AI Backend – provides API to run object detection on uploaded images

UI Frontend – provides a browser-based interface to upload images and view detection results

System Components

**AI Backend**

* Implemented using flask
* Runs a Yolov8-nano model
* Provides a /detect endpoint that:

1. Accepts an image.
2. Runs detection and return result as JSON.
3. Saves the annotated image and output in backend/outputs.

**UI Frontend**

* Implemented using Flask with Jinja2 templates.
* Provides a web form **index.html** to upload image
* Forwards uploaded images to AI backend.
* Display detection results:

1. JSON data
2. Annotated image inline in the browser

**Workflow**

* The user opens the UI in a browser at http://127.0.0.1:8000.
* The user uploads an image file.
* The UI forwards the image to the AI backend (http://127.0.0.1:5000/detect).
* The AI backend:

1. Runs YOLO inference.
2. Saves results in backend/outputs/.
3. Returns JSON with detections.

* The UI:

1. Displays the JSON.
2. Fetches and displays the annotated image.

API Details

**AI Backend**

* POST /detect

Input: multipart form-data, field name image.

Output (JSON):

{

"status": "ok",

"result\_json": "outputs/sample\_20250919.json",

"result\_image": "outputs/sample\_20250919.png",

"data": {

"source\_filename": "sample.jpg",

"processed\_at": "20250919T153210",

"num\_detections": 2,

"objects": [

{ "label": "person", "confidence": 0.97, "bbox": [30, 40, 200, 300] },

{ "label": "dog", "confidence": 0.88, "bbox": [220, 120, 400, 350] }

]

}

}

* GET /outputs/<filename>

Returns annotated images or JSON stored in outputs/.

**UI Frontend**

**Get /**

* Displays upload form

**POST /**

* Forwards image to backend
* Renders JSON results and annotated image

SETUP

1. Copy the project files
2. Create and activate a Python virtual environment

Windows:

python -m venv venv

venv\Scripts\activate

Linus\Mac:

Python3 -m venv venv

Source venv/bin/activate

1. Install dependencies:

pip install -r requirements.txt

1. Run AI Backend

python backend/server.py

This runs on http://127.0.0:5000

1. Run UI frontend:

python frontend/app.py

This runs on <http://127.0.0:8000>

1. Open browser at <http://127.0.0:8000> and upload an image.