Звіт за практичну роботу №5. Сегментація зображення.

Автор: Ясногородський Нікіта Вікторович, ТУ-12-22-Б1ІПЗ

Опис завдання

Метою цієї практичної роботи було розробити програмне забезпечення, яке дозволяє накладати різні фільтри на зображення та виконувати сегментацію обраного зображення.

Алгоритм сегментації

Сегментація зображення полягає у його розділенні на окремі частини або області, які відображають об’єкти чи області інтересу. Цей процес сприяє відокремленню об’єктів на зображенні від тла чи інших об’єктів.

Для виконання сегментації використовуються різноманітні алгоритми, такі як порогова обробка, водоподіл, алгоритми кластеризації та інші. У даному прикладі ми застосуємо порогову обробку для розділення зображення на області на основі певного порогу яскравості для забезпечення простоти виконання.

Опис програмного забезпечення

Програма має наступний функціонал:

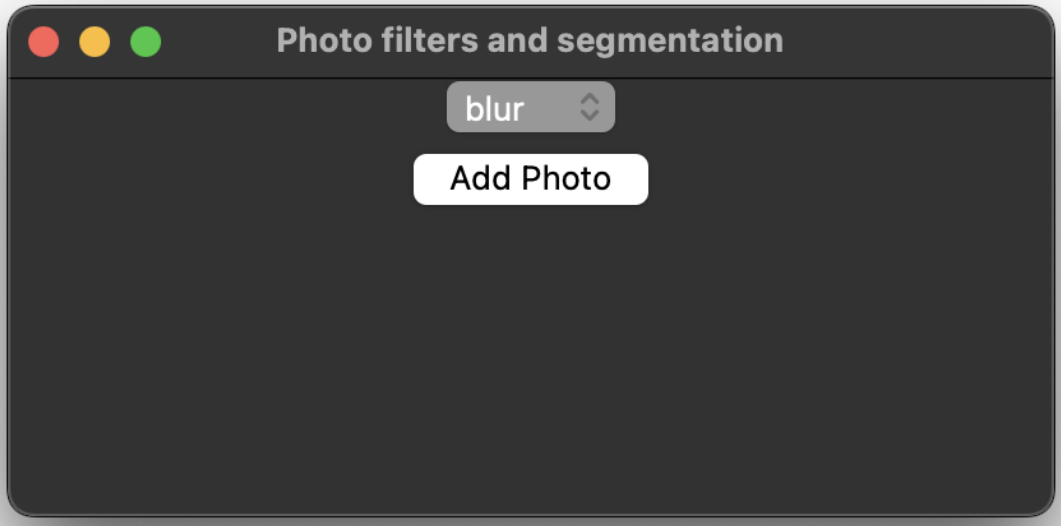
- Завантаження зображення з пристрою користувача.
- Вибір фільтрів з графічного інтерфейсу.
- Накладання обраних фільтрів.
- Сегментація зображення на основі порогової обробки.
- Відображення оригінального зображення, зображення з накладеними фільтрами та сегментованого зображення.

```
from enum import StrEnum, auto
from tkinter import Button, OptionMenu, StringVar, Tk, filedialog

import cv2
import numpy as np


class FilterOptions(StrEnum):
    BlackAndWhite = auto()
    Negative = auto()
    Blur = auto()
```

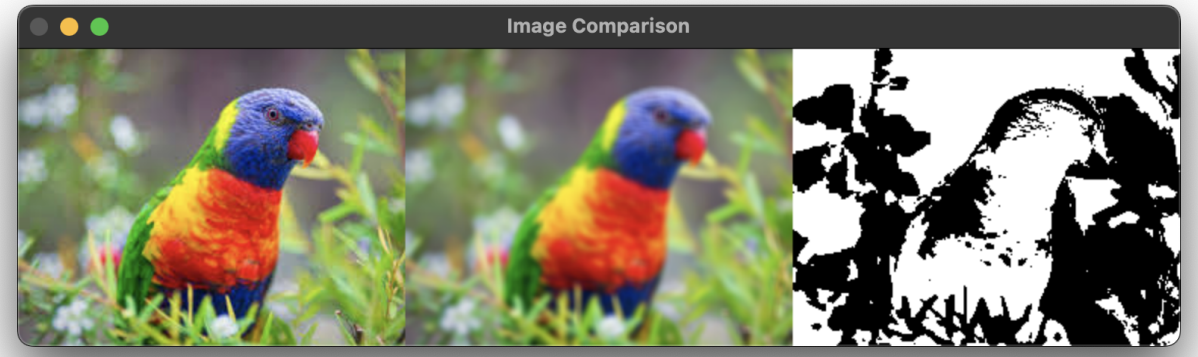
```
def apply_filter(image, filter_type):
    match filter_type:
        case FilterOptions.BlackAndWhite:
            return cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        case FilterOptions.Negative:
            return cv2.bitwise_not(image)
        case FilterOptions.Blur:
            return cv2.filter2D(
                src=image, ddepth=-1, kernel=np.ones((5, 5), np.float32) / 25
            )
        case _:
            return image


def segment_image(image):
    if len(image.shape) == 3:  # change to gray if colourful
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    _, binary = cv2.threshold(image, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
    return binary


def make_img_3_channel(image):
    if len(image.shape) != 3:
        return cv2.cvtColor(image, cv2.COLOR_GRAY2BGR)
    return image


def open_image(filter_type):
    file_path = filedialog.askopenfilename()
    image = cv2.imread(file_path)
    filtered_image = apply_filter(image, filter_type)
    segmented_image = segment_image(image)
    cv2.imshow(
        "Image Comparison",
        np.hstack(
            list(map(make_img_3_channel, [image, filtered_image, segmented_image]))
        ),
    )
    cv2.waitKey(0)
    cv2.destroyAllWindows()


def render():
    root = Tk()
    root.title("Photo filters and segmentation")
    filter_var = StringVar(root)
    filter_var.set(FilterOptions.Blur)
    filter_options = OptionMenu(
        root,
        filter_var,
```

```
        FilterOptions.BlackAndWhite,
        FilterOptions.Negative,
        FilterOptions.Blur,
    )
    filter_options.pack()

    open_button = Button(
        root, text="Add Photo", command=lambda: open_image(filter_var.get())
    )
    open_button.pack()
    root.mainloop()


render()
```

Головний інтерфейс програми:



Фільтр "Блюр" з оригінальним та сегментованим зображення для порівняння:



Висновок

Під час виконання даної практичної роботи було успішно розроблено програмне забезпечення, яке дозволяє застосовувати різноманітні фільтри до зображень та виконувати їх сегментацію на основі порогової обробки. Для цього були використані інструменти мови програмування Python та бібліотеки Tkinter та OpenCV.