

Звіт за практичну роботу №6.

Двовимірне обертання зображення навколо осі.

Завдання

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

Виконання

1. Дослідження алгоритмів обертання: Перед початком роботи було вивчено різні алгоритми для обертання зображення навколо осей x та y . Були розглянуті математичні аспекти обертання та його вплив на зображення.
2. Розширення програмного забезпечення: На основі отриманих знань було розширено програмне забезпечення, додавши функціонал обертання зображення. Для цього використовувалася бібліотека OpenCV для обробки зображень.
3. Тестування та візуалізація: Після реалізації функціоналу обертання зображення було проведено тестування програми з різними зображеннями та кутами обертання. Результати тестування були візуалізовані для аналізу.

Результати

1. Успішно реалізовано функціонал обертання зображення навколо осей x та y .
2. Проведено тестування програми з різними зображеннями та кутами обертання.
3. Отримано результати візуалізації обернутих зображень для аналізу.

Програма була розроблена з використанням мови програмування Python та бібліотеки Tkinter для створення графічного інтерфейсу. Для маніпуляції зображень було використано бібліотеку OpenCV.

```
from tkinter import *
from tkinter import filedialog

import cv2
import numpy as np

def apply_filter(image, filter_type):
```

```

if filter_type == "Чорно-білий":
    return cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
elif filter_type == "Сепія":
    kernel = np.array(
        [[0.393, 0.769, 0.189], [0.349, 0.686, 0.168], [0.272, 0.534, 0.131]]
    )
    return cv2.filter2D(image, -1, kernel)
elif filter_type == "Негатив":
    return cv2.bitwise_not(image)
else:
    return image

def segment_image(image):
    if len(image.shape) == 3: # Перевіряємо, чи є зображення кольоровим
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    else:
        gray = image
    _, binary = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
    return binary

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():
    file_path = filedialog.askopenfilename()
    image = cv2.imread(file_path)
    cv2.imshow("Image Comparison", np.hstack([image, image, image]))

    filter_type = filter_var.get()
    filtered_image = apply_filter(image, filter_type)
    if len(filtered_image.shape) != 3:
        filtered_image = cv2.cvtColor(
            filtered_image, cv2.COLOR_GRAY2BGR
        ) # Перетворюємо у формат з трьома каналами

    segmented_image = segment_image(image)
    segmented_image = cv2.cvtColor(
        segmented_image, cv2.COLOR_GRAY2BGR
    ) # Перетворюємо у формат з трьома каналами

    rotation_degree = int(rotation_var.get())
    rotated_image = rotate_image(image, rotation_degree)

    cv2.imshow(
        "Image Comparison",
        np.hstack([image, filtered_image, segmented_image, rotated_image]),

```

```

)

cv2.waitKey(0)
cv2.destroyAllWindows()

# Створення графічного інтерфейсу Tkinter
root = Tk()
root.title("Накладання фільтрів, сегментація та обертання зображення")

# Вибір фільтра
filter_var = StringVar(root)
filter_var.set("Чорно-білий")
filter_options = OptionMenu(root, filter_var, "Чорно-білий", "Сепія", "Негатив")
filter_options.pack()

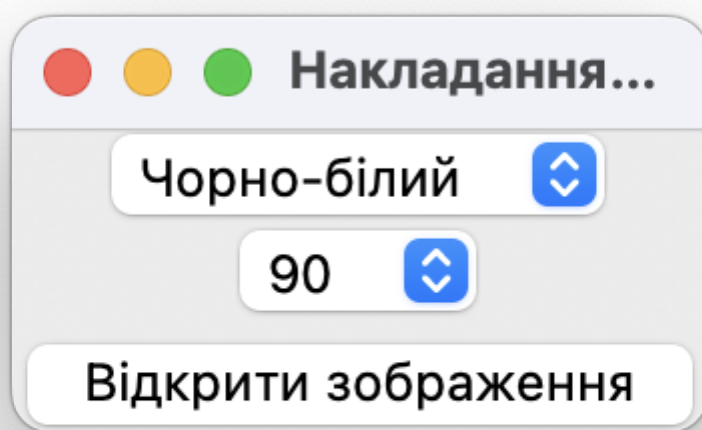
# Вибір ступеня обертання
rotation_var = StringVar(root)
rotation_var.set("0")
rotation_options = OptionMenu(root, rotation_var, "0", "90", "180", "270")
rotation_options.pack()

# Кнопка відкриття зображення
open_button = Button(root, text="Відкрити зображення", command=open_image)
open_button.pack()

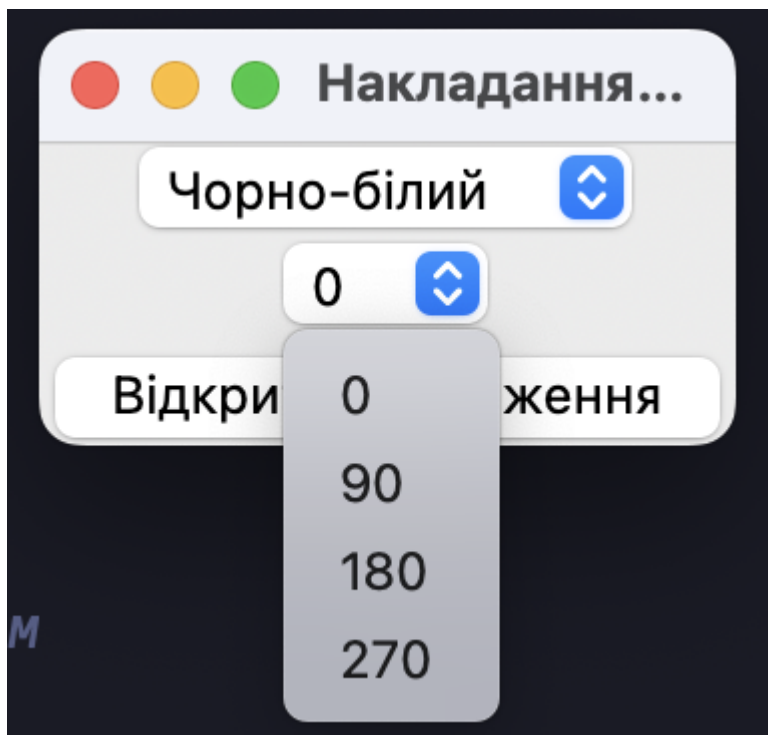
root.mainloop()

```

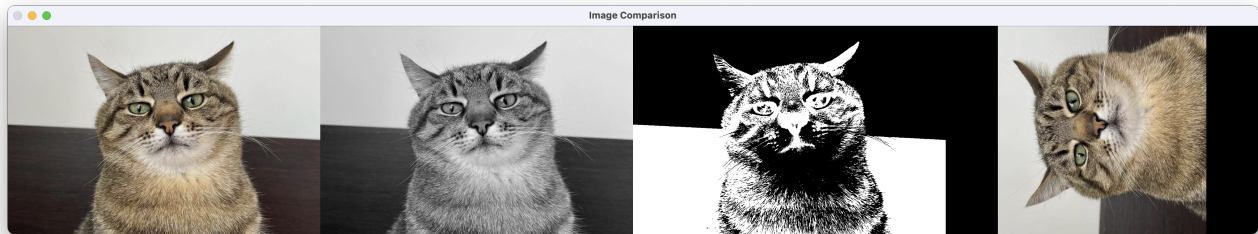
Головний інтерфейс програми:



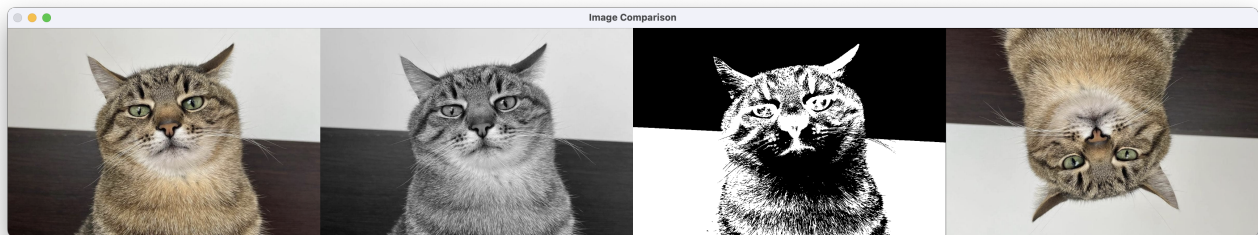
Можливість обрати кут оберту:



Оберт 90 градусів:



Оберт 180 градусів:



Висновок

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