

Олашин Олександр Олександрович

студент

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

Олашин Александр Александрович

студент

Национального технического университета Украины

«Киевский политехнический институт имени Игоря Сикорского»

Olashyn Oleksandr

Student of the

National Technical University of Ukraine

«Igor Sikorsky Kyiv Polytechnic Institute»

СИСТЕМА ПОКРАЩЕННЯ АЛГОРИТМІВ СТИСКУ ЗОБРАЖЕННЯ СИСТЕМА УЛУЧШЕННЯ АЛГОРИТМОВ СЖАТИЯ ИЗОБРАЖЕНИЯ SYSTEM FOR IMPROVING IMAGE COMPRESSION ALGORITHMS

Анотація. Запропоновано архітектуру системи на основі нейронних мереж для покращення алгоритмів стиску зображення. Проведено порівняння ефективності роботи даної системи та інших систем стиску зображення.

Ключові слова: нейронні мережі, алгоритми стиску зображення.

Аннотация. Предложено архитектуру системы на основе нейронных сетей для улучшения алгоритмов сжатия изображений. Проведено сравнение эффективности работы данной системы и других систем сжатия изображений.

Ключевые слова: нейронные сети, алгоритмы сжатия изображений.

Summary. A system architecture based on neural networks is proposed to improve image compression algorithms. The efficiency of this system and other image compression systems is compared.

Key words: neural networks, image compression algorithms.

Технології розвиваються надзвичайно стрімко. Їх розвиток прискорюється з кожним роком. Однак деякі речі як не дивно залишаються доволі інертними до змін. Яскравим прикладом є алгоритми стиску зображень (або ж формати зображень). Так PNG було створено в 1997, а JPEG і того раніше — в 1992 році. Здається це було нещодавно, однак по міркам сучасного технологічного зростання цілу вічність тому. Дані формати залишаються надзвичайно популярними і досі домінують на ринку зображень [1].

Однак на превеликий жаль вимоги до якості зображень з поступом технологій тільки зростають. Так якщо раніше основним критерієм до алгоритмів було стиснути зображення якомога сильніше, то зараз при покращенні дисплеїв якість зображення грає не менш важливу роль ніж його компактність.

Ціллю даної роботи є створення інструменту, який не змінюючи існуючі алгоритми кодування та декодування зможе представити інструмент, який суттєво покращить якість зображень та їх розмір.

Існують різні підходи до вирішення задачі покращення алгоритмів стиску зображення. Більшість дослідників в своїй роботі опираються на різноманітні типи нейронних мереж, як найбільш перспективного інструменту. Наприклад, варто відзначити роботу Dong [2], в якій запропоновано метод покращення роздільної здатності зображення на основі згорткової мережі. Іншим прикладом є стаття Google Research Group [3], де чудово описано застосування рекурентної нейронної мережі для задачі стиску зображення і приведено приклади, що ілюструють ефективність даної мережі.

У даній роботі, як основний інструмент було обрано саме згорткову нейронну мережу, через її дуже високі показники ефективності для розв'язку задачі, а також через меншу кількість ресурсів необхідних для тренування такої мережі.

Запропонована система має наступний вигляд — вона складається з двох нейронних мереж, які застосовують звичайний кодек зображення для самої операції стиску (рисунок 1).

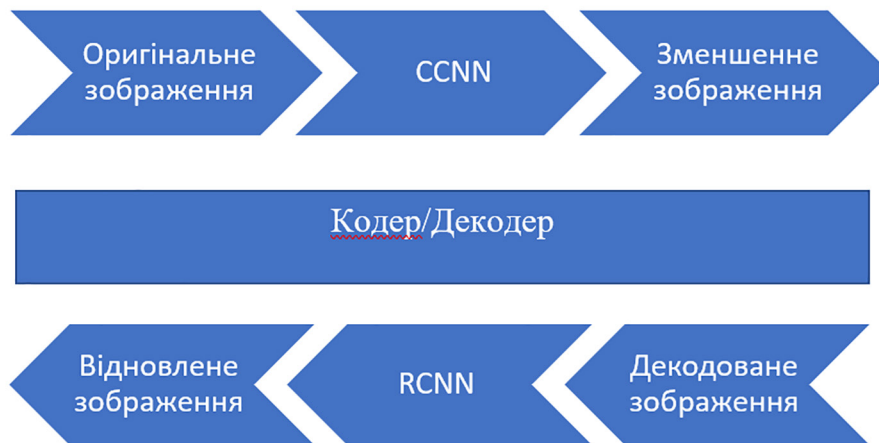


Рис. 1. Архітектура запропонованої системи



Рис. 2. Тестові зображення

Завдання першої мережі — створити зменшену версію оригінального зображення, яка зберігає всі основні структурні елементи. Потім зменшене зображення стискається за допомогою стандартного кодеку (наприклад JPEG). Для відновлення зображення використовується друга мережа — вона відновлює, покращує та збільшує декодоване зображення.

Перший шар являє собою згортковий шар, що використовує, де ReLU слугує функцією активації. Цей шар використовується для того, щоб отримати основні характеристики зображення. Фільтри мають розмір $3 \times 3 \times c$, де c — кількість каналів вихідного зображення. В даній роботі використовується 48 фільтрів (через обмеження обчислювальних потужностей). Нелінійність досягається використанням ReLU як функції активації. Наступний шар використовується для зменшення зображення та покращення отриманих характеристик. Для цього застосовується операція згортки з кроком, що дорівнює два. Фільтри даного шару мають розміри $3 \times 3 \times 48$ і ReLU також застосовується як функція активації. Для останнього третього шару використовують с фільтрів розміру $3 \times 3 \times 48$ для створення компактного представлення зображення.

RCNN складається з 18 шарів. Перший шар складається з 48 фільтрів розміром $3 \times 3 \times c$ і використовується для створення 48 карт характеристик. Як функція активації використовується ReLU. Потім йдуть 16 шарів використовуються 48 фільтрів розміром $3 \times 3 \times 48$. Варто зазначити, що в кожному з цих шарів також використовується нормалізація між

операціями згортки та ReLU. Для останнього шару використовується с фільтрів розміром $3 \times 3 \times 48$ для відновлення зображення.

Систему було натреновано на 200 зображеннях, розміру 180×180 . Для тестування було обрано 5 популярних зображень, що використовуються для тестування нейронних мереж (рисунок 2).

Систему було порівняно зі стандартним алгоритмом JPEG, а також ж ARCNN [4] — однією з найкращих нейронних мереж для стиску зображення. Також були окремо протестовані мережі по окремо та система загалом. Результати можна побачити в таблиці 1 та таблиці 2.

Якість отриманих зображень перевірялись за допомогою двох основних показників — коефіцієнту пікового сигналу до шумового співвідношення (PSNR) та індексу структурної схожості (SSIM).

Як можна побачити з наведених вище таблиць — запропонована система суттєво перевищила звичайний JPEG (при чому JPEG використовувався, як кодер/декодер в системі). Також по результативності система майже досягнула результатів ARCNN — яка вважається взірцем нейронних мереж для стиску зображень.

Висновки. Серед різноманітних підходів до вирішення задачі стиску зображень, в цій роботі було обрано той, який в перспективі повинен дати найбільшу універсальність. Так використання нейронних мереж, не як основного інструменту для стиску зображення, а як допоміжного елементу в системі зі стандартним алгоритмом стиску довело

Таблиця 1

Результати роботи за критерієм PSNR (dB)

Метод	Метелик	Дім	Папуга	Листя	Фотограф	Середній результат
JPEG	22.58	27.77	26.19	22.49	24.45	20.18
ARCNN	25.64	29.68	28.13	25.07	25.27	21.63
CCNN	22.85	27.9	27.01	22.85	24.78	20.508
RCNN	24.55	28.34	27.57	24.23	25.17	21.062
Запропонована система	25.78	28.48	28.01	25.03	25.58	21.576

Таблиця 2

Результати роботи за критерієм SSIM

Метод	Метелик	Дім	Папуга	Листя	Фотограф	Середній результат
JPEG	0.7378	0.7733	0.7581	0.7775	0.7283	0.60744
ARCNN	0.8741	0.8209	0.8446	0.8983	0.7674	0.66624
CCNN	0.7201	0.8019	0.8232	0.7761	0.7582	0.63188
RCNN	0.867	0.8189	0.8411	0.8643	0.7543	0.65572
Запропонована система	0.8732	0.8321	0.8367	0.8821	0.7677	0.66372

свою ефективність. Результати таблиць 1 та 2 показують, що фінальний результат роботи системи є надзвичайно якісним — середні показники PSNR дорівнював 21.576, що лише трохи менший показ-

нику однієї з найкращих мереж на даний момент ARCNN (21.63), а результат SSIM 0.66372 ще менше відстав від ARCNN (0.66624).

Література

1. JPEG, PNG, or GIF? The Ultimate Cheat Sheet of Image File Formats [Електронний ресурс]: [Веб-сайт]. — Електронні дані. — Jami Oetting — 28.07.2017 — Режим доступу: <https://blog.hubspot.com/agency/image-file-formats-infographic>
2. Image super-resolution using deep convolutional networks / [C. Dong, C. C. Loy, K. He, X. Tang.] // IEEE transactions on pattern analysis and machine intelligence — 2016. — Vol. 38. — PP. 295–307.
3. Image Compression with Neural Networks [Електронний ресурс]: [Веб-сайт]. — Електронні дані. — Nick Johnston, David Minnen — 29.09.2016 — Режим доступу: <https://research.googleblog.com/2016/09/image-compression-with-neural-networks.html>
4. Deep Convolution Networks for Compression Artifacts Reduction [Електронний ресурс]: [Веб-сайт]. — Електронні дані. — Ke Yu, Chao Dong, Chen Change Loy, Xiaoou Tang — Режим доступу: <http://mmlab.ie.cuhk.edu.hk/projects/ARCNN.html>