

УДК 004.925

<https://doi.org/10.35546/kntu2078-4481.2022.1.6>

О.М. МІХАЙЛУЦА

Запорізький національний університет, м. Запоріжжя

ORCID: 0000-0003-2935-7997

Є.В. МЕЛІХОВ

Запорізький національний університет, м. Запоріжжя

ORCID: 0000-0002-9856-9291

Д.К. БЕЗОТОСНИЙ

Запорізький національний університет, м. Запоріжжя

Ю.О. ЛИМАРЕНКО

Запорізький національний університет, м. Запоріжжя

ORCID: 0000-0002-1643-6939

АНАЛІЗ СУЧАСНИХ ДОСЯГНЕНЬ ПРОГРАМНОЇ ІНЖЕНЕРІЇ ДЛЯ СТВОРЕННЯ УНІВЕРСАЛЬНОГО АЛГОРИТМУ ASCII-ART КОНВЕРТАЦІЇ

У роботі розглядаються переваги використання ASCII-art технологій. Широка доступність і популярність текстових комунікаційних каналів спонукають до використання ASCII-art для представлення зображень. Проаналізовано сучасні підходи створення рисунку у стилі ASCII-art. В ході проведеного дослідження були показані різні способи створення ASCII-рисуноків за допомогою комп'ютерних програм. Виявлено ефективні сучасні алгоритми конвертації зображень у ASCII-формат. Запропоновано досконалий алгоритм конвертації довільного зображення у ASCII-формат, який надає можливість оцінити всі переваги ASCII-art, такі як: невеликий розмір рисунку, універсальність, інтернаціональність.

Спроектовано та розроблено досконалий та зручний мобільний застосунок, що надає можливість завантаження цільового зображення з внутрішнього сховища або миттєвого фото з камери та оптимізує процес конвертації зображень в ASCII-art. Спроектована система є програмним додатком, який надає більш розширений функціонал порівняно з існуючими аналогами, та незалежним продуктом, який не базується на будь-яких існуючих системах. Для створення програмного застосунку використана статично типізована об'єктно орієнтована мова програмування Kotlin, у якості інтегрованого середовища розробки обрано AndroidStudio. Додаток призначено для одноосібного користування. Алгоритм конвертації зображень у ASCII-формат складається з наступних етапів: підготовка зображення для обробки, що дозволяє усунути некоректне масштабування зображення у самому застосунку; обробка зображення.

Проведено експерименти з самостійного створення ASCII-art рисуноків. Отримані результати свідчать про те, що створення модулів для конвертації ASCII art зображень дозволить зменшити обсяг інформації, якій зберігається на серверах, залучити нових користувачів завдяки слідуванню молодіжному напрямку разом із дотриманням світових тенденцій та популяризувати ASCII-графіку серед вже зареєстрованих користувачів. Застосування розробленого додатку дозволить творчим особистостям розширити шляхи самореалізації за допомогою сучасних технологій долаючи обмеження, спричинені початком глобальної пандемії COVID-19. Рисунки символами – це універсальний засіб спілкування, покликаний поєднати людство, подолати бар'єри природних мов та державні кордони.

Ключові слова: ASCII art, конвертація зображення, мобільний застосунок, проектування, Kotlin.

Е.Н. МІХАЙЛУЦА

Запорізький національний університет, г. Запоріжжя

ORCID: 0000-0003-2935-7997

Е.В. МЕЛІХОВ

Запорізький національний університет, г. Запоріжжя

ORCID: 0000-0002-9856-9291

Д.К. БЕЗОТОСНИЙ

Запорізький національний університет, г. Запоріжжя

Ю.А. ЛИМАРЕНКО

Запорізький національний університет, г. Запоріжжя

ORCID: 0000-0002-1643-6939

АНАЛИЗ СОВРЕМЕННЫХ ДОСТИЖЕНИЙ ПРОГРАММНОЙ ИНЖЕНЕРИИ ДЛЯ СОЗДАНИЯ УНИВЕРСАЛЬНОГО АЛГОРИТМА ASCII-ART КОНВЕРТАЦИИ

В работе рассматриваются преимущества использования ASCII-art технологий. Широкая доступность и популярность текстовых коммуникационных каналов побуждают к использованию ASCII-art для представления изображений. Проанализированы современные подходы к созданию рисунка в стиле ASCII-art. В ходе проведенного исследования были показаны разные способы создания ASCII-рисунков с помощью компьютерных программ. Выявлены эффективные современные алгоритмы конвертации изображений в формат ASCII. Предложен совершенный алгоритм конвертации произвольного изображения в формат ASCII, который позволяет оценить все преимущества ASCII-art, такие как: небольшой размер рисунка, универсальность, интернациональность.

Спроектировано и разработано универсальное мобильное приложение, которое дает возможность загрузки целевого изображения из внутреннего хранилища или мгновенного фото с камеры и оптимизирует процесс конвертации изображений в ASCII-art. Для создания программного приложения использован статически типизированный объектно-ориентированный язык программирования Kotlin, в качестве интегрированной среды разработки выбрана AndroidStudio. Приложение предназначено для единичного использования. Алгоритм конвертации изображений в ASCII-формат состоит из следующих этапов: подготовка изображения для обработки, что позволяет устранить некорректное масштабирование изображения в самом отношении; обработка изображения.

Проведены эксперименты по самостоятельному созданию ASCII-art рисунков. Полученные результаты свидетельствуют о том, что создание модулей для конвертации ASCII art изображений позволит снизить объем информации, хранящейся на серверах, привлечь новых пользователей благодаря следованию молодежному направлению вместе с соблюдением мировых тенденций и популяризовать ASCII-графику среди уже зарегистрированных пользователей. Применение разработанного приложения позволит творческим личностям расширить пути самореализации с помощью современных технологий, преодолевая ограничения, вызванные началом глобальной пандемии COVID-19. Рисунки символами – это универсальное средство общения, призванное соединить человечество, преодолеть барьеры естественных языков и государственные границы.

Ключевые слова: ASCII art, конвертация изображения, мобильное приложение, проектирование, Kotlin.

O. MIKHAILUTSA

Zaporizhzhia National University, Zaporizhzhia

ORCID: 0000-0003-2935-7997

Y. MELIKHOV

Zaporizhzhia National University, Zaporizhzhia

ORCID: 0000-0002-9856-9291

D. BEZOTOSNYI

Zaporizhzhia National University, Zaporizhzhia

Y. LYMARENKO

Zaporizhzhia National University, Zaporizhzhia

ORCID: 0000-0002-1643-6939

ANALYSIS OF MODERN ADVANCES IN SOFTWARE ENGINEERING FOR CREATING A UNIVERSAL ASCII-ART CONVERSION ALGORITHM

The paper considers the advantages of using ASCII-art technologies. The wide availability and popularity of textual communication channels encourages the use of ASCII-art to represent images. Modern approaches to creating a drawing in the ASCII-art style are analyzed. In the course of the study, various ways of creating ASCII drawings using computer programs were shown. Efficient modern algorithms for converting images into ASCII format have been identified. A perfect algorithm for converting an arbitrary image into ASCII format is proposed, which allows you to evaluate all the advantages of ASCII-art, such as small size of the picture, universality, internationality.

Designed and developed a perfect and convenient mobile application that allows you to download the target image from the internal storage or instant photo from the camera and optimizes the process of converting images to ASCII-art. The designed system is a software application that provides more advanced functionality than existing analogues and is an independent product that is not based on any existing systems. To create a software application, a statically typed object-oriented programming language Kotlin was used, and AndroidStudio was chosen as an integrated development environment. The application is intended for sole use. The algorithm for converting images into an ASCII format consists of the following steps: preparing an image for processing, which makes it possible to eliminate incorrect image scaling in the ratio itself and image processing.

Experiments were carried out on the independent creation of ASCII-art drawings. The obtained results indicate that the creation of modules for converting ASCII art images will reduce the amount of information

stored on servers, attract new users by following the youth direction along with global trends and popularize ASCII graphics among already registered users. The use of the developed application will allow creative individuals to expand the ways of self-realization with the help of modern technologies, overcoming the limitations caused by the onset of the global COVID-19 pandemic. Drawings with symbols are a universal means of communication designed to connect humanity, overcome the barriers of natural languages and state borders.

Keywords: ASCII-art, image conversion, mobile application, Kotlin.

Постановка проблеми

Технологічні, інноваційні процеси та відкриття мають колосальний вплив на розвиток суспільства та наше повсякденне життя. Сучасні наукові відкриття змушують забути про важливість творчості. Історично підтверджено, що люди створюють мистецтво для будь-якої нової технології. Одним із прикладів цього є мистецтво, створене для спілкування за допомогою рисунків. У разі, коли необхідно відправити не деталізоване зображення в чат-форумі без підтримки медіа файлів або зберігати його в класичній реляційній базі даних, можна використовувати застосунки, які конвертують звичайну картинку у набір текстових символів - ASCII-графіку. У сучасному цифровому світі важко уникнути мистецтва ASCII. Це і смайлики, і аватарки, і інтернет мему, а також фотографії, зроблені з тексту: букв, цифр і спеціальних символів з таблиці Юнікоду. Актуальність у тому, що мистецтво створення картинок із знаків продовжує розвиватися, попри появу потужних графічних редакторів. З розвитком соціальних мереж мистецтво малювати символами набуває все більшої популярності у нас у країні та у світі, адже малюнки символами – це відмінний спосіб висловити свої емоції, це мова, яка зрозуміла без перекладу.

Сьогодні спеціалістами в області інформаційних технологій розробляються мобільні додатки, які дозволяють вирішувати величезну кількість завдань. Протягом останніх років по-казник, що характеризує рівень запиту на мобільні пристрої, постійно зростає. Така статистика дозволяє зробити висновок про те, що розробка мобільних додатків є актуальною і доцільною.

Отже перед роботою ставиться задача аналізу сучасних технологій створення рисунку у стилі ASCII-art, а також їх застосування для проектування системи конвертації довільного зображення у ASCII-формат.

Аналіз останніх досліджень і публікацій

Виникнення стандарту ASCII (American Standard Code for Information Interchange) датується 60-ми роками ХХ століття. Він є формою образотворчого мистецтва, що використовує символи ASCII для представлення зображень [1]. При цьому застосовується палітра, яка складається з буквених, цифрових символів та знаків пунктуації з 95 символів таблиці ASCII [2].

У перших поколіннях комп'ютерів вже були можливості відображати графіку. Популярним було відображення графіки за допомогою текстових символів та виведення отриманих зображень на екран терміналу та принтер. На сучасному етапі розвитку комп'ютерної техніки та мережевих систем існує велика кількість форматів та можливостей для передачі фотографій та малюнків. Однак досить часто виникає необхідність для відтворення зображень у «незвичному» текстовому форматі, що викликає цілу низку складних технічних проблем. Зрозумілим постає бажання автоматизації подібного перетворення, для реалізації чого в останні роки науковцями запропоновано декілька методів, підходів та алгоритмів. Один з таких підходів, а саме автоматичне перетворення бінарних зображень у форматі ASCII було розглянуто у роботі [3] з точки зору проблеми оптимізації. Застосовувалося невід'ємне матричне розкладання з метою перетворення чорно-білих зображень в ASCII-малюнки.

Іншим перспективним підходом для автоматичного створення ASCII-зображень, який досліджувався авторами у роботі [4], є застосування структур. Його сутність полягає в апроксимації структури основної лінії вмісту еталонного зображення з формою символів. З оглядом на людське сприйняття зображень, розроблена схема їх виділення, яка базується на моделі конгруентності фаз. Використання штучного інтелекту для підвищення якості автоматизованого перетворення досліджено у роботі [5], зокрема навчання згорткової мережі проводилося на малюнках художників.

Більш складним завданням для ASCII-art як форми мистецтва є перетворення фотографій, оскільки досить важко у цьому випадку знаходити основну структуру для відтворення. Подоланню цієї проблеми присвячена стаття [6], у якій автори пропонують новий алгоритм модуляції для відокремлення структури зображення від текстури.

Всі перераховані вище інновації дозволяють конвертувати ASCII-art зображення швидше і зручніше. Бурхливий розвиток соціальних мереж та месенджерів, а також їх домінування у мобільних пристроях зумовлюють створювати застосунки, які базуються на мобільних платформах Android та IOS. Проектування та розробка досконалого та зручного програмного застосунку надасть можливість оцінити всі переваги ASCII-art, а саме:

- невеликий розмір рисунку [7];
- універсальність [8];

- інтернаціональність [9].

Аналіз сучасних підходів до створення ASCII-art

На сьогоднішній день існує достатньо літератури, що описує проблему конвертації довільного зображення, існує також кілька додатків, що вирішують дану задачу, наприклад додаток від команди розробників «N Studio» – Ascii Art Generator, застосунок для конвертації зображень у ASCII-art формат у вигляді тексту або картинки – ASCII Image Generator, програмний додаток для конвертації зображень у ASCII-art формат, якій підтримує конвертування вхідного зображення у текст або картинку – Photo to ASCII Text Art.

До найбільш суттєвих недоліків розглянутих застосунків можна віднести:

- Відсутність можливості конвертації зображення у текст (вихідні файли створюються лише у форматі зображень).
- Відсутність можливості налаштування конвертації (перетворення фактично існуючого зображення у ASCII-art без функції вибору символів або режимів).
- Меню налаштувань надто неінформативне. Наприклад, немає необхідності налаштування кольору символів, адже ASCII-art повинно зберігати відтінки базового зображення без їхньої зміни. А налаштування кількості символів взагалі може бути автоматичним (у тому випадку, коли користувач власноруч чітко визначить бажані символи, їх кількість буде визначена виходячи з переліку, а у випадку програмного заповнення – автоматично, виходячи з самого зображення).
- Недостатня якість цільового зображення. Після виконання конвертації картинка не має високої чіткості та єдності. Можна чітко розрізнити дефекти (у вигляді горизонтальних смуг), які на вхідному зображенні відсутні (можливо наявні недосконалості або помилки у роботі алгоритму).
- Відсутність зручного інструменту для форматування зображення. Після обрання вхідного зображення у середині екрану з'являється область виділення. Проте, можливість пересування самої зони по робочій поверхні відсутня.

Вивчивши історію тестових малюнків, провівши експерименти з самостійного створення ASCII-art і знайшовши застосування цього виду мистецтва в сучасному світі Інтернету, можна сказати, що ASCII-art сьогодні затребуване в рекламі та в соціальних мережах.

Формулювання мети дослідження

Метою дослідження є вивчення переваг існуючих аналогів та методів конвертації зображень у ASCII-формат для проектування та розробки універсального мобільного застосунку, що надає можливість завантаження цільового зображення з внутрішнього сховища або миттєвого фото з камери та оптимізує процес конвертації зображень в ASCII-art.

Викладення основного матеріалу дослідження

Використання сучасних досягнень програмної інженерії для створення досконалого алгоритму ASCII-art конвертації

Застосування чи модифікація існуючих рішень, зважаючи на вартість їх придбання та надлишкову або місцями недостатню функціональність, не є розумним рішенням для вирішення поставлених задач. Доцільним є розроблення програмного комплексу для якісної та зручної конвертації довільного зображення у ASCII-формат, діаграма варіантів використання для якого наведена на рисунку 1.

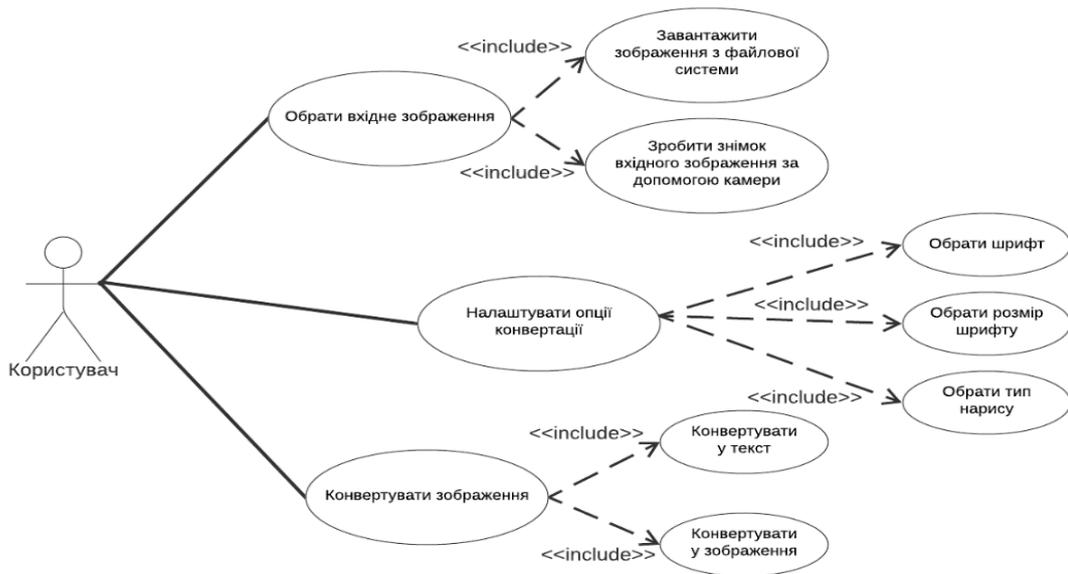


Рис.1. Діаграма варіантів використання

Спроекована система є програмним додатком, який надає більш розширений функціонал порівняно з існуючими аналогами, та незалежним продуктом, який не базується на будь-яких існуючих системах і відповідає наступним функціональним вимогам:

- завантаження початкового зображення для конвертації з внутрішнього сховища або миттєвого фото з камери;
- можливість конвертації обраного зображення в необхідний графічний формат або в текстовий файл;
- збереження результату (вихідного файлу) до внутрішнього сховища (в довільну директорію на пристрої);
- копіювання результату конвертації (вихідного зображення або тексту) у буфер обміну;
- налаштування опцій конвертації цільового зображення, а саме шрифту, розміру символів та накреслення тексту;
- інформування клієнта про стан виконання конвертації (перегляд стану конвертації за допомогою індикатора виконання).

Додаток призначено для одноосібного користування, тобто у інформаційній системі є лише один клас з можливостями користувача.

Алгоритм конвертації зображень у ASCII-формат складається з наступних етапів:

- підготовка зображення для обробки;
- обробка зображення.

Вхідними даними для конвертації завжди є зображення та набір символів, а вихідними даними алгоритму є відформатований текст. Однак слід відзначити, що користувачам результат цікавий саме у форматі зображення. Саме тому перед завершенням етапу перетворення вхідних даних у вихідні, отриманий текст друкується на екрані у вигляді картинки. Хоча, якщо є така необхідність, результат можна відобразити і у форматі текстового рядку.

Запропонована версія алгоритму була реалізована для мобільних пристроїв під керуванням ОС «Android». Перша частина підготовки зображення для конвертації пов'язана з усуненням проблеми, яка стосується відмінності співвідношення довжини та ширини екрану в різних пристроях. Оскільки ця проблема призводить до некоректного масштабування зображення у самому застосунку, користувач власноруч обирає прямокутну область вхідного зображення, довжина та ширина якої прямо пропорційні параметрам екрану конкретного пристрою.

Друга частина підготовки зображення для конвертації полягає у порівнянні співвідношення ширини вхідного зображення та екрану пристрою у пікселях за допомогою формули:

$$\text{imageWidth} \leq \frac{\text{screenWidth}}{\text{scale}},$$

де *imageWidth* – ширина вхідного зображення, *screenWidth* – ширина екрану застосунку, а *scale* – це число, яке позначає ширину символу моноширинного шрифту певного розміру (у застосунку використовуються значення 12 для розміру шрифту, якому відповідає значення 7 для змінної *scale*).

У разі, якщо умова виконується, то зображення залишається без змін. В іншому випадку зображення перемасштабується нові розміри, а саме:

$$\text{resultWidth} \leq \frac{\text{screenWidth}}{\text{scale}},$$

$$\text{resultHeight} \leq \text{resultWidth} \frac{\text{imageHeight}}{\text{imageWidth}},$$

де resultWidth – ширина вихідного зображення, screenWidth – ширина екрану застосунку, scale – це число, яке позначає ширину символу моноширинного шрифту певного розміру (у застосунку використовуються значення 12 для розміру шрифту, якому відповідає значення 7 для змінної scale), resultHeight – висота вихідного зображення, imageHeight – висота вхідного зображення.

Процес обробки підготованого зображення полягає у необхідності розбиття усього зображення на комірки розміром 1x2 пікселів. По-суті, другий (нижній) піксель в комірці ігнорується, оскільки якщо розбити зображення на квадратні комірки і замінювати кожен символом ASCII, який умовно займає 2 комірки, то вихідне зображення вийде в два рази більшим за вертикаллю. У внутрішньому циклі виконується прохід по коміркам в рядку зліва на право, а у зовнішньому циклі виконується прохід по рядках зверху вниз. На конкретній ітерації потрібно замінити два пікселя в комірці на деякий символ із вхідного рядку. Процес роботи алгоритму відбувається наступним чином. Береться перший піксель комірки та переводиться у *Grayscale mode*. При цьому утворюється *Grayscale picture* – зображення, у якому різні кольори передаються за допомогою відтінків сірого кольору. Так як значення пікселю – це ціле число, то перші два біти позначають значення складової *blue*, наступні два – складової *green*, останні два – складової *red*. Виконавши виділення цих значень на змінні, перетворення пікселя в *Grayscale mode* здійснюється за формулою:

$$RGB[A] \text{ to Gray} : Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

Отримане значення знаходиться в діапазоні від 0 до 255, де 0 відповідає білому кольору, а 255 – чорному, крім того, чим менше отримане значення, тим відтінок сірого світліший, а чим більше – темніший. Якщо провести аналогію з символами ASCII, то деякі з них займають більшу площу в своїй комірці і здаються «темнішими», а деякі займають меншу площу у порівнянні з іншими і здаються «світлішими». Обравши масив таких символів, наприклад у кількості n , і відсортувавши їх від найтемнішого до найсвітлішого, розділивши 255 на n проміжків, де порядковий номер проміжку буде відповідати індексу в масиві символів – можна замінити отримане значення *Gray* на значення із масиву, яке відповідає цьому проміжку і додати цей символ до результуючого рядку. А у випадку роботи з *Colored ASCII*, потрібно зберегти початковий піксель, тобто його колір і на етапі промальовування тексту замінити чорний колір конкретного символу на збережений колір пікселя цієї комірки. В кінці внутрішнього циклу, тобто перед новим рядком з комірками, в результуючий рядок також потрібно додати символ кінця рядку. В результаті отримуємо текст, у якому за рядками записані символи, кожен з яких відповідає комірці вхідного зображення.

Спираючись на вищезазначений алгоритм та беручи до уваги функціональні вимоги, для створення програмного застосунку використовується статично типізована об'єктно орієнтована мова програмування Kotlin [10]. Оскільки розроблений додаток створювався для застосування на мобільних пристроях, то у якості інтегрованого середовища розробки обрано Android Studio. Воно є офіційно інтегрованим середовищем розробки для створення застосунків під операційну систему Android на основі IntelliJ IDEA.

Проект програмного забезпечення для системи конвертації зображень у ASCII-art

Кінцевий продукт являє собою програмний застосунок, який є конвертором зображень у ASCII-формат. Вікно головної сторінки містить назву програмного застосунку, поле виводу медіа-контенту (у разі обрання цільової картини вона буде тут відображена, а після виконання конвертації буде наведене вихідне зображення). Окрім цього, вікно головної сторінки має чотири функціональні кнопки. Три з них знаходяться під полем виводу медіа-контенту та відповідають безпосередньо за обрання вхідного зображення, його конвертацію та збереження у вигляді зображення до пам'яті пристрою. Четверта знаходиться у правому верхньому куті екрану та відкриває субменю з трьох пунктів, за допомогою яких можна: відкрити вікно налаштувань, скопіювати вихідне зображення у буфер обміну у вигляді картини або у текстовому форматі.

Перед виконанням конвертації користувач обирає вхідне зображення (SELECT) та бажану картинку за допомогою системних утиліт (графічний переглядач, файловий менеджер). Після її погодження програмний застосунок надає можливість користувачу виокремити необхідну йому область та завантажити її до додатку. Результатом виконаних дій буде вхідне зображення (поле виводу медіа-контенту).

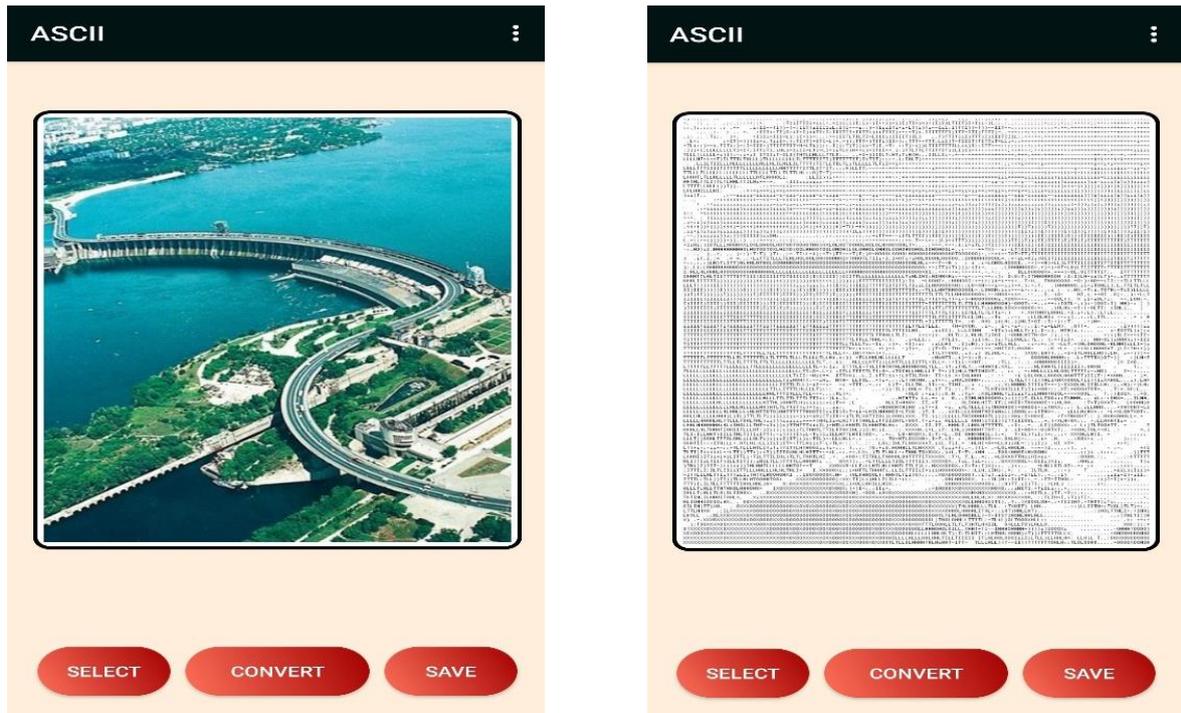


Рис.2. Конвертація зображення у ASCII-art

Після цього є можливість або відразу конвертувати зображення у ASCII-art, або попередньо налаштувати параметри конвертації (також у програмному застосунку передбачена можливість для вибору бажаних функцій навіть перед обранням вхідної картинки). У разі автоматичного виконання конвертації вихідна картинка одночасно відображається у полі виводу медіа-контенту (рис. 2). Вихідне зображення може бути збережене (SAVE) або скопійоване через використання кнопки виклику субменю. Окрім того, користувач може налаштувати цільовий файл згідно власних потреб, адже у різних ситуаціях (або в залежності від використовуваної платформи) у кожного зображення може бути окреме призначення. Для присвоєння цільовій картинці необхідних параметрів користувач повинен, за допомогою субменю, відкрити сторінку налаштувань конвертації вихідного зображення. Після встановлення бажаних характеристик та погодження зі змінами, користувачу знов відкривається головна сторінка застосунку.

На відкритій сторінці користувач може обрати один з трьох параметрів (рис. 3):



Рис. 3. Вікно налаштувань

1. «Colored ASCII» – кольоровий формат вихідного зображення, який має назву ANSI-art. Його особливість полягає в тому, що ASCII-символи, з яких складається картинка, мають відтінки оригінального малюнку, що дозволяє більш детально передавати його зміст. Проте, такий файл не можна зберегти у вигляді тексту. Справа в тому, що більшість редакторів, сервісів та платформ підтримують символи лише чорного кольору, в той час, як зображення ANSI-формату є кольоровим.

2. «Increase contrast of the original image» – збільшення контрастності зображення для покращення якості конвертації. Використовується у випадку, коли вихідний файл є не дуже чітким. Подібна ситуація може відбутися при використанні картинки із плавним переходом кольорів, що не дає змогу людському оку виокремити елементи зображення після виконання конвертації. Задля уникнення цього випадку на вхідній картинці підвищується рівень контрастності (для збереження чіткості разом із цим збільшується яскравість), після чого відбувається звичайний процес конвертації.

3. «Use custom characters» – задати власні символи, з яких буде будуватися вихідна картинка. Ця функція може знадобитися при створенні комерційного зображення. В цьому випадку, у якості символів, які використовуються при конвертації, можна задати щось, що пов'язано з замовником ASCII-art. Наприклад, назву підприємства або рекламованого продукту. Все зображення буде побудоване з використанням зазначених символів.

Висновки

В ході проведеного дослідження були показані різні способи створення ASCII-рисуноків за допомогою комп'ютерних програм. В даній роботі було запропоновано досконалий алгоритм ASCII-art конвертації та розроблено програмний модуль для його реалізації. Отримані результати свідчать про те, що створення модулів для конвертації ASCII-art зображень дозволить зменшити обсяг інформації, якій зберігається на серверах, залучити нових користувачів завдяки слідуванню молодіжному напрямку разом із дотриманням світових тенденцій та популяризувати ASCII-графіку серед вже зареєстрованих користувачів. Застосування розробленого додатку дозволить творчим особистостям розширити шляхи самореалізації за допомогою сучасних технологій долаючи обмеження, спричинені початком глобальної пандемії COVID-19. Рисунок символами – це універсальний засіб спілкування, покликаний поєднати людство, подолати бар'єри природних мов та державні кордони.

Список використаної літератури

1. Lewis A. ASCII 87 Success Secrets - 87 Most Asked Questions on ASCII -What You Need to Know : Emereo Pty Limited, 2014, 80p.
2. Fogelman R. ASCII Graphic Glitch Art : Graphic Glitch Art - Technology + Art + Design : Createspace Independent Publishing Platform, 2013, 372p.
3. O'Grady P.D., Rickard S.T. Automatic ASCII Art Conversion of Binary Images Using Non-Negative Constraints. IET Irish Signals and Systems Conference. IEEE, 2008, pp.186-191.
4. Xu X., Zhang L., Wong T. Structure-based ASCII art. ACM Transactions on Graphics. IEEE Transactions on Visualization and Computer Graphics. 2010, vol.29, no.4, pp.52:1–52:9.
5. Akiyama O. ASCII Art Synthesis with Convolutional Networks. 31st Conference on Neural Information Processing Systems. NIPS, 2017, pp.74-79.
6. Xu X., Zhang L., Xie M., Liu X., Qin J., Wong T. ASCII art synthesis from natural photographs. IEEE Transactions on Visualization and Computer Graphics. 2017, vol.23, no.8, pp.1910-1923.
7. Tallón-Ballesteros A.J. Fuzzy Systems and Data Mining VI : Proceedings of FSDM 2020 : IOS Press, 2020, 812p.
8. Desmond Paul H. Computer Art: Ascii Art, Fractal Art, Digital Art, Ars Electronica Center, Digital Media, Interactive Art : Books LLC, 2010, 372p.
9. Majid al-Rifaie M., Ursyn A., Ursyn T. The Art of Coding : The Language of Drawing, Graphics, and Animation : Chapman and Hall/CRC, 2020, 318p.
10. Samuel S., Bocutiu S. Learn Kotlin Programming: A comprehensive guide to OOP, functions, concurrency, and coroutines in Kotlin 1.3, 2nd Edition : Packt Publishing, 2019, 514p.

References

1. Lewis, A. (2014). ASCII 87 Success Secrets - 87 Most Asked Questions on ASCII -What You Need to Know: Emereo Pty Limited [in English].
2. Fogelman, R. (2013). ASCII Graphic Glitch Art: Createspace Independent Publishing Platform [in English].
3. Tallón-Ballesteros, A.J. (2020). Fuzzy Systems and Data Mining VI: IOS Press [in English].
4. Desmond, P.H. (2010). Computer Art: Ascii Art, Fractal Art, Digital Art, Ars Electronica Center, Digital Media, Interactive Art: Books LLC [in English].
5. Majid al-Rifaie, M., Ursyn, A., Ursyn, T. (2020). The Art of Coding: The Language of Drawing, Graphics, and Animation: Chapman and Hall/CRC-[in English].

6. O'Grady, P.D., Rickard, S.T. (2008). Automatic ASCII Art Conversion of Binary Images Using Non-Negative Constraints. *IET Irish Signals and Systems Conference*, 186-191 [in English]. doi.org/10.1049/cp:20080660.
7. Xu, X., Zhang, L., Wong, T. (2010). Structure-based ASCII art. *IEEE Transactions on Visualization and Computer Graphics*, 29(4), 52:1–52:9 [in English]. doi.org/10.1145/1833349.1778789.
8. Akiyama, O. (2017). ASCII Art Synthesis with Convolutional Networks. *31st Conference on Neural Information Processing Systems*, 74-79 [in English].
9. Xu, X., Zhang, L., Xie, M., Liu, X., Qin, J., Wong, T. (2017). ASCII art synthesis from natural photographs. *IEEE Transactions on Visualization and Computer Graphics*, 23(8), 1910-1923 [in English]. doi.org/10.1109/TVCG.2016.2569084.
10. Samuel, S., Bocutiu, S. (2019). Learn Kotlin Programming: A comprehensive guide to OOP, functions, concurrency, and coroutines in Kotlin 1.3, 2nd Edition : Packt Publishing [in English].