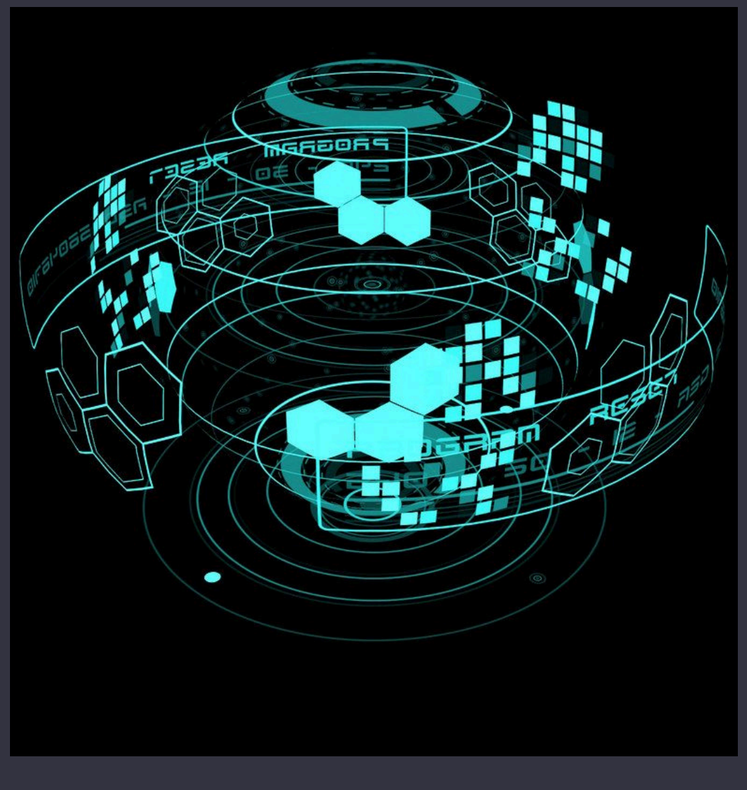
Allpueli: Real-Time AI Emergency Detection System

01

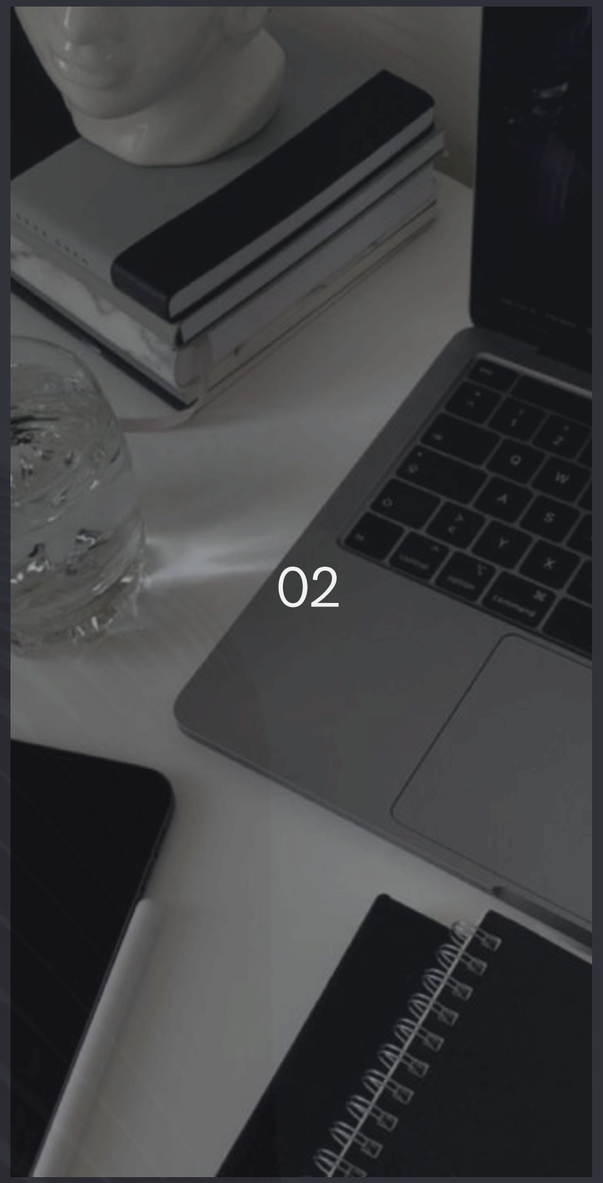
Alp Eren Aydin, Alexander Mateides, Samuel Luis Štúber, Amalia Vinciunaite



Our business proposal

- Monitors public transport dashcams to detect emergencies in real-time.

- Helps emergency services respond faster while protecting passenger privacy.



Who can use it?

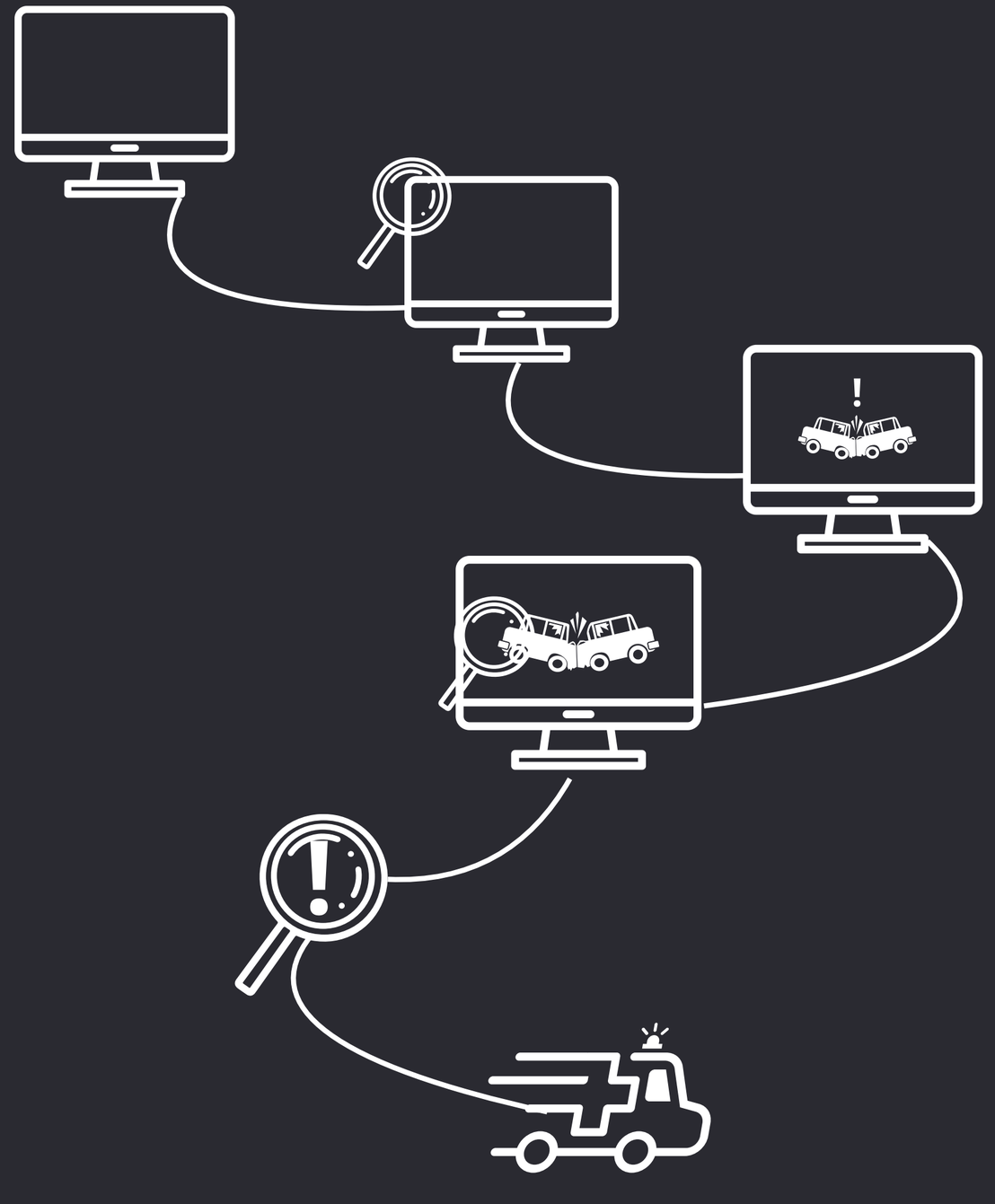
CITY SERVICES

EMERGENCY SERVICES

DAILY USE

04

How does it work?



Why AI?

SPEED

- Allows fast analysis of great volumes of data

EFFICIENCY

- Delegates only the important actions to human operators

FUTURE IMPROVEMENT

- Can use historical data to update itself continuously

Challenges

Time to analyze the video (5-10 minutes for short segment) made iteration difficult

We didn’t manage to consistently detect vehicle lane for speed estimation

The video didn’t have enough metadata (such as geolocation, kinetics data)

07

Risks & ethics

PRIVACY

- Possible identification of faces or other personal data.
- Risk of video stream leakage or misuse of access.

FAIRNESS

- Biased datasets may cause the algorithm to unfairly detect ‘dangerous events’ more frequently or miss some in different circumstances

FALSE POSITIVES

- False detection of a ‘dangerous’ situation may lead to unnecessary intervention or deployment of emergency services.

Mitigation

PRIVACY

- Use of automatic face blurring or other de-identification methods.
- Storing data only when an incident is confirmed, rather than continuously

FAIRNESS

- Implementing a ‘human-in-the-loop’ procedure: critical decisions are not made fully automatically.

FALSE POSITIVES

- Adjusting optimal detection thresholds.
- Using a multistage verification process (automatic model first and then a human operator).

```
base = os.path.basename(vid)
trimmed_path = f"trim_{base}"

fourcc = cv2.VideoWriter_fourcc(*'mp4v')
out = cv2.VideoWriter(trimmed_path, fourcc, fps, (w, h))

frame_count = 0
saved_frames = 0

while True:
    ret, frame = cap.read()
    if not ret:
        break

    if MAX_FRAMES != -1 and frame_count >= MAX_FRAMES:
        break

    # Save every Nth frame
    if frame_count % FRAME_SKIP == 0:
        out.write(frame)
        saved_frames += 1
```

Process & pipeline

Video analysis

- uses Python script with multiple models and CV methods to detect the “dangerous” events

Frontend

- Based on React TS

Backend

- Python FastAPI server

App

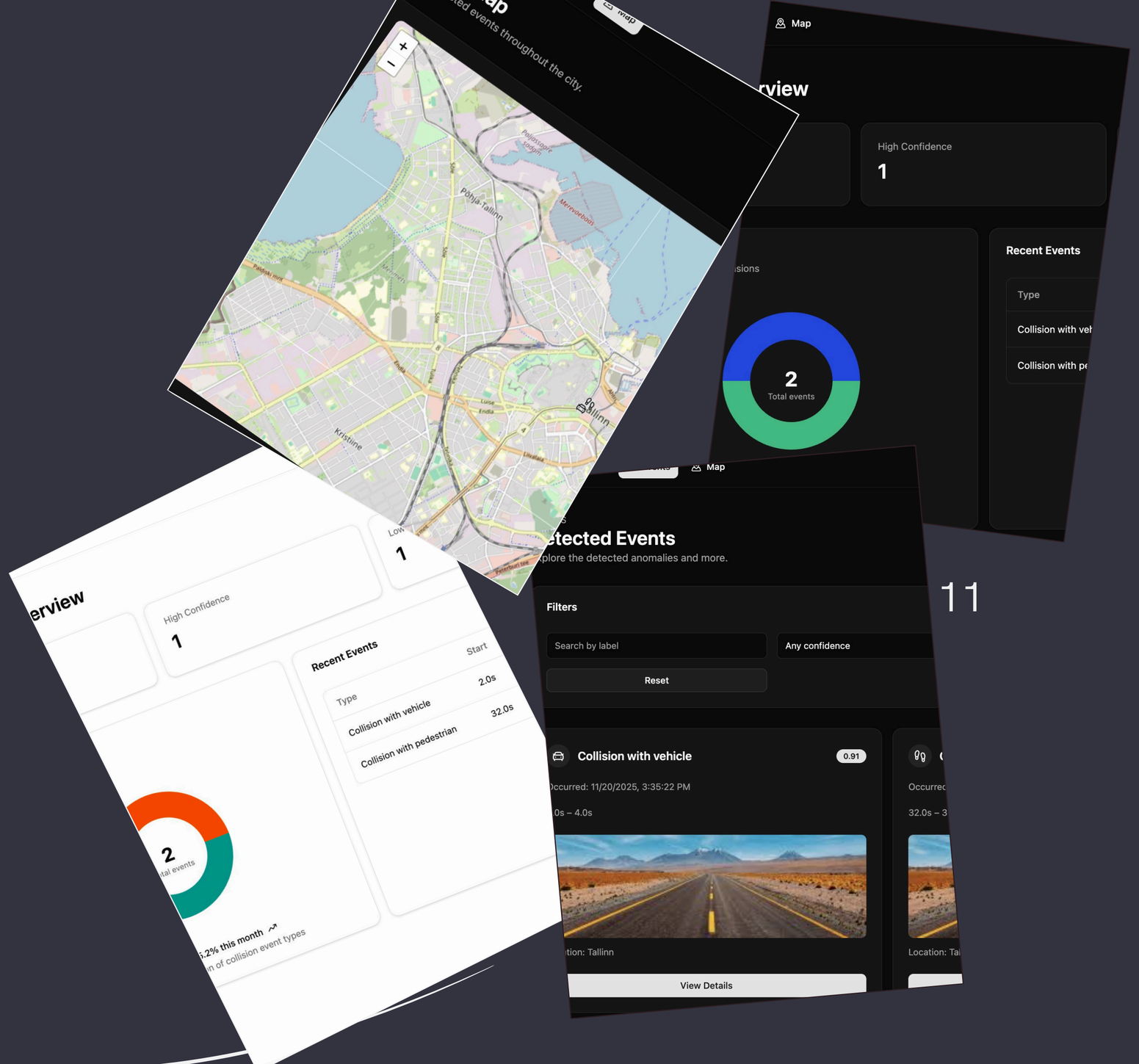
- Fully dockerized, ready for deployment

Next steps & Improvements

- Use data from dedicated sensors (geolocation, velocity) instead of using video for estimation
  - More accurate results
  - Saves time for analysis
- Implement real-time streaming/analysis
- Improve GUI and features (report, organizing events)
- Add face/car plate blurring to protect privacy
- Optionally allow non-public vehicles to join the program

Result

- APP


Map
High Confidence
1
Recent Events
Type
Collision with vehicle
Collision with pedestrian
2
Total events
Filters
Search by label
Any confidence
Reset
Collision with vehicle
0.91
View Details

Thank you for listening!