

МЕТА РОБОТИ

Застосувати теоретичні знання з основ роботи з бібліотекою tkinter на мові Python, навички використання бібліотеки matplotlib, а також об'єктно-орієнтований підхід до проектування програм, і навчитися розробляти скрипти для інженерних додатків з графічним інтерфейсом.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вивчити документацію до бібліотеки Pillow і написати скрипт з визначенням класу, що реалізує користувацький інтерфейс для виконання наступних функцій:

- 1) відкриття файлу із зображенням будь-якого допустимого графічного формату;
- 2) відображення зображення та інформації про формат;
- 3) * Установка значень для виконання функцій 4-5;
- 4) створення зменшеної копії вихідного зображення;
- 5) геометричні перетворення мініатюри, фільтрація, перетворення формату і вставка в вихідне зображення відповідно до варіанту (див. табл.1);
- 6) збереження зміненого зображення в фай і реалізацією роботи з об'єктом цього класу для запуску віконного програми.

13.	Вертикальний дзеркальний образ	DETAIL	8 бітів, чорно-біле	У лівого краю посередині
-----	-----------------------------------	--------	---------------------	-----------------------------

Завдання 2. Вивчити документацію до бібліотеки OpenCV і написати скрипт з визначенням і роботою об'єктів класу, що реалізує користувацький інтерфейс для виконання наступних функцій:

- 1) відкриття файлу із зображенням будь-якого допустимого графічного формату;
- 2) * Установка значень для виконання функцій 3-4;
- 3) зміна розмірів зображення;
- 4) геометричні перетворення зображення, зміна кольорового простору,

фільтрація і виконання операцій із зображенням відповідно до варіанту (див. табл.2);

5) відображення вихідного зображення і після кожної зміни;

6) збереження змінених зображень у файли

і реалізацією роботи з об'єктом цього класу для запуску віконного програми.

	(вертикальне)			
14.	Проективне перетворення (горизонтальне)	з ефектом зсуву	YUV	Виділення меж zerocross

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі Task1_13:

Вхідні дані (ім'я, опис, тип, обмеження):

file_path – шлях до файлу, string

Вихідні дані (ім'я, опис, тип):

Вивід результату у інтерфейсі

Лістинг коду вирішення задачі наведено в дод. А.2 (стор. 6).

Екран роботи програми показаний на рис. Б.1

Завдання 2. Вирішення задачі Task2_14:

Вхідні дані (ім'я, опис, тип, обмеження):

file_path – шлях до файлу, string

Вихідні дані (ім'я, опис, тип):

Вивід результату у інтерфейсі

Лістинг коду вирішення задачі наведено в дод. А.3 (стор. 7).

Екран роботи програми показаний на рис. Б.2

ВИСНОВКИ

1.Pillow:

Бібліотека для роботи із зображеннями. Дозволяє відкривати, змінювати розмір, обрізати, накладати фільтри та зберігати зображення у різних форматах.

2.OpenCV:

Потужна бібліотека для обробки зображень і комп'ютерного зору. Підтримує різноманітні операції, включаючи обробку кольорів, розпізнавання об'єктів та застосування фільтрів.

3.Розробка віконних додатків:

Використовуючи Pillow або OpenCV і бібліотеку Tkinter, можна створювати додатки для:

- Завантаження зображень із файлу.
- Обробки зображень (фільтри, зміна розмірів, обертання тощо).
- Збереження змінених зображень.
- Відображення результатів у графічному інтерфейсі.

ДОДАТОК А.1

Лістинг головного меню завдання

```
from lab6_1 import Window1
from lab6_2 import Window2
import tkinter

task_list = {
    "1": (Window1, "lab6_1-322-v13-Zubov-Yevgen", "1024x768"),
    "2": (Window2, "lab6_2-322-v13-Zubov-Yevgen", "1024x768")
}

choice = input("Please, choose the task 1-2 (0-EXIT): ")
while choice != "0":
    # якщо даний ключ є у словнику
    if choice in task_list.keys():
        # Створення відповідного вікна
        application = tkinter.Tk()
        window_class, window_name, window_size = task_list.get(choice)
        window = window_class(application)
        application.geometry(window_size)
        application.title(window_name)
        application.mainloop()
    else:
        print("Wrong task number!")
    choice = input("Please, choose the task again (0-EXIT): ")
```

ДОДАТОК А.2

Лістинг завдання №1

```
import tkinter
from tkinter.filedialog import askopenfile
from tkinter import messagebox

from PIL import ImageFilter, Image, ImageOps

class Window1(tkinter.Frame):
    def __init__(self, parent):

        super().__init__(parent)
        self.parent = parent
        self.pack(fill=tkinter.BOTH, expand=1)
        self.config(bg="#A43820")
        self.create_widgets()
        self.grid_rowconfigure(0, weight=1)
        self.grid_rowconfigure(1, weight=1)
        self.grid_rowconfigure(2, weight=1)
        self.grid_columnconfigure(0, weight=1)
        self.grid_columnconfigure(1, weight=1)
        self.grid_columnconfigure(2, weight=1)

        self.file_path = tkinter.StringVar(value="None")

    def create_widgets(self):

        self.lb1 = tkinter.Label(self, text = "Оберіть зображення", bg="#A43820",
fg = "white")
        self.btn1 = tkinter.Button(self, text = "Обрати зображення", command =
self.image_choose)
```

```

        self.btn2 = tkinter.Button(self, text="Відкрити зображення у форматі 8біт",
command=self.image_8bit)
        self.btn3 = tkinter.Button(self, text="Відкрити зображення з фільтром
DETAIL", command=self.detailfilter_image)
        self.btn4 = tkinter.Button(self, text="Відкрити відзеркалене
зображення",command=self.rotated_image)

        self.lb1.grid(row=0, column=0, sticky=tkinter.NSEW)
        self.btn1.grid(row=1, column=0, sticky=tkinter.NSEW)
        self.btn2.grid(row=1, column=1, sticky=tkinter.NSEW)
        self.btn3.grid(row=2, column=0, sticky=tkinter.NSEW)
        self.btn4.grid(row=2, column=1, sticky=tkinter.NSEW)

def image_choose(self):
    f = askopenfile(mode='rb', defaultextension=".jpg",
                    filetypes=(("Image files", "*.jpg"), ("All files", "*.
*"))))

    if f is not None:
        full_path = f.name
        self.file_path = f.name

def image_8bit(self):
    if self.file_path is not None:
        try:
            with Image.open(self.file_path) as img:
                img.load()
                img_8bit = img.convert('L')
                img_8bit.show()
        except Exception as e:
            messagebox.showerror("Помилка", f"Файл неможливо відкрити")
    else:
        messagebox.showerror("Помилка", "Файл не обрано")

def detailfilter_image(self):
    if self.file_path is not None:
        try:
            with Image.open(self.file_path) as img:
                img1 = img.filter(ImageFilter.DETAIL)
                width, height = img.size
                img.paste((255, 0, 0), (width - 117, 9, width - 9, 86))
                img.paste(img1, (width - 113, 10))
                img1.show()

        except Exception as e:
            messagebox.showerror("Помилка", f"Файл неможливо відкрити")
    else:
        messagebox.showerror("Помилка", "Файл не обрано")

def rotated_image(self):
    if self.file_path is not None:
        try:
            with Image.open(self.file_path) as img:
                mirrored_img = ImageOps.mirror(img)
                mirrored_img.show()

        except Exception as e:
            messagebox.showerror("Помилка", f"Файл неможливо відкрити")
    else:
        messagebox.showerror("Помилка", "Файл не обрано")

```

ДОДАТОК А.3

Лістинг завдання №2

```
import tkinter
from tkinter.filedialog import askopenfile
from tkinter import messagebox
import numpy as np

import cv2

class Window2(tkinter.Frame):
    def __init__(self, parent):

        super().__init__(parent)
        self.parent = parent
        self.pack(fill=tkinter.BOTH, expand=1)
        self.config(bg="#A43820")
        self.create_widgets()
        self.grid_rowconfigure(0, weight=1)
        self.grid_rowconfigure(1, weight=1)
        self.grid_rowconfigure(2, weight=1)
        self.grid_columnconfigure(0, weight=1)
        self.grid_columnconfigure(1, weight=1)
        self.grid_columnconfigure(2, weight=1)

        self.file_path = tkinter.StringVar(value="None")

    def create_widgets(self):

        self.lb1 = tkinter.Label(self, text = "Оберіть зображення", bg="#A43820",
fg = "white")
        self.btn1 = tkinter.Button(self, text="Обрати зображення",
command=self.choose_image)
        self.btn2 = tkinter.Button(self, text = "Відкрити зображення з режимом
YUV", command =self.yuv_image)
        self.btn3 = tkinter.Button(self, text = "Відкрити зображення з проєктивним
перетворенням ", command = self.proective_transform)
        self.btn4 = tkinter.Button(self, text="Відкрити зображення з виділенням меж
zerocross ",command=self.moved_image)

        self.lb1.grid(row=0, column=0, sticky=tkinter.NSEW)
        self.btn1.grid(row=1, column=0, sticky=tkinter.NSEW)
        self.btn2.grid(row=1, column=1, sticky=tkinter.NSEW)
        self.btn3.grid(row=1, column=2, sticky=tkinter.NSEW)
        self.btn4.grid(row=2, column=0, sticky=tkinter.NSEW)

    def choose_image(self):
        f = askopenfile(mode='rb', defaultextension=".jpg",
                        filetypes=(("Image files", "*.jpg"), ("All files", "*.*")))

        if f is not None:
            self.file_path = f.name

    def yuv_image(self):
        if self.file_path != "None":
            try:
                img = cv2.imread(self.file_path)
                yuv_img = cv2.cvtColor(img, cv2.COLOR_BGR2YUV)
                cv2.imshow("yuv_image", yuv_img)
                cv2.waitKey(0)
                cv2.destroyWindow("yuv_image")
            except Exception as e:
```



```

        messagebox.showerror("Помилка", f"Файл неможливо відкрити")
    else:
        messagebox.showerror("Помилка", "Файл не обрано")

def proective_transform(self):
    if self.file_path != "None":
        try:
            img = cv2.imread(self.file_path)
            height, width = img.shape[:2]
            src_points = np.float32([[0, 0], [width - 1, 0], [0, height - 1],
[width - 1, height - 1]])
            dst_points = np.float32([[50, 50], [width - 50, 100], [0, height -
100], [width, height]])
            matrix = cv2.getPerspectiveTransform(src_points, dst_points)
            transformed_img = cv2.warpPerspective(img, matrix, (width, height))

            cv2.imshow("Проективне перетворення", transformed_img)
            cv2.waitKey(0)
            cv2.destroyAllWindows("Проективне перетворення")
        except Exception as e:
            messagebox.showerror("Помилка", f"Файл неможливо відкрити")
    else:
        messagebox.showerror("Помилка", "Файл не обрано")

def moved_image(self):
    if self.file_path != "None":
        try:
            image = cv2.imread(self.file_path, cv2.IMREAD_GRAYSCALE)
            blurred = cv2.GaussianBlur(image, (5, 5), 0)
            laplacian = cv2.Laplacian(blurred, cv2.CV_64F)
            laplacian_normalized = cv2.normalize(laplacian, None, 0, 255,
cv2.NORM_MINMAX)
            zero_cross = np.zeros_like(laplacian_normalized, dtype=np.uint8)
            for i in range(1, laplacian.shape[0] - 1):
                for j in range(1, laplacian.shape[1] - 1):
                    patch = laplacian[i - 1:i + 2, j - 1:j + 2]
                    if np.min(patch) < 0 and np.max(patch) > 0:
                        zero_cross[i, j] = 255
            cv2.imshow("Zero Crossing", zero_cross)
            cv2.waitKey(0)
            cv2.destroyAllWindows("Zero Crossing")
        except Exception as e:
            messagebox.showerror("Помилка", f"Файл неможливо відкрити")
    else:
        messagebox.showerror("Помилка", "Файл не обрано")

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

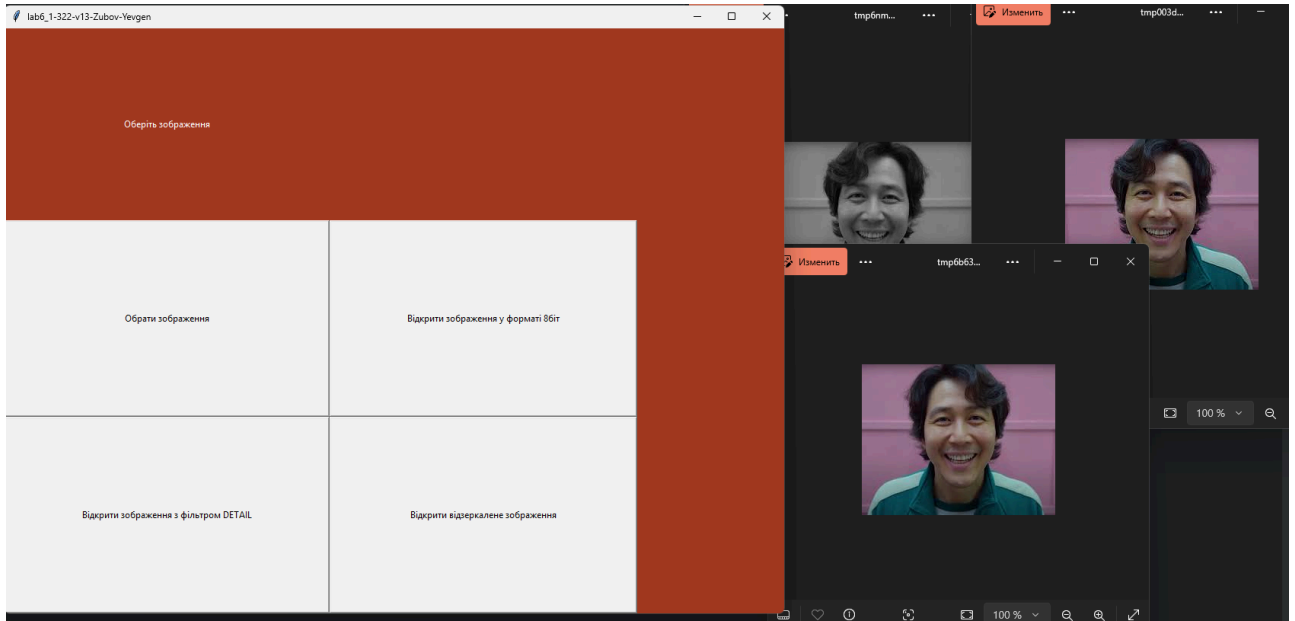
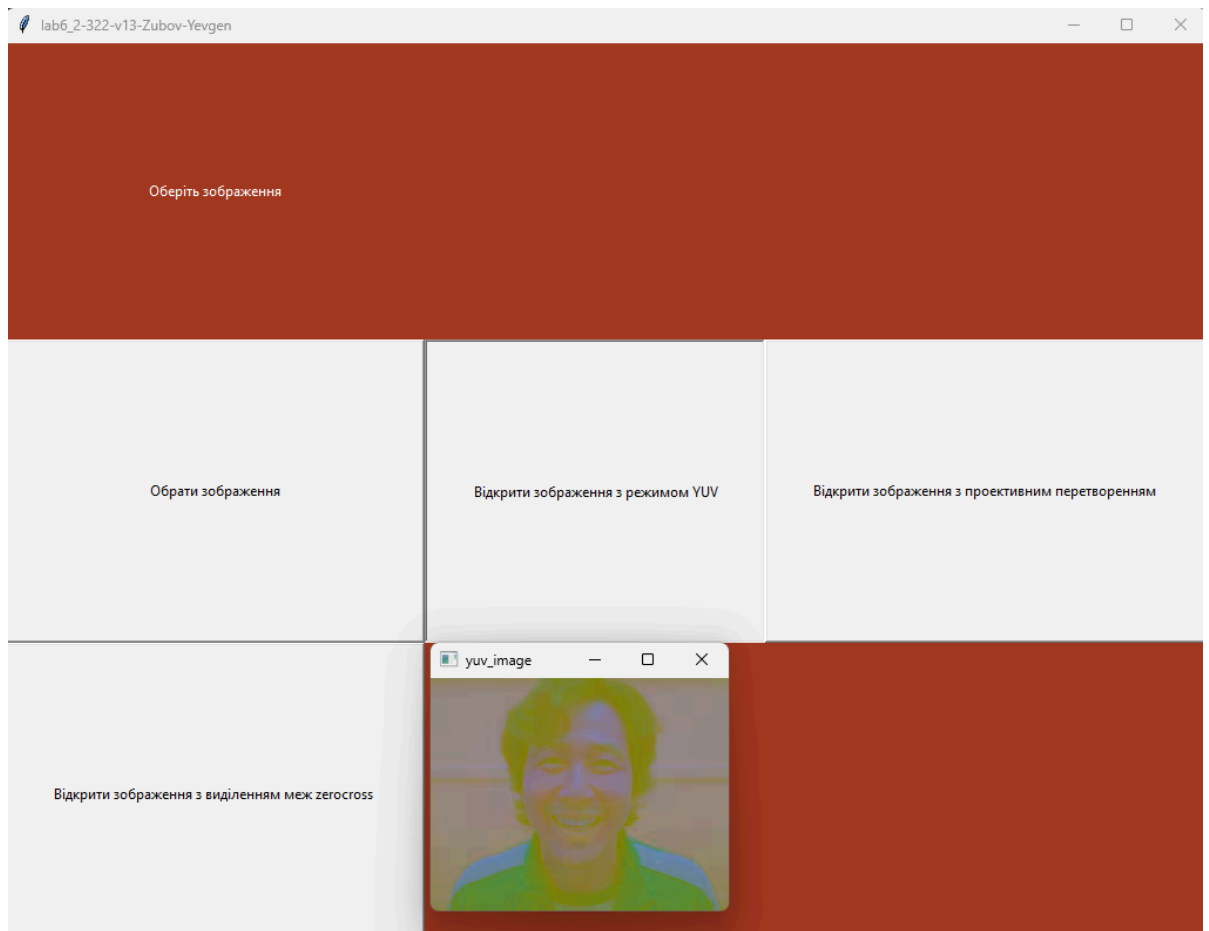


Рисунок Б.1 – Екран виконання програми для вирішення завдання
Task1_13



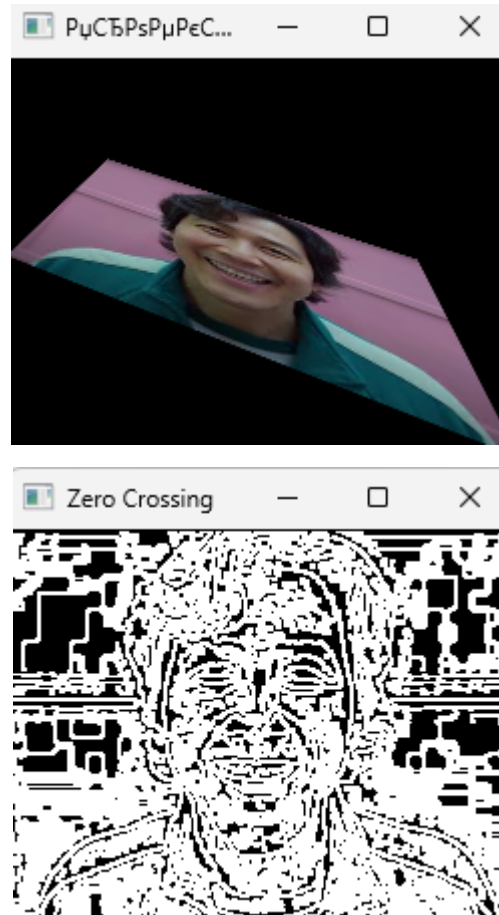


Рисунок Б.2 – Екран виконання програми для вирішення завдання
Task2_14