

УДК 004.04/.67

ІНТЕГРАЦІЯ ЦИФРОВОЇ ОБРОБКИ ЗОБРАЖЕНЬ В МЕСЕНДЖЕР-БОТ

Мартіросян Р. К., студент, ruben.martirosyan@kname.edu.ua,
ХНУМГ імені О. М. Бекетова
Братерська Н. М., асистент, nataliia.braterska@kname.edu.ua,
ХНУМГ ім. О. М. Бекетова

У сучасному світі цифрове оброблення зображень стає все більш важливим і поширеним. Цей процес охоплює різні методи та алгоритми, які дають змогу змінювати та покращувати якість зображень, а також витягувати з них корисну інформацію. Одним зі способів використання цифрової обробки зображень є її інтеграція в месенджер-ботів.

Суть обробки зображень полягає в застосуванні різних операцій і фільтрів до пікселів зображення з метою зміни його зовнішнього вигляду або видалення інформації з нього. У контексті месенджер-ботів, цифрове опрацювання зображень може бути використане для аналізу вмісту зображень, розпізнавання об'єктів на них, зміни розмірів або формату зображень і багато чого іншого.

Принцип обробки тексту із зображень у Python зазвичай включає в себе використання бібліотеки OpenCV для завантаження зображень, а потім застосування різних методів обробки, таких як фільтри, сегментація і розпізнавання об'єктів. Для розпізнавання тексту на зображеннях можна використовувати бібліотеку pytesseract, яка надає простий інтерфейс для роботи з OCR (оптичним розпізнаванням символів). Принцип роботи з цією бібліотекою полягає в наступному: програма отримує на вході зображення, після цього попередньо обробляє його, після чого безпосередньо і відбувається робота з pytesseract, який використовує залежну для неї бібліотеку Leptonica і Trained Data Set (набір файлів, які містять дані для навчання моделі Tesseract на конкретній мові або складних виразах). Після цього відбувається пост-обробка результату, за підсумком якої ми отримуємо текст (рис. 1).

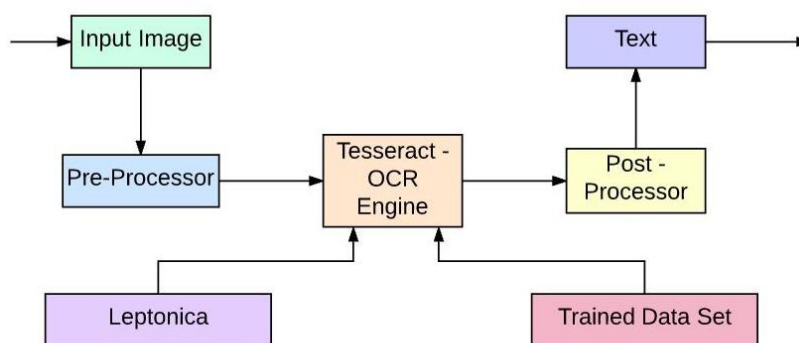


Рисунок 1 – Принцип розпізнавання символів в бібліотеці pytesseract

Принцип інтеграції цієї обробки в телеграм-бота полягає в написанні коду, який буде взаємодіяти з месенджером Telegram через його API. Для цього можна використовувати бібліотеку `python-telegram-bot`, яка надає зручні інструменти для створення та управління ботами в Telegram. У середині бота можна реалізувати функціонал, на основі якого бот отримує зображення від користувача, потім застосовує до нього обрані методи обробки зображень, після чого аналізує отриманий текст і відправляє результат назад користувачеві. Цей процес може бути автоматизовано, що дає змогу користувачам швидко отримувати результати обробки зображень прямо в месенджері.

Для реалізації цієї функціональності необхідно використовувати Python для написання коду бота, а також, як було вище згадано, бібліотеки `OpenCV` і `pytesseract` для обробки зображень і розпізнавання тексту. Крім того, знадобиться бібліотека `python-telegram-bot` для роботи з Telegram API.

Таблиця 1 – Бібліотеки, необхідні для реалізації телеграм-боту з розпізнаванням тексту, та їх характеристика

Назва бібліотеки	Характеристика (який функціонал вона забезпечує)
OpenCV	Бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях.
Pytesseract	Вільна програма для розпізнавання текстів.
Python-telegram-bot	Ця бібліотека надає асинхронний інтерфейс на чистому Python для API Telegram Bot. На додаток до чистої реалізації API, ця бібліотека містить ряд високорівневих класів, щоб зробити розробку ботів легкою і простою.

У результаті успішної інтеграції цифрової обробки зображень у месенджер-бота, користувачі отримують значне розширення можливостей взаємодії з ботом. Відправляючи зображення, вони можуть отримувати оброблені результати в різних форматах, що значно збагачує їхній досвід спілкування з ботом. У нашому випадку, користувачі можуть надсилати фотографії з текстом, і отримувати у відповідь розпізнаний текст, що спрощує процес обміну інформацією та покращує доступність контенту. Наприклад, якщо користувач захоче швидко розпізнати текст зі світлини книжки, рецепта чи документа, месенджер-бот, обробляючи зображення, надасть текстову версію для подальшого використання.

Список використаних джерел

1. opencv-python. PyPI. [Електронний ресурс] / Режим доступу: <https://pypi.org/project/opencv-python/>
2. pytesseract. PyPI. [Електронний ресурс] / Режим доступу: <https://pypi.org/project/pytesseract/>
3. python-telegram-bot. PyPI. [Електронний ресурс] / Режим доступу: <https://pypi.org/project/python-telegram-bot/>
4. Solem J. E. Programming computer vision with python: tools and algorithms for analyzing images. O'Reilly Media, Incorporated, 2012. 264 p.
5. Zelic F. How to OCR with Tesseract in Python with Pytesseract and OpenCV?. Nanonets Intelligent Automation, and Business Process AI Blog. [Електронний ресурс] / Режим доступу: <https://nanonets.com/blog/ocr-with-tesseract/>