КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Звіт

з самостійної роботи

«Розпізнавання образів та обробка зображень за допомогою OpenCV»

з теми

«Розпізнавання друкованого тексту »

Виконав
студент 4 курсу
групи ТК-41
факультету комп'ютерних наук
та кібернетики
Некряч Владислав Вадимович

Київ
2023

OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим кодом) — 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.

Бібліотека розроблена Intel і нині підтримується Willow Garage та Itseez. Сирцевий код бібліотеки написаний мовою C++ і поширюється під ліцензією BSD. Біндинги підготовлені для різних мов програмування, таких як Python, Java, Ruby, Matlab, Lua та інших. Може вільно використовуватися в академічних та комерційних цілях.

Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору. Алгоритми OpenCV застосовують у таких сферах:

- Аналіз та обробка зображень
- Системи з розпізнавання обличчя
- Ідентифікації об'єктів
- Розпізнавання жестів на відео
- Відстежування переміщення камери
- Побудова 3D моделей об'єктів
- Створення 3D хмар точок зі стерео камер
- Склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю
- Система взаємодії людини з комп'ютером
- Пошуку схожих зображень із бази даних

- Усування ефекту червоних очей при фотозйомці зі спалахом
- Стеження за рухом очей
- Аналіз руху
- Ідентифікація об'єктів
- Сегментація зображення
- Трекінґ відео
- Розпізнавання елементів сцени і додавання маркерів для створення доповненої реальності

та інші.

Функціонал OpenCV:

- **Робота із структурами даних**

Для зберігання та роботи із зображеннями OpenCV використовує вектори та скаляри, матриці та діапазони. Вони дозволяють проводити математичні перетворення, орієнтуватися на зображення і виконувати безліч інших дій.

- **Видозміна зображень**

За допомогою OpenCV з картинкою можна працювати як у графічному редакторі: обрізати, збільшувати чи зменшувати, обертати. В основному програмісти використовують цю можливість для попередньої підготовки картинки перед її розшифровкою, наприклад, обрізають непотрібні частини.

- **Додавання ефектів**

Картинку можна зробити у відтінках сірого або повністю чорно-білого. Це важливо для алгоритмів розпізнавання, які працюють із знебарвленими зображеннями. Можна змінювати колірний тон, розмивати, згладжувати чи геометрично змінювати картинку.

- **Малювання поверх зображення**

На картинку можна нанести лінії та геометричні фігури, зробити підпис, наприклад, щоб виділити знайдену програмою особу. Часто це використовується в мобільних програмах для камери: квадрат навколо обличчя людини під час зйомки означає, що програма розпізнала його.- Розпізнавання об'єктів

Для розпізнавання елементів OpenCV використовуються обриси об'єктів, сегментація за кольорами, вбудовані методи розпізнавання, які можна настроювати залежно від об'єкта і чутливості алгоритму.

- **Робота з відеороликами**

Нові версії бібліотеки підтримують роботу не лише з картинками, а й із відео. Вони можуть зчитувати ролики з використанням кодеків, аналізувати те, що відбувається в них, відстежувати рухи та елементи. Це корисно, наприклад, при програмуванні робота, що рухається, або створенні ПЗ для камери відеоспостереження.

Реалізація

Для імплементації проекту мною була використана мова програмування високого рівня Python.

Завантажуємо бібліотеку cv2 та pytesseract

```
import cv2
import pytesseract
```

Pytesseract - це бібліотека Python, яка надає інтерфейс до Tesseract-OCR - безкоштовного та відкритого движка оптичного розпізнавання символів (OCR), який може розпізнавати текст зі зображень. Pytesseract робить процес розпізнавання тексту на зображенні простішим, дозволяючи розробникам легко інтегрувати OCR-функціональність в їхні програми на Python.

Основні функції Pytesseract:

1. Розпізнавання тексту зі зображень: Pytesseract може розпізнавати текст зі зображень, включаючи скановані документи, фотографії та інші типи зображень.

2. Мовна підтримка: Pytesseract підтримує більше 100 мов, включаючи українську, англійську та інші.

3. Висока точність: Tesseract-OCR, на якому ґрунтується Pytesseract, має високу точність розпізнавання тексту зі зображень, що дозволяє отримувати якісні результати.

4. Легкість використання: Pytesseract має простий інтерфейс, який дозволяє легко інтегрувати OCR-функціональність в програму на Python.

5. Налаштування: Pytesseract надає можливість налаштування параметрів OCR-розпізнавання, що дозволяє досягти більш якісних результатів.

Зчитуємо зображення з текстом, який ми будемо зчитувати, за допомогою

cv2.imread(), переводимо його у градації сірого за допомогою функції

cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

```
img = cv2.imread("sample123123.jpg")
grayimg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Застосуємо порогове значення до перетвореного зображення за допомогою

функції cv2.threshold. Як метод вибору порогового значення ми вибрали метод Оцу та обернену бінарну порогову функцію.

Метод порогової обробки Оцу - це техніка сегментації зображення, яка використовується для автоматичного визначення оптимального значення порогу для розділення переднього та заднього плану пікселів у сірому зображенні. Він базується на ідеї мінімізації внутрішньокласової дисперсії, яка вимірює розподіл інтенсивностей пікселів в кожному класі (передньому та задньому планах). Алгоритм працює за допомогою обчислення гістограми вхідного зображення, а

потім ітеративного обчислення значення порогу, яке мінімізує внутрішньокласову дисперсію. Це значення порогу використовується для бінаризації зображення, що призводить до отримання бінарного зображення, де передні пікселі встановлені на білий колір, а задні пікселі встановлені на чорний.

Метод порогової обробки Оцу часто використовується в застосунках комп'ютерного зору для завдань обробки зображень, таких як виявлення об'єктів, сегментація та вилучення ознак. Він особливо корисний для зображень з двохмодальним розподілом інтенсивності, де є чіткі значення переднього та заднього плану пікселів.

Щоб визначити форму розпізнаваного тексту як прямокутник, використаємо функцію cv2.getStructuringElement(cv2.MORPH_RECT, (20, 20)), розмір прямокутника (розпізнаваного текстового ядра) – 20 на 20. Після вибору правильного текстового ядра розширимо текстові блоки за допомогою функції cv2.dilate, що збільшує ймовірність коректного визначення блоків на зображенні.

```
ret, thresh1 = cv2.threshold(grayimg, 0, 255, cv2.THRESH_OTSU |
cv2.THRESH_BINARY_INV)
rect_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (20, 20))
dilation = cv2.dilate(thresh1, rect_kernel, iterations=1)
```

Знайдемо контури зображень за допомогою функції cv2.findContours().

```
contours, hierarchy = cv2.findContours(dilation, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
im2 = img.copy()
```

Тепер по кожному з контурів зображень знайдемо їх координати x, y та ширину і висоту за допомогою функції cv2.boundingRect(). Малюємо на зображенні за допомогою ф-ції cv2.rectangle() прямокутники за допомогою отриманих координат. Обрізаємо область і передаємо її у функцію pytesseract.image_to_string(cropped). Pytesseract визначає текст на зображенні та записує результати у файл recognized2.txt.

```
for cnt in contours:
    x, y, w, h = cv2.boundingRect(cnt)
```

```
rect = cv2.rectangle(im2, (x, y), (x + w, y + w), (0, 255, 0), 2)
cropped = im2[y:y + h, x:x + w]
file = open("recognized2.txt", "a")
text = pytesseract.image_to_string(cropped)
file.write(text)
file.close()
```

РЕЗУЛЬТАТИ

input

What is Bash Script?

Bash (Bourne Again Shell) is the command language/command interpreter introduced as the **shell (Bourne Shell)** replacement. **Brian Fox** developed it in **1989** for the **GNU Project**. A Bash script contains a set of commands to automate the task execution.

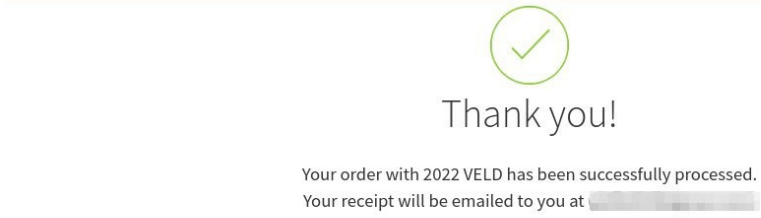
It also includes **imperative programming**, such as conditional constructs, loops, functions, etc. Hence, **Bash is a powerful tool** in Linux used for data crunching, automated backups, system administration, custom script creation, web development, and many more. A Bash script is a text file having the .sh as the extension.

Output

Italso includes imperative programming, such as conditional constructs, loops, functions, etc. Hence, Bash is a powerful tool
in Linux used for data crunching, automated backups, system administration, custom script creation, web development, and
many more. A Bash script is a text file having the .sh as the extension.
What is Bash Script?

Bash (Bourne Again Shell) is the command language/command interpreter introduced as the shell (Bourne Shell)
replacement. Brian Fox developed it in 1989 for the GNU Project. A Bash script contains a set of commands to automate the
task execution.

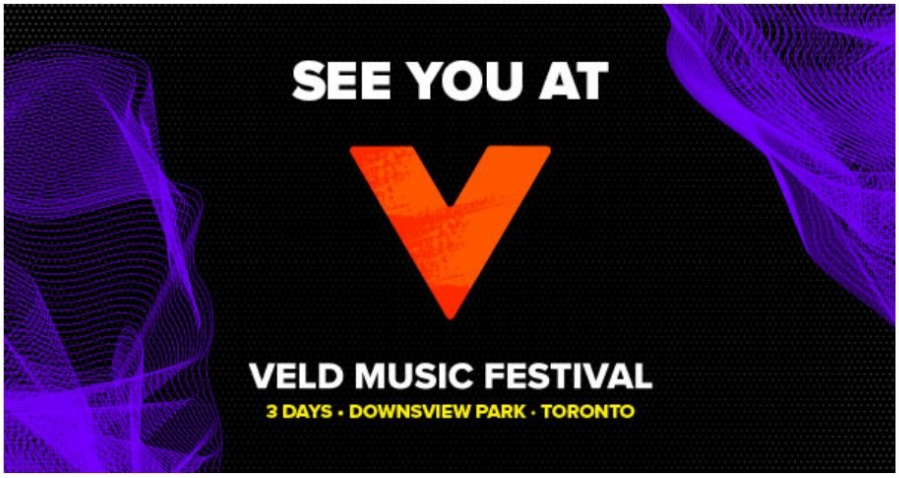
Input



Thank you!

Your order with 2022 VELD has been successfully processed.
Your receipt will be emailed to you at

Order #128664671



Output:

```
Order #128664671

SEE YOU AT

VELD MUSIC FESTIVAL

3 DAYS - DOWNSVIEW PARK - TORONTO

Your order with 2022 VELD has been successfully processed.
Your receipt will be emailed to you at

Thank you!
```