

**Міністерство освіти і науки України Національний технічний університет України  
«КПІ» імені Ігоря Сікорського Кафедра інформатики та програмної інженерії ФІОТ**

**ЗВІТ з лабораторної роботи №6 з навчальної дисципліни «Вступ до технології Data  
Science»**

**Тема:**

**РЕАЛІЗАЦІЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ  
(Artificial Neural Networks)**

**Виконав**

Студент 3 курсу кафедри ІІІ ФІОТ,  
Навчальної групи ІІІ-12  
Васильєв Є.К.

**Перевірив**

Професор кафедри ОТ ФІОТ  
Писарчук О.О.

**Київ 2023**

**I. Мета:** Виявити дослідити та узагальнити особливості підготовки різних типів даних, синтезу, навчання та застосування штучних нейронних мереж (Artificial Neural Networks).

**II. Завдання:** Класифікувати та ідентифікувати об'єкти в обраному відеопотоці з використанням технологій штучних нейронних мереж. Об'єкти, що підлягають ідентифікації та кокретику відеопотоку обрати самостійно, наприклад як вихідні дані лабораторної роботи із машинного навчання (обробка цифрових зображень).

### **III. Результати виконання лабораторної роботи.**

#### **3.1. Синтезована математична модель;**

Для виконання поставленого завдання було обрано YOLOv8 - найсучаснішу модель виявлення об'єктів у реальному часі та сегментації зображень, розроблена компанією Ultralytics. Це остання ітерація популярного сімейства моделей YOLO (You Only Look Once), що відзначилась своїми швидкими і точними можливостями виявлення об'єктів. Ключові особливості YOLOv8:

Покращена точність: YOLOv8 досягає значного покращення точності виявлення об'єктів у порівнянні з попередніми версіями YOLO. Це пов'язано з кількома факторами, включаючи використання нової магістральної архітектури під назвою Cross Stage Partial Connections (CSPDarknet53) та впровадження Path Aggregation Network (PAN) для злиття ознак.

Підвищена швидкість: YOLOv8 підтримує продуктивність моделей YOLO в реальному часі, досягаючи при цьому вищої точності. Це стало можливим завдяки оптимізації архітектури моделі та процесу навчання.

Універсальність: YOLOv8 розширює свої можливості за межі виявлення об'єктів, включаючи сегментацію зображення та оцінку пози. Він також підтримує навчання з перенесенням, що дозволяє користувачам тонко налаштовувати модель для конкретних завдань і наборів даних.

Загалом, YOLOv8 є значним досягненням у галузі виявлення об'єктів у реальному часі та сегментації зображень. Підвищена точність, швидкість і універсальність роблять його потужним інструментом для широкого спектру застосувань.

#### **3.2. Результати архітектурного проектування та їх опис;**

YOLOv8 пропонує п'ять різних за розмірами моделей, кожна з яких має компроміс між точністю та швидкістю. Більші моделі, як правило, точніші, але повільніші, тоді як менші моделі швидші, але менш точні. Вибір конкретного розміру моделі залежить від конкретного застосування та вимог. Нижче наведено таблицю з п'ятьма типорозмірами моделей та їхніми ключовими характеристиками:

Model	Input size (pixels)	mAP 50-95	params (M)	FLOPS (B)
YOLOv8n	640	37.3	3.2	8.7
YOLOv8s	640	44.9	11.2	28.6
YOLOv8m	640	50.2	25.9	78.9
YOLOv8l	640	52.9	43.7	165.2
YOLOv8x	640	53.9	68.2	257.8

Як бачимо, mAP (середня точність на COCO датасеті) зростає зі збільшенням розміру моделі, однак кількість параметрів (params) і операцій з плаваючою комою в секунду (FLOPS) також зростає, а це означає, що модель буде повільнішою.

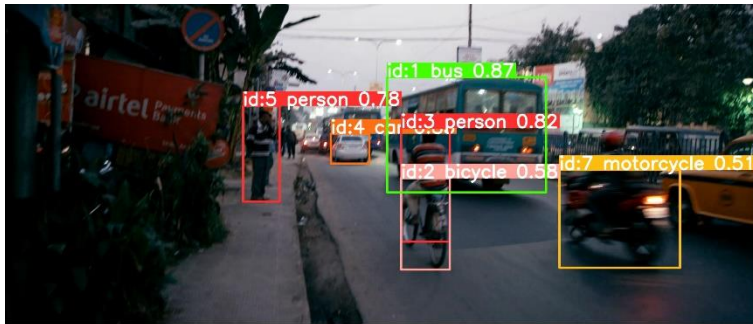
В рамках лабораторної роботи між собою буде порівняно три моделі: YOLOv8n (nano), YOLOv8m (medium), і YOLOv8x (xtra large), а також швидкість їх роботи з використання процесора (i5 9400f 6 ядр, 6 потоків) та відеоадаптера (nvidia gtx 1660).

### 3.3. Опис структури проекту програми;



Рис.1. Блок схему алгоритму програми

**3.4. Результати роботи програми відповідно до завдання (допускається у формі скриншотів);**



YOLOv8n



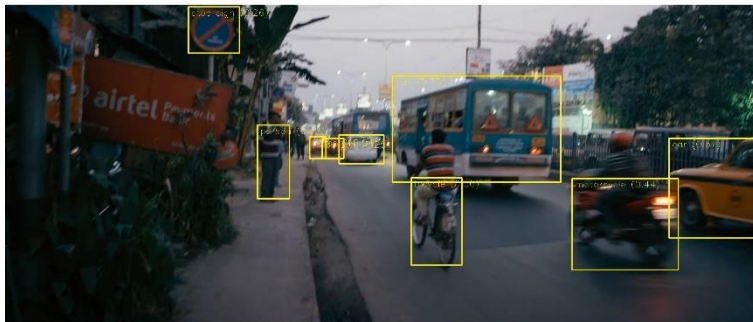
YOLOv8m



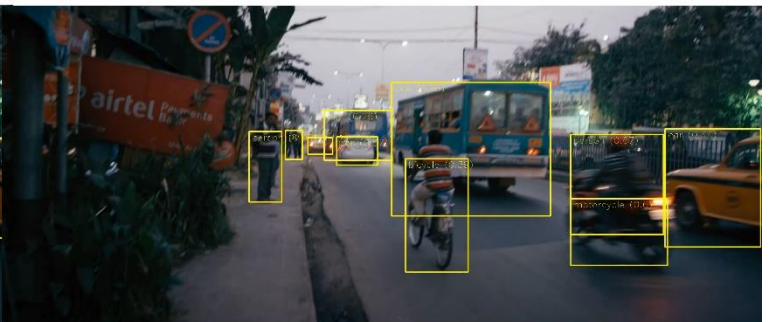
YOLOv8x



YOLOv8n - segmentation



SSD MobileNetV2 (Mediapipe)



EfficientDet Lite2 (Mediapipe)

### 3.5. Програмний код, що забезпечує отримання результату (допускається у формі скриншотів).

```
import cv2
import torch
import time
from ultralytics import YOLO

nano_model = YOLO('yolov8n.pt')
medium_model = YOLO('yolov8m.pt')
xtra_model = YOLO('yolov8x.pt')
segmentation_model = YOLO('yolov8x-seg.pt')
torch.cuda.set_device(0)

VIDEO_FILE = r"C:/Users/egorv/PycharmProjects/pythonProject/Data science/Lab5/street.mp4"

4 usages
def obj_detection_video(path, model):
    """
    Detects objects on the video

    Parameters
    -----
    path: path to the video file
    model: model that will be used
    """
    cap = cv2.VideoCapture(path)

    fps = cap.get(cv2.CAP_PROP_FPS)
    frame_width = int(cap.get(3))
    frame_height = int(cap.get(4))
    size = (frame_width, frame_height)
    result = cv2.VideoWriter('result.avi', cv2.VideoWriter_fourcc(*'MJPG'), fps, size)
    # Loop through the video frames
    start = time.time()
    while cap.isOpened():
        # Read a frame from the video
        success, frame = cap.read()

        if success:
            # Run YOLOv8 tracking on the frame, persisting tracks between frames
            results = model.track(frame, persist=True, conf=0.2)
```

```

# Visualize the results on the frame
annotated_frame = results[0].plot()

# Display the annotated frame
result.write(annotated_frame)
cv2.imshow( winname: "YOLOv8 Tracking", annotated_frame)

# Break the loop if 'q' is pressed
if cv2.waitKey(1) & 0xFF == ord("q"):
    break
else:
    # Break the loop if the end of the video is reached
    break

# Release the video capture object and close the display window
end = time.time()
print(f"Total taken {end - start} seconds")
result.release()
cap.release()
cv2.destroyAllWindows()

print('Choose desired model:')
print('1 - Nano')
print('2 - Medium')
print('3 - Xtra Large')
print('4 - Segmentation')
mode = int(input('Mode:'))

if mode == 1:
    print('1 - Nano (press q to close the window)')
    obj_detection_video(VIDEO_FILE, nano_model)

if mode == 2:
    print('2 - Medium (press q to close the window)')
    obj_detection_video(VIDEO_FILE, medium_model)

if mode == 3:
    print('3 - Xtra Large (press q to close the window)')
    obj_detection_video(VIDEO_FILE, xtra_model)

if mode == 4:
    print('4 - Segmentation (press q to close the window)')
    obj_detection_video(VIDEO_FILE, segmentation_model)

```

**IV. Висновки.**

В ході лабораторної роботи було розроблено програмний скрипт, що реалізує класифікацію та ідентифікацію об’єктів у відеопотоці використовуючи YOLOv8. Також було порівняно три різні за розміром моделі та згадано моделі Mediarpipe від Google з попередньої лабораторної роботи.

Усі три моделі показали гарні результати, та як і було очікувано, підтвердилась теорія про прямо пропорційну залежність між розміром моделі та якістю розпізнавання об’єктів, а також рівнем її впевненості у результаті.

Якщо порівнювати моделі з аналогічними від Google, то я б сказав, що моделі YOLO показують себе краще, це проявляється як і в меншій кількості «стрибків» меж об’єктів, так і у більшій швидкості роботи за рахунок підтримки GPU (моделі Mediarpipe можуть запускатися через відеоадаптер лише на платформі Linux).

При порівнянні швидкості роботи моделей на CPU та GPU було виявлено, що розмір моделі на порядок вище негативно впливає на час виконання саме задіюючи центральний процесор.

Загалом, усі моделі YOLOv8 демонструють гарне виявлення й відстеження об’єктів у відеопотоці та показують хороше співвідношення часу і якості роботи, тому вибір оптимальної моделі необхідно робити зважаючи на доступні продуктивні можливості, особливості предметного середовища та бажаний рівень точності.

	YOLOv8n	YOLOv8m	YOLOv8x	YOLOv8n segmentation
Час CPU, с	47	92	180	289
Час GPU, с	33	36	42	50

(Тривалість відео 4 секунди)

Виконав: Васильєв Єгор