

Звіт за практичну роботу №6. Двовимірне обертання зображення навколо осі.

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

Завдання

Метою практичної роботи було розширення функціональності програми, реалізованої у попередній практичній роботі, шляхом додавання можливості обертання зображення навколо осей **x** та **y**.

Хід роботи

1. Функціонал обертання зображення навколо осей **x** та **y** успішно реалізовано.
2. Було проведено тестування програми з використанням різних зображень та кутів обертання.
3. Отримано результати візуалізації обернених зображень з метою проведення аналізу.

```
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 rotate_image(image, degree):
    rows, cols, _ = image.shape
    M = cv2.getRotationMatrix2D((cols / 2, rows / 2), degree, 1)
    return cv2.warpAffine(image, M, (cols, rows))

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

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

```

```

)
filter_options.pack()

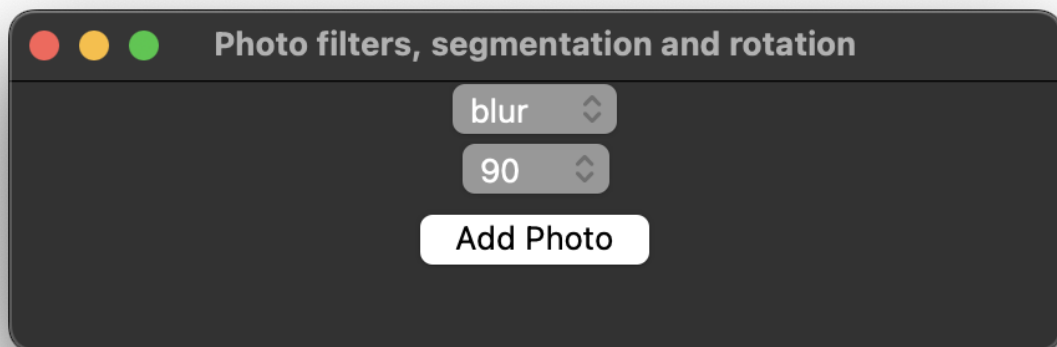
rotation_var = StringVar(root)
rotation_var.set("90")
rotation_options = OptionMenu(root, rotation_var, "90", "180", "270")
rotation_options.pack()

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

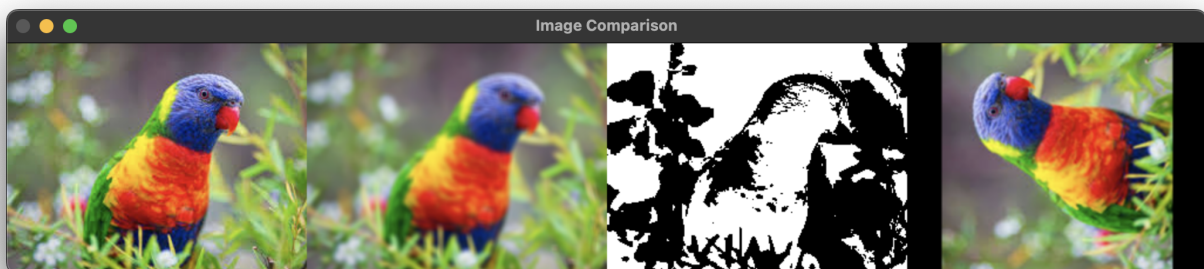
render()

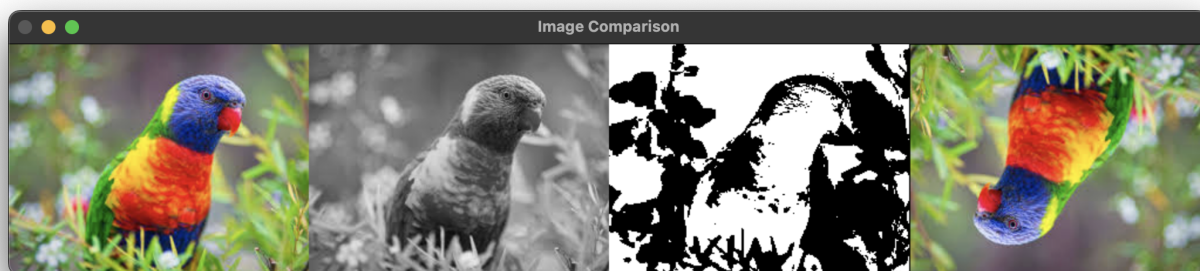
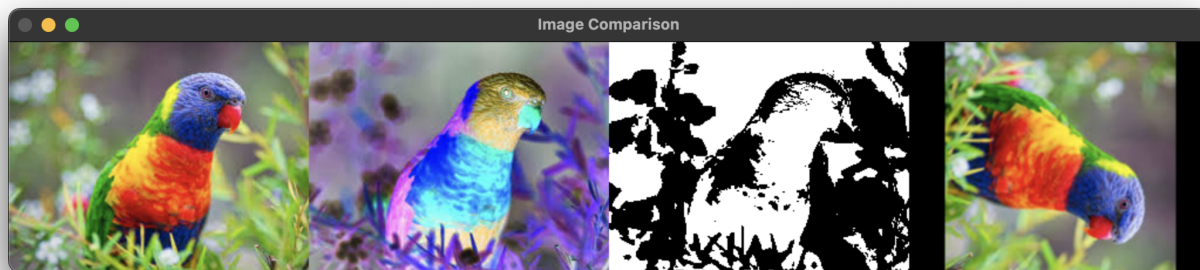
```

Інтерфейс програми:



Оберти 90, 180 та 270 градусів:





Висновок

Виконання практичної роботи дозволило успішно розширити функціональність програмного забезпечення для обробки зображень. Додавання можливості обертання зображення навколо осей x та y зробило програму більш універсальною та корисною для використання. У ході роботи були отримані нові навички у роботі з обробкою зображень та застосуванні алгоритмів обертання.