ділових інтересів;
- складання інвестиційної пропозиції (оферти), тобто стислої характеристики стартап-проекту для попереднього ознайомлення інвесторів;
- планування заходів щодо просування оферти: визначення каналів комунікації, планування заходів з просування в рамках обраних каналів;
- планування ресурсів для реалізації заходів щодо просування оферти.

Визначені етапи, реалізовані вчасно та послідовно створюють передумови для успішного старту проекту на ринку.

#### 4.1. Опис ідеї проекту

В рамках підпункту було проаналізовано і подано у вигляді таблиць:

- зміст ідеї (що пропонується);
- можливі напрямки застосування;
- основні вигоди, які може отримати користувач товару (за кожним напрямком застосування);
- чим товар відрізняється від існуючих аналогів або замінників.

Зміст ідеї проекту, що дає в цілому картину запропонованого рішення наведено у табл. 5.1.

Таблиця 5.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка WEB-додатку для генерації логотипів за допомогою нейронних мереж.	<ol style="list-style-type: none"> <li>1. Генерування логотипів компаній.</li> <li>2. Генерування логотипів брендів</li> </ol>	<ol style="list-style-type: none"> <li>1. Генерація безкінечної кількості логотипів за необхідністю</li> <li>2. Унікальність логотипів</li> <li>3. Відсутність авторських прав на логотипи</li> </ol>

Аналіз потенційних техніко-економічних переваг проекту (чим відрізняється від існуючих аналогів та замінників) у порівнянні з конкурентними пропозиціями передбачає:

- визначення набору техніко-економічних властивостей та характеристик проекту;
- визначення попереднього набору конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та збір інформації щодо значень техніко-економічних показників для ідеї проекту та конкурентних проектів відповідно до зазначеного вище переліку;
- проведення порівняльного аналізу показників.

Для майбутнього проекту визначені показники, що мають:

- а) гірші показники (W, слабкі);
- б) аналогічні показники (N, нейтральні);
- в) кращі показники (S, сильні) (табл.5.2).

Таблиця 5.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкуренти			
1	Формат сервісу	Web-App	Web-App		+	
2	Кросплатформеність	+	+		+	
3	Вартість генерації	Безкоштовно	Безкоштовно		+	
4	Налаштування	-	+			+
5	Унікальність	+	+		+	
6	Кількість генерацій	Безлімітна	Зменшується з кількістю	+		

Зазначений вище перелік сильних, слабких та нейтральних характеристик потенційного проекту є основою для формулювання його конкурентоспроможності.

#### 4.2. Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару).

Визначення того, чи можливе здійснення ідеї проекту з технологічної точки зору, тобто визначення технологічної здійсненності проекту, передбачає аналіз таких складових (табл.5.3):

- технологія виготовлення товару згідно проекту;
- чи існують технології для виготовлення товару;
- доступність технологій виготовлення авторам проекту.

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	WEB-додаток	HTML, CSS, JavaScript, React.js, PHP	+	+
2	Нейронні мережі	Mongo DB, Tensorflow.js, Python,	+	+

За результатами аналізу таблиці робиться висновок щодо можливості технологічної реалізації проекту: так чи ні, а також технологічного шляху, яким це доцільно зробити (з поміж названих технологій обираються такі, що доступні авторам проекту та є наявними на ринку).

Висновок: реалізація продукту з технологічної точки зору можлива.

#### 4.3. Аналіз ринкових можливостей запуску стартап-проєкту

Визначення можливостей ринку, які можна використати під час впровадження проєкту на ринку. Аналіз ринкових загроз, які можуть перешкодити успішній реалізації стартап-проєкту, що дозволяє спланувати напрями розвитку проєкту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проєктів-конкурентів. Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку табл.5.4.

Таблиця 5.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн./ум.од	1000000
3	Динаміка ринку	зростає
4	Наявність обмежень для входу	відсутні
5	Специфічні вимоги до стандартизації та сертифікації	відсутні
6	Середня норма рентабельності в галузі або по ринку, %	R=30%

Висновок: враховуючи яка існує кількість головних гравців на ринку, а також зростаючу динаміку ринку та середню норму рентабельності є можливість зробити висновок, що ринок є привабливим для реалізації проєкту.

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Потреба у створенні логотипів	Будь-яка компанія, бренд	Потребують високої якості логотипів. Потребують закладати значення в логотип Готові замовляти індивідуальні логотипи.	Логотипи повинні бути високої якості зображення

Після проведення аналізу потенційних груп клієнтів було визначено умови ринкового середовища. Для цього було створено таблиці з факторами, що сприяють впровадженню проекту на ринок, та факторів, що цьому перешкоджають (табл.5.6, табл.4.5).

Таблиця 5.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Наявність конкурентів котрі пропонують подібні рішення	1. Розробка унікальних характеристик товару; 2. Збільшення якості товару
2	Інвестиції	Недостатнє фінансування	1. Залучення додаткових інвесторів; 2. Краудфандинг
3	Технічний	Складність технологій	Інвестиції в технології
4	Кадровий	Відсутність кваліфікованих кадрів	Залучення фахівців зі сторони

Таблиця 5.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Надання нових рішень	Розробка нового продукту; Вихід продукції на ринок; Захоплення максимальної долі ринку
2	Науково-технічний	Стрімкий розвиток технологій	Використання нових технологій
4	Монетизація реклами	Розміщення реклами на застосунку	Отримання доходу від розміщення реклами

Далі було проведено аналіз пропозиції. Було визначено загальні риси ринкової конкуренції (табл.5.8).

Таблиця 5.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Існують конкуренти на ринку. Товар конкурентів принципово відрізняється від запропонованого товару	Розробка якісного продукту, щоб запропонувати альтернативні рішення клієнтам
2	Рівень конкурентної боротьби: світовий	Всі конкуруючі компанії знаходяться в будь-якому місці і пропонують свій товар в інтернеті для будь-якої аудиторії	Вихід на ринок з принципово новим підходом до рішення проблеми; Впровадження маркетингових стратегій на основних Інтернет ресурсах задля охоплення максимальної кількості потенційних користувачів; Роз'яснення серед користувачів унікальності продукту.
3	Галузева ознака: міжгалузева	Запропонований продукт може використовуватися в будь-якій сфері діяльності компаній	Розробка максимально зрозумілого інтерфейсу користувача; Врахування розмірів аудиторії; Відсутність концентрації на окремих типах клієнтів
4	Конкуренція за видами товарів: товарно-видова	Конкуренція між товарами одного виду.	Розробка функціоналу, що відрізняється від технологій конкурентів за своєю суттю.
5	Характер за конкурентними перевагами: не цінова	Не цінова – надання функціональності, що відрізняється від аналогів за своєю суттю.	Пропозиція можливостей, які не надають крнкуренти.
6	За інтенсивністю: марочна	Наявність особливого знаку що відрізняє продукт від аналогічних продуктів	Впровадження унікальної назви. Реклама бренду

Далі було проведено аналіз галузевої конкуренції за моделлю М. Портера (табл.5.9).

За результатами аналізу було сформульовано висновок щодо можливість впровадження продукту на ринок з огляду на конкурентну ситуацію. Також було сформульовано висновок щодо властивостей, які

повинен мати проект, щоб витримати конкуренцію на ринку і зайняти конкурентну нішу.

Даний висновок було враховано у процесі формування переліку факторів конкурентоспроможності. На основі дослідження конкурентного середовища, а також із урахуванням властивостей стартап-проекту (табл.5.2), вимог споживачів до товару (табл.5.5) та факторів маркетингового середовища (табл.5.6, табл.5.7) був визначений перелік факторів конкурентоспроможності. Аналіз оформлюється за табл.5.9.

Таблиця 5.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Logaster	Існує вірогідність виходу нових конкурентів	-	Потребують унікальних логотипів з гарною якістю зображення	Частково присутні, можуть з'явитися нові
Висновки	Існує інтенсивна конкурентна боротьба в галузі	Слід враховувати можливість виходу конкурентів на ринок	-	Клієнти диктують умови функціонування на ринку.	Рішення існують, можуть з'явитися нові

Проаналізувавши властивості ринку з огляду на конкурентну ситуацію можна зробити висновок, що оскільки кожний з існуючих продуктів має свій підхід до вирішення задачі, то процес виходу на ринок представляється можливою, але слід враховувати, що рівень конкуренції високий.

Збір інформації про конкурентів проводиться з метою зробити власний проект більш конкурентоспроможним. При цьому є можливість з більшою упевненістю пропонувати свої послуги потенційним споживачам. Ця процедура необхідна тому, що можливі споживачі та партнери підприємства взагалі не знають, чим відрізняються діючі виробники на ринку.

Для успішного виходу на ринок продукт повинен робити акценти на своїй відмінності від конкурентів.

Таблиця 5.10. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Web-application	Дає можливість використання програмного забезпечення будь-де для будь-якого користувача. Можливість кросплатформенного використання
2	Ціна рішення	Можливість генерувати логотипи безкоштовно
3	Унікальність товару	Унікальність логотипів необхідна для клієнтів
4	Різноманітність товару	Різноманітність логотипів є перевагою для клієнтів

На основі значених вище факторів конкурентоспроможності (табл.5.10) було проведено аналіз сильних та слабких сторін проекту (табл.5.11).

Таблиця 5.11. Порівняльний аналіз сильних та слабких сторін

№	Фактори конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим							
			-3	-2	-1	0	+1	+2	+3	
1	Web-application	20	+							
2	Ціна рішення	17				+				
3	Унікальність товару	19				+				
4	Різноманітність товару	19			+					

Заключною частиною аналізу можливостей ринкового впровадження проекту є формулювання SWOT-аналізу (матриці аналізу слабких (Weak) та сильних (Strength) сторін, можливостей (Opportunities) та загроз (Troubles) (табл.5.11) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл.5.12).

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками

(прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 5.12. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> <li>– Кроссплатформенність</li> <li>– Ціна рішення</li> <li>– Унікальність</li> <li>– Різноманітність</li> </ul>	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> <li>– Реклама</li> <li>– Сміслове навантаження логотипів</li> </ul>
<p>Можливості (O):</p> <ul style="list-style-type: none"> <li>– Відсутність схожих технологій у конкурентів</li> </ul>	<p>Загрози (T):</p> <ul style="list-style-type: none"> <li>– Наявність конкурентів з альтернативними рішеннями</li> </ul>

На основі SWOT-аналізу були сформульовані альтернативи ринкової поведінки, тобто перелік заходів, необхідних для виведення проекту на ринок та приблизний оптимальний час для його реалізації з огляду на можливість виведення конкурентами потенційних проектів.

Сформульовані альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (табл. 5.13).

Таблиця 5.13. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розробка алгоритмів управління смисловою частиною зображення	20%	7-8 місяців
2	Відсутність реклами на перших етапах роботи рішення	80%	4-5 місяців
3	Розміщення рекламних постів у соціальних мережах, реклама по всім каналам	90%	2-3 тижні
4	Презентація товару на тематичних конференціях	80%	1-2 місяці

#### 4.4. Розроблення ринкової стратегії проекту

Першим кроком розроблення ринкової стратегії є визначення стратегії охоплення ринку. Для цього було проведено опис цільових груп потенційних споживачів (табл.5.14).

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільового сегменту	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Малі компанії	Висока	Висока	Висока	Середня
2	Середні компанії	Середня	Середня	Висока	Середня
3	Великі компанії	Низька	Низька	Висока	Середня
4	Молоді компанії	Висока	Висока	Висока	Середня
5	Старі компанії	Низька	Низька	Висока	Середня

Згідно з проведеним вище аналізом можна зробити висновок, що цільова група, що підходить найбільше для поширення даного програмного продукту є молоді невеликі компанії. Такі результати отримано тому, що саме такі компанії схильні до використання новітніх технологій а також не мають коштів для закупки логотипів у спеціалізованих професійних компаній. Великі компанії, що давно на ринку схиляються до традиційних способів ведення справ. Далі у табл. 4.15 наведено визначення базової стратегії розвитку.

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Розміщення рекламних постів у соціальних мережах, реклама по всім каналам	Проведення реклами, освітлення унікальності рішення в інтернет мережі. Створення рекламних постів в соціальних мережах та в основних пошукових системах.	Унікальність технології запропонованої в рішенні. Відмінність від аналогічних рішень.	Стратегія диференціації

Наступним кроком є визначення стратегії конкурентної поведінки (табл.5.16).

Таблиця 5.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрхідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки існують як конкурентні рішення, так і товари-замінники	Концентрація на існуючих клієнтів	Рішення не буде копіювати аналогів. Буде обрано власний підхід до рішення проблеми	Стратегія заняття конкурентної ніші

На основі того, які потреби мають споживачі з обраних сегментів щодо постачальника та його продукту (див. табл.5.5), а також в залежності від обраної базової стратегії розвитку (табл.5.15) та стратегії конкурентної поведінки (табл.5.16) розроблено стратегію позиціонування (табл.5.17), що полягає у визначенні ринкової позиції, за якою споживачі будуть відрізняти торгівельну марку/проект від інших.

Таблиця 5.17. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувану комплексну позицію власного проекту
1	Якість зображень	Спеціалізації	Використання передових алгоритмів	Якість, чіткість, осмисленість
2	Унікальність зображень	Спеціалізації	Відсутність подібних рішень	Відмінність, унікальність

Відповідно до приведеної вище інформації та проаналізувавши її можна зробити висновок, що як базова стратегія розвитку та як базова стратегія конкурентної поведінки для стартап-компанії найкраще підходить стратегія спеціалізації.

#### 4.5. Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач.

Для цього у табл.5.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Висока якість логотипів	Якість забезпечується передовими алгоритмами нейронних мереж	Потрібно забезпечити якість зображень
2	Унікальність логотипів	Унікальність забезпечується алгоритмами нейронних мереж	Наявність аналогічних зображень неможлива

Розроблено трирівневу маркетингову модель товару: уточняється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл.5.19).

Таблиця 5.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	WEB-додаток для генерації логотипів.		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Зручність та простота у користуванні	Нм	Ергономічні
	Унікальність зображень	Нм	Технологічні
	Кросплатформеність	Нм	Технологічні
	Якість: буде проведено тестування		
	Система; веб-сайт, що буде надавати доступ для системи.		
Назва товару: NeuraLogo			
3. Товар із підкріпленням	До продажу: 1 місяць без реклами		
	Після продажу: постійне допрацювання алгоритмів		
За рахунок чого потенційний товар буде захищено від копіювання: унікальність технологій.			

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари аналоги або товари субститути (табл. 5.20).

Таблиця 5.20. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
0 грн	0 – 60000 / рік	10000000грн / рік	60000 / рік

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 5.21):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 5.21. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Вибір послуги на веб-сайті	Web-сайт	Виробник - клієнт	Web-сайт

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 5.22).

Таблиця 5.22. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Клієнти обиратимуть зручніший товар з потрібними функціями.	Веб-сайт	Унікальність	Демонстрація переваг для клієнтів.	Аргументація переваги над конкурентами, демонстрація результатів роботи.

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

#### Висновки до розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проекту було встановлено що проект є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої імплементації проекту.

## ВИСНОВОК

В ході розробки магістерської дисертації досліджено генеративні можливості інтелектуальних робототехнічних систем. В ході роботи над дисертацією стало зрозуміло що фокус уваги буде падати на генерацію новизни в рамках нейронних мереж. Проаналізувавши літературу можна сказати, що останнім часом досягнуто певних успіхів у генеративному моделюванні. Однак більшість робіт стосується того, як навчити мережу створювати відомі об'єкти, а література здебільшого стурбована генеруванням реалістичних об'єктів а не вивченням та пропонуванням нових завдань.

В роботі проведено аналіз сучасного стану проблеми. Тобто визначено мету та задачі дослідження. В ході аналізу виявлено проблеми предметної області а також її особливості. Проведено аналіз існуючих підходів до оцінки генеративних моделей, підходів до рішення задачі генерації та аналіз існуючих рішень.

На основі отриманих результатів проведено аналіз інформаційного забезпечення. У даному розділі роботи описано обрану генеративну мережу GAN, її архітектуру та компоненти. Також проведено вибір середовища та засобів розробки, описано причини вибору саме таких інструментів. Здійснено проектування моделі та показано її архітектуру.

Наступним кроком стала розробка алгоритмічного та програмного забезпечення. У цьому розділі описано тренувальні данні для даної моделі, описаний процес та особливості навчання моделі

Як результат опису моделі з інформаційної та алгоритмічно-програмної сторони став розділ експерименту. У даному розділі використано описану в попередніх розділах мережу для випробування методів оцінки генеративних моделей. В ході дослідження проведено експерименти з трьома різними налаштуваннями мереж для їх подальшого порівняння. Результати експериментів були досить вдалими. Експеримент показав, що можна використовувати не тільки людську суб'єктивну оцінку, але й чисельну розроблену оцінку.

Останнім розділом роботи є розробка стартап-проекту. В даному розділі обґрунтовано актуальність створення системи, веб-додатку для генерації логотипів компаній та брендів. У розділі розробки стартапу був здійснений опис ідеї, аналіз умов ринку та стратегії впровадження проекту в таких умовах. Була розроблено організаційний план проекту, в якому проведений аналіз обсягу робіт, витрат та графіку реалізації цих ресурсів. В завершення проведено фінансово-економічний аналіз та оцінка ризиків проекту.

Загальним висновком, який можна зробити проаналізувавши всі розділи роботи є те, що генеративні можливості інтелектуальних робототехнічних систем дійсно є актуальним полем для дослідження. Ця область нейронних мереж стає популярнішою щороку. Кількість робіт пов'язаних з цією темою зростає. Розвиток досліджень рухають як створення нових архітектур мереж, так і створення нових методів оцінки, один з яких використано в роботі.

## ПЕРЕЛІК ДЖЕРЕЛ

1. Tim Salimans. Improved techniques for training GANs / Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen. arXiv preprint // arXiv:1606.03498, 2016.
2. Pascal Vincent. Extracting and composing robust features with denoising autoencoders / Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol // In Proceedings of the 25th international conference on Machine learning, pp. 1096–1103. ACM, 2008.
3. Armand Hatchuel. Ck design theory: an advanced formulation / Benoit Weil // Research in engineering design, 19(4):181–192, 2009
4. Akın Kazakçı. Conceptive artificial intelligence // Insights from design theory. In International Design Conference DESIGN 2014, pp. 1–16, 2014.
5. Geoffrey E Hinton. A fast learning algorithm for deep belief nets / Geoffrey E Hinton, Simon Osindero, Yee-Whye The // Neural computation, 18(7):1527–1554, 2006.
6. Diederik P Kingma. Auto-encoding variational bayes / Diederik P Kingma, Max Welling // arXiv preprint arXiv:1312.6114, 2013.
7. Antti Rasmus. Semi-supervised learning with ladder networks. In Advances in Neural Information Processing Systems / Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, Tapani Raiko // pp. 3532–3540, 2015.
8. Lars Maaløe. Auxiliary deep generative models / Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, Ole Winther // arXiv preprint arXiv:1602.05473, 2016.
9. William T Freeman. Example-based superresolution. IEEE Computer graphics and Applications / William T Freeman, Thouis R Jones, and Egon C Pasztor // 22(2):56–65, 2002.
10. Chao Dong. Image super-resolution using deep convolutional networks / Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang // arXiv:1501.00092, 2014.
11. Christian Ledig. Photo-realistic single image super-resolution using a generative adversarial network / Christian Ledig, Lucas Theis, Ferenc Huszár, Jose

- Caballero, Andrew Aitken, Alykhan Tejani, Johannes Tutz, Zehan Wang, and Wenzhe Shi // arXiv preprint arXiv:1609.04802, 2016.
12. Olivier Breuleux. Unlearning for better mixing / Olivier Breuleux, Yoshua Bengio, Pascal Vincent. Unlearning for better mixing // Universite de Montreal/DIRO, 2009.
  13. Lucas Theis. Generative image modeling using spatial LSTMs. In Advances in Neural Information Processing Systems / Lucas Theis, Matthias Bethge // pp. 1918–1926, 2015.
  14. Jason Yosinski. Understanding neural networks through deep visualization / Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson // arXiv:1506.06579, 2015.
  15. Emily L Denton, Soumith Chintala, Rob Fergus, et al., pp. 1486–1494, 2015.
  16. Alec Radford. Unsupervised representation learning with deep convolutional generative adversarial networks / Alec Radford, Luke Metz, and Soumith Chintala // arXiv:1511.06434, 2015.
  17. Alexey Dosovitskiy. Generating images with perceptual similarity metrics based on deep networks. In Advances in Neural Information Processing Systems / Alexey Dosovitskiy and Thomas Brox // pp. 658–666, 2016.
  18. Brenden M Lake. Humanlevel concept learning through probabilistic program induction. Science / Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum // 350 (6266):1332–1338, 2015.
  19. Xiang Zhang. Character-level convolutional networks for text classification. In Advances in neural information processing systems / Xiang Zhang, Junbo Zhao, and Yann LeCun // pp. 649–657, 2015.
  20. Alex Krizhevsky. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems / Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton // pp. 1097–1105, 2012.
  21. Ryan Dahl. Pixel recursive super resolution / Ryan Dahl, Mohammad Norouzi, and Jonathon Shlens // arXiv:1702.00783, 2017.

22. Ronan Collobert. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* / Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa // 12:2493–2537, 2011.
23. David E Rumelhart. Learning representations by back-propagating errors / David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams // 323:9, 1986.
24. Ian Goodfellow. Generative adversarial nets. In *Advances in Neural Information Processing Systems* / Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio // pp. 2672–2680, 2014.
25. Mehdi Mirza. Conditional generative adversarial nets / Mehdi Mirza and Simon Osindero // arXiv:1411.1784, 2014.
26. Jianxiong Xiao. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR)* / Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba // pp. 3485– 3492. IEEE, 2010.
27. Jia Deng. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition* / Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei // 2009. CVPR 2009. IEEE Conference on, pp. 248–255. IEEE, 2009.
28. Alexey Dosovitskiy. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems* / Alexey Dosovitskiy and Thomas Brox // pp. 658–666, 2016.
29. Raymond Yeh. Semantic image inpainting with perceptual and contextual losses / Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do // arXiv:1607.07539, 2016.
30. Razvan Pascanu. Understanding the exploding gradient problem / Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio // CoRR, abs/1211.5063, 2012.
31. Scott E Reed. Learning what and where to draw. In *Advances In Neural Information Processing Systems* / Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee // pp. 217–225, 2016.

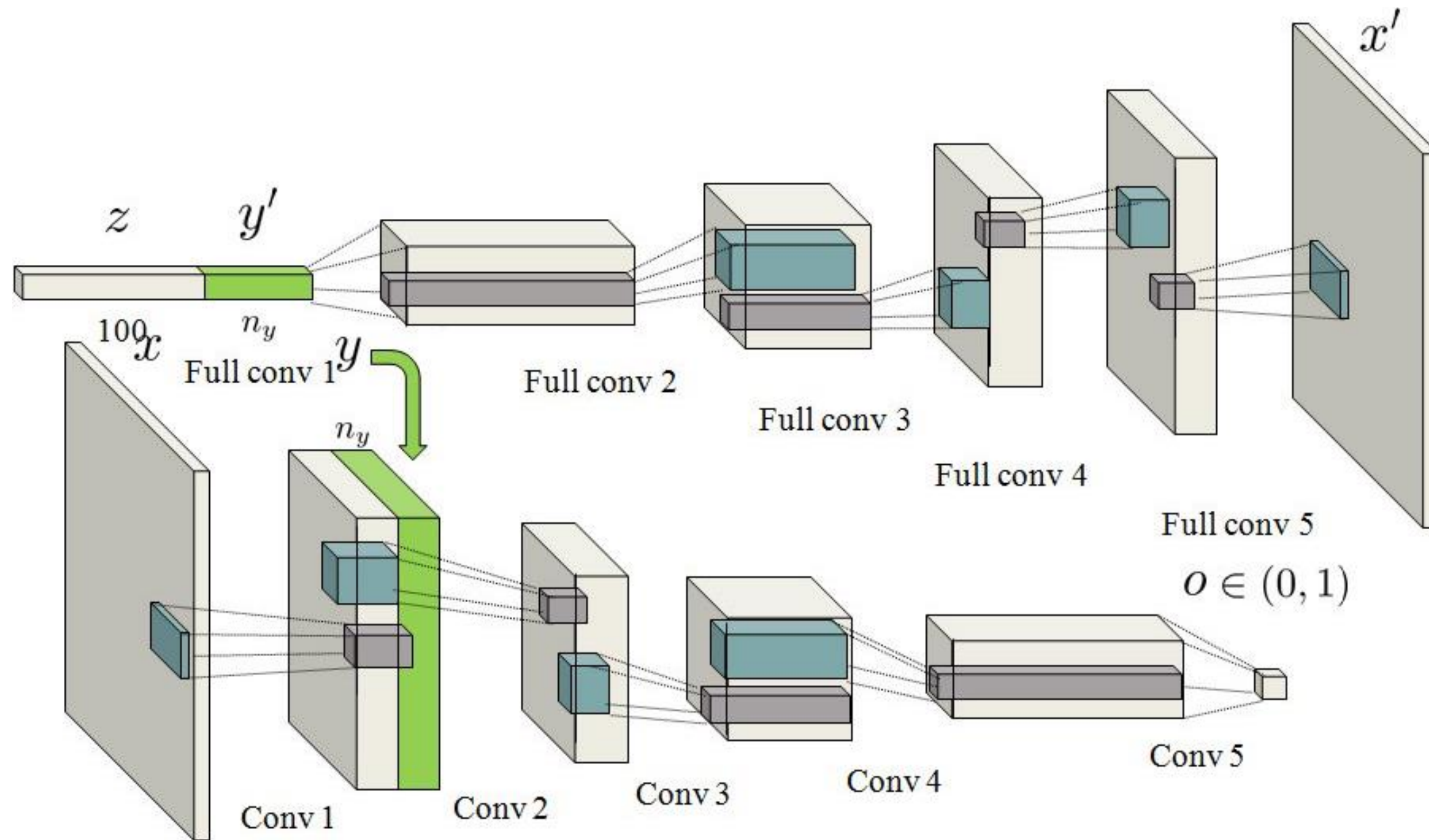
32. Phillip Isola. Image-to-image translation with conditional adversarial networks / Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros // arXiv:1611.07004, 2016.
33. Yanghua Jin. Towards the automatic anime characters creation with generative adversarial networks / Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang // arXiv preprint arXiv:1708.05509, 2017.
34. Róisín Loughran. Application domains considered in computational creativity / Róisín Loughran and Michael O'Neill // 2017.
35. Julian Togelius. Search-based procedural content generation: A taxonomy and survey / Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne // IEEE Transactions on Computational Intelligence and AI in Games, 3 (3):172–186, 2011.
36. Alec Radford. Unsupervised representation learning with deep convolutional generative adversarial networks / Alec Radford, Luke Metz, and Soumith Chintala // arXiv:1511.06434, 2015.
37. Python [Электронный ресурс] // Wikipedia. – 2019. – Режим доступа до ресурсу: <https://ru.wikipedia.org/wiki/Python>
38. Tensorflow [Электронный ресурс] // Wikipedia. – 2019. – Режим доступа до ресурсу: <https://www.tensorflow.org/about/>
39. PyCharm [Электронный ресурс] // Wikipedia. – 2019. – Режим доступа до ресурсу: <https://www.jetbrains.com/ru-ru/pycharm/features/>
40. ReLU [Электронный ресурс] // Wikipedia. – 2019. – Режим доступа до ресурсу: <https://uk.wikipedia.org/wiki/ReLU>
41. LLD [Электронный ресурс] // Wikipedia. – 2019. – Режим доступа до ресурсу: <https://data.vision.ethz/sagea/lll/>
42. ImageNet [Электронный ресурс] // Wikipedia. – 2019. – Режим доступа до ресурсу: <http://image-net.org/about-overview>
43. Stochastic gradient descent [Электронный ресурс] // Wikipedia. – 2018. – Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Stochastic\\_gradient\\_descent](https://en.wikipedia.org/wiki/Stochastic_gradient_descent)

## ДОДАТКИ

ДОДАТОК А  
ІЛЮСТРАЦІЙНІ МАТЕРІАЛИ

ДОДАТОК Б  
ПЕРЕВІРКА НА СПВПАДІННЯ

# Архітектура DCGAN

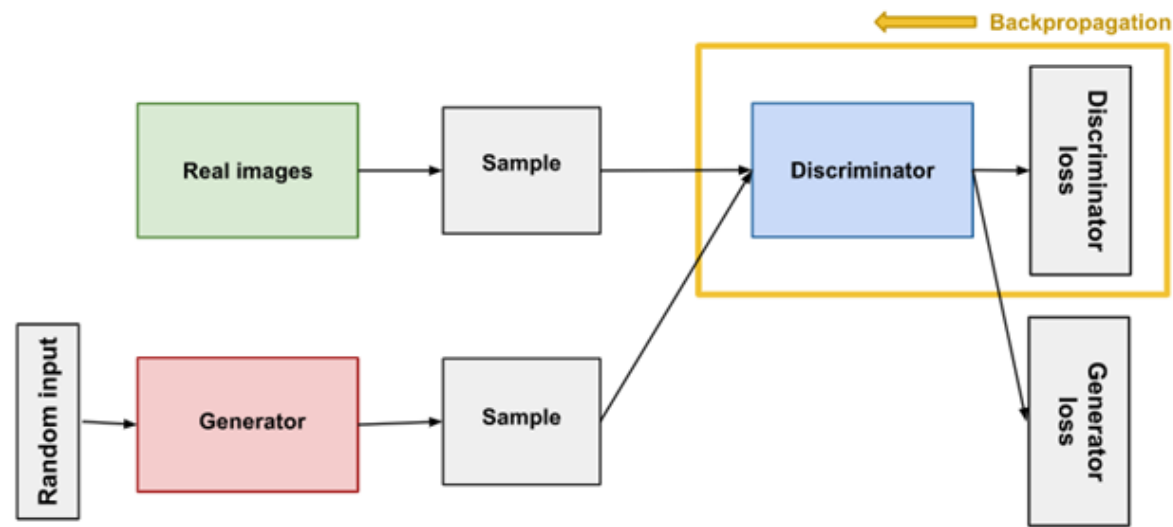


Демонстраційний плакат №6  
до дипломної роботи на тему  
«Дослідження генеративних можливостей інтелектуальних  
робототехнічних систем»

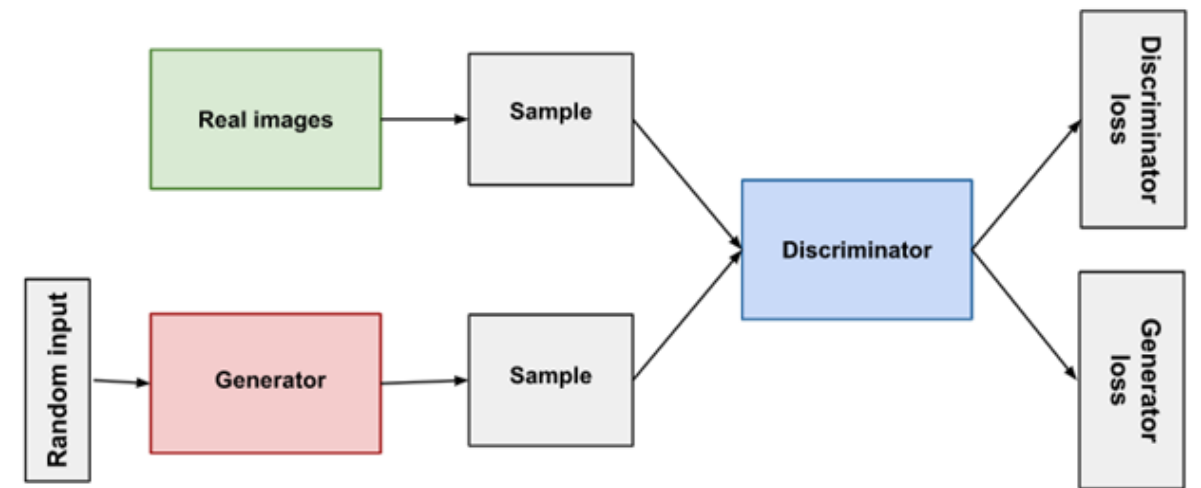
Виконав: студент гр. ІК-82мп Гарькавий С.В.  
Керівник: к.т.н., доцент Олійник В.В.

# Архітектура DCGAN

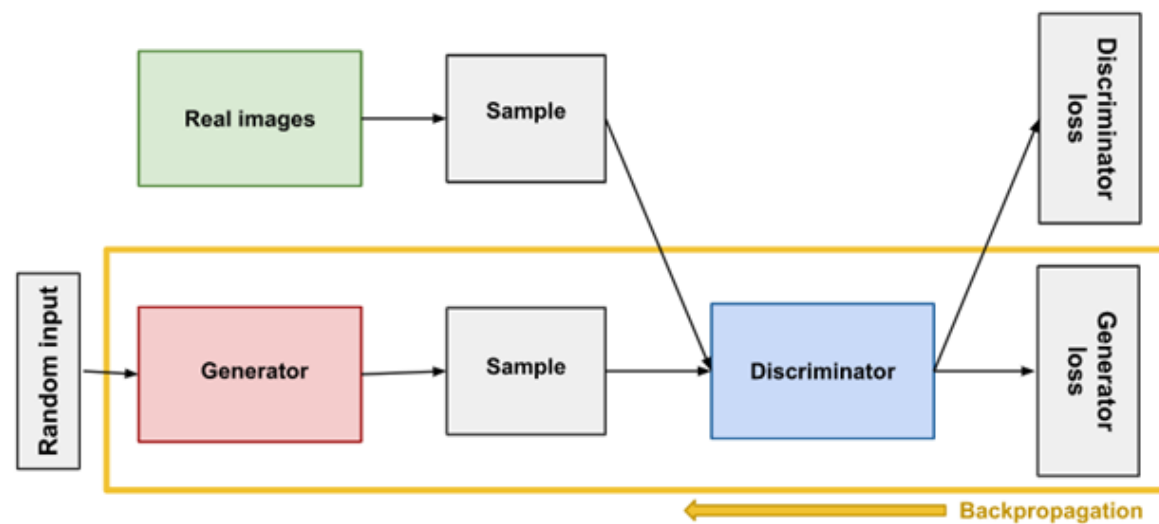
## Дискримінатор GAN



## GAN



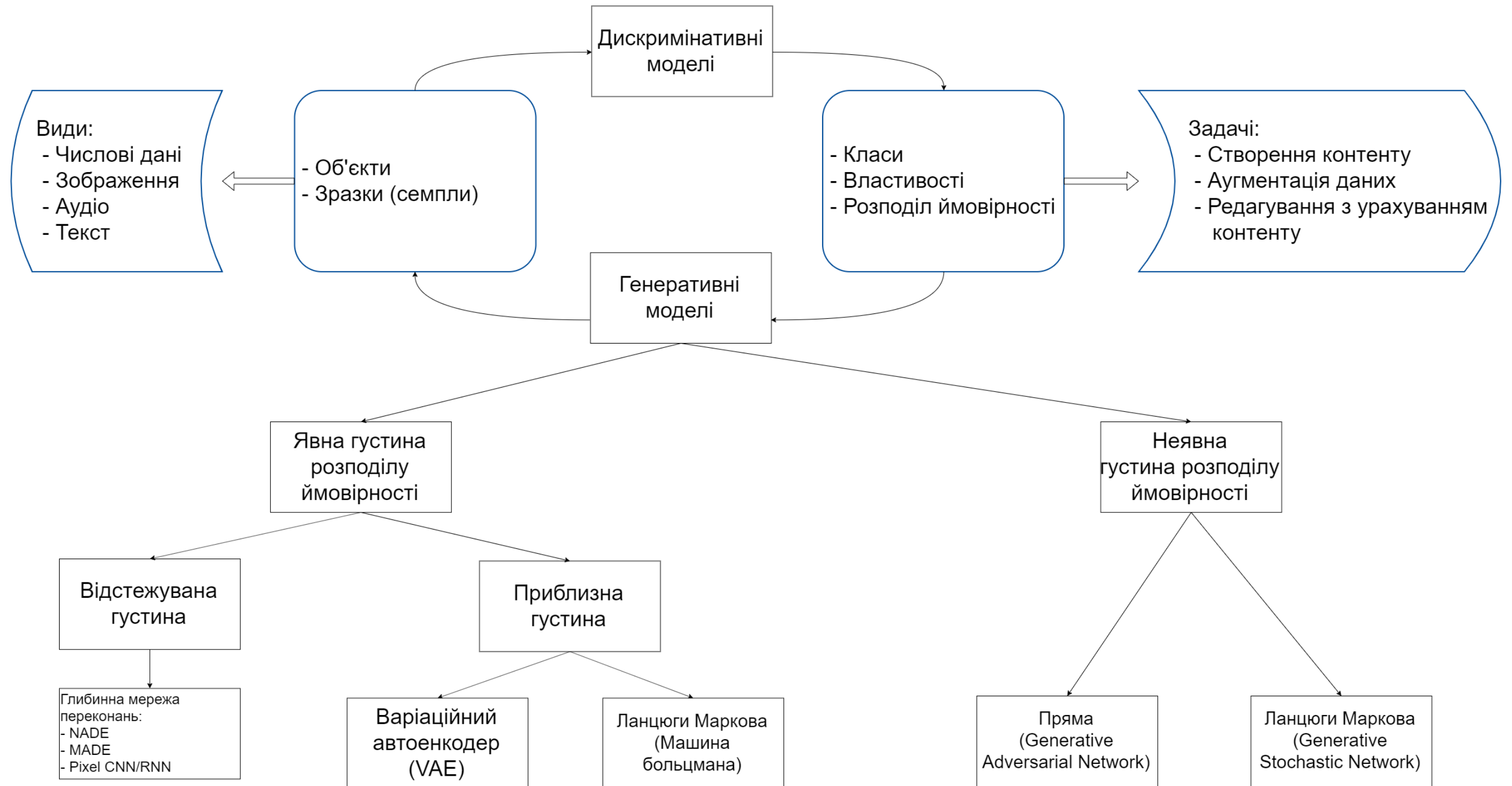
## Генератор GAN



Демонстраційний плакат №5  
до дипломної роботи на тему  
«Дослідження генеративних можливостей інтелектуальних  
робототехнічних систем»

Виконав: студент гр. ІК-82мп Гарькавий С.В.  
Керівник: к.т.н., доцент Олійник В.В.

# ІРС

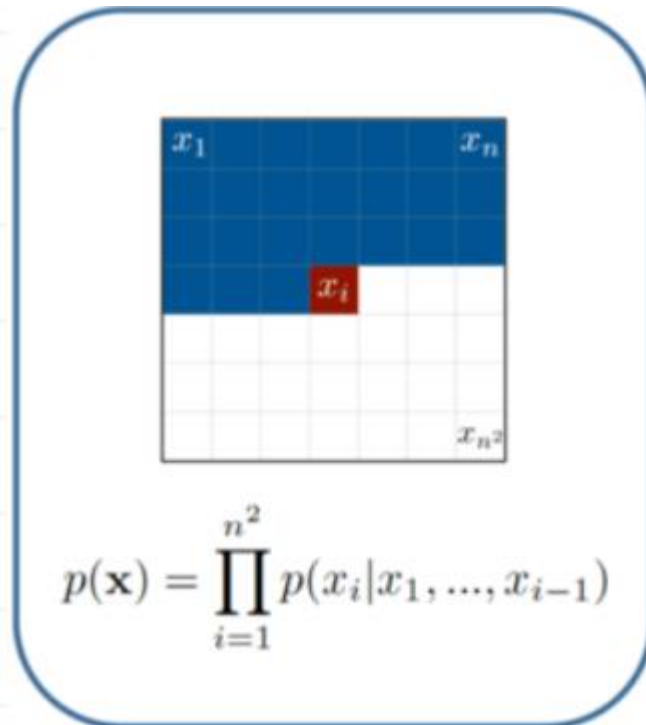


Демонстраційний плакат №1  
до дипломної роботи на тему  
«Дослідження генеративних можливостей інтелектуальних  
робототехнічних систем»

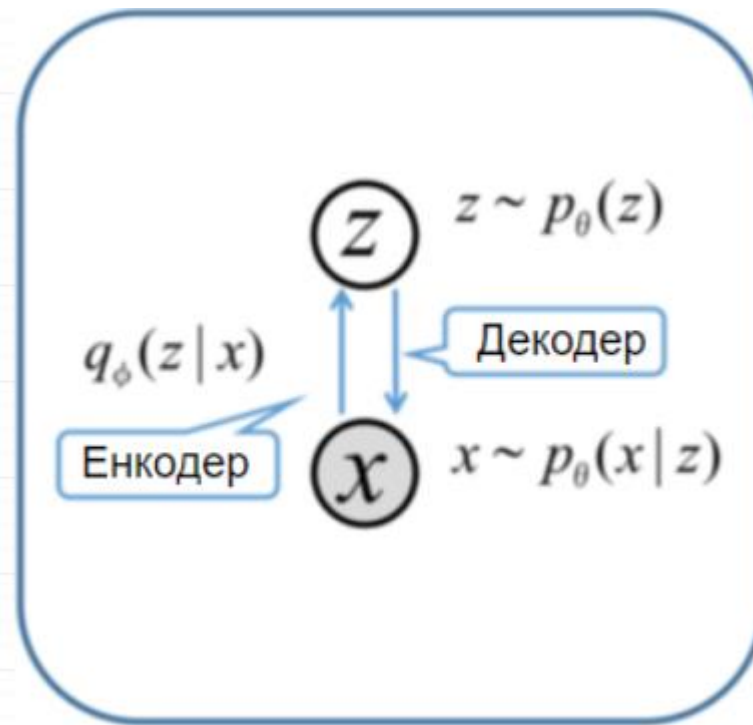
Виконав: студент гр. ІК-82мп Гарькавий С.В.  
Керівник: к.т.н., доцент Олійник В.В.

# Генеративні моделі

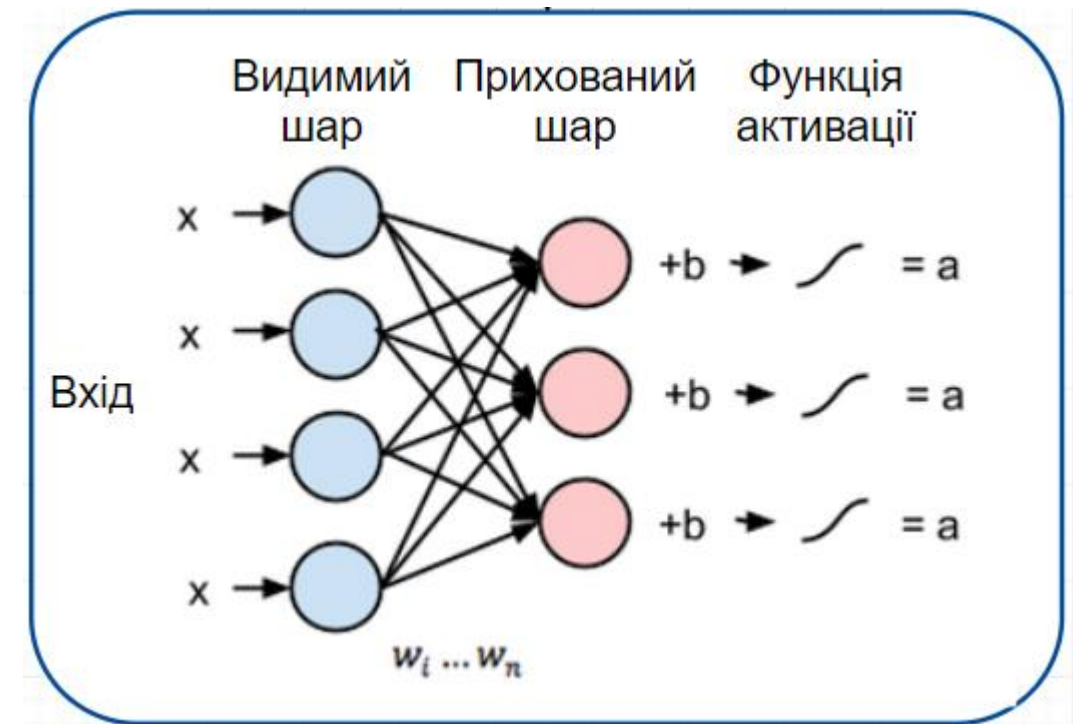
RNN



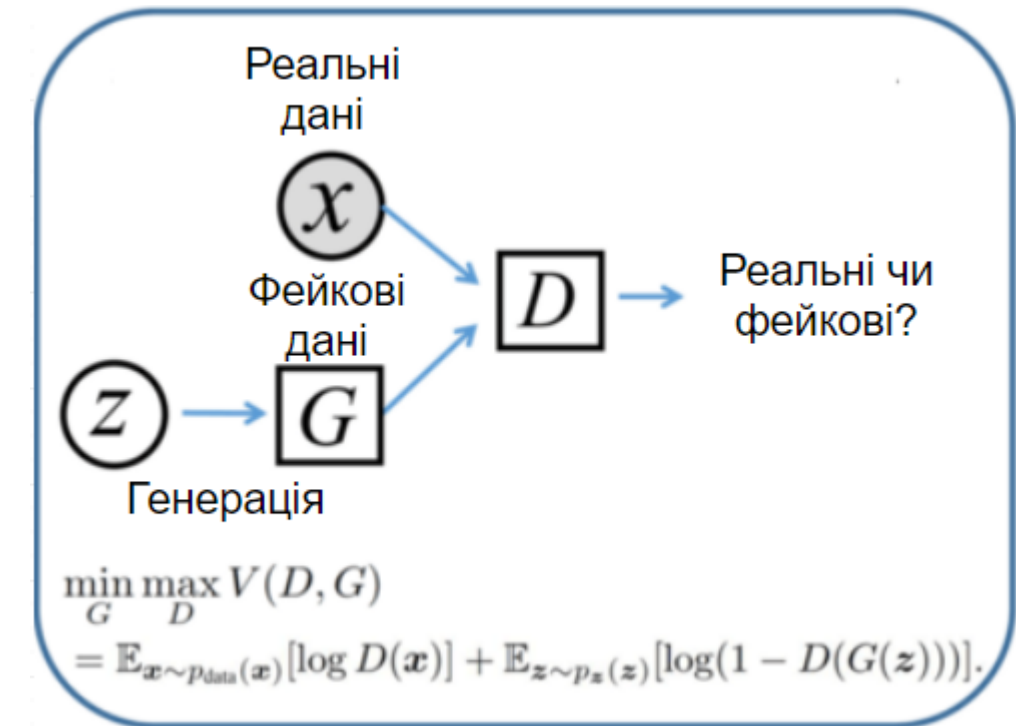
VAE



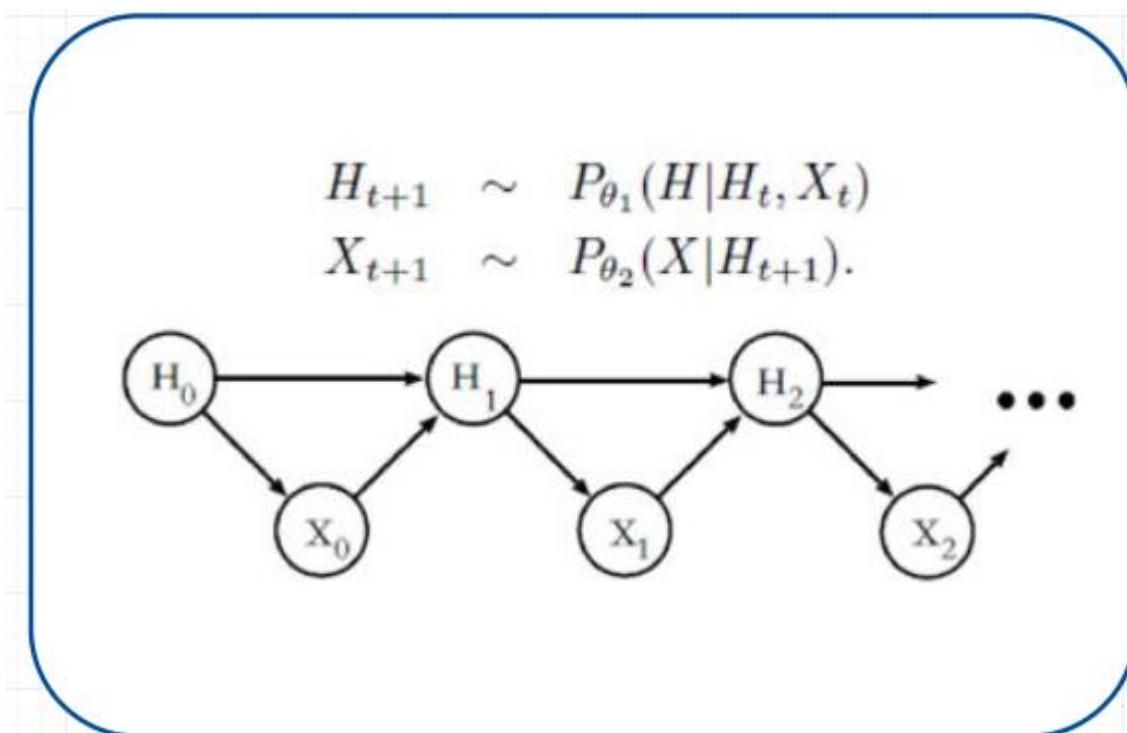
Boltzmann machine



GAN



GSN



Демонстраційний плакат №2  
до дипломної роботи на тему  
«Дослідження генеративних можливостей інтелектуальних  
робототехнічних систем»

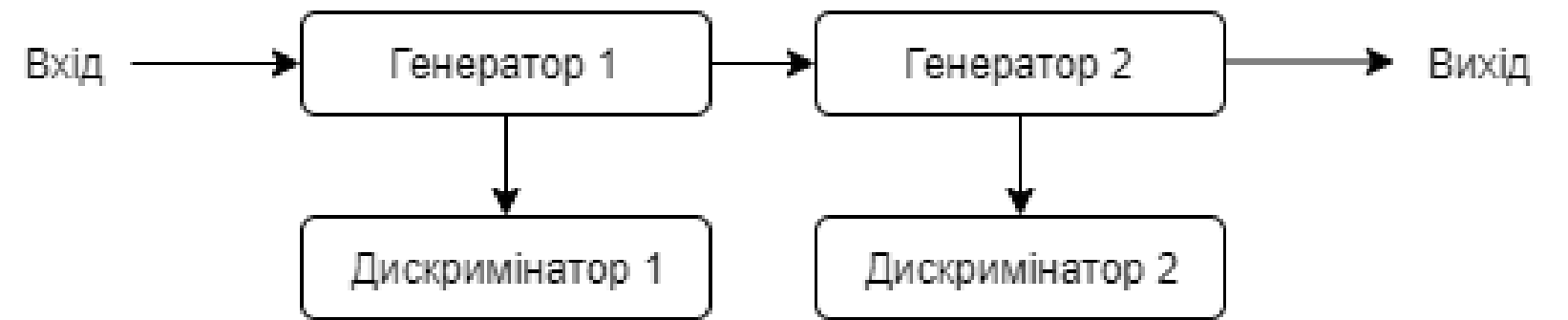
Виконав: студент гр. ІК-82мп Гарькавий С.В.  
Керівник: к.т.н., доцент Олійник В.В.

# Методи генерації даних

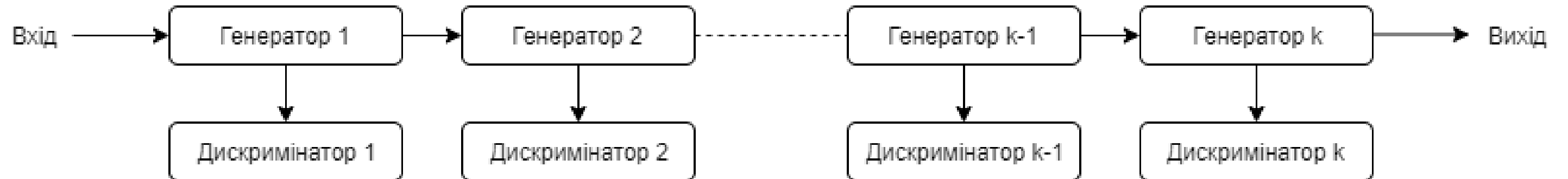
## Прямий метод



## Ієрархічний метод



## Ітераційний метод



Демонстраційний плакат №4  
до дипломної роботи на тему  
«Дослідження генеративних можливостей інтелектуальних  
робототехнічних систем»

Виконав: студент гр. ІК-82мп Гарькавий С.В.  
Керівник: к.т.н., доцент Олійник В.В.

# Результати роботи системи

Результати першого експерименту



Результати другого експерименту



Результати третього експерименту



Результати роботи мережі



Демонстраційний плакат №3  
до дипломної роботи на тему  
«Дослідження генеративних можливостей інтелектуальних  
робототехнічних систем»

Виконав: студент гр. ІК-82мп Гарькавий С.В.  
Керівник: к.т.н., доцент Олійник В.В.