

Міністерство освіти і науки України

Національний технічний університет України „КПІ імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Звіт до лабораторної роботи №1

З дисципліни «Штучний інтелект в задачах обробки зображень»

|  |  |
| --- | --- |
| Перевірив:  доц. Нікітін В.А. | Виконала:  Студентка 3 курсу, гр. ІС-13  Росновська О.О. |

Київ 2024 р.

**Лабораторна робота №1.**

**Тема:** Основи роботи з OpenCV

**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.

# Хід роботи

Для початку відкриємо зображення. Аби не дублювати постійно код, я написала для цього функцію, яка використовується надалі.

def show\_i(name, img):  
 cv2.imshow(name, img)  
 cv2.waitKey()  
 cv2.destroyAllWindows()  
  
  
img = cv2.imread('image.jpg')

show\_i("shinji", img)

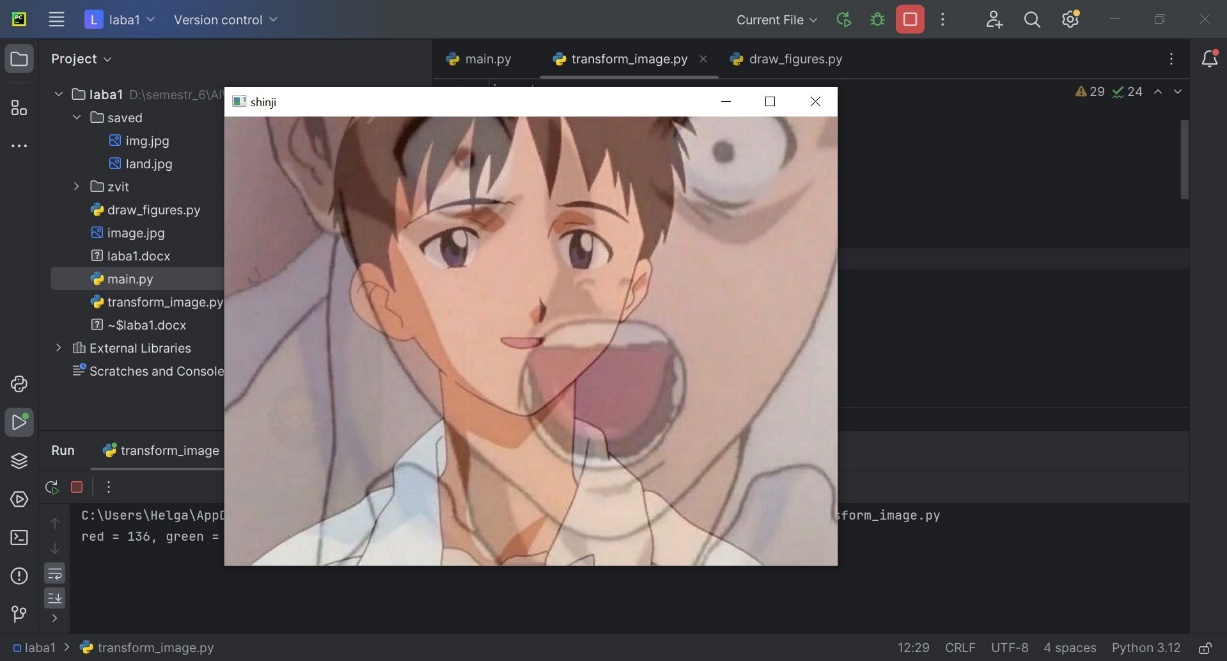


Рис.1 Відкрите зображення

Тепер аби виділити не все зображення, а деякий його фрагмент

piece = img[60:260, 320:689]

show\_i("piece shinji", piece)

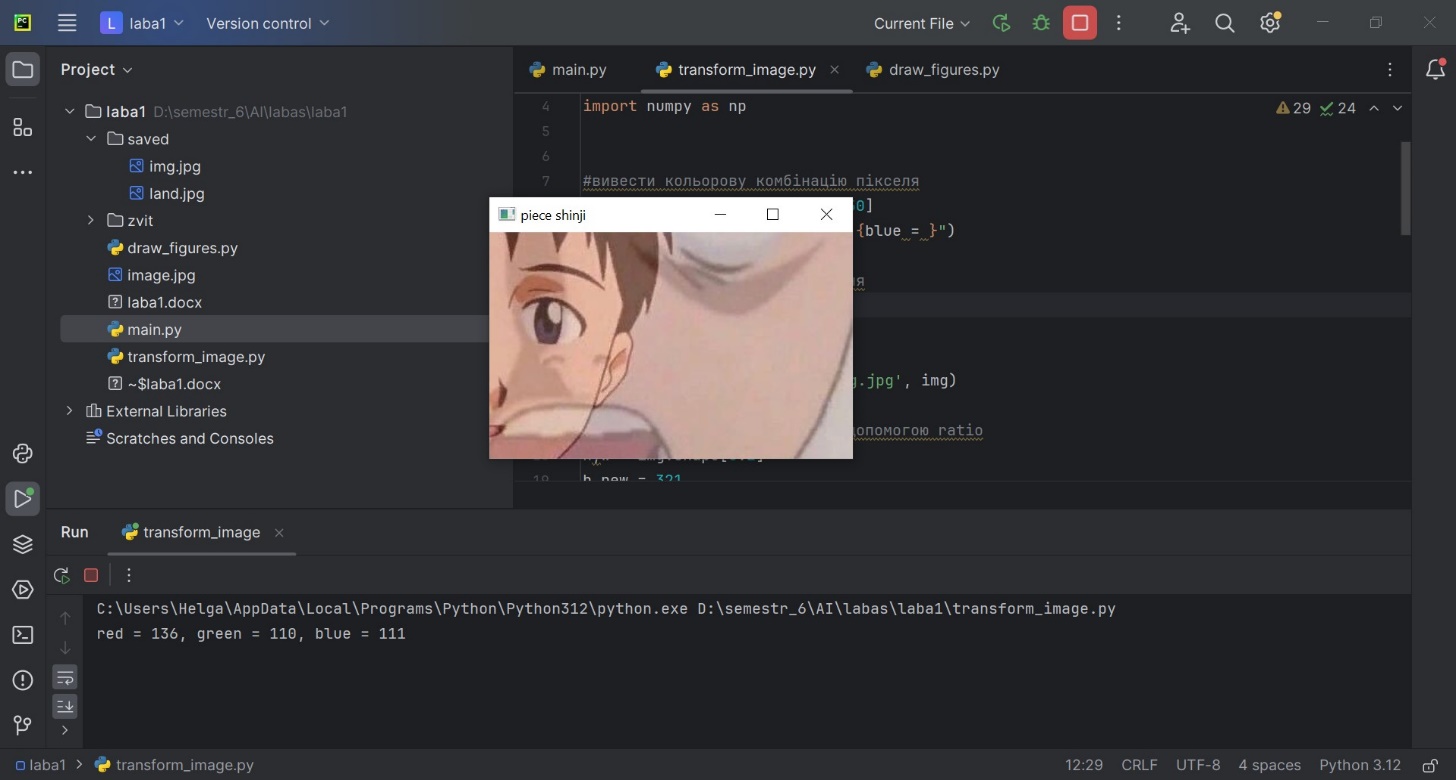


Рис.2 Фрагмент зображення

Для зміни розміру зображення я використала обидва методи. Метод розрахунку ratio та завдяки бібліотеки imutils.

#зміна розміру зображення за допомогою ratio  
h,w = img.shape[0:2]  
h\_new = 321  
ratio = w/h  
w\_new=int(h\_new\*ratio)  
resized\_1 = cv2.resize(img,(w\_new,h\_new))  
  
#зміна розміру зображення за допомогою imutils  
resized\_2 = imutils.resize(img, width = 321)

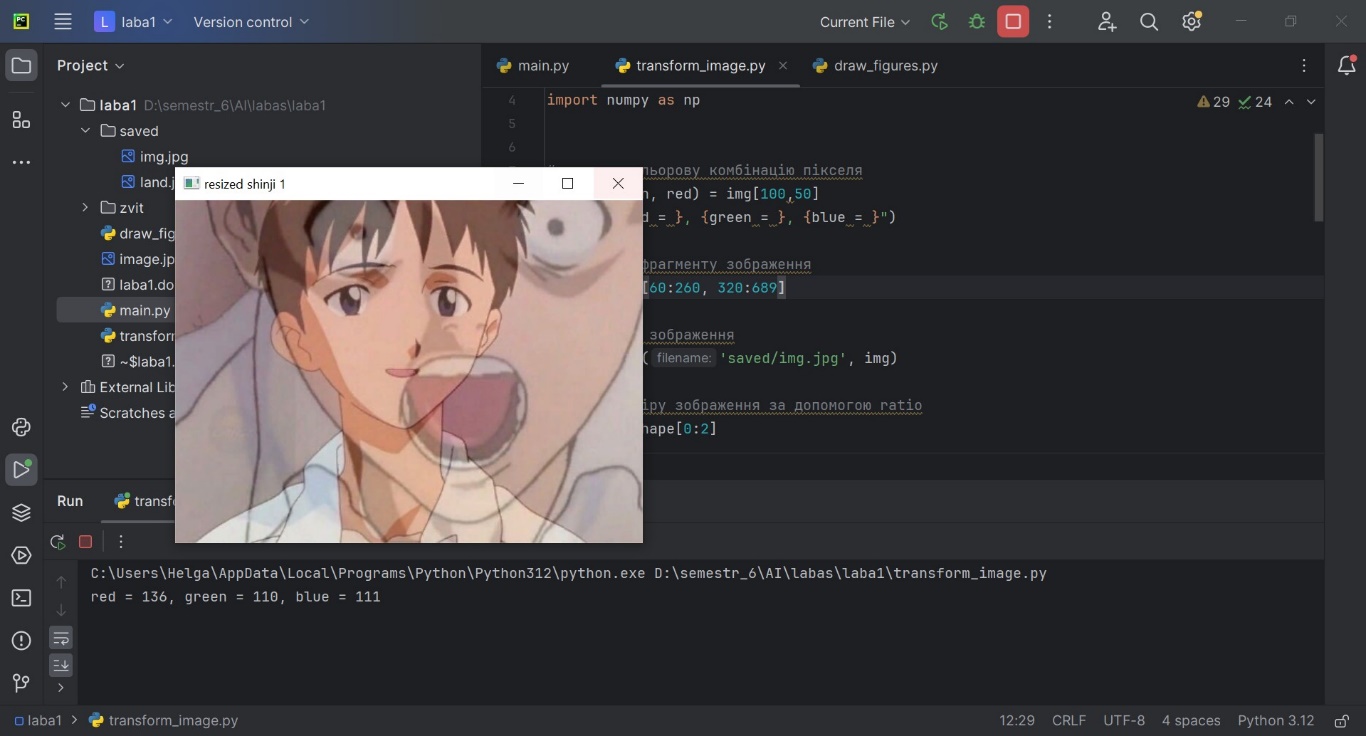


Рис.3 Зміна розміру ratio

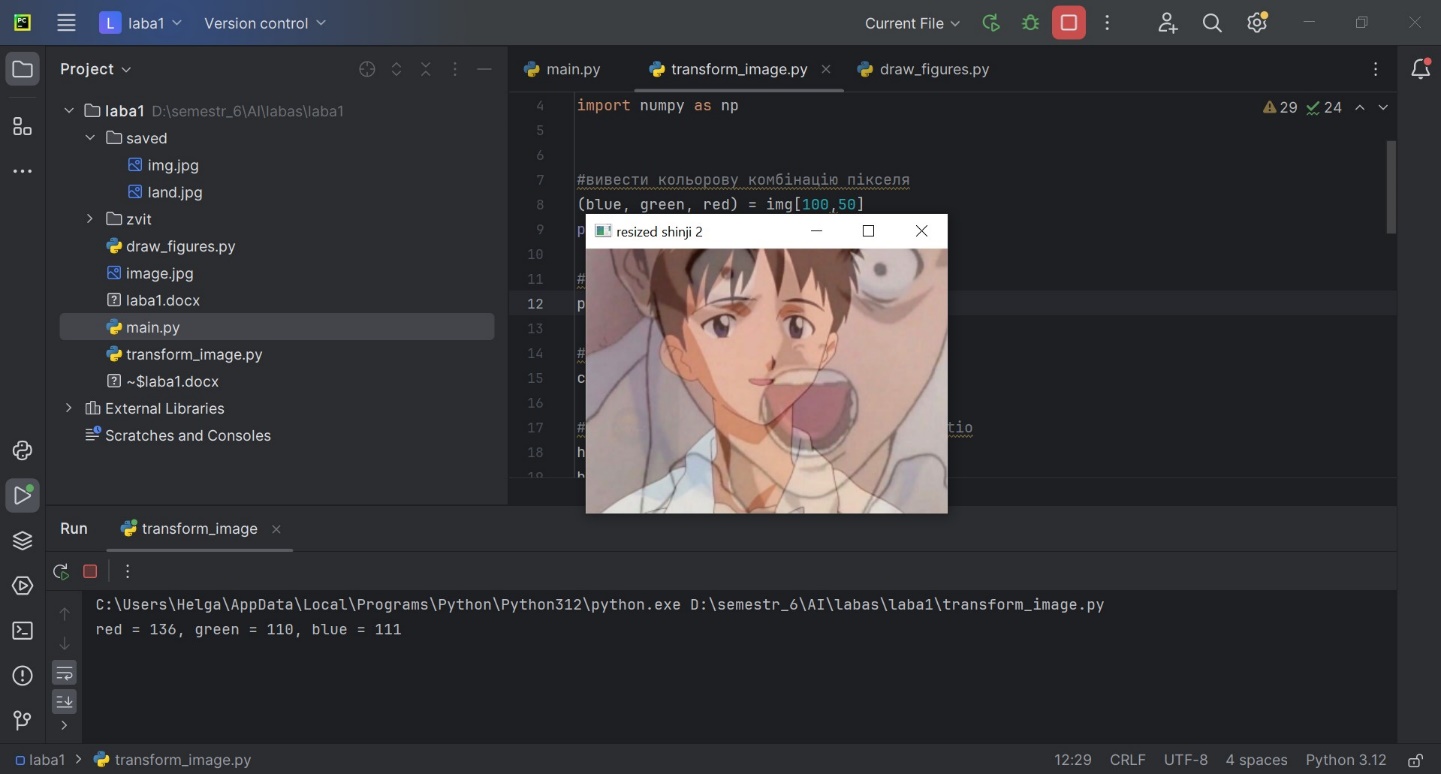


Рис.4 Зміна розміру imutils

Для повертання зображення я також використала обидва методи. Метод розрахунку центру зображення та завдяки бібліотеки imutils.

#поворот зображення  
h,w = resized\_1.shape[0:2]  
center = (w//2, h//2)  
M = cv2.getRotationMatrix2D(center, -45, 1)  
rotated\_1 = cv2.warpAffine(resized\_1, M, (w,h))  
  
#поворот зображення imutils  
rotated\_2 = imutils.rotate(resized\_2, 45)

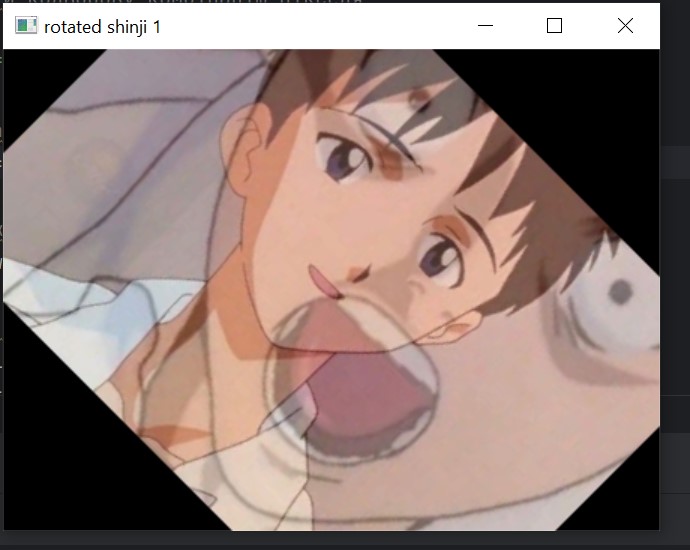


Рис.5 Поворот за розрахунками

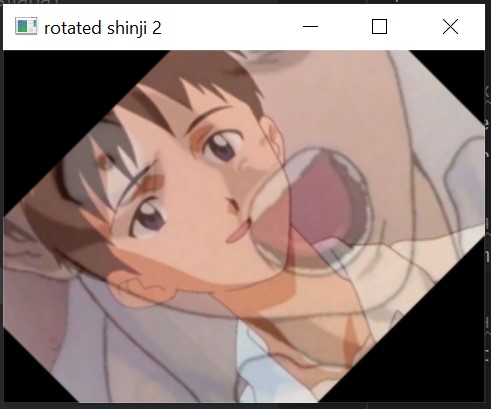


Рис.6 Поворот за допомогою imutils

Для того, аби розмити зображення було використано Блюр по Гаусу. Влаштована функція приймає такі параметри, як зображення, розмір ряда фільтра (ширина та висота) та стандартне відхилення за горизонтальними та вертикальними напрямками. Для порівняння початкове та розмите зображення склеїла.

#розмите зображення  
blurred = cv2.GaussianBlur(resized\_2, (11,11), 0)  
  
#склеювання  
collage = np.hstack((resized\_2, blurred))

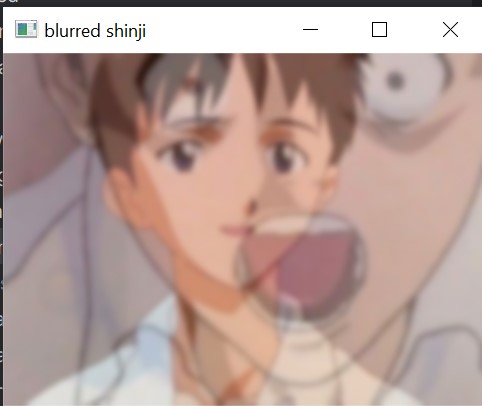


Рис.7 Розмите зображення

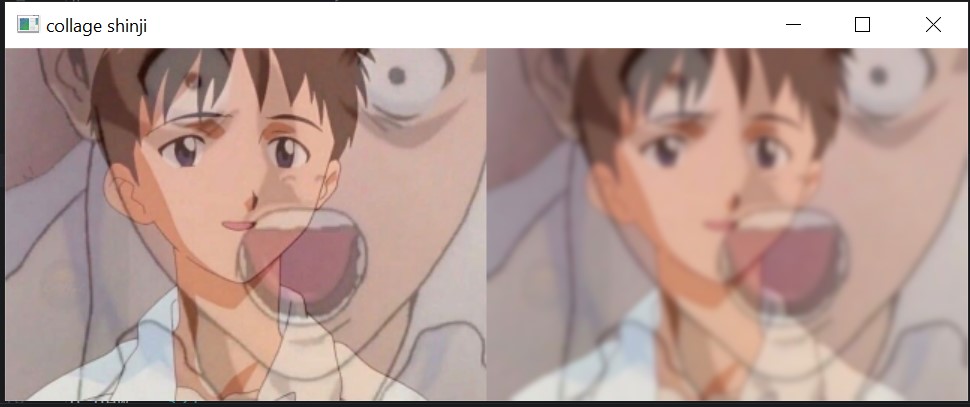


Рис.8 Порівняння зображень

Тепер на вихідному зображені потренуюсь малювати геометричні фігури. Спочатку намалюємо прямокутник. Для цього скористуємось cv2.rectangle(), який приймає зображення, дві координатні точки, колір та товщину ліній.

cv2.rectangle(img, (150,170), (450, 70), (0,0,255), 3)

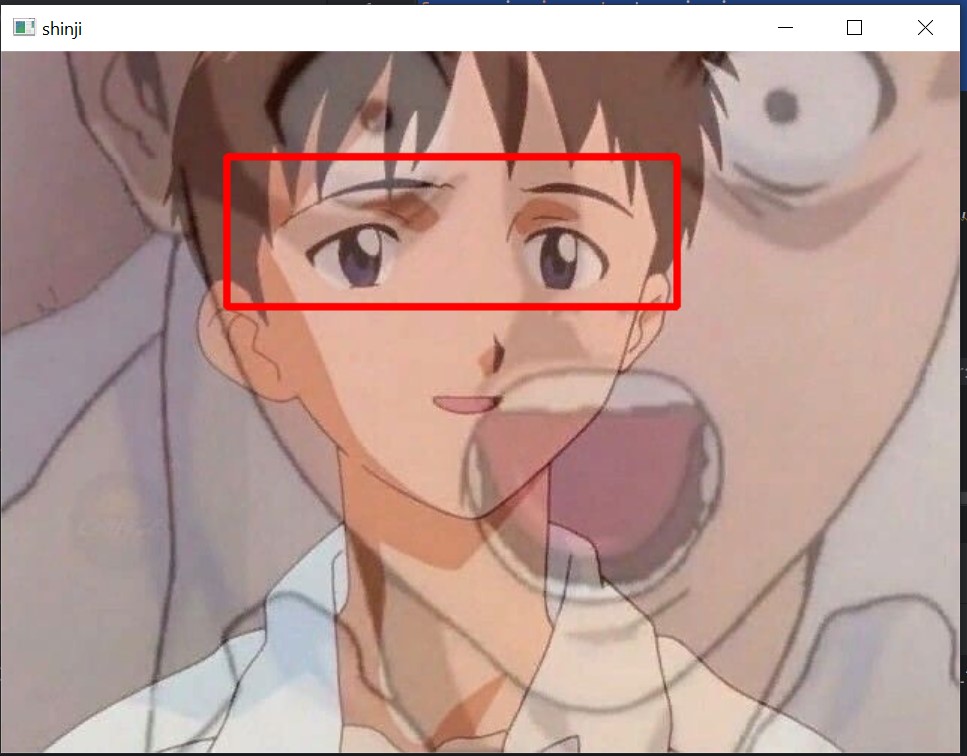


Рис.9 Прямокутник

Аби намалювати лінію скористаюсь cv2.line(), що приймає зображення, початкову та кінцеву координату, колір та товщину ліній.

cv2.line(img, (0,0), (610,468), (255,0,0), 3)

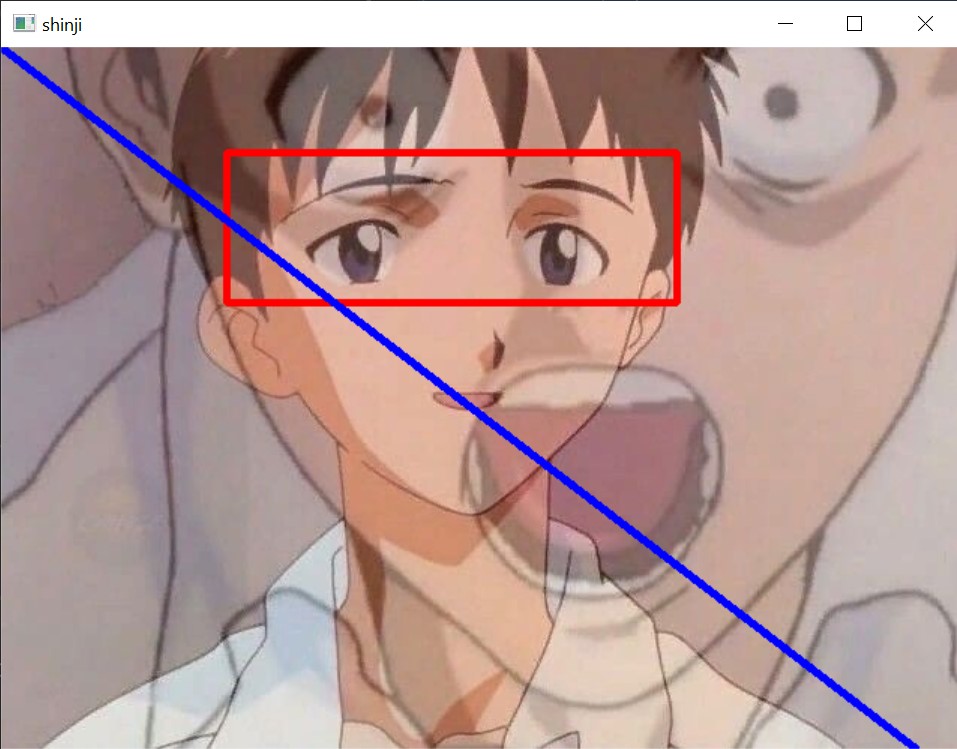


Рис.10 Лінія

Трикутник вже вважається полігональною фігурою. Спочатку задамо масив с координатами та використаємо в polylines().

points = np.array([[198, 231], [315, 311], [400, 233], [198, 231]])  
cv2.polylines(img, np.int32([points]), 1, (255, 255, 255), 3)

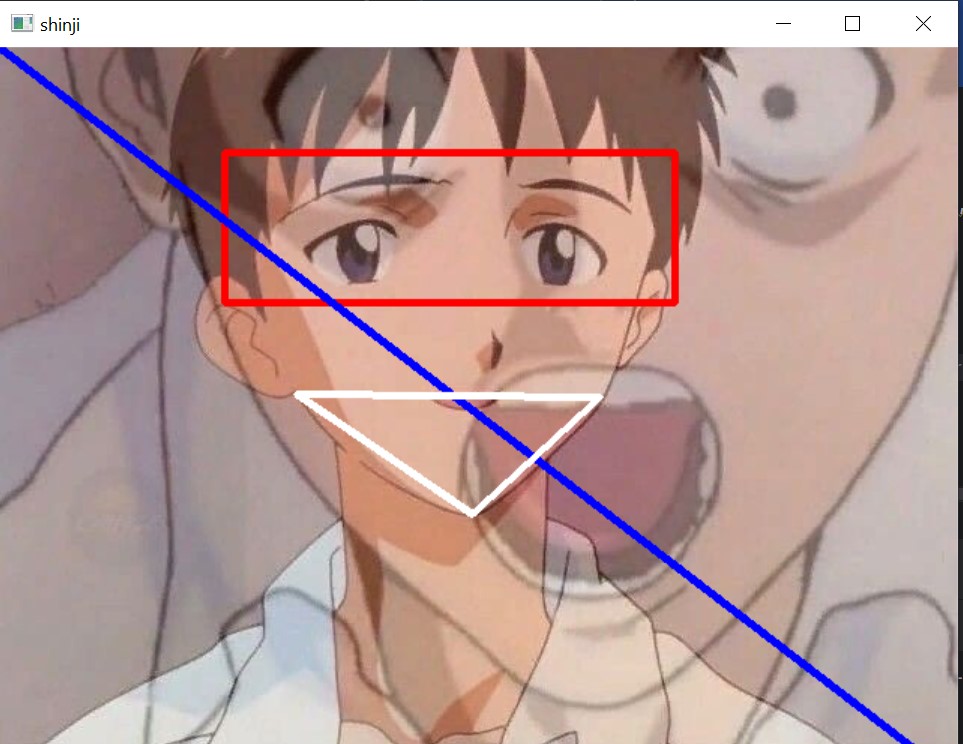


Рис.11 Трикутник

Для кола необхідно зазначити зображення, координату центру та радіус.

cv2.circle(img, (397, 289), 85, (0, 0, 255), 3)

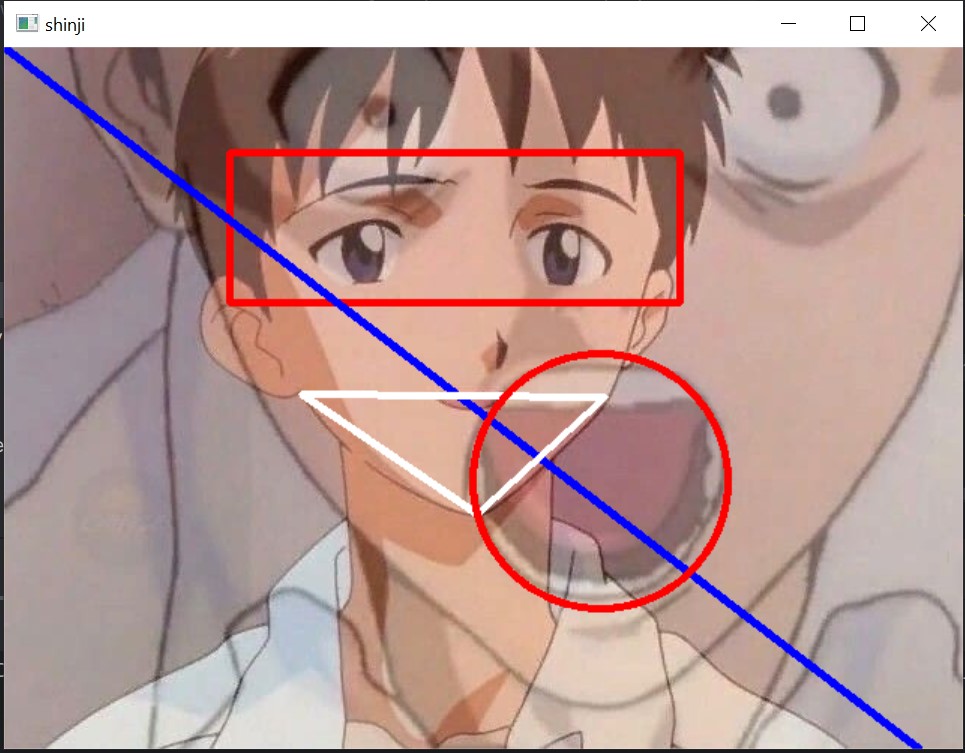


Рис.12 Коло

Тепер додамо напис. Визначимо шрифт та використаємо cv2.putText(), куди передамо зображення, текст, положення, шрифт, колір, товщину.

font = cv2.FONT\_HERSHEY\_SCRIPT\_COMPLEX  
cv2.putText(  
 img, 'evangelion', (30,420), font, 2, (255,255,255), 4, cv2.LINE\_4  
)

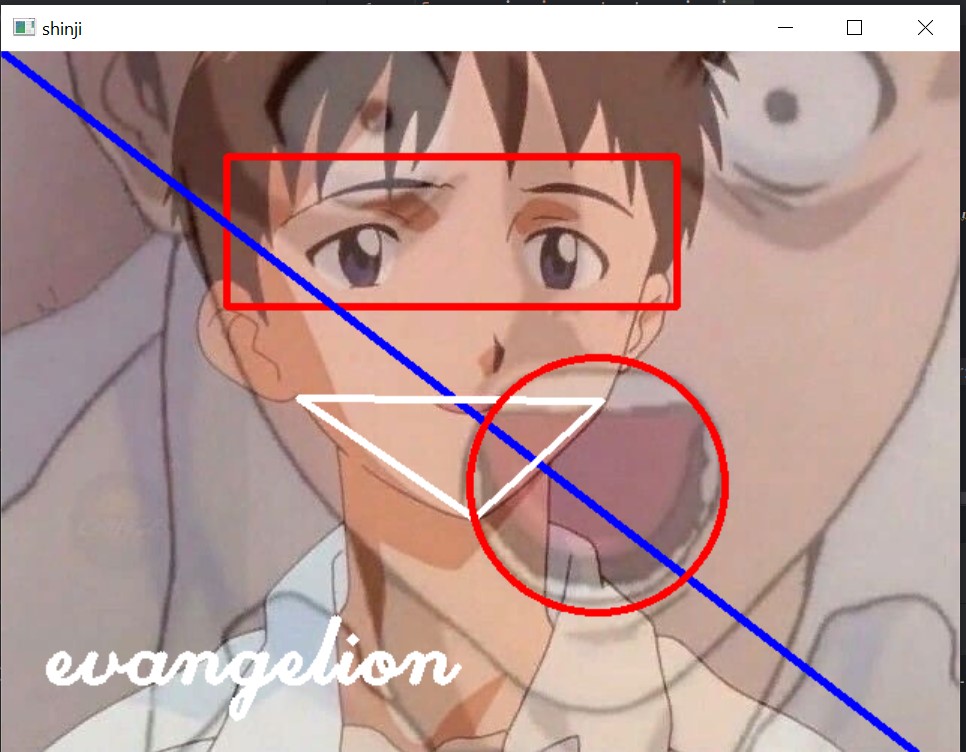


Рис.13 Напис

# Контрольні питання

1. Що таке OpenCV?

OpenCV — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях, відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях.

1. Що таке піксель?

Піксель - це найменша одиниця дискретизації на цифровому зображенні або екрані. Він представляється як точка з певним кольором або інтенсивністю на зображенні. Кожен піксель має свої координати у просторі зображення та значення, що визначає його колір або яскравість. У піксельних зображеннях кольори представлені різними значеннями каналів (наприклад, червоний, зелений, синій для RGB) або яскравості (у відтінках сірого).

1. Як встановити пакет opencv-python?

OpenCV має огортку у вигляді Python модулю. Встановити цей модуль можно за допомогою пакетного менеджера pip. Необхідно прописати:

Pip3 install opencv-python

1. Як розмити зображення?

Для того, аби розмити зображення можна використати Блюр по Гаусу. Влаштована функція приймає такі параметри, як зображення, розмір ряда фільтра (ширина та висота) та стандартне відхилення за горизонтальними та вертикальними напрямками.

cv2.GaussianBlur(resized\_2, (11,11), 0)

1. Які функції використовуються для малювання та відображення тексту?

|  |  |
| --- | --- |
| line() | лінія |
| rectangle() | прямокутник, квадрат |
| circle() | коло |
| polylines() | трикутник та будь який багатокутник |
| putText() | текст |

# Висновок

Отже, під час цієї лабораторної роботи я ознайомилась з операціями читання, відображення та збереження зображень з використанням пакету для мови програмування Python.

Масштабувала, повертала та вирізала частини зображення. Також малювала геометричні фігури, такі як коло, прямокутник, лінія, трикутник та додала текст.