GlobalProVision Risk Analysis

Objective:

The aim of this project is to detect specific objects—**knife**, **phone**, and **pen**—from a live camera feed via a web interface, present these objects as text, and then perform a **risk analysis** based on the detected objects and display the results to the user.

Expected Outputs:

- Detected objects will be listed as text in real time on the web UI.
- The web UI will be minimalistic and user-friendly.
- All detected objects will be grouped and displayed under a heading on the interface.

Technologies:

Duty	**Technology**
Image Processing & Detection	Python + YOLO
Backend	Next.js/React
Frontend	Next.js/React
Video	Webcam

Role Distribution:

Backend API Developer: Mehmet Said Hüseyinoğlu

Image Processing Developer: Beyza Nur Büyükdemir

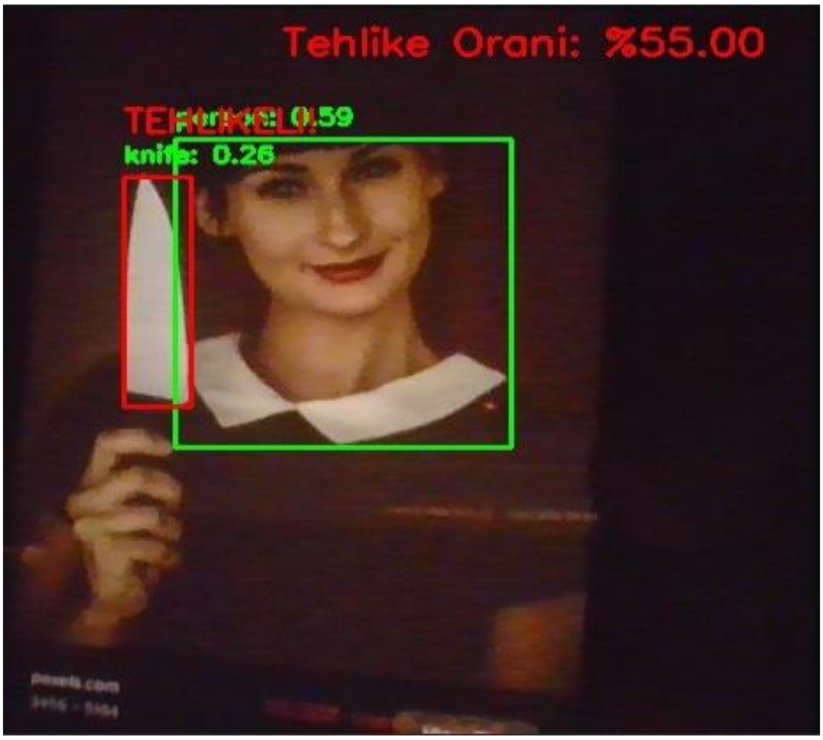
Frontend/Web UI Developer: Batuhan Acar

Integration & Data Flow Tester**:** Hüseyin Alperen Kavukçu

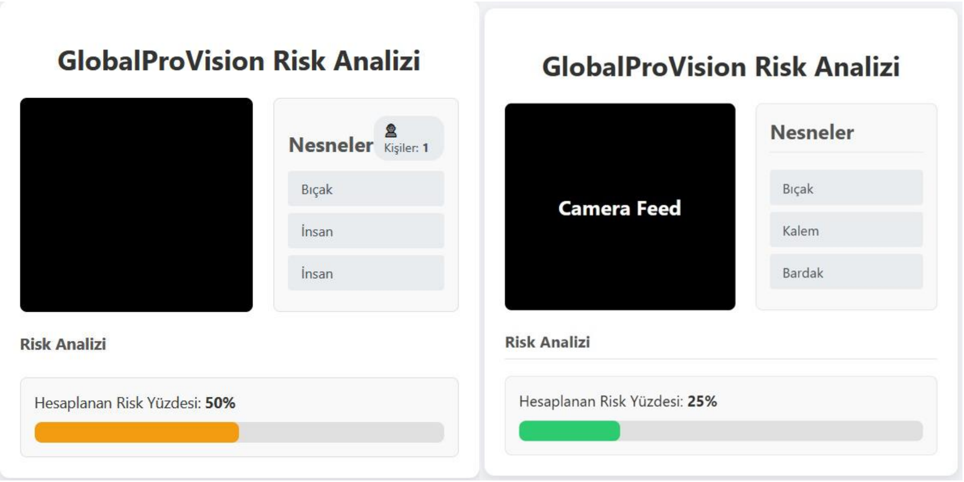
UX & Documentation Lead: Ekin Doğan

Results:

1. The object detection and image processing model was successfully implemented**.**
The knife, phone, and pen objects were trained and recognized by the model.
When an object is detected, it will appear inside a green bounding box on the screen. If the perceived threat increases, the box will turn red. The risk level will also be shown in the top-right corner of the screen.



2. A simple and user-friendly web UI was developed**.** Detected objects will appear live on the right side of the interface. The calculated risk value will be displayed at the bottom. The interface helps users understand; what objects are in the room, the risk score. the reason behind the calculated risk


GlobalProVision Risk Analizi

Nesneler — Kişiler: 1

- Bıçak
- İnsan
- İnsan

Risk Analizi

Hesaplanan Risk Yüzdesi: **50%**

GlobalProVision Risk Analizi

Camera Feed

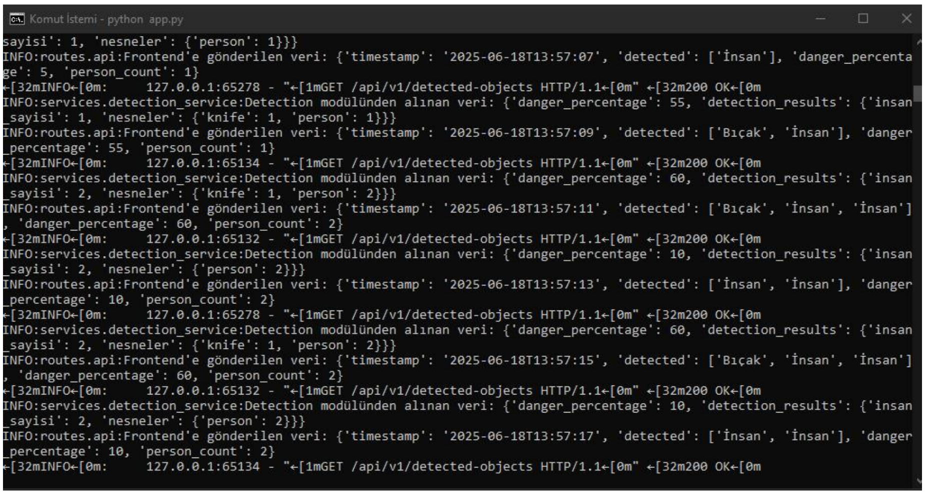
Nesneler

- Bıçak
- Kalem
- Bardak

Risk Analizi

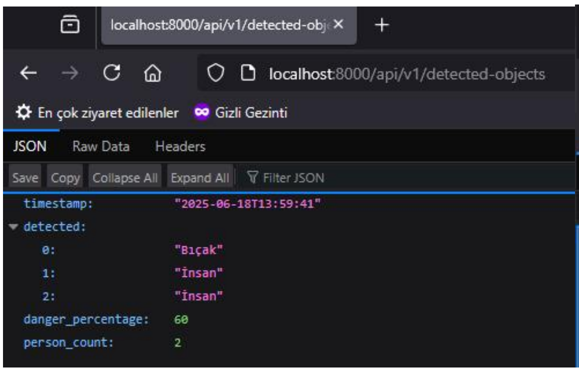
Hesaplanan Risk Yüzdesi: **25%**

3. API development and data integration were completed. Real-time data from the backend is successfully displayed on the frontend.

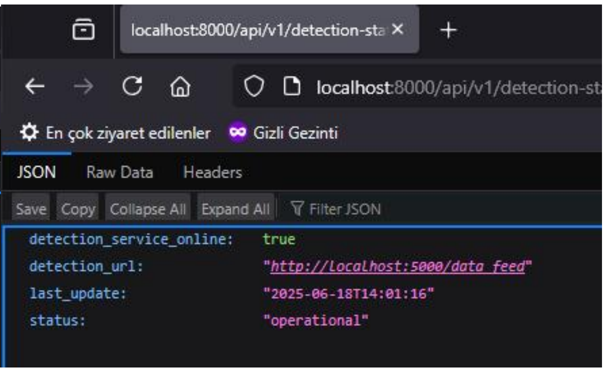


```
sayisi': 1, 'nesneler': {'person': 1}}}
INFO:routes.api:Frontend'e gönderilen veri: {'timestamp': '2025-06-18T13:57:07', 'detected': ['İnsan'], 'danger_percenta
ge': 5, 'person_count': 1}
←[32mINFO←[0m:     127.0.0.1:65278 - "←[1mGET /api/v1/detected-objects HTTP/1.1←[0m" ←[32m200 OK←[0m
INFO:services.detection_service:Detection modülünden alınan veri: {'danger_percentage': 55, 'detection_results': {'insan
_sayisi': 1, 'nesneler': {'knife': 1, 'person': 1}}}
INFO:routes.api:Frontend'e gönderilen veri: {'timestamp': '2025-06-18T13:57:09', 'detected': ['Bıçak', 'İnsan'], 'danger
_percentage': 55, 'person_count': 1}
←[32mINFO←[0m:     127.0.0.1:65134 - "←[1mGET /api/v1/detected-objects HTTP/1.1←[0m" ←[32m200 OK←[0m
INFO:services.detection_service:Detection modülünden alınan veri: {'danger_percentage': 60, 'detection_results': {'insan
_sayisi': 2, 'nesneler': {'knife': 1, 'person': 2}}}
INFO:routes.api:Frontend'e gönderilen veri: {'timestamp': '2025-06-18T13:57:11', 'detected': ['Bıçak', 'İnsan', 'İnsan']
, 'danger_percentage': 60, 'person_count': 2}
←[32mINFO←[0m:     127.0.0.1:65132 - "←[1mGET /api/v1/detected-objects HTTP/1.1←[0m" ←[32m200 OK←[0m
INFO:services.detection_service:Detection modülünden alınan veri: {'danger_percentage': 10, 'detection_results': {'insan
_sayisi': 2, 'nesneler': {'person': 2}}}
INFO:routes.api:Frontend'e gönderilen veri: {'timestamp': '2025-06-18T13:57:13', 'detected': ['İnsan', 'İnsan'], 'danger
_percentage': 10, 'person_count': 2}
←[32mINFO←[0m:     127.0.0.1:65278 - "←[1mGET /api/v1/detected-objects HTTP/1.1←[0m" ←[32m200 OK←[0m
INFO:services.detection_service:Detection modülünden alınan veri: {'danger_percentage': 60, 'detection_results': {'insan
_sayisi': 2, 'nesneler': {'knife': 1, 'person': 2}}}
INFO:routes.api:Frontend'e gönderilen veri: {'timestamp': '2025-06-18T13:57:15', 'detected': ['Bıçak', 'İnsan', 'İnsan']
, 'danger_percentage': 60, 'person_count': 2}
←[32mINFO←[0m:     127.0.0.1:65132 - "←[1mGET /api/v1/detected-objects HTTP/1.1←[0m" ←[32m200 OK←[0m
INFO:services.detection_service:Detection modülünden alınan veri: {'danger_percentage': 10, 'detection_results': {'insan
_sayisi': 2, 'nesneler': {'person': 2}}}
INFO:routes.api:Frontend'e gönderilen veri: {'timestamp': '2025-06-18T13:57:17', 'detected': ['İnsan', 'İnsan'], 'danger
_percentage': 10, 'person_count': 2}
←[32mINFO←[0m:     127.0.0.1:65134 - "←[1mGET /api/v1/detected-objects HTTP/1.1←[0m" ←[32m200 OK←[0m
```

4. Testing was completed. All data pipelines were verified and confirmed to function correctly. The project is finalized.

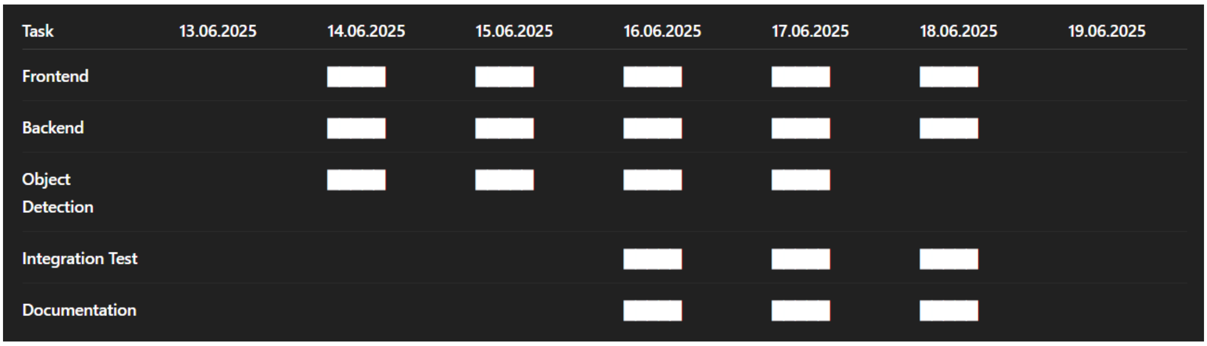


```
timestamp:            "2025-06-18T13:59:41"
▾ detected:
    0:                "Bıçak"
    1:                "İnsan"
    2:                "İnsan"
danger_percentage:    60
person_count:         2
```



```
detection_service_online:    true
detection_url:               "http://localhost:5000/data_feed"
last_update:                 "2025-06-18T14:01:16"
status:                      "operational"
```

Project Timeline:



Task	13.06.2025	14.06.2025	15.06.2025	16.06.2025	17.06.2025	18.06.2025	19.06.2025
Frontend		■	■	■	■	■	
Backend		■	■	■	■	■	
Object Detection		■	■	■	■		
Integration Test				■	■	■	
Documentation				■	■	■	

References:

https://github.com/SIYAKS-ARES/GlobalProVision