МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет електроніки і комп’ютерних технологій

Кафедра системного проєктування

**Звіт**

про виконання лабораторної роботи № 5

з дисципліни «Цифрова обробка зображень»

«Дискретна двовимірна згортка сигналів»

**Виконав:**

студент групи Феп-31

Линва Віталій

**Перевірив:**

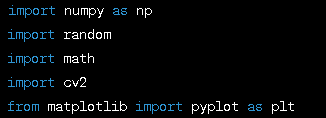
Проф. Половинко І. І.

**Львів – 2023**

**Хід роботи**

У рамках цієї лабораторної роботи було реалізовано накладання водяного знаку на зображення за допомогою бібліотеки OpenCV та NumPy. Накладання водяного знаку дозволяє захистити авторські права на зображення або вказати його походження.

Для початку були імпортовані необхідні бібліотеки:



Після цього було зчитано зображення водяного знаку та основне зображення:



Далі були отримані розміри зчитаних зображень:



На основному зображенні була визначена область, до якої буде накладений водяний знак:



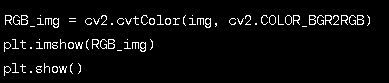
Потім водяний знак був накладений на основне зображення:



Накладене зображення було збережено з водяним знаком:



Нарешті, результат був відображений на екрані:



Вивід:

  
  
**Висновок:** Цей код демонструє основні кроки для накладання водяного знаку на зображення за допомогою OpenCV. Він може бути використаний для захисту авторських прав на зображення або вказівки на походження.

import numpy as np

import random

import math

import cv2

from matplotlib import pyplot as plt

#img read

logo = cv2.imread("Labs\Lab5\logo.png")

img = cv2.imread("Labs\Lab5\img.jpg")

#Отримую розміри зчитаних зображень

(wH, wW) = logo.shape[:2]

(h, w) = img.shape[:2]

#Визначення області "Destination" на основному зображенні, до якого буде накладений водяний знак

destination = img[h - wH - 10:h - 10, w - wW - 10:w - 10]

#Накладання водяного знаку

result = cv2.addWeighted(destination, 1, logo, 0.7, 0)

img[h - wH - 10:h - 10, w - wW - 10:w - 10] = result

#Збереження зображення з водяним знаком

cv2.imwrite("Labs\Lab5\watermarked.jpg", img)

#Відображення зображення

RGB\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

plt.imshow(RGB\_img)

plt.show()