Лабораторна робота №3

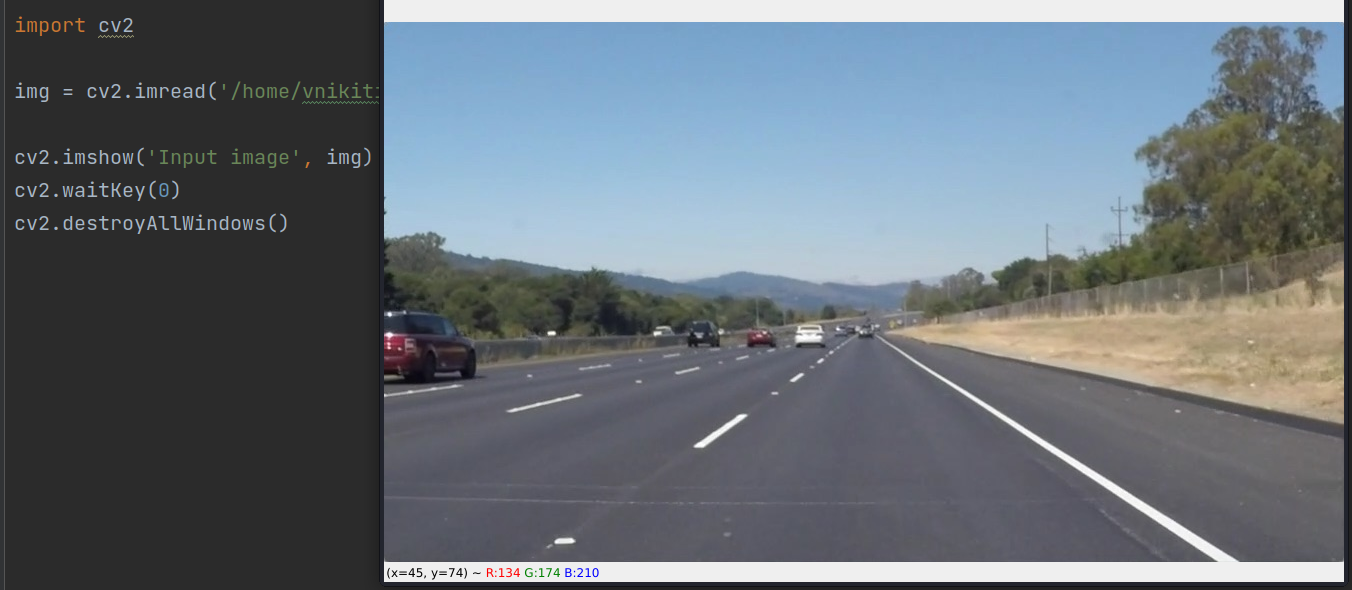
Розмітка дорожньої лінії засобами OpenCV

**Мета:** навчитися виконувати розмітку дорожних ліній на відео засобами

OpenCV в режимі реального часу

**ТЕОРЕТИЧНІ ВІДОМОСТІ:**

Завантаження зображення:

Рисунок 1 — Читання та відображення вихідного файлу

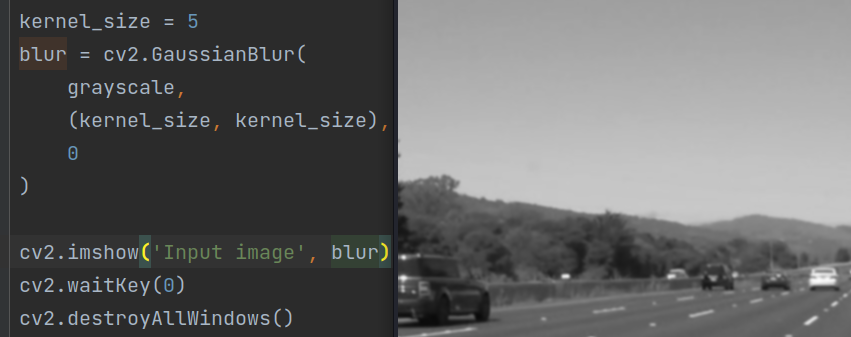
Предобробка даних, фільтрація від шуму та векторизація зображення:

1) конвертуємо в сіре зображення

Рисунок 2 — Конвертація в сіре зображення

Для багатьох етапів предобробки зображення ідеально перетворити його на зображення у градаціях сірого, оскільки простіше працювати із зображенням лише з 1 каналом (без необхідності враховувати 3 канали в кольорових зображеннях), а поставлене завдання робить не потрібно, щоб зображення було кольоровим, оскільки смуги все ще можна спостерігати на зображеннях у відтінках сірого, заощаджуючи багато часу та ресурсів на обчислення.

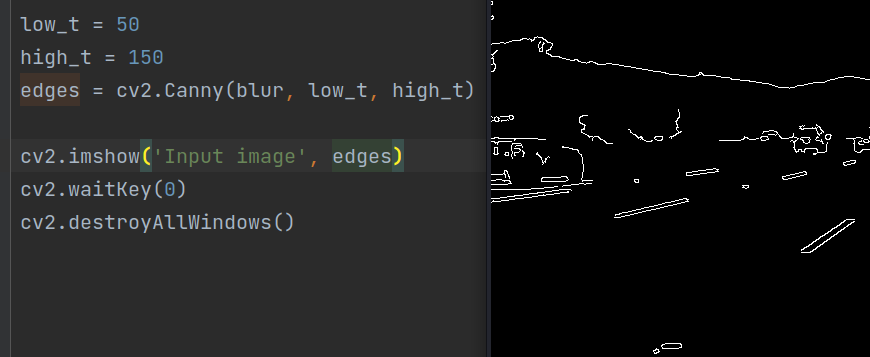
2) розмивання за фільтром Гаусса

Рисунок 3 — Розмиття зображення

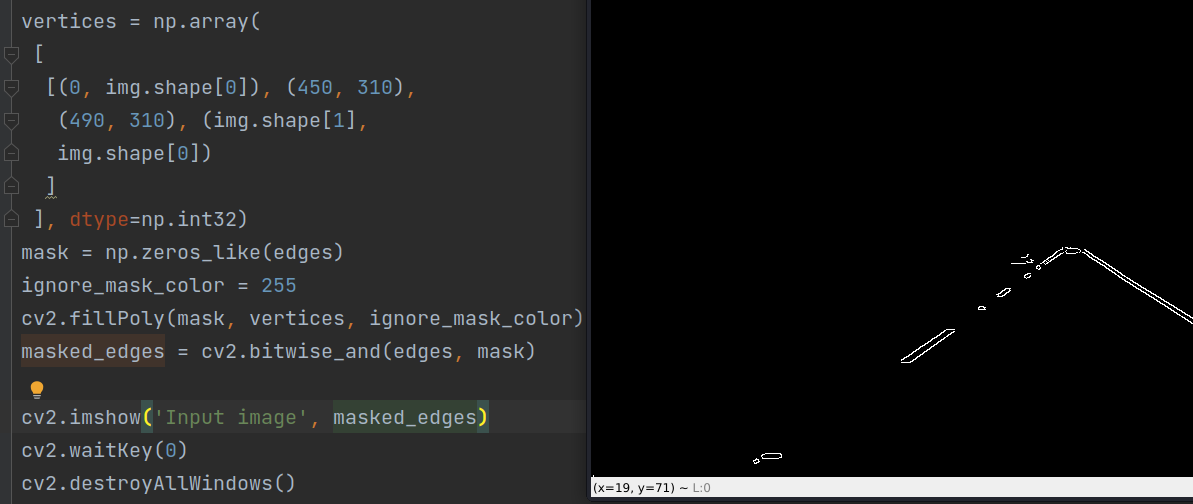
Як правило, причиною застосування розмиття до зображень є те, що ми хочемо видалити шум і згладити зображення. Вхідний кадр або зображення зазвичай містить багато небажаного шуму, який іноді сприяє помилковим виявленням. Таким чином, розмиття зображення усуває багато непотрібного шуму, зберігаючи наш об’єкт інтересу (смуги) достатньо видимим, щоб його можна було виявити.

3) застосування алгоритму Кенні

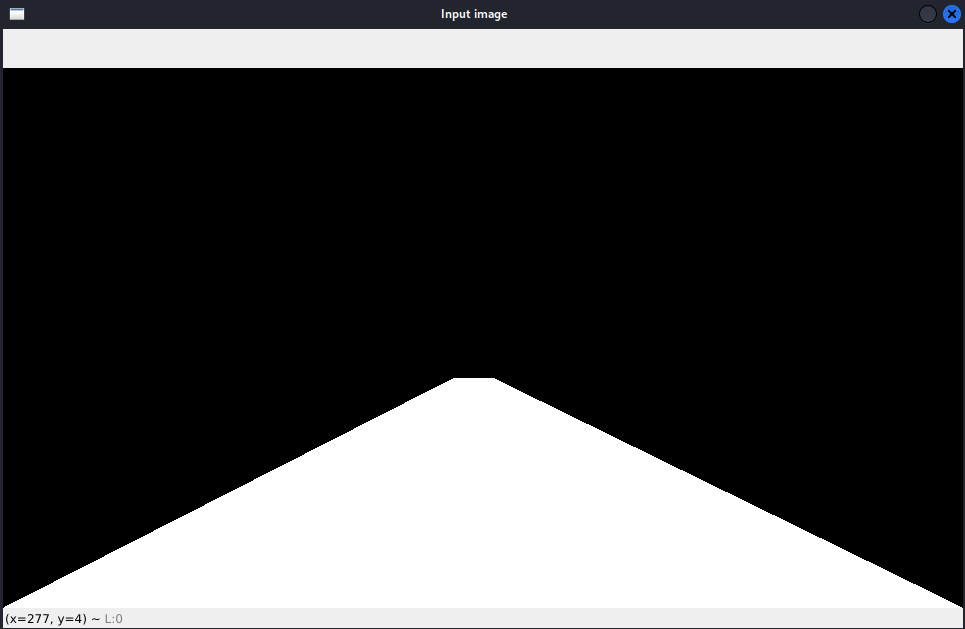
Даний алгоритм оптимального визначення меж, який розраховує градієнти інтенсивності зображення, а потім за допомогою двух порогів видаляє незначні межі, залишая лише потрібні. Результат зображено на рис. 5.

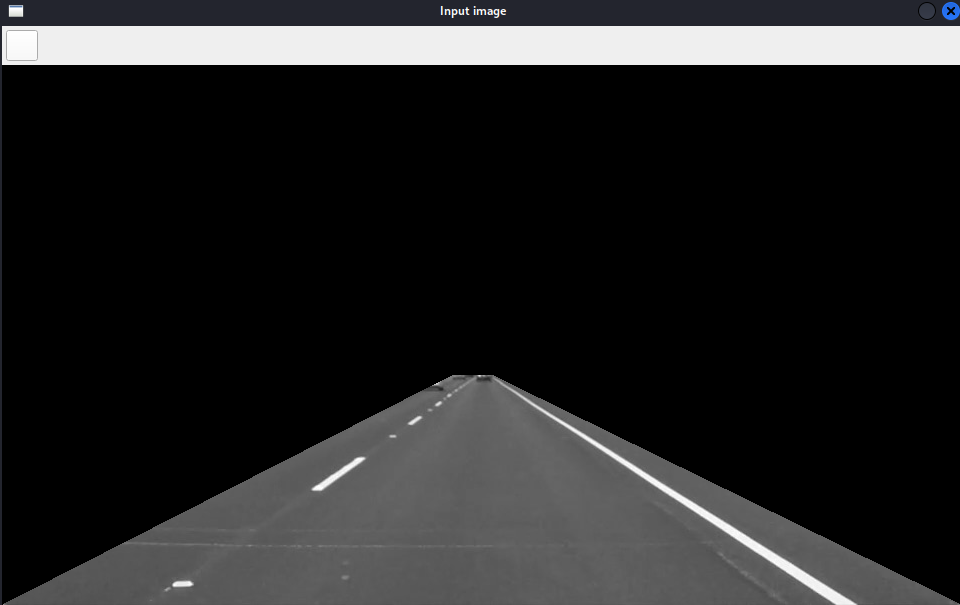
Рисунок 4 — Застосування алгоритму Кенні

4) створення набору вершин для маски

Рисунок 5 - створення набору вершин для маски

cv2.fillPoly заповнює область, визначену вершинами, білими пікселями (ignore\_mask\_color = 255), і ми об’єднуємо знайдений край і маску разом за допомогою cv2.bitwise\_and.

Рисунок 6 — маска

Рисунок 7 — замасковане зображення

5) перетворення Хафа

Щоб знайти лінії на зображенні, ми використовуємо перетворення ліній Гафа. У перші роки навчання в школі нас вчили «за замовчуванням» рівняння лінії y = mx + c. Насправді це декартове представлення можна трансформувати в полярну систему з параметрами (ρ, θ) замість (m, c). Тоді рівняння набуває вигляду:

ρ = x cos (θ) + y sin (θ)

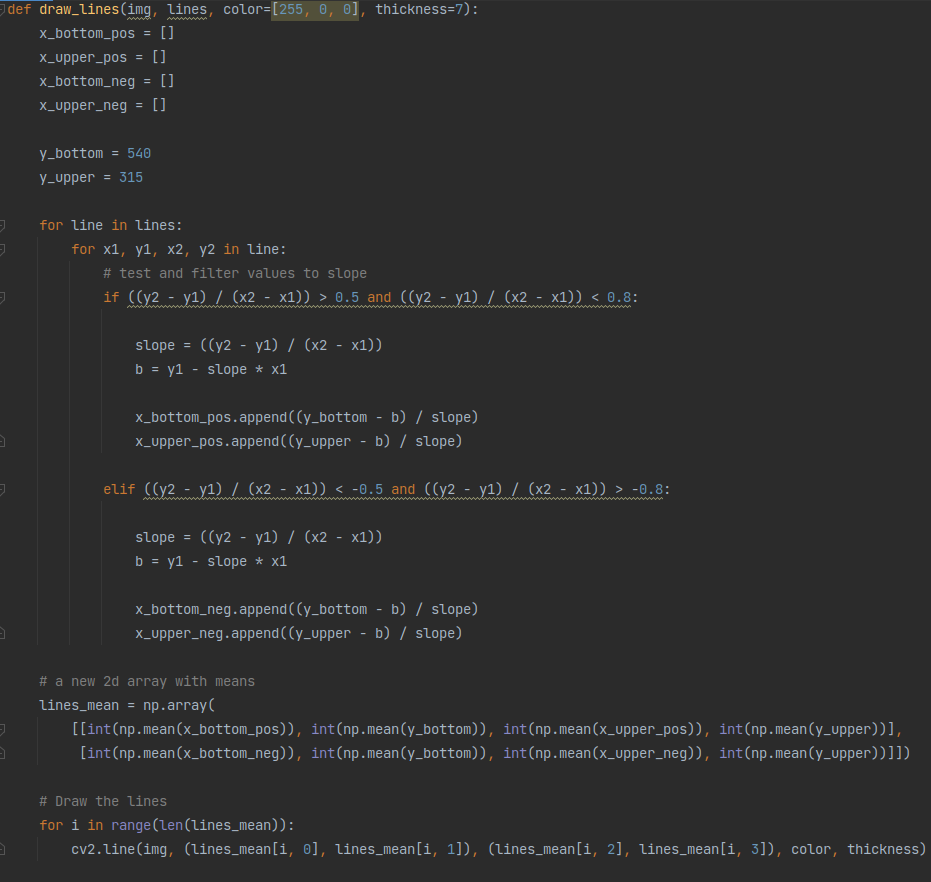
Рисунок 8 — функція нанесення ліній

Рисунок 9 — застосування перетворення Хафа

Для розпізнавання розмітки на відео достатньо перенести написаний код у функцію process\_image і застосувати її до кожного фрейму з відео (рис. 10).

Рисунок 10 - обробка відеофайлу

**ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №3**

Перед виконанням лабораторної роботи рекомендується ознайомитись з відповідним розділом лекційного матеріалу курсу та засвоїти теоретичний матеріал.

**ВИМОГИ ДО ОФОРМЛЕННЯ ЗВІТУ**

Звіт з лабораторної роботи обов’язково повинен містити наступну інформацію:

* назва комп’ютерного практикуму;
* мета роботи;
* відповіді на завдання у текстовому форматі та графічними зображеннями за необхідності.

**Завдання на лабораторну роботу:**

1. Проробити з будь-якою фотографією процедури, які описані в теоретичних відомостях;
2. Зробити розпізнавання розмітки з будь-якого відеофайлу.

**Контрольні запитання:**

1. Чому краще обробляти сіре зображення?

2. Навіщо робити розмиття зображення перед розпізнаванням?

3. Що таке алгоритм Кенні?

4. Що таке перетворення Хафа?

5. Яка мета маски?

**Навчальні матеріали та ресурси:**

* OpenCV. URL: <https://docs.opencv.org/4.x/>
* opencv-python:URL: <https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html>