



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Комп'ютерний практикум №2
з дисципліни
«Штучний інтелект в задачах обробки зображень»
на тему
«Прогнозування та розпізнавання об'єктів на фото та відео»

Виконала:
студентка гр. IA–34
Потейчук С. А.

Перевірив:
асистент Тюляков Д. І.

Мета: Навчитися працювати з бібліотекою ImageAI

Програма роботи

1. Завантаження моделі та детекція на зображенні
2. Обробка відео
3. Відстеження об'єктів
4. Результати роботи

Теоретичні відомості

Розпізнавання об'єктів (Object Detection) — це напрям комп'ютерного зору, що поєднує дві задачі: класифікацію (визначення типу об'єкта) та локалізацію (визначення його координат на зображенні). На відміну від простої класифікації, детекція дозволяє знаходити декілька об'єктів одночасно та визначати їх положення за допомогою обмежувальних рамок (bounding boxes).

Однією з найпоширеніших сучасних архітектур є YOLO (You Only Look Once). Вона належить до одностадійних (one-stage) детекторів, що виконують передбачення координат і класів об'єктів за один прохід нейронної мережі. Це забезпечує високу швидкість роботи та можливість використання в режимі реального часу.

YOLOv8, розроблена компанією Ultralytics, реалізована на базі бібліотеки PyTorch і підтримує як детекцію об'єктів на зображеннях, так і їх відстеження (tracking) на відео.

Для відстеження використовується алгоритм багатооб'єктного трекінгу (Multi-Object Tracking), який дозволяє присвоювати кожному об'єкту унікальний ідентифікатор та зберігати його між кадрами відео. Це дає змогу аналізувати рух об'єктів та уникати повторного підрахунку.

Виконання роботи

У програмі реалізовано три основні функції: детекція на зображенні, детекція на відео та відстеження об'єктів.

1. Завантаження моделі та детекція на зображенні

Тут відбувається завантаження попередньо навченої моделі YOLOv8. Вона використовується для передбачення класів об'єктів та їх координат.

```
33 def detect_obj(source, conf=0.6, scale=1):
34     model = YOLO("models/yolov8m.pt")
35     results = model(source, conf=conf)
36
37     annotated_frame = results[0].plot()
38     annotated_frame = resize_frame(annotated_frame, scale=scale)
39
40     counts = {}
41     for box in results[0].boxes:
42         cls_id = int(box.cls[0])
43         label = model.names[cls_id]
44         counts[label] = counts.get(label, 0) + 1
45
46     if isinstance(source, str):
47         basename = os.path.splitext(os.path.basename(source))[0]
48         # output_path = f"output/{source[6:-4]}_out.jpg"
49         output_path = f"output/{basename}_{conf}_out.jpg"
50         cv.imwrite(output_path, annotated_frame)
51
52     return annotated_frame, counts
```

Модель аналізує зображення та повертає список знайдених об'єктів. Параметр conf задає мінімальний поріг довіри (confidence threshold). Отримані об'єкти містять координати рамки, клас об'єкта та ймовірність.

Підрахунок кількості об'єктів виконується шляхом проходження по results[0].boxes та групування за класами.

2. Обробка відео

```
55  def detect_obj_video(video_path, conf=0.6, scale=1):
56      capture = cv.VideoCapture(video_path)
57
58      width = int(capture.get(cv.CAP_PROP_FRAME_WIDTH))
59      height = int(capture.get(cv.CAP_PROP_FRAME_HEIGHT))
60      fps = capture.get(cv.CAP_PROP_FPS)
61
62      basename = os.path.splitext(os.path.basename(video_path))[0]
63      output_path = f"output/{basename}_{conf}_frame_out.mp4"
64      fourcc = cv.VideoWriter_fourcc(*"mp4v")
65
66      out = cv.VideoWriter(output_path, fourcc, fps, (width, height))
67
68      while capture.isOpened():
69          ret, frame = capture.read()
70          if not ret:
71              break
72
73          annotated_frame, counts = detect_obj(frame, conf)
74
75          out.write(annotated_frame)
76          cv.imshow("per frame", resize_frame(annotated_frame, scale))
77
78          if cv.waitKey(1) & 0xFF==ord('q'):
79              break
80
81      capture.release()
82      out.release()
```

Відео відкривається через OpenCV.

Кожен кадр зчитується в циклі та передається в модель для обробки. Після цього результати зберігаються у новий відеофайл за допомогою VideoWriter.

3. Відстеження об'єктів

```
85  def track_objects(video_path, conf=0.6, scale=1):
86      model = YOLO("models/yolov8m.pt")
87
88      capture = cv.VideoCapture(video_path)
89
90      width = int(capture.get(cv.CAP_PROP_FRAME_WIDTH))
91      height = int(capture.get(cv.CAP_PROP_FRAME_HEIGHT))
92      fps = int(capture.get(cv.CAP_PROP_FPS))
93
94      basename = os.path.splitext(os.path.basename(video_path))[0]
95      output_path = f"output/{basename}_{conf}_track_out.mp4"
96      fourcc = cv.VideoWriter_fourcc(*"mp4v")
97
98      out = cv.VideoWriter(output_path, fourcc, fps, (width, height))
99
100     while capture.isOpened():
101         ret, frame = capture.read()
102         if not ret:
103             break
104
105         results = model.track(frame, conf=conf, persist=True)
106         annotated_frame = results[0].plot()
107
108         out.write(annotated_frame)
109         cv.imshow("tracking", resize_frame(annotated_frame, scale))
110
111         if cv.waitKey(1) & 0xFF==ord('q'):
112             break
113
114         out.release()
115         capture.release()
```

Метод `track()` виконує не тільки детекцію, але й відстеження об'єктів між кадрами. Параметр `persist=True` дозволяє зберігати ідентифікатори об'єктів між кадрами. Таким чином кожному фрукту присвоюється унікальний ID, що не змінюється під час його руху у відео.

4. Результати роботи

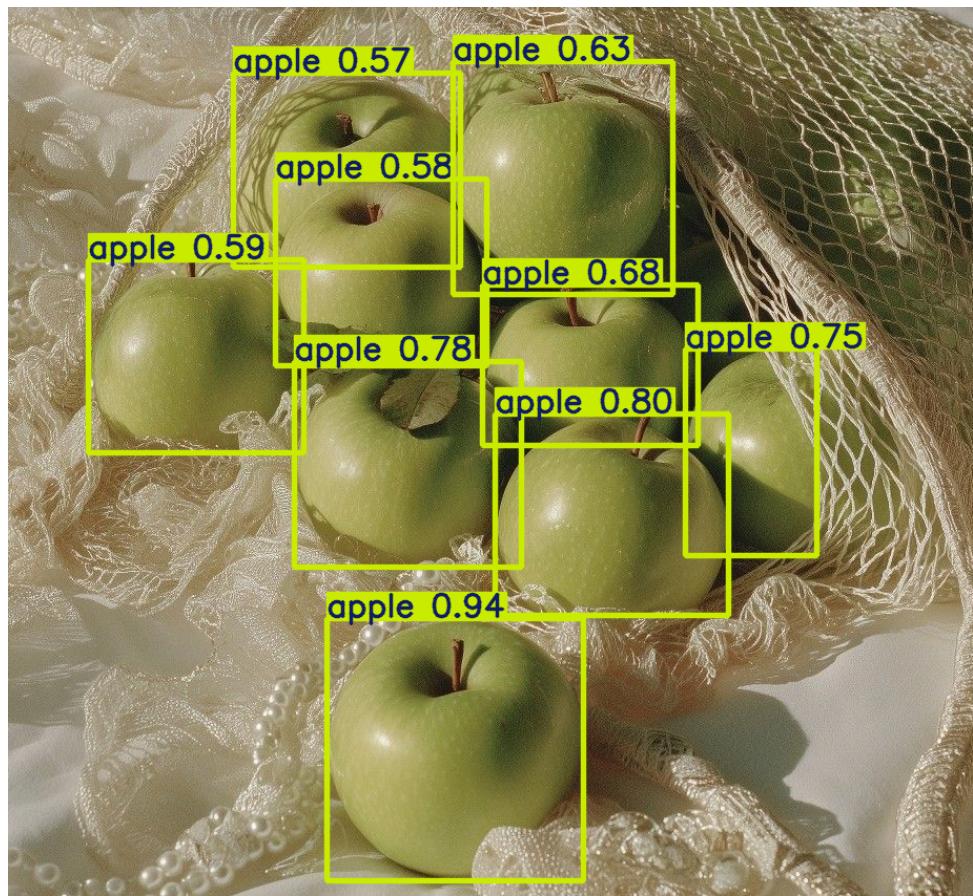


Рисунок 1 — Детекція об'єктів на зображенні

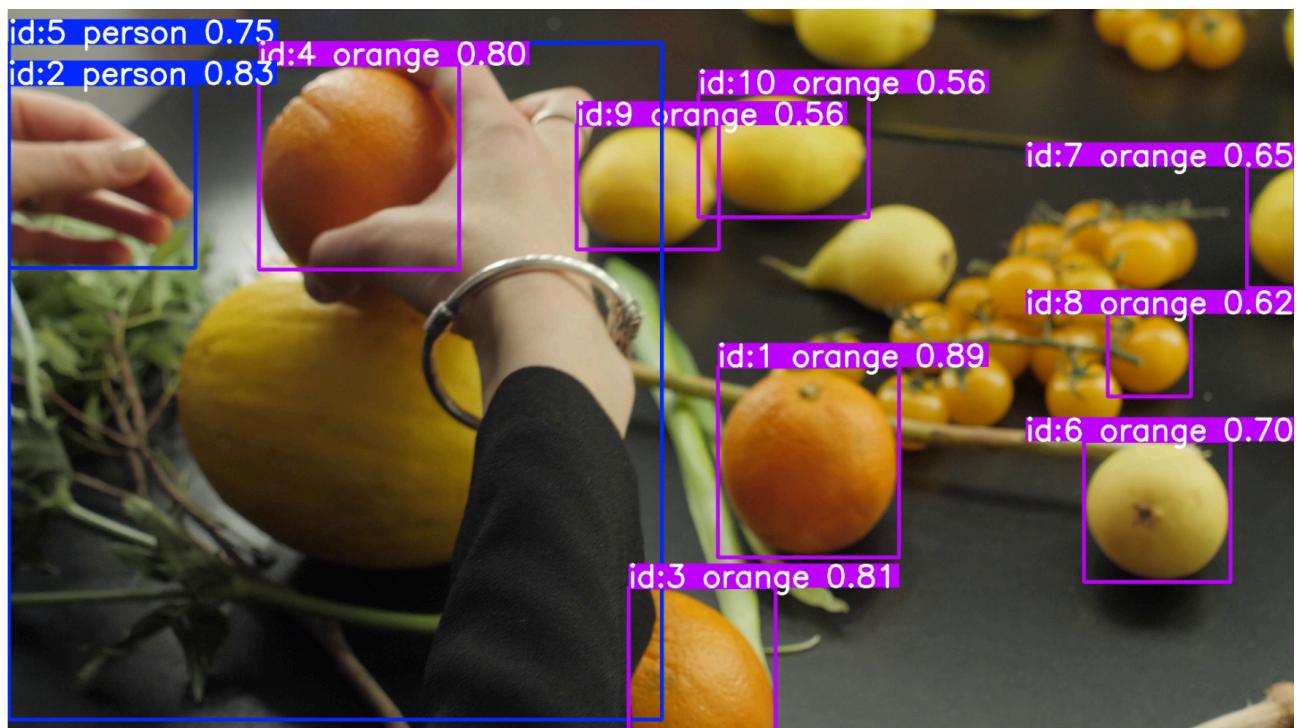


Рисунок 2 — Відстеження об'єктів на відео

Висновок

У ході виконання лабораторної роботи було досліджено сучасні методи розпізнавання та відстеження об'єктів за допомогою нейронних мереж. Було використано архітектуру YOLOv8, яка забезпечує високу швидкість та точність детекції.

Розроблена програма дозволяє автоматично знаходити фрукти на зображеннях, а також виконувати детекцію та відстеження об'єктів на відео. Завдяки використанню алгоритму багатооб'єктного трекінгу кожному об'єкту присвоюється унікальний ідентифікатор, що дозволяє аналізувати його рух у часі.