

TUBES UAS **PENGOLAHAN CITRA DIGITAL**



KELOMPOK 7 - 2B

ANGGOTA KELOMPOK



DAFFA AL GHIFARI
231511038



M. RAIHAN PRATAMA
231511055



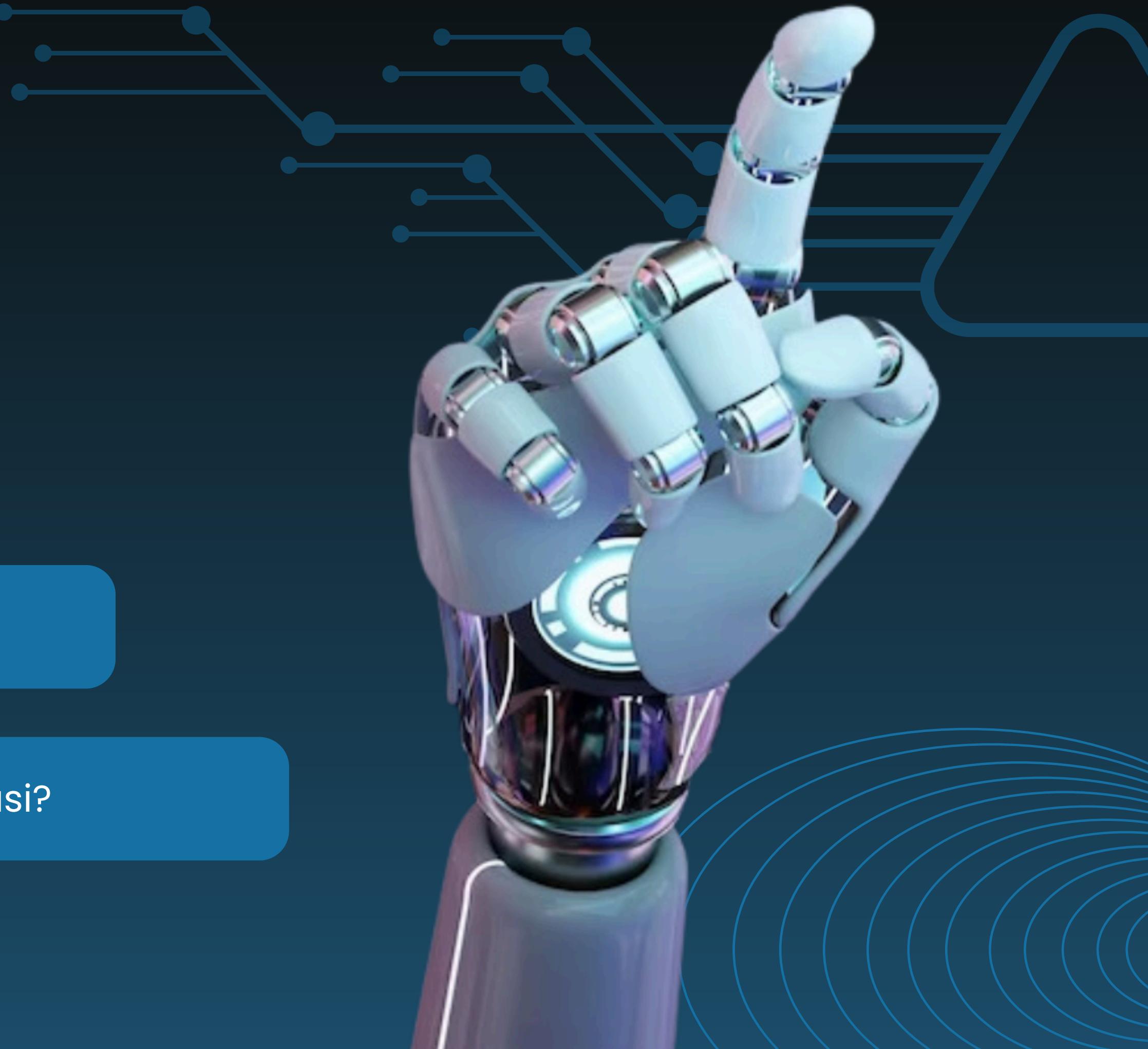
TRESNARDI FATHU R.
231511062

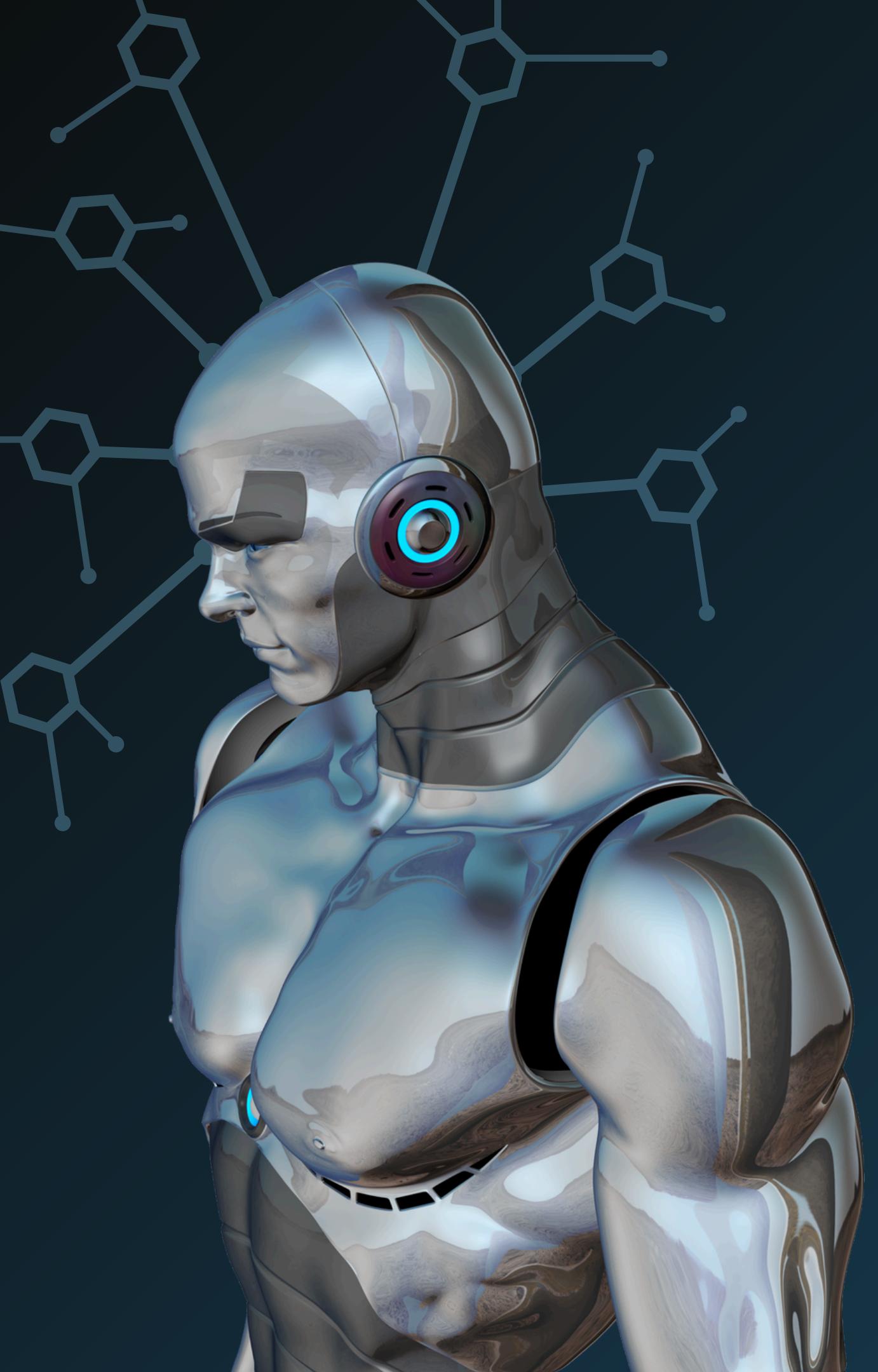
Introduction

 Apa itu Deteksi Objek Real-time?

 Tantangan Inferensi AI di Edge

 Mengapa Orange Pi 5 Menjadi Solusi?



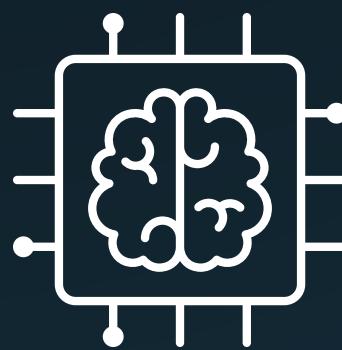


Apa itu Deteksi Objek Real-time?

Kemampuan komputer untuk mengidentifikasi dan melokalisasi objek dalam gambar atau video secara instan.

Penting dalam berbagai aplikasi: keamanan (pengawasan), otomasi (mobil otonom, robot industri), kesehatan (diagnosis medis), hingga ritel (analisis perilaku pelanggan).

Tantangan Inferensi AI di Edge



Model AI modern, khususnya untuk deteksi objek, seringkali membutuhkan daya komputasi tinggi. Dimana ini akan membatasi penerapan di perangkat edge (perangkat yang dekat dengan sumber data, bukan cloud) karena keterbatasan sumber daya.



Mengapa Orange Pi 5 Menjadi Solusi?



SBD & NPU

Munculnya Single Board Computers (SBC) dengan Neural Processing Unit (NPU) dedicated menawarkan harapan baru.

[Read More](#)

ORANGE PI 5

Orange Pi 5, dengan NPU-nya, menjanjikan komputasi AI yang signifikan dalam ukuran dan konsumsi daya yang kecil.

[Read More](#)



TUJUAN PROYEK

Mengimplementasikan dan mengevaluasi kinerja model deteksi objek YOLOv8 pada perangkat komputasi Orange Pi 5.

Mendemonstrasikan inferensi deteksi objek secara real-time pada perangkat edge Orange Pi 5.

Memanfaatkan NPU (Neural Processing Unit) yang terintegrasi pada Orange Pi 5 untuk akselerasi performa AI.

Menganalisis metrik kinerja (seperti Frame per Second/FPS dan latensi) dari model yang telah dikonversi, dan menyajikan visualisasi hasil deteksi yang efektif dan mudah dipahami.

PENGENALAN YOLOv8

You Only Look Once (Versi 8)



Konsep Dasar YOLO

Sebuah arsitektur deep learning yang revolusioner untuk deteksi objek.

Berbeda dari metode tradisional, YOLO memprediksi bounding box dan probabilitas kelas objek dalam satu kali forward pass melalui jaringan. Inilah alasan di balik namanya: "You Only Look Once."

KEUNGGULAN UTAMA

Kecepatan: Sangat cepat, ideal untuk aplikasi real-time.

Akurasi: Mampu mencapai akurasi yang kompetitif dengan metode lain.

Efisiensi: Menggunakan lebih sedikit komputasi dibandingkan arsitektur dua tahap.

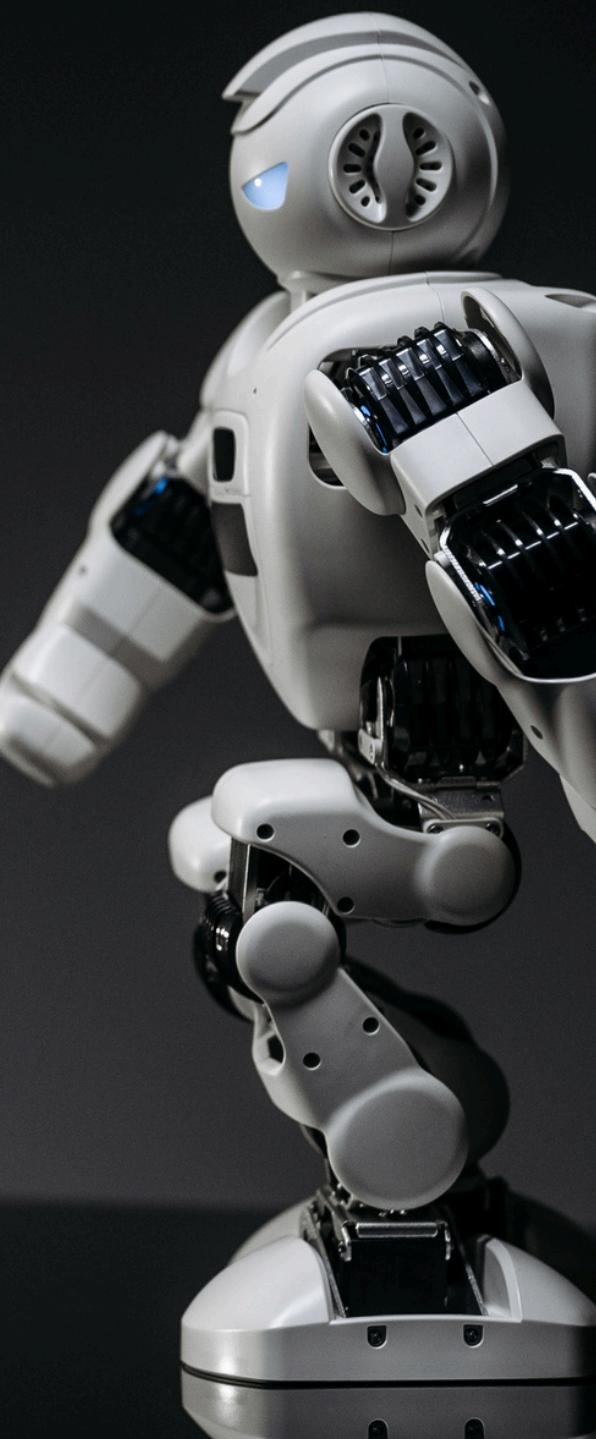


MENGAPA YOLOv8?

YOLOv8 adalah iterasi dari keluarga YOLO yang dikembangkan oleh Ultralytics.

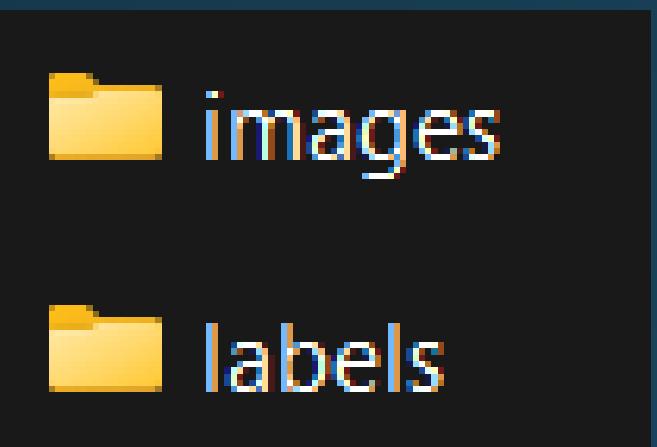
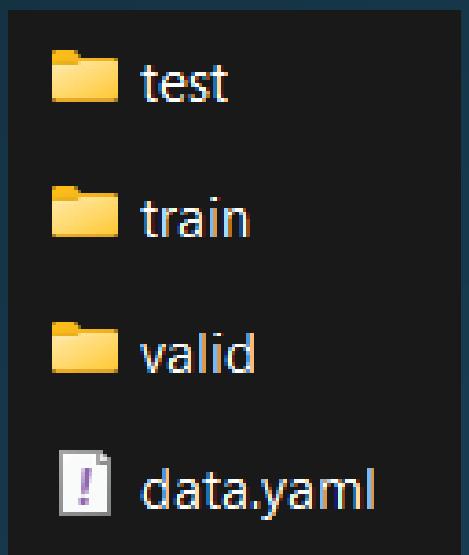
Menawarkan peningkatan dalam kecepatan inferensi dan akurasi dibandingkan versi sebelumnya.

Dukungan yang lebih baik untuk ekspor ke berbagai format inferensi, termasuk yang kompatibel dengan NPU.



TRAINING MODEL YOLOv8

Mengapa Pelatihan Model Diperlukan? Meskipun YOLOv8 sudah tersedia sebagai model pre-trained, untuk aplikasi spesifik dengan objek atau lingkungan yang unik, pelatihan ulang atau fine-tuning model adalah langkah krusial. Selain itu ini juga akan memastikan model dapat mengenali objek-objek relevan dalam dataset kustom dengan akurasi tinggi.



TRAINING MODEL YOLOv8



Universe Q

← Back

waste-detection
Object Detection

GENERAL

Overview

DATA

Images 9178

Dataset 10

Analytics

DEPLOY

Model 1

API Docs

9.2k images

waste-detection Computer Vision Project

AI project Updated 2 years ago

9.9k views 319 downloads

TAGS

Object Detection Model snap

CLASSES (24)

battery can cardboard_bowl cardboard_box chemical_plastic_bottle
chemical_plastic_gallon chemical_spray_can coltello light_bulb paint_bucket
plastic_bag plastic_bottle plastic_bottle_cap plastic_box
plastic_cultery plastic_cup plastic_cup_lid plastic_spoon reuseable_paper

METRICS

mAP@50 99.5% Precision 99.4% Recall 100.0%

Try This Model
Drop an image or browse your device

LANGKAH - LANGKAH TRAINING MODEL

Model Awal: Menggunakan model YOLOv8n.pt pre-trained sebagai dasar, mempercepat pelatihan dan meningkatkan efisiensi.

Dataset Kustom: Dilatih menggunakan dataset gambar sampah dari Roboflow Universe (Waste Detection), yang telah dianotasi dan diatur dalam format YOLO.

Proses Fine-tuning:
50 epoch pelatihan lanjutan pada gambar berukuran 640x640 piksel dengan batch size 16.

Memanfaatkan akselerasi GPU (device=0) untuk kecepatan optimal.

Parameter Penting (Default YOLOv8):

Optimizer: Otomatis dipilih (misal: AdamW/SGD) untuk penyesuaian bobot.

Laju Pembelajaran: Disesuaikan secara dinamis untuk konvergensi optimal.

Data Augmentation: Otomatis aktif (augment=True) untuk meningkatkan generalisasi model.

Regularisasi: Mencegah overfitting (misal: weight decay).

Early Stopping: Otomatis menghentikan pelatihan jika performa tidak meningkat (misal: patience=100).

LANGKAH - LANGKAH TRAINING MODEL

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... ^

48/50    2.05G  0.7545  0.509   1.137   21      640: 100%|██████████| 848/848 [02:53<00:00, 4.87it/s]
          Class Images Instances Box(P) R       mAP50 mAP50-95): 100%|██████████| 51/51 [00:12<00:00, 4.14it/s
          all   1607    2594     0.89    0.862    0.9    0.731

Epoch    GPU_mem  box_loss  cls_loss dfl_loss Instances Size
49/50    2.04G   0.7457  0.5018   1.131   24      640: 100%|██████████| 848/848 [02:54<00:00, 4.87it/s]
          Class Images Instances Box(P) R       mAP50 mAP50-95): 100%|██████████| 51/51 [00:12<00:00, 4.21it/s
          all   1607    2594     0.888   0.866    0.9    0.732

Epoch    GPU_mem  box_loss  cls_loss dfl_loss Instances Size
50/50    2.04G   0.7367  0.4943   1.123   24      640: 100%|██████████| 848/848 [02:54<00:00, 4.87it/s]
          Class Images Instances Box(P) R       mAP50 mAP50-95): 100%|██████████| 51/51 [00:11<00:00, 4.29it/s
          all   1607    2594     0.887   0.864    0.9    0.732

0 epochs completed in 2.646 hours.
Optimizer stripped from runs\detect\waste_detection3\weights\last.pt, 6.3MB
Optimizer stripped from runs\detect\waste_detection3\weights\best.pt, 6.3MB

Validating runs\detect\waste_detection3\weights\best.pt...
Intralytics 8.3.151 Python-3.12.6 torch-2.7.1+cu118 CUDA:0 (NVIDIA GeForce RTX 3050 Laptop GPU, 4096MiB)
Model summary (fused): 72 layers, 3,009,938 parameters, 0 gradients, 8.1 GFLOPs
          Class Images Instances Box(P) R       mAP50 mAP50-95): 100%|██████████| 51/51 [00:12<00:00, 4.05it/s

powershell
python
```

Output: Model akhir best.pt siap untuk inferensi deteksi sampah.

DAFTAR KLASIFIKASI SAMPAH

Recyclable

- Carboard box
- Can
- Plastic Bottle cap
- Plastic Bottle
- Reuseable paper

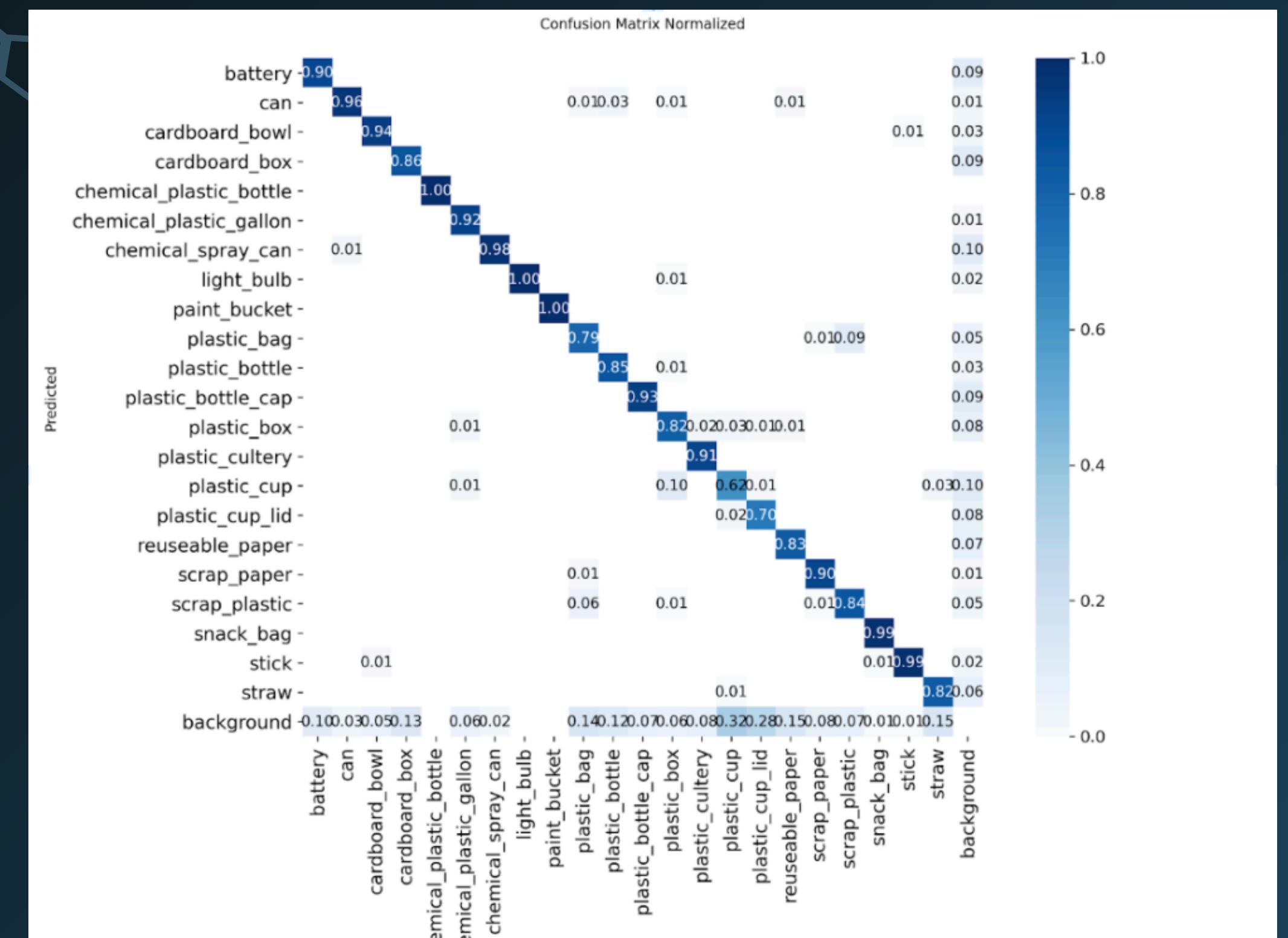
Non-Recyclable

- Plastic bag
- Scrap Paper
- Stick
- Plastic Cup
- Snack bag
- Plastic Box
- Straw
- Plastic cup lid
- Scrap Plastic
- Cardboard bowl
- Plastic cutlery

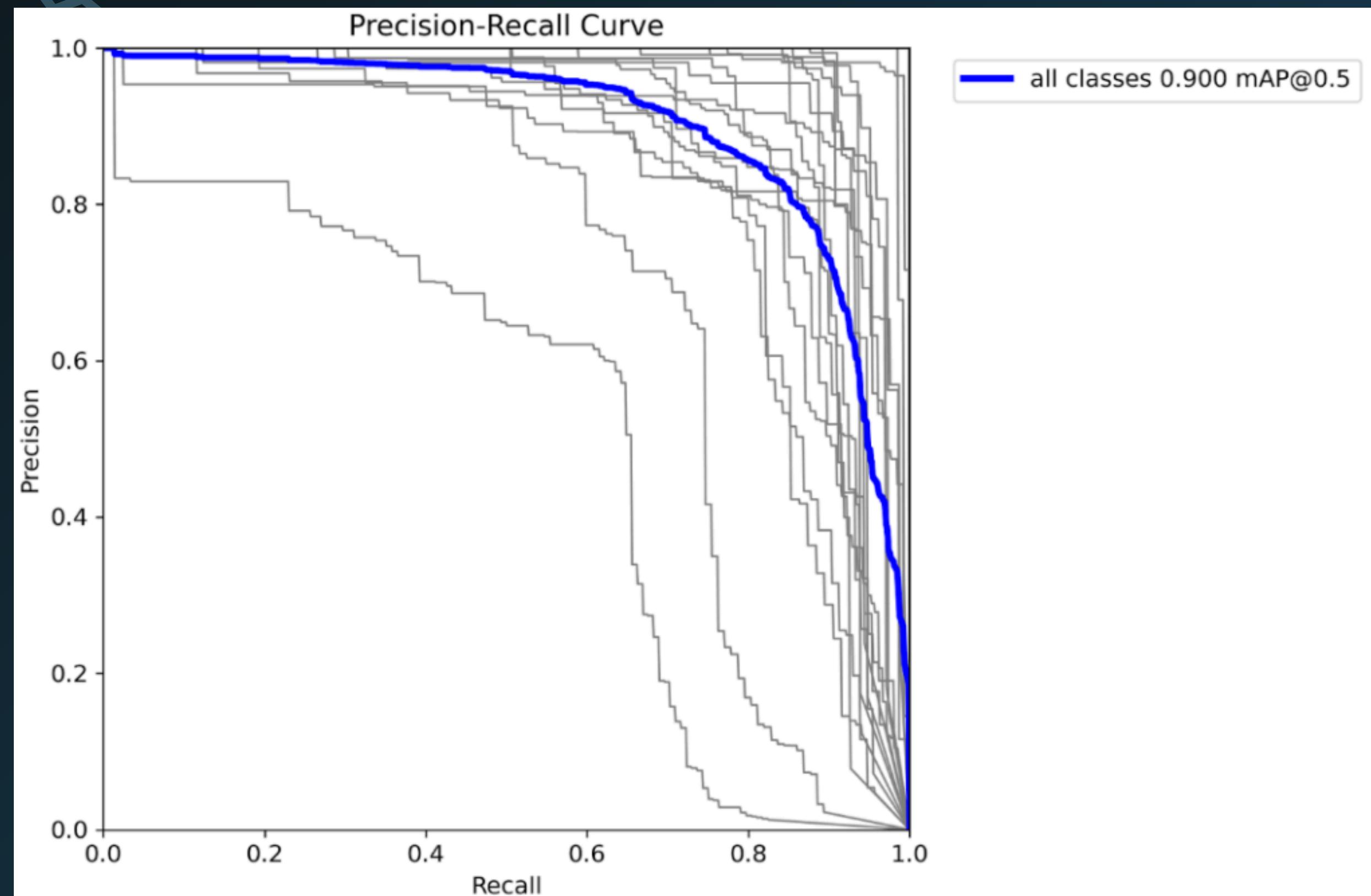
Hazardous

- Battery
- Chemical Spray can
- Chemical plastic bottle
- Chemical Plastic gallon
- Light bulb
- Paint bucket

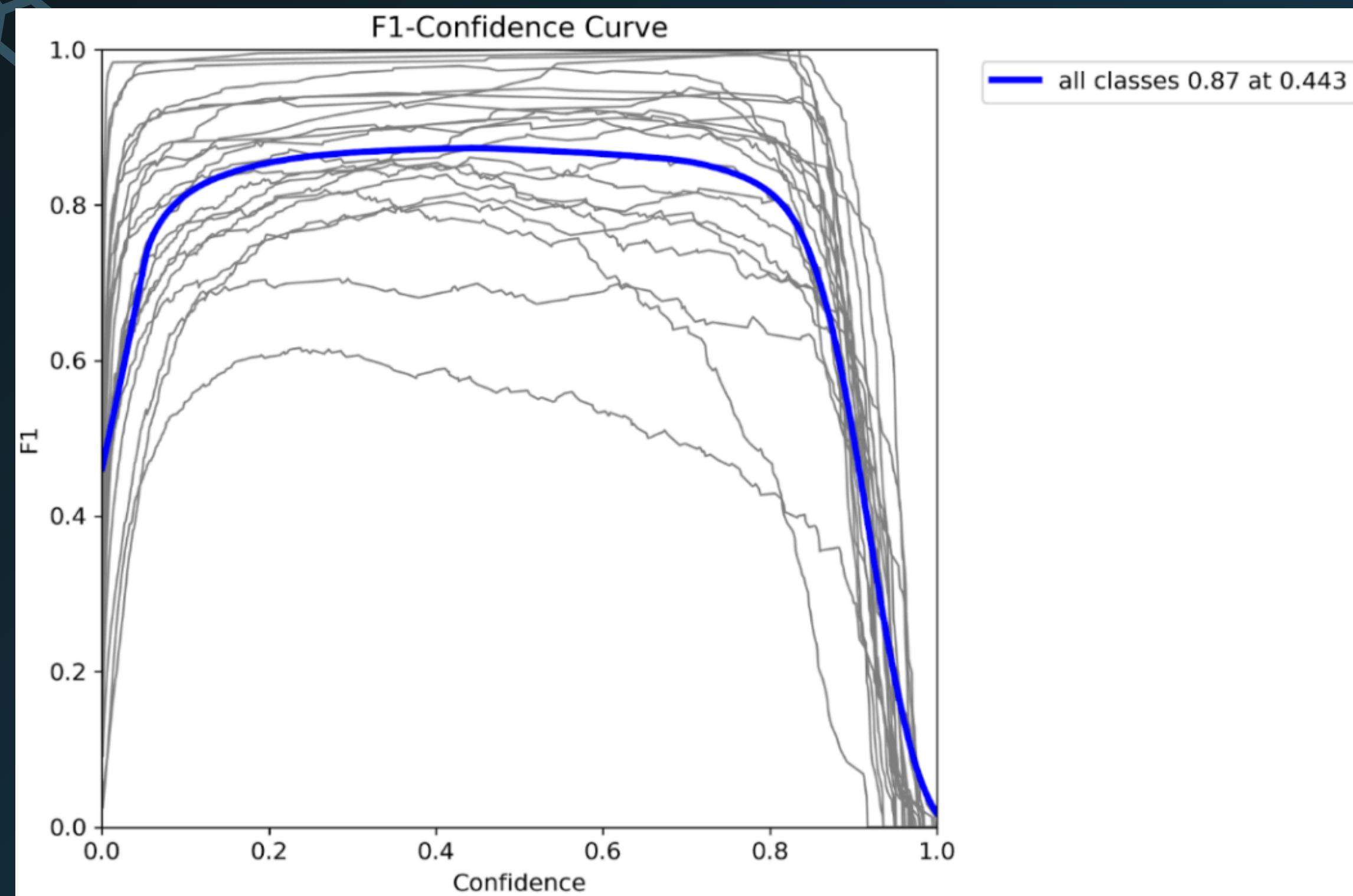
GRAFIK HASIL TRAINING - CONFUSSION MATRIX



GRAFIK HASIL TRAINING - PR CURVE



GRAFIK HASIL TRAINING - F1 CURVE



Mengintegrasikan Deteksi dengan FastAPI

Mengapa FastAPI?

- Cepat: Framework web modern untuk membangun API performa tinggi di Python.
- Mudah Digunakan: Sintaksis intuitif dan dokumentasi interaktif (Swagger UI otomatis).
- Asynchronous: Mendukung operasi asynchronous (async/await) untuk menangani permintaan secara efisien.

Peran dalam Proyek:

- Menyediakan interface agar model deteksi sampah dapat diakses dan digunakan oleh aplikasi lain (misal: frontend web).
- Menyajikan hasil deteksi (gambar dengan bounding box dan klasifikasi jenis sampah) secara real-time.

Endpoint API Utama

/detect (POST)

- Menerima gambar input (melalui upload file).
- Menjalankan inferensi YOLOv8n pada gambar tersebut.
- Mengklasifikasikan sampah yang terdeteksi (daur ulang, non-daur ulang, berbahaya).
- Mengembalikan gambar hasil deteksi (dengan bounding box) dan klasifikasi jenis sampah dalam format JSON.

**/video_feed
(GET)**

- Membuat stream video real-time dari webcam.
- Menjalankan inferensi YOLOv8n pada setiap frame video.
- Mengirimkan frame yang sudah diproses (dengan bounding box dan FPS) kembali ke frontend dalam format MJPEG stream.

Endpoint API Utama

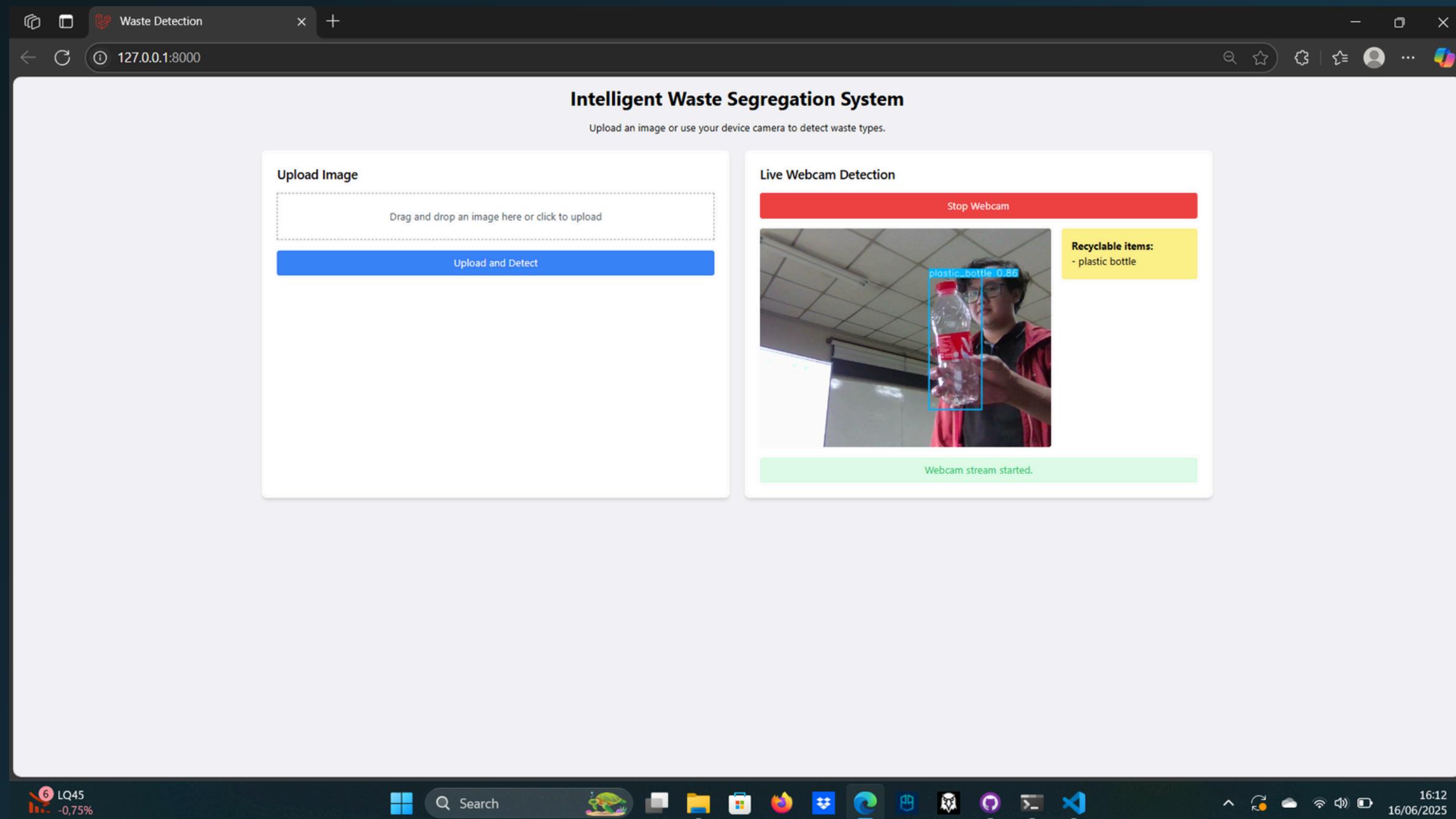
/webcam_classification (GET)

- Menyediakan endpoint terpisah untuk mendapatkan hasil klasifikasi sampah terbaru dari stream webcam secara real-time dalam format JSON, memungkinkan frontend menampilkan teks klasifikasi secara dinamis.

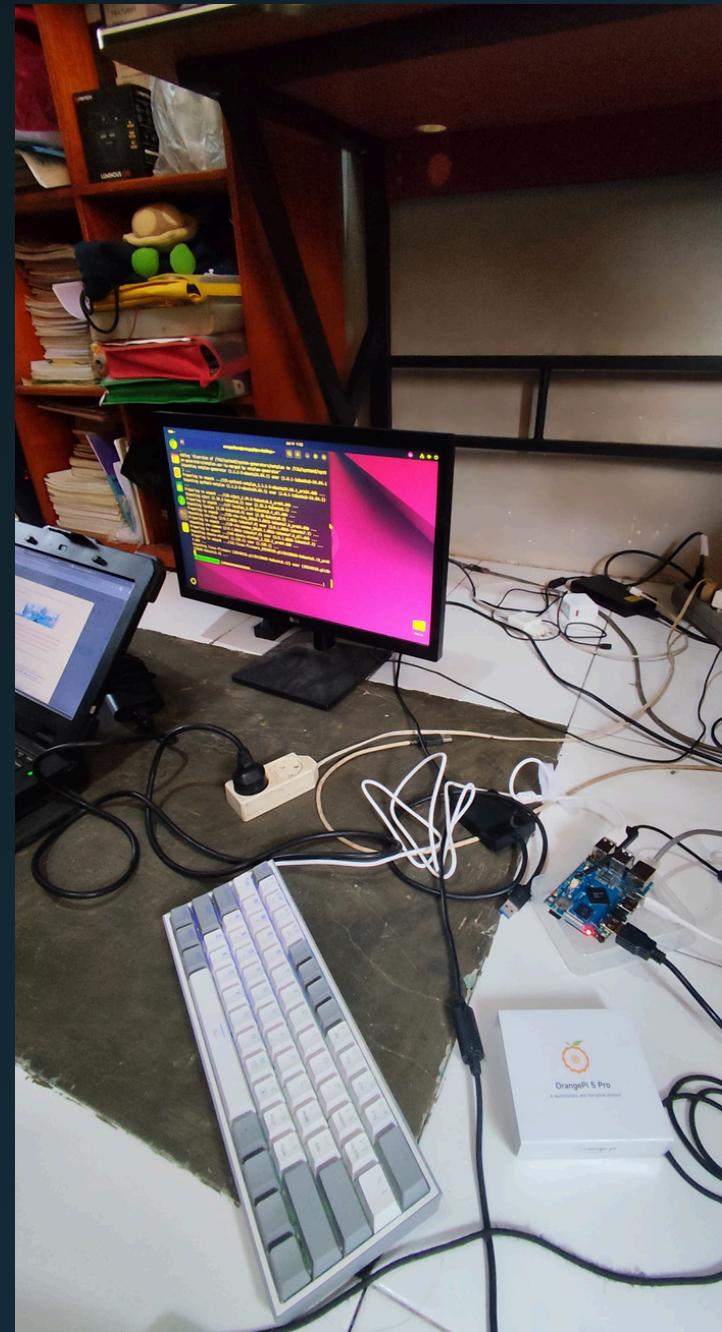
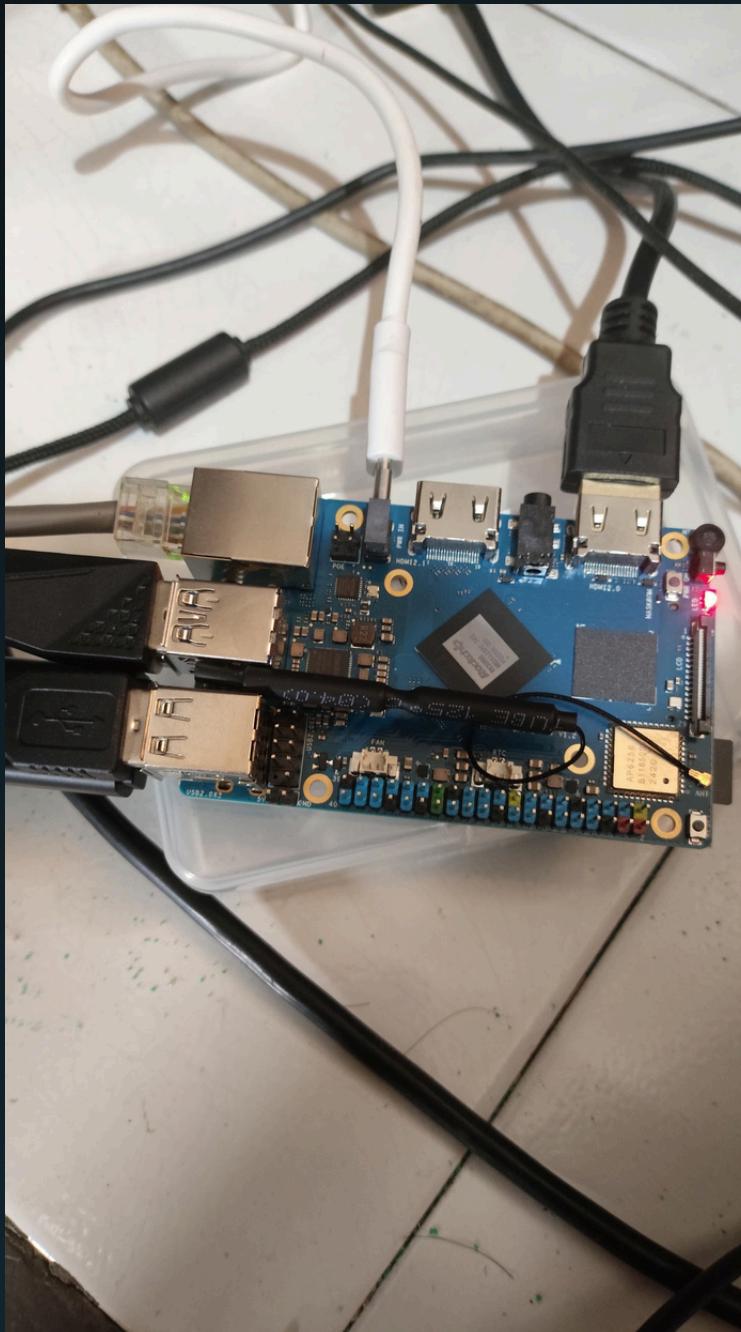
/stop_webcam_backend (POST)

- Menerima sinyal dari frontend untuk menghentikan stream webcam yang sedang berjalan, mengelola sumber daya dengan efisien.

HALAMAN WEB



SETUP ORANGE PI 5 PRO SERTA KENDALA DAN SOLUSI



Karena kami baru pertama kali memegang perangkat orange pi 5 pro, kami mengalami beberapa kesulitan saat melakukan setup, untuk menyalakan perangkat ini saja, kami masih kesulitan saat pertama kali mencobanya.

Selanjutnya kami terlebih dahulu mencari OS yang support terhadap perangkat ini, file berupa img.xyz yang kita lakukan flash ke dalam SDCard yang telah disediakan.

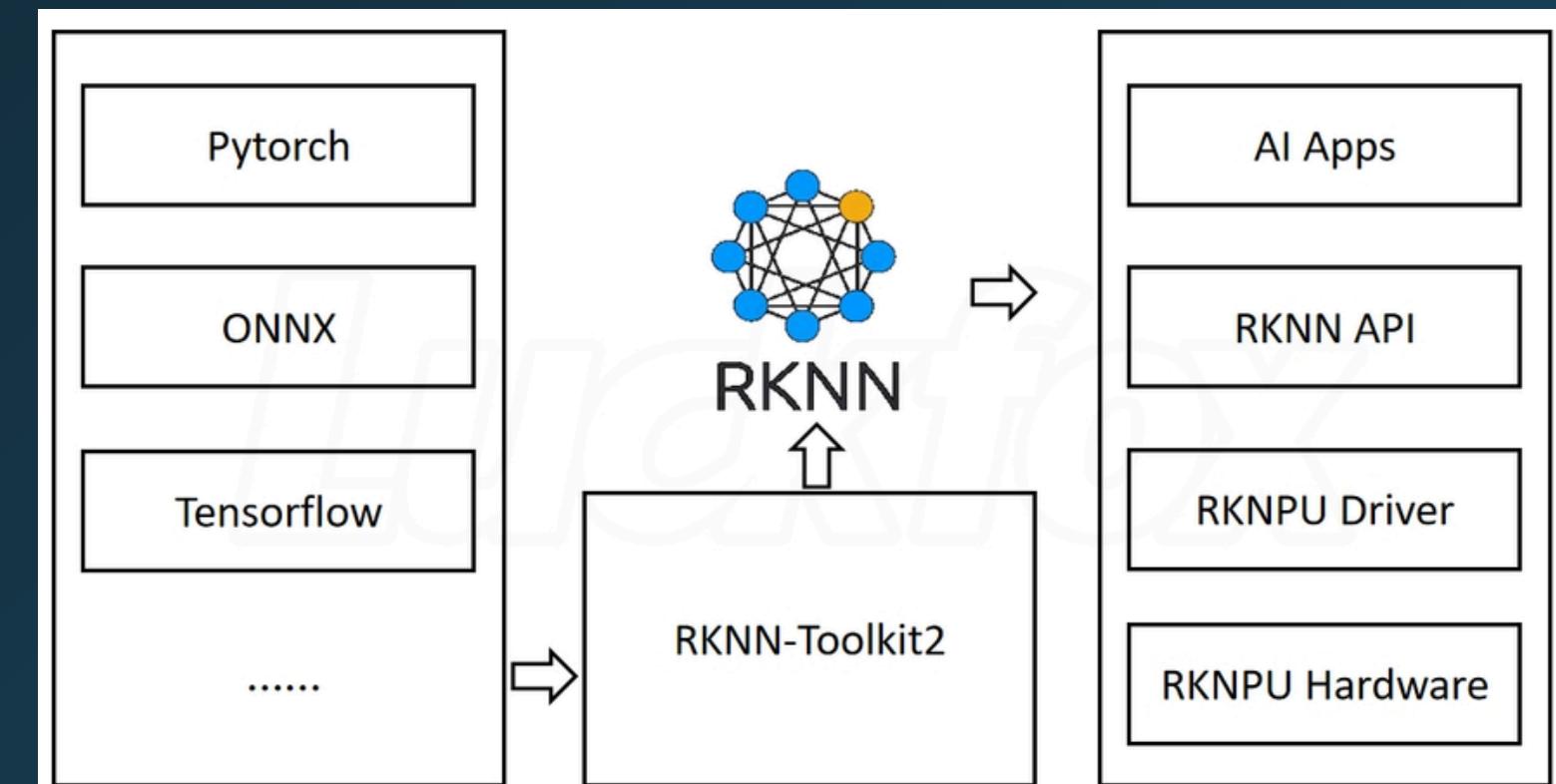
Percobaan pertama kami mengalami kendala dimana OS tersebut masih belum stabil sehingga menyebabkan perangkat tiba tiba mati. Maka dari itu kami mencari OS yang benar benar support dari website resmi orange pi. Dan pada akhirnya kami berhasil melakukan setup

KONVERSI MODEL PT KE RKNN

Model PyTorch diekspor terlebih dahulu ke format ONNX (Open Neural Network Exchange) sebagai perantara universal.

Dari ONNX, model kemudian dikonversi ke format RKNN (.rknn) menggunakan RKNN Toolkit, yang spesifik untuk NPU Rockchip.

Karena kita menggunakan Orange Pi 5 Pro, kita memilih target yang sesuai yaitu rk3588, karena dari orange Pi 5 Pro itu memiliki spesifikasi NPU Rockchip rk3588s





DUA PENDEKATAN KONVERSI



RKNN TOOLKIT 2

Konversi Manual terlebih dahulu dari model PT, lalu ke model ONNX, dan terakhir ke model RKNN. dimana kita membuat 2 file.

[Read More](#)

ULTRALYTICS

Langsung konversi dari PT ke RKNN (tetapi ada proses konversi ke model ONNX) dalam satu program

[Read More](#)

KENDALA SAAT KONVERSI DAN SOLUSI

Saat melakukan konversi model, kami menghadapi kesulitan dalam mengubah model .pt ke format .rknn. Banyak tutorial yang kami ikuti berakhir dengan kegagalan, membuat model menjadi rusak dan tidak mampu mendeteksi objek sama sekali.

Berangkat dari masalah tersebut, kami melakukan analisis mendalam dengan mencoba dua skenario konversi:

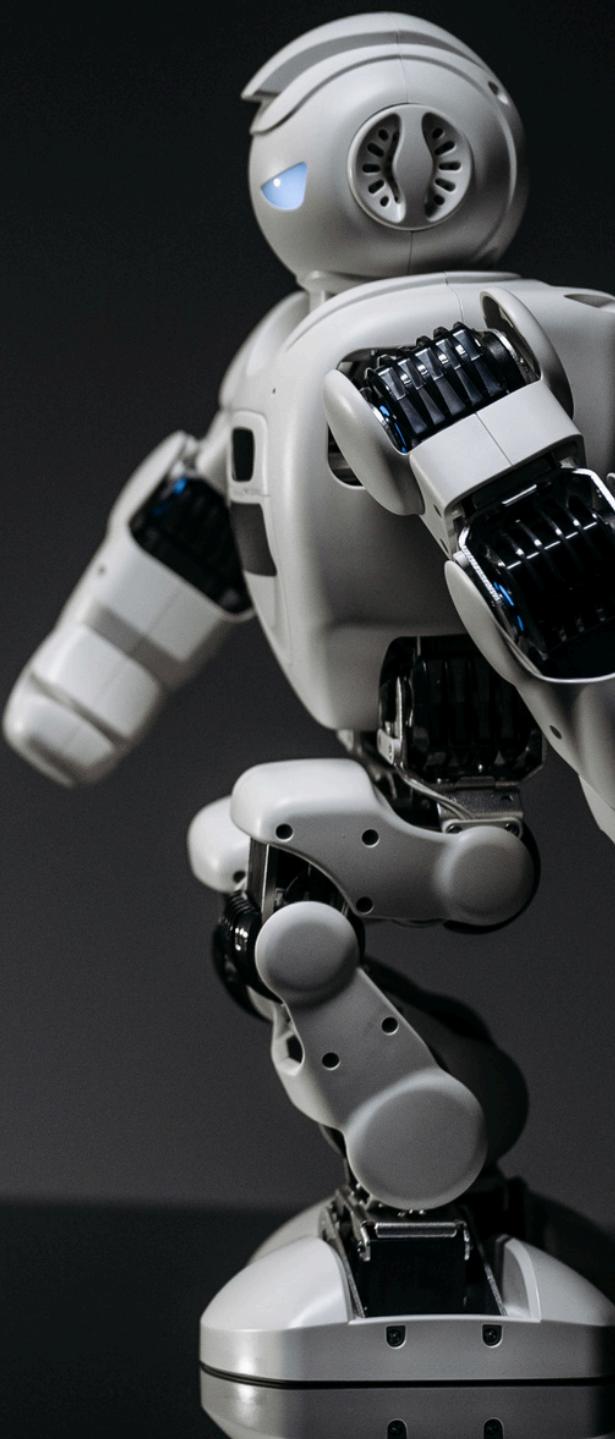
1. Dengan Kuantisasi: Pada skenario ini, model dikonversi dengan proses kuantisasi untuk mengoptimalkan ukuran dan kecepatan. Namun, hasilnya model malah tidak dapat berfungsi dan mendeteksi objek.
2. Tanpa Kuantisasi: Kami mencoba kembali proses konversi tanpa menerapkan kuantisasi. Hasilnya, model berhasil dikonversi dan dapat mendeteksi objek dengan baik.

Dari percobaan ini, kami menyimpulkan bahwa kuantisasi menjadi kendala utama dalam keberhasilan konversi model kami ke format .rknn, dan konversi tanpa kuantisasi menjadi solusi yang efektif.

KONVERSI PROGRAM FAST API

Bukan lagi ultralytics.YOLO untuk inferensi, melainkan rknnlite.api.RKNNLite.

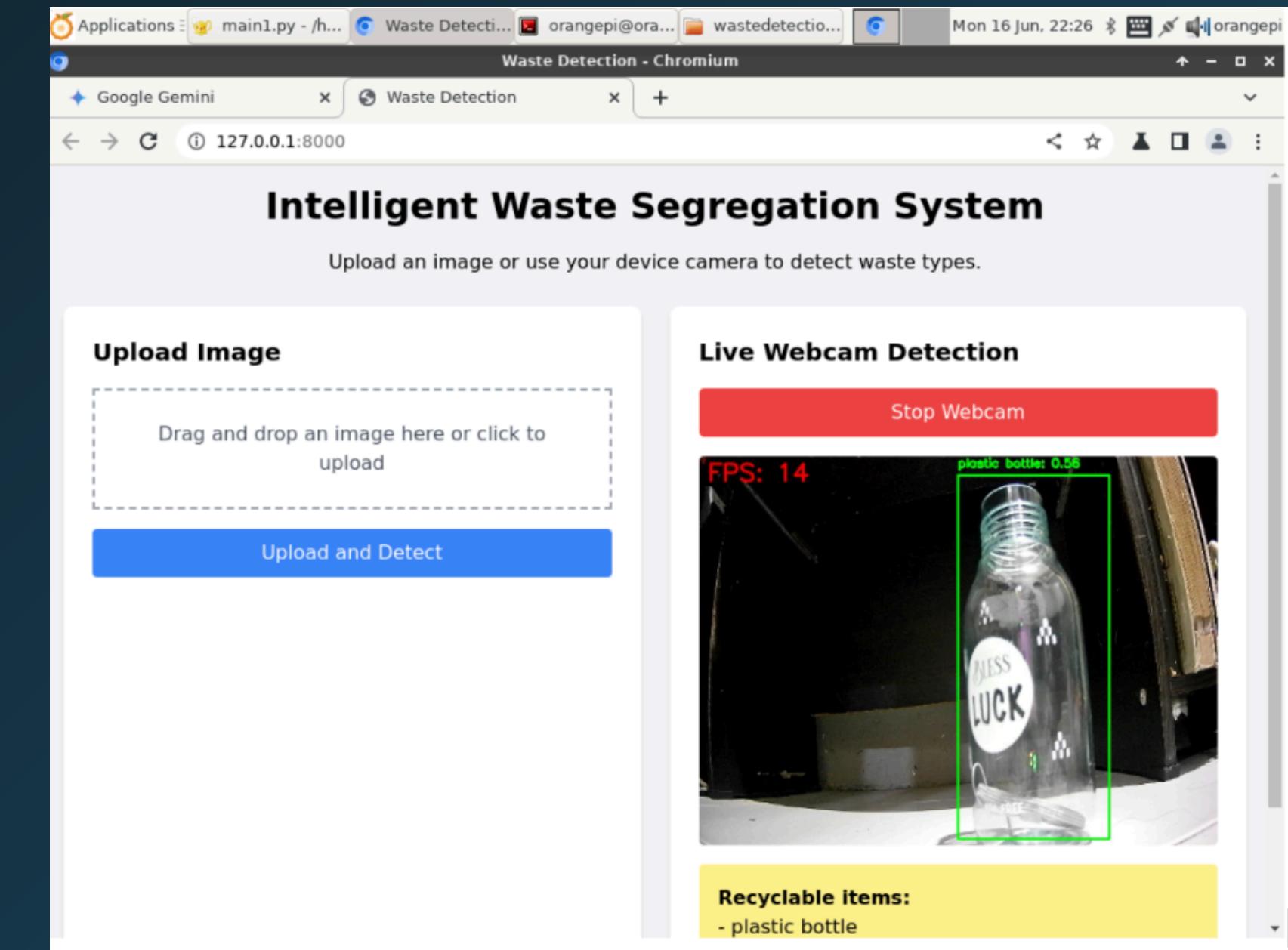
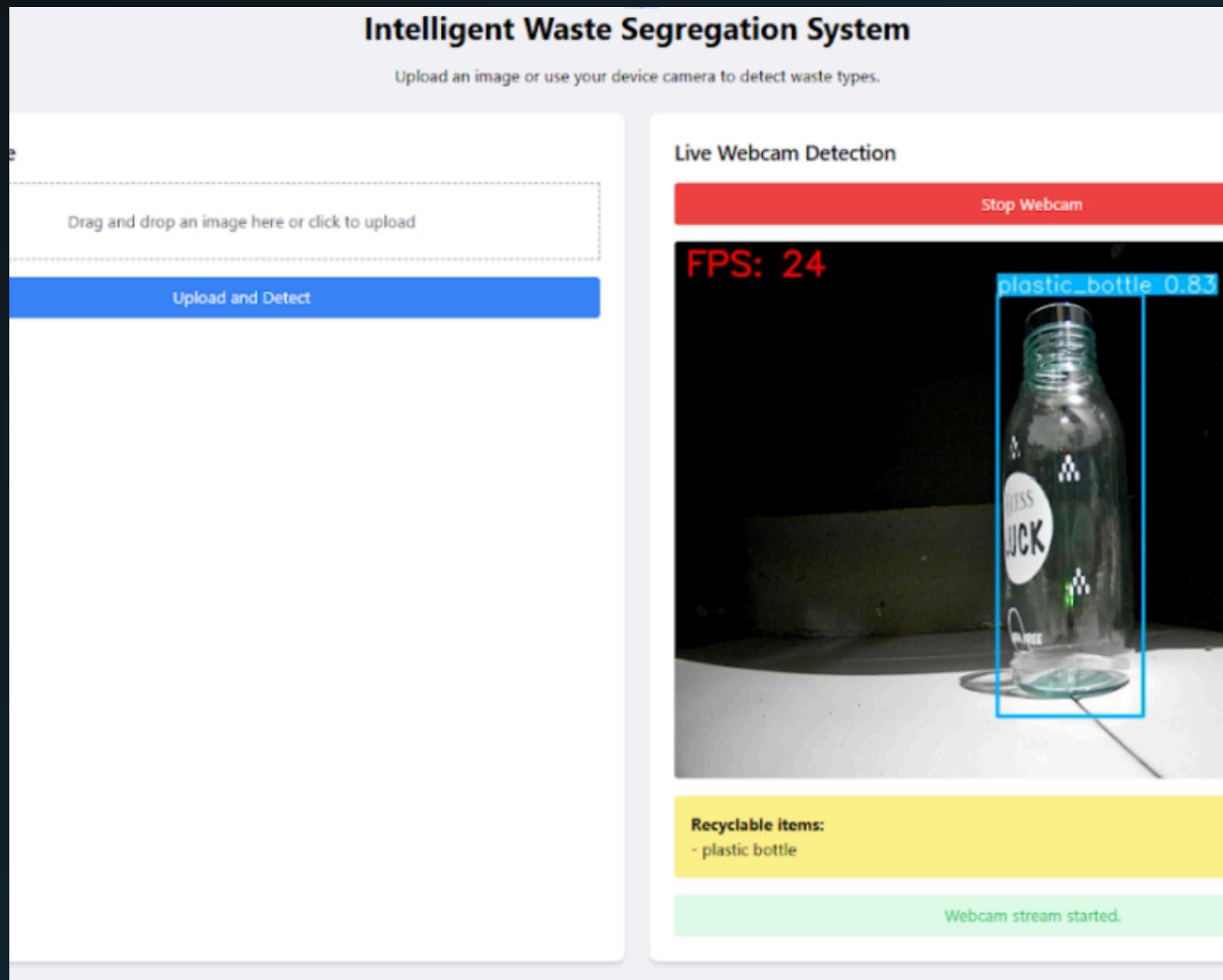
Inferensi terjadi di NPU. rknn_model.init_runtime() dan rknn_model.inference() adalah kunci yang membuat deteksi sangat cepat di Orange Pi 5.



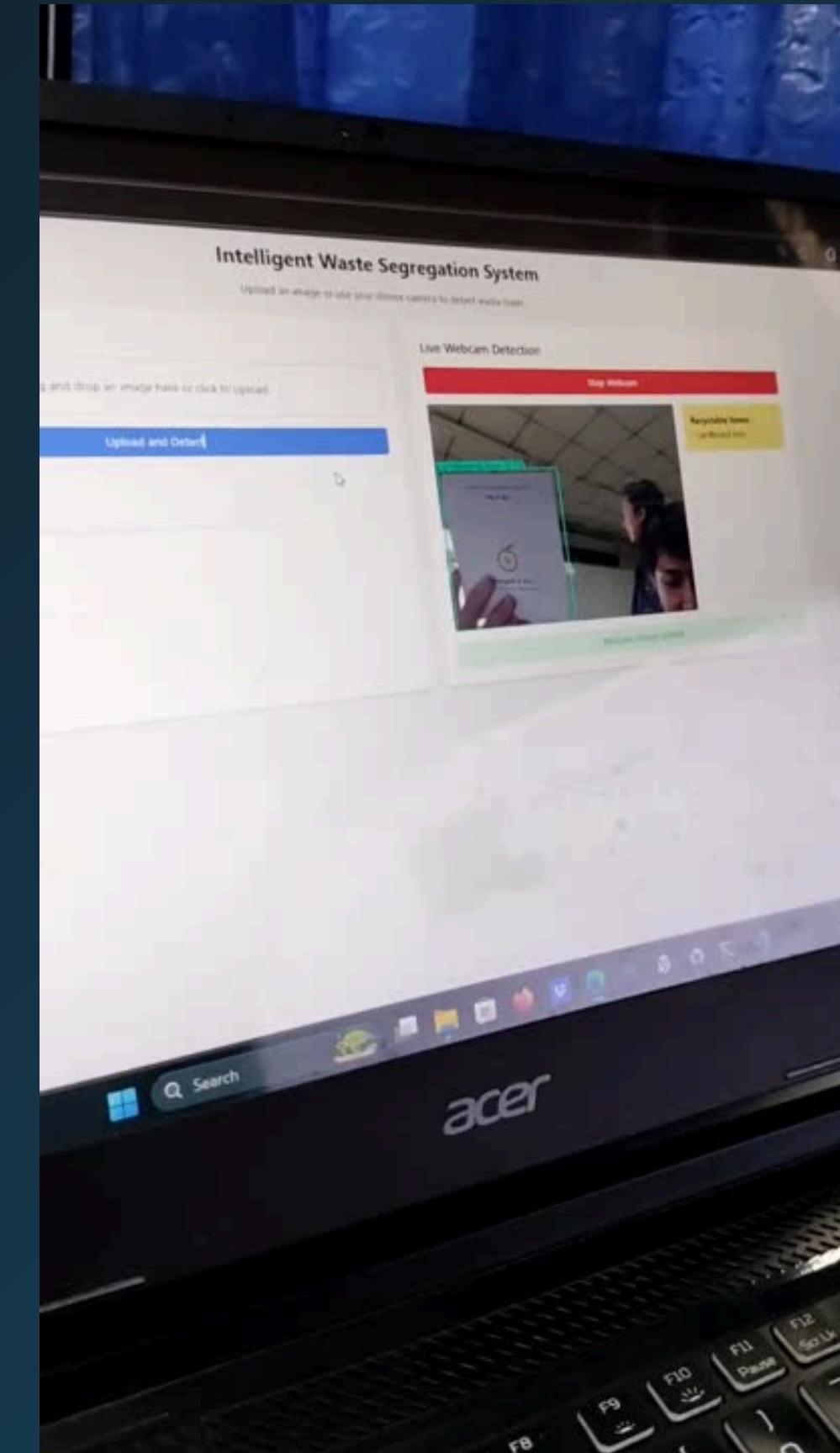
PERBANDINGAN PERFORMA RKNN DENGAN PC

PC

RKNN ORANGE PI 5 PRO



PEMBUKTIAN PROGRAM BERJALAN PADA PERANGKAT ORANGE PI 5 PRO



KESIMPULAN

Proyek ini telah berhasil mencapai tujuannya dalam mengembangkan prototipe sistem segregasi sampah cerdas. Sebuah model deteksi objek berbasis YOLOv8n berhasil dilatih untuk mengenali 22 jenis sampah dan mengklasifikasikannya ke dalam tiga kategori utama dengan performa yang cukup untuk demonstrasi fungsionalitas. Keberhasilan implementasi model ke dalam dua prototipe aplikasi web yang fungsional (API dan Perangkat ORANGE PI 5 PRO membuktikan kelayakan teknis dan konsep dari solusi yang diusulkan untuk otomatisasi pemilahan sampah.

Secara keseluruhan, berdasarkan hasil percobaan kami, Desktop PC dengan Intel i7 Gen 12 dan RTX 3050 Mobile memang memberikan performa akurasi dan FPS yang lebih tinggi dibandingkan Orange Pi 5 Pro dengan model RKNN. Ini sesuai dengan ekspektasi teoritis di mana GPU diskrit yang lebih besar dan lebih mahal menawarkan kekuatan komputasi yang lebih besar. Namun, penemuan kunci kami adalah bahwa Orange Pi 5 Pro, dengan NPU RK3588S dan model RKNN yang dikonversi tanpa kuantisasi, mampu mencapai akurasi yang tetap tinggi dan kecepatan deteksi yang sangat fungsional dalam batasan perangkat edge. Ini memvalidasi Orange Pi 5 Pro sebagai *platform* yang sangat efisien dan efektif untuk aplikasi deteksi objek *real-time* di lingkungan *embedded*.

KONTRIBUSI



DAFFA AL GHIFARI
231511038

- Mencari Dataset untuk dilakukan training pada model pretrained yolo8n.pt untuk menghasilkan model best.pt yang akan dapat lebih baik dalam mendekksi sampah
- Melakukan Training Model untuk menghasilkan model.pt dengan dataset yang telah dicari.
- Membuat WEB Fast API untuk menjalankan program deteksi sampah menggunakan model .pt (Pytorch) pada komputer.
- Melakukan Konversi Model PT (Pytorch) ke RKNN
- Membuat WEB Fast API untuk menjalankan program deteksi sampah menggunakan model RKNN pada perangkat Orange Pi 5 PRO
- Membuat file IPYNB

KONTRIBUSI



M. RAIHAN PRATAMA
231511055

- Eksplorasi Program Sejenis (Pendeteksi Sampah)
- Eksplorasi Cara Kerja YOLOv8
- Melakukan Percobaan Program Deteksi Sampah YOLOv8
- Menyusun Dokumen Proses Analisis

KONTRIBUSI



TRESNARDI FATHU R.
231511062

- SETUP perangkat Orange Pi 5 PRO, meliputi
 - Eksplorasi Cara Penggunaan Orange Pi 5 Pro.
 - SETUP Kabel, SDCard, dan lainnya.
 - Instalasi Operation System UBUNTU.
- Melakukan Konversi Model PT (Pytorch) ke RKNN
- Membuat WEB Fast API untuk menjalankan program deteksi sampah menggunakan model RKNN pada perangkat Orange Pi 5 PRO
- Menyusun Slide Presentasi

Thank You!



KELOMPOK 7

