

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

**Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту**

До захисту допущено:

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«___» _____ 20__ р.

**Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системи і методи штучного інтелекту»
спеціальності 122 «Комп'ютерні науки»
на тему: «Інтелектуальна система визначення стилів картин»**

Виконала:

студентка IV курсу, групи КІ-21
Левцун Анастасія Костянтинівна

Керівник:

професорка кафедри штучного інтелекту, д. т. н., професор,
Чумаченко Олена Іллівна

Консультант з нормоконтролю:

фахівчиня I категорії кафедри штучного інтелекту,
к.т.н., доцент Комариста Богдана Миколаївна

Рецензент:

старший викладач кафедри системного проектування,
доктор філософії, Яременко Вадим Сергійович

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студентка _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«15» січня 2026 р.

ЗАВДАННЯ

на дипломну роботу студенту

Левцун Анастасії Костянтинівні

1. Тема роботи «Інтелектуальна система визначення стилів картин», керівник роботи Чумаченко Олена Іллівна, професор кафедри ШІ, д. т. н., затверджені наказом по НН ІПСА від «27» травня 2026 р. № 1944-с.
2. Термін подання студентом роботи «05» червня 2026 року.
3. Вихідні дані до роботи – набір даних для задачі класифікації.
4. Зміст роботи: аналіз задачі класифікації художніх стилів, огляд математичних основ роботи, розробка запропонованого підходу, опис програмної реалізації системи та демонстрація результатів.
5. Перелік ілюстративного матеріалу: електронна презентація.
6. Дата видачі завдання «02» лютого 2026 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Вивчення літератури за темою	20.04.2026	Виконано
2	Підготовка першого розділу	27.04.2026	Виконано
3	Підготовка другого розділу	04.05.2026	Виконано
4	Підготовка третього розділу	11.05.2026	Виконано
5	Розробка програмного продукту	18.05.2026	Виконано
6	Підготовка четвертого розділу	25.05.2026	Виконано
7	Оформлення дипломної роботи	01.06.2026	Виконано
8	Підготовка презентації доповіді	03.06.2026	Виконано

Студент

Анастасія ЛЕВЦУН

Керівник

Олена ЧУМАЧЕНКО

РЕФЕРАТ

Дипломна робота: 88 с., 17 рис., 9 табл., 37 посилань, додаток.

КЛАСИФІКАЦІЯ ХУДОЖНІХ КАРТИН, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ГЕНЕРАТИВНО-ЗМАГАЛЬНІ МЕРЕЖІ, БАЛАНСУВАННЯ ДАНИХ.

Об'єктом дослідження є процес автоматизованого розпізнавання художніх стилів картин на основі методів глибокого машинного навчання.

Предметом дослідження є архітектура та методи навчання інтелектуальної системи, що поєднує генеративну модель WGAN-GP для балансування датасету та згорткову нейронну мережу для класифікації.

Метою роботи є розробка та аналіз ефективності інтелектуальної системи визначення художніх стилів картин для забезпечення коректної класифікації в умовах нерівномірного розподілу навчальних даних.

Для досягнення мети використано методи глибокого та трансферного навчання, згорткову нейронну мережу ResNet-18 та генеративну модель WGAN-GP. У результаті дослідження створено адаптивну стратегію балансування набору даних WikiArt шляхом синтезу зображень, розроблено оригінальний алгоритм запобігання mode collapse та підвищено загальну точність класифікації з 61% до 72%. Також впроваджено імовірнісний підхід для виведення трьох найімовірніших стилів і верифіковано логіку моделі через теплові карти. Розроблену систему рекомендується використовувати для автоматизованої каталогізації музейних фондів, інтеграції в пошукові механізми мистецьких цифрових архівів, а також як основу для освітніх та рекомендаційних платформ.

ABSTRACT

Master's thesis: 88 p., 17 figures, 9 tables, 37 references, appendix.

PAINTING CLASSIFICATION, CONVOLUTIONAL NEURAL NETWORKS, GENERATIVE ADVERSARIAL NETWORKS, DATA BALANCING.

The object of the study is the process of automated recognition of artistic styles of paintings based on deep machine learning methods.

The subject of the study is the architecture and training methods of an intelligent system that combines the generative WGAN-GP model for dataset balancing and a convolutional neural network for classification.

The purpose of the work is to develop and evaluate the effectiveness of an intelligent system for painting style recognition, ensuring accurate classification under the conditions of imbalanced training data.

To achieve this, deep and transfer learning techniques were applied, utilizing the ResNet-18 convolutional neural network and the WGAN-GP generative model. As a result of the research, an adaptive balancing strategy for the WikiArt dataset was established via image synthesis, a novel algorithm to prevent mode collapse was developed, and the overall classification accuracy was improved from 61% to 72%. Furthermore, a probabilistic approach was implemented to output the top three most likely styles, and the model's decision-making logic was verified using heatmaps. The developed system is recommended for the automated cataloging of museum collections, integration into the search engines of digital art archives, and as a foundation for educational and recommendation platforms.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ТЕОРЕТИЧНІ ОСОБЛИВОСТІ ДОСЛІДЖЕННЯ.....	10
1.1 Задача класифікації художніх стилів та її особливості	10
1.2 Обґрунтування використання генеративно-змагальних мереж.....	12
1.2.1 Базова структура GAN	13
1.2.2 Класифікація GAN	15
1.3 Проблема mode collapse та причини її виникнення.....	17
Висновки до розділу 1	22
РОЗДІЛ 2 ТЕОРЕТИЧНІ ТА МАТЕМАТИЧНІ ОСНОВИ ОБРАНИХ МОДЕЛЕЙ.....	24
2.1 Алгоритми запобігання mode collapse.....	24
2.1.1 Архітектура WGAN-GP	25
2.1.2 Регуляризаційні підходи та сучасні алгоритми	26
2.2 Згорткові нейронні мережі.....	28
2.3 Огляд сучасних досліджень у сфері класифікації художніх картин	35
2.4 Постановка задачі.....	40
Висновки до розділу 2	41
РОЗДІЛ 3 ЗАПРОПОНОВАНИЙ ПІДХІД ДО ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	42
3.1 Опис навчальної вибірки	42
3.2 Спеціалізований алгоритм запобігання mode collapse	46
3.2.1 Двокритеріальна оцінка ризику	47
3.2.2 Конфігурація захисних методів	51
3.4 Обґрунтування вибору ResNet-18	53
Висновки до розділу 3	58

РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ.....	59
4.1 Вибір інструментальних засобів розробки	59
4.2 Архітектура програмного забезпечення	60
4.3 Метрики оцінки якості класифікатора	65
4.5 Результати роботи запропонованого підходу	67
4.5 Можливості вдосконалення моделі та подальші експерименти.....	74
Висновки до розділу 4	76
ВИСНОВКИ	77
ПЕРЕЛІК ПОСИЛАНЬ	78
ДОДАТОК А ОСНОВНІ СКЛАДНИКИ ЛІСТИНГУ ПРОГРАМИ.....	83

ВСТУП

Класифікація¹ зображень є однією з фундаментальних задач комп'ютерного зору, для якої протягом десятиліть було розроблено широкий спектр методів. Однак більшість підходів традиційно зосереджувалися на аналізі натуральних зображень – об'єктів реального світу, таких як люди, тварини чи транспорт. На відміну від них, художні твори мають іншу природу: вони є результатом творчого процесу, у якому поєднуються індивідуальний стиль, техніка й світогляд художника або цілої мистецької течії. Саме тому класифікація художніх зображень має унікальні складнощі та потребує спеціалізованих підходів.

Сучасний розвиток технологій оцифрування спричинив появу величезних онлайн-бібліотек художніх творів. Музеї й галереї активно інтегрують свої фонди у цифрові колекції, що робить мистецькі ресурси широко доступними. Це супроводжується потребою у створенні інтелектуальних систем, які забезпечуватимуть ефективний пошук, архівування, індексацію й аналіз візуального контенту.

Створення інтелектуальної системи визначення стилів картин є актуальним завданням сучасної комп'ютерної візуалізації. Воно має важливе значення не лише для технічної галузі, а й для культурної та історичної сфер, сприяючи розвитку технологій обробки художнього контенту, збереженню культурної спадщини й удосконаленню сучасних мультимедійних систем.

Об'єкт роботи – це інтелектуальна система автоматизованого розпізнавання та класифікації візуального мистецтва.

¹Тут і нижче використано такий інструмент штучного інтелекту як чат-бот з генеративним штучним інтелектом Gemini виключно для корегування та редагування тексту, створеного автором цієї дипломної роботи, на основі автоматизованої перевірки граматики, структури та стилю, що відповідає Політиці використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського (протокол №11 Вченої ради КПІ ім. Ігоря Сікорського від 11 грудня 2023 р.).

Предмет роботи – це метод глибокого машинного навчання для визначення художніх стилів на базі архітектури ResNet-18 та модель генеративної мережі для балансування вибірки цільового набору даних.

Мета роботи – розробити та проаналізувати ефективність інтелектуальної системи, здатної автоматично визначати художній стиль картин, а також вирішення проблеми незбалансованості мистецьких наборів даних за допомогою методів генерації синтетичних зображень.

Виходячи з мети роботи, було поставлено такі завдання:

- проаналізувати існуючі методи класифікації та обґрунтувати доцільність використання запропонованої моделі;
- застосувати генеративну модель для синтезу нових зображень з метою усунення класового дисбалансу;
- дослідити проблему виникнення mode collapse та знайти шляхи її подолання;
- провести навчання моделі на збалансованій вибірці та здійснити класифікацію зображень;
- оцінити точність класифікації.

Удосконалена розроблена система може бути використана цифровими архівами та музеями для автоматичної каталогізації колекцій, інтегрована в пошукові механізми мистецьких платформ для пошуку за візуальною схожістю.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ТЕОРЕТИЧНІ ОСОБЛИВОСТІ ДОСЛІДЖЕННЯ

1.1 Задача класифікації художніх стилів та її особливості

У межах аналізу мистецтва важливою задачею є визначення художнього стилю. Стиль – це комплекс характеристик, таких як вибір колористики, композиції, освітлення, типові мазки пензля, які формують своєрідний «візуальний підпис» твору. Він дозволяє віднести роботу до певного мистецького руху – імпресіонізму, кубізму, бароко, реалізму тощо. Попри те, що для досвідчених мистецтвознавців деякі стилі можуть бути впізнаваними, їх межі далеко не завжди чітко визначені, а різні художники можуть одночасно працювати у кількох стилях. Це зумовлює складність автоматизованого розмежування та створює значний науковий інтерес до задачі класифікації художніх стилів.

Автоматична класифікація художнього стилю живопису залишається однією з найбільш нетривіальних та науково цікавих задач у сучасній інформатиці. Традиційно ідентифікація художнього стилю, визначення авторства чи приналежності до певної епохи вимагала глибокої експертної оцінки фахівцями-мистецтвознавцями. Проте цей класичний підхід має суттєві обмеження. Він є повільним, суб'єктивним, потребує значних ресурсів і стає абсолютно немасштабованим, коли йдеться про необхідність обробки мільйонів оцифрованих творів мистецтва у сучасних базах даних.

З огляду на ці обмеження, розпізнавання художніх стилів на основі цифрових зображень стало одним із найперспективніших напрямків досліджень у галузі комп'ютерного зору. На відміну від стандартних завдань, таких як ідентифікація облич, транспортних засобів чи повсякденних об'єктів (де класифікація надійно спирається на чіткі фізичні контури та структурні властивості), художній стиль є значно складнішою категорією. Картини

мають багатшарову структуру, яка вимагає від алгоритмів здатності аналізувати не лише об'єкти на полотні, а й саму манеру їх зображення.

Процес комп'ютерного аналізу та класифікації живопису стикається з низкою специфічних перешкод.

1. Відсутність чітких меж та візуальне перекривання. Еволюція мистецьких напрямів відбувається поступово і не має суворих часових рамок. Картини різних епох можуть містити багато спільних рис, а деякі стилі, як-от романтизм і символізм, мають значну візуальну схожість. Це призводить до того, що алгоритми класифікації часто плутають їх через накладання характерних ознак.
2. Художній стиль не визначається якоюсь однією ознакою. Це складна, багаторівнева система, що формується із взаємодії багатьох формальних елементів: напрямку та текстури мазків пензля, специфіки палітри кольорів, геометричної форми, маси, загальної композиції та емоційного забарвлення.
3. Нестандартність та відхилення від канонів. Мистецтво базується на креативності, яка нерідко проявляється у свідомому порушенні правил стилю. Автори можуть змішувати різні техніки або експериментувати, створюючи унікальні візуальні патерни, які вкрай важко піддаються математичній формалізації та алгоритмічному опису.

Саме ця багатогранність, неоднозначність та наявність прихованих стильових ознак роблять класифікацію художніх картини надзвичайно складним завданням. Вона залишається викликом не лише для досвідчених мистецтвознавців, але й для найсучасніших неймереж. Водночас ця проблема стимулює розробку і вдосконалення принципово нових методів комп'ютерного зору, які здатні глибше розуміти контекст та розпізнавати найтонші нюанси художніх манер.

Для вирішення задачі класифікації художніх стилів дослідники застосовують згорткові нейронні мережі, трансформерні архітектури та генеративні моделі, кожен з яких має власні переваги та обмеження, які детально розглянуто далі в роботі.

1.2 Обґрунтування використання генеративно-змагальних мереж

Актуальність застосування GAN у задачах аналізу візуального мистецтва, зумовлена проблемою критичного дисбалансу навчальних даних. Доступні цифрові архіви характеризуються значною нерівномірністю: популярні напрямки, такі як імпресіонізм чи реалізм, представлені тисячами творів, тоді як авангардні або вузькоспеціалізовані стилі, можуть налічувати лише обмежену кількість якісних цифрових копій. Навчання класифікатора на таких незбалансованих вибірках неминуче призводить до перенавчання на домінуючих класах та втрати здатності системи розпізнавати унікальні ознаки класів меншості.

Традиційні методи геометричної аугментації (англ. data augmentation) – повороти, зсуви, віддзеркалення не вирішують цю проблему повністю, оскільки вони лише дублюють існуючу інформацію, не створюючи нових візуальних концептів. Натомість генеративно-змагальні мережі здатні синтезувати принципово нові зображення, що відтворюють статистичні характеристики розподілу цільового стилю. Генератор, навчений виключно на зразках конкретного художнього напрямку, засвоює характерні текстурні паттерни, колірні відношення та фактуру мазків, що дозволяє створювати синтетичні зображення для збагачення недопредставлених класів.

У контексті даної роботи генеративна модель використовується як інструмент балансування датасету. Для кожного класу з недостатньою

кількістю реальних зображень навчається генеративна мережа, яка синтезує необхідну кількість додаткових зразків. Такий підхід забезпечує репрезентативність навчальної вибірки для всіх досліджуваних стилів, що є передумовою коректного навчання класифікатора без зміщення у бік мажоритарних класів.

1.2.1 Базова структура GAN

Концепція генеративно-змагальних мереж була вперше запропонована Іеном Гудфеллоу та його колегами у 2014 році як новий фреймворк для неконтрольованого машинного навчання [2]. В основі моделі лежить змагальна гра двох нейронних мереж, які функціонують у режимі нульової суми. Перша мережа, генератор G , приймає на вхід вектор випадкового шуму z , отриманий з апіорного розподілу p_z і намагається відобразити його у простір даних, формуючи розподіл згенерованих зразків p_g . Друга мережа, дискримінатор D , отримує на вхід як справжні зображення x із цільового розподілу навчальної вибірки p_{data} , так і згенеровані «фейки» $G(z)$. Мета дискримінатора полягає у розрахунку ймовірності того, що отриманий зразок походить з реального розподілу p_{data} , а не від генератора. Математично цей процес формалізується як задача мінімаксної оптимізації цільової функції $V(D, G)$, який представлений у формулі (1.1), де генератор намагається мінімізувати функцію, а дискримінатор – максимізувати її.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(x)))]. \quad (1.1)$$

У цій формуляції дискримінатор оптимізується для того, щоб $D(x)$ наближалось до 1 для реальних даних, і $D(G(z))$ наближалось до 0 для

синтетичних. Натомість генератор оптимізує свої ваги таким чином, щоб змусити $D(G(z))$ наближатися до 1. Згідно з теоретичним аналізом, якщо обидві мережі мають достатню репрезентативну здатність і оптимізація проводиться у просторі неперервних функцій, система досягає рівноваги Неша (англ. Nash Equilibrium), за якої розподіл генератора ідеально збігається з розподілом реальних даних ($p_g = p_{data}$), а дискримінатор видає ймовірність 0.5 для всіх вхідних зразків, втрачаючи здатність розрізняти їхнє походження [2]. Проте досягнення цієї теоретичної рівноваги на практиці під час навчання багат шарових нейронних мереж методом стохастичного градієнтного спуску є надзвичайно складним завданням, що призводить до нестабільної конвергенції та деградації генератора.

Отже, підсумувавши вищезазначене, модель GAN складається з двох конкуруючих нейронних мереж, які навчаються одночасно.

1. Генератор, який приймає на вхід вектор випадкового шуму і намагається згенерувати синтетичне зображення, яке б максимально точно імітувало справжні дані цільового класу.
2. Дискримінатор, який отримує на вхід як справжні зображення із навчальної вибірки, так і згенеровані зразки. Він виконує роль критика, намагаючись відрізнити справжні зображення від згенерованих.

Кінцевою метою процесу навчання є досягнення стану рівноваги, за якого генератор створює настільки реалістичні дані, що дискримінатор помиляється приблизно у половині випадків, втрачаючи здатність до ідентифікації походження зразка. Загальний принцип роботи генеративно-змагальної мережі зображений на рисунку 1.1.

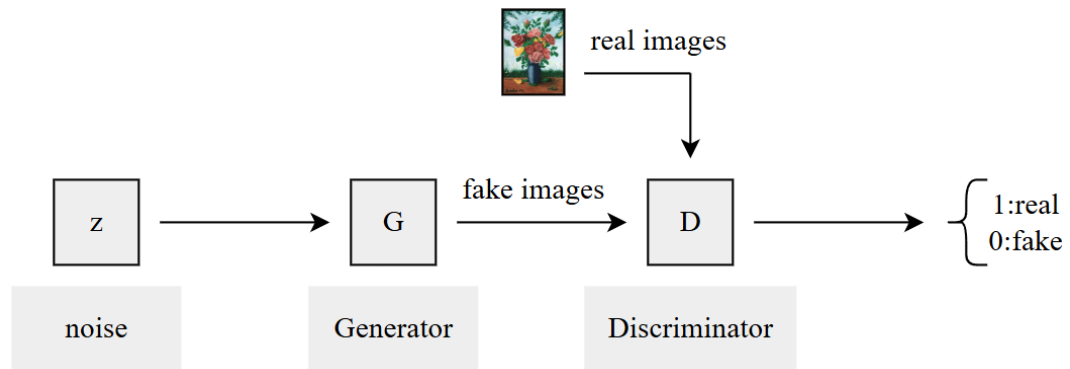


Рисунок 1.1 – Принцип роботи GAN

1.2.2 Класифікація GAN

Моделі на основі оптимізації цільової функції. Ця група архітектур фокусується на зміні математичного апарату оцінки помилки між генератором та дискримінатором для забезпечення стабільного навчання. Архітектура WGAN-GP використовує відстань Вассерштейна для вимірювання різниці між реальним та згенерованим розподілами. Додавання штрафу градієнта (Gradient Penalty) усуває необхідність жорсткого обмеження ваг, що значно стабілізує процес навчання та робить модель високостійкою до втрати варіативності [16]. LSGAN (Least Squares GAN) замінює стандартну крос-ентропійну функцію втрат функцією на основі методу найменших квадратів. Це запобігає проблемі зникнення градієнта для зразків, які знаходяться на правильному боці межі ухвалення рішення, але ще далеко від реальних даних [3]. В основі RGAN / RaGAN (Relativistic GAN) лежить релятивістський дискримінатор, який оцінює не абсолютну ймовірність того, що зразок реальний, а ймовірність того, що реальний зразок «більш реальний», ніж згенерований. Це збалансовує швидкість навчання генератора та дискримінатора [4]. QAGAN (Quality-Aware GAN) інтегрує

метрики оцінки якості зображень безпосередньо у функцію втрат як регуляризатори. Це змушує генератор звертати увагу не лише на статистичний розподіл, а й на піксельну та структурну цілісність результату [5].

Умовні моделі. Ці моделі дозволяють керувати процесом генерації шляхом подачі додаткової інформації, такої як мітки класів або текстові описи. CGAN (Conditional GAN) є базовою умовною архітектурою, де мітка класу об'єднується як до вхідного вектора шуму генератора, так і до вхідного зображення дискримінатора. У моделі ACGAN (Auxiliary Classifier GAN) дискримінатор не лише відрізняє реальні зображення від згенерованих, але й має додатковий класифікаційний вихід для передбачення класу зображення. Це покращує якість генерації конкретних класів. ReACGAN вдосконалює цей підхід для вирішення проблеми нестабільності на великих датасетах [6]. ArtGAN є розширенням умовних мереж для роботи з художніми зображеннями. Дискримінатор функціонує як автокодувальник, що дозволяє розплутувати ознаки стилю та контенту через механізм Denoising Feature Matching [7].

Моделі, орієнтовані на запобігання mode collapse. PacGAN змінює дискримінатор таким чином, щоб він приймав на вхід набір з певної кількості зображень одночасно (або всі реальні, або всі згенеровані). Якщо генератор страждає від втрати варіативності, дискримінатор миттєво це виявляє, оскільки в цьому наборі будуть повтори [8]. SetGAN використовує екваріантні шари для аналізу наборів зображень. Замість поштучної оцінки, модель порівнює статистичні властивості цілих батчів, забезпечуючи жорсткий контроль як за внутрішньокласовим, так і за міжкласовим різноманіттям [9]. VEEGAN вводить додаткову мережу-реконструктор, яка намагається відобразити згенероване зображення назад у вихідний вектор шуму. Цей механізм гарантує, що генератор використовує весь прихований простір, запобігаючи відкиданню цілих мод розподілу [10].

Еволюційні та просторові моделі. PEGANs (Phased Evolutionary GANs) застосовує еволюційні алгоритми для навчання популяції з кількох генераторів. Замість того, щоб покладатися на один градієнтний спуск, система використовує оператори мутації. Додатковий модуль Self-Attention дозволяє моделі враховувати віддалені просторові залежності на зображенні (глобальну композицію) [11].

Моделі для роботи зі стилями та високою роздільною здатністю. CycleGAN використовує дві пари генераторів та дискримінаторів для перенесення стилю між двома доменами (наприклад, «фотографія» —> «картина Ван Гога») без необхідності мати ідентичні за контентом зображення. Навчання базується на циклічній узгодженості (cycle-consistency loss) [12]. StyleGAN є однією з найпотужніших архітектур для високоякісної генерації. Модель відмовляється від традиційного подання шуму на вхід першого шару. Натомість шум подається через мережу відображення (mapping network), утворюючи проміжний вектор простору стилів. Це дозволяє розділити контроль над макроструктурами (композиція) та мікроструктурами (текстури, кольори) [13]. StyleGAN-ADA (Adaptive Discriminator Augmentation) розв'язує проблему навчання GAN на малих наборах даних. Вона застосовує адаптивну диференційовану аугментацію до зображень перед тим, як вони потраплять до дискримінатора, що запобігає його перенавчанню [14].

1.3 Проблема mode collapse та причини її виникнення

Одним із найбільш поширених та складних для вирішення сценаріїв збою під час навчання GAN є явище mode collapse, що проілюстровано на рисунку 1.2 (на базі моделі CGAN).



Рисунок 1.2 – Приклад генерації художніх картин

Це явище виникає тоді, коли генератор знаходить слабе місце дискримінатора. Замість того, щоб створювати різноманітні дані, він починає видавати лише кілька однакових варіантів, які точно проходять перевірку. У найгіршому випадку генератор зациклюється і постійно створює лише один зразок. Оскільки для нормального навчання системи потрібні різні зображення, такі одноманітні результати стають абсолютно непридатними для використання.

Основна причина цього явища ховається у тому, як саме математично розраховується різниця між реальними та створеними даними. У стандартних

моделях, коли дискримінатор починає працювати ідеально, головне завдання системи зводиться до зменшення математичної похибки, яка відома як дивергенція Єнсена-Шеннона (Jensen-Shannon, JS). Цей спосіб підрахунку, зі свого боку, спирається на іншу базову формулу — Кульбака-Лейблера (KL). [16]. Збій виникає через те, що реальні дані (зокрема зображення високої якості) займають дуже малу частину у великому просторі всіх можливих ознак. Математично ймовірність того, що реальні та згенеровані дані перетнуться, дорівнює нулю. Коли ці розподіли не перетинаються, дивергенція KL прямує до нескінченності, а дивергенція JS різко стає сталою величиною. Як наслідок, градієнт дивергенції JS щодо параметрів генератора дорівнює нулю. Цей процес називають зникненням градієнта (*vanishing gradient*). Якщо дискримінатор починає працювати занадто добре, генератор перестає отримувати корисний сигнал для свого покращення і не знає, як наблизити свої результати до реальних [1]. Не маючи чіткого напрямку, генератор починає зводити всі свої результати до тієї точки, де дискримінатор ще помиляється, щоб отримувати максимальні оцінки для вузького набору зразків.

Інша причина полягає у нерівномірності дивергенції KL. Коли генератор намагається охопити різні варіанти реальних даних, він неминуче створює проміжні зразки, яких насправді не існує. Система суворо штрафує за такі помилки, але майже не реагує, якщо генератор просто ігнорує частину існуючих реальних даних. Тому генератору стає вигідно зосередитися лише на кількох безпечних варіантах, які гарантують високі оцінки дискримінатора, але призводять до втрати загального різноманіття.

Катастрофічне забування (англ. *catastrophic forgetting*) у дискримінаторі також сильно сприяє цьому збою. Оскільки дискримінатор навчається поступово, він схильний забувати ті варіанти, які генератор перестав створювати на попередніх етапах. Як тільки генератор знаходить новий вдалий варіант і починає його постійно використовувати, дискримінатор з

часом вчиться визначати його як фейк, але при цьому втрачає здатність розпізнавати попередні зразки. Це призводить до того, що генератор просто циклічно стрибає між різними варіантами, так і не досягаючи загального розмаїття [20].

З моменту свого створення алгоритми GAN зазнали інтенсивної еволюції, що була здебільшого продиктована необхідністю подолання mode collapse та покращення стабільності навчання. На ранніх етапах розвитку генеративного моделювання оцінка якості та розмаїття згенерованих зображень покладалася виключно на візуальний огляд, що унеможливлювало об'єктивне порівняння різних підходів. Поява глибоких згорткових архітектур, таких як DCGAN у 2015 році, запровадила певні структурні стандарти (використання пакетної нормалізації та відмова від повністю зв'язних шарів), що дещо зменшило нестабільність, але не вирішило проблему зникнення градієнтів та однакове генерування зразків на фундаментальному рівні [23].

Перший значний крок у формалізації проблеми відбувся у 2016 році з розробкою метрики Inception Score (IS). Цей показник використовує попередньо навчену на наборі даних ImageNet мережу Inception-v3 для вилучення ознак і обчислення якості та різноманітності зображень. IS розраховує експоненту дивергенції KL між умовним розподілом класів $p(y|x)$ (що характеризує впевненість класифікатора в тому, що зображення містить чіткий об'єкт) та маргінальним розподілом $p(y)$ (що характеризує варіативність згенерованих класів). Хоча високий показник IS вказує на те, що модель генерує різноманітні і якісні зображення, він має серйозну ваду: IS не здійснює жодного порівняння синтетичних зображень із розподілом цільової навчальної вибірки. Крім того, IS є нечутливим до внутрішньокласового mode collapse – якщо генератор продукує ідеальне, але єдине зображення для кожного класу, показник IS буде штучно завищеним.

Революцією в оцінюванні генеративних моделей стало впровадження у 2017 році метрики Fréchet Inception Distance. FID безпосередньо вимірює відстань між ймовірнісними розподілами реальних та згенерованих зображень у глибокому просторі ознак. Використовуючи активації шару pooling тієї ж мережі Inception-v3, FID апроксимує ці активації як багатовимірні гауссівські розподіли та обчислює відстань Фреше (також відому як відстань Вассерштейна) між ними, як зазначено у формулі (1.2) [18].

$$FID = \|\mu_r - \mu_g\|_2^2 + Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}\right), \quad (1.2)$$

де μ_r, μ_g – вектори середніх значень, а Σ_r, Σ_g – коваріаційні матриці для реальних і згенерованих ознак відповідно. FID став галузевим стандартом завдяки своїй високій чутливості до mode collapse: якщо генератор втрачає різноманітність, коваріаційна матриця згенерованих зразків Σ_g суттєво зменшується, що призводить до різкого збільшення значення відстані. Додаткові інструменти, такі як розрахунок Birthday Paradox та оцінка метрики Coverage, згодом були розроблені для безпосереднього підрахунку кількості унікальних зразків, які генератор здатний відтворити, проте FID залишається найбільш комплексним та загальноприйнятим індикатором [18].

У 2016 році були запропоновані інші алгоритми. Метод міні-батч дискримінації (англ. minibatch discrimination) дозволяє дискримінатору аналізувати одразу цілу групу зображень. Він оцінює, наскільки вони схожі між собою. Якщо картинки виявляються майже однаковими, дискримінатор відхиляє їх, що змушує генератор щоразу створювати різні об'єкти. Раніше згаданий метод PacGAN, який об'єднує m зображень у загальний пакет, і дискримінатор приймає одне спільне рішення для всіх. Якщо генератор дає збій, і всі m зразків виявляються однаковими, дискримінатору дуже легко

виявити обман. Цей підхід легко впровадити, і він чудово підтримує різноманітність результатів [16]. Unrolled GAN діють на випередження. Тут генератор намагається обійти дискримінатор не лише в поточний момент, а й передбачити його реакцію на декілька майбутніх кроків вперед. Оскільки генератор розуміє, що за k кроків дискримінатор легко розпізнає хитрощі з однаковими картинками, йому стає не вигідно їх створювати. Це дуже надійний метод, але він вимагає значних обчислювальних ресурсів комп'ютера [17].

Період з 2018 по 2023 роки ознаменувався впровадженням методів прогресивного зростання (ProGAN) та спектральної нормалізації (Spectral Normalization), які стабілізували ліпшицеві константи мереж без необхідності застосування складних штрафних санкцій. Останні досягнення 2024-2026 років спрямовані на поєднання регуляризації різноманітності (Diversity Regularization) та методів боротьби з катастрофічним забуванням, таких як архітектура AMAT (Adaptive Multi Adversarial Training), що адаптивно додає нові дискримінатори для збереження історії генерації. Новітні метрики, такі як DMD-GEN, що базуються на динамічній декомпозиції, починають застосовуватися для ідентифікації збою у складних часових рядах і неструктурованих даних, розширюючи сферу застосування теорії на нові горизонти [24].

Висновки до розділу 1

У першому розділі було проведено всебічний аналіз предметної галузі, пов'язаної з автоматизованою класифікацією художніх стилів живопису. Дослідження засвідчило, що ця задача є значно складнішою за традиційне розпізнавання об'єктів реального світу через відсутність чітких візуальних

меж між різними мистецькими напрямками. Було встановлено, що головною перешкодою для якісного навчання нейронних мереж є критичний дисбаланс навчальних даних, де популярні стилі представлені тисячами зразків, а вузькоспеціалізовані – лише сотнями. Це створює умови, за яких класичні моделі схильні ігнорувати рідкісні класи. Для розв'язання цієї проблеми було теоретично обґрунтовано доцільність інтеграції генеративно-змагальних мереж як інструменту для синтетичного розширення вибірки. Водночас аналіз виявив фундаментальну вразливість до mode collapse, що полягає у втраті варіативності згенерованих зображень. Тому було визначено необхідність пошуку методів, які дозволять стабілізувати процес генерації та забезпечити збалансовану вибірку даних для подальшого навчання класифікатора.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ТА МАТЕМАТИЧНІ ОСНОВИ ОБРАНИХ МОДЕЛЕЙ

2.1 Алгоритми запобігання mode collapse

У контексті цієї роботи ризик виникнення mode collapse є високим з огляду на специфіку задачі: генеративна мережа навчається на вибірках окремих художніх стилів, частина яких налічує лише кілька сотень реальних зображень. За таких умов дискримінатор здатний порівняно швидко «запам'ятати» всі наявні реальні зразки, що позбавляє генератор змістовного навчального сигналу та провокує колапс різноманітності згенерованих зображень. На початковому етапі дослідження було розглянуто умовну генеративно-змагальну мережу як альтернативний підхід до балансування датасету. CGAN навчає єдину мережу для всіх 15 стилів одночасно використовуючи мітку класу як додаткову умову генерації. Однак практична реалізація CGAN для датасету WikiArt зіткнулась з критичною проблемою mode collapse: при одночасному навчанні на 15 класах з різним рівнем представленості генератор систематично колапсував до генерації обмеженого набору стилів ігноруючи міноритарні класи. Застосованих методів регуляризації виявилось недостатньо для подолання нестабільності при такій кількості умовних класів. Це підтвердило доцільність переходу до підходу з окремими моделями WGAN-GP для кожного класу, що усуває конкуренцію між класами під час навчання. Для систематичного подолання цієї проблеми в реалізованій системі застосовано комплекс взаємодоповнювальних методів.

2.1.1 Архітектура WGAN-GP

Фундаментальним проривом у подоланні зникнення градієнтів та запобіганні mode collapse став перехід до метрики відстані Вассерштейна. На відміну від дивергенцій KL та JS, відстань Вассерштейна обчислює мінімальні витрати на перенесення даних, необхідні для перетворення одного розподілу на інший [15]. Математично відстань Вассерштейна визначається за формулою (2.1).

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|], \quad (2.1)$$

де $\Pi(P_r, P_g)$ – множина всіх спільних розподілів $\gamma(x, y)$, маргінальні розподіли яких дорівнюють P_r та P_g [3].

Головна перевага цього підходу полягає в тому, що він працює плавно і без збоїв, навіть якщо набори даних взагалі не перетинаються. Це гарантує, що генератор завжди отримуватиме корисні сигнали для навчання, незалежно від того, наскільки його результати відрізняються від справжніх.

Для того, щоб нейромережа могла правильно розрахувати цю відстань, за теоремою Канторовича-Рубінштейна дискримінатор повинен мати спеціальну математичну властивість – бути 1-ліпшицевим (1-Lipschitz continuous). У першій версії WGAN для цього використовували відсікання ваг (weight clipping) – штучно обмежували параметри критика вузьким діапазоном. Хоча це і зробило процес навчання плавнішим, виникли нові проблеми. Жорстке обмеження сильно заважає нейромережі нормально працювати, оскільки більшість її параметрів просто скупчуються на самих краях дозволеного діапазону. Через це ставало дуже складно налаштовувати великі мережі, що часто призводило до неконтрольованого зростання або ж повного зникнення сигналів під час навчання.

Для вирішення цих обмежень було розроблено архітектуру WGAN-GP. Цей алгоритм повністю відмовився від відсікання ваг, замінивши його на градієнтний штраф, який додається безпосередньо до цільової функції дискримінатора. Метод ґрунтується на математичному спостереженні: диференційована функція є 1-ліпшицевою тоді й лише тоді, коли норма її градієнта майже скрізь не перевищує 1. Функція втрат критика у WGAN-GP набуває вигляду формули (2.2).

$$L_D = \mathbb{E}_{x \sim P_g}[D(x)] - \mathbb{E}_{x \sim P_r}[D(x)] + \lambda \mathbb{E}_{\hat{x} \sim P_{\hat{x}}} |(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2|, \quad (2.2)$$

де λ є коефіцієнтом штрафу (як правило, встановлюється на рівні 10), а $\hat{x}$ – це точки, випадково вибрані шляхом лінійної інтерполяції між парами реальних та згенерованих зразків. Цей штраф змушує норму градієнта дискримінатора щодо його вхідних даних наближатися до 1 у тих областях простору ознак, де зосереджена ймовірнісна маса [16].

WGAN-GP забезпечує стабільне навчання без зникнення градієнтів та значно ефективніше запобігає mode collapse, оскільки генератор завжди отримує чіткий напрямок для оптимізації, що дозволяє йому досліджувати весь латентний простір. Саме архітектура WGAN-GP була обрана, як фундамент генеративного модуля в інтелектуальній системі розпізнавання стилів картин, дозволяючи згенерувати синтетичні полотна з високим рівнем різноманітності.

2.1.2 Регуляризаційні підходи та сучасні алгоритми

Попри успіхи структурних методів, найбільш прямолінійним шляхом вирішення проблеми є впровадження функцій регуляризації, що

безпосередньо штрафують мережу за відсутність різноманітності. Суть підходу полягає у максимізації відношення відстані між згенерованими зображеннями до відстані між їхніми латентними векторами: генератор змушений забезпечувати, щоб дві різні точки латентного простору z_1 та z_2 завжди проектувалися в різні візуальні простори $G(z_1)$ та $G(z_2)$ [21].

Для підвищення якості синтезу картин, до базової архітектури слід інтегрувати функцію Diversity Loss. Тоді цільова функція генератора модифікується до формули (2.3).

$$L_G = -\mathbb{E}[D(G(z_1))] + \alpha \cdot L_{div}, \quad (2.3)$$

де $L_{div} = -\mathbb{E}[\|G(z_1) - G(z_2)\|]$ обчислює від'ємне середнє попіксельне відхилення між згенерованими зображеннями, а α є ваговим коефіцієнтом регуляризації. Застосування цієї функції змушує генератор максимізувати дисперсію вихідних даних, протидіючи тенденції WGAN концентруватися лише на найбільш виражених текстурах стилю.

Для додаткового захисту від перенавчання дискримінатора, що може спровокувати збій на міноритарних класах, до реальних зображень перед аналізом додається гаусівський шум, а також використовуються шари dropout. Крім того, впровадження методів диференційованої аугментації, штучно збільшує різноманітність навчальної вибірки, що є критичним для забезпечення стабільності WGAN-GP на малих класах.

Дослідження 2024-2025 років пропонують нові горизонти у регуляризації. Модель HingeRLC-GAN поєднує функцію втрат Hinge Loss із регуляризацією контролю втрат (Regularized Loss Control, RLC). RLC забороняє надмірну впевненість дискримінатора шляхом додавання суми квадратів норм його градієнтів до функції втрат. Це призводить до згладжування меж прийняття рішень, що не дозволяє дискримінатору надто

швидко відкидати зразки генератора, надаючи йому простір для дослідження нових мод і досягнення видатних показників FID [22].

Іншим проривним алгоритмом є АМАТ, який вирішує проблему катастрофічного забування в дискримінаторі шляхом динамічного адаптивного розгортання нових паралельних дискримінаторів під час навчання. Коли система фіксує втрату певної ознаки розподілу, вона ініціалізує новий дискримінатор, який спеціалізується на збереженні пам'яті про цю специфічну ділянку простору, гарантуючи повне покриття всіх концептів навіть на складних датасетах [23]. Методи спектральної регуляризації (Spectral Regularization) довели свою ефективність, оскільки вони стабілізують вагові коефіцієнти дискримінатора. Це дозволяє фіксувати ознаки наближення mode collapse на найраніших етапах. Крім того, варто відзначити метрику DMD-GEN. Вона дозволяє виявляти проблему mode collapse у складних часових рядах, тим самим доводячи, що цей збій характерний не лише для генерації зображень [24].

2.2 Згорткові нейронні мережі

Завдяки своїй ефективності в розпізнаванні закономірностей, людський мозок може швидко ідентифікувати такі абстрактні поняття, як стиль. Штучні нейронні мережі – це математичні структури, здатні навчатися та згодом розпізнавати закономірності. З огляду на це, метою даної роботи є використання згорткових нейронних мереж як методу ідентифікації та класифікації стилів художніх картин.

Згорткові нейронні мережі є фундаментом сучасної обробки зображень. Архітектура типової CNN є послідовністю різних типів шарів, кожен з яких виконує специфічне математичне перетворення вхідного тензора даних.

Convolutional Layer. Це основний шар мережі, відповідальний за витягування локальних ознак. Операція згортки полягає у ковзанні набору фільтрів (ядер) певного розміру (наприклад, 3x3 або 5x5) по вхідному зображенню з певним кроком (stride). Під час ковзання виконується операція матричного скалярного добутку між значеннями пікселів зображення та вагами фільтра, після чого додається значення зміщення (bias). Математично операція згортки для генерації карти активацій (activation map) A описується формулою (2.4).

$$A_{i,j,k} = \sum_{p=1}^f \sum_{q=1}^f \sum_{r=1}^c I'_{i+p-1,j+q-1,r} \cdot K_{p,q,r,k} + b_k, \quad (2.4)$$

де $A_{i,j,k}$ – значення активації для пікселя з координатами (i,j) на k -му каналі, I' – нормалізоване вхідне зображення, K – тензор ваг фільтра, f – просторовий розмір фільтра, c – кількість каналів у вхідному зображенні (наприклад, 3 для простору RGB). Розмір фільтра відіграє критичну роль у визначенні художнього стилю [2]. Хоча сучасні мережі тяжіють до використання малих фільтрів 3x3, дослідження показують, що збільшення розміру ядра до 5x5 на перших шарах дозволяє охопити ширшу площу мазків пензля, що емпірично підвищує точність ідентифікації на 2-3% у задачах класифікації живопису.

Pooling Layer. Цей шар застосовується для зменшення просторової розмірності (ширини та висоти) карт активацій, що зменшує кількість параметрів мережі та обчислювальне навантаження, а також забезпечує інваріантність моделі до невеликих трансляційних зсувів об'єктів [31]. Найпоширенішим є використання операції Max Pooling (вибір максимального значення в локальному вікні розміром, як правило, 2x2 з кроком 2). Операція Max Pooling для області визначається за формулою (2.5).

$$P_{i,j,k} = \max_{p,q \in \{0,1\}} A_{2i+p,2j+q,k}. \quad (2.5)$$

Однак, стандартний Max Pooling може призводити до втрати важливих високочастотних деталей художнього стилю. Для вирішення цього в передових архітектурах застосовують операцію розмитого пулінгу (Blur Pool), яка комбінує субдискретизацію з низькочастотною фільтрацією (low-pass filtering). Цей підхід ефективно зменшує високочастотний шум, зберігаючи при цьому критично важливі текстурні характеристики картини, мінімізуючи ефект згладжування (aliasing).

Fully Connected Layer. Це повнозв'язний шар, який розташовується в кінці нейронної мережі і відповідає за фінальну класифікацію. У цьому шарі кожен нейрон з'єднаний з усіма нейронами попереднього шару. Вихід j -го нейрона y_j обчислюється за формулою (2.6) як зважена сума його входів, пропущена через нелінійну функцію активації σ .

$$y_j = \sigma(\sum_i W_{ij}x_i + b_j), \quad (2.6)$$

де W_{ij} – ваги з'єднань, x_i – входні значення, b_j – зміщення (bias). У випадку багатокласової класифікації стилів, на виході останнього шару використовується функція активації Softmax, яка перетворює "сирі" виходи мережі (logits) у нормований розподіл ймовірностей, сума яких дорівнює одиниці. Ймовірність належності зображення до конкретного художнього стилю k обчислюється за формулою (2.7).

$$\hat{z}_k = \frac{\exp(z_k)}{\sum_{i=1}^K \exp(z_i)}, \quad (2.7)$$

де K – загальна кількість стилів у наборі даних. Навчання класифікатора оптимізується шляхом мінімізації функції втрат категоріальної крос-ентропії (Categorical Cross-Entropy Loss), як написано у формулі (2.8).

$$Loss = -\sum_{k=1}^K y_k \log(\hat{y}_k), \quad (2.8)$$

де y_k – істинна мітка класу (у форматі one-hot encoding), а $\hat{y}_k$ – передбачена моделлю ймовірність.

Функції активації та проблема зникнення градієнта. Для введення нелінійності у систему використовуються функції активації. Історично стандартом була функція ReLU (Rectified Linear Unit), яка визначається як $f(x) = \max(0, x)$. Проте ReLU страждає від проблеми "відмирання нейронів" (dying ReLU), оскільки градієнт для від'ємних вхідних значень дорівнює нулю, що блокує оновлення ваг під час зворотного поширення помилки (backpropagation). У задачах аналізу мистецтва, де негативні значення можуть нести важливу інформацію про колірні контрасти, доцільно використовувати покращені функції, такі як LeakyReLU – формула (2.9) або CELU (Continuously Differentiable Exponential Linear Units) [30]. Використання CELU запобігає "відмиранню" нейронів і емпірично підвищує точність класифікації художніх зображень, забезпечуючи стабільніший розрахунок градієнтів.

$$f(x) = \begin{cases} x, & \text{якщо } x > 0 \\ 0.01x, & \text{якщо } x \leq 0 \end{cases} \quad (2.9)$$

Вибір правильної архітектури CNN є критичним для забезпечення високої точності розпізнавання стилів. У цьому контексті робота має на меті порівняти та проаналізувати відмінності в застосуванні та архітектурі передових CNN, таких як VGG, залишкові нейронні мережі ResNets та Xception.

ResNet. ResNet, скорочення від Residual Networks (Залишкові нейронні мережі) – це нейронна мережа, яка використовується переважно для завдань комп'ютерного зору. Головною інновацією цієї мережі стало те, що вона дозволила успішно навчати глибокі нейронні мережі з понад 150 шарами, що раніше було складно реалізувати через відому проблему зникаючого градієнта. Нова концепція, запроваджена в ResNet – це пропуски з'єднань» (skip connections). Вони дозволяють градієнту поширюватися безпосередньо до шарів, які страждають від згаданої проблеми [27]. Реалізація є відносно простою: між шарами, які раніше були лише послідовними, тепер існують прямі зв'язки, як показано на рисунку 2.1.

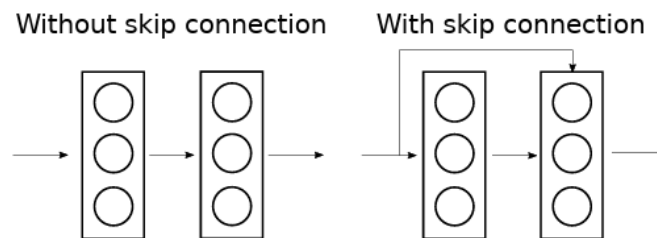


Рисунок 2.1 – Структура базового блоку архітектури ResNet [11]

VGG. VGG – це модель глибоких згорткових нейронних мереж для розпізнавання зображень у великих масштабах, розроблена групою Visual Geometry Group [26]. Характеризуються використанням виключно малих згорткових фільтрів (3x3) у поєднанні з max pooling. Незважаючи на популярність, вони генерують гігантську кількість параметрів, що призводить до повільного навчання та високого ризику перенавчання. Процес обробки виглядає так:

- 1) на вхід подається RGB-зображення фіксованого розміру (224×224);
- 2) зображення проходить через серію згорткових шарів;

- 3) після деяких з цих шарів застосовується шар max-pooling розміром 2×2 із кроком 2;
- 4) наприкінці розташовані три fully connected шари — перший має 4096 каналів, а останній — 1000, що дозволяє класифікувати зображення на 1000 класів.

Повну архітектурну схему можна побачити на рисунку 2.2.

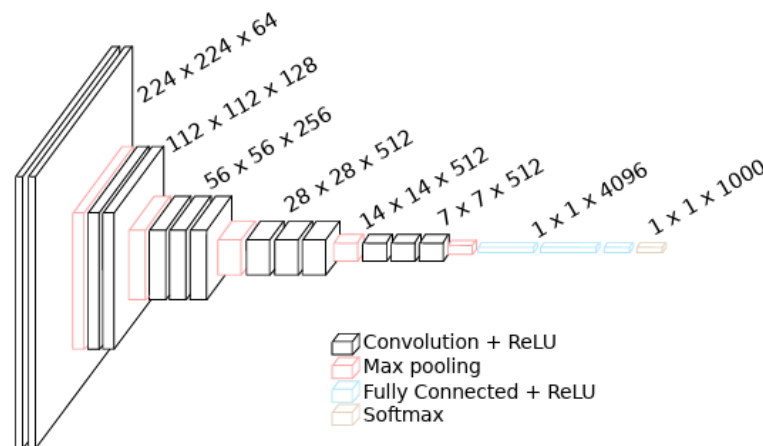


Рисунок 2.2 – Схема архітектури моделі VGG [26]

Хception. Архітектура Хception, опублікована у 2016 році, є результатом еволюції згорткових нейронних мереж. Вона розроблена компанією Google і базується на архітектурі Inception-V3. Модель Хception замінює стандартні згортки на глибинно-роздільні. Її алгоритм наступний [28]:

- 1) канали зображення розділяються;
- 2) до кожного каналу застосовуються специфічні згорткові фільтри;
- 3) після цього виконується поштучна згортка (pointwise convolution), тобто застосовується фільтр 1×1 після глибинної згортки.

Це радикально зменшує кількість параметрів мережі та підвищує точність класифікації на збалансованих вибірках.

Attention Mechanisms. Останнім словом у комп'ютерному зорі є впровадження модулів уваги. Оскільки стилі живопису часто розрізняються лише за мікроскопічними нюансами мазків, традиційні згорткові шари можуть витягувати забагато фонового шуму. Впровадження потрійної уваги, яка складається з трьох паралельних гілок для аналізу взаємодії між просторовим розташуванням пікселів та каналами кольору, дозволяє мережі прицільно фокусуватися виключно на критично важливих текстурних паттернах, повністю ігноруючи надлишкову інформацію. Це знижує обчислювальне навантаження та підвищує точність розпізнавання дуже схожих стилів на понад 11% [35].

Vision Transformers та MLP-Mixer. Новітні архітектури взагалі відмовляються від операції згортки. Модель ViT розбиває вхідне зображення картини на сітку неперекривних фрагментів (патчів), перетворює їх на лінійні вектори і обробляє за допомогою механізму само-уваги (Self-Attention), подібно до того, як обробляється текст у мовних моделях. Це дозволяє моделі фіксувати глобальні семантичні взаємозв'язки між дуже віддаленими фрагментами полотна. Архітектура MLP-Mixer йде ще далі, базуючись виключно на багатошарових перцептронах для змішування інформації між просторовими локаціями та каналами. Тренування таких архітектур з нуля на складних датасетах мистецтва доводить їхню високу конкурентоспроможність.

У контексті оптимізації архітектур згорткових нейронних мереж важливу роль відіграють методи структурно-параметричного синтезу [19]. Цей підхід дозволяє формалізувати вибір типу нейромережі відповідно до складності задачі та оптимізувати її топологію для досягнення високої точності за мінімальних обчислювальних витрат. Інтеграція таких методів обґрунтовує стратегію навчання та є доцільною у задачах класифікації творів

мистецтва, оскільки забезпечує ефективну адаптацію моделей до незбалансованих вибірок і підвищує загальну стійкість системи до перенавчання.

2.3 Огляд сучасних досліджень у сфері класифікації художніх картин

Тема класифікації творів образотворчого мистецтва викликає постійний інтерес дослідників останні кілька років. При першій спробі класифікації картин використовували наївний байєсівський класифікатор на основі локальних ознак. Більшість ранніх робіт із класифікації стилів мали спільну методологію:

- виділення різних ознак зображення (форма, текстура, межі та колір);
- їх використання для навчання класифікаторів, таких як метод опорних векторів (SVM), багатошаровий перцептрон (MLP) або k-найближчих сусідів (k-NN).

Нещодавно стрімкий прогрес у цій сфері відбувся завдяки значному розвитку комп'ютерного зору через впровадження згорткових нейронних мереж. Спочатку CNN застосовувались виключно як екстрактори ознак. Дослідники показали, що активації шарів CNN (попередньо навченої на ImageNet на звичайних фото) як ознаки дають високі результати в розпізнаванні стилів і перевершують ручні алгоритми.

Ще кращих результатів досягнуто завдяки fine-tuning попередньо навченої мережі на новому цільовому наборі даних. Цей підхід ефективніший за просте вилучення ознак і успішно застосовується для різноманітних мистецьких класифікацій.

Хоча донавчання потребує менше даних, ніж навчання глибокої CNN з нуля, великий обсяг зображень залишається обов'язковою умовою. Поява колекції WikiArt (англ. Visual Art Encyclopedia) стала стандартом для порівняння методів глибокого навчання у цій сфері. Також використовуються інші бази:

- Web Gallery of Art (WGA) – понад 40 тис. зображень;
- датасет Rijksmuseum challenge;
- Painting-91 – 4266 зображень 91 художника;
- Pandora – 7724 зображення з 12 мистецьких течій;
- OmniART – понад 1 млн фоторепродукцій.

На основі використаних даних та методологій варто виділити кілька робіт для порівняння. Вперше набір даних WikiArt було адаптовано для класифікації художників, стилів і жанрів у роботі [29]. Найкращий результат забезпечив метод злиття ознак (feature fusion), що включав CNN-ознаки. Потім на тому ж наборі даних аналітики виконали fine-tuning CNN. Вони довели, що цей підхід перевершує, як використання лише вилучених CNN-ознак, так і навчання мережі з нуля на картинах [30]. А в наступних роботах було досягнуто ще вищої продуктивності завдяки fine-tuning глибших моделей. Для класифікації стилів за допомогою трансферного навчання було порівняно ефективність шести архітектур CNN (Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121 та EfficientNet B3), а також досліджено вплив різних оптимізаторів і швидкостей навчання [31]. Архітектура InceptionResNetV2 виявилася найточнішою моделлю для класифікації стилів на обох наборах даних. Найвищого результату вона досягла у поєднанні з оптимізатором Adam та швидкістю навчання $1e - 4$.

У недавньому дослідженні [32] запропоновано архітектуру ArtSAGENet, яка об'єднує CNN та графові нейронні мережі (GNN) для одночасного вивчення візуальних і семантичних ознак із використанням

багатозадачного навчання. Також доведено, що використання GAN для аугментації даних покращує роботу моделі EfficientNet B0 [33].

Попередні роботи були зосереджені на різних методологіях і моделях CNN. Проте ключем до високих результатів є правильний вибір гіперпараметрів. Вплив різних оптимізаторів у сфері класифікації мистецтва ще не досліджувався, хоча це активно вивчалось в інших галузях. Наприклад, у дослідженні [34] запропоновано інноваційну двоетапну архітектуру глибокого навчання для семантичної класифікації стилів образотворчого мистецтва. Головна особливість цього методу полягає у просторовому поділі вхідного зображення на п'ять окремих фрагментів. На першому етапі ці фрагменти незалежно аналізуються глибокою згортковою нейромережею (наприклад, Inceptionv3 або ResNet-50). На другому етапі отримані ймовірнісні вектори для кожного фрагмента об'єднуються та подаються на вхід додатковій нейронній мережі для прийняття фінального рішення. Експерименти на трьох стандартизованих наборах даних довели, що така архітектура суттєво перевершує традиційні базові підходи. Дослідники встановили, що алгоритм досягає максимальної ефективності, коли локальний аналіз окремих ділянок полотна поєднується з цілісним аналізом загальної композиції. У цьому випадку друга стадія класифікації ефективно виконує функцію оптимізації, нівелюючи помилкові прогнози, зроблені під час аналізу окремих фрагментів.

У дослідженні 2021 року [35] проведено масштабний порівняльний аналіз семи сучасних архітектур згорткових нейронних мереж (зокрема варіацій ResNet, EfficientNet_B3 та ResNeSt) для автоматичної класифікації творів мистецтва. Головною метою роботи був розподіл картин за трьома критеріями: ідентифікація художника, визначення художнього стилю та жанру. Тестування алгоритмів здійснювалося на трьох стандартизованих наборах даних різного масштабу: Painting-91, WikiArt-WikiPaintings та незбалансованому MultitaskPainting100K. Важливим висновком дослідження

стала практично доведена необхідність застосування трансферного навчання. Автори підтвердили, що використання моделей, попередньо навчених на масивах звичайних фотографій (база ImageNet), суттєво підвищує точність класифікації живопису, особливо під час роботи з невеликими базами даних. Шляхом оптимізації структури нейромереж дослідникам вдалося перевершити попередні наукові результати та встановити нові рекорди точності для всіх трьох наборів даних. Найвищу загальну ефективність продемонстрували моделі EfficientNet_B3 та ResNeSt50. Зокрема, точність ідентифікації художника на базі WikiArt досягла 91,73%, а розпізнавання стилю на базі Painting-91 склало 79,23%. Окрім кількісних метрик, автори дослідили специфіку алгоритмічного «сприйняття» живопису за допомогою методів візуалізації. Аналіз матриць невідповідностей показав, що штучний інтелект з високою точністю (до 95%) розпізнає унікальні стилі, такі як абстрактний експресіонізм, проте часто стикається з труднощами при диференціації історично неперервних та візуально схожих напрямів, нерідко плутаючи бароко з неокласицизмом, а реалізм — із романтизмом.

У 2025 році було запропоновано підхід до вирішення проблеми автоматичної класифікації художніх стилів із використанням оптимізованої згорткової нейронної мережі MobileNetV2 [36]. Метою роботи було створення ефективної системи для розпізнавання восьми класичних напрямів живопису на основі вибірки з набору даних WikiArt. Для підвищення здатності моделі до узагальнення та компенсації обмеженого обсягу даних, автор застосував методи аугментації зображень і трансферного навчання, використовуючи попередньо навчені ваги з масиву ImageNet. Результати експерименту продемонстрували, що запропонована модель досягає загальної точності 67,6%. Найвищу ефективність класифікації було зафіксовано для яскраво виражених стилів: кубізму (89,1%) та імпресіонізму (73,1%). Водночас дослідження виявило суттєві обмеження алгоритмів комп'ютерного зору при роботі з перехідними історичними стилями та

творами, що мають спільні візуальні характеристики. Зокрема, картини епохи романтизму класифікувалися з найнижчою точністю (53,3%) через змішування стилістичних рис. Для подолання цієї проблеми у майбутніх дослідженнях автор пропонує перехід до багатоетикеткової класифікації (multi-label classification), що дозволить фіксувати співіснування кількох стилів на одному полотні, а також залучення зовнішніх контекстних даних (наприклад, року створення твору чи біографії художника).

У контексті розвитку сучасних методів комп'ютерного зору для вирішення задач класифікації, окремої уваги заслуговують архітектури, що стають альтернативою класичним CNN. Зокрема, у дослідженні [37] розглядається застосування моделей Vision Transformer (ViT) та MLP Mixer для складної задачі розпізнавання художніх стилів картин. Особливість цих моделей полягає у відмові від традиційних згорток: ViT покладається на механізми уваги для обробки фрагментів зображення як послідовностей, тоді як MLP Mixer використовує виключно архітектуру багат шарових перцептронів для крос-локаційних та поканальних операцій. Дослідники здійснили навчання моделей «з нуля» на масштабному наборі даних WikiArt. Результати експериментів продемонстрували конкурентоспроможність обох підходів: точність розпізнавання склала 39,89% для ViT (при використанні оптимізатора Adam) та 39,59% для MLP Mixer. Крім того, проведене порівняння гіперпараметрів підтвердило перевагу алгоритму Adam над іншими методами оптимізації (SGD, RMSProp тощо) у задачах такого типу. Це дослідження підтверджує, що новітні архітектури, які не використовують CNN, здатні ефективно виділяти складні високорівневі ознаки зображень і є дієвим інструментом для задач багатокласової класифікації в умовах високої варіативності даних.

2.4 Постановка задачі

Головним завданням цієї роботи є розробка інтелектуальної системи визначення художніх стилів, здатної з високою точністю класифікувати твори образотворчого мистецтва за стилем. Фундаментальною проблемою, що постає під час реалізації такого підходу, є природний класовий дисбаланс у наявних базах даних, оскільки кількість збережених та оцифрованих картин суттєво відрізняється залежно від конкретної епохи чи стилю. Навчання глибоких згорткових нейронних мереж безпосередньо на таких незбалансованих масивах інформації неминуче призводить до зміщення передбачень моделі в бік домінуючих класів та критичного зниження загальної узагальнювальної здатності алгоритмів.

Для розв'язання цієї проблеми в роботі пропонується комплексний підхід, що об'єднує процеси генерації та класифікації. Відмовляючись від традиційних, але обмежених методів простої просторової чи колірної аугментації, які не здатні відтворити унікальні текстурні особливості та складну семантику живопису, дослідження фокусується на застосуванні генеративно-змагальних мереж. Зокрема, передбачається використання архітектури WGAN-GP для синтезу нових зображень тих стилів, що є недостатньо представленими у вибірці. При цьому особлива увага приділяється механізмам запобігання втраті різноманітності згенерованих зразків, щоб забезпечити створення унікальних та варіативних візуальних даних, а не генерацію одноманітних патернів.

Центральним дослідницьким питанням роботи є встановлення закономірностей впливу синтетично розширеної вибірки на ефективність навчання алгоритмів розпізнавання. Мета полягає не лише у досягненні максимальних показників точності класифікації, але й у глибокому аналізі меж застосовності генеративного балансування. Важливим аспектом є

розуміння умов, за яких синтетичні дані дійсно сприяють формуванню стійких ознак для класифікатора, та виявлення ситуацій, коли такий синтез вносить деструктивний шум чи артефакти. Таким чином, результати дослідження мають як вагому прикладну цінність у вигляді побудови надійної системи аналізу мистецтва, так і наукову значущість для розвитку методів комп'ютерного зору в задачах зі складними візуальними категоріями.

Висновки до розділу 2

Другий розділ присвячено дослідженню теоретичних та математичних засад обраних моделей штучного інтелекту. На основі проведеного аналізу системи було обрано архітектуру WGAN-GP для етапу генерації. Додатково було розглянуто механізм градієнтного штрафу, який накладає обмеження на ліпшицеву константу мережі, забезпечуючи плавність навчання. Для гарантування візуальної різноманітності картин було обґрунтовано впровадження функції Diversity Loss, яка безпосередньо штрафує генератор за створення однотипних зразків. У контексті задачі класифікації було проведено порівняльний аналіз архітектур згорткових нейронних мереж. Це закладає надійний теоретичний фундамент для побудови інтелектуальної системи, де генеративна модель виступає засобом підготовки даних, а згорткова мережа – інструментом фінального аналізу та розпізнавання стилів.

РОЗДІЛ 3 ЗАПРОПОНОВАНИЙ ПІДХІД ДО ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1 Опис навчальної вибірки

Ефективність будь-якої моделі глибокого навчання прямо пропорційна якості та обсягу вхідних даних, на яких вона навчається. Для задач комп'ютерного аналізу мистецтва загальновизнаним «золотим стандартом» та галузевим еталоном є використання цифрового архіву WikiArt.

Цей масив даних містить понад 80000 – 100000 високоякісних зображень творів мистецтва, зібраних від понад 1000 до 2000 художників, що хронологічно охоплюють період від епохи Раннього Відродження (XV століття) до сучасних мистецьких напрямків. Архітектура метаданих WikiArt дозволяє групувати картини за 27 різними стилістичними категоріями, 45 жанрами та національними школами.

Найголовнішим інженерним викликом датасету WikiArt є його глибока незбалансованість. Статистичний аналіз показує, що такі стилі, як імпресіонізм або реалізм, містять понад 10000 – 13000 зразків картин кожен, тоді як такі вузькі напрямки, як синтетичний кубізм або пуантилізм, можуть налічувати ледве більше сотні екземплярів. Тренування нейронної мережі на таких «сирих» даних призведе до того, що модель постійно передбачатиме лише мажоритарні класи задля мінімізації загальної помилки.

Для навчання було відібрано 15 художніх стилів, що представляють різні епохи: абстрактний експресіонізм, постімпресіонізм, імпресіонізм, романтизм, експресіонізм, фовізм, укійо-е, реалізм, модерн, поп-арт, пуантилізм, примітивізм, мінімалізм, бароко, кубізм, які зображені на рисунку 3.1 відповідно.

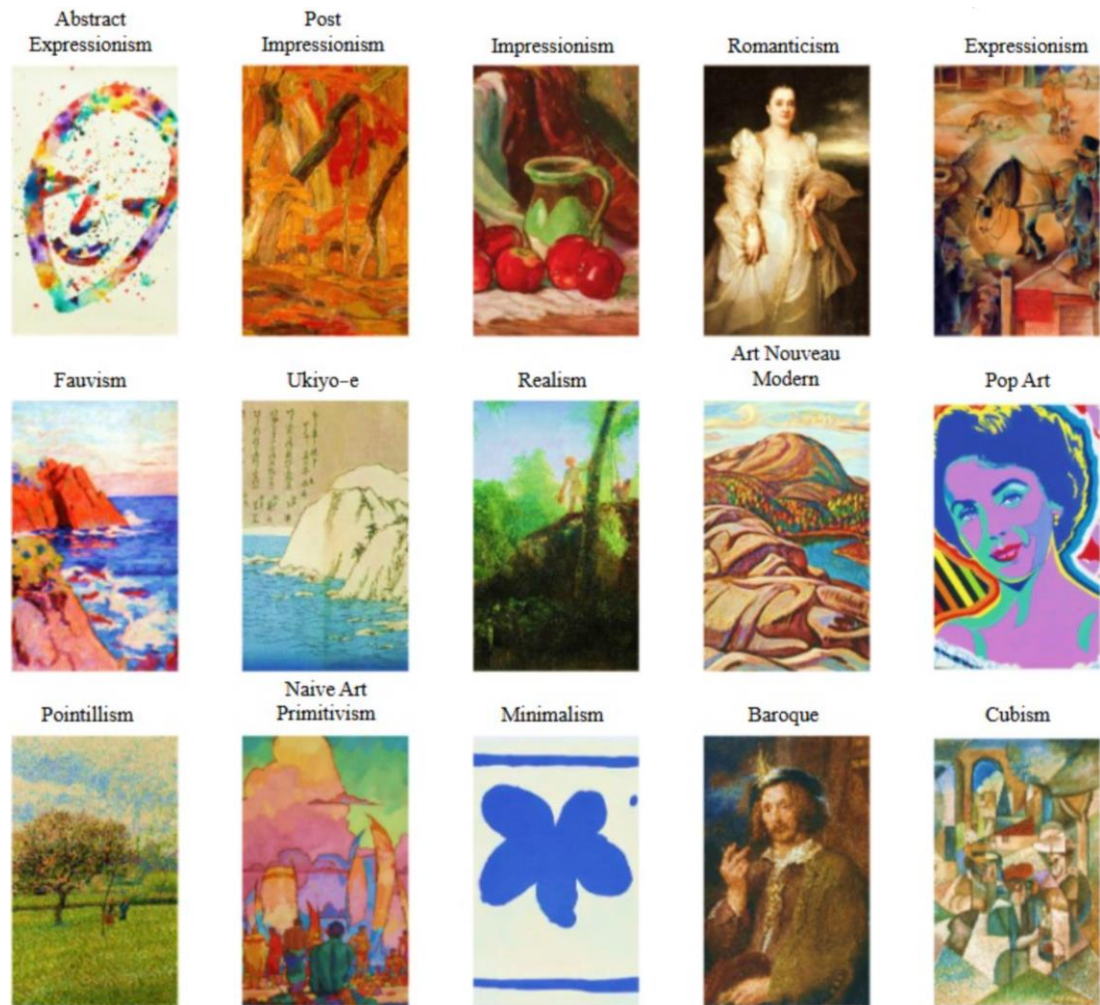


Рисунок 3.1 – Приклади картин з датасету WikiArt

Вихідний масив даних характеризується значною нерівномірністю розподілу зображень між класами, що є типовим для реальних мистецтвознавчих архівів. Кількісні показники вихідного розподілу зображень за стилями до моменту обробки наведено на рисунку 3.2.

Дані є суттєво незбалансованим. Така нерівномірність є фундаментальним викликом для навчання будь-якої нейронної мережі, оскільки без відповідних заходів модель буде схильна надавати перевагу мажоритарним класам, мінімізуючи загальну помилку за рахунок якості класифікації меншин.

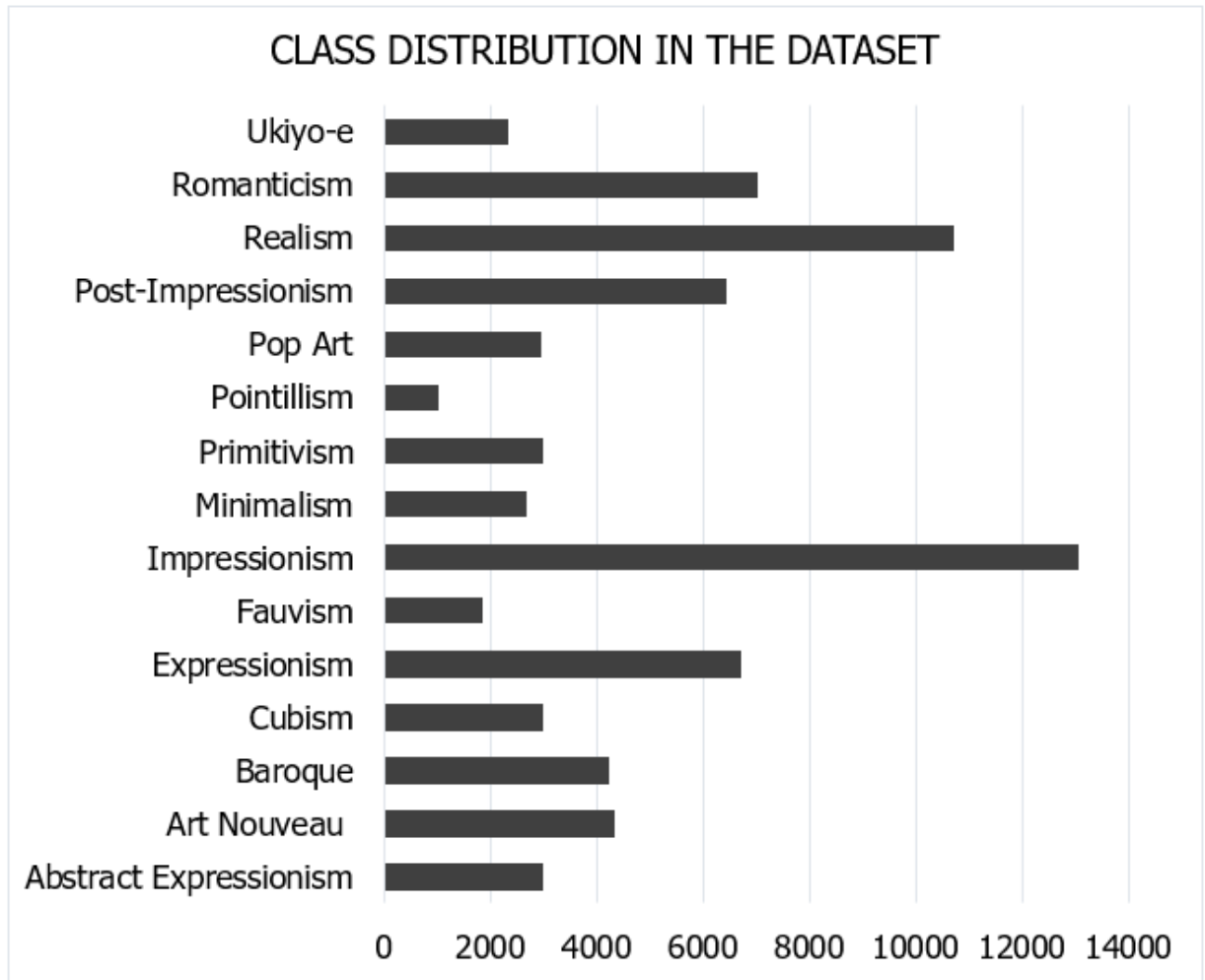


Рисунок 3.2 – Розподіл зображень по художніх стилях

Вирішення проблеми дисбалансу класів. Навчання класифікатора на незбалансованих даних призводить до зміщення моделі на користь домінуючих класів. Для вирішення цієї проблеми застосовується двоетапна стратегія.

1. Для класів, чисельність яких перевищувала попередньо визначену кількість одиниць, було виконано випадковий відбір репрезентативної підмножини. Це дозволило скоротити надмірні дані без втрати ключових ознак стилю.
2. Для менших класів, обсяг яких був недостатнім, використовується WGAN-GP. Даний метод дозволить

згенерувати синтетичні зображення, які за своїми візуальними характеристиками відповідають специфіці конкретних художніх стилів.

Аугментація зображень. Усі зображення проходять попередню обробку перед подачею до нейронної мережі. Традиційне алгоритмічне стиснення зображень великої роздільної здатності безповоротно знищує мікротекsturну інформацію, важливу для ідентифікації стилю. Для забезпечення балансу між обчислювальною складністю та збереженням стилістичних ознак, система імплементує приведення зображень до уніфікованого розміру: 128×128 пікселі для модуля генерації та 224×224 пікселі для класифікатора.

Для підвищення різноманітності навчальних прикладів та запобігання перенавчанню застосовується розширений набір трансформацій.

1. **Resize:** зображення масштабується безпосередньо до цільового розміру 128×128 для генератора та 224×224 для класифікатора.
2. **RandomHorizontalFlip:** горизонтальне віддзеркалення з імовірністю 0,5. Оскільки художній стиль інваріантний до дзеркального відображення, ця операція дозволяє подвоїти обсяг даних без внесення семантичних помилок.
3. **RandomRotation:** випадковий поворот у діапазоні $\pm 15^\circ$. Це забезпечує інваріантність до орієнтації композиції.
4. **ColorJitter:** випадкові зміни яскравості ($\pm 20\%$) та контрасту ($\pm 20\%$). Це моделює варіативність умов освітлення та передачі кольору під час фотографування картин у різних галереях.

Нормалізація. Для модуля WGAN-GP значення пікселів переводяться до діапазону $[-1, 1]$ за формулою (3.1).

$$x_{norm} = \frac{x - 0,5}{0.5}, \quad (3.1)$$

де $x \in [0,1]$ – початкове значення каналу. Це відповідає функції активації Tanh на виході генератора та забезпечує стабільність навчання.

Для класифікатора застосовується нормалізація згідно з параметрами моделі ImageNet: середнє значення [0.485; 0.456; 0.406] та стандартне відхилення [0.229; 0.224; 0.225] для каналів RGB відповідно. Для валідаційної та тестової вибірок використовуються лише детерміновані перетворення (Resize та нормалізація) для забезпечення об'єктивності оцінки.

3.2 Спеціалізований алгоритм запобігання mode collapse

Під час розробки системи балансування датасету WikiArt було виявлено фундаментальне протиріччя між стандартним підходом до запобігання mode collapse та реальними умовами задачі. Традиційні методи стабілізації навчання генеративно-змагальних мереж застосовуються однаково для всіх класів незалежно від їхніх характеристик. Однак датасет WikiArt характеризується суттєвою нерівномірністю розподілу – кількість зображень варіюється від кількох сотень до кількох тисяч на клас, що робить однакову стратегію захисту неоптимальною.

Застосування повного набору регуляризаційних методів до всіх класів без врахування цієї різниці є неефективним з двох причин. По-перше, для класів з великою кількістю зображень такий підхід є надлишковим. Захисні методи, що необхідні при малій вибірці лише штучно погіршують якість генерації для добре представлених класів. По-друге, для класів з малою кількістю зображень стандартна конфігурація може виявитись недостатньою без врахування специфіки конкретної навчальної вибірки.

Це спостереження стало основою для розробки оригінального підходу – спеціалізованого алгоритму запобігання mode collapse, що автоматично

визначає необхідний рівень захисту на основі двох незалежно обчислюваних характеристик кожного класу.

3.2.1 Двокритеріальна оцінка ризику

Ключова концепція запропонованого алгоритму полягає у введенні двокритеріальної оцінки ймовірності mode collapse. На відміну від наявних підходів, де рішення про застосування методів захисту приймається на основі одного параметра – як правило, лише кількості навчальних прикладів – запропонована система враховує два незалежних критерії: відносну кількість зображень та візуальну складність стилю.

Критерій 1 – Відносна кількість зображень. Перший критерій оцінює кількість реальних зображень класу відносно статистики всього датасету. Принципова відмінність від фіксованих меж полягає в тому, що критерій обчислюється динамічно на основі середнього значення за формулою (3.2) та стандартного відхилення за формулою (3.3) розподілу кількості зображень за всіма класами.

$$\mu = \frac{1}{K} \sum_{k=1}^K N_k, \quad (3.2)$$

$$\sigma = \sqrt{\frac{1}{K} \sum_{k=1}^K (N_k - \mu)^2}, \quad (3.3)$$

де N_k – кількість зображень класу k , K – загальна кількість класів, μ – середня кількість зображень по всіх класах датасету, σ – стандартне відхилення. Показники визначаються як:

- межа критичної вразливості: $\tau_{crit} = \mu - \sigma$;
- межа помірної вразливості: $\tau_{mod} = \mu$.

Такий підхід робить алгоритм незалежним від конкретного датасету – при застосуванні до іншого набору даних значення автоматично перераховуються відповідно до його розподілу без будь-яких змін у коді. Для датасету WikiArt з 15 класами ці показники набувають конкретних числових значень залежно від поточного розподілу класів після балансування. Визначення ймовірності за першим критерієм:

- $N_k < \tau_{\text{crit}}$: оцінка +2 (критична ймовірність за кількістю);
- $\tau_{\text{crit}} \leq N_k < \tau_{\text{mod}}$: оцінка +1 (помірна ймовірність);
- $N_k \geq \tau_{\text{mod}}$: оцінка 0 (низька).

Критерій 2 – Візуальна складність стилю. Другий критерій визначений у формулі (3.4) оцінює наскільки складним є художній стиль із погляду генеративної моделі. Складність стилю вимірюється через середню ймовірність дисперсії пікселів вибірки зображень класу.

$$C_k = \frac{1}{M} \sum_{i=1}^M \text{Var}(I_i), \quad (3.4)$$

де I_i – i -те зображення вибірки, M – розмір вибірки (30 зображень), $\text{Var}(\cdot)$ – дисперсія значень пікселів.

Стилі з деталізованими текстурами, складними кольоровими переходами та різноманітними композиціями характеризуються високою дисперсією значень пікселів. Стилі з простими геометричними формами та однотонними заливками демонструють низьку візуальну неоднорідність. Висока варіація означає, що дискримінатор очікує складніших та різноманітніших ознак від генератора – знайти простий «трюк», що постійно обманює такий дискримінатор значно важче.

Оскільки датасет WikiArt містить виключно високоякісні репродукції картин, абсолютні значення дисперсії пікселів для всіх стилів є апріорі високими. Тому фіксовані порогові значення не забезпечують диференціації між класами. Для вирішення цієї проблеми запропоновано адаптивний

підхід: пороги визначаються відносно статистики самого датасету як $\mu_C \pm \sigma_C$, де μ_C – середнє значення складності по всіх класах, σ_C – стандартне відхилення. Визначення ймовірності за другим критерієм:

- $C_k > \mu_C + \sigma_C$: оцінка +1 (складність вища за норму – генератору важче відтворити різноманіття ознак);
- $\mu_C - \sigma_C \leq C_k \leq \mu_C + \sigma_C$: оцінка 0 (середня складність);
- $C_k < \mu_C - \sigma_C$: оцінка -1 (складність нижча за норму – підвищена ймовірність mode collapse через простоту стилю).

Комбінована оцінка ризику. Фінальна оцінка ризику обчислюється за формулою (3.5) як сума оцінок за обома критеріями.

$$R_k = S_{count} + S_{complexity}, \quad (3.5)$$

де S_{count} – оцінка за першим критерієм, $S_{complexity}$ – оцінка за другим критерієм. Відповідність між комбінованою оцінкою та рівнем ризику:

- $R_k \geq 2$: максимальний показник нестабільності;
- $R_k = 1$: помірний показник нестабільності;
- $R_k \leq 0$: низький показник нестабільності.

Таблицю всіх можливих комбінацій критеріїв та відповідних показників нестабільності наведено у таблиці 3.1.

Поєднання двох критеріїв дозволяє системі формувати більш точний профіль ризику для кожного класу індивідуально, що є принциповою відмінністю від уніфікованих стратегій навчання GAN. Результуюча комбінована оцінка визначає не лише факт застосування захисних механізмів, але й їх конкретний склад та інтенсивність – тому реалізується адаптивна, а не фіксована стратегія протидії mode collapse.

Таблиця 3.1 – Аналіз умов виникнення mode collapse у задачі генерації художніх стилів

<i>Обсяг навчальної вибірки</i>	<i>Складність стилю</i>	<i>Комбінована оцінка</i>	<i>Показник нестабільності</i>	<i>Характер вразливості</i>
<i>Малий</i> ($< \mu - \sigma$)	Висока	+3	Максимальний	Дискримінатор швидко перенавчається; генератор не здатен відтворити складні патерни
<i>Малий</i> ($< \mu - \sigma$)	Середня	+2	Критичний	Дискримінатор перенавчається; обмежений сигнал для генератора
<i>Малий</i> ($< \mu - \sigma$)	Низька	+1	Помірний	Дискримінатор перенавчається але генератор легко знаходить базові ознаки
<i>Середній</i> ($\mu - \sigma - \mu$)	Висока	+2	Критичний	Частковий ризик перенавчання при складних ознаках
<i>Середній</i> ($\mu - \sigma - \mu$)	Середня	+1	Помірний	Збалансований ризик; collapse можливий при несприятливій ініціалізації
<i>Середній</i> ($\mu - \sigma - \mu$)	Низька	0	Низький	Достатньо даних для простого стилю
<i>Високий</i> ($> \mu$)	Висока	+1	Помірний	Різноманітність вибірки компенсує складність стилю
<i>Високий</i> ($> \mu$)	Середня	0	Низький	Стандартне навчання без додаткового захисту
<i>Високий</i> ($> \mu$)	Низька	-1	Мінімальний	Collapse практично неможливий

3.2.2 Конфігурація захисних методів

На основі комбінованої оцінки ризику алгоритм автоматично визначає конфігурацію захисних методів. Система оперує п'ятьма методами – двома постійними та трьома адаптивними. Постійні методи застосовуються для всіх класів незалежно від рівня ризику оскільки є частиною базової архітектури WGAN-GP та забезпечують мінімально необхідну стабільність навчання. Gradient Penalty математично обмежує нахил функції дискримінатора, забезпечуючи стабільний градієнтний сигнал. Gradient Clipping запобігає різким стрибкам ваг мереж при нестабільних градієнтах.

Адаптивні методи активуються лише за наявності відповідного показника нестабільності. Dropout у дискримінаторі обмежує здатність запам'ятовувати малу вибірку – активується при помірному та критичному ризику. Гаусівський шум з амплітудою розмиває межу між реальними та синтетичними зображеннями – активується лише при критичному ризику. Diversity Loss явно штрафує генератор за генерацію однотипних виходів – активується при помірному та критичному ризику з різними коефіцієнтами α . Параметр оновлення дискримінатора, n_critic , також адаптується: при максимальному показнику зменшується до 2, що обмежує перевагу дискримінатора, при низькому – відповідає стандартному значенню 5. Повну конфігурацію захисних методів для кожного рівня вразливості наведено у таблиці 3.2.

Таблиця 3.2 – Конфігурація методів залежно від рівня вразливості

Метод	Максимальний	Помірний	Низький
Gradient Penalty ($\lambda = 10$)	+	+	+
Gradient Clipping (норма = 1.0)	+	+	+
Dropout у дискримінаторі ($p = 0.25$)	+	+	–
Гаусівський шум ($\sigma = 0.05$)	+	–	–
Diversity Loss	+	+	–

Статичний показник нестабільності на основі характеристик класу є необхідною, але недостатньою умовою надійного захисту від mode collapse. Навіть при правильно визначеній початковій конфігурації, collapse може виникнути у будь-який момент навчання через несприятливу ініціалізацію ваг або нестабільну динаміку градієнтів. Для вирішення цієї проблеми запропонований алгоритм доповнено системою динамічного моніторингу. Кожні 10 епох система аналізує динаміку функцій втрат у ковзному вікні останніх п'яти епох та обчислює три індикатори відхилення від нормального навчання. Індикатор застрягання генератора виявляється коли стандартне відхилення функції втрат генератора за останні 5 епох менше 0.5 при одночасно великому абсолютному значенні. Це означає що генератор перестав покращуватись. Індикатор домінування дискримінатора виявляється, коли функція втрат стабільно зменшується протягом п'яти послідовних епох. Це означає, що дискримінатор навчається значно швидше за генератор. Індикатор загального дисбалансу виявляється, коли різниця між середніми значеннями функцій втрат генератора та критика перевищує порогове значення 20. При виявленні домінування дискримінатора коефіцієнт Diversity Loss збільшується на 0.1 до максимального значення 1.0. При виявленні застрягання генератора параметр для оновлень дискримінатора зменшується на 1 до мінімального значення 1. Таким чином алгоритм здатен реагувати на ранні ознаки нестабільності ще до повного колапсу різноманітності виходів генератора.

3.2.3 Результати застосування алгоритму

Запропонований алгоритм було застосовано до всіх класів датасету WikiArt, що потребували балансування. Перед початком балансування

система автоматично обчислила статистику розподілу датасету та встановила адаптивні межі нестабільності. Більшість класів отримала низький рівень нестабільності та мінімальний набір методів запобігання – лише Gradient Penalty та Gradient Clipping. Для кількох класів з меншою кількістю зображень система активувала Dropout та Diversity Loss. Лише один клас отримав максимальний рівень ризику з повним набором захисних методів включаючи гаусівський шум.

Принципово важливим результатом є те що для найбільших класів система автоматично вимкнула всі адаптивні методи захисту. Це дозволило цим класам навчатись у стандартному режимі WGAN-GP без штучних обмежень, що позитивно вплинуло на якість генерації. Динамічний моніторинг протягом навчання зафіксував окремі епізоди активації індикаторів для класів з помірним рівнем нестабільності. У кожному з таких випадків система автоматично скоригувала параметри, що дозволило продовжити навчання без повної втрати різноманітності виходів генератора. Тому запропонований алгоритм реалізує принцип мінімально необхідної регуляризації – кожен клас отримує рівно такий захист, який обумовлений його об'єктивними характеристиками. Це відрізняє розроблену систему від наявних підходів, де всі класи обробляються однаково, та є оригінальним авторським внеском, що підвищує ефективність балансування для проблемних класів.

3.4 Обґрунтування вибору ResNet-18

Для задачі класифікації художніх стилів обрано архітектуру ResNet-18 з попередньо навченими вагами на датасеті ImageNet. Залишкові з'єднання ResNet вирішують проблему зникання градієнтів при глибокому навчанні та

забезпечують ефективне навчання навіть при обмеженому датасеті. Вибір ResNet-18 обумовлений співвідношенням між обчислювальними ресурсами та якістю: модель має 11 мільйонів параметрів та демонструє достатню виразність для задачі класифікації художніх стилів. У рамках дослідження була спроба використати глибший варіант архітектури – ResNet-50, але показники не продемонстрували ефективніших результатів. Також для кращого розуміння принципу вибору моделі – було проведено порівняльний аналіз різних архітектур (див. табл. 3.3).

Таблиця 3.3 – Характеристики досліджуваних архітектур

<i>Модель</i>	<i>Шарів</i>	<i>Параметрів</i>	<i>Розмір</i>	<i>Архітектура</i>
CNN baseline	9	~5 М	~20 МБ	Послідовні згортки
ResNet-18	18	11.7 М	45 МБ	Залишкові блоки
ResNet-50	50	25.6 М	96 МБ	Залишкові блоки
EfficientNet-B0	82	5.3 М	21 МБ	Комбіноване масштабування
VGG16	16	138 М	515 МБ	Лінійна
InceptionV3	48	23.9 М	89 МБ	Модулі Inception

Незважаючи на те що у порівняльному аналізі досліджувались архітектури ResNet-18, ResNet-50, EfficientNet-B0, VGG16, InceptionV3 та CNN Baseline, основною моделлю системи обрано саме ResNet-18. Цей вибір обумовлений не лише показниками точності, а й сукупністю практичних міркувань, що є критичними в умовах обмежених обчислювальних ресурсів.

ResNet-18 демонструє оптимальне співвідношення між точністю класифікації та обчислювальними витратами. При 11.7 мільйонах параметрів та розмірі 45 МБ модель забезпечує точність 72% — лише на 1% нижчу за ResNet-50 з 25.6 мільйонами параметрів (див. табл. 3.4). VGG16 при 138 мільйонах параметрів та розмірі 515 МБ займає непропорційно великий обсяг відеопам'яті, що є критичним обмеженням для GPU. EfficientNet-B0 при меншій кількості параметрів забезпечує порівнянну точність, однак складніший процес налаштування гіперпараметрів. InceptionV3 вимагає вхідний розмір 299×299 , що збільшує споживання пам'яті без суттєвого

покращення результату для цієї задачі. Отже, ResNet-18 є найбільш збалансованим вибором для задачі класифікації художніх стилів в умовах наявного апаратного забезпечення.

Таблиця 3.4 – Таблиця результатів усіх моделей

Модель	Accuracy/%	F1/%	Presicion/%	Recall/%	AUC
CNN baseline	45	0.46	0.49	0.47	0.889
ResNet-18	72	0.72	0.72	0.72	0.961
ResNet-50	73	0.73	0.72	0.73	0.963
EfficientNet-B0	74	0.74	0.72	0.74	0.974
VGG16	69	0.69	0.70	0.70	0.962
InceptionV3	74	0.73	0.73	0.72	0.966

На рисунку наведено лінійний графік, який відображає результати порівняльного аналізу метрики AUC для шести архітектур нейронних мереж. Візуалізація демонструє суттєву перевагу глибоких моделей над базовою архітектурою CNN у розв'язанні задачі класифікації художніх стилів. Усі глибокі мережі показують високу та стабільну здатність до узагальнення зі значеннями AUC понад 0.96, причому найвищу ефективність серед досліджуваних алгоритмів забезпечує архітектура EfficientNet-B0.

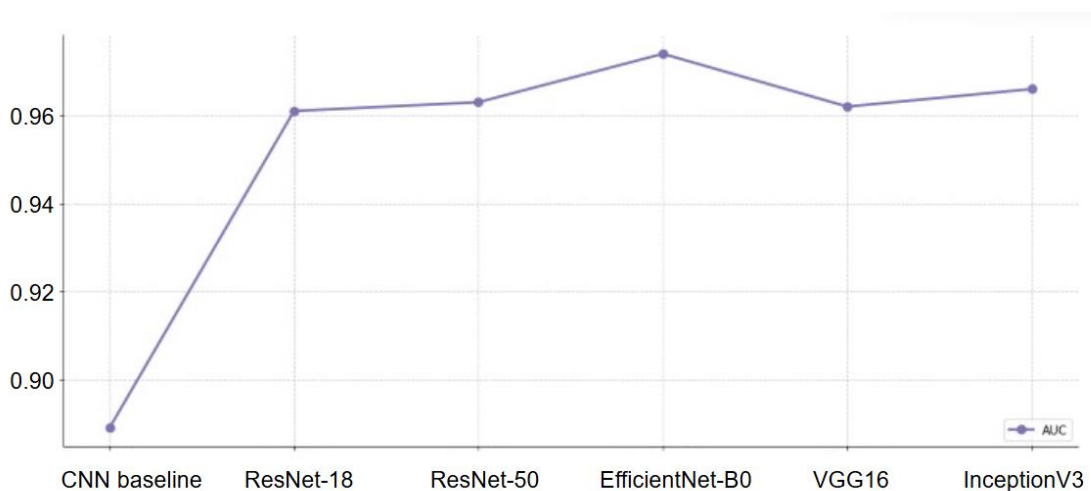


Рисунок 3.3 – Значення показника AUC для досліджуваних архітектур

У цій роботі застосовується стратегія донавчання усіх параметрів моделі з малим темпом навчання. На відміну від feature extraction де заморожуються всі шари крім фінального, fine-tuning дозволяє моделі адаптувати представлення під специфіку художніх зображень зберігаючи при цьому загальні візуальні концепти. Два перших залишкових блоки ResNet-18 заморожуються повністю оскільки містять низькорівневі детектори країв та текстур, що є універсальними та не потребують адаптації. Глибші шари, що кодують більш специфічні ознаки навчаються з темпом $lr = 3 \times 10^{-5}$, що є на порядок меншим за стандартний темп навчання з нуля.

Стандартний фінальний шар ResNet-18 розрахований на 1000 класів датасету ImageNet замінюється новим блоком адаптованим до 15 художніх стилів. Новий класифікаційний блок складається з шару Dropout, що виконує функцію регуляризації запобігаючи перенавчанню на відносно невеликому датасеті, та лінійного шару що відображає 512-вимірний вихід передостаннього шару ResNet-18 у простір з 15 класами. Застосування Dropout безпосередньо перед класифікаційним шаром обумовлено тим що саме на цьому рівні мережа формує найбільш специфічні для задачі представлення, що є найбільш схильними до перенавчання. Коефіцієнт обраний емпірично як компроміс між достатньою регуляризацією та збереженням інформативності фінальних представлень.

Порівняння результатів класифікатора до та після балансування.

Для кількісної оцінки ефективності запропонованої стратегії балансування проведено порівняльний експеримент: класифікатор ResNet-18 навчався на оригінальному незбалансованому датасеті WikiArt без застосування WGAN-GP. Результати порівняння наведено у таблиці 3.5. Застосування балансування датасету забезпечило підвищення загальної точності класифікації з 61% до 72%. Найбільш суттєве покращення спостерігається саме для міноритарних класів: фовізм (+0.41), поп-арт (+0.34), мінімалізм

(+0.25) та пуантилізм (+0.23) – тобто тих стилів, для яких WGAN-GP генерував найбільшу кількість синтетичних зображень.

Таблиця 3.5 – Порівняння підходів до балансування

<i>Метрика</i>	<i>БЕЗ балансування</i>	<i>З балансуванням</i>
Accuracy	61%	72%
Macro F1	0.62	0.72
Weighted F1	0.60	0.72

Функція втрат на тренувальній вибірці демонструє стабільне монотонне спадання, що свідчить про коректний перебіг оптимізаційного процесу. Точність на валідаційній вибірці стрімко зростала протягом перших п'яти епох та надалі стабілізувалась у діапазоні 71–72%, що підтверджує відсутність критичного перенавчання (рис. 3.6).

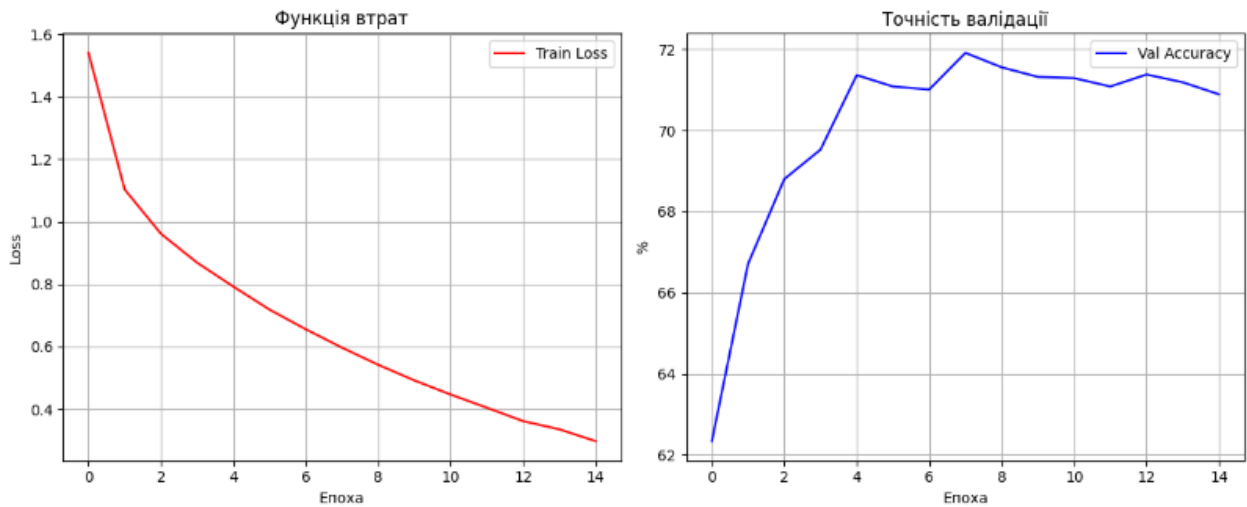


Рисунок 3.4 – Динаміка функції втрат та точності класифікатора під час навчання

Висновки до розділу 3

У третьому розділі представлено опис запропонованого підходу до проєктування інтелектуальної системи. Було реалізовано адаптивну стратегію балансування даних, яка поєднує метод випадкової недодискретизації для переповнених класів та синтез нових зображень. Це дозволило сформувати рівномірний та репрезентативний набір даних для 15 різних художніх стилів. Аналіз динаміки функцій втрат та точності на валідаційній вибірці підтвердив відсутність перенавчання та високу узагальнювальну здатність моделі. Важливою частиною розділу став аналіз впливу якості генерації на фінальний результат. Також було окреслено перспективи подальшого розвитку системи, що включають перехід до глибших архітектур та розширення функціональності до мультизадачної класифікації, що дозволить одночасно визначати стиль, жанр, автора та історичну епоху твору.

РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

4.1 Вибір інструментальних засобів розробки

Розробка інтелектуальної системи класифікації художніх стилів вимагає ретельного підбору інструментального стеку, який забезпечив би як обчислювальну ефективність навчання глибоких нейронних мереж, так і зручність реалізації складних архітектурних компонентів.

Розробка та експериментальне дослідження інтелектуальної системи проводились на апаратному забезпеченні з певними характеристиками. Обчислення на графічному процесорі здійснювались за допомогою дискретної відеокарти NVIDIA GeForce GTX 1660 Ti з обсягом відеопам'яті 6 ГБ GDDR6. Оперативна пам'ять системи – не менше 16 ГБ, дисковий простір – не менше 20 ГБ для розміщення датасету WikiArt та синтетично згенерованих зображень.

Обсяг відеопам'яті 6 ГБ визначив ключові архітектурні обмеження системи: розмір батчу встановлено рівним 32, розмір зображень для WGAN-GP – 128×128 пікселі, що є компромісом між якістю генерації та доступними обчислювальними ресурсами.

Програмне середовище базується на операційній системі Windows з інтерпретатором Python 3.10, драйверами NVIDIA CUDA 11.8 для GPU-прискорення та середовищем розробки Visual Studio Code з розширенням Jupyter Notebook.

Мовою реалізації обрано Python завдяки широкій екосистемі бібліотек глибокого навчання. Основним фреймворком є PyTorch, що забезпечує динамічний обчислювальний граф, зручний інтерфейс CUDA та гнучкість реалізації кастомних функцій втрат. Порівняно з TensorFlow, PyTorch демонструє вищу зручність налагодження та більшу поширеність у сучасних дослідженнях генеративних моделей.

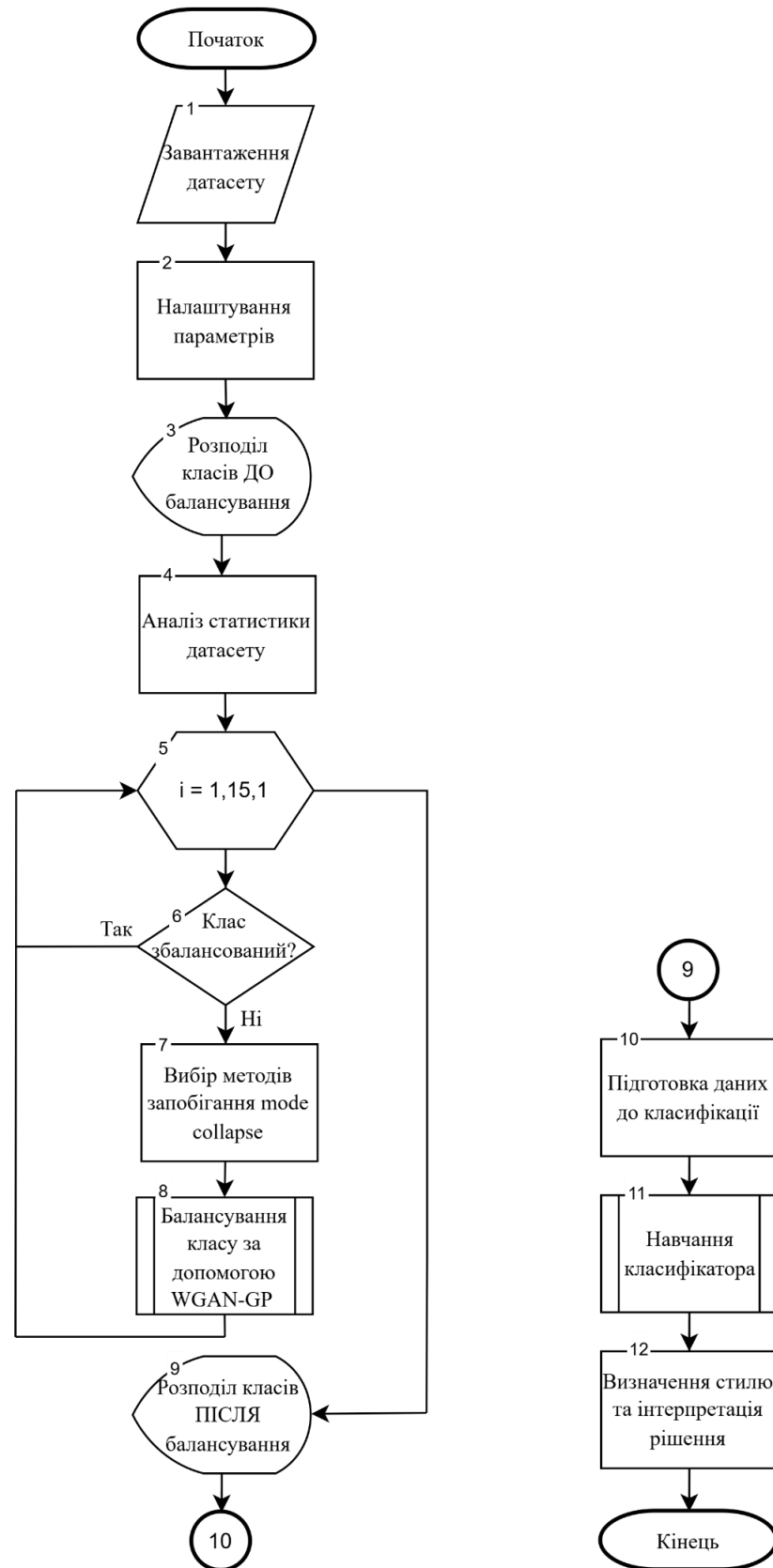
4.2 Архітектура програмного забезпечення

Програмний продукт реалізовано у вигляді модульного скрипту з розподілом функціональних блоків, що взаємодіють між собою через чітко визначені інтерфейси. Загальна архітектура системи організована за принципом послідовного пайплайну, де вихід кожного модуля є входом наступного. Загальний потік виконання системи наведено на блок-схемі 4.1.

Опис основних модулів системи наведено у таблиці 4.1.

Таблиця 4.1 – Функціональні модулі системи

<i>Модуль</i>	<i>Призначення</i>
Generator	Генерування синтетичних зображень на основі вхідного латентного вектора
Discriminator	Оцінювання реалістичності зображень (виконує функцію критика в архітектурі WGAN-GP)
gradient_penalty	Обчислення градієнтного штрафу для забезпечення умови ліпшицевої неперервності дискримінатора
balance_class_with_wgan	Керування процесом навчання WGAN-GP та генеруванням синтетичних даних для усунення класового дисбалансу
predict_style_with_uncertainty	Класифікація стилю зображення з одночасним обчисленням індексу невизначеності
visualize_gradcam	Побудова теплових карт активації для візуального пояснення рішень класифікатора
EarlyStopping	Моніторинг метрик на валідаційній вибірці та зупинка навчання для запобігання перенавчанню моделі
FilteredArtDataset	Завантаження, фільтрація та формування збалансованого набору даних із застосуванням методів випадкового відбору
train_classifier	Організація повного циклу навчання, валідації та тестування згорткової нейронної мережі для задачі класифікації



Блок-схема 4.1 — Блок-схема загального потоку виконання системи

Принцип побудови та модифікація архітектури WGAN-GP.

Архітектура генератора реалізована як повністю згорткова мережа-декодер, що перетворює латентний вектор розмірністю 128 на кольорове зображення через п'ять послідовних блоків транспонованих згортки. Детальну архітектуру генератора зображено на рисунку 4.1.

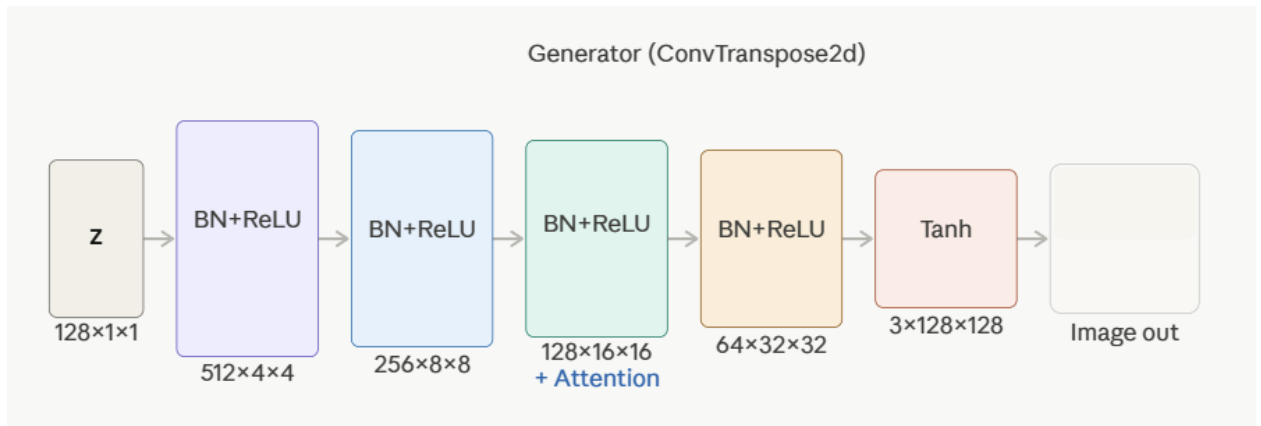


Рисунок 4.1 – Архітектура генератора²

Після кожного шару (крім останнього) застосовується BatchNormalization та функція активації ReLU. Фінальний шар використовує активацію Tanh для нормалізації виходу до діапазону $[-1, 1]$. До архітектури генератора інтегровано механізм Self-Attention між четвертим та п'ятим блоками транспонованих згортки. На відміну від стандартних згорткових шарів, що аналізують лише локальні просторові околиці, шар самоуваги обчислює залежності між усіма просторовими позиціями карти ознак одночасно. Це дозволяє генератору враховувати глобальні стилістичні закономірності при синтезі зображень – наприклад рівномірний розподіл текстурних елементів, характерний для ряду художніх стилів.

Архітектура дискримінатора реалізована як згорткова мережа, що отримує зображення 128×128 та повертає скалярну оцінку без обмеження

²Зображення створено за допомогою чат-боту з генеративним штучним інтелектом Gemini.

діапазону. Для дотримання умови Ліпшиця у перших двох блоках застосовується InstanceNormalization. На відміну від класичної архітектури WGAN-GP де дискримінатор не містить шарів регуляризації, у цій роботі до перших двох згорткових блоків дискримінатора додано шари Dropout2d. Це рішення є частиною розробленого алгоритму адаптивного запобігання mode collapse – при критичному та помірному рівнях ризику Dropout обмежує здатність дискримінатора запам'ятовувати малі вибірки реальних зображень. Це підтримує конкурентний баланс між мережами протягом усього процесу навчання. Детальну архітектуру проілюстровано на рисунку 4.2.

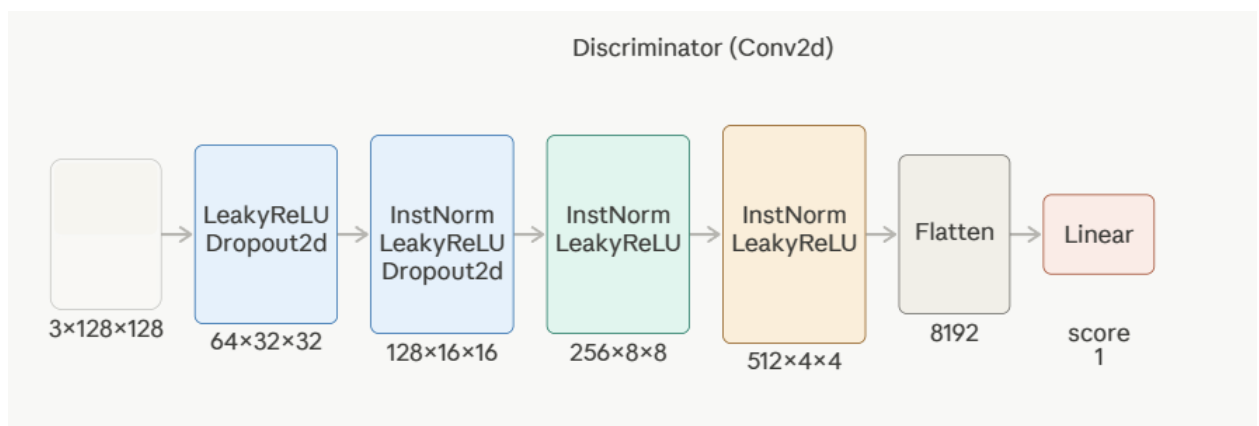


Рисунок 4.2 – Архітектура дискримінатора³

Після завершення навчання генератор переводиться у режим оцінювання та послідовно генерує необхідну кількість синтетичних зображень. Кожне зображення денормалізується з діапазону $[-1, 1]$ до $[0, 1]$, затискається операцією clamp для запобігання виходу за межі допустимих значень та зберігається у відповідному для зображення форматі безпосередньо до директорії конкретного художнього стилю. Усі шари використовують LeakyReLU як функцію активації.

³Зображення створено за допомогою чат-боту з генеративним штучним інтелектом Gemini.

Повний лістинг програмного коду розробленої інтелектуальної системи наведено у додатку А.

Гіперпараметри навчання WGAN-GP. Навчання генеративної моделі здійснювалось з використанням оптимізатора Adam з параметрами що відповідають стандартним рекомендаціям для WGAN-GP. Відсутність імпульсу першого моменту ($\beta^1 = 0.0$) забезпечує стабільність змагального навчання, тоді як асиметрія темпів навчання між генератором та дискримінатором компенсує різницю у частоті їх оновлень. Коефіцієнт градієнтного штрафу (10) та кількість ітерацій критика на одне оновлення генератора ($n_{critic} = 2$) обрані відповідно до оригінальної специфікації WGAN-GP. Проте останній параметр може динамічно коригуватися механізмом адаптивного моніторингу. Цільова кількість зображень визначається адаптивно формулою (4.1).

$$N_{target} = \min(3000, 2 \cdot N_{real}), \quad (4.1)$$

де N_{real} – поточна кількість реальних зображень класу. Це обмежує частку синтетичних зображень до 50% від загальної кількості зображень класу.

Гіперпараметри навчання класифікатора. Класифікатор ResNet-18 навчається методом fine-tuning усіх незаморожених параметрів з використанням оптимізатора Adam та L2-регуляризації ($weight\ decay = 10^{-4}$). Малий темп навчання (3×10^{-5}) обраний свідомо для збереження корисних представлень засвоєних під час попереднього навчання на ImageNet. Нормалізація вхідних зображень розміром 224×224 пікселів виконується за статистиками ImageNet, що забезпечує узгодженість вхідного розподілу з умовами попереднього навчання. Функцією втрат є категоріальна крос-ентропія. Для запобігання перенавчанню реалізовано механізм раннього зупинення, що моніторить точність на валідаційній вибірці та зберігає

найкращу версію моделі протягом усього процесу навчання. Повний перелік гіперпараметрів системи записано до таблиці 4.2.

Таблиця 4.2 – Основні гіперпараметри системи

<i>Параметр</i>	<i>Генератор</i>	<i>Класифікатор</i>
Оптимізатор	Adam	Adam
Темп навчання G	10^{-4}	3×10^{-5}
Темп навчання D	3×10^{-5}	—
β_1, β_2	0.0, 0.9	0.9, 0.999
Weight decay	—	10^{-4}
Розмір батчу	16	32
Кількість епох	100	30
Розмір зображення	128×128	224×224
Розмірність латентного простору	128	—
Коефіцієнт GP	10	—
n_{critic}	2	—
Коефіцієнт Diversity Loss	0.5	—
Dropout	0.25	0.4
Early Stopping patience	—	7

4.3 Метрики оцінки якості класифікатора

Для комплексної оцінки якості розробленої системи класифікації художніх стилів використовується набір метрик, кожна з яких характеризує окремий аспект ефективності моделі.

Введемо стандартні позначення: TP (True Positive) – кількість коректно розпізнаних зразків класу, FP (False Positive) – кількість хибно віднесених до класу, FN (False Negative) – кількість пропущених зразків класу, TN (True Negative) – кількість коректно відхилених зразків.

Перелік реалізованих метрик оцінки якості наведено у таблиці 4.3.

Таблиця 4.3 – Метрики оцінки якості класифікатора

Метрика	Формула	Призначення
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Загальна частка коректних передбачень
Precision	$\frac{TP}{TP + FP}$	Частка дійсно позитивних зразків серед усіх, які модель класифікувала як позитивні
Recall	$\frac{TP}{TP + FN}$	Частка знайдених зразків класу серед усіх реальних зразків
F1-score	$\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$	Гармонійне середнє між Precision та Recall, що забезпечує збалансовану оцінку
Macro F1-score	$\frac{1}{K} \sum_{k=1}^K F1_k,$ де $K = 15$ – кількість художніх стилів	Обчислює середнє арифметичне F1 для всіх класів, оцінюючи кожен клас як рівнозначний
Weighted F1	$\frac{\sum_{k=1}^K n_k \cdot F1_k}{\sum_{k=1}^K n_k},$ де n_k – кількість зразків класу k у тестовій вибірці	Обчислює F1 для кожного класу, але бере зважене середнє пропорційно до кількості зразків
AUC-ROC	$\int_0^1 TPR(FPR) d(FPR)$	Характеризує здатність моделі розрізняти зразки класу від усіх інших

4.5 Результати роботи запропонованого підходу

Після застосування адаптивної стратегії балансування на основі WGAN-GP ефективний розподіл класів для навчання набув більш рівномірного характеру.

Процес навчання генеративної моделі відрізнявся залежно від обсягу реальних зображень у класі. Для класів з достатньою кількістю даних спостерігалась стабільна динаміка функцій втрат генератора та дискримінатора з поступовою конвергенцією, що свідчить про ефективне засвоєння стилістичних ознак без ознак mode collapse. Для класів з обмеженою вибіркою, динаміка навчання виявилась більш нестабільною, що є очікуваним явищем при роботі з малими датасетами. Приклади генерації певних стилів зображено на рисунку 4.3. Застосований комплекс методів запобігання mode collapse дозволив утримати процес навчання в прийнятних межах та забезпечити генерацію стилістично різноманітних синтетичних зображень.

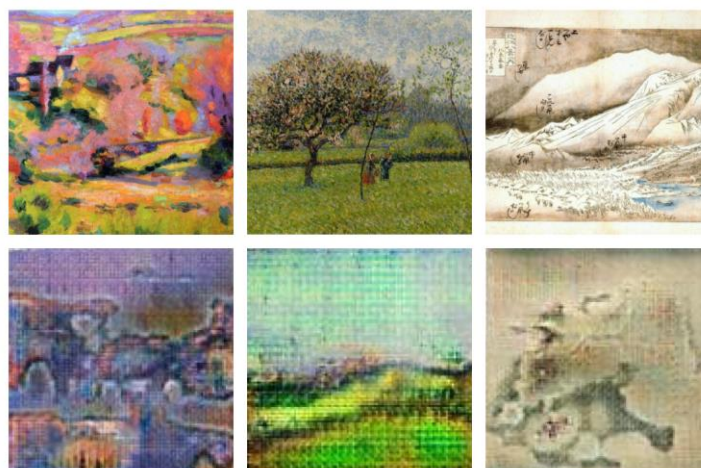


Рисунок 4.3 – Результат генерації картин у стилі фовізм, пуантилізм та укійо-е

Синтетичні зображення не претендують на фотореалістичність через обмежений розмір. Їхня роль полягає у статистичному розширенні простору ознак міноритарних класів, що підтверджується покращенням точності класифікатора з 61% до 72%. Модель демонструє помірну та збалансовану ефективність із загальними показниками (Precision, Recall, F1-score). Майже однакові значення макро- та зваженого усереднення (0.71 та 0.72) свідчать про те, що класифікатор працює стабільно і не має сильного перекосу в бік розпізнавання лише більших класів.

Детальні показники якості для кожного стилю наведено у таблиці 4.4.

Таблиця 4.4 – Показники якості класифікації за художніми стилями

<i>Стиль</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
Abstract Expressionism	0.72	0.75	0.73
Art Nouveau Modern	0.60	0.63	0.62
Baroque	0.75	0.81	0.78
Cubism	0.73	0.81	0.77
Expressionism	0.51	0.48	0.49
Fauvism	0.85	0.82	0.84
Impressionism	0.58	0.61	0.59
Minimalism	0.91	0.93	0.92
Naive Art Primitivism	0.76	0.68	0.72
Pointillism	0.98	0.93	0.96
Pop Art	0.82	0.79	0.80
Post Impressionism	0.52	0.48	0.50
Realism	0.47	0.55	0.50
Romanticism	0.67	0.55	0.60
Ukiyo e	0.96	0.97	0.96

Візуалізація простору ознак та матриця помилок. Для оцінки якості роботи інтелектуальної системи та аналізу її здатності розрізняти художні стилі використано два методи: візуалізацію простору ознак методом t-SNE на рисунку 4.4 та матрицю помилок на рисунку 4.5.

Графік t-SNE відображає багатовимірний простір ознак передостаннього шару нейромережі ResNet-18 у двовимірному вигляді. Кожна точка відповідає окремому зображенню з тестової вибірки. Він призначений для якісного аналізу здатності моделі групувати картини за

їхніми візуальними характеристиками. Окремі скупчення точок свідчать про чітке виділення унікальних ознак стилю, наприклад, для мінімалізму. Змішані області вказують на візуальну подібність стилів і складність їх розділення для алгоритму, наприклад, імпресіонізм та реалізм.

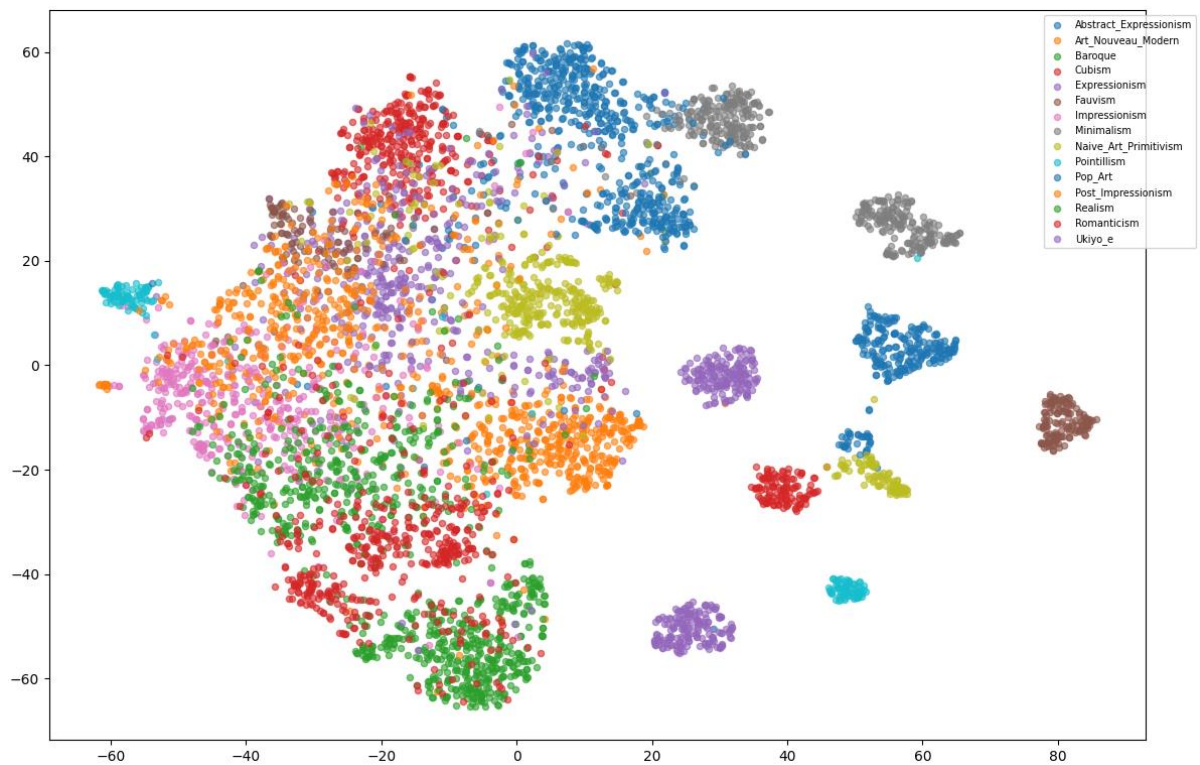


Рисунок 4.4 – Візуалізація простору ознак

Матриця помилок демонструє кількісні результати класифікації на тестовій вибірці. У рядках вказані справжні класи зображень, у стовпцях – класи, які передбачила модель. Вона призначена для детального кількісного аналізу точності моделі та виявлення закономірностей у хибних класифікаціях. Значення на головній діагоналі показують кількість правильних відповідей. Значення поза діагоналлю вказують на кількість помилок. Матриця чисельно підтверджує візуальні дані графіка t-SNE. Найвищі показники помилок фіксуються саме між тими стилями, чії кластери перетинаються у просторі ознак.

Спільне використання цих методів забезпечує комплексне оцінювання системи. t-SNE пояснює логіку формування ознак моделлю, а матриця помилок надає точні метрики її ефективності.

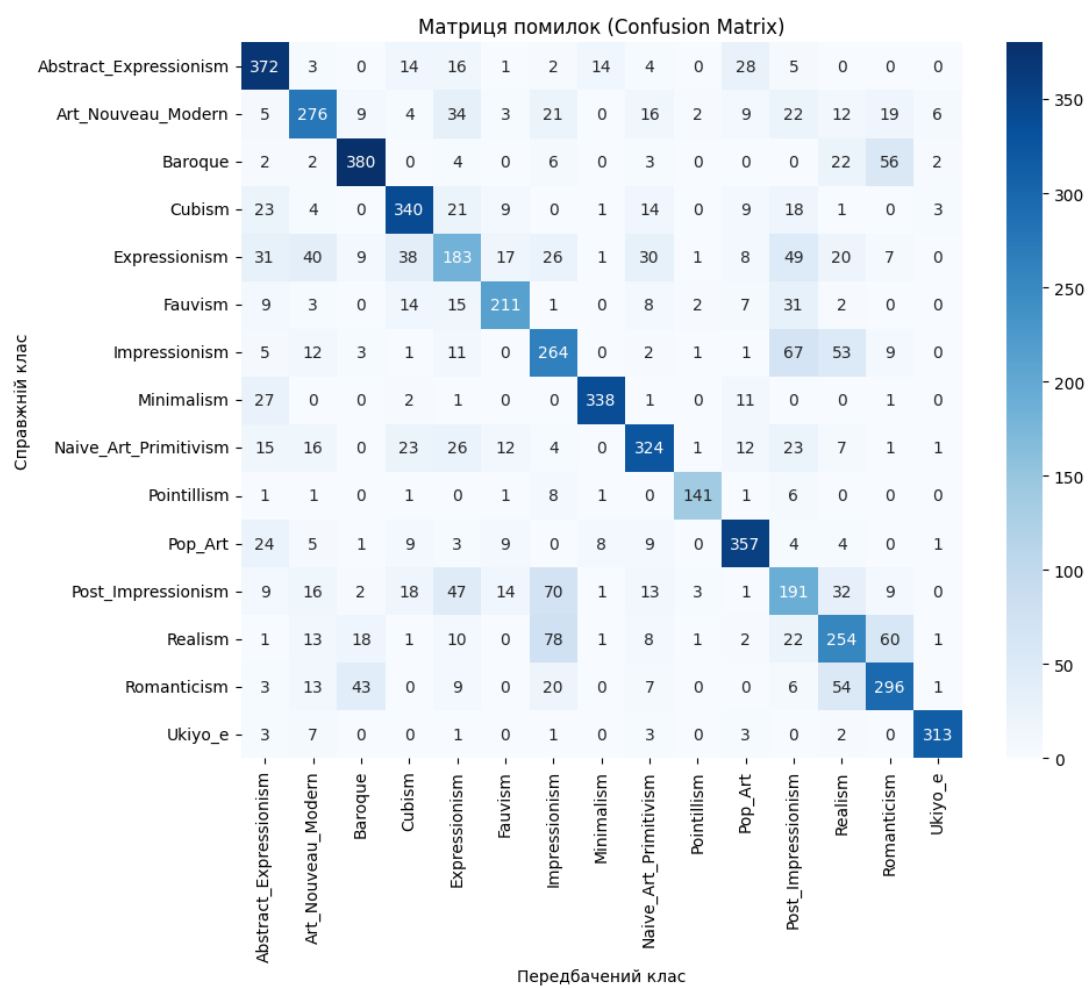


Рисунок 4.5 – Матриця помилок класифікації художніх стилів

Обґрунтування імовірнісного підходу та оцінки невизначеності.
Класифікація творів живопису є складним завданням для систем комп'ютерного зору через глибоко суб'єктивну та безперервну природу мистецтва. Перехід від одного стилю до іншого, наприклад, від імпресіонізму до постімпресіонізму, супроводжувався поступовою еволюцією технік. На одному полотні часто поєднуються ознаки одразу кількох шкіл. Тому використання класичної жорсткої класифікації, яка видає лише один

безальтернативний клас із найвищою ймовірністю, є методично неповним і не відображає реальної картини мистецтвознавчого аналізу.

Для вирішення цієї проблеми у розробленій інтелектуальній системі впроваджено модуль оцінки невизначеності та виведення топ-3 найімовірніших стилів. Такий підхід дозволяє системі не приховувати власні сумніви, а відкрито демонструвати розподіл ймовірностей. Розрахунок загального індексу невизначеності (ентропії) допомагає класифікувати результати за зонами впевненості та відфільтровувати складні, перехідні або нетипові випадки.

Практичне тестування модуля підтвердило високу ефективність такого підходу. Під час аналізу типових робіт система демонструє однозначні результати. Наприклад, для пастельної роботи Клод Моне «Дорога до Ветеля» модель зафіксувала показник упевненості на рівні 96.3% для імпресіонізму, виділивши постімпресіонізм як найближчу альтернативу з часткою лише 3.6% (рисунок 4.6). Індекс невизначеності становить 0,125, це свідчить про те, що архітектура безпомилково ідентифікувала легку текстуру та динаміку простору, притаманні саме цій техніці та віднесла результат до зони високої впевненості.

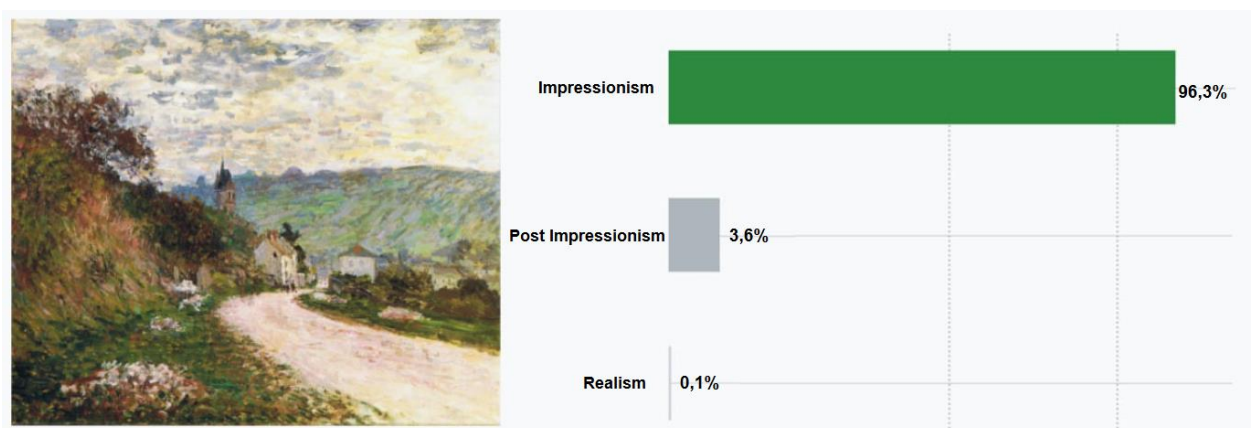


Рисунок 4.6 – Візуалізація результатів роботи класифікаційного модуля

Водночас імовірнісний підхід розкриває свій потенціал під час аналізу складніших художніх творів. Зокрема, під час обробки міського пейзажу мережа згенерувала розподіл у 75.5% для імпресіонізму та 21.0% для постімпресіонізму (рисунок 4.7). Індекс невизначеності зріс до 0.257, віднісши результат до середньої зони впевненості. З точки зору мистецтвознавства такий результат є абсолютно виправданим, адже полотно містить складне поєднання розмитих тональних переходів та більш чітких, фактурних мазків. Нейронна мережа не стала штучно завищувати впевненість одного класу, а математично точно продемонструвала синкретичну природу картини, зафіксувавши її в межевому стані між двома спорідненими напрямками.

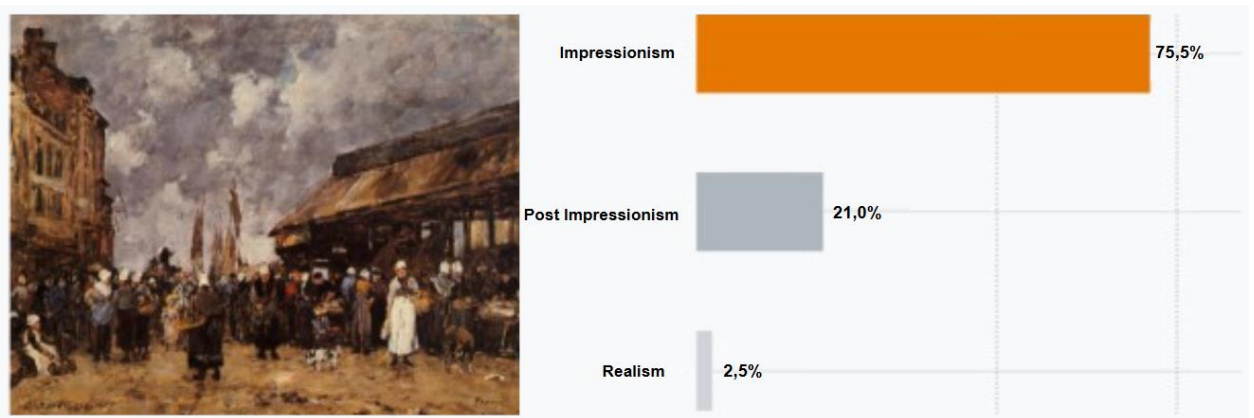


Рисунок 4.7 – Результати роботи визначення художнього стилю на тестовому зображенні

Обґрунтування використання алгоритму Grad-CAM. Використання алгоритму Grad-CAM (Gradient-weighted Class Activation Mapping) є необхідним етапом для забезпечення прозорості та інтерпретованості рішень розробленої системи. Одним із головних недоліків глибокого навчання є проблема, коли неможливо достеменно визначити, на основі яких саме візуальних ознак нейромережа приймає рішення. Завжди існує ризик того,

що модель навчилася розпізнавати хибні кореляції – наприклад, колір фону, залишки рами на краях фотографії або водяні знаки, замість того, щоб аналізувати художню техніку.

Візуалізація за допомогою теплових карт дозволяє зазирнути всередину процесу класифікації. Теплі кольори (червоні та жовті зони) на накладених картах чітко підсвічують ті піксельні області та патерни мазків, які мали найбільшу математичну вагу у фінальному згортковому шарі моделі.

Інтеграція алгоритму Grad-CAM дозволила верифікувати внутрішню логіку системи та довести, що вона орієнтується на релевантні ознаки (рис. 4.8)

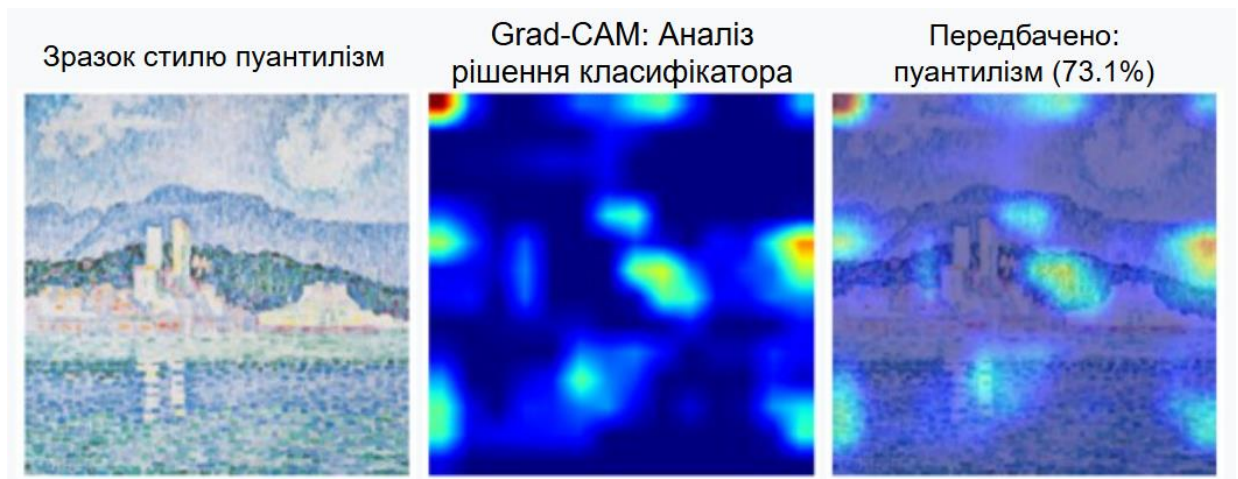


Рисунок 4.8 – Аналіз релевантних візуальних ознак художнього стилю за допомогою теплових карт

Таким чином, модуль пояснювального штучного інтелекту (англ. eXplainable Artificial Intelligence, XAI) підтвердив, що навіть при хибній класифікації архітектура логічно аналізує реальні структурні паттерни зображення, а не випадковий шум.

4.5 Можливості вдосконалення моделі та подальші експерименти

Розроблена інтелектуальна система класифікації художніх стилів демонструє високі результати в межах поставленого завдання, проте відкриває широкі можливості для подальшого вдосконалення як на рівні архітектури, так і в загальній концепції.

Одним із напрямків покращення є перехід до глибших згорткових мереж. Це збільшить виразність моделі та дозволить значно краще розпізнавати тонкі відмінності між спорідненими стилями. Перспективним також є поєднання класичних згорткових шарів, які виділяють локальні деталі, з механізмами уваги для моделювання загальної композиції. Такий підхід дасть змогу системі одночасно аналізувати як мікроструктуру мазків, так і просторову організацію полотна загалом.

Суттєвого покращення точності можна досягти шляхом обробки зображень вищої роздільної здатності, що збереже більше важливих стилістичних деталей. Розширення методів аугментації підвищить стійкість моделі до різної якості фотографій картин. Для складних класів варто застосувати алгоритми зважування помилок, щоб система приділяла більше уваги рідкісним мистецьким напрямкам.

Поточна версія системи охоплює 15 художніх стилів, проте логічним кроком є її розширення до повної бази даних – 27 категорій. Це ускладнить класифікацію та потребуватиме більшої кількості згенерованих даних, але суттєво підвищить практичну цінність рішення. Додатково можна реалізувати мультизадачну архітектуру для одночасного визначення стилю, жанру, автора та історичного періоду твору.

Генеративну складову системи також можна оптимізувати:

- 1) створення синтетичних зразків більшого розміру та кращої деталізації;

- 2) використання умовних генеративних мереж дозволить навчати єдину модель для всіх класів одночасно, що значно скоротить час підготовки даних;
- 3) застосування дифузійних моделей здатне забезпечити генерацію складних художніх текстур із ще вищою якістю та різноманітністю.

Найбільш амбітним етапом розвитку є створення на базі прототипу повноцінної інтелектуальної платформи для аналізу візуального мистецтва. Вона може об'єднати наступні функціональні модулі:

- аналітичний веб-інтерфейс для завантаження картин та миттєвого отримання результатів класифікації з візуальними поясненнями;
- пошукову систему для знаходження картин зі схожою стилістикою у цифрових архівах;
- модуль автоматичної каталогізації для музеїв та галерей, що автоматизує присвоєння стилістичних тегів великим колекціям;
- рекомендаційну систему для платформ і аукціонів, яка пропонуватиме твори на основі візуальних вподобань користувача;
- освітній інтерактивний модуль для студентів, який не лише класифікуватиме стиль, а й надаватиме пояснення його характерних ознак та мистецтвознавчий контекст.

Для реалізації такої платформи знадобиться розгортання хмарної інфраструктури, розробка програмних інтерфейсів (API) та створення баз даних для швидкого пошуку.

Отже, розроблена система є надійним науковим фундаментом для побудови комплексного рішення у сфері автоматизованого аналізу та каталогізації візуального мистецтва.

Висновки до розділу 4

Четвертий розділ містить опис програмної реалізації розробленої інтелектуальної системи. Для створення продукту було обрано мову програмування Python та фреймворк PyTorch. Система побудована за модульним принципом, де кожен блок – від завантажувача даних до модуля оцінки якості – функціонує як частина єдиного обчислювального процесу. Було детально описано реалізацію класів генератора та дискримінатора, а також процедуру fine-tuning класифікатора з використанням попередньо навчених ваг ImageNet. Для об'єктивного підтвердження працездатності системи було впроваджено комплексний набір метрик оцінки. Програмна реалізація підтвердила технічну придатність обраного стеку технологій для вирішення поставленої задачі. Розроблена архітектура коду є масштабованою, що дозволяє легко додавати нові стилі або змінювати архітектурні блоки мереж без необхідності повної перебудови системи.

ВИСНОВКИ

У цій дипломній роботі представлено актуальну та технічно складну проблему автоматичного розпізнавання та класифікації художніх стилів за допомогою методів глибокого навчання.

У ході дослідження було розв'язано низку наукових та практичних завдань, що дозволило досягти поставленої мети. Насамперед було проведено глибокий аналіз специфіки образотворчого мистецтва як об'єкта цифрової обробки та ідентифіковано основні чинники, що ускладнюють процес класифікації. Ключовим результатом роботи стала розробка комбінованого підходу, який інтегрує генеративно-змагальні мережі Вассерштейна з градієнтним штрафом для створення синтетичних зразків живопису. Це дозволило подолати проблему класового дисбалансу датасету WikiArt та забезпечити стабільне навчання класифікаційної моделі на основі архітектури ResNet-18. Експериментальна перевірка підтвердила, що запропонована система демонструє високу точність розпізнавання стилів та здатна ефективно працювати навіть із тими напрямками мистецтва, які мають обмежену кількість реальних зразків. Практичне значення отриманих результатів полягає у можливості використання розробленої системи для автоматичної каталогізації музейних фондів, створення інтелектуальних пошукових механізмів у цифрових архівах та розробки освітніх платформ для вивчення історії мистецтв. Робота закладає підґрунтя для подальших досліджень у сфері автоматизованого аналізу візуальної естетики за допомогою новітніх методів глибокого навчання.

ПЕРЕЛІК ПОСИЛАНЬ

1. Будума Н., Локашо Н. Основи глибокого навчання. Створення алгоритмів для штучного інтелекту наступного покоління. Київ, 2020. 304 с.
2. Goodfellow I. et al. Generative adversarial nets. *Advances in Neural Information Processing Systems*. 2014. Vol. 27.
3. Mao X., Li Q., Xie H., Lau R. Y., Wang Z., Smolley S. P. Least squares generative adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*. 2017. P. 2794–2802.
4. Jolicoeur-Martineau A. The relativistic discriminator: a key element missing from standard GAN. arXiv:1807.00734, 2018, DOI: <https://doi.org/10.48550/arXiv.1807.00734> (дата звернення: 28.01.26).
5. Kancharla P., Channappayya S. S. Quality aware generative adversarial networks. *Advances in Neural Information Processing Systems*. 2019. P. 2948–2958.
6. Kang M., Shim J., Cho M., Park J. Rebooting ACGAN: auxiliary classifier GANs with stable training. *Advances in Neural Information Processing Systems*. 2021. Vol. 34. P. 23505–23518.
7. Tan W. R., Chan C. S., Aguirre H. E., Tanaka K. Improved ArtGAN for conditional synthesis of natural image and artwork. *IEEE Transactions on Image Processing*. 2017. Vol. 28, № 1. P. 394–409.
8. Lin Z., Khetan A., Fanti G., Oh S. PacGAN: the power of two samples in generative adversarial networks. *Advances in Neural Information Processing Systems*. 2018. Vol. 31.
9. Ferrero A., Elhabian S., Whitaker R. SetGANs: enforcing distributional accuracy in generative adversarial networks. arXiv:1907.00109, 2019, DOI: <https://doi.org/10.48550/arXiv.1907.00109> (дата звернення: 20.02.26).

10. Srivastava A., Valkov L., Russell C. [et al.]. VEEGAN: reducing mode collapse in GANs using implicit variational learning. *Advances in Neural Information Processing Systems*. 2017. Vol. 30.
11. Xue Y., Tong W., Neri F., Zhang Y. PEGANs: phased evolutionary generative adversarial networks with self-attention module. *Mathematics*. 2022. Vol. 10, № 15. P. 2792.
12. Zhu J. Y., Park T., Isola P., Efros A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*. 2017. P. 2223–2232.
13. Karras T., Laine S., Aila T. A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019. P. 4401–4410.
14. Karras T., Aittala M., Hellsten J. [et al.]. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 12104–12114.
15. Arjovsky M., Chintala S., Bottou L. Wasserstein generative adversarial networks. *International Conference on Machine Learning*. PMLR, 2017. P. 214–223.
16. Gulrajani I., Ahmed F., Arjovsky M. [et al.]. Improved training of Wasserstein GANs. *Advances in Neural Information Processing Systems*. 2017. Vol. 30.
17. Metz L., Poole B., Pfau D., Sohl-Dickstein J. Unrolled generative adversarial networks. arXiv:1611.02163, 2016, DOI: <https://doi.org/10.48550/arXiv.1611.02163> (дата звернення: 03.02.26).
18. Lin Z., Khetan A., Fanti G., Oh S. PacGAN: the power of two samples in generative adversarial networks. *Advances in Neural Information Processing Systems*. 2018. Vol. 31.

19. Чумаченко О. І. Структурно-параметричний синтез гібридних нейронних мереж : дис. ... д-ра техн. наук : 05.13.23 / КПП ім. Ігоря Сікорського. Київ, 2019.
20. Thanh-Tung H., Tran T., Venkatesh S. On catastrophic forgetting and mode collapse in generative adversarial networks. *arXiv:1807.04015*, 2018, DOI: <https://doi.org/10.48550/arXiv.1807.04015> (дата звернення 10.03.2026).
21. Mao Q., Lee H. Y., Tseng H. Y. [et al.]. Mode seeking generative adversarial networks for diverse image synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019. P. 1429–1437.
22. Goni O., Arka H. S., Halder M. [et al.]. HingeRLC-GAN: combating mode collapse with hinge loss and RLC regularization. *arXiv:2503.19074*, 2025, DOI: https://doi.org/10.1007/978-3-031-78389-0_25 (дата звернення 10.03.2026).
23. Mangalam K. [et al.]. Overcoming mode collapse with adaptive multi adversarial training. *arXiv:2112.14406*, 2021, DOI: <https://doi.org/10.48550/arXiv.2112.14406> (дата звернення 13.04.2026).
24. Abbahaddou Y., Aboussalah A. M. Grassmannian geometry meets dynamic mode decomposition in DMD-GEN: a new metric for mode collapse in time series generative models. *arXiv:2412.11292*, 2024, DOI: <https://doi.org/10.48550/arXiv.2412.11292> (дата звернення 14.04.2026).
25. Radford A., Metz L., Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*, 2015, DOI: <https://doi.org/10.48550/arXiv.1511.06434> (дата звернення 06.05.2026).
26. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014, DOI: <https://doi.org/10.48550/arXiv.1409.1556> (дата звернення 19.04.2026).

27. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. arXiv:1512.03385, 2015, DOI: <https://doi.org/10.48550/arXiv.1512.03385> (дата звернення 25.03.2026).
28. Chollet F. Xception: deep learning with depthwise separable convolutions. arXiv:1610.02357. 2016. DOI: <https://doi.org/10.48550/arXiv.1610.02357> (дата звернення 22.03.2026).
29. Saleh B., Elgammal A. Large-scale classification of fine-art paintings: learning the right metric on the right feature. *International Journal for Digital Art History*. 2016. № 2. DOI: <https://doi.org/10.11588/dah.2016.2.23376> (дата звернення 01.05.2026).
30. Hentschel C., Wiradarma T. P., Sack H. Fine tuning CNNs with scarce training data adapting ImageNet to art epoch classification. *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016. P. 3693–3697. DOI: <https://doi.org/10.1109/ICIP.2016.7533049> (дата звернення 11.05.2026).
31. Menai B., Babahenini M. The effect of optimizers on CNN architectures for art style classification. *International Journal of Computing and Digital Systems*. 2023. Vol. 31. P. 353–360.
32. Efthymiou S., Rudinac S., Kackovic M. [et al.]. Graph neural networks for knowledge enhanced visual representation of paintings. 2021. DOI: <https://doi.org/10.1145/3474085.3475586> (дата звернення 07.05.2026).
33. Perez S. P., Cozman F. G. How to generate synthetic paintings to improve art style classification. *Brazilian Conference on Intelligent Systems*. Springer, 2021. P. 238–253.
34. Sandoval C., Pirogova E., Lech M. Two-stage deep learning approach to the classification of fine-art paintings. *IEEE Access*. 2019. DOI: <https://doi.org/10.1109/ACCESS.2019.2907986> (дата звернення 04.05.2026).

35. Zhao W., Zhou D., Qiu X., Jiang W. Compare the performance of the models in art classification. *PLoS ONE*. 2021. Vol. 16, № 3. P. e0248414. DOI: <https://doi.org/10.1371/journal.pone.0248414> (дата звернення 15.04.2026).
36. Wang Z. Art-style classification using MobileNetV2: a deep learning approach. *Proceedings of the 2nd International Conference on Data Science and Engineering (ICDSE 2025)*. 2025. P. 16–20. DOI: <https://doi.org/10.5220/0013677300004670> (дата звернення 23.05.2026).
37. Iliadis L. A., Nikolaidis S., Sarigiannidis P. [et al.]. Artwork style recognition using vision transformers and MLP Mixer. *Technologies*. 2021. Vol. 10, № 1. Art. 2.

ДОДАТОК А ОСНОВНІ СКЛАДНИКИ ЛІСТИНГУ ПРОГРАМИ

Повний лістинг коду розміщено за посиланням:

<https://github.com/bugs-bunny-nastia/painting-classification-system>

#1. Глобальні параметри системи:

```
# WGAN-GP
BATCH_SIZE_WGAN = 16
IMAGE_SIZE      = 128
LATENT_DIM       = 128
WGAN_EPOCHS      = 100
NGF, NDF         = 64, 64
```

Класифікатор

```
BATCH_SIZE_CLF   = 32
CLASSIFIER_EPOCHS = 30
TARGET_COUNT      = 3000
```

#2. Обчислення візуальної складності стилю:

```
def compute_style_complexity(files: list, n_samples: int = 30) -> float:
    sample = random.sample(files, min(n_samples, len(files)))
    variances = []
    for f in sample:
        img = np.array(
            Image.open(f).convert("RGB").resize((64, 64)),
            dtype=np.float32
        )
        variances.append(np.var(img))
    return float(np.mean(variances))
```

#3. Адаптивна оцінка ризику mode collapse:

```
# Адаптивні пороги кількості
threshold_critical = mean_count - std_count
threshold_moderate = mean_count
```

Критерій 1: відносна кількість зображень

```
if current_count < threshold_critical:
    count_score = 2
elif current_count < threshold_moderate:
    count_score = 1
else:
    count_score = 0
```

Критерій 2: візуальна складність (адаптивні пороги)

```
complexity = compute_style_complexity(files)
```

```

if complexity > COMPLEXITY_HIGH:  #  $\mu_C + \sigma_C$ 
    complexity_score = 1
elif complexity > COMPLEXITY_LOW:  #  $\mu_C - \sigma_C$ 
    complexity_score = 0
else:
    complexity_score = -1

# Комбінована оцінка → стратегія
total_score = count_score + complexity_score

if total_score >= 2:
    risk = 'МАКСИМАЛЬНИЙ'
    use_dropout=True; use_gaussian_noise=True
    use_diversity_loss=True; diversity_alpha=0.8; n_critic=2
elif total_score == 1:
    risk = 'ПОМІРНИЙ'
    use_dropout=True; use_gaussian_noise=False
    use_diversity_loss=True; diversity_alpha=0.5; n_critic=3
else:
    risk = 'НИЗЬКИЙ'
    use_dropout=False; use_gaussian_noise=False
    use_diversity_loss=False; diversity_alpha=0.0; n_critic=3

#Архітектура Generator з Self-Attention:
class SelfAttention(nn.Module):
    def __init__(self, in_channels):
        super().__init__()
        self.query = nn.Conv2d(in_channels, in_channels // 8, 1)
        self.key = nn.Conv2d(in_channels, in_channels // 8, 1)
        self.value = nn.Conv2d(in_channels, in_channels, 1)
        self.gamma = nn.Parameter(torch.zeros(1))

    def forward(self, x):
        b, C, w, h = x.size()
        N = w * h
        query = self.query(x).view(b, -1, N).permute(0, 2, 1)
        key = self.key(x).view(b, -1, N)
        energy = torch.bmm(query, key)
        attention = F.softmax(energy, dim=-1)
        value = self.value(x).view(b, -1, N)
        out = torch.bmm(value, attention.permute(0, 2, 1))
        out = out.view(b, C, w, h)
        return self.gamma * out + x

class Generator(nn.Module):
    def __init__(self):
        super().__init__()
        self.model = nn.Sequential(
            nn.ConvTranspose2d(LATENT_DIM, NGF*16, 4, 1, 0),

```

```

        nn.BatchNorm2d(NGF*16), nn.ReLU(True),
        nn.ConvTranspose2d(NGF*16, NGF*8, 4, 2, 1),
        nn.BatchNorm2d(NGF*8), nn.ReLU(True),
        nn.ConvTranspose2d(NGF*8, NGF*4, 4, 2, 1),
        nn.BatchNorm2d(NGF*4), nn.ReLU(True),
        nn.ConvTranspose2d(NGF*4, NGF*2, 4, 2, 1),
        nn.BatchNorm2d(NGF*2), nn.ReLU(True),
        SelfAttention(NGF*2),          # механізм уваги
        nn.ConvTranspose2d(NGF*2, NGF, 4, 2, 1),
        nn.BatchNorm2d(NGF), nn.ReLU(True),
        nn.ConvTranspose2d(NGF, 3, 4, 2, 1),
        nn.Tanh()
    )

    def forward(self, x):
        return self.model(x)

# Архітектура Discriminator:
class Discriminator(nn.Module):
    def __init__(self, use_dropout: bool = True):
        super().__init__()
        layers = [nn.Conv2d(3, NDF, 4, 2, 1), nn.LeakyReLU(0.2, True)]
        if use_dropout:
            layers.append(nn.Dropout2d(0.25))
        layers += [nn.Conv2d(NDF, NDF*2, 4, 2, 1),
                   nn.InstanceNorm2d(NDF*2), nn.LeakyReLU(0.2, True)]
        if use_dropout:
            layers.append(nn.Dropout2d(0.25))
        layers += [
            nn.Conv2d(NDF*2, NDF*4, 4, 2, 1),
            nn.InstanceNorm2d(NDF*4), nn.LeakyReLU(0.2, True),
            nn.Conv2d(NDF*4, NDF*8, 4, 2, 1),
            nn.InstanceNorm2d(NDF*8), nn.LeakyReLU(0.2, True),
            nn.Conv2d(NDF*8, NDF*8, 4, 2, 1),
            nn.InstanceNorm2d(NDF*8), nn.LeakyReLU(0.2, True),
            nn.Flatten(),
            nn.Linear((NDF*8)*4*4, 1)
        ]
        self.model = nn.Sequential(*layers)

    def forward(self, x):
        return self.model(x)

#Gradient Penalty:
def gradient_penalty(D, real, fake):
    b = real.size(0)
    eps = torch.rand(b, 1, 1, 1, device=device)
    interp = (eps * real + (1 - eps) * fake).requires_grad_(True)
    d_interp = D(interp)

```

```

grads      = torch.autograd.grad(
    outputs=d_interp, inputs=interp,
    grad_outputs=torch.ones_like(d_interp),
    create_graph=True, retain_graph=True
)[0]
return ((grads.view(b, -1).norm(2, dim=1) - 1) ** 2).mean()

#Цикл навчання WGAN з адаптивними методами:
for real in dataloader:
    real, b_size = real.to(device), real.size(0)

    # Навчання дискримінатора
    for _ in range(n_critic):
        opt_D.zero_grad()
        z      = torch.randn(b_size, LATENT_DIM, 1, 1, device=device)
        fake = G(z).detach()
        gp     = gradient_penalty(D, real, fake)
        real_in = (real + torch.randn_like(real) * 0.05
                    if use_gaussian_noise else real)
        loss_D = D(fake).mean() - D(real_in).mean() + 10 * gp
        loss_D.backward()
        torch.nn.utils.clip_grad_norm_(D.parameters(), 1.0)
        opt_D.step()

    # Навчання генератора
    opt_G.zero_grad()
    if use_diversity_loss:
        z1, z2 = (torch.randn(b_size, LATENT_DIM, 1, 1, device=device),
                  torch.randn(b_size, LATENT_DIM, 1, 1, device=device))
        f1, f2  = G(z1), G(z2)
        loss_adv = -D(f1).mean()
        n_pixels = f1.numel() / f1.size(0)
        div_loss = -torch.mean(torch.abs(f1 - f2)) / n_pixels
        loss_G    = loss_adv + diversity_alpha * div_loss
    else:
        loss_G = -D(G(torch.randn(b_size, LATENT_DIM, 1, 1,
                                   device=device))).mean()

    loss_G.backward()
    torch.nn.utils.clip_grad_norm_(G.parameters(), 1.0)
    opt_G.step()

#Динамічний моніторинг collapse:
if (epoch + 1) % 10 == 0 and len(g_losses) >= 5:
    g_std  = np.std(g_losses[-5:])
    g_mean = np.mean(g_losses[-5:])
    d_trend = d_losses[-1] - d_losses[-5]
    gap      = abs(g_mean - np.mean(d_losses[-5:]))

    if g_std < 0.5 and g_mean > 10:          # генератор застряг

```

```

        n_critic = max(n_critic - 1, 1)
    if d_trend < -5:
        diversity_alpha = min(diversity_alpha + 0.1, 1.0)
        use_diversity_loss = True
    if gap > 20:
        print(f"⚠ Епоха {epoch+1}: ДИСБАЛАНС_НАВЧАННЯ")

#TransformWrapper – усунення витоку даних:
class TransformWrapper(Dataset):
    """Застосовує індивідуальний трансформ до кожної підмножини
    (train/val/test) без зміни оригінального датасету."""
    def __init__(self, subset: Subset, transform):
        self.subset = subset
        self.transform = transform

    def __len__(self):
        return len(self.subset)

    def __getitem__(self, idx):
        img, label = self.subset[idx]
        return self.transform(img), label

# Використання:
train_ds = TransformWrapper(train_sub, train_transform)
val_ds = TransformWrapper(val_sub, eval_transform)
test_ds = TransformWrapper(test_sub, eval_transform)

#Налаштування ResNet-18 з Transfer Learning:
model = resnet18(weights=ResNet18_Weights.DEFAULT)
model.fc = nn.Sequential(
    nn.Dropout(0.4),
    nn.Linear(model.fc.in_features, len(SELECTED_STYLES))
)

# Заморожування перших двох блоків
for name, param in model.named_parameters():
    if 'layer1' in name or 'layer2' in name:
        param.requires_grad = False

optimizer = torch.optim.Adam(model.parameters(), lr=3e-5, weight_decay=1e-4)
scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(
    optimizer, T_max=CLASSIFIER_EPOCHS, eta_min=1e-6
)

#Модуль невизначеності:
probs = F.softmax(model(img_tensor), dim=1)[0]
top3_probs, top3_idx = torch.topk(probs, 3)

```

```

# Ентропійна невизначеність
entropy          = -torch.sum(probs * torch.log(probs + 1e-8))
max_entropy      = torch.log(torch.tensor(float(len(SELECTED_STYLES))))
uncertainty_score = (entropy / max_entropy).item() # ∈ [0, 1]

# Зони впевненості
if top3_probs[0] >= 0.80:
    zone = "ВИСОКА"
elif top3_probs[0] >= 0.50:
    zone = "СЕРЕДНЯ"
else:
    zone = "МЕЖОВА"

```