

УДК 004.415.25

Льовкін В.М.¹, Ступаков К.В.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. КНТ-229 НУ «Запорізька політехніка»

АЛГОРИТМИ СТИСНЕННЯ ФОТОГРАФІЙ ШЛЯХОМ СПРОЩЕННЯ КОЛЬОРОВОЇ СХЕМИ ЗОБРАЖЕННЯ

На сьогоднішній день людство дуже широко застосовує фотографування: камери, вбудовані в смартфони й інші пристрої, ще активніше розвинули дану активність порівняно з фотоапаратами, що обумовлено широким колом причин, починаючи з суто побутових та закінчуючи науково-дослідними цілями. Все це стрімко збільшує обсяги від цифрованої інформації та змушує винаходити більш оптимальні методи керування нею. Оскільки з плином часу якість фотографій неперервно покращується, зростає й обсяг, який займає зображення на накопичувальних пристроях. Але не всі зображення потребують відображення у повній якості, що підтверджує сучасне використання фотографій в соціальних мережах, де дуже часто відображається зображення у стисненому вигляді з втратами. Використання такого підходу призводить до неможливості перегляду деталей при спробі масштабування, спотворенні деталей. Тож задача розробки методів і алгоритмів стискання графічних файлів є вкрай важливою та актуальною.

Існує ряд методів і алгоритмів, які можуть використовуватися для вирішення цієї задачі, включаючи JPEG, вейвлет-перетворення, перетворення Фур'є, фрактальне стиснення, векторне квантування тощо. Концепція частини алгоритмів полягає в геометричних перетвореннях, що фактично можуть призвести до втрати сутнісних деталей, спотворення форм, а це для інформування користувача може мати значний негативний вплив. Інша група алгоритмів ґрунтується на спрощенні кольорової схеми.

Ідея, взята за основу в даній роботі для розробки програмного забезпечення і базового алгоритму, полягає у виявленні основних кольорів та заміні інших кольорів основними. Для знаходження панівних кольорів використано метод К-найближчих сусідів, оскільки наперед невідомо, які саме кольори переважатимуть на конкретному зображенні. Обраний метод порівнює кольори пікселів фотографії та встановлює їм кольори, які обрані для кожного кластера.

Стиснення відбувається за рахунок зменшення загальної кількості кольорів та їх відтінків, що спрощує внутрішню структуру зображення.

Головна перевага підходу полягає у якісному збереженні контурів фрагментів зображення, що в свою чергу не позбавляє фотографію інформативності. Простота використаного підходу сприяє його використанню на великій кількості ресурсів потенційно великого обсягу.

Алгоритм, покладений в основу стиснення фотографій в програмі, має наступну послідовність кроків: завантаження зображення, яке має бути стиснутим, зміна розмірів зображення, підготовка його для обробки, виконання кластеризації кольорів, повернення зображення до початкових розмірів, збереження копії вхідного зображення з внесеними змінами.

Програмне забезпечення створювалось за допомогою мови Python – високорівневої мови програмування, ідеологія якої спрямована на швидке розв’язання задач широкого кола зі зниженими вимогами до ресурсів за рахунок наявності великої кількості пакетів, створених іншими розробниками. У підсумку для створення застосунку використано: пакет NumPy для зміни розмірів зображень [1], пакет sklearn для кластеризації кольорів [2], пакет Pillow для роботи з файлом зображення [3].

Безпосередньо зміна розмірів зображення відбувається у 2 кроки: визначається висота, ширина, кількість каналів зображення та формується кортеж, а далі другим кроком функцією reshape багатомірний масив перетворюється у двомірний, кожний рядок якого відповідає одному пікселю зображення, а стовпець – значенню відповідного каналу кольору.

Для реалізації кластеризації створюється екземпляр класу KMeans з визначенням кількості кластерів, тобто кольорів, до яких потрібно спростити зображення. Навчання моделі реалізується через використання методу fit на масиві зображення, яке є матрицею пікселів. Після цього методом predict визначаються кластери для всіх точок в масиві зображення, ґрунтуючись на результаті навчання моделі. Наступним кроком, використовуючи атрибут cluster_centers_ створеного об’єкта, знаходяться центри кластерів, визначених при навчанні моделі. Таким чином для кожного пікселя зображення встановлюється найближчий центр кластера, після чого піксель приймає колір центру кластера.

Повернення зображення до початкових розмірів проходить у 3 кроки: за допомогою функції reshape перетворюється масив кольорів у форму вихідного зображення, виконується поелементне множення на максимальну інтенсивність відповідного каналу, за допомогою функції fromarray підпаketу Image пакету Pillow створюється новий об’єкт зображення з масиву даних типу uint8.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. NumPy documentation [Electronic resource]. – Access mode : <https://numpy.org/doc/stable/>.

2. sklearn API [Electronic resource]. – Access mode : <https://scikit-learn.org/stable/modules/classes.html>.
3. Pillow documentation [Electronic resource]. – Access mode : <https://pillow.readthedocs.io/en/stable/>.