Науково-практичний звіт на тему

“Локалізація та розпізнавання дорожніх знаків по камері відеоспостереження”

Бегішева В. О., студентка 4 курсу

механіко-математичного факультету,

групи комп’ютерна математика

Анотація.

Робота присвячена дослідженню моделей машинного навчання для задачі локалізації та розпізнавання знаків дорожнього руху на відео. Основною метою дослідження було натренувати модель, що здатна локалізувати та відслідковувати дорожні знаки, використовуючи відео з камери, встановленої на лобовому склі автомобіля, а також визначати тип знаку. Для тренування моделі був використаний набір даних Mapillary Traffic Sign Dataset. В роботі була створена всласна модель локалізації та розпізнавання обєктів. Але вона не працювала коректно. Тому була натренована вже існуюча модель yolov5s на запропонованому наборі данних. Робота моделі була перевірина на зображеннях та безпосередньо на відео.

Abstract.

This work is dedicated to the study of machine learning models for the task of road sign localization and recognition in video. The main goal of the research was to train a model capable of localizing and tracking traffic signs using video from a windshield-mounted camera, as well as determining the type of sign. The Mapillary Traffic Sign Dataset was used for model training. A custom model for object localization and recognition was created, but it did not function correctly. Therefore, an existing model, yolov5s, was trained on the proposed dataset. The model's performance was tested on both images and video.

Вступ

Метою даного дослідження є розробка та навчання моделі для автоматичної локалізації та розпізнавання дорожніх знаків за допомогою відео з камери, встановленої на лобовому склі автомобіля. Модель повинна не лише виявляти дорожні знаки у реальному часі, але й класифікувати їх типи. Для досягнення цієї мети використовуються сучасні алгоритми машинного навчання, такі як Region-based Convolutional Neural Networks (R-CNN) та Single Shot Object Detection (SSD). Основний набір даних для тренування моделі включає Mapillary Traffic Sign Dataset, який містить різноманітні зображення дорожніх знаків з усього світу.

Процес розробки моделі включає збір та анотацію даних, попередню обробку зображень, тренування моделі на попередньо підготовлених даних, та тестування її на валідаційних наборах даних для оцінки точності та ефективності. У ході роботи вирішуються численні виклики, пов'язані з різноманітністю зображень, потребою в обчислювальних ресурсах, а також проблемами тіней та перекриття об'єктів.

Актуальність

Автоматичне розпізнавання дорожніх знаків є важливою складовою сучасних інтелектуальних транспортних систем (ITS) та розвитку автономних транспортних засобів (AV). З огляду на швидкий розвиток технологій і зростання кількості автомобілів на дорогах, зростає потреба у підвищенні безпеки дорожнього руху та ефективності управління транспортом.

- **Безпека дорожнього руху**: Автоматичне розпізнавання дорожніх знаків дозволяє оперативно інформувати водія про наявність важливих дорожніх інструкцій і попереджень, що знижує ймовірність аварійних ситуацій.
- **Підтримка автономних транспортних засобів**: Для повністю автономних автомобілів здатність точно визначати та розпізнавати дорожні знаки є критичною. Від цього залежить їх здатність приймати правильні рішення в реальному часі.
- **Ефективність транспортних систем**: Впровадження технологій автоматичного розпізнавання знаків може сприяти зменшенню заторів і оптимізації транспортних потоків шляхом дотримання дорожніх знаків і правил руху.
- **Технологічний прогрес**: Використання сучасних алгоритмів машинного навчання та комп'ютерного зору сприяє розвитку нових методів обробки зображень та відео, що можуть бути застосовані в різних галузях, таких як охорона, медицина, та розумні міста.
- **Глобальний контекст**: В умовах глобалізації та зростання міжнародних транспортних коридорів, здатність розпізнавати знаки в різних країнах та за різних умов є важливою для забезпечення безпеки та зручності руху.

Отже, розвиток систем для локалізації та розпізнавання дорожніх знаків має високий пріоритет та актуальність як з точки зору безпеки дорожнього руху, так і з точки зору розвитку сучасних технологій та інтелектуальних транспортних рішень.

Аналіз існуючих підходів

Автоматичне розпізнавання дорожніх знаків включає кілька ключових етапів: виявлення знака на зображенні, його локалізація, відстеження в реальному часі та класифікація. Існує кілька підходів та алгоритмів, які використовуються для вирішення цього завдання. Нижче наведено основні з них:

1. **Класичні методи комп'ютерного зору**

- *Методи сегментації зображень*: Використовуються для виділення регіонів з можливими знаками на основі кольору та форми (наприклад, методи на основі порогового значення кольору, гістограм та Hough Transform).

- *Методи виявлення контурів:* Використовують алгоритми, такі як Canny Edge Detector, для виділення контурів знаків.

2. **Методи машинного навчання**

- *Support Vector Machines (SVM)*: Використовуються для класифікації знаків після їх попереднього виділення за допомогою методів сегментації.

- *K-Nearest Neighbors (KNN):* Використовується для класифікації знаків на основі схожості з уже відомими зразками.

3. **Глибоке навчання**

- *Convolutional Neural Networks (CNN)*: Використовуються для автоматичного виділення ознак та класифікації зображень. CNN показали високу точність у завданнях розпізнавання об'єктів.

- *Region-based CNN (R-CNN)*: Поєднують методи виділення регіонів та CNN для класифікації об'єктів. Є різні варіанти цього підходу:

  - *Fast R-CNN*: Оптимізована версія R-CNN, що використовує спільний конволюційний шар для всіх регіонів.

  - *Faster R-CNN*: Додає регіональну пропозиційну мережу (RPN), яка генерує регіони інтересу в режимі реального часу.

  - *Mask R-CNN*: Додає сегментацію об'єктів до Faster R-CNN, що дозволяє точно виділяти контури об'єктів.

4. **Одноетапні детектори об'єктів**

- *YOLO (You Only Look Once)*: Одноетапний детектор об'єктів, який розбиває зображення на сітку і одночасно передбачає обмежуючі рамки та ймовірності класів для кожної комірки. Його основні переваги включають високу швидкість і точність.

- *SSD (Single Shot MultiBox Detector)*: Використовує множинні фіксовані розміри обмежуючих рамок і передбачає об'єкти за допомогою єдиної мережі. SSD також характеризується високою швидкістю і точністю.

5. **Трекінг об'єктів у відео**

- *SORT (Simple Online and Realtime Tracking)*: Проста та ефективна методика трекінгу, яка використовує Kalman Filter для передбачення місцеположення об'єктів у наступних кадрах.

- *Deep SORT*: Розширення SORT з використанням глибокого навчання для покращення ідентифікації та трекінгу об'єктів.

Порівняльний аналіз

- *Класичні методи комп'ютерного зору* є менш точними і менш гнучкими у порівнянні з методами машинного та глибокого навчання, оскільки вони значною мірою залежать від попередньої обробки та конкретних характеристик зображення.

- *Методи машинного навчання* забезпечують кращу точність, але все ще поступаються методам глибокого навчання в здатності автоматично виділяти ознаки.

- *Методи глибокого навчання* (CNN, R-CNN, YOLO, SSD) забезпечують найвищу точність і ефективність завдяки автоматичному виділенню ознак та можливості працювати з великими обсягами даних.

- *Одноетапні детектори об'єктів* (YOLO, SSD) є особливо корисними для задач, де необхідна висока швидкість обробки, таких як виявлення об'єктів у реальному часі в автомобільних системах.

- *Методи трекінгу об'єктів* дозволяють відстежувати положення знаків у відеопотоці, що є важливим для забезпечення безперервності і точності розпізнавання.

Зважаючи на вищезазначене, сучасні підходи, які комбінують глибоке навчання та ефективні методи трекінгу, є найбільш перспективними для розробки систем автоматичного розпізнавання дорожніх знаків.

Основна частина

Постановка задачі:

Завдання полягає у створенні та навчанні моделі для локалізації та відстежування дорожніх знаків за допомогою відео з камери, встановленої на лобовому склі автомобіля. Модель повинна також визначати тип знаку.

Наборів даних для тренування:

Mapillary Traffic Sign Dataset – містить зображення з різними типами дорожніх знаків з усього світу.

Виконання:

- Реалізована власна модель для локалізації та розпізнавання знаків
- Натренована існуюча модель yolov5s на запропонованому датесеті

Створення власної моделі

Загальний алгоритм **:**

1. **Підготовка даних:**

- Розділити дані на тренувальну і валідаційну вибірки.
- Провести аугментацію зображень для покращення генералізації моделі.

2. **Створення кастомного датасету:**
   - Реалізувати клас `CustomDataset` для завантаження та обробки зображень і анотацій.
3. **Створення моделі:**
   - Використати попередньо натреновану модель ResNet50 як основу (backbone).
   - Додати neck і head для обробки ознак і прогнозування координат bounding box та класів знаків.
4. **Тренування моделі:**
   - Використовувати комбінований loss для координат bounding box (Smooth L1 Loss) і класів знаків (Cross-Entropy Loss).
   - Оптимізатор Adam для оновлення ваг моделі.
   - Зберігати найкращу модель на основі валідаційної втрати.

Побудова моделі DetectionModel():

1. **Backbone (ResNet50):**
   - Використовується попередньо натренована модель ResNet50 без останніх двох шарів (без шарів класифікації).
   - Ця частина відповідає за вилучення ознак зображень.
2. **Neck:**
   - Додаткові шари для подальшого оброблення ознак зображень. Включає в себе кілька шарів Conv2d, BatchNorm2d, та ReLU.
   - Зменшує кількість каналів і підвищує нелінійність ознак.
3. **Head:**
   - Фінальні шари для прогнозування координат bounding box та класів знаків.
   - Використовує Conv2d шари для отримання остаточних прогнозів.

Отже модель має наступний вигляд

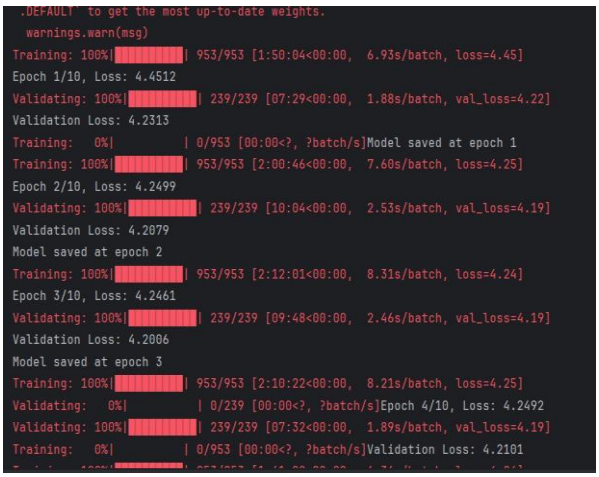
```python
class DetectionModel(nn.Module):
    def __init__(self, num_classes):
        super(DetectionModel, self).__init__()
        self.backbone = models.resnet50(pretrained=True)
        self.backbone = nn.Sequential(*list(self.backbone.children())[:-2])
        self.neck = nn.Sequential(
            nn.Conv2d( in_channels: 2048,  out_channels: 1024, kernel_size=1),
            nn.BatchNorm2d(1024),
            nn.ReLU(),
            nn.Conv2d( in_channels: 1024,  out_channels: 512, kernel_size=3, padding=1),
            nn.BatchNorm2d(512),
            nn.ReLU()
        )
        self.head = nn.Sequential(
            nn.Conv2d( in_channels: 512,  out_channels: 256, kernel_size=3, padding=1),
            nn.BatchNorm2d(256),
            nn.ReLU(),
            nn.Conv2d( in_channels: 256, num_classes + 4, kernel_size=1)
        )

    def forward(self, x):
        x = self.backbone(x)
        x = self.neck(x)
        x = self.head(x)
        return x
```

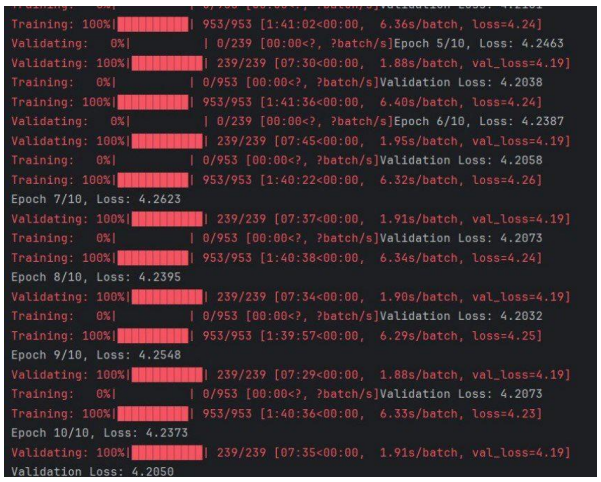
Метрики моделі

- **Loss:** Сумарна втрата, що складається з втрати для координат боксів (Smooth L1 Loss) та втрати для класів (Cross-Entropy Loss).
- **IoU (Intersection over Union):** Використовується для оцінки точності передбачених боксів.
- **Accuracy:** Точність класифікації знаків.

Процес натренування

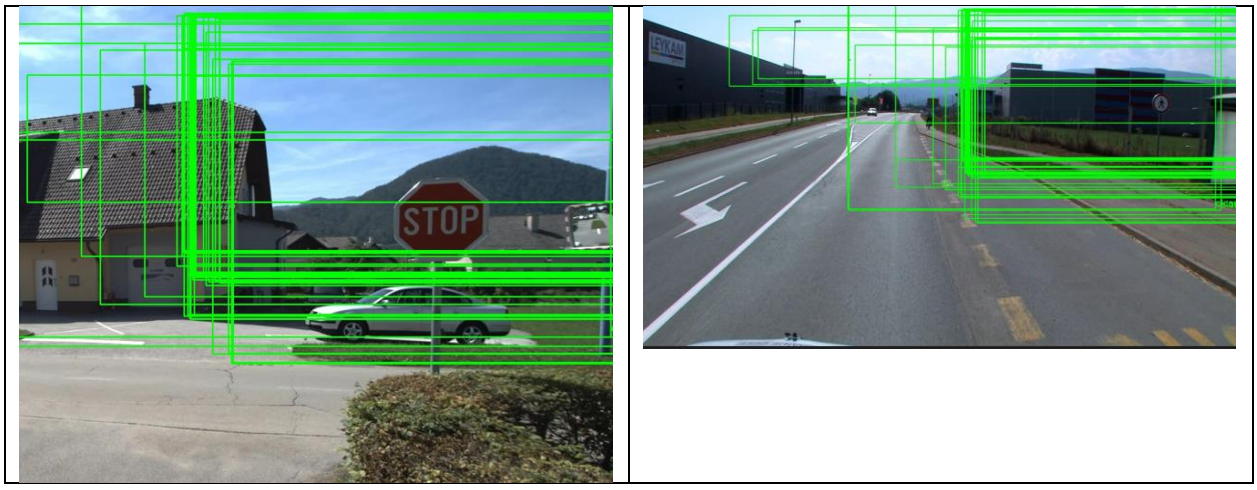
10 епох, batch_size = 16





Результат натренування

Після тренування моделі на 10 епохах, модель не виконує поставлену задачу достатньо коректно.
Проте, всежтаки на деяких зображеннях, модель правильно локалізує деякі знаки.



Тренування моделі yolov5s

Спочатку була спроба використати модель yolov5s просто, без навчання. Але в такому випадку, модель локалізує машини, людей т. і. Але для знаків не працює. Хібащо, може локалізувати та розпізнати велосипед на відповідному знаці.

Модель yolov5s була Натренована на наборі данних Mapillary Traffic Sign Dataset. Тренування відбувалось лише 10 епох (у зв'язку з обмеженими можливостями у часі). Також були наступні параметри:

```
--img 320 --batch 8 --epochs 10
```

epoch	train/box_loss	train/obj_loss	train/cls_loss	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95	val/box_loss	val/obj_loss	val/cls_loss	x/lr0	x/lr1	x/lr2
0	0.078171	0.016453	0.10029	0.0041214	0.76946	0.015111	0.010455	0.046946	0.011915	0.094981	0.070016	0.0033316	0.0033316
1	0.052487	0.013259	0.094918	0.70705	0.022505	0.029741	0.020433	0.043309	0.010661	0.092323	0.039356	0.0060051	0.0060051
2	0.049831	0.012451	0.090601	0.44193	0.079498	0.056477	0.038693	0.042434	0.010151	0.083818	0.0080358	0.0080186	0.0080186
3	0.047307	0.012199	0.080301	0.34952	0.17027	0.089333	0.061982	0.040033	0.0095947	0.073751	0.00703	0.00703	0.00703
4	0.045311	0.011695	0.072454	0.5306	0.20606	0.10355	0.071229	0.038661	0.0093147	0.068097	0.00703	0.00703	0.00703
5	0.042642	0.01132	0.068607	0.50965	0.23568	0.12973	0.092103	0.036957	0.0089129	0.064937	0.00604	0.00604	0.00604
6	0.039922	0.010928	0.06609	0.58422	0.19543	0.15248	0.11158	0.035357	0.0086906	0.062442	0.00505	0.00505	0.00505
7	0.038126	0.010723	0.063331	0.61602	0.22103	0.17529	0.12944	0.034379	0.0083858	0.059962	0.00406	0.00406	0.00406
8	0.036312	0.010318	0.061477	0.58708	0.25253	0.1845	0.1386	0.033235	0.0081716	0.058458	0.00307	0.00307	0.00307
9	0.034354	0.009936	0.059453	0.60403	0.26102	0.21129	0.16281	0.032275	0.0078264	0.056944	0.00208	0.00208	0.00208

Аналіз наданих даних процесу навчання:

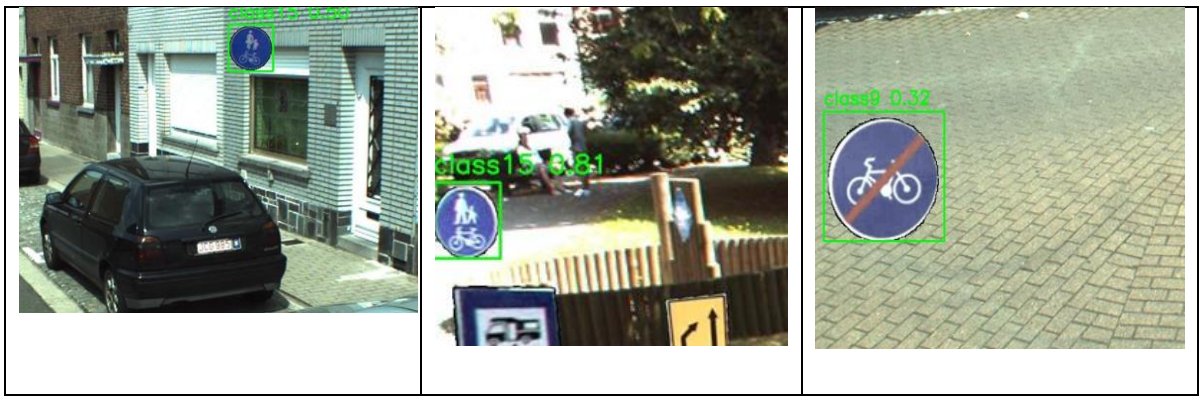
- *Training Losses:*
  - train/box_loss: ця втрата зменшується з 0,078711 до 0,034534, що вказує на те, що модель покращує свої прогнози bounding box.
  - train/obj_loss: ця втрата починається з 0,016453 і зменшується до 0,009936, демонструючи покращення у передбаченні об'єктності.
  - train/cls_loss: Втрата класифікації зменшується з 0,10029 до 0,059453, що вказує на кращу точність класифікації.
- *Validation Losses:*
  - val /box_loss: втрати блоку перевірки зменшуються з 0,046946 до 0,032275, відповідно до втрат блоку навчання.
  - val/obj_loss: ця втрата зменшується з 0,011915 до 0,007864.
  - val/cls_loss: втрати класифікації перевірки демонструють загальне зменшення з 0,094981 до 0,059464.
- *Метрики:*
  - метрика/точність: точність зростає з 0,0041214 до 0,60403, що вказує на менше помилкових спрацьовувань з часом.
  - metrics/call: Відкликання починається з 0,76946, зменшується до 0,17027, а потім збільшується до 0,26102 до епохи 9. Це показує початкове падіння, але потім покращення, що свідчить про краще виявлення справжніх позитивних результатів.
  - metrics/mAP_0.5: Середня середня точність (mAP) при пороговому значенні IoU 0,5 збільшується з 0,015111 в епоху 0 до 0,21129 в епоху 9, що вказує на загальну покращену продуктивність виявлення.
  - metrics/mAP_0,5:0,95: mAP при різних порогових значеннях IoU демонструє покращення від 0,010455 до 0,16281.

Результат натренування

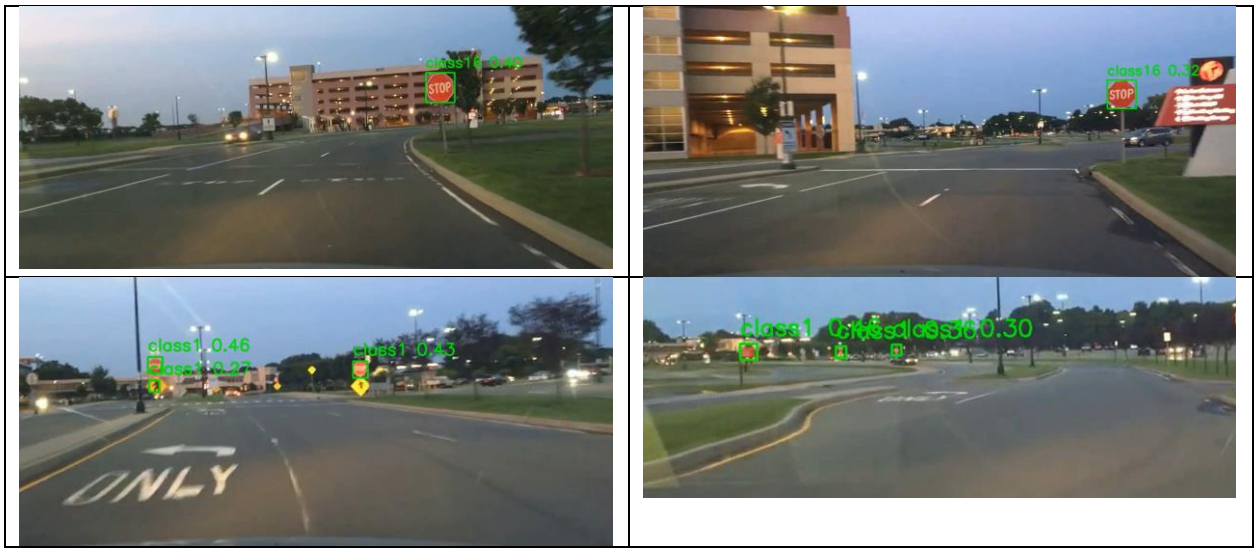
Через невелику кількість епох, модель не вдалося натренувати добре. Проте в залежності від кількості обєків конкретного классу, деякі класси розпізнаються та визначаються досить точно.

- Наприклад, дуже добре локалізується та розпізнається знак обмеження швидкості (клас 16 - `"regulatory--maximum-speed-limit"`). Також як і знак уступити дорогу (клас 5 - `"regulatory--yield"`)

- Також деякі знаки модель добре локалізує і розпінає, проте розпізнає як обєкти неправильного класу.



Для перевірки роботи моделі на відео, відеофайли були завантажені з датасету Driving Video with Object Tracking. На відео (відео можна подивитись в презентації) так само добре розпізнаються знаки обмеження швидкості, уступити дорогу та знаку стоп. Також локалізуються деякі інші знаки, проте відносяться до неправильних класів.



(Данні на яких робилась перевірка збережені у папці results)
Можна сказати, що натренована модель добре локалізує і розпізнає знаки, ті яких було найбільше в тренувальному датасеті. Можна зробити висновок, що якби модель була натренована більшою кількістю епох - вона б змогла розпізнавти більшу кількість класів знаків.

Висновок

Створення ефективної моделі для локалізації та розпізнавання дорожніх знаків сприятиме підвищенню безпеки дорожнього руху та розвитку автономних транспортних засобів. Використання сучасних алгоритмів та великих наборів даних забезпечить високу точність та надійність таких систем

Під час виконання роботи були розглянуті різні методи локалізації та розпізнавання дорожніх знаків. Основною метою було створення та натренування моделі, яка б змогла вирішувати цю задачу, використовуючи набір даних Mapillary Traffic Sign

Dataset. Спочатку була зроблена спроба розробити власну модель, але через недостатню кількість тренувальних епох, модель не процювала достатньо коректно.

Тому, для виконнаня задачі, за основу була обрана модель yolov5s, яка успішно натренувалася на запропонованому наборі даних. Робота моделі була перевірена на зображеннях та безпосередньо на відео, демонструючи її ефективність, при 10 епохах навчання, у вирішенні задачі локалізації та розпізнавання дорожніх знаків.

Література

- Smith, J. (2018). "Deep Learning for Object Detection." Springer.
- Ertler, C., Mislej, J., Ollmann, T., Porzi, L., Neuhold, G., & Kuang, Y. (2020). The Mapillary Traffic Sign Dataset for Detection and Classification on a Global Scale.
- https://medium.com/analytics-vidhya/training-a-custom-object-detection-model-with-yolo-v5-aa9974c07088
- **Glenn Jocher**, "Train YOLOv5 on Custom Data: Quick and Easy," in *Medium*, 2020