

рення та розгортання для загального користування моделі згорткової архітектури, що вирішує завдання багатокласової класифікації стану рослин за кольоровими зображеннями їх листя.

Висновки. Дослідження проведено з метою вирішення задачі діагностики хвороб рослин. Як вхідні дані використано датасет, що включає зображення здорових рослин і зображення для п'яти видів захворювань рослин. Розроблено інформаційну систему, що включає такі етапи: автоматичну сегментацію, отримання ознак, класифікацію методами машинного навчання. Проведено попередню обробку вхідних зображень з метою забезпечення повноти сегментації листя та ознак, що характеризують їх аномалії, які пов'язані з захворюванням рослин. Розроблена модель може бути використана при проектуванні програмних комплексів, які будуть орієнтовані на вирішення задач класифікації об'єктів, що задані у вигляді зображень.

Список використаних літературних джерел

- [1] Акинин М.В., Никифоров М.Б., Таганов А.И. Нейросетевые системы искусственного интеллекта в задачах обработки изображений / – М.: Горячая линия – Телеком, 2017. – 152 с.
- [2] Содем Ян. Программирование компьютерного зрения на языке Python / – Москва: ДМК Пресс, 2016. – 312 с.

Дмитро Квик, Світлана Яцишин*

магістрант кафедри КН, НЛТУ України, Львів, Україна. kvyk@nltu.edu.ua.

* завідувач кафедри ІПЗ, НЛТУ України, Львів, Україна.

<https://orcid.org/0000-0001-5200-4837>, yatsyshyn@nltu.edu.ua.

Мобільний застосунок для формування зображення за ескізом з використанням методів штучного інтелекту

Анотація – У результаті виконання проекту було охарактеризовано метод генерування зображень з вихідного нарису та досліджені доступні підходи та алгоритми для програмної реалізації. На основі цього був розроблений програмний продукт, який здатен виконувати генерацію зображень на основі вхідних нарисів. Проведене тестування підтвердило правильність та ефективність

генерації зображень з використанням розробленого програмного забезпечення. Кросплатформений додаток, розроблений для платформ Android, IOS та Web, надає користувачам зручний інструмент для створення унікальних зображень на основі ескізів. Розроблена програмна система відкриває нові можливості для творчості та використання штучного інтелекту у генерації зображень, сприяючи розширенню сфери застосування в мистецтві, дизайні та інших галузях.

Програмна система реалізована з використанням мов програмування Python і Dart, фреймворків Flask і Flutter та бібліотек Tensorflow і Keras.

Ключові слова – Python, Flask, Dart, Flutter, TensorFlow, Keras, штучний інтелект.

Генерація зображень за допомогою штучного інтелекту надає нові можливості для художників, дизайнерів та творчих професіоналів, що допомагає збагачувати їх творчий процес. Також ця технологія дозволяє автоматизувати процес створення графічних елементів у графічному дизайні та рекламі, що прискорює розробку та знижує витрати.

Нейронні мережі – це математичні моделі, що намагаються відтворити роботу людського мозку. Вони складаються зі з'єднаних штучних нейронів, які працюють разом для вирішення певної задачі. Нейрони взаємодіють між собою, передаючи інформацію через ваги, які визначають силу зв'язку між нейронами. Їх можна використовувати задля вирішення різноманітних завдань, серед яких можна виділити класифікацію зображень та їхню генерацію, визначення об'єктів, машинний переклад, розпізнавання мови та багато інших.

Для генерації зображень в основному використовуються моделі генеративно-змагальних мереж (GAN) – пари нейронних мереж, генератора і дискримінатора, які змагаються один з одним. Генератор намагається створити реалістичні зображення, тоді як дискримінатор намагається відрізнити справжні зображення від зображень, згенерованих генератором. Ця конкурентна взаємодія між двома моделями допомагає покращити якість згенерованих зображень з часом [1].

Існує кілька видів GAN-моделей, які використовуються для генерації зображень за допомогою штучного інтелекту. Серед них: Deep Convolutional GAN (DCGAN), що використовує глибокі згорткові мережі для генерації високоякісних зображень, Conditional GAN (cGAN), що розширює базову GAN-модель, додаючи можливість контролювати генерацію зображень за допомогою умовних міток [2], Progressive GAN (ProGAN), яка є моделлю, що прогресивно збільшує якість зображень під час навчання та CycleGAN, що використовується для перетворення зображень між двома різними доменами без наявності парних тренувальних даних [3].

З наведених моделей, під окреслену предметну область найбільше підходить cGAN модель, адже вона дозволяє подати нарис користувача, виконаний у зовнішньому додатку, у якості вхідного зображення, задати необхідні параметри та отримати максимально наближене до вхідних параметрів зображення. Умовна генеративно-змагальна мережа (cGAN) є розширенням базової GAN-моделі, яке додає можливість контролювати генерацію зображень за допомогою умовних міток або параметрів. Вона відкриває широкі можливості для контрольованої генерації зображень, дозволяючи створювати специфічні зображення з урахуванням вхідної умови та дозволяє зробити модель більш гнучкою та універсальною для різних завдань, де потрібно контролювати процес генерації. Однією з її переваг є можливість створювати унікальні зображення, які задовольняють специфічні вимоги або креативні ідеї користувача.

Проте, у cGAN є деякі виклики. Одним з них є необхідність наявності достатньої кількості маркованих даних, що позначають умови генерації, для тренування моделі. Відсутність або недостатність таких даних може призвести до обмежень у контролі генерації [1].

Розробка кросплатформеного додатку для створення ескізів та генерації зображень з використанням штучного інтелекту вимагає вибору відповідних технологій. Для цього проекту використовувалися мови програмування Dart та Python, а також фреймворки Flutter і Flask.

Окрім цього, було обрано TensorFlow для розробки програмної системи, оскільки ця відкрита бібліотека машинного навчання, розроблена Google, надає широкий спектр функціональності та підтримує різні мови програмування, а також Keras як високорівневу бібліотеку для швидкого створення моделей нейронних мереж з простим API та вбудованою підтримкою розподіленого навчання [6].

Клієнтський додаток розроблено з використанням Material3 для створення сучасних інтерфейсів. Material3 пропонує нові компоненти, анімації та кольорову палітру. На сторінці малювання є кнопки Clear, Undo і Save (рис 1, а). Немає можливості налаштувати параметри курсора і завантажити зображення з файлів. Після малювання користувач вводить опис, зберігає зображення і отримує підтвердження (рис 1, б). У вкладці Browse користувач може переглядати і збільшувати зображення (рис 1, в).

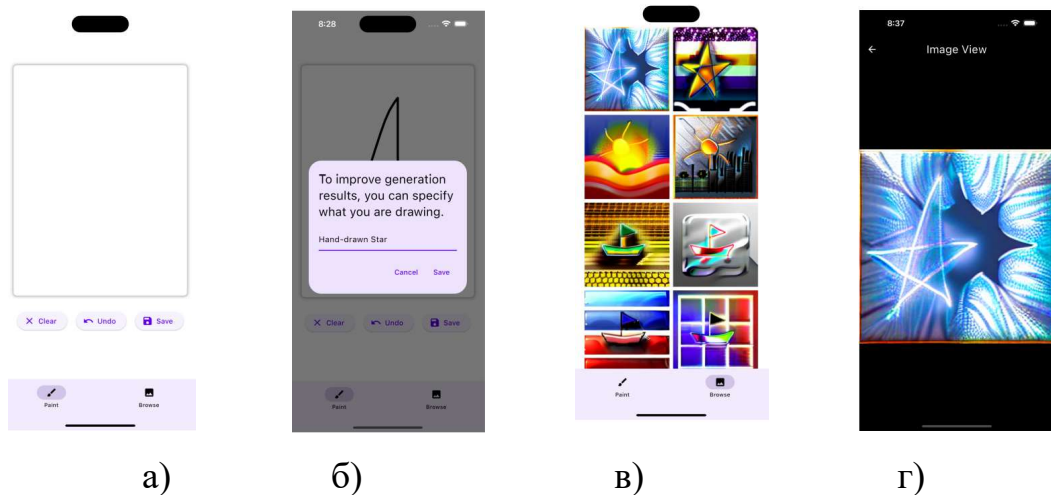


Рисунок 1 – Додаток: а) початковий екран; б) діалог для введення опису намальованого зображення; в) вкладка Browse; г) перегляд згенерованого зображення

Розроблений додаток для створення ескізів та генерації зображень з використанням штучного інтелекту надає користувачам можливість створювати унікальні ескізи, налаштовувати параметри курсору та вказувати опис зображення для покращення результатів генерації. Після генерації зображення, користувачам доступний перегляд отриманих результатів (рис 1, в). Система до-

ступна для платформ Android, iOS та Web, забезпечуючи користувачам можливість використовувати її на різних пристроях.

Висновок. Розроблений додаток надає користувачу можливість створювати унікальні зображення, використовуючи вбудовані засоби створення ескізів. Це відкриває безліч нових можливостей у відтворенні віртуальних світів, розширеній реальності, створенні реалістичних персонажів та об'єктів, візуалізації наукових досліджень та багато інших сфер.

Список використаних літературних джерел

- [1] Navin Manaswi Generative Adversarial Networks with Industrial Use Cases: Learning how to build GAN applications for Retail, Healthcare, Telecom, Media, Education, and HRTech – 2020, 134 с.
- [2] Alberto Miola Flutter Complete Reference: Create beautiful, fast and native apps for any device – 2020, 760 с.
- [3] Hany John Hands-On Generative Adversarial Networks with PyTorch 1.x: Implement next-generation neural networks to build powerful GAN models using Python / John Hany, Greg Walters – 2019, 312 с.
- [4] Getting Started with Flutter Bloc Pattern [електронний ресурс]. – режим доступу: <https://www.mitrais.com/news-updates/getting-started-with-flutter-bloc-pattern/> (2022).
- [5] Luciano Ramalho Fluent Python: Clear, Concise, and Effective Programming 2nd Edition - 2022, 1012 с.
- [6] Liu, Ruiyuan & Mao, Jian. (2018). Research on Improved Canny Edge Detection Algorithm. MATEC Web of Conferences. 232. 03053. 10.1051/mateconf/201823203053.

Максим Борух, Ігор Капран*

магістрант кафедри КН, НЛТУ України, Львів, Україна.

* старший викладач кафедри КН, НЛТУ України, Львів, Україна.

<https://orcid.org/0000-0001-9666-9531>. kapran@nltu.edu.ua