УДК 004.8 DOI https://doi.org/10.35546/kntu2078-4481.2026.1.29

О. В. ВЕРЕТЮК
аспірант кафедри систем автоматизованого проектування
Національний університет «Львівська політехніка»
ORCID: 0009-0009-2340-4458

Н. А. АНДРУЩАК
кандидат технічних наук,
доцент кафедри систем автоматизованого проектування
Національний університет «Львівська політехніка»
ORCID: 0000-0002-8248-404X

# ОПТИМІЗАЦІЯ НЕЙРОННИХ МЕРЕЖ В ДИФУЗІЙНИХ МОДЕЛЯХ ДЛЯ ГЕНЕРАЦІЇ ЕСКІЗІВ З ВИКОРИСТАННЯМ КРИВИХ БЕЗ'Є

*У цій статті представлено та обґрунтовано інноваційний підхід до оптимізації складного процесу генерування ескізів як графічного контенту, що є критично важливим етапом для автоматизації процесу малювання. Актуальність роботи продиктована необхідністю зменшення обчислювального навантаження сучасних генеративних моделей при збереженні високої якості вихідного зображення. На основі проведеного аналізу недоліків растрової графіки було вирішено використати альтернативний тип абстрактних даних – криві Без'є, на противагу загальноприйнятим піксельним сіткам. Цей тип даних надає значно ширші можливості в побудові невеликих, але ефективних моделей з більш якісним та масштабованим результатом генерування. В результаті експериментів було отримано модель, яка здатна успішно генерувати нові дані з вхідного ескізу, а також детально проаналізовано показники стабільності та варіативності процесу генерування. У роботі проведено ґрунтовний порівняльний аналіз кількості параметрів нейронної мережі, необхідних для генерації графіки у піксельному форматі та за допомогою кривих Без'є. Встановлено, що використання кривих суттєво зменшує кількість вхідних параметрів та обчислювальну складність мережі. Наприклад, для створення моделі при генеруванні піксельного зображення 32 × 32 потрібно 1024 вхідні параметри (якщо пікселі представлені у чорно-білому форматі) або 3072 (якщо у форматі RGB). Це значно перевищує показник у понад 800 параметрів моделі, яка використовувалась для генерування масштабованих кривих у даному дослідженні. Ключовою перевагою нового підходу визначено забезпечення ідеальної масштабованості: згенеровані дані можуть бути легко збільшені або зменшені до будь-якого розміру з мінімальною втратою якості, що є неможливим у традиційних піксельних (растрових) форматах. Також варто зауважити, що якість згенерованого зображення визначається лише потенціалом архітектури нейромережі, а не обмеженням фіксованої роздільної здатності, притаманним піксельним форматам.*

***Ключові слова:*** *оптимізація, нейронні мережі, графічний контент, масштабованість, криві Без'є.*

O. V. VERETIUK
Postgraduate Student at the Department of Computer Design Systems
Lviv Polytechnic National University
ORCID: 0009-0009-2340-4458

N. A. ANDRUSHCHAK
PhD, Associate Professor at the Department of Computer Design Systems
Lviv Polytechnic National University
ORCID: 0000-0002-8248-404X

## OPTIMIZATION OF NEURAL NETWORKS IN DIFFUSION MODELS FOR SKETCH GENERATION USING BEZIER CURVE

*This article presents a comprehensive study regarding an innovative approach to optimising the computational process of generating artistic sketches, specifically focusing on the significant task of automating digital drawing workflows. Following a detailed and rigorous analysis of existing methodologies, a strategic decision was made to utilize parametric curve-based abstract data structures – specifically Bézier curves – as a superior alternative to the traditional, widely utilized raster pixel representations. This fundamental transition to using mathematical curve descriptions provides significantly broader opportunities for constructing compact, efficient models that yield higher quality and fully scalable generation results. Through this research, a robust model capable of generating novel geometric data from an input*

© О. В. Веретюк, Н. А. Андрущак, 2026
Стаття поширюється на умовах відкритої ліцензії CC BY 4.0

*sketch was obtained, and the stability, consistency, and variability of the generation process were rigorously analyzed to ensure reliability. A critical component of this study involves a comparative analysis of the neural network parameters required for generating graphics in standard pixel formats versus the proposed Bézier curve method. The empirical results demonstrated that the employment of Bézier curves significantly reduces both the number of necessary input parameters and the overall computational complexity of the neural network architecture. To illustrate this computational disparity, creating a model to generate a standard 32 × 32 pixel image requires 1,024 input parameters (if the pixels are represented in black and white format) or 3,072 parameters (if in RGB format). In stark contrast, the model utilized to generate scalable curves requires only 800 parameters, representing a substantial reduction in computational load while maintaining structural integrity. The pivotal advantage of this new approach lies in its provision of perfect scalability; unlike traditional pixel (raster) formats, the generated data can be effortlessly enlarged or reduced to any dimension with zero loss of visual fidelity. Consequently, the quality of the generated image is no longer bound by the fixed resolution limitations inherent in raster formats but is determined solely by the potential and sophistication of the neural network architecture.*

***Key words:*** *optimization, neural networks, graphic content, scalability, Bezier curves.*

**Постановка проблеми**

Використання генеративного штучного інтелекту для генерування графічного контенту стає більш популярним з кожним днем. Проте, його використання у вирішенні практичних проблем є все ще обмеженим. Однією з проблем є висока ціна тренування та використання таких моделей, що зменшує доступність для роботи з цими технологіями для малих компаній або досліджень з малими бюджетами. Особливо це стосується тих індустрій, які пов'язані з процесом малювання.

**Аналіз останніх досліджень і публікацій**

Можливості використання штучного інтелекту для покращення або автоматизації процесу генерування контенту для різних індустрій активно досліджується [1][2]. В основі сучасних моделей для генерації графічного контенту лежить методологія дифузійних моделей [3], які довели свою ефективність в генеруванні високоякісного контенту.

В той самий час проводяться дослідження вартості тренування та використання моделей, які показують тенденцію до збільшення вартості [4], що робить важливим, як пошук підходів оптимізації так й практичного використання.

**Формулювання мети дослідження**

Метою роботи є розробка підходу для зменшення складності нейромережі у генеративних моделях, при збереженні якості та можливостей для практичного застосування.

**Викладення основного матеріалу дослідження**

Використання штучного інтелекту в різних його формах стає більш звичним та популярнішим з кожним днем. Однією з галузей штучного інтелекту, яка отримала найбільший поштовх останні роки став генеративний штучний інтелект. [5][6]

Генеративний штучний інтелект – це методологія штучного інтелекту, яка зосереджена на моделях та методах, які здатні породжувати новий контент відповідно, до даних, на яких такі моделі тренуються. Сьогодні існує доволі багато різних моделей, проєктів та інструментів для генерування різного типу даних: ChatGPT[7] для генерування та аналізу текстів, DALL-E[8] для створення високоякісних зображень, ElevenLabs[9] для створення аудіо та музики, Sora[10] для відео. Це лише невеликий перелік існуючих досягнень, який поповнюється кожен день, як схожими проєктами так і новими рішеннями, які розширюють можливості штучного інтелекту.

Однією з найбільш потенціальних галузей, які стрімко розвиваються, є власне генерування графічного контенту в різних його формах. Сьогодні існує великий вибір, як у виборі генеративних моделей для дослідження, так й у готових інструментах, де будь хто може створити нове зображення або відео ввівши декілька слів у текстове поле. Існують два основні типи моделей в генеруванні такого типу контенту. Генеративні змагальні мережі (GAN) [11], які довгий час були основним підходом, та дифузійні моделі [12], які набули популярності в останні роки, та на яких базується більшість відомих інструментів для генерування зображень та відео.

Як відомо, генеративні змагальні мережі мають архітектуру, яка складається з двох мереж, генератора та дискримінатора. Генератор пробує згенерувати контент, а дискримінатор намагається визначити чи контент відповідає дійсності. Ці дві мережі навчаються одночасно і з часом тренування генератор починає надавати все кращі та кращі екземпляри. Проте цей підхід є повноцінною чорною коробкою, процес генерування якої доволі важко контролювати, а значення похибки, яке визначається за допомогою дискримінанта при тренуванні, може не бути репрезентаційною, адже за великим рахунком залежить виключно від здатності моделі дискримінанта вчитись в тому ж темпі, що й модель генератора. В основі дифузійних моделей лежать два процеси: руйнація вхідних даних та їх відновлення за певну кількість кроків. Руйнація здійснюється за допомогою додавання випадкового шуму, який часто береться з гаусового розподілу. Цей шум додається визначену кількість разів, до того моменту поки вхідні дані самі не перетворяться в гаусовий шум. Відновлення ж відбувається за рахунок передбачення

стану даних в зворотньому порядку. Дуже часто передбачається власне шум, який додавався на тому чи іншому кроці для виведення результату. В основі цих процесів лежить математичний апарат, що є важливою відмінністю від змагальних мереж. Це дозволяє отримати кращий контроль над процесами генерування, а похибка обчислюється з виведених формул

Сьогодні вже описані різні типи дифузійних моделей, які мають свої переваги та недоліки. В даній статті використовується класична модель, яка носить назву Імовірнісні дифузійна модель з усуненням шуму або Denoising Diffusion Probabilistic Models (DDPM)[13] в оригіналі. В її основі, як й в інших дифузійних моделей, лежать три основні формули: формула руйнації, формула генерування та формула тренування.

Для прикладу, формулу руйнації або накладання шумів можна записати із використанням рівняння 1, яке може бути записано як:

$$x_t = \sqrt{1-b_t} \cdot x_{t-1} + \sqrt{b_t} \cdot \varepsilon, \tag{1}$$

де $x_t$ – зображення (або дані) після додавання шуму, $b_t$ – коефіцієнт шуму, $x_{t-1}$ – зображення (або дані) після додавання шуму на кроці $t-1$, ε –шум отриманий з гаусового розподілу

Проте на практиці використовується оптимізована формула, яка дозволяє опинитись на обраному кроці руйнування без проходження попередніх кроків. Це дозволяє заощадити, як використання обчислювальних ресурсів, так й часу тренування. Ця формула може бути записана як:

$$x_t = \sqrt{\overline{a}_t} \cdot x_0 + \sqrt{1-\overline{a}_t} \cdot \varepsilon, \tag{2}$$

де $x_t$ – зображення (або дані) після додавання шуму, $\overline{a}_t$ – накопичений коефіцієнт збереження, $x_0$ – початкове зображення (або дані), ε – шум отриманий з гаусового розподілу

В основі цієї формули лежить накопичений коефіцієнт збереження, який дозволяє оминути додаткових кроків при обчисленні стану зображення на тому чи іншому кроці руйнації.

Формула для накопичуваного коефіцієнту збереження

$$\overline{a}_t = \prod_{s=1}^{t} a_s, \tag{3}$$

де $\overline{a}_t$ – накопичений коефіцієнт збереження, $a_s$ – коефіцієнт збереження на кроці $s$

Формула коефіцієнту збереження

$$a_t = 1 - b_t, \tag{4}$$

де $a_t$ – коефіцієнт збереження, $b_t$ – коефіцієнт шуму

Формула генерації нових даних

$$x_{t-1} = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{b_t}{\sqrt{1-\overline{a}_t}}\varepsilon(x_t, t)\right) + \sigma_t z, \tag{5}$$

де $x_{t-1}$ – зображення (або дані) отримані з даних на кроці $t$, $x_t$ – поточне зображення (або дані), $\overline{a}_t$ – накопичений коефіцієнт збереження, $\varepsilon(x_t, t)$ – передбачений шум (вихід з нейромережі), $\sigma_t$ – значення наближене до квадратного кореню з $b_t$. Варто зауважити, що в оригіналі для $\sigma_t$ використовується окрема формула, яка виводиться як з $b_t$ так й з $\overline{a}_t$, проте це спрощення не дуже сильно впливає на кінцевий результат [14], тому було вирішено обрати спрощений варіант.

Формула втрат (середньоквадратичне відхилення), яка мінімізується під час навчання

$$L_{simple} = E_{t,x_0,\varepsilon}\left[\left\|\varepsilon - \varepsilon_\theta(x_t, t)\right\|^2\right], \tag{6}$$

де $E_{t,x_0,\varepsilon}$ – усереднення за випадковими вибірками (оцінка Монте-Карло) часу $t$, початкового прикладу $x_0$ та шуму ε, $x_0$ початкове зображення (дані) з датасету, $t$ – крок дифузії, ε – реальний шум, який додається під час процесу шумування, $x_t$ – зашумлена версія $x_0$ на кроці $t$, $\varepsilon_\theta(x_t, t)$ – шум, який передбачає нейромережа.

Як відомо звичайним типом даних, який використовується в генеративних мережах є піксель. Це найпопулярніший тип представлення зображень та графічного контенту. Будь який контент може бути представлений набором пікселів від фотографій до малюнків, що дає можливість швидко сформувати тренувальний набір даних та почати роботу над моделлю. Проте незважаючи на універсальність цього типу даних в нього є одна велика проблема – цей тип даних не є оптимізованим, адже залежить від розширення, яке визначає як якість так й наскільки великим може бути зображення. Більше того, тренування моделі зазвичай має обмеження: одна модель – одне розширення для невеликих моделей, адже нейронні мережі мають визначену кількість параметрів на вході й на виході, що буде відповідати кількості пікселів. Проте, варто зауважити, що існують рішення, які дозволяють мати можливість вибору розширення при генеруванні, але це вимагає великих обчислювальних потужностей як при

навчання так й при генеруванні, що може бути неефективним використанням ресурсів для вирішення тих чи інших задач.

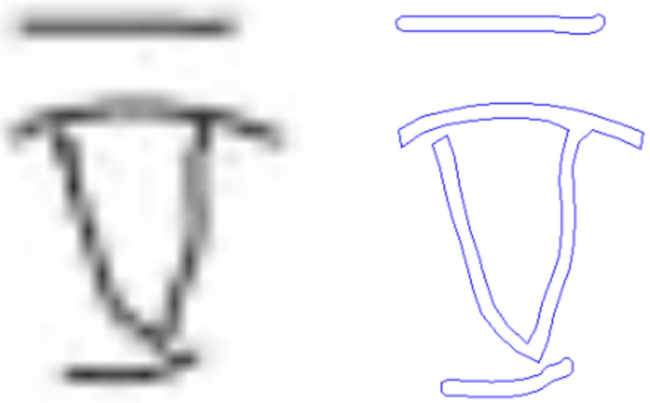
Використання більш абстрактних типів даних, може бути більш доцільним для вирішення спеціалізованих задач та проблем. Наприклад криві Без'є можуть бути ефективним та оптимізованим типом для даних для роботи з ескізами, що може бути корисним в спрощенні та пришвидшенні процесу малювання.

**Опис експерименту**

В межах статті використовувалась нейронна мережа з 800 вхідними та 800 вихідними параметрами, що відповідає 400 точкам, які складаються з двох координат, та є доволі низькою кількістю параметрів для генеративної мережі. В той час для генерування картинки навіть розміром 32X32 потрібно 1024 параметрів, якщо пікселі представлені чорно-білому форматі, або 3072, якщо у форматі RGB.

На рис. 1 представлено відображення одного ескізу, при використанні піксельного типу даних в розширенні 32X32, та кривих Без'є, загальна кількість точок, яких не перевищує 800.

Варто зауважити, що для того, щоб мати можливість показати зображення в достатньому розмірі, треба було перевести зображення у формат 32X32, а потім до формату 256X256, що сприяло ще більший деградації якості. В той самий час масштабування не є проблемою для кривих Без'є, адже вони складаються з точок з координатами, що дозволяє збільшувати або зменшувати зображення з мінімальними втратами якості та даних. В той самий час, якщо є потреба в роботі з більш сучасними розширеннями, як наприклад 1280X720, кількість параметрів буде дорівнювати 921 600 для чорно-білих пікселів, або 2 764 800 для пікселів у форматі RGB, що багатократно збільшує складність нейронної мережі, та кількість обчислювальних ресурсів потрібних для тренування.



*а)* *б)*

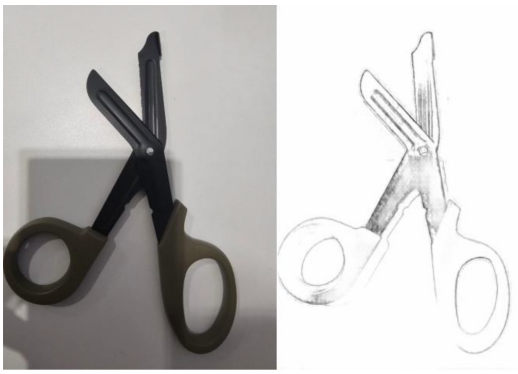
**Рис. 1. Порівняння зображень: *а*) у піксельному форматі, *б*) у форматі кривих Без'є**

Набір даних був побудований на основі малюнків очей в популярному японському анімаційному стилі. Однією з найбільших проблем при роботі з даними є їх переведення у потрібний формат. На противагу піксельним зображенням, які є у великій кількості в мережі інтернет, графічні дані у вигляді точок або кривих Без'є важко знайти у вільному доступі. Було вирішено використати програмний підхід, який дозволяє перевести більшу частину зображення у формат ескізу, який надалі може бути оброблений за допомогою бібліотеки для роботи з графічним контентом OpenCV для видобутку кривих.

Для переведення даних у ескіз був використаний відкритий інструмент[15], який використовує GAN моделі, щоб прибрати кольори, зайві деталі й залишає зображення, яке складається з чорних ліній на білому фоні. Цей інструмент базується на статті, в якій досліджують можливості генерації фото з вхідного ескізу[16].

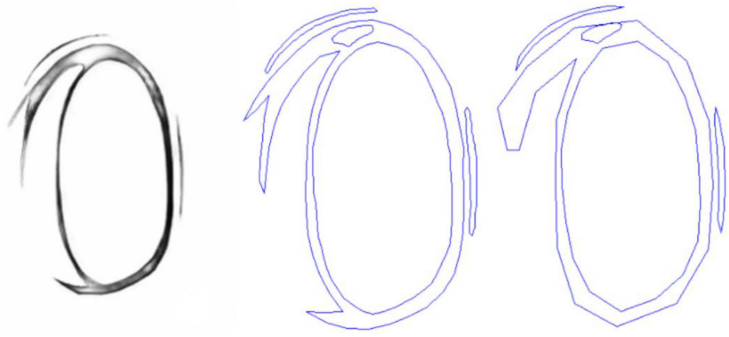
Далі за допомогою бібліотек OpenCV і skimage, та з використанням розробленого алгоритму виділяється набір кривих, кожна з яких далі розділяється на менші криві, які складаються з чотирьох точок. Алгоритм дозволяє обирати максимальну кількість кривих та точок, з яких складаються криві, на які можна розділити отримане зображення, це дозволяє регулювати якість та деталізованість даних. Приклади переведення піксельного ескізу в криві різної деталізованості можна побачити на рис. 3.

Надалі, ці дані вже можна використовувати при тренуванні дифузійної моделі. Проте при першому тренуванні моделі була знайдена велика проблема пов'язана з якістю згенерованих результатів. Якщо лишити формат даних, коли кожна окрема крива передбачається окремо, результат генерування буде складатись з рваних кривих. Це пов'язано, зі специфікою роботи генеративних мереж. Вони насправді не генерують дані, а лише апроксимують можливе місце знаходження даних. Відповідно спільні точки між кривими, можуть бути нескінченно близько, проте між ними все одно буде лишатись відстань, яка зробить деградацію відображення більшою при



*а)* *б)*

**Рис. 2. Приклад переведення піксельного зображення у формат ескізу, *а*) оригінальне зображення, *б*) зображення перетворене у ескіз**



а) б) в)

**Рис. 3. Приклад а) оригінального ескізу, б) ескіз з кривих, які були розділені на небільше ніж 5 кривих та 100 точок в кожній, в) ескіз намальований з кривих, які були розділені на небільше ніж 5 кривих та 30 точок в кожній**

масштабуванні картинки. В якості рішення цієї проблеми, було запроповано перетворити всі криві на одну велику криву, яка складається з усіх точок, які присутні на ескізі. Більше того цей підхід дозволяє, ще більше оптимізувати дані, адже спільні точки будуть представлені лише один раз. Для отримання можливості відновлення минулого формату даних, індекси розділення кривих та спільних точок зберігаються в тимчасовій базі даних.

На рис. 4 можна побачити приклад розірваних або накладених одну на одну кривих у випадку використання окремих кривих, як вхідних та вихідних даних при генеруванні.

**Рис. 4. Приклад згенерованого зображення при генеруванні окремих кривих**

Було підготовлено набір даних, який складається з 213 ескізів очей різної деталізованості у форматі японського анімаційного стилю.

Для тренування генеративних моделей зазвичай використовується інфраструктура побудована на відеокартах, адже вони надають набагато більший обчислювальний потенціал для тренування нейронних мереж ніж

процесори. Сьогодні існує велика кількість різних сервісів, які надають в оренду різні типи відеокарт для тренування. Одним з таких сервісів є Google Colab. Цей інструмент надає обмежений вибір інфраструктури для тренування, проте підтримує формат Jupyter Notebook, який дозволяє швидку побудову програмного забезпечення для тренування й тестування моделі в межах сервісу.

В наслідок експериментування з параметрами дифузійних моделей було натреновано декілька різних екземплярів. Результати генерування показані нижче відносяться до останньої моделі та є найбільш вдалими. Параметри моделі, які були використані: кількість кроків $T$ – 500, верхня межа $b_t$ – 0.0005, нижня межа $b_t$ – 0.000001 та кількість тренувальних епох – 500 000. Нейронна мережа має підтримку 800 вхідних та вихідних параметрів, які представляють собою координати 400 окремих точок.

**Перевірка отриманих результатів**

Для дифузійних моделей існує два типи перевірки якості. Перший це використання числового параметра, який розраховується під час тренування й визначає похибку (MSE) при передбачення шуму, який має бути прибраний від даних, які передбачаються. Графік зменшення похибки показаний на рис. 5. Другим є прямий візуальний, тобто оцінка якості згенерованого контента людиною.

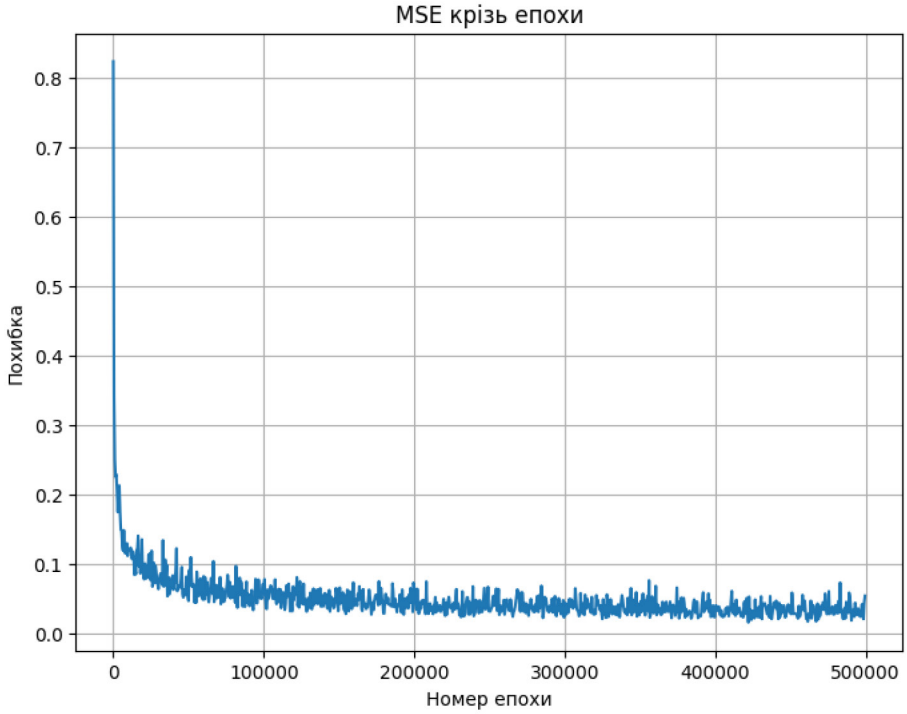
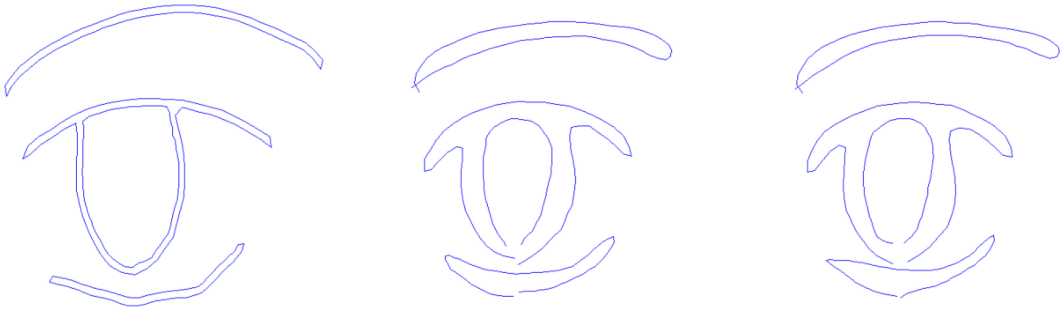



**Рис. 5. Графік зменшення похибки під час тренування – треба завжди вказувати розмірності на графіку**

Як видно з рис. 5, похибка різко зменшується на початкових 50 000 епохах та має певну тенденцію зменшення похибки, але в результаті виходить на певне плато. Фінальна похибка – 0.05413289368, в той час як початкова – 0.8232346177. Відповідно відбулось покращення точності в ~16 разів. Проте в генеративних мережах так чи інакше числова похибка може не бути репрезентативною, тому вимагається візуальна перевірка принаймні декількох результатів генерування. Нижче представлені результати генерування з вхідного ескізу в чотирьох варіантах. Кожен екземпляр має вхідне зображення та два згенерованих з нього, що потрібно для оцінки стабільності та різноманітності генерування. Результати генерування представлені на рис 6, 7, 8, 9.

На основі результатів генерування можна зробити перелік висновків. Перший, що модель повторює риси оригінального ескізу, але додає додаткові деталі. Другий це стабільність вихідних результатів генерування. Можна побачити, що ескізи згенеровані з одного зображення мають багато схожих рис, але не є ідентичні. Це може бути важливою властивістю при роботі з ескізами, адже дозволяє мати більший контроль над процесом генерування, що є важливим для практичного застосування. Варто зауважити, що випадковість генерування можна збільшити, якщо змінити верхню та нижню межу $b_t$. Третій це здатність моделі розуміти де знаходяться нульові значення. Як було описано вище, ця модель має 800 вхідних та вихідних параметрів, тобто здатна працювати не більш ніж з 400 точками. Проте різні вхідні ескізи можуть мати різну кількість точок після етапу обробки. Ці точки, які не є задіяними в процесі генерації модель накопичує біля нуля, або в лівій верхній частині згенерованих зображень. Це важливо, адже робить модель більш універсальною, адже дає можливість гнучкого збільшення чи зменшення



а) б) в)

**Рис. 6. Результат генерування а) оригінальний ескіз б) перший згенерований ескіз в) другий згенерований ескіз**

а) б) в)

**Рис. 7. Результат генерування а) оригінальний ескіз б) перший згенерований ескіз в) другий згенерований ескіз**

а) б) в)

**Рис. 8. Результат генерування а) оригінальний ескіз б) перший згенерований ескіз в) другий згенерований ескіз**

а) б в)

**Рис. 9. Результат генерування а) оригінальний ескіз б) перший згенерований ескіз в) другий згенерований ескіз**

масштабів генерування незалежно від розширення екрану. Це дозволяє отримати якісні результати генерування для моделей невеликого розміру, що неможливо при використанні піксельного формату даних.

Варто окремо відзначити ціну тренування цієї моделі. Google Colab використовує обчислювальний блок, як одиницю вимірювання витрат на тренування. Загалом тренування зайняло ~36 обчислювальний блоків. Ціна одного обчислювального блока на момент написання статті складає близько 5 гривень, відповідно ціна вихідної моделі є 180 гривень. Також варто зауважити, що кількість даних, яка було використана для тренування моделі є доволі малою. Тому зважаючи на малу ціну та обмежений набір даних результати генерування можна вважати вдалими.

Є декілька можливих рішень для покращення якості генерування: збільшення набору тренувальних даних та покращення його якості, збільшення складності моделі за рахунок внутрішніх шарів нейронної мережі та збільшення використання обчислювальних блоків.

На рис. 10 можна побачити приклад згенерованих зображень[17] в масштабі 32X32. Для генерування цих зображень використовувалась модель, яка має 3072 вхідних та вихідних параметрів, що в рази більше ніж в моделі, яка працює з кривими. Проте незважаючи на більшу кількість вхідних параметрів є перелік проблем: результати є розмитими на противагу точному відображенню при використанні кривих, їх неможливо масштабувати без великої втрати якості, для збільшення якості зображень треба збільшити складність вхідного та вихідного шару нейронної мережі, що в той самий час вимагатиме набагато складнішої внутрішньої архітектури нейронної мережі не тільки для збільшення якості генерування, а й для підтримки більшої кількості вхідних параметрів. Збільшення розширення згенерованого зображення до сучасних стандартів вимагатиме дуже глибоку й комплексну архітектуру для нейромережі тільки для того, щоб мати можливість опрацьовувати велику кількість вхідних та вихідних параметрів. Більше того при роботі з ескізами велика частина пікселів не буде задіяна в генеруванні, а отже сприятиме неефективному використанню обчислювальних ресурсів.

**Рис. 10. Результат генерування в піксельному форматі**

## Висновки

У ході проведеного дослідження було запропоновано підхід для оптимізованого та якісного генерування графічного контенту у дифузійних моделях. Цей підхід базується на використанні кривих Без'є як основного типу даних на противагу широко використовуваним пікселям. Реалізація та експерементальні дослідження були проведені з використанням мов Python та бібліотек PyTorch, pandas, OpenCV та skimage. Як середовище тренування та розробки використовувався Google Colab з Jupyter Notebook. У процесі експериментів досліджувалось можливість використання кривих як типу даних та які оптимізаційні можливості це дає.

Аналіз результатів показав, що дифузійні моделі можуть ефективно використовуватись для генерації даних у форматі кривих Без'є. Ключовою перевагою цього підходу є значне зменшення кількості параметрів моделі та обчислювальної складності. Для порівняння: якщо генерація растрового зображення 32 × 32 вимагає 1024 (якщо пікселі у чорно-білому форматі) або 3072 (RGB) параметри, то для моделі, що генерує криві, було достатньо лише 800 параметрів. Це дозволяє збільшувати якість згенерованого контенту виключно за рахунок ускладнення архітектури нейромережі, а не через збільшення обсягу вхідних/вихідних даних, як у піксельних форматах. Таким чином, якість обмежується лише потенціалом моделі, а не фіксованою роздільною здатністю, та забезпечує ідеальну масштабованість результату без втрати якості.

Ефективність підходу було підтверджено внаслідок експериментів та порівняння результатів генерування при використанні кривих та при використанні пікселів. Також було окреслено додаткові кроки, які можуть призвести до поліпшення моделі та результатів її генерування.

Результати описані в цій статті доводять важливість використання альтернативних типів даних для практичного застосування генеративних мереж в таких індустріях як анімація, малювання та дизайн.

**Список використаної літератури**

1. Hutson, J., & Robertson, B. (2023). A Matter of Perspective: A Case Study in the Use of AI-Generative Art in the Drawing Classroom. International Journal of New Media, Technology & the Arts, 18.
2. Porvari, E. (2024). Exploring the potential of AI utilization in the interpretation and digitization of legacy drawings.
3. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y.,... & Yang, M. H. (2023). Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, *56*(4), 1–39.
4. Cottier, B., Rahman, R., Fattorini, L., Maslej, N., Besiroglu, T., & Owen, D. (2024). The rising costs of training frontier AI models. *arXiv preprint arXiv:2405.21015*.
5. Mudrinić, D., & Šoda, I. (2024, November). The Rise of Generative Artificial Intelligence in Business. In 2024 IEEE 17th International Scientific Conference on Informatics (Informatics) (pp. 493–498). IEEE.
6. Kusetogullari, A., Kusetogullari, H., Andersson, M., & Gorschek, T. (2025). GenAI in entrepreneurship: A systematic review of generative artificial intelligence in entrepreneurship research: Current issues and future directions. arXiv preprint arXiv:2505.05523.
7. Zhang, M., & Li, J. (2021). A commentary of GPT-3 in MIT Technology Review 2021. Fundamental Research, 1(6), 831-833.
8. Puteikis, K., & Mameniškienė, R. (2024). Artificial intelligence: Can it help us better grasp the idea of epilepsy? An exploratory dialogue with ChatGPT and DALL· E 2. Epilepsy & Behavior, 156, 109822.
9. Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R.,... & Sun, L. (2024). Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv preprint arXiv:2402.17177.
10. Samson, G. (2025, June). Perspectives on Generative Sound Design: A Generative Soundscapes Showcase. In Arts (Vol. 14, No. 3, p. 67). MDPI.
11. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S.,... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.
12. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015, June). Deep unsupervised learning using nonequilibrium thermodynamics. In International conference on machine learning (pp. 2256–2265). pmlr.
13. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems, 33, 6840-6851.
14. Nichol, A. Q., & Dhariwal, P. (2021, July). Improved denoising diffusion probabilistic models. In International conference on machine learning (pp. 8162–8171). PMLR.
15. Xiang, X., Liu, D., Yang, X., Zhu, Y., & Shen, X. (2021). Anime2Sketch: A Sketch Extractor for Anime Arts with Deep Networks [Computer software]. GitHub. https://github.com/Mukosame/Anime2Sketch
16. Xiang, X., Liu, D., Yang, X., Zhu, Y., Shen, X., & Allebach, J. P. (2022). Adversarial open domain adaptation for sketch-to-photo synthesis. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1434–1444).
17. Веретюк, О., Огерук, Б., & Андрущак, Н. (2025). Використання шумів для стабільного генерування зображень у дифузійних моделях. Комп'ютерні системи проєктування. Теорія і практика. 7(1), 113–123. https://doi.org/10.23939/cds2025.01.113

**References**

1. Hutson, J., & Robertson, B. (2023). A Matter of Perspective: A Case Study in the Use of AI-Generative Art in the Drawing Classroom. International Journal of New Media, Technology & the Arts, 18.
2. Porvari, E. (2024). Exploring the potential of AI utilization in the interpretation and digitization of legacy drawings.
3. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y.,... & Yang, M. H. (2023). Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, *56*(4), 1–39.
4. Cottier, B., Rahman, R., Fattorini, L., Maslej, N., Besiroglu, T., & Owen, D. (2024). The rising costs of training frontier AI models. *arXiv preprint arXiv:2405.21015*.
5. Mudrinić, D., & Šoda, I. (2024, November). The Rise of Generative Artificial Intelligence in Business. In 2024 IEEE 17th International Scientific Conference on Informatics (Informatics) (pp. 493–498). IEEE.
6. Kusetogullari, A., Kusetogullari, H., Andersson, M., & Gorschek, T. (2025). GenAI in entrepreneurship: A systematic review of generative artificial intelligence in entrepreneurship research: Current issues and future directions. arXiv preprint arXiv:2505.05523.
7. Zhang, M., & Li, J. (2021). A commentary of GPT-3 in MIT Technology Review 2021. Fundamental Research, 1(6), 831–833.

8. Puteikis, K., & Mameniškienė, R. (2024). Artificial intelligence: Can it help us better grasp the idea of epilepsy? An exploratory dialogue with ChatGPT and DALL· E 2. Epilepsy & Behavior, 156, 109822.

9. Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R.,... & Sun, L. (2024). Sora: A review on background, technology, limitations, and opportunities of large vision models. arXiv preprint arXiv:2402.17177.

10. Samson, G. (2025, June). Perspectives on Generative Sound Design: A Generative Soundscapes Showcase. In Arts (Vol. 14, No. 3, p. 67). MDPI.

11. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S.,... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

12. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015, June). Deep unsupervised learning using nonequilibrium thermodynamics. In International conference on machine learning (pp. 2256–2265). pmlr.

13. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems, 33, 6840–6851.

14. Nichol, A. Q., & Dhariwal, P. (2021, July). Improved denoising diffusion probabilistic models. In International conference on machine learning (pp. 8162–8171). PMLR.

15. Xiang, X., Liu, D., Yang, X., Zhu, Y., & Shen, X. (2021). Anime2Sketch: A Sketch Extractor for Anime Arts with Deep Networks [Computer software]. GitHub. https://github.com/Mukosame/Anime2Sketch

16. Xiang, X., Liu, D., Yang, X., Zhu, Y., Shen, X., & Allebach, J. P. (2022). Adversarial open domain adaptation for sketch-to-photo synthesis. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1434–1444).

17. Veretiuk, O., Oheruk, B., & Andrushchak, N. (2025). Using noises for stable image generation in diffusion models. Computer Science and Systems, 7(1), 113–123. https://doi.org/10.23939/cds2025.01.113

*Дата першого надходження статті до видання: 17.01.2026*
*Дата прийняття статті до друку після рецензування: 20.02.2026*
*Дата публікації (оприлюднення) статті: 30.04.2026*