




Zadanie 04


Przygotowanie modelu YOLO do wykrywania jedzenia na obrazach

W dzisiejszym zadaniu będziesz pracować z gotową siecią neuronową YOLO, przystosowaną do znajdowania obiektów na obrazach. Jednakże YOLO domyślnie nie rozpoznaje kategorii "jedzenie", dlatego dotrenujemy model, aby nauczył się rozpoznawać ten typ obiektów.

- Dla punktów oznaczonych ikoną  podaj opis realizacji, w tym kluczowe klasy, metody lub funkcje oraz ewentualnie fragmenty kodu źródłowego.
- Dla punktów oznaczonych ikoną  załącz w raporcie obraz prezentujący efekt wykonanej operacji.
- Dla punktów oznaczonych ikoną  załącz wykres przedstawiający proces optymalizacji (uczenia).





Korzystanie z YOLO

YOLO to szybki i wydajny detektor obiektów, który łatwo wdrożyć. Istnieje wiele wersji YOLO, jednak zasady korzystania z nich są podobne. Wybierz dowolną wersję YOLO (testowane na YOLOv8). Model YOLO występuje w różnych rozmiarach; do testów początkowych możesz użyć dowolnej wielkości, natomiast do fine-tuningu zalecana jest wersja mniejsza (np. "nano"), która jest bardziej wydajna na komputerze osobistym.

1. Zainstaluj wybraną wersję i wielkość YOLO (np. `yolov8m.pt`) i przetestuj jej działanie na obrazie zawierającym psa i kota obok siebie, aby upewnić się, że wykrywa obiekty poprawnie. 

Przystosowanie YOLO do wykrywania jedzenia

Naszym celem jest wykorzystanie YOLO do detekcji jedzenia, jednak domyślny model nie rozpoznaje tej kategorii. Dlatego przeprowadzimy dodatkowe szkolenie, aby model nauczył się klasy „jedzenie”. Wytrenowany model zostanie użyty do cenzurowania potraw na obrazach.

2. Pobierz zbiór danych treningowych zawierający obrazy z etykietami. Możesz skorzystać z otwartego zbioru danych Open Images Dataset V7. Pobierz obrazy z kategorii „food” oraz przygotuj etykiety kompatybilne z YOLO. Skrypt [downloader.py](#) oraz pliki CSV (z listą obrazów i ich koordynatami) będą pomocne: [manual](#).
3. Przeanalizuj pliki CSV (np. przy użyciu biblioteki pandas), wygeneruj plik tekstowy zgodny z wymaganiami skryptu `downloader.py`, a następnie pobierz odpowiednią ilość obrazów treningowych i walidacyjnych, zapisując ich etykiety do plików tekstowych. Ile przykładów udało się pobrać? 
4. YOLO wymaga pliku YAML z informacjami o zbiorze danych do treningu. Stwórz taki plik – możesz zrobić to ręcznie lub przy użyciu biblioteki pyYAML. Opisz, jak to zrobiłeś. 
5. Przystąp do treningu. Użyj mniejszej wersji YOLO (np. `yolov8n.pt`) i dotrenuj ją przez 5–10 epok na przygotowanym zbiorze danych. Przedstaw wykres funkcji kosztu (loss). 
6. Przetestuj wytrenowany model na kilku obrazach, które nie były używane podczas treningu (np. ze zbioru testowego Open Images). 

7. Wykorzystaj wytrenowany model do cenzurowania jedzenia na obrazach. Wybierz kilka obrazów zawierających jedzenie (np. ze zbioru testowego Open Images) i zastosuj efekt rozmycia (blur) jedynie na obszarach, gdzie model wykrywa jedzenie. Opisz metodę rozmycia oraz bibliotekę, której używasz do tego zadania. Pokaż efekt końcowy na kilku przykładowych obrazach z ocenionym jedzeniem.

