

**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота № 1**

з дисципліни  
«Штучний інтелект в задачах обробки зображень»

Виконав:

студент групи ІМ-11  
Царик Микола Миколайович  
варіант відповідно до списку: 22

Перевірів:

Нікітін Валерій Андрійович

**Київ 2023**

**Мета:** ознайомитись з операціями читання, відображення та збереження зображень з використанням пакету для мови програмування Python.

**Завдання:** Проробити всі приклади, які наведені у теоретичних відомостях.

1. Встановити OpenCV
2. Зчитати зображення
3. Відобразити зображення
4. Зберегти файл
5. Отримати доступ до певного пікселю
6. Зробити вирізання зображення
7. Змінити розмір зображення
8. Повернути зображення
9. Розмити зображення
10. Склеїти два зображення
11. Намалювати прямокутник
12. Намалювати лінію
13. Намалювати коло
14. Розмістити текст

**Хід виконання:**

### 1) Команда для встановлення OpenCV

```
pip3 install opencv-python
```

### 2) Код для зчитування зображення

```
img = cv2.imread("Lab1\data\image.jpg")  
imgGray = cv2.imread("Lab1\data\image.jpg",0)
```

Тут ми використали нуль для того щоб зробити зображення чорно-білим

### 3) Код для відображення зображень.

```
images = [img, imgGray, imgCropped, imgResized, imgRotated,  
  
#Послідовне відображення зображень  
for image in images:  
    cv2.imshow("Image", image)  
    cv2.waitKey()  
  
cv2.destroyAllWindows()
```

Відобразити зображення я вирішив через цикл щоб код був компактнішим та читабельнішим

#### 4) Код для зберігання зображення

```
# Зберігання файлу  
cv2.imwrite("Lab1\data\newImage.jpg",img)
```

В цьому випадку перший аргумент - шлях куди зберігати зображення, а другий - це саме зображення

#### 5) Код для отримання доступу до певного пікселю

```
# Звернення до пікселю  
(b,g,r) = img[100,50]  
print("red =" , r , ", green =", g, ", blue =",b)
```

Оскільки `img` це масив пікселів , які представляють зображення, ми можемо звернутися до будь якого. В цьому прикладі взявши піксель ми відображаємо три характеристики його кольору (b,g,r)

#### 6) Код для вирізання зображень

```
imgCropped = img[145:620, 60:400]
```

Тут використовуємо певний проміжок по висоті та ширині щоб вказати, яку саме області ми хочемо вирізати

#### 7) Код для змінення розміру зображення

Для початку імпортуємо модуль `imutils`

```
import utils
```

Його ми використаємо для більш простого обчислення майбутніх розмірів зображення

```
imgResized = utils.resize(img, width=300)
```

Тут нам достатньо лише вказати зображення та бажану характеристику висоти чи ширини, і друга характеристика автоматично підлаштується під першу в результаті чого отримаємо зображення з правильними пропорціями.

Також є другий метод де ми можемо порахувати пропорцію вручну

```
# Зміна розміру в cv2
# height,width,  = img.shape[0:2]
# ratio = width/height;
# newHeight = 300;
# newWidth = int(newHeight * ratio)
# imgResized = cv2.resize(img, (newWidth,newHeight))
```

## 8) Код для повороту зображення

```
imgRotated = imutils.rotate(img, 45)
```

В цьому випадку ми знову можемо використати `imutils` та його метод `rotate`, в якому першим аргументом буде зображення, а другим – кут нахилу.

Як і в минулому випадку, поворот можна зробити і без використання `imutils`. Для цього треба задати координати центра повороту та зробити матрицю нахилу

```
# Поворот в cv2
# height, width = img.shape[:2]
# center = (width // 2, height//2)
# rotationMatrix = cv2.getRotationMatrix2D(center, 45, 1) # 45 - градуси повороту, 1 - зберегти розміри
# imgRotated = cv2.warpAffine(img, rotationMatrix, (width, height))
```

## 9) Код для розмиття зображення

```
imgBlurred = cv2.GaussianBlur(img, (11,11), 0)
```

```
# 11,11 - ступінь розмиття, 0 - стандартне відхилення (автоматично)
```

`cv2` надає вбудовану функцію розмиття зображення, нам треба лише вказати саме зображення, ступінь розмиття, та стандартне відхилення (0 – значення за замовчуванням)

## 10) Код для склеювання зображень

Нам доведеться імпортувати модуль `numpy` тому що він надає необхідну нам функцію поєднання зображень.

```
import numpy
```

Після імпорту нам достатньо лише викликати відповідний метод `hstack()` та вказати зображення що будуть горизонтально склеєні.

```
imgUnited = numpy.hstack((img, imgBlurred))
```

## 11) Код для малювання прямокутника

```
cv2.rectangle(imgForRectangle, (60,145), (400,620), (0,0,255), 3)
```

Для малювання прямокутника використовується метод `rectangle()`, в який ми передаємо зображення на якому малюватимемо, координати початку і кінцю, колір та товщину лінії.

## 12) Код для малювання лінії

```
cv2.line(imgForLine, (265,125), (255,355), (255,0,0), 4)
```

Лінія малюється методом `line()` в який ми також передаємо зображення на якому малюватимемо, координати початку і кінцю, колір та товщину лінії.

Крім того можна намалювати одразу багато об'єднаних ліній методом `polylines`. Для цього створимо масив точок, який потім передамо аргументом та поміж іншим вкажемо значення «1» щоб остання точка поєдналася з першою і фігура була «замкнена»

```
points = numpy.array([[185,160],[255,145],[335,175],[350,250],[325,325]])  
cv2.polylines(imgForPolylines, numpy.int32([points]), 1, (0,255,0), 4)
```

## 13) Код для малювання кола

```
cv2.circle(imgForCircle, (255,255), 90, (212,173,49), 3)
```

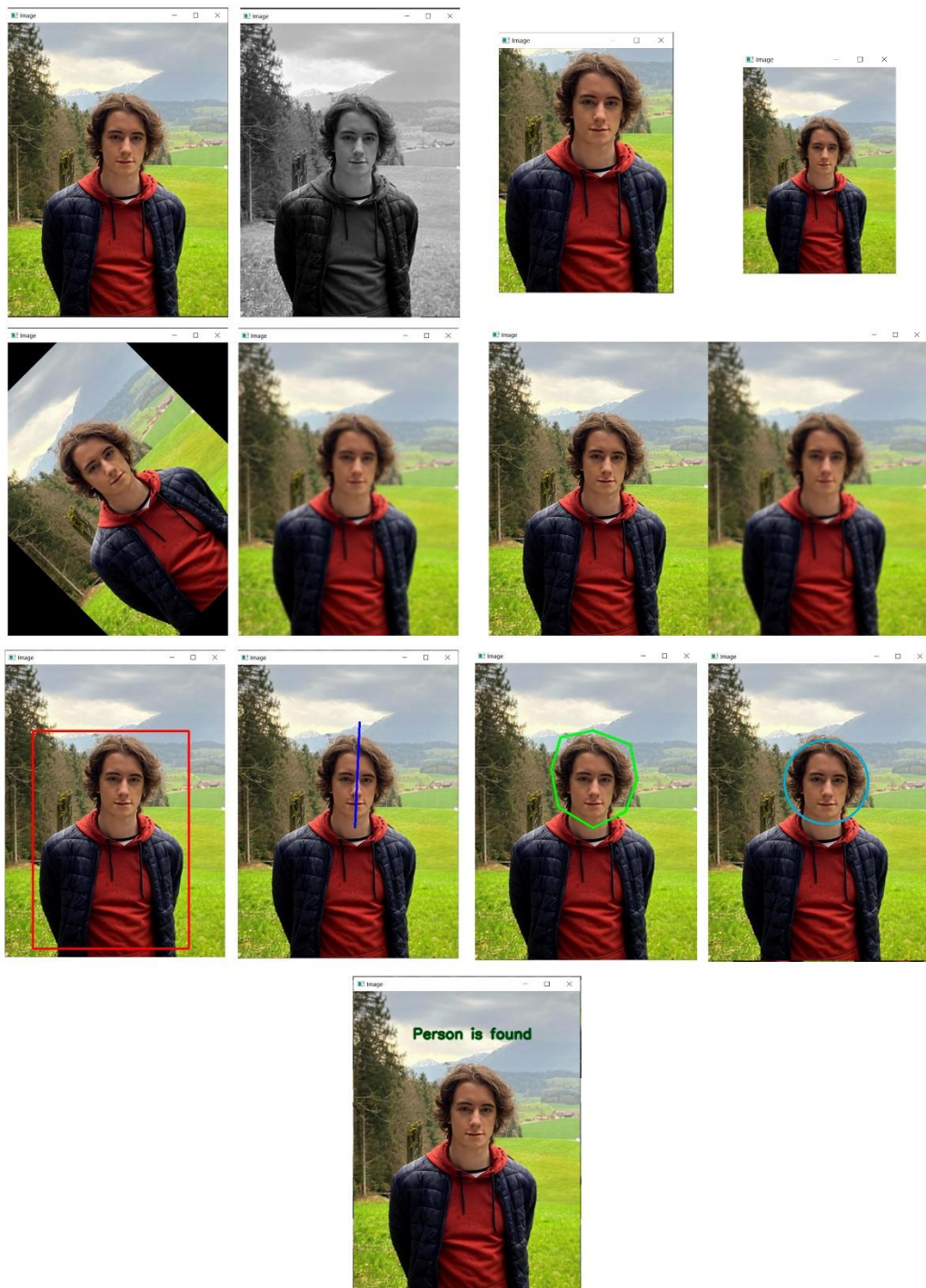
Коло малюється функцією `circle()` в яку передаємо зображення, координати центру кола, радіус, колір та товщину ліній.

## 14) Код для додавання тексту

```
font = cv2.FONT_ITALIC  
cv2.putText(imgForText, "Person is found", (125,100) , font, 1 ,(16,74,9),4, cv2.LINE_AA)
```

Для тексту спочатку створимо змінну з шрифтом, а потім використаємо функцію `putText()` в яку передамо зображення, текст, початкові координати, шрифт, масштабування тексту, колір, товщину лінії та тип лінії

## Результат



## Висновок

В ході виконання лабораторної, окрім навичок роботи з бібліотекою cv2 для редагування зображень я зробив кілька висновків

- 1) OpenCV надає дуже широкий спектр функцій для роботи з зображеннями.
- 2) Стандартні інструменти бібліотеки cv2 можуть виконувати більшість необхідних маніпуляцій з зображенням, але зручніше деякі дії виконувати допоміжними модулями такими як `imutils`, це економить час та робить код простішим.
- 3) Через те що OpenCV був створений багато років тому, в ньому залишився тогочасний стандарт BGR кольору замість RGB, треба мати це на увазі щоб кольорова гамма в програмі співпадала з очікуваною.
- 4) У функціях модулю cv2 при вказанні розмірів спочатку йде висота, а тільки потім ширина (а не навпаки). Не дивлячись на це, при вказанні координат формат заповнення все ще (x,y), не (y,x). Така особливість може збити з толку при обрізанні зображення або вказанні точок для малювання.

## Контрольні питання

### 1. Що таке OpenCV?

OpenCV (Open Source Computer Vision Library) - це бібліотека з відкритим вихідним кодом, яка містить набір функцій для роботи з комп'ютерним зором і обробки зображень. Вона широко використовується для розв'язання завдань машинного зору, таких як розпізнавання облич, виявлення об'єктів, відстеження руху, розпізнавання тексту тощо.

### 2. Що таке піксель?

Піксель - це найменша одиниця зображення на екрані дисплея чи у цифровому зображенні. Він представляє собою один кольоровий або відтінковий пункт, який складається з числових значень, що визначають його кольорову інформацію.

### 3. Як встановити пакет opencv-python?

Для встановлення пакету opencv-python можна використовувати менеджер пакетів `pip`. Просто виконайте наступну команду в командному рядку:

```
pip3 install opencv-python
```

#### 4. Як розмити зображення?

Для розмиття зображення в OpenCV використовується функція `cv2.GaussianBlur()`, яка застосовує розмиття за вказаними параметрами.

Простий приклад коду для розмиття зображення:

```
imgBlurred = cv2.GaussianBlur(img, (11,11), 0)
```

#### 5. Які функції використовуються для малювання та відображення тексту?

Для малювання та відображення тексту в OpenCV використовуються функції, такі як `cv2.putText()` для відображення тексту на зображенні та `cv2.line()`, `cv2.rectangle()`, `cv2.circle()` для малювання ліній, прямокутників та кола відповідно.

```
cv2.rectangle(imgForRectangle, (60,145), (400,620), (0,0,255), 3)
cv2.line(imgForLine, (265,125), (255,355), (255,0,0), 4)
cv2.circle(imgForCircle, (255,255), 90, (212,173,49), 3)
cv2.putText(imgForText, "Person is found", (125,100), cv2.FONT_ITALIC, 1, (16,74,9), 4, cv2.LINE_AA)
```