Neodata — Segmentation & Training Pipeline (YOLO + SAM3)

Demo site!

Sažetak

Naša ideja je da ćemo koristiti prvo YOLO za detekciju fasadnih cropova i SAM3 (u Colab GPU okruženju) za pixel-level segmentaciju komponenti (tin, screw, hole, glass, seal). Iz maski se izračunava pokrivenost i prostorna distribucija te se primjenjuju heuristike (thresholdi za coverage, veličina/oblik maski, zonalna analiza) koje odlučuju o pojavi i confidenceu defekta. Zbog ograničenja lokalnog hardvera, segmentirani izlazi su ručno proizvedeni u Colabu i spremljeni u `segmented_output_improved/` te se koriste za demonstraciju i evaluaciju!

Na kraju radimo jednostavno web sučelje gdje radnik može laganim načinom detektirati anomalije.

Kod korištenja stranice je važno napomenuti da se backend svako malo zbog neaktivnosti šalje u SLEEP. **Zato je važno pričekati pri inicijalnome testiranju, oko 5 minuta da se pokrene backend.**

Uvod

Cilj: dobiti pouzdane pixel-level maske za specifične komponente na fasadama radi automatske detekcije defekata.

Razlog Colab workflowa: lokalni HW nema dovoljno GPU/ram za SAM3 treniranje/inferenciju, zato se koristi Colab (GPU) + Drive.

Arhitektura i flow (visoki nivo)

Ulaz: originalne slike (TRAIN / positive / negative).

Korak A — YOLO (ultralytics): detekcija fasada → cropovi (facade_crops)!

Korak B — SAM3 (text-prompt driven segmentation) u Colabu: za svaki crop više promptova po komponenti → skup maski. Tražimo najbolji klasifikator tj. prompt da damo SEM3 modelu.

Korak C — Mask post-processing: morphology (opening/closing), filter po veličini, kombiniranje preklapajućih maski.

Izlaz: overlay slike, konture, komponentne croppove i *_masks.json metapodaci (spremno u segmented_output_improved/).

Upotreba u demo: backend čita te JSON-ove (bez ponovnog pokretanja modela) i radi analizu u **combined_detector.py**!

Sadržaj važnih datoteka

SAM_Improved_Segmentation.ipynb — Glavni Colab notebook (YOLO -> SAM3 -> postprocess -> save).

classify/combined_detector.py — Logika koja kombinira detekcije i računa metrike.

segmented_output_improved/ — finalni izlazi: overlay, contours, _masks.json datoteke.

Notebook — SAM_Improved_Segmentation.ipynb (detalji)

Drive mount, instalacija: pillow-heif, ultralytics, decord/av, sam3.

HuggingFace login potreban za SAM3 weights.

Glavne funkcije: detect_and_crop_facades, segment_with_text_prompt, segment_with_multiple_prompts, refine_mask_morphology, combine_overlapping_masks, save_results_json, process_folder.

Format izlaznih JSON-ova (_masks.json)

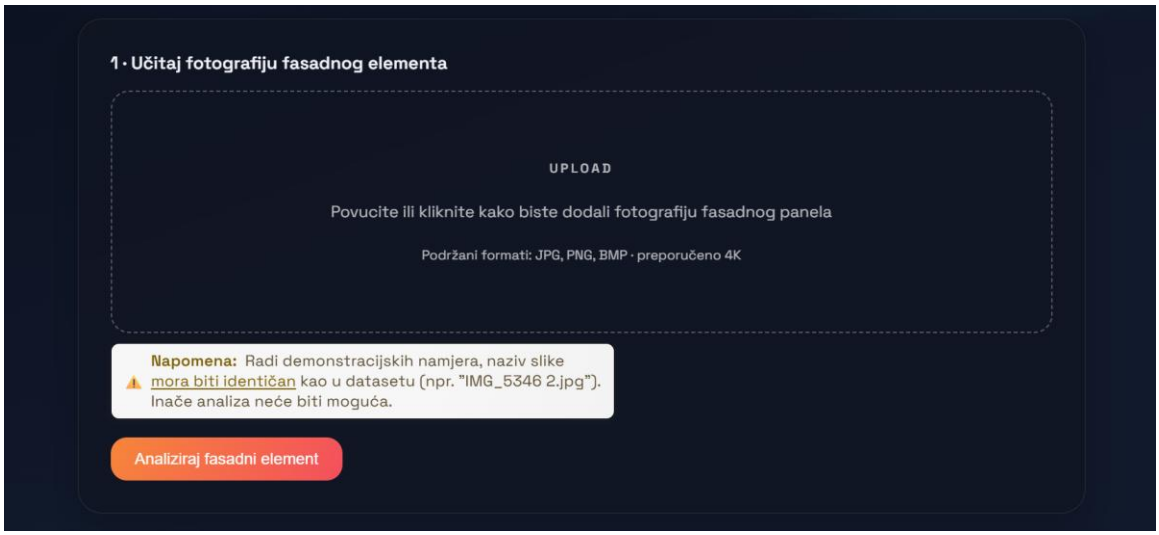
Struktura: { 'image': ..., 'size':[w,h], 'components': { 'tin': [{index, area_pixels, coverage_percent, bbox, score}, ...], ... } }

Overlay i contours slike za vizualizaciju.

Analiza i evaluacija (classify)

combined_detector.py učitava *_masks.json, računa pokrivenost komponenti, izvlači listu defekata i sprema combined_results.json.

Alati: final_report.py, check_results.py, analyze_missed.py, analyze_zones.py.



Rezultati

Generirani izvještaj se nalazi u combined_results.json!

Sljedeće je dobiveno u terminalu kada detektiramo defekte na TRAIN datasetu:

Total images: 55

Detected defects: 54

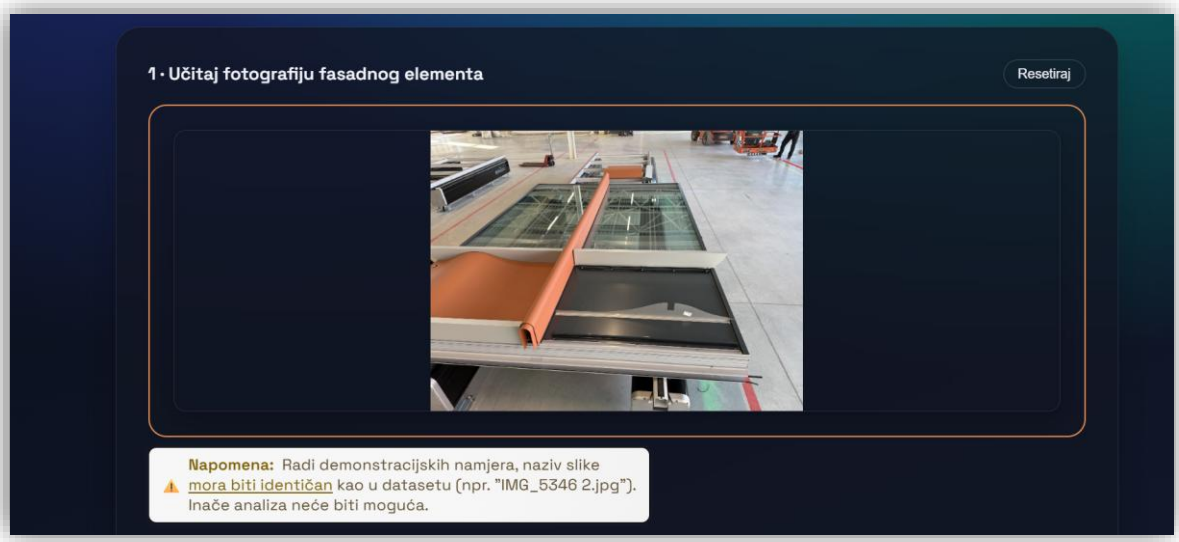
Correct matches (TP): 37

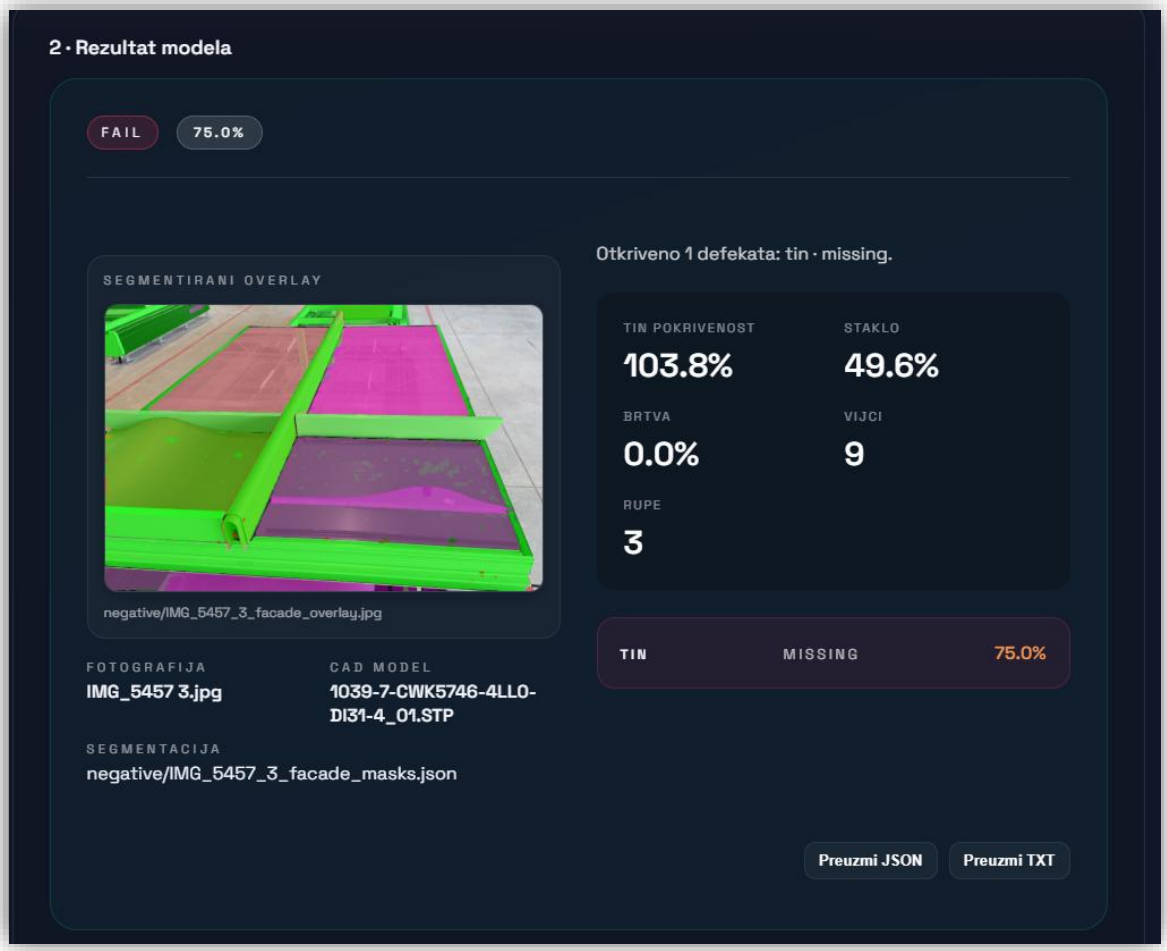
False positives (FP): 17

SCORE: 20 (TP - FP = 37 - 17)

PRECISION: 68.5%

Primjer:


1 · Učitaj fotografiju fasadnog elementa
Resetiraj
Napomena: Radi demonstracijskih namjera, naziv slike mora biti identičan kao u datasetu (npr. "IMG_5346 2.jpg"). Inače analiza neće biti moguća.


2 · Rezultat modela
FAIL
75.0%
SEGMENTIRANI OVERLAY
negative/IMG_5457_3_facade_overlay.jpg
Otkriveno 1 defekata: tin · missing.
TIN POKRIVENOST
103.8%
STAKLO
49.6%
BRTVA
0.0%
VIJCI
9
RUPE
3
TIN
MISSING
75.0%
FOTOGRAFIJA
IMG_5457 3.jpg
CAD MODEL
1039-7-CWK5746-4LL0-DI31-4_01.STP
SEGMENTACIJA
negative/IMG_5457_3_facade_masks.json
Preuzmi JSON
Preuzmi TXT

Zašto koristimo statičke outpute

Trenutno nemamo dovoljno brza računala te smo morali implementirati brzo hands-on rješenje. Brzi i jeftin backend bez GPU — učitavanje JSON umjesto modela.

Ograničenje: ne može analizirati arbitrary novu sliku bez odgovarajuće preprocess pipeline.

Reproducibilnost i preporučeno okruženje

Preporučeno: Google Colab Pro/Pro+, Python 3.10/3.11, ultralytics, pillow-heif, decord/av, opencv-python, numpy, matplotlib.

Instalacija primjer: pip install pillow pillow-heif ultralytics opencv-python numpy matplotlib decord

Pokretanje korak-po-korak (u Colab)

1) Mount Drive → postavi DRIVE_PATH, OUTPUT_DIR, YOLO_WEIGHTS.

2) Pokreni ćelije instalacije paketa.

3) Pokreni YOLO detekciju za facade crops.

4) Pokreni segmentaciju (process_folder za positive i negative).

5) Provjeri OUTPUT_DIR segmented_output_improved.

Integracija s demonstracijom

Spremi segmented_output_improved u repozitorij ili Docker image kako bi backend mogao čitati JSON-ove.

combined_detector.py koristi te JSON-ove; za arbitrary uploads treba backend endpoint koji poziva process_image nad novom slikom (zahtijeva GPU/duže vrijeme).

Ograničenja i preporuke

SAM3 zahtijeva značajne resurse; razmotriti optimizacije (tiling, niža rezolucija) ili manji model.

Bolje prompt engineering i augmentacija smanjuju ručni rad.

Sačuvati verzije paketa i model checkpoints na Drive za reproducibilnost.

Tehnički annex (naredbe)

YOLO treniranje (ultralytics):

yolo task=detect mode=train model=yolov8n.pt data=dataset.yaml epochs=50 imgsz=640

Lokalna analiza: python neodata2/combined_detector.py

Instalacija: pip install pillow pillow-heif ultralytics opencv-python numpy matplotlib

Za II. Rješenje – “Industrijsko rješenje”

Poslušajte naš pitch!!

Tim FERSADA