# 1 Робота з геометричними фігурами

Бібліотека для роботи з зображеннями OpenCV надає широкі можливості і їх частиною є функції для накладання на зображення різноманітних геометричних фігур, наприклад: прямокутників, еліпсів, ліній і т.д..

Такі функції, зазвичай, використовуються для виділення певних об’єктів на зображенні, щоб акцентувати на них увагу людини або штучного інтелекту.

Ціль цього модуля полягає в тому, щоб навчитись використовувати функції накладання геометричних фігур на зображення.

Під час виконання вище описаного завдання було:

* Проведено аналіз теоретичних відомостей про запропоновані функції: перелік аргументів та їх допустимі значення.
* Протестовано кожну з функцій фігур у роботі з різними наборами аргументів.
* Протестовано можливість використання «генератора випадкових чисел» як аргумента для використовуваних функцій.

Результат роботи програми

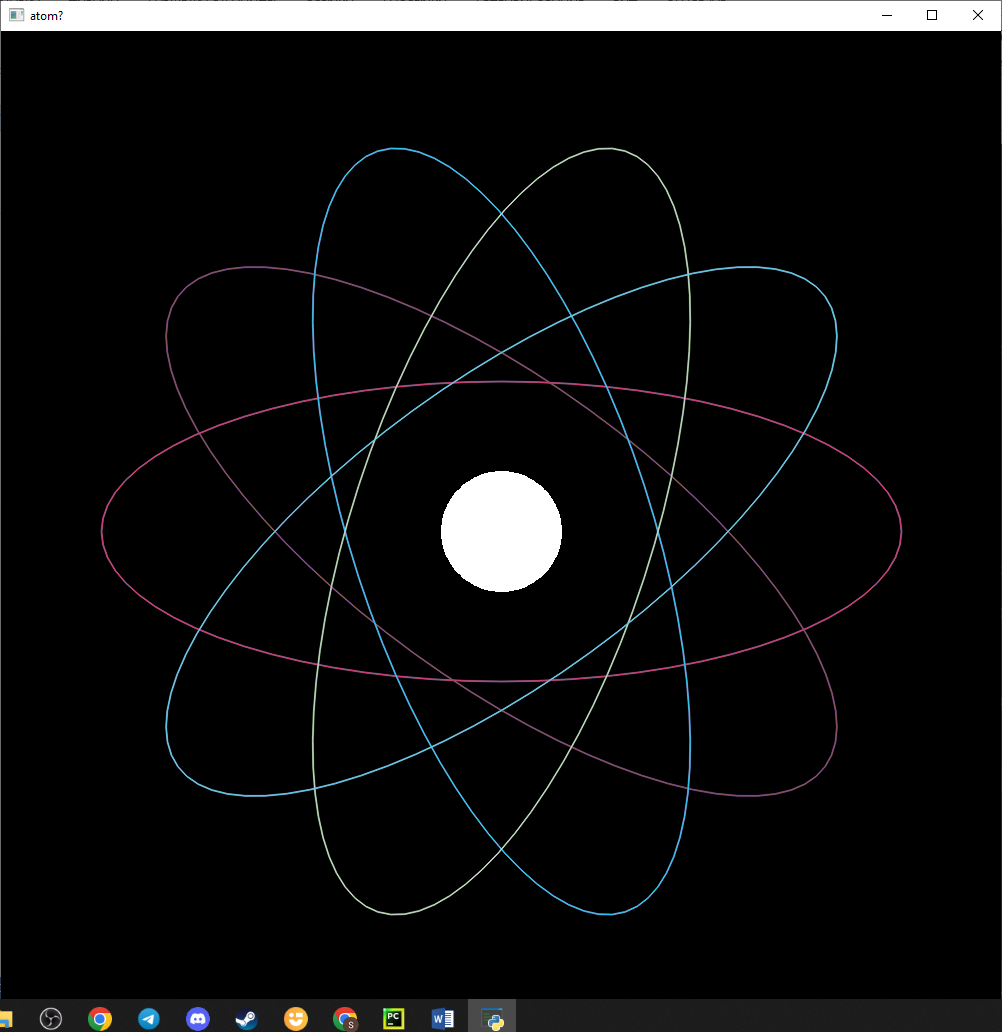


Рисунок 1.1 – Використання функцій ellipse() та circle().

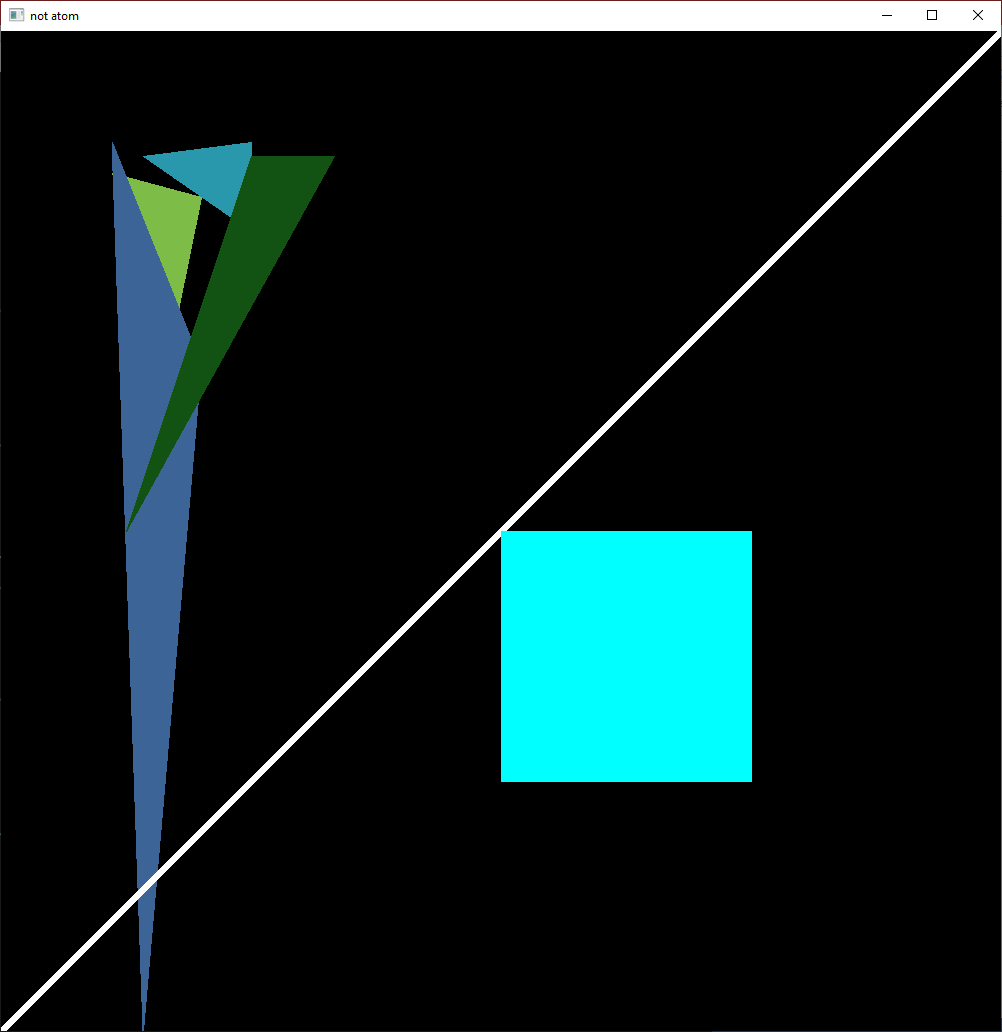


Рисунок 1.2 – Використання функцій line(), rectangle() та fillPoly().

Результатом роботи стали два вікна, на одному з яких показано виведення фігур за чітким алгоритмом, а на іншому – частина фігур (полігони) виведена з випадковими аргументами координат та кольору.

Поставлене завдання було виконано шляхом тестування запропонованих функцій з різними аргументами, заданими як вручну, так і за допомогою «генератора випадкових чисел».

Програмний код, написаний для тестування вище описаних функцій знаходиться в папці «code\venv» у файлі «module\_1\_drawing\_basic\_objects.py».

# 2 Фільтри розмиття зображень

Можливість накладання на зображення ефектів розмиття (blur) є корисною функцією у випадках, коли потрібно зменшити чіткість зображення задля зменшення шуму або його деталізації.

Роботу ефектів розмиття було перевірено з використанням чотирьох різних функцій з бібліотеки opencv.

blur() – функція для накладання розмиття на зображення за допомогою зсуву пікселів чітко по осях «Х» та «Y» на певне значення, яке передається одновимірним масивом в якості аргументу функції. Результат роботи функції показано на рисунку 2.1, де верхнє зображення є оригінальним, а нижнє – це оригінальне зображення з накладеним на нього ефектом розмиття. Приклад роботи наступних функцій буде показано аналогічним способом.



Рисунок 2.1 – Накладання звичайного ефекту розмиття.

У наведеному прикладі розмиття накладається лише по осі «Y», тому деталізація суттєво зменшилась, але можливим є і розмиття по двох осях, що дозволяє досягнути зменшення деталізації без втрати відносно чітких меж. Зсув пікселів чітко по двох осях має свій недолік – різке розмиття, спровоковане зсувом лише по прямій.

Наступна функція – це Гаусове розмиття або GaussianBlur(). Вона схожа на попередню функцію розмиття (blur()) хоча б тим, що зсув проходить чітко по двох осях, але є і суттєві відмінності, наприклад:

* Менш різкий зсув, що дозволяє зберегти межі об’єктів близькі до оригіналу не зважаючи на зменшення деталізації.
* Наявність додаткового, обов’язкового аргументу «sigmaX», який і робить ефект цієї функції суттєво іншим відносно попередньої, бо дозволяє збільшити або зменшити різкість зсуву.

Приклад роботи функції розмиття GaussianBlur() в порівнянні з оригінальним зображенням показано на рисунку 2.2, а різницю між попередньою функцією та GaussianBlur(), з ідентичними аргументами, показано на рисунку 2.3.



Рисунок 2.2 – Накладання ефекту Гаусового розмиття.

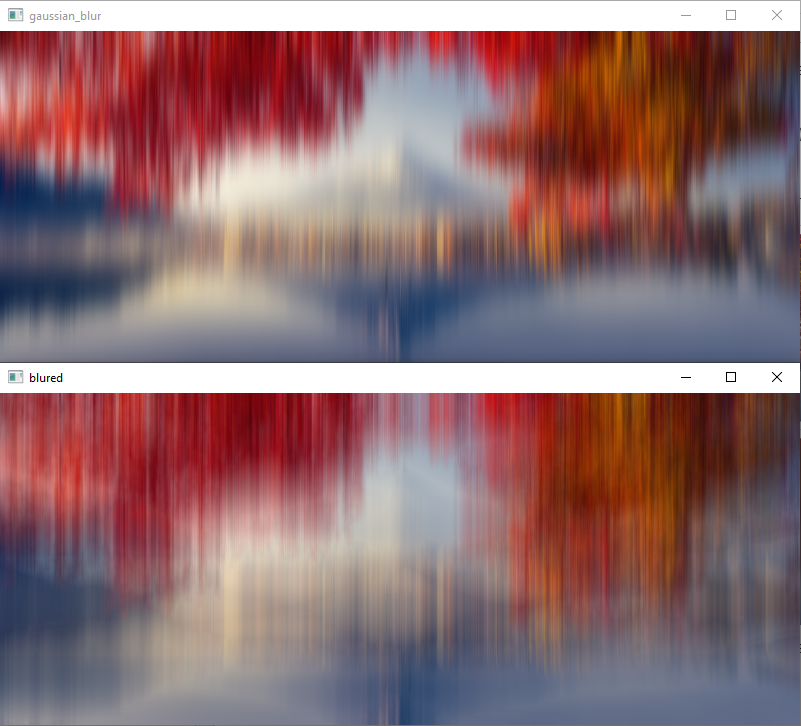


Рисунок 2.3 – Різниця роботи функцій GaussianBlur() та blur().

Наступна суттєво відрізняється від двох вище описаних функцій як результатом і принципом роботи, так і за кількістю аргументів, а її назва – medianBlur().

Медіана – це щось, що розміщається посередині чогось, наприклад число в центрі масиву чисел, і ця функція працює за схожим принципом.

Ця функціє вимагає лише один чисельний аргумент, який і відповідає за розмиття зображення.

Зображення є великим двовимірним масивом чисел і числовий аргумент функції medianBlur() відповідає за кількість елементів масиву, між якими і буде знаходитись медіана, саме тому цей аргумент є непарним, що і спричиняє зсув (розмиття) на самому зображені.

Результатом роботи цієї функції є зменшення деталізації та чіткості зображення, але межі об’єктів стають більш плавними.

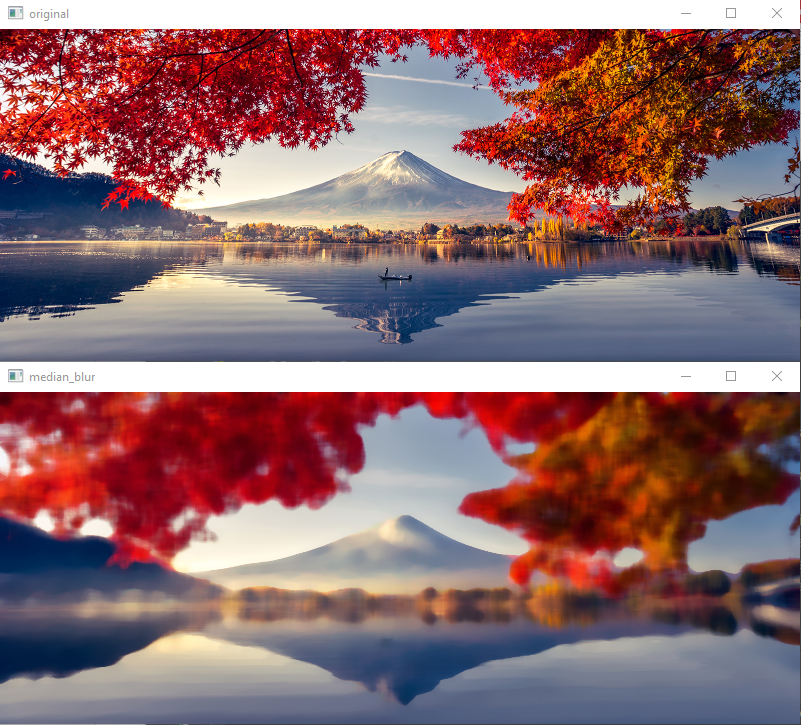


Рисунок 2.4 – Робота функції medianBlur().

Останньою функцією для зменшення деталізації зображення яку було протестовано є bilateralFilter(). В цій функції передбачено використання трьох числових аргументів. Суть роботи цього фільтра полягає в тому, щоб виявити області навколо пікселів з певним діаметром (аргумент 1) коліри в яких відрізняються не більше певного числа (аргумент 2) відстань між якими не перевищує значення третього аргумента, щоб в результаті зменшити різницю в кольорах між цими областями.

Отже, ця функція дозволяє зменшити кількість кольорів в певних областях, що дозволяє полегшити класифікацію об’єктів на зображенні за кольором.



Рисунок 2.5 – Робота функцій bilateralFilter() і putText().

Також було використано функцію додавання тексту на зображення putText(), що показано на рисунку 2.5.

Програмний код, написаний для тестування вище описаних функцій знаходиться в папці «code\venv» у файлі «module\_2\_using\_blur\_effects.py».