# Scooper Violation Detection System

A smart computer vision system to detect violations of workers not using a scooper tool while working inside a pizza preparation area (ROI - Region of Interest). Built using \*\*YOLOv8\*\*, \*\*FastAPI\*\*, and a \*\*React + TailwindCSS\*\* dashboard.

---

## Project Structure

scooper-violation-system/

│

├── frontend/ # React frontend app with TailwindCSS

│ ├── src/

│ │ ├── App.jsx # Main component

│ │ └── index.css

│ ├── index.html

│ ├── tailwind.config.js

│ ├── postcss.config.js

│ └── package.json

│

├── streaming-service/ # FastAPI backend service

│ ├── main.py # Core detection & video streaming logic

│ ├── models/

│ │ └── best.pt # Trained YOLOv8 model from Roboflow

│ └── videos/

│ └── sample.mp4 # Test video for simulation

│

├── violations.db # SQLite database for violations

├── violations/ # Snapshot images of violations

└── README.md # This file

---

## How to Run the Project

### 1. Start the Backend (FastAPI)

```bash

cd streaming-service

pip install -r requirements.txt

uvicorn main:app --reload

This will:



Serve live video at http://localhost:8000/video/stream





Provide a violation count API at http://localhost:8000/violations/count



2. Start the Frontend (React)

cd frontend

npm install

npm run dev

Then open:

http://localhost:5173

🎯 How It Works

1.The video is read frame by frame using OpenCV.

2.

3.Each frame is passed to a YOLOv8 model (best.pt) to detect:

hand

scooper

4.It checks whether any detected hand is inside the ROI box.

5.If a hand is inside the ROI and not holding a scooper, it logs a violation:

6.Saves the frame to violations/

Inserts a new row into violations.db

7.A post-check function (review\_recent\_violations) reviews the last 5 violations:

8.If a scooper is later found near a hand in any image, that violation is removed.

🖥️ Frontend Dashboard



Streams live annotated video (/video/stream)





Displays real-time violation count, updated every 3 seconds via /violations/count.



🧠 Technical Details



Model: YOLOv8 trained on Roboflow to detect:



hand

scooper

ROI (Region of Interest): Defined as a fixed box (x1=150, y1=250, x2=550, y2=750)





Violation condition:



A hand is inside ROI

No scooper detected close to the hand (based on center distance)

Database:



violations.db using SQLite

Table: violations (id, timestamp, image\_path, violation\_type)

📦 Requirements

Backend (Python)

fastapi

uvicorn

ultralytics

opencv-python

Frontend (Node.js)

react

vite

tailwindcss

📸 Example Violation Entry



Image: violations/violation\_20250717\_103045\_123456.jpg





Timestamp: 2025-07-17 10:30:45





Type: hand\_in\_roi\_without\_scooper



📌 Notes



ROI area is drawn directly on the video stream (blue rectangle).





Violations are automatically reviewed after being saved.





Images are saved in /violations/ and deleted if false positive.





Video is streamed from videos/Sah w b3dha ghalt (2).mp4.



✅ Status

All core features required by the original PDF specification have been completed:



✅ Object detection using YOLOv8





✅ ROI violation logic implemented





✅ Real-time video stream with overlay





✅ SQLite violation database





✅ Auto-review of false violations





✅ Simple, functional frontend UI



---

Let me know if you’d like a downloadable `.zip` or to push this to GitHub with you — and I can guide you on that in one command.

Mistakes  
counter in UI is not working  
but work in database