

ІННОВАЦІЙНІ МЕТОДИ ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ПРОЄКТУВАННЯ ШРИФТІВ У ЦИФРОВОМУ ДИЗАЙНІ

Семенів М.Р.

к.т.н., доцент, кафедра Інформаційних технологій видавничої справи,
Національний університет «Львівська політехніка»
ORCID ID: 0000-0003-0798-1916

Кинаш Ю.Є.

к.ф.-м.н., доцент, кафедра Інформаційних технологій видавничої справи,
Національний університет «Львівська політехніка»
ORCID ID: 0000-0002-3762-3215

Кустра Н.О.

к.т.н., доцент, кафедра Інформаційних технологій видавничої справи,
Національний університет «Львівська політехніка»
ORCID ID: 0000-0002-3562-2032

Логойда М.М.

к.т.н., доцент, кафедра Інформаційних технологій видавничої справи,
Національний університет «Львівська політехніка»
ORCID ID: 0000-0001-7597-7973

***Анотація.** Проаналізовано інноваційні методи та нейромережеві архітектури для генерування гліфів і перенесення стилю на шрифтові форми. Запропоновано методологію концептуальної добудови шрифту. Практичну апробацію здійснено із застосуванням CNN-based Style Transfer, DALL-E, Gemini 3 Flash Image, GPT Image 2 та Flux.2. Результати оцінено за стилістичною відповідністю, читабельністю, збереженням пропорцій.*

***Ключові слова:** шрифт, AI-інструменти, перенесення стилю, гліф, кирилиця, морфометричний аналіз.*

Вступ

У цифровому дизайні текст є важливою візуальною складовою, яка визначає характер комунікації, ритм композиції та емоційне сприйняття повідомлення. Вибір шрифту впливає на те, як користувач сприйме зміст тексту: як нейтральний, офіційний, динамічний, елегантний, технологічний чи експериментальний. Типографіка – це мистецтво й система організації тексту у візуальному просторі і є важливим історичним та культурним елементом, який відображає характерні особливості певних епох. Як частина дизайну, вона є однією із найкращих сфер для експериментів, впровадження нестандартних рішень та звернень до національної ідентичності.

Історія дизайну шрифтів показує, що якісно розроблені гарнітури можуть залишатися актуальними протягом тривалого часу: Garamond, Baskerville, Bodoni, Didot, Times New Roman, Futura, Univers, Helvetica й досі активно

використовуються у верстці, брендингу, інтерфейсах та рекламній комунікації. Водночас розробка шрифту, особливо призначеного для суцільного читання, вимагає тривалої роботи над формою гліфів, ритмом набору та міжлітерними інтервалами. Це підтверджує досвід Крістіана Шварца, який у 2004–2010 роках здійснював цифрове відновлення Neue Haas Grotesk з максимальною увагою до оригінальних форм [1].

На цьому тлі дизайн шрифтів дедалі більше виходить за межі традиційного малювання окремих гліфів і переходить у сферу системного моделювання графічної форми. Цей перехід можна розглядати як зміну парадигми: від створення окремих статичних символів до формування шрифтової системи, побудованої на правилах, залежностях та параметрах. Дизайнери шрифтів можуть використовувати алгоритмічні та інтелектуальні системи, здатні синтезувати, варіювати та інтерпретувати візуальні особливості шрифтів. Про це свідчить розвиток варіативних шрифтів (*variable fonts*), у яких один файл шрифту може містити кілька стилів і забезпечувати безперервну варіацію за осями насиченості, ширини або іншого параметра. Специфікація *OpenType Font Variations* закріпила параметричність як повноцінний інструмент цифрової типографіки [2].

Розвиток дизайну шрифтів пов'язаний із залученням інтелектуальних систем: генеративних моделей, стилізації за допомогою нейронних мереж та методів автоматизованого відтворення графічних ознак. Ці технології відкривають нові можливості для створення стилістично узгоджених наборів гліфів, але не виключають ролі дизайнера, оскільки шрифт вимагає аналізу читабельності, цілісності, технічної правильності контурів та культурної релевантності. Такі інструменти автоматизують окремі етапи роботи, а також здатні аналізувати та відтворювати складні графічні характеристики. Дослідження С. Азаді та ін. [3] демонструє, що технології перенесення стилю шрифту можуть переносити стилістичні особливості заданих гліфів на нові символи, формуючи узгоджений набір шрифту на основі обмеженої кількості зразків. Відтак, інноваційні методи та інтелектуальні системи перетворюють типографіку на динамічну систему, в якій дизайнер виступає архітектором ідей, а технології є засобом їх реалізації.

Стрімкий розвиток технологій штучного інтелекту зумовлює переосмислення класичних підходів до створення шрифтів у сучасному цифровому дизайні. В процесі проектування шрифту з'являється можливість використання цифрових систем, здатних аналізувати, генерувати, варіювати та адаптувати графічні форми. У цьому контексті особливої актуальності набуває створення кирилических шрифтів, зокрема таких, що відображають українську культурну та національну ідентичність. Тривалий час візуальний простір України значною мірою формувався під впливом шрифтів російського походження, однак впродовж останнього десятиліття помітно активізувався розвиток українського шрифтового дизайну [4].

Мета та задачі дослідження

Актуальність дослідження зумовлена необхідністю вивчення інноваційних інструментів, які дають змогу поєднати професійні принципи шрифтового дизайну та автоматизовану обробку графічних форм. Особливе значення має трансформація візуальної мови українського мистецтва у форму сучасного кириличного шрифту.

Метою роботи є дослідження можливостей інноваційних методів та інтелектуальних систем у проектуванні цифрових шрифтів, а також їх практичне випробування у процесі створення кирилично-шрифтової гарнітури на основі інтерпретації художнього стилю та характерних формотворчих ознак одного з українських художників.

Для досягнення мети визначено наступні завдання.

1. Систематизувати сучасні підходи до шрифтового дизайну, сформувавши таксономію методів.
2. Розробити методіку проектування шрифту на основі нейромережевої інтерпретації художнього оригіналу та подальшої обробки форм.
3. Провести порівняльний морфометричний аналіз результатів генерації моделей генерування зображень для оцінки їхньої придатності до створення цілісних шрифтових систем.

Основна частина

Варіативні шрифти як передумова інтелектуальної типографіки

У розвитку цифрового шрифтового дизайну важливим етапом стало впровадження варіативних шрифтів (variable fonts), офіційно представлених як частина стандарту OpenType 1.8 за участю Adobe, Apple, Google і Microsoft [5]. Їхня поява змінила уявлення про шрифт як про набір окремих статичних файлів на зразок Regular, Bold, Italic або Condensed. У специфікації OpenType Font Variations зазначено, що варіативний шрифт може містити кілька шрифтових накреслень у межах одного ресурсу та забезпечувати безперервну зміну форми за визначеними осями дизайну, наприклад насиченістю, шириною, нахилом або оптичним розміром [2]. Це дало змогу розглядати шрифт не лише як завершений набір контурів, а як керовану систему варіативних графічних форм.

Основна інновація варіативних шрифтів полягає в тому, що вони не потребують зберігання окремих шрифтових файлів для кожного накреслення. Натомість у межах одного ресурсу зберігаються базові контури та варіаційні дані, які забезпечують обчислення проміжних станів гліфа в межах заданого дизайну. У шрифтовому файлі визначаються осі варіативності, діапазони їхніх значень і окремі іменовані позиції, що можуть відповідати звичним накресленням, наприклад Light, Regular або Bold. За глобальний опис осей і таких іменованих позицій відповідає таблиця fvar, тоді як конкретні варіаційні

дані для контурів та інших параметрів можуть зберігатися в інших таблицях формату OpenType [2].

З погляду формального опису варіативний шрифт можна розглядати як систему, у якій цільова форма гліфа залежить від набору параметрів. У спрощеному вигляді це можна подати так:

$$f(w, s, o) = \text{Glyph}_{\text{target}}, \quad (1)$$

де w – насиченість (*weight*);
 s – ширина (*width*);
 o – оптичний розмір (*optical size*).

Така модель демонструє, що форма гліфа може бути описана через залежність від змінних, які впливають на його графічні характеристики. Саме тому варіативні шрифти можна розглядати як важливу передумову інтелектуальної типографіки: якщо шрифтову форму подано як параметричну систему, її характеристики можуть бути не лише вручну налаштовані дизайнером, а й проаналізовані, модифіковані або оптимізовані алгоритмічними та нейромережевими методами.

Із цим безпосередньо пов'язана концепція параметричного дизайну. У шрифтовому проектуванні вона означає перехід від створення кожної літери як окремого завершеного рисунка до побудови системи правил, залежностей і контрольованих параметрів. Такими параметрами можуть бути товщина штриха, ширина знака, висота малих літер, висота або довжина виносних елементів, радіус заокруглень, контраст між основними й додатковими штрихами, нахил, форма засічок, характер внутрішніх просвітів і міжлітерні інтервали. У межах такого підходу дизайнер працює не лише з окремими векторними вузлами, а з логікою побудови всієї шрифтової системи.

Параметричний підхід змінює професійну роль дизайнера. Він уже не лише креслить контури гліфів, а проектує модель, у якій форма літери виникає внаслідок взаємодії параметрів. Такий підхід можна пояснити через логіку “скелета” літери: спочатку визначається базова конструкція знака, а потім до неї застосовуються змінні характеристики – товщина штриха, контраст, завершення, заокруглення або нахил. Завдяки цьому одна система може породжувати низку споріднених накреслень, зберігаючи стилістичну єдність гарнітури.

Отже, варіативні шрифти та параметричний дизайн формують проміжну ланку між класичним шрифтовим проектуванням і сучасними інтелектуальними системами. Вони переводять шрифтову форму в площину керованих змінних, що робить її придатною для алгоритмічного аналізу, автоматизованої модифікації та подальшої інтеграції з методами машинного навчання [2, 6]. Саме цей перехід є важливим для досліджень у сфері генерації гліфів, перенесення стилю та створення адаптивних шрифтових систем.

Машинне навчання та нейромережеві архітектури в генерації гліфів

Генеративно-змагальні мережі (Generative Adversarial Networks, GANs) у шрифтовому дизайні застосовуються для автоматичної генерації гліфів (графічне зображення знака у шрифті), перенесення стилістичних ознак між знаками, доповнення наборів шрифтових гарнітур і створення нових накреслень на основі обмеженої кількості зразків. Модель може навчатися на наявних бібліотеках шрифтів і виявляти закономірності побудови літер: пропорції, товщину штрихів, форму завершень, нахил, контраст, декоративні елементи та ритм знакової системи [3, 7].

Механізм GAN ґрунтується на взаємодії двох нейромереж: генератора (G) і дискримінатора (D). Математична основа функціонування генеративно-змагальних мереж базується на теорії ігор, зокрема на концепції мінімаксної гри з нульовою сумою. Наведене рівняння (2) описує цільову функцію $V(D, G)$, де дві мережі перебувають у стані постійної конкуренції за досягнення стану рівноваги Неша [8]. Генератор намагається створити новий гліф, який виглядав би як справжній шрифт. Дискримінатор (D) намагається відрізнити згенерований гліф від реального шрифту з навчальної вибірки. Математично це описується як мінімаксна гра:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (2)$$

де $\min_G \max_D$ – стратегічна мета навчання;

$E_{x \sim p_{data}(x)} [\log D(x)]$ – математичне сподівання логарифма ймовірності того, що Дискримінатор правильно ідентифікує реальний шрифт x з наявного датасету p_{data} ;

$G(z)$ – результат роботи Генератора;

z – вектор випадкового шуму;

$D(G(z))$ – оцінка Дискримінатором згенерованого об'єкта;

$E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$ – відповідає за штрафування Генератора, якщо Дискримінатор викриває підробку.

У процесі навчання генератор поступово вдосконалює якість результату, намагаючись створити гліфи, які зберігають потрібну структуру літери й водночас відповідають заданому стилю.

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) у шрифтовому дизайні застосовуються для розпізнавання стилістичних ознак гліфів, класифікації шрифтів і перенесення художнього стилю (style transfer). Їхня перевага полягає у здатності аналізувати локальні графічні характеристики: контури, товщину штрихів, серифи, нахил, ритм ліній, фактуру та декоративні елементи [9]. Згорткові мережі відіграють роль «очей» системи.

Завдяки ієрархічній структурі вони аналізують зображення гліфа на різних рівнях:

- нижні шари розпізнають базові елементи: кути, краї, напрям штрихів;
- верхні шари ідентифікують складні ознаки: наявність та форму засічок (serifs), контрастність штрихів, характер з'єднань.

Саме CNN дозволяють реалізувати концепцію Style Transfer (перенесення стилю). Наприклад, маючи лише літеру «А» у новому художньому виконанні, мережа виокремлює «код стилю» і накладає його на структуру інших літер алфавіту [10, 11].

Аналіз актуальних досліджень у галузі інтелектуального шрифтового дизайну свідчить про активне використання нейромережових архітектур для аналізу, генерування та стилістичної трансформації гліфів. Такі системи здатні виявляти закономірності побудови шрифтової форми, зокрема пропорції, контрастність, насиченість, засічки та декоративні ознаки. На сучасному етапі можна виокремити кілька ключових технологічних напрямів. Перший напрям охоплює моделі для генерації великих шрифтових бібліотек, зокрема DCFont [12], SCFont [13] та FontRNN [14]. Ці системи спрямовані на автоматичне створення великої кількості символів для складних ієрогліфічних писемностей на основі обмеженого набору зразків. Зокрема, DCFont генерує бібліотеку GB2312 із 6763 китайських символів, а FontRNN моделює символи через послідовності точок і траєкторії штрихів, враховуючи динаміку письма.

Паралельно розвивається група GAN-моделей, орієнтованих на перенесення стилю. Multi-Content GAN, TET-GAN та AGIS-Net [3, 15, 16] дають змогу переносити на нові гліфи не лише базові типографічні ознаки, а й складні декоративні ефекти: фактуру, колірні переходи, орнаменти. Такі моделі є важливими для завдань, у яких потрібно інтерпретувати художній почерк і поширити його на ширший набір символів. У межах few-shot font generation, зокрема в моделях DM-Font, LF-Font і MX-Font [17-19], увага зосереджується на локальних стильових ознаках і компонентах гліфа. Це дає змогу не копіювати шрифт як цілісне зображення, а точніше відтворювати окремі структурні та графічні елементи форми.

Важливим етапом стали структурні та компонентно-орієнтовані моделі, зокрема DG-Font [20] і CG-GAN [21]. Їхнє завдання полягає у збереженні конструктивної основи символу під час зміни стилю. Водночас CNN-підходи доцільно розглядати як інструменти аналізу візуальних ознак і перенесення стилю, тоді як SVG-VAE та DeepSVG [22, 23] репрезентують інший напрям – роботу з векторними та латентними представленнями графічної форми. Такі моделі важливі для шрифтового дизайну, оскільки фінальний шрифт має бути не лише зображенням, а технічно коректною системою контурів.

Одним із найактивніших напрямів останніх років є дифузійні моделі, зокрема Diff-Font і FontDiffuser [24-26]. Вони розглядають генерування шрифту як процес поступового уточнення зображення та дають змогу досягати високої якості стилізації, особливо у завданнях one-shot і few-shot font generation. Окремого значення набуває міжписемне перенесення стилю, представлене, зокрема, моделлю FTransGAN [27]. Такі підходи відкривають можливості для

адаптації стилістичних ознак між різними системами письма, однак потребують додаткового контролю графічної, мовної та культурної специфіки.

Отже, сучасні інтелектуальні методи створюють технологічну основу для генерування, стилістичної інтерпретації та розвитку шрифтових форм, зокрема в контексті актуалізації української візуальної спадщини в цифровому середовищі.

На основі аналізу наукових публікацій побудовано таксономію інтелектуальних методів генерування шрифтів і перенесення стилю на гліфи (рис. 1). В основу класифікації покладено функціональне призначення моделей і принцип їхньої роботи: генерація великих шрифтових бібліотек, перенесення стилю, генерування шрифтів з невеликої кількості зразків, компонентне моделювання, CNN-підходи, векторні та латентні моделі, дифузійні моделі й міжписемне перенесення стилю. Запропонована схема візуалізує результати типологізації джерел та представляє досліджувану сферу як цілісну систему взаємопов'язаних напрямів.

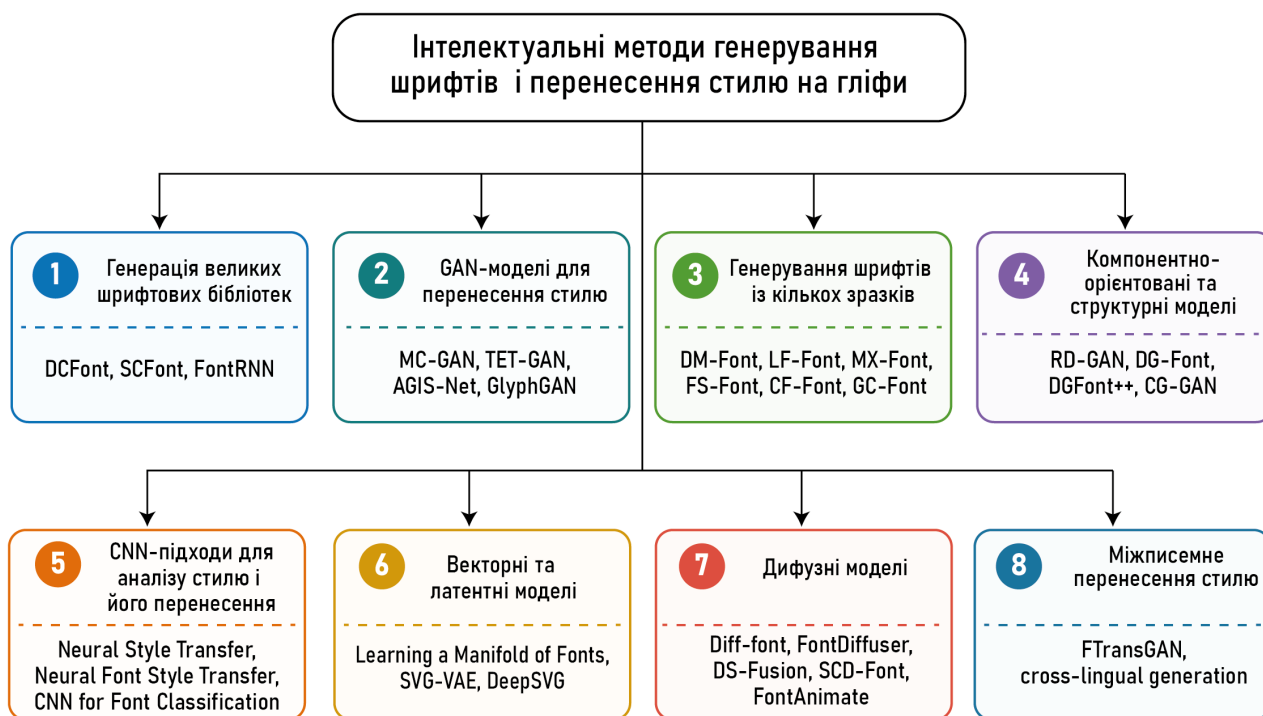


Рисунок 1 – Таксономія інтелектуальних методів генерування шрифтів і перенесення стилю на гліфи

Інструментальні засоби інтелектуального шрифтового проєктування

Використовуючи AI-інструменти дизайнери можуть інтегрувати алгоритмічні й інтелектуальні підходи у професійну практику без необхідності самостійного програмування. У сучасному цифровому дизайні функціонує низка вебзастосунків та AI-інструментів, які мають різне функціональне призначення: підтримують добір шрифтових пар, розпізнавання гарнітур, генеративний пошук стилістики та формування гарнітур.

Fontjoy є інструментом для автоматизованого добору шрифтових пар, що використовує технології машинного навчання [28]. Сервіс дає змогу генерувати нові комбінації, фіксувати вибрані шрифти, вручну змінювати окремі гарнітури та редагувати текст прикладу, налаштовувати рівень контрастності шрифтових пар за допомогою повзунка (рис. 2).



Рисунок 2 – Інтерфейс інструмента для автоматизованого добору шрифтових пар Fontjoy

Окрему групу становлять інструменти розпізнавання шрифтів за растровим зображенням. До них належать Adobe Retype [29], WhatTheFont [30], Font Matcherator by Fontspring [31]. Вони дають змогу завантажити зображення або виділити текстовий фрагмент і отримати перелік ідентичних чи подібних гарнітур. Такі сервіси є корисними для пошуку аналогів, вивчення візуальної подібності між гарнітурами та підготовки матеріалу для подальшого шрифтового проектування. Adobe Retype в Illustrator, зокрема, призначений для ідентифікації шрифтів із зображень або контурного тексту, тоді як WhatTheFont і Matcherator працюють як вебінструменти пошуку шрифту за зображенням.

Для генеративного створення шрифтової форми доцільно залучати інструменти, що працюють із візуальними референсами та векторною графікою. Adobe Illustrator Text to Vector Graphic, побудований на технологіях Adobe Firefly, дає змогу створювати редаговану векторну графіку на основі текстового опису [32]. Подібну роль може виконувати Recraft, який позиціонується як AI-платформа для дизайнерів і підтримує генерацію векторної графіки, ілюстрацій, логотипів, рекламних матеріалів і мокапів [33]. Важливо, що такі інструменти дають змогу працювати із масштабованими SVG-формами, що є доцільно при проектуванні шрифтової гарнітури.

Для дослідження, пов'язаного з інтерпретацією художньої мови українського митця у формі кириличного шрифту, найдоцільніше поєднувати ці інструменти послідовно: генеративний пошук стилю, параметричне осмислення форми, векторне редагування та фінальне формування гарнітури.

Методологія проєктування шрифту на основі художнього стилю

Проєктування шрифту на основі художнього стилю передбачає збір та аналіз творів мистецтва, виявлення їхніх характерних ознак, перенесення цих ознак на літерні форми, генерування кількох варіантів гліфів за допомогою AI-інструментів, порівняння отриманих результатів за визначеними критеріями, подальшу векторизацію та формування файлу шрифту.

Для практичної апробації методології обрано творчість Володимира Юрчишина – одного з відомих представників української книжкової графіки та шрифтового мистецтва. Його роботи поєднують глибоке знання історичної шрифтової культури, зокрема стародруків, із самобутнім авторським трактуванням літерної форми. У графіці Юрчишина важливу роль відіграють контраст чорного й білого, фактурність ліногравюри, орнаментальні мотиви, динамічна лінія, ритмічна організація композиції та активне використання негативного простору [34-36].

Юрчишин розглядав типографіку не як допоміжний елемент книжкового оформлення, а як повноцінну частину візуальної оповіді. У його роботах шрифт взаємодіє з орнаментом, а літери зберігають зв'язок із традиціями українських стародруків і скоропису. Саме творчість Володимира Юрчишина (рис. 3) розглядається як художнє першоджерело для створення кириличного акцидентного шрифту. Такий вибір є доцільним, оскільки акцидентний шрифт допускає вищий рівень художньої інтерпретації, ніж текстова гарнітура, і дає змогу перетворити характерні ознаки графіки Юрчишина на виразну систему кирилических гліфів.



Рисунок 3 – Зразки творчості Володимира Юрчишина [34]

У практичній частині дослідження використано два підходи. Перший підхід передбачає застосування CNN-based Style Transfer для перенесення художньої манери на форму літер. Для цього було підготовлено базові літерні

форми, які виконували функцію змістової структури, а також ілюстрацію художника, що використовувалася як джерело стилю. Перенесення стилю здійснювалося у середовищі Google Colab з використанням бібліотеки PyTorch. Для виділення ознак використовувалася згорткова нейронна мережа VGG-19 із бібліотеки torchvision.models. Модель не навчалася заново, а використовувалася як екстрактор ознак: її проміжні згорткові шари порівнювати зміст літери та стиль художнього зображення. Форма літери розглядалася як основа, а художнє зображення – як стилістичний референс.

Другий підхід ґрунтується на використанні вже наявного шрифтового зразка для побудови відсутніх літер кириличного алфавіту. За допомогою Recraft, DALL-E та Gemini створюються відсутні літери кириличного алфавіту в подібному стилі. З метою аналізу потенціалу AI-інструментів у підтриманні стилістичної єдності між наявними та новоствореними гліфами. На відміну від першого підходу, де основним завданням є перенесення художньої фактури на форму літери, другий підхід спрямований на побудову шрифтової системи на основі обмеженого набору літер.

Після генерації результатів проводиться порівняльне оцінювання. До основних критеріїв доцільно віднести стилістичну відповідність роботам художника, читабельність літер, цілісність шрифтової системи, технічну придатність до векторизації, рівень декоративності, кількість артефактів, потребу в ручному доопрацюванні та потенціал для створення повної кириличної гарнітури.

Завершальний етап передбачає постобробку та професійне шрифтове доопрацювання. Отримані результати потрібно векторизувати, відредагувати контури, уніфікувати пропорції, вирівняти висоту знаків, скоригувати внутрішні просвіти та налаштувати міжлітерні інтервали. Після цього гліфи можуть бути імпортовані у шрифтовий редактор, наприклад FontLab, для формування робочого шрифтового файлу. На цьому етапі важливо перевірити не лише окремі літери, а й їхню взаємодію у словах, коротких фразах і композиційних макетах.

Таким чином, запропонована методологія поєднує збір та аналіз творів мистецтва, використання AI-інструментів, порівняльне оцінювання та редагування. Створення шрифту розглядається як дослідницько-дизайнерський процес: від аналізу візуальної мови українського художника – до формування акцидентної кириличної гарнітури з виразною культурною та стилістичною ідентичністю.

Результати досліджень

У межах практичної частини дослідження було перевірено можливості AI-інструментів для інтерпретації художнього стилю Володимира Юрчишина у процесі створення кириличного акцидентного шрифту. Експеримент охоплював два основні підходи: перенесення стилю на базові літерні форми за допомогою CNN-based Style Transfer та побудову відсутніх літер кириличного алфавіту на

основі наявного шрифтового зразка із застосуванням Recraft, DALL-E та Gemini. На першому етапі було використано підхід CNN-based Style Transfer (рис. 4). Для цього в середовищі Google Colab було підготовлено два типи зображень: базові форми літер, які виконували функцію структурної основи, та ілюстрацію Володимира Юрчишина, що використовувалася як джерело стилістичних ознак. У результаті було отримано варіанти літер із перенесеними фактурними та декоративними характеристиками зображення.



Рисунок 4 – Приклади перенесення стилю ілюстрації Володимира Юрчишина на базові літерні форми методом CNN-based Style Transfer

Однак результати цього підходу виявилися обмежено придатними для подальшого шрифтового проектування. У частині зразків спостерігалось розмиття, поява зайвих текстурних плям і зниження читабельності. Тому цей підхід доцільно розглядати не як самостійний спосіб створення гліфів, а як допоміжний інструмент попереднього стилістичного пошуку.

У межах експерименту було проаналізовано придатність різних AI-інструментів для створення кириличного алфавіту на основі наявного шрифтового зразка. Вихідний напис використовується як опорний візуальний матеріал, а відсутні літери генерувалися з урахуванням його стилістичних і конструктивних характеристик. Це дало змогу оцінити не лише здатність інструментів відтворювати загальну художню манеру, а й точність збереження ключових параметрів літерної форми: висоти знаків, пропорцій, товщини основних і з'єднувальних штрихів, контрастності, форми засічок і характеру напливів.

Практичну цінність у цьому процесі продемонстрував Recraft (рис. 5). Його перевага полягає в тому, що в межах одного AI-інструменту можна здійснювати як векторизацію наявного напису, так і генерування нових літер у подібному стилі. Завдяки цьому Recraft виявився зручним для побудови цілісного робочого циклу: від підготовки опорного шрифтового зразка до створення нових гліфів. Водночас ефективність цього інструменту істотно залежить від якості промпта. Крім того, для отримання точніших результатів доцільно генерувати по одній літері, оскільки при спробі створювати кілька знаків одночасно знижується контроль над пропорціями, характером штрихів і деталями форми.

ТАРАС ШЕВЧЕНКО ПОЕМИ

Рисунок 5 – Результат векторизації Recraft

У процесі дослідження було порівняно результати генерування літер за допомогою моделей DALL·E, Gemini 3 Flash Image, GPT Image 2 та Flux.2. Основну увагу зосереджено на здатності моделей відтворювати стилістику вихідного шрифтового зразка, зокрема конструкцію літер, пропорції, характер штрихів, форму засічок, наявність діакритичних знаків та загальну цілісність шрифтової системи.

Аналіз згенерованих зображень показав, що досліджувані інструменти по-різному інтерпретують стилістичні та конструктивні ознаки вихідних літер. Найбільш наближеним до зразка виявився результат, отриманий за допомогою Gemini 3 Flash Image. У згенерованих літерах точніше збережено висоту знаків, співвідношення товщини основних і з'єднувальних штрихів, а також характер засічок. Це свідчить про кращу здатність моделі підтримувати візуальну єдність літер у межах заданої шрифтової стилістики (рис. 6).

Рисунок 6 – Результати побудови кирилических літер засобами Gemini 3 Flash Image

Модель DALL·E продемонструвала менш стабільні результати. У частині згенерованих літер спостерігалися відхилення у висоті знаків, нерівномірність контрасту між основними та допоміжними штрихами, а також зміни у формі кінцевих елементів. Це знижує цілісність шрифтового набору та ускладнює подальше використання результатів як основи для розроблення гарнітури (рис. 7). Водночас результати, отримані за допомогою DALL·E, можуть бути корисними на етапі пошуку загальної візуальної ідеї або стилістичного напрямку.

Рисунок 7 – Результати побудови кирилических літер засобами DALL-E

GPT Image 2 забезпечила результати, близькі до Gemini 3 Flash Image за загальною формою літер і відтворенням характеру шрифтового зразка. Модель достатньо точно передала основні пропорції, пластику штрихів і загальну стилістику літер (рис. 8). Проте в окремих випадках було виявлено неточності у побудові нижніх виносних елементів, зокрема в літерах Ц та Щ. Нерівномірність

цих елементів порушує ритм шрифтового набору та потребує додаткового ручного коригування.



Рисунок 8 – Результати добування кирилических літер засобами GPT Image 2

Модель Flux.2 згенерувала літери, які загалом стилістично подібні до вихідного зразка. Водночас було зафіксовано суттєві помилки у відтворенні окремих кирилических знаків (рис. 9). Зокрема, модель не згенерувала літеру Ї, а до літери Л помилково додала діакритичний знак. Такі результати свідчать про обмежену стабільність моделі під час роботи з кирилицею.



Рисунок 9 – Результати добування кирилических літер засобами Flux.2

У процесі порівняльного аналізу було верифіковано основні розмірні характеристики векторизованого зразка та гліфів, згенерованих нейромережами. Для об'єктивізації порівняння використано метод морфометричного аналізу прямих знаків (літери «Н» як еталона та «Й» як результату генерації), що мають ідентичну конструктивну основу.

Для кількісного оцінювання стилістичної відповідності було розраховано ключові типографічні показники за наступними формулами.

1. Насиченість (S) – відношення товщини основного штриха до висоти знака.

$$S = \frac{T_{main}}{H}, \quad (3)$$

де T_{main} – товщина основного штриха;
 H – висота літери.

2. Контрастність (C) – співвідношення товщини з'єднувального та основного штрихів.

$$C = \frac{T_{con}}{T_{main}}, \quad (4)$$

де T_{con} – товщина з'єднувального штриха.

Чим менше значення CS , тим вищою є графічна контрастність шрифту.

3. Відносна ширина (W_{rel}) – показник пропорційності вічка знака.

$$W_{rel} = \frac{W}{H}, \quad (5)$$

де W – ширина знака.

Результати вимірювань (Adobe Illustrator, інструмент «Measure») засвідчили, що модель Gemini 3 Flash Image продемонструвала найвищу точність відтворення метричних параметрів щодо вихідного зразка. Висота гліфа у Gemini 3 (19,613 мм) є майже ідентичною еталону (19,5 мм). Моделі GPT Image 2 (20,015 мм) та Flux (19,939 мм) також показали високу стабільність за висотою. Натомість DALL-E виявив суттєву похибку у масштабуванні (24,88 мм).

Аналіз пропорційності знаків через показник відносної ширини (W_{rel}) підтвердив лідерство GPT Image 2 (0,832) та Gemini 3 (0,853), які максимально наближені до значення оригінального зразка (0,836). Модель Flux сформувала дещо вужче вічко знака (0,782), а DALL-E – найбільш стиснуту форму (0,76).

За показником насиченості (S) Gemini 3 ($S = 0,336$) та вихідний зразок ($S_{ref} = 0,323$) утворюють найбільш узгоджену пару. Моделі Flux ($S = 0,311$) та GPT Image 2 ($S = 0,309$) продемонстрували результати, що є близькими до еталона, проте візуально виглядають дещо світлішими. Найменшу відповідність за насиченістю показав DALL-E ($S = 0,291$).

Дослідження контрастності (C) виявило, що найбільш автентичне співвідношення штрихів забезпечують GPT Image 2 ($C = 0,08$) та Flux ($C = 0,085$), що є максимально близьким до показника художнього оригіналу ($C_{ref} = 0,077$). Gemini 3 продемонстрував тенденцію до зменшення контрасту ($C = 0,092$), тоді як DALL-E схильний до надмірного контрастування елементів ($C = 0,068$).

Порівняльне оцінювання результатів синтезу кирилических гліфів проведено за допомогою методу експертного візуально-морфологічного аналізу. Для верифікації художньо-стилістичної та конструктивної якості використано 5-бальну шкалу, де 1 бал відповідає критично низькій точності, а 5 балів – високій стилістичній автентичності та технічній придатності до фінального шрифтового редагування.

Комплексне оцінювання моделей (табл. 1) показало, що Gemini 3 Flash Image є найбільш прецизійним інструментом для професійної побудови гарнітури (28 з 30 балів) серед розглянутих моделей, продемонструвавши стабільність пропорцій та стилістики.

Таблиця 1 – Експертне оцінювання результатів генерації кирилических гліфів

Критерій оцінювання	DALL-E	Gemini 3 Flash Image	GPT Image 2	Flux.2
Стилістична відповідність художньому джерелу / опорному зразку	3	5	4	4
Читабельність літер	5	5	5	4
Цілісність шрифтової системи	3	4	4	4
Збереження висоти та пропорцій літер	3	5	5	4
Збереження товщини основного та з'єднувального штриха	3	5	4	4
Форма засічок і наплівів	3	4	4	3

Модель GPT Image 2 виявилася кращою у збереженні внутрішньолітерного простору, проте потребує корекції насиченості штриха. Flux не забезпечує читабельність літер та є недоліки у формуванні засічок, що ускладнює векторизацію. Найнижчу ефективність виявила модель DALL-E, яка через значні відхилення у системній цілісності та пропорціях придатна лише для ескізного пошуку, попри високу читабельність знаків.

Отримані спостереження дають підстави стверджувати, що для створення кириличної акцидентної гарнітури найбільш доцільним є використання AI-інструментів не як повністю автономних генераторів, а як засобів пошуку та побудови літерних форм із подальшим дизайнерським доопрацюванням.

Найперспективнішим у цьому процесі виявився підхід, у якому поєднуються векторизація наявного зразка, генерація відсутніх літер за його мотивами та подальше професійне редагування результатів.

Висновки

У результаті проведеного дослідження систематизовано сучасні підходи до інтелектуального генерування шрифтів і перенесення стилю на гліфи. На основі аналізу наукових публікацій побудовано таксономію методів, у якій виокремлено основні напрями. Проаналізовано інструментальні засоби, які можуть бути використані у практиці інтелектуального шрифтового проєктування. Встановлено, що сучасні AI-інструменти та вебзастосунки не замінюють професійної роботи шрифтового дизайнера, але розширюють можливості пошуку, аналізу й попередньої генерації форм.

Запропоновано методологію проєктування кириличного акцидентного шрифту на основі художнього стилю. Вона передбачає аналіз творчості художника, виокремлення характерних формотворчих ознак, підготовку опорного шрифтового зразка, апробацію AI-інструментів для перенесення стилю та побудови відсутніх літер, а також подальше професійне доопрацювання гліфів. Морфометричний аналіз та апробація генеративних моделей підтвердили, що найвищу точність у побудові гліфів забезпечує Gemini 3 Flash Image, демонструючи стабільність метричних параметрів (насиченість, контраст, відносна ширина літери).

Таким чином, дослідження засвідчило, що інноваційні методи та AI-інструменти можуть бути ефективно залучені до процесу створення шрифтів, особливо на етапах аналізу стилю, пошуку формотворчих рішень, побудови відсутніх літер і підготовки графічних основ. Водночас кінцевий результат потребує професійного контролю дизайнера, оскільки шрифт є не лише візуальним образом, а складною системою гліфів, що має відповідати вимогам читабельності, технічної коректності та культурної виразності.

Список літератури.

1. Schwartz, C. (n. d.). Neue Haas Grotesk. <https://www.christianschwartz.com/haasgrotesk.shtml>
2. Microsoft Typography. (n. d.). OpenType font variations overview. Microsoft Learn. <https://learn.microsoft.com/en-us/typography/opentype/spec/otvaroverview>.
3. Azadi, S., Fisher, M., Kim, V., Wang, Z., Shechtman, E., & Darrell, T. (2017). Multi-content GAN for few-shot font style transfer. arXiv. <https://arxiv.org/abs/1712.00516>.
4. Огородник, С. (2022). Чому шрифти відображають історію та як народжується шрифтовий патріотизм. Lyuk Media. <https://lyuk.media/city/fonts-reflect-history/>.
5. Google Open Source Blog. (2016). Introducing OpenType font variations. <https://opensource.googleblog.com/2016/09/introducing-opentype-font-variations.html>.
6. web.dev. (n. d.). Introduction to variable fonts on the web. <https://web.dev/articles/variable-fonts>.
7. Hayashi, H., Abe, K., & Uchida, S. (2019). GlyphGAN: Style-consistent font generation based on generative adversarial networks. Knowledge-Based Systems, 186, Article 104927. <https://doi.org/10.1016/j.knosys.2019.104927>.
8. Campbell, N.D.F., & Kautz, J. (2014). Learning a manifold of fonts. https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf.
9. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521, 436-444. <https://doi.org/10.1038/nature14539>.
10. Gatys, L.A., Ecker, A.S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. Computer Vision and Pattern Recognition. (p. 2414-2423).
11. Atarsaikhan, G., Iwana, B. K., Narusawa, A., Yanai, K., & Uchida, S. (2017). Neural font style transfer. Document Analysis and Recognition (ICDAR). Vol. 1. (p. 51-56).
12. Jiang, Y., Lian, Z., Tang, Y., & Xiao, J. (2017). DCFont: An end-to-end deep Chinese font generation system. SIGGRAPH Asia 2017 Technical Briefs. Article 22. (p. 1-4). Association for Computing Machinery. <https://doi.org/10.1145/3145749.3149440>
13. Jiang, Y., Lian, Z., Tang, Y., & Xiao, J. (2019). SCFont: Structure-guided Chinese font generation via deep stacked networks. AAAI Conference on Artificial Intelligence, 33(1), 4015-4022. <https://cdn.aaai.org/ojs/4294/4294-13-7348-1-10-20190706.pdf>.
14. Tang, S., Xia, Z., Lian, Z., & Xiao, J. (n. d.). FontRNN: Generating large-scale Chinese fonts via recurrent neural network. <https://xiazeqing.github.io/FontRNN/documents/paper.pdf>.
15. Yang, S., Liu, J., Wang, W., & Guo, Z. (2019). TET-GAN: Text effects transfer via stylization and destylization. AAAI Conference on Artificial Intelligence, 33(1), 1238-1245. <https://cdn.aaai.org/ojs/3919/3919-13-6978-1-10-20190702.pdf>.
16. Gao, Y., Guo, Y., Lian, Z., Tang, Y., & Xiao, J. (2019). Artistic glyph image synthesis via one-stage few-shot learning. ACM Transactions on Graphics, 38(6), Article 185. <https://doi.org/10.1145/3355089.3356574>.
17. Cha, J., Chun, S., Lee, G., Lee, B., Kim, S., & Lee, H. (2020). Few-shot compositional font generation with dual memory. Computer Vision – ECCV 2020. https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123640715.pdf.
18. Park, S., Chun, S., Cha, J., Lee, B., & Shim, H. (2021). Few-shot font generation with localized style representations and factorization. AAAI Conference on Artificial Intelligence, 35(3), 2393-2402. <https://cdn.aaai.org/ojs/16340/16340-13-19834-1-2-20210518.pdf>.
19. Park, S., Chun, S., Cha, J., Lee, B., & Shim, H. (2021). Multiple heads are better than one: Few-shot font generation with multiple localized experts. IEEE/CVF International Conference on Computer Vision. (p. 13900-13909). https://openaccess.thecvf.com/content/ICCV2021/papers/Park_Multiple_Heads_Are_Better_Than_One_Few-Shot_Font_Generation_With_ICCV_2021_paper.pdf.
20. Xie, Y., Chen, X., Sun, L., & Lu, Y. (2021). DG-Font: Deformable generative networks for unsupervised font generation. IEEE/CVF Conference on Computer Vision and Pattern Recognition. (p. 5130-5140). https://openaccess.thecvf.com/content/CVPR2021/papers/Xie_DG-Font_Deformable_Generative_Networks_for_Unsupervised_Font_Generation_CVPR_2021_paper.pdf.

21. Kong, Y., Luo, C., Ma, W., Zhu, Q., Zhu, S., Yuan, N., & Jin, L. (2022). Look closer to supervise better: One-shot font generation via component-based discriminator. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (p. 13482-13491). https://openaccess.thecvf.com/content/CVPR2022/papers/Kong_Look_Closer_To_Supervise_Better_One-Shot_Font_Generation_via_Component-Based_CVPR_2022_paper.pdf.
22. Lopes, R. G., Ha, D., Eck, D., & Shlens, J. (2019). A learned representation for scalable vector graphics. *IEEE/CVF International Conference on Computer Vision*. (p. 7930-7939). https://openaccess.thecvf.com/content_ICCV_2019/papers/Lopes_A_Learned_Representation_for_Scalable_Vector_Graphics_ICCV_2019_paper.pdf.
23. Carlier, A., Danelljan, M., Alahi, A., & Timofte, R. (2020). DeepSVG: A hierarchical generative network for vector graphics animation. *arXiv*. <https://arxiv.org/pdf/2007.11301>.
24. He, H., Wang, X., Wang, Y., & Lian, Z. (2024). Diff-Font: Diffusion model for robust one-shot font generation. *International Journal of Computer Vision*. <https://doi.org/10.1007/s11263-024-02137-0>.
25. Yang, Z., Peng, D., Kong, Y., Zhang, Y., Yao, C., & Jin, L. (2024). FontDiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. *AAAI Conference on Artificial Intelligence*, 38(7), 6603-6611. <https://doi.org/10.1609/aaai.v38i7.28482>.
26. SGD-font: Style and glyph decoupling for one-shot font generation. (2025). *Knowledge-Based Systems*. <https://doi.org/10.1016/j.knosys.2025.114600>.
27. Li, C., Taniguchi, Y., Lu, M., & Konomi, S. (2021). Few-shot font style transfer between different languages. *IEEE/CVF Winter Conference on Applications of Computer Vision*. (p. 433-442). https://openaccess.thecvf.com/content/WACV2021/papers/Li_Few-Shot_Font_Style_Transfer_Between_Different_Languages_WACV_2021_paper.pdf.
28. Fontjoy. (n. d.). Generate font combinations with deep learning. <https://fontjoy.com/>.
29. Adobe. (n. d.). Identify fonts using Retype. Adobe Help Center. <https://helpx.adobe.com/illustrator/desktop/design-with-text/fonts-and-scripts/identify-fonts-using-retype.html>.
30. MyFonts. (n. d.). WhatTheFont. <https://www.myfonts.com/pages/whatthefont>.
31. Fontspring. (n. d.). Fontspring Matcherator. <https://www.fontspring.com/matcherator>.
32. Adobe. (n. d.). Text to Vector Graphic in Adobe Illustrator. <https://www.adobe.com/products/illustrator/text-to-vector-graphic.html>.
33. Recraft. (n. d.). Recraft. <https://www.recraft.ai/>.
34. Володимир Юрчишин. Мистецтво книги: Вибране з колекції Музею книги і друкарства України: Каталог. (2015). Музей книги і друкарства України.
35. Дудник, І. (2013). Ранній Юрчишин. Кирилівські читання. <https://cyreading.blogspot.com/2013/09/blog-post.html>.
36. Ламонова, О.В. (2012). Мистецтво книжкового оформлення: Творчість Володимира Юрчишина. *Образотворче мистецтво*, 12, 76-80.