

ЛАБОРАТОРНА РОБОТА №8

Визначення швидкості руху транспортних засобів за відеозаписом

1.1 Мета роботи

Дослідити відеозапис в середовищі нейромережі та скласти графік на базі отриманих даних.

1.2. Короткі теоретичні відомості

You Only Look Once (YOLO) — це серія систем виявлення об'єктів у реальному часі на основі згорткових нейронних мереж. Вперше представлений Джозефом Редмоном та ін. у 2015 році YOLO зазнала кількох ітерацій і вдосконалень, ставши однією з найпопулярніших систем виявлення об'єктів.

Назва «Ви дивитеся лише один раз» означає, що алгоритм вимагає лише одного проходу прямого розповсюдження через нейронну мережу, щоб зробити прогнози, на відміну від попередніх методів на основі регіональних пропозицій, таких як R-CNN, які потребують тисячі для одного зображення.

В програмі виконано пошук об'єктів на відео за допомогою Python і бібліотеки OpenCV, а також бібліотек numpy та art. Нейромережа YOLO була використана для визначення класів на відео послідовності.

З її допомогою визначаються найімовірніші класи об'єктів на зображенні ('person', 'bicycle', 'car', 'motorbike', 'bus', 'truck'), а також координати їх кордонів, які були використані для відображення рамки (прямокутника). Приклад визначення класів об'єктів на зображенні (кадр з відеопослідовності), а саме в даному випадку авто (car) представлено на рис.2.3-2.4.

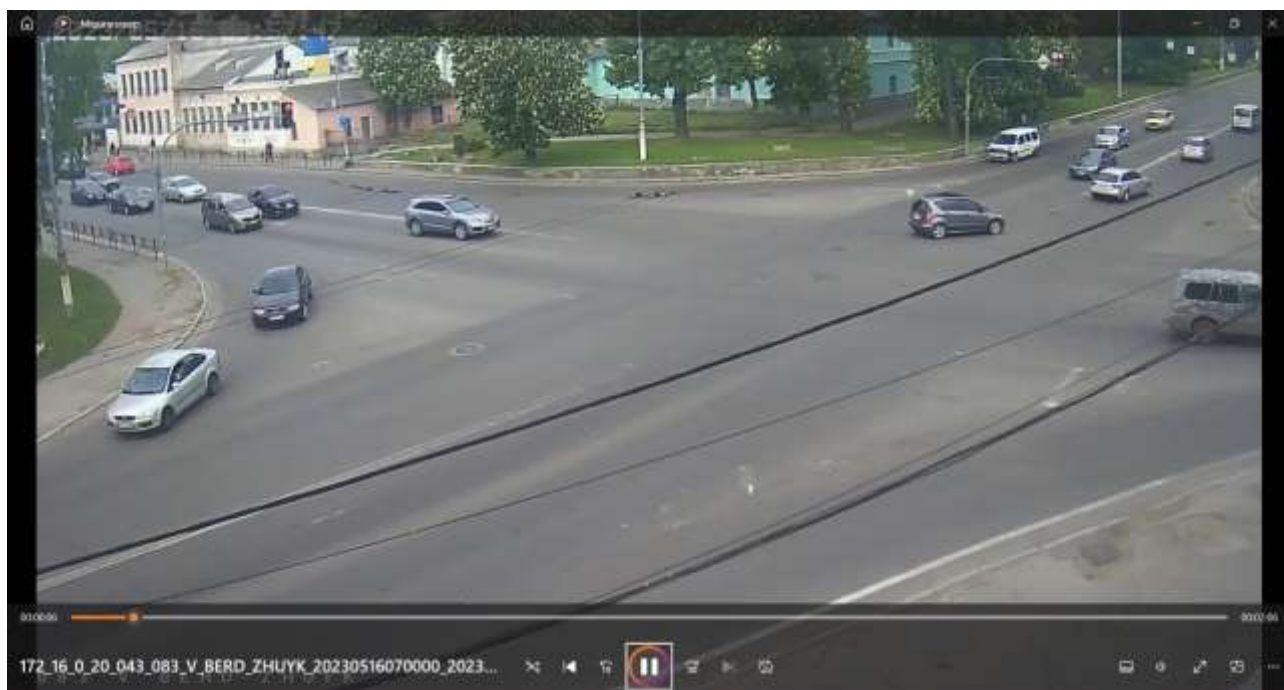


Рисунок 2.3 Кадр з відеопослідовності для подальшого визначення класів об'єктів на зображенні



Рисунок 2.4. Кадр з відеопослідовності з визначеними класами об'єктів (авто, «саг») на зображенні

Головна функція визначає координати X, Y центру мас зображення, потім добуток кожної координати на довжину/ширину, далі математичним чином визначаються координати обмежуючого прямокутника для промальовування рамки (боксу). Код представлений нижче:

```
center_x = int(obj[0] * width)
center_y = int(obj[1] * height)
obj_width = int(obj[2] * width)
obj_height = int(obj[3] * height)
box = [center_x - obj_width // 2, center_y - obj_height // 2,
       obj_width, obj_height]
```

Програма виводить координати центра мас, а також час для кожної пари координат.

```
coordinat.append(center_x)
coordinat.append(center_y)
end_time = time.time() - start_time
sum_time = float(sum_time) + end_time
sum_time = round(sum_time, 4)
final_image = index(center_x, center_y, final_image, coordinat, sum_time)
```

Приклад роботи програми з визначенням класів об'єктів та виведенням координат та час їх зміни приведено на рис.2.5.



Рисунок 2.5. Кадр з відеопослідовності з визначеними класами об'єктів (авто, «саг») на зображенні, визначеними координатами центру мас та часу

Далі отримані дані передаються для визначення переміщення та швидкості руху об'єктів.

Вхідні дані для відеопотоку представлені нижче в програмі:

```
# Capturing a picture from a video
video_camera_capture = cv2.VideoCapture(video)
out = cv2.VideoWriter('captured.avi',1 , 20.0, (640,480))
```

Отже, відповідно 20 кадрів в секунду та розширення зображення 640 на 480.

Центр мас – це уявна геометрична точка, що характеризує розподіл маси в тілі (системі тіл) і радіус-вектор якої виражається через радіус-вектори x_i , y_i окремих точок за формулою (2.1):

$$\vec{x}_c = \frac{\sum m_i x_i}{m_i}, \vec{y}_c = \frac{\sum m_i y_i}{m_i} \quad (2.1)$$

де $\sum m_i = M$.

Центр мас є точкою прикладання не тільки рівнодійної сил тяжіння, але і рівнодійної будь-яких масових сил за умови. Що сили, що діють на елементи тіла (точки системи) паралельні між собою.

Коефіцієнт масштабування для зображення 640 на 480 становить $K=6,2505$. Виміряні дані поточних координат представлені в д.т. Щоб отримати виміряні поточні координати в метрах потрібно перемножити на коефіцієнт масштабування та поділити на 100 (2.2).

$$x_i = \frac{K\vec{x}_c}{100}, y_i = \frac{K\vec{y}_c}{100} \quad (2.2)$$

Переміщення розраховується, як модуль різниці між поточним та попереднім значенням координати. Відповідно, вважаємо рух транспортного засобу зліва направо та зворотно по координаті x_i , зверху вниз та зворотно по координаті y_i (2.3).

$$r_i = |x_i - x_{i-1}| \quad (2.3)$$

Поточну швидкість v_i розраховується, як модуль різниці між наступним значенням та попереднім поділений на час кадру. Час кадру t_i в програмі не фіксований (2.4).

$$v_i = \frac{|x_{i+1} - x_{i-1}|}{t_i} \quad (2.4)$$

Далі обчислюються загладжувані координати, переміщення та швидкість по формулі 2.5:

$$\hat{\theta}_n = (1 - \xi) \theta_n + \xi \hat{\theta}_{n-1} \quad (2.5)$$

де θ_n – вимірне значення параметра на n -кроці; $\hat{\theta}_n$ – зглажене значення параметра в n -м огляді; ξ – стала величина, що має сенс коефіцієнта згладжування. [7]

Коефіцієнт згладжування обирається для кожного транспортного засобу враховуючи інтенсивність маневру. При дослідженні було відібрано три екземпляра транспортних засобів: легкове авто, автобус на дев'ять посадкових місць та вантажівка. В програмі нейромережею YOLO це ідентифікувалося відповідно car, bus, truck.

Якість експоненційного згладжування оцінюється за величиною випадкових та динамічних помилок згладжування.

Дисперсія відносної сумарної помилки згладжувальних дорівнює

$$\frac{\sigma_\Sigma^2}{\sigma_\theta^2} = \frac{(1 - \xi)^2}{(1 + \xi)} + \frac{\xi^2}{(1 - \xi)^2} s^2, \quad (2.6)$$

де $s^2 = \frac{(\Delta^{(1)} \theta_n)^2}{\sigma_\theta^2}$ - відносна інтенсивність можливих змін параметрів руху.

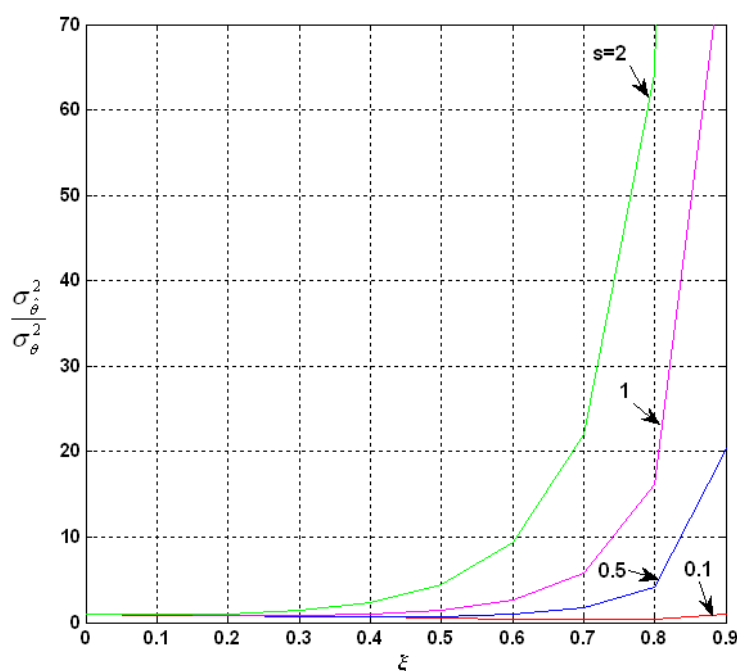


Рис.2.6. Графіки залежності дисперсії відносної сумарної помилки згладжування параметра від коефіцієнта згладжування ξ при різних інтенсивностях маневру

На рис. 2.6 побудовані графіки залежності дисперсії відносної сумарної помилки згладжування параметра від коефіцієнта згладжування ξ при різних інтенсивностях змін цього параметра. Графіки мають екстремуми типу мінімуму, які зміщуються вліво при збільшенні інтенсивності змін. Звідси можна зробити висновок про доцільність зміни

коефіцієнта згладжування ξ з метою підтримки похибки на мінімальному або близькому до мінімального рівня для кожного конкретного значення інтенсивності змін параметра руху ОУ. З графіків похибки також випливає, що вибір коефіцієнтів ξ для кожного значення інтенсивності змін не критичний, так як екстремуми графіків пологі.

Для знаходження оптимального значення ξ_{opt} за критерієм мінімуму дисперсії сумарної помилки продиференціюємо вираз (2.6) по ξ і прирівняємо результат до нуля. Остаточне рівняння для ξ_{opt} має вигляд (2.7):

$$(1 - \xi_{opt}^2)(1 - \xi_{opt})^3 + 2(1 - \xi_{opt})^4 - 2\xi_{opt}(1 + \xi_{opt})^2 s^2 = 0. \quad (2.7)$$

Рис.2.7. Графіки знаходження оптимального значення ξ_{opt} за критерієм мінімуму дисперсії сумарної помилки

1.3. Підготовка до роботи

При підготовці до роботи необхідно:

- Ознайомитися з теоретичними відомостями
- Встановити програму нейромережу

1.4. Порядок роботи:

1. Встановити програму нейромережі
2. Запустити наданий відеофайл в середовищі нейромережі
3. Отримати дані обробки відеофайлу
4. Побудувати графік на базі отриманих даних.