**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Факультет комп'ютерних наук та кібернетики**

**РЕФЕРАТ**

на тему: **«Підрахунок кількості об'єктів, що перетинають певну лінію на сцені»**

Виконав:

Студент групи ТК-41

Спeцiaльнocтi «Iнфopмaтикa»

Франчук Костянтин Валерійович

**Київ - 2025**

**Зміст**

[Вступ 3](#_is8s4msogaiy)

[Теоретичне підґрунтя 3](#_cav7ol3q7kwj)

[2.1 Комп’ютерний зір 3](#_uvym6eb9221x)

[2.2 Модель YOLOv8 4](#_eryua1xnuzm0)

[2.3 Визначення центроїда об’єкта 4](#_e222c97ng0hi)

[2.4 Трекінг за евклідовою відстанню 4](#_wok10diltlzg)

[2.5 Виявлення перетину лінії 4](#_79q0vxjuo99m)

[Практичне використання 4](#_blbtgaqbx55i)

[3.1 Автоматизований облік транспортного потоку 5](#_773k3de778uw)

[3.2 Системи відеоспостереження та безпеки 5](#_6351wanlv9nb)

[3.3 Використання у розумних містах 5](#_1m89pqs2g3kq)

[3.4 Гнучкість та масштабованість 5](#_az9q96lkyfe7)

[Опис реалізації програми 6](#_n4ze9huey23u)

[4.1 Завантаження моделі та відео 6](#_vrrdx89evmrm)

[4.2 Обробка кадру та виявлення об’єктів 6](#_og99p1370kfh)

[4.3 Визначення центроїдів об’єктів 6](#_yqojgdrs8lh3)

[4.4 Простий трекінг об’єктів 7](#_1lb9t7lhf2n9)

[4.5 Визначення перетину контрольної лінії 7](#_iu1fz8ego6ip)

[4.6 Відображення результатів 7](#_tcr8ns11fkiv)

[4.7 Оптимізація швидкодії 7](#_x7j3sarrrjuj)

[Висновки 8](#_1n6ua92onmwc)

[**Список використаних джерел 9**](#_4kxq0qddixlw)

### **Вступ**

У сучасному світі зростає потреба в автоматизованих системах моніторингу дорожнього руху, особливо у великих містах, де кількість транспортних засобів постійно збільшується. Ручний підрахунок автомобілів на перехрестях, шосе або в зоні паркування є неефективним, трудомістким і суб'єктивним. Саме тому виникає необхідність створення програмного забезпечення, яке здатне автоматично аналізувати відеопотоки та підраховувати кількість транспортних засобів.

Метою даної роботи є розробка програми, що здійснює підрахунок автомобілів, які перетинають певну контрольну лінію у відеозаписі. У процесі розробки використовуються сучасні технології комп’ютерного зору, зокрема бібліотека OpenCV [1], а також глибокі нейронні мережі, зокрема модель YOLOv8 [2], яка забезпечує високу точність виявлення об’єктів у реальному часі. YOLO відома своєю здатністю швидко та ефективно виконувати задачі розпізнавання об’єктів на зображеннях і відео.

Реалізована програма має чітке практичне значення: вона може бути використана для аналізу дорожньої ситуації, регулювання транспортних потоків, обліку автомобілів на в’їздах до територій, у зонах паркування або в межах систем міста. Також ця розробка демонструє ефективне поєднання штучного інтелекту та класичних алгоритмів трекінгу в реальних прикладних задачах.

### **Теоретичне підґрунтя**

Розробка системи автоматичного підрахунку автомобілів базується на технологіях комп’ютерного зору та методах глибокого навчанняЮ, а саме модель YOLOv8, бібліотеку OpenCV.

#### **2.1 Комп’ютерний зір**

Комп'ю́терний зір або Комп'ютерне бачення — теорія та технологія створення машин, які можуть проводити виявляння, відстежування та визначення об'єктів. У контексті даної роботи він використовується для обробки відео, виявлення об'єктів (для зручності і наглядності було обрано автомобілі, але можна розширити і до інших об’єктів) та аналізу їхніх координат для виявлення перетину умовної лінії.

#### **2.2 Модель YOLOv8**

YOLO (You Only Look Once) — це комп'ютерна технологія, пов'язана з комп'ютерним баченням та обробкою зображень, яка має справу з виявлянням примірників семантичних об'єктів певного класу (таких як люди, будівлі чи автомобілі) у цифрових зображеннях та відео. YOLOv8 — одна з найновіших і найбільш оптимізованих версій, розроблена компанією Ultralytics. Вона поєднує високу точність виявлення з низькою затримкою, що робить її ідеальною для задач реального часу. У даній програмі використовується полегшена версія yolov8n.pt, що забезпечує добрий баланс між швидкістю та якістю [3].

#### **2.3 Визначення центроїда об’єкта**

Центроїдом вважається геометричний центр прямокутника, що охоплює виявлений об’єкт (bounding box). Центроїди використовуються для трекінгу: якщо центроїд об'єкта рухається та перетинає певну координатну лінію (наприклад, як реалізовано у програмі горизонтальну лінію посередині кадру), програма визначає, що транспортний засіб пройшов через контрольну зону.

#### **2.4 Трекінг за евклідовою відстанню**

Для збереження ідентичності об’єктів між кадрами використовується простий алгоритм на основі евклідової відстані між центроїдами.Тобто, якщо відстань між новим центроїдом та існуючим у попередньому кадрі є меншою за встановлений поріг (40 пікселів), то вважається, що це один і той самий об’єкт. Такий підхід дозволяє відстежувати об'єкти без складних моделей трекінгу, забезпечуючи високу швидкість обробки [4].

#### **2.5 Виявлення перетину лінії**

Програма визначає, чи перетнув об'єкт контрольну лінію, шляхом аналізу координат центроїда в попередньому та поточному кадрі. Якщо координата y центроїда змінилась з більшої на меншу відносно лінії (або навпаки) — це вважається перетином, і лічильник збільшується на 1.

### **Практичне використання**

Розроблена програма має широке практичне застосування в реальних умовах, особливо у сфері автоматизації транспортного моніторингу, аналізу трафіку, а також у системах безпеки. Завдяки використанню сучасних інструментів комп’ютерного зору та штучного інтелекту, вона може ефективно виконувати підрахунок транспортних засобів у реальному часі з високою точністю.

#### **3.1 Автоматизований облік транспортного потоку**

Одним із ключових напрямків використання програми є автоматичний облік автомобілів, що перетинають певну ділянку дороги. Це може бути перехрестя, в’їзд на парковку, контрольна зона на платній дорозі або ділянка біля світлофора. Зібрана інформація може використовуватись для аналізу завантаженості дороги, планування транспортної розв’язки чи оптимізації роботи світлофорів.

#### **3.2 Системи відеоспостереження та безпеки**

Програму можна інтегрувати в інтелектуальні системи відеоспостереження для автоматичного моніторингу транспортного руху. Наприклад, у випадку перевантаженого в’їзду на закриту територію система може миттєво попередити охорону або відповідні служби. Або ж у зоні митного контролю або на приватних підприємствах такий інструмент допомагає вести точний облік в’їзду та виїзду автомобілів без участі оператора, що скорочує витрати підприємства на працівників і унеможливлює корупцію та крадіжки.

#### **3.3 Використання у розумних містах**

У концепції «розумного міста» [6] велике значення мають системи, що збирають та обробляють дані про транспортну ситуацію. Програма може стати елементом загальноміської мережі аналітики дорожнього руху. Завдяки накопиченню статистики, міська влада може приймати обґрунтовані рішення щодо інфраструктурних змін, зонування, розширення доріг або впровадження нових маршрутів громадського транспорту.

#### **3.4 Гнучкість та масштабованість**

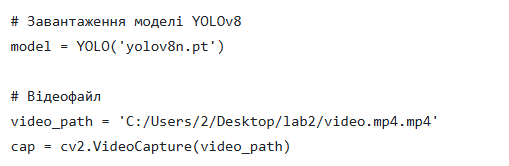
Програма легко адаптується під різні типи відео: вона може працювати як з попередньо записаними файлами, так і з потоковим відео з камер спостереження. Її можна налаштувати для виявлення не лише автомобілів, а й автобусів, вантажівок чи пішоходів. Це досягається зміною класів об’єктів у моделі YOLO. Така гнучкість дозволяє інтегрувати її в різноманітні сфери, від промислових зон до центрів міст.

### **Опис реалізації програми**

Програма реалізована мовою Python з використанням бібліотек OpenCV та Ultralytics YOLOv8. Основна мета — підрахунок транспортних засобів, які перетинають контрольну горизонтальну лінію на відео. Для цього реалізовано послідовність етапів обробки відео, виявлення об'єктів, трекінгу та підрахунку.

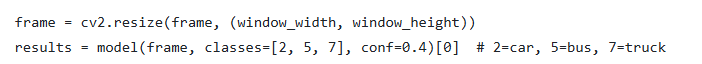
#### **4.1 Завантаження моделі та відео**

На початку роботи відбувається завантаження попередньо натренованої моделі yolov8n.pt, що спеціалізується на задачах детекції об’єктів у реальному часі. Відео зчитується за допомогою cv2.VideoCapture [5].



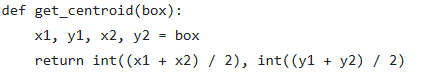
#### **4.2 Обробка кадру та виявлення об’єктів**

Кожен кадр відео зменшується до розміру 800×600 пікселів. Далі модель YOLO застосовується для виявлення транспортних засобів (автомобілів, автобусів, вантажівок), які належать до класів 2, 5 та 7.



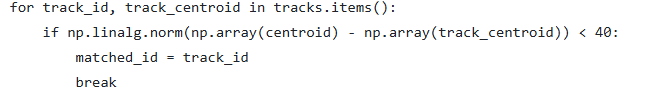
#### **4.3 Визначення центроїдів об’єктів**

Для кожного виявленого об'єкта обчислюється центроїд прямокутника (bounding box). Центроїди використовуються як база для трекінгу.



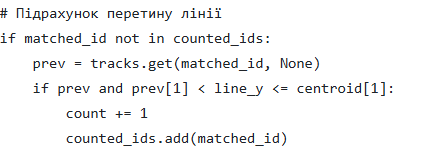
#### **4.4 Простий трекінг об’єктів**

Для спрощення ідентифікації рухомих об’єктів у кожному кадрі застосовується евклідова відстань між центроїдами. Якщо новий центроїд близький до попереднього — вважається, що це той самий об’єкт.



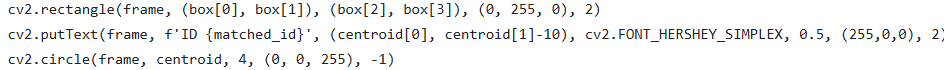
#### **4.5 Визначення перетину контрольної лінії**

У центрі кадру проводиться горизонтальна лінія. Якщо об’єкт рухається й перетинає цю лінію (зміна координати y центроїда з меншої на більшу відносно line\_y), програма зараховує це як перетин.



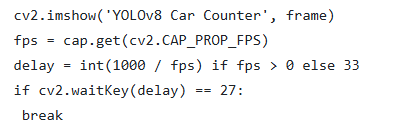
#### **4.6 Відображення результатів**

На кадрі відображається: рамка навколо об’єкта, ID об’єкта, червона лінія контролю, загальний лічильник, швидкість обробки кадрів (FPS).



#### **4.7 Оптимізація швидкодії**

Програма обробляє кожен другий кадр для підвищення продуктивності, а також динамічно визначає затримку відтворення відео на основі FPS:



Посилання на гіт: <https://github.com/Kasat1k/Roac_lab_2/blob/main/main>

### **Висновки**

У ході виконання даної роботи було розроблено програму, що здійснює автоматичний підрахунок автомобілів, які перетинають контрольну лінію на відео. Програма реалізована з використанням бібліотеки OpenCV та сучасної моделі глибокого навчання YOLOv8, яка забезпечує точне та швидке виявлення транспортних засобів у режимі реального часу.

Використання моделі YOLOv8 у поєднанні з алгоритмом трекінгу за центроїдами дало змогу реалізувати точну та стабільну систему підрахунку. Навіть у складних відеоумовах (зміни освітлення, перехресний рух) алгоритм показав надійну роботу. Також, завдяки оптимізованій моделі yolov8n.pt та простій логіці трекінгу програма може працювати в реальному часі без значного навантаження на систему, що робить її придатною для використання на звичайних комп’ютерах. Система може бути ефективно використана у різних сферах життя: обліку трафіку, відеоспостереження, автоматизації паркувальних комплексів, митного контролю, транспортної аналітики в межах розумного міста.

Програму можна розширити, наприклад, додавши збереження статистики до бази даних, розробити графічний інтерфейс та розширити список об’єктів для розпізнавання, але це залежить від потреб користувача.

Таким чином, на мою думку, реалізований проєкт демонструє, як сучасні технології комп’ютерного зору та штучного інтелекту можуть ефективно вирішувати прикладні задачі аналізу відео. Він має потенціал для практичного впровадження в різних галузях і може служити основою для створення масштабніших систем аналітики дорожнього руху.

## **Список використаних джерел**

1. **OpenCV: Open Source Computer Vision Library** [Електронний ресурс]. – Режим доступу:<https://opencv.org/>
2. **Ultralytics YOLOv8 Documentation** [Електронний ресурс]. – Режим доступу: <https://docs.ultralytics.com>
3. Redmon, J., & Farhadi, A. (2018). **YOLOv3: An Incremental Improvement**
4. **PyImageSearch – Object Tracking with OpenCV** [Електронний ресурс]. – Режим доступу:<https://pyimagesearch.com>
5. **OpenCV-Python Tutorials** [Електронний ресурс]. – Режим доступу: <https://docs.opencv.org/>
6. **Smart City** [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D1%83%D0%BC%D0%BD%D0%B5_%D0%BC%D1%96%D1%81%D1%82%D0%BE>