

Trash Detection

Konstanty Kaszubski

Jakub Jagodziński

Lucjan Jaworski

8 lutego 2026

1 Opis rzeczywistego problemu

Temat: Automatyzacja wstępnej segregacji odpadów komunalnych w oparciu o wizyjne systemy detekcji. Implementacja systemu tego typu pozwala na zastąpienie ludzi maszynami w pracy, która nie jest zbyt atrakcyjna.

Cel (Efekt): Stworzenie modelu, który jest w stanie w czasie rzeczywistym zidentyfikować i zlokalizować odpady na obrazie wizyjnym, klasyfikując je do jednej z pięciu frakcji recyklingowych: Metal, Plastik, Szkło, Papier oraz Bio.

Motywacja: Ręczne sortowanie odpadów jest procesem nieefektywnym, kosztownym i obarczonym błędami ludzkimi oraz ryzykiem sanitarnym. Wprowadzenie automatyzacji opartej na wizji komputerowej pozwala na zwiększenie przepustowości linii sortowniczych oraz poprawę czystości odzyskanego surowca, co jest kluczowe w obliczu rosnących wymogów prawnych UE dotyczących poziomów recyklingu. Jest to realizacja konkretnego, rzeczywistego problemu optymalizacji procesu przemysłowego.

Dane wejściowe: Obrazy cyfrowe (klatki wideo) przedstawiające odpady w różnym stopniu zniekształcenia (zgniecenia, zabrudzenia). Mierzone będą: lokalizacja obiektu (współrzędne ramki) oraz pewność predykcji klasy.

Powiązanie z AI: Projekt dotyczy zagadnień Computer Vision (widzenie komputerowe), a w szczególności zadania Object Detection. Model uczy się rozpoznawać cechy wizualne odpadów na podstawie dostarczonego zbioru treningowego (Supervised Learning).

2 State of the art (Przegląd znanych koncepcji)

W literaturze przedmiotu wyróżnia się trzy główne podejścia do problemu detekcji obiektów, różniące się architekturą i balansem między szybkością a precyzją:

2.1 Metody klasyczne (Sliding Window + HOG + SVM):

- Opis: Podejście bazujące na przesuwaniu okna po obrazie, ekstrakcji cech za pomocą histogramu zorientowanych gradientów (HOG) i klasyfikacji maszyną wektorów nośnych (SVM).

- Mocne strony: Działa na słabszym sprzęcie, nie wymaga GPU, wytłumaczalność matematyczna.
- Słabe strony: Bardzo niska skuteczność przy obiektach zdeformowanych (zgniecone puszki), brak odporności na zmiany oświetlenia, powolne działanie przy dużej rozdzielczości.

2.2 Detektory dwustadialne (Two-stage detectors, np. Faster R-CNN):

- Opis: Sieci neuronowe, które najpierw generują propozycje regionów (Region Proposal Network), a dopiero potem klasyfikują zawartość każdego regionu.
- Mocne strony: Bardzo wysoka precyza (mAP) i dokładność w lokalizacji małych obiektów.
- Słabe strony: Wysoka złożoność obliczeniowa i niska prędkość (często poniżej 5-10 FPS), co dyskwalifikuje je w zastosowaniach na szybkich taśmach sortowniczych.

2.3 Detektory jednostadialne (One-stage detectors, np. rodzina YOLO - You Only Look Once):

- Opis: Traktują detekcję jako problem regresji, przewidując ramki i prawdopodobieństwa klas dla całego obrazu w jednym przejściu sieci przez koder i dekoder.
- Mocne strony: Praca w czasie rzeczywistym (Real-time), łatwa implementacja na urządzeniach brzegowych (np. Jetson Nano), dobry balans między szybkością a precyza.
- Słabe strony: Historycznie gorsze radzenie sobie z małymi obiektami w grupach (choć nowsze wersje jak v8/v11 niwelują ten problem).

3 Opis wybranej koncepcji

Jako rozwiązanie wybrano architekturę YOLOv8 (You Only Look Once version 8), należącą do grupy detektorów jednostadialnych (single-stage detectors). Model ten wybrano ze względu na kompromis między szybkością działania (niedobraną na taśmie produkcyjnej) a precyza detekcji.

3.1 Architektura sieci neuronowej

W przeciwieństwie do starszych wersji opartych na mechanizmie kotwic (anchors), YOLOv8 wykorzystuje podejście Anchor-Free. Sieć przewiduje bezpośrednio środek obiektu, co zwiększa elastyczność przy wykrywaniu odpadów o nieregularnych kształtach (np. silnie zgniecone puszki). Model składa się z trzech kluczowych modułów:

- Backbone (CSPDarknet53): Odpowiada za ekstrakcję cech z obrazu (faktura, krawędzie) przy użyciu splotów.
- Neck (PANet): Łączy cechy z różnych poziomów, co pozwala na jednoczesne wykrywanie obiektów małych (nakrętki) i dużych (kartony).
- Head (Decoupled): Rozdziela proces decyzyjny na dwie niezależne ścieżki: klasyfikację (co to jest?) i regresję ramki (gdzie to jest?), co zwiększa precyza (mAP).

3.2 Dane i Inżynieria Cech

Do wytrenowania modelu wykorzystano publicznie dostępny zbiór ProjectVerba (Roboflow Universe).

- **Preprocessing:** Oryginalny zbiór zawierał 42 klasy o skrajnym niebalansowaniu (np. dominacja aluminium nad elektroniką). W ramach projektu przeprowadzono agregację klas (remapping) do 5 głównych frakcji sortowniczych (Metal, Plastik, Szkło, Papier, Bio), usuwając klasy niereprezentatywne (<50 próbek), co jest kluczowe dla stabilności treningu.
- **Dostępność:** Dane są publiczne, jednak wymagały autorskiego skryptu czyszczącego i mapującego etykiety.

3.3 Wyjście algorytmu i metoda działania

Algorytm przetwarza obraz w siatce (grid) i zwraca listę wykrytych obiektów, gdzie każdy opisany jest wektorem:

$$[x_{center}, y_{center}, width, height, class_id, confidence]$$

Wizualnie jest to ramka otaczająca odpad z etykietą frakcji. Wybrano wariant modelu Nano (yolov8n) ze względu na priorytet szybkości inferencji.

3.4 Wymagania wdrożeniowe i procedura testowa

Do wdrożenia systemu w warunkach rzeczywistych (Rzeczywisty świat) wymagana jest:

- Kamera przemysłowa nad taśmociągiem.
- Oświetlenie eliminujące cienie (np. pierścieniowe lub backlight dla plastików).
- Jednostka obliczeniowa z akceleratorem GPU (np. NVIDIA Jetson) do inferencji w czasie rzeczywistym.

3.5 Procedura testowania i problemy

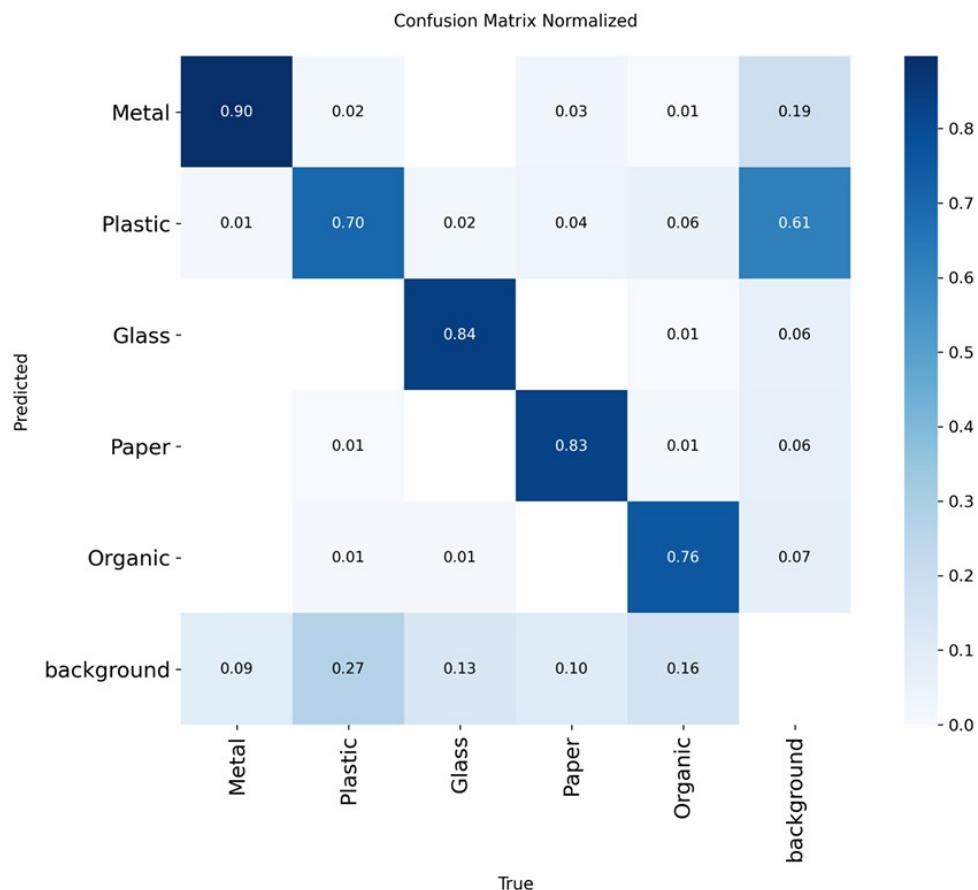
1. **Podział zbioru na:** Treningowy (70%), Walidacyjny (20%), Testowy (10%).
2. **Główna metryka:** mAP@0.5 (Mean Average Precision przy progu IoU 0.5) oraz Confusion Matrix (Macierz Pomyłek) do weryfikacji, czy model nie myli np. szkła z plastikiem.
3. **Zidentyfikowane problemy:** Deformacja obiektów (zgniecionie butelki wyglądają inaczej niż całe), zabrudzenia etykiet, nachodzenie na siebie obiektów (okluzja).

4 Wyniki eksperymentu i demonstracja działania

4.1 Metryki uczenia

Proces uczenia przeprowadzono na 25 epokach. Finalna skuteczność modelu (mAP@0.5) dla wszystkich klas wyniosła **0.80**. Szczegółowa analiza skuteczności per klasa wykazała istotne różnice w detekcji materiałów:

- Papier: 0.895 (Najwyższa skuteczność – obiekty o wyraźnej teksturze).
- Metal: 0.865
- Szkło: 0.808
- Bio: 0.793
- Plastik: 0.639 (Najniższa skuteczność).



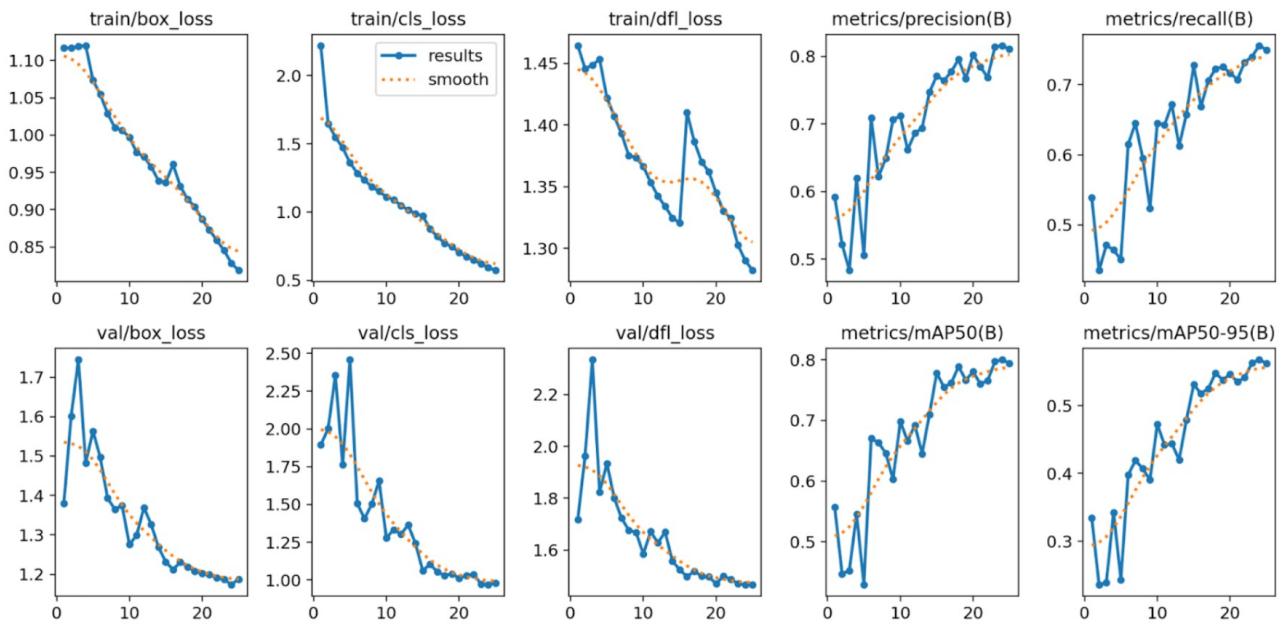
Rysunek 1: Macierz pomylek (Confusion Matrix).

4.2 Wnioski z analizy błędów

Niższy wynik dla klasy Plastic wynika z fizycznej charakterystyki odpadów – przezroczystości butelek oraz dużej wariancji kształtów (zgniecenia). Jest to typowe wyzwanie w wizyjnych systemach sortowania, które w warunkach przemysłowych rozwiązuje się poprzez zastosowanie czujników NIR (podczerwieni), a nie tylko kamery RGB.

4.3 Przebieg treningu

Poniższe wykresy prezentują spadek funkcji straty (Loss) oraz wzrost precyzji w trakcie 25 epok treningowych.



Rysunek 2: Przebieg funkcji straty (Loss) i precyzji (mAP) w trakcie 25 epok

4.4 Opis Demo

Przygotowano skrypt w języku Python wykorzystujący bibliotekę ultralytics. System pobiera obraz wejściowy, przeprowadza inferencję i nanosi ramki detekcji (Bounding Boxes) wraz z etykietą klasy i poziomem pewności.



Rysunek 3: Przykładowa detekcja obiektu klasy 'Plastic' .