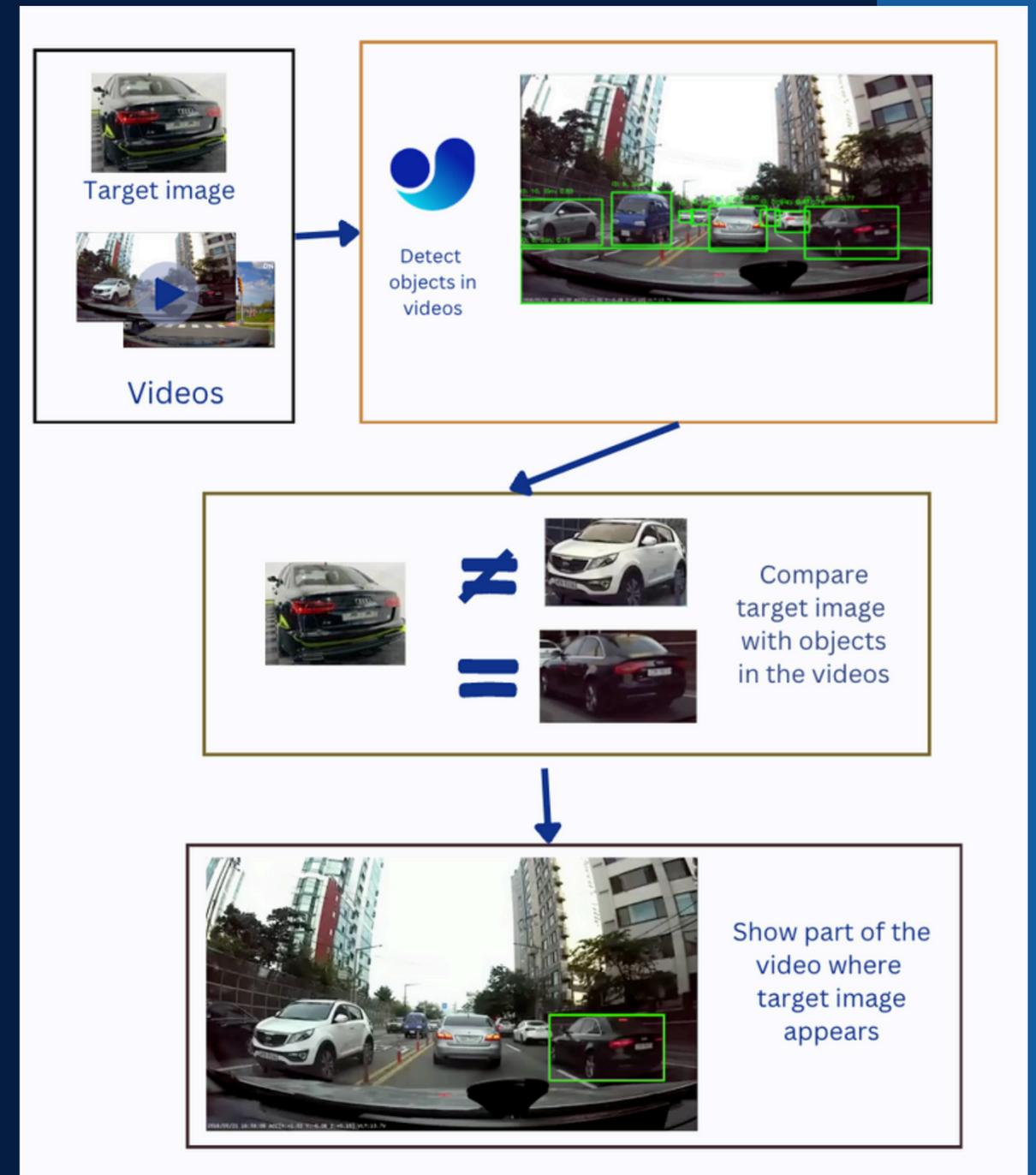


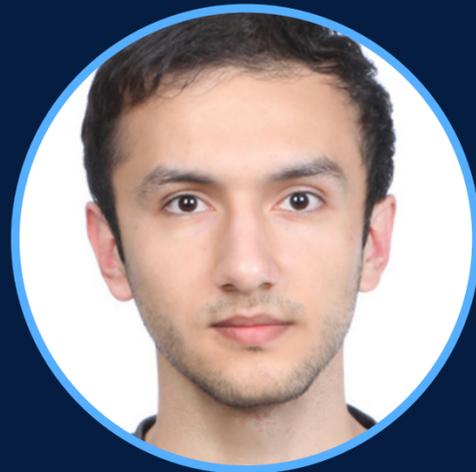


CRIME SCENE DETECTION

Final project presentation
Presented by: AI Ninjas



OUR TEAM



Sukhrob
Ikromov

Leader



Andrei
Laptev

Co-leader and
presentation maker



Shruthi
Atkuri

Report writer



Sherzod
Uralov

Presenter

PARTICIPATING COMPANY

GMDSOFT

GMDSOFT

worldwide leading research group of mobile
and digital forensics.



AGENDA

01

Problem

02

Solution

03

Design

04

Result

05

Demo

06

Next solutions

GMD
SOFT

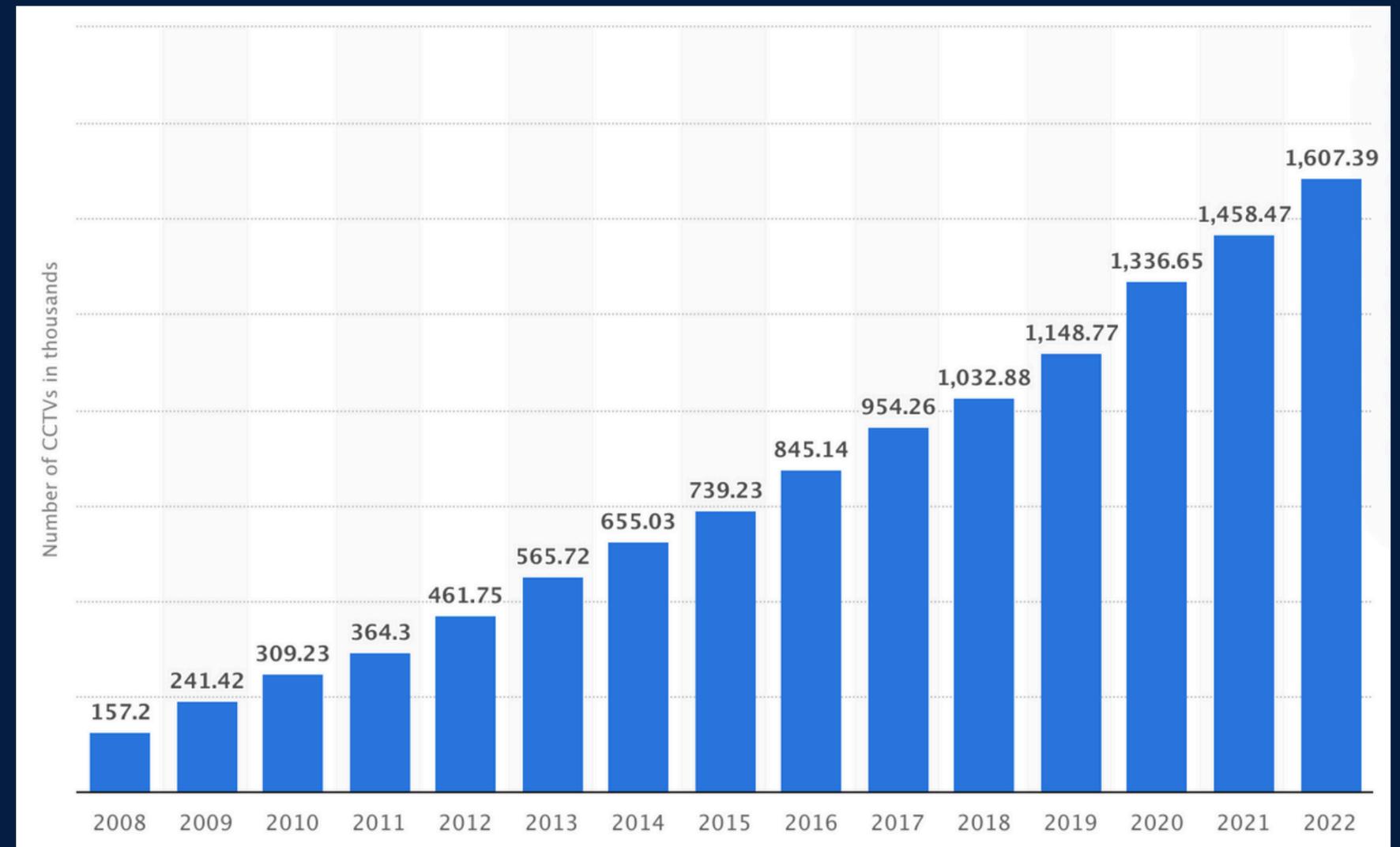
PROBLEM



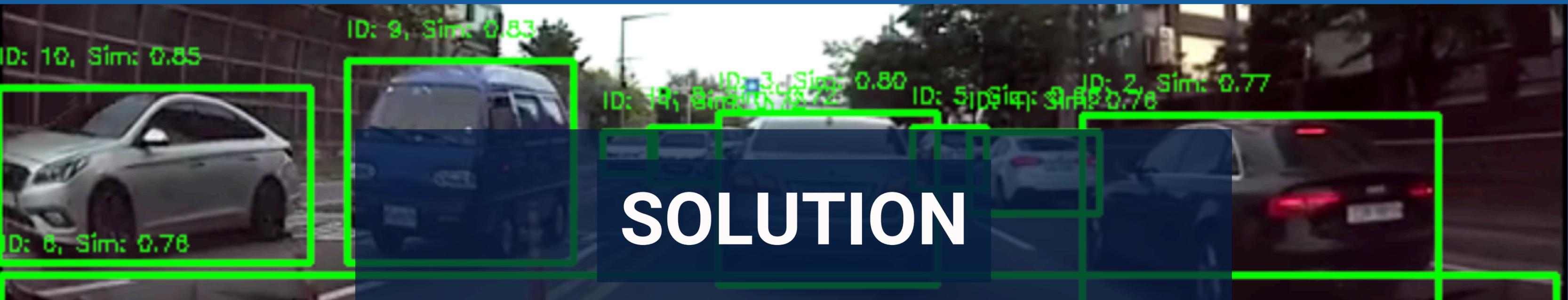
- **More videos:** The amount of video evidence increases.



- **Review pressure:** Tedious for humans.
- **Investigation length:** Longer & more complex.



Number of CCTV cameras in Korea



SOLUTION

Detect objects



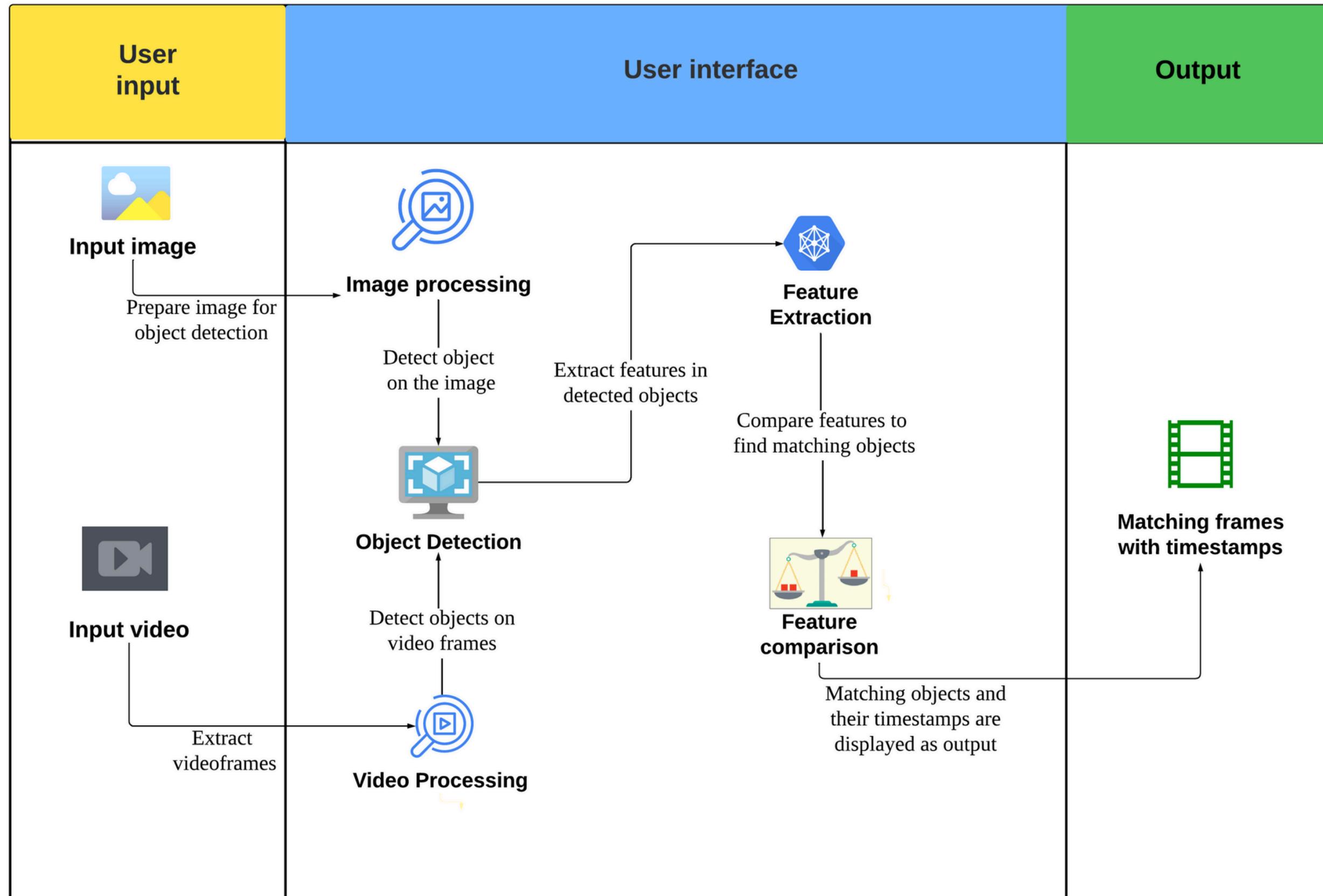
Compare target and detected objects



Save part of a video where target object appears



System Architecture Diagram

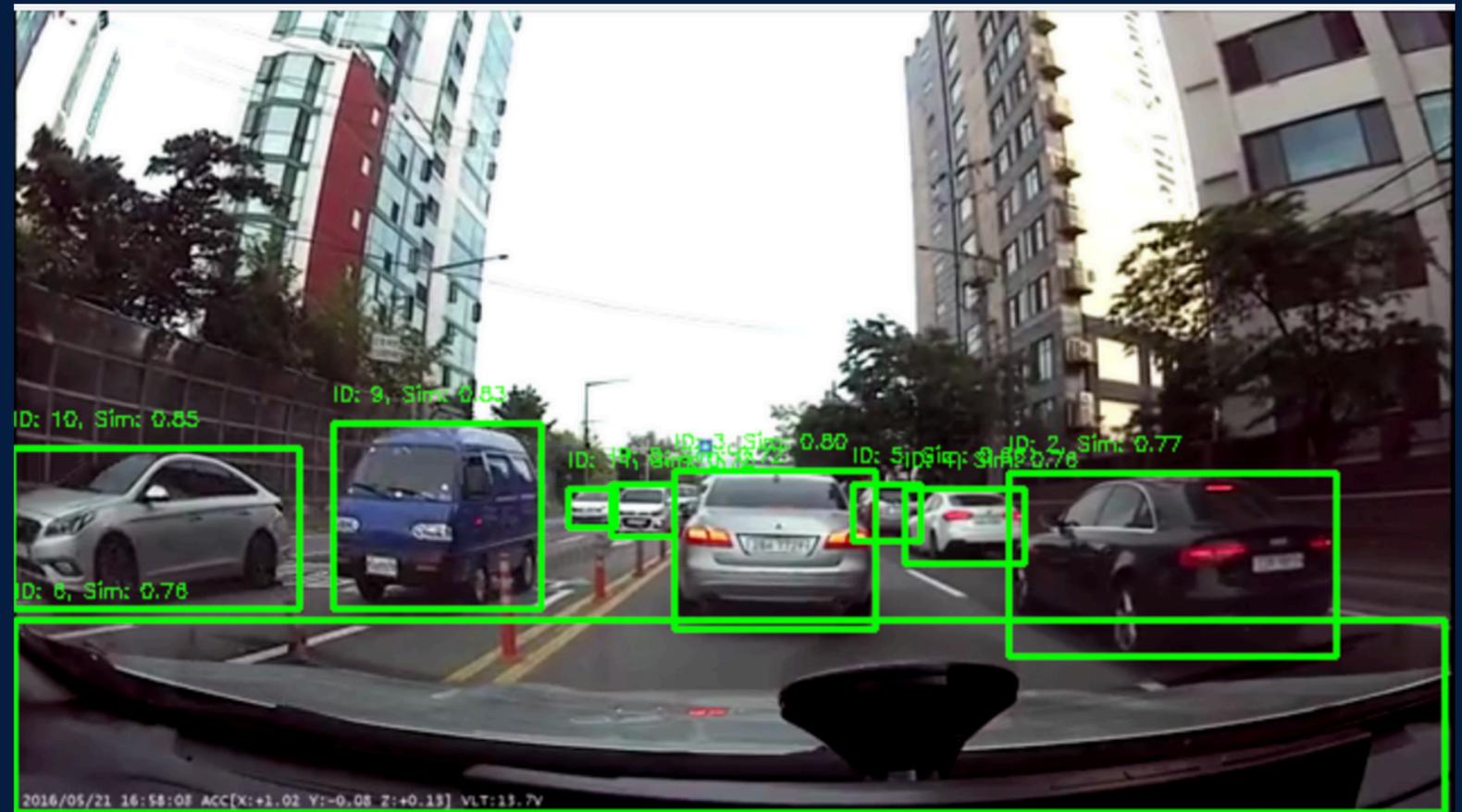




DESIGN

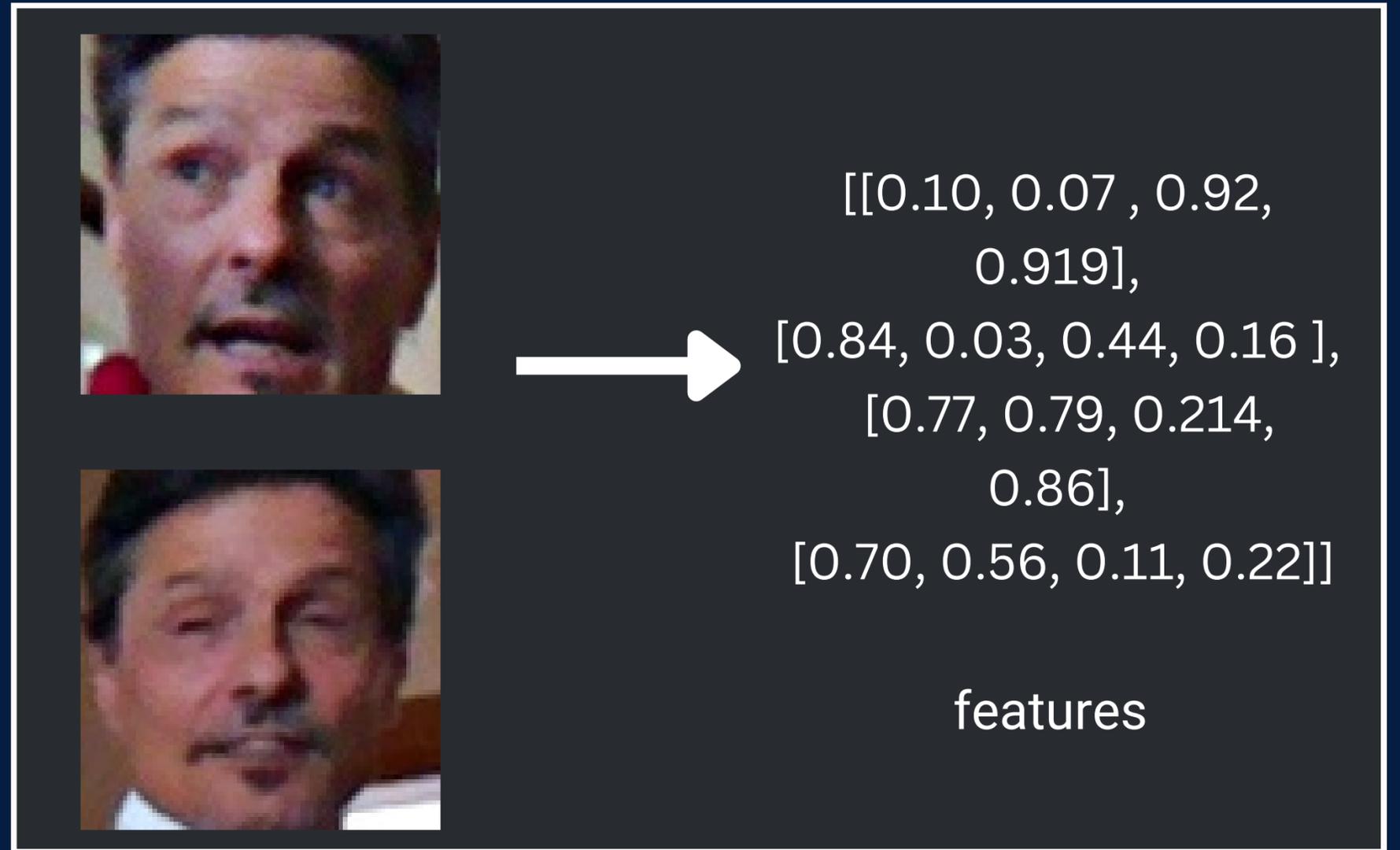
DETECTION

- Detect objects
- Follow (track) objects in each frame
- Save their coordinates and unique ID



FEATURE EXTRACTION & COMPARISON

- Get the most defining features of an image.
- Compare features of different images.
- Most important features of identical objects will be similar.



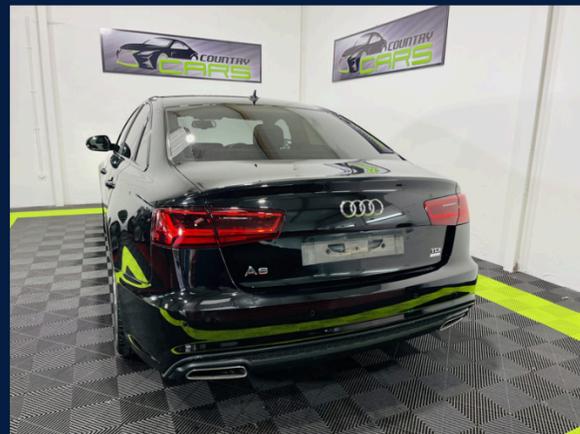
FINAL APPLICATION

1. Target object retrieval
2. Video processing
3. Comparison
4. Output saving



```
> database
> files
> models
> outputs
.gitignore
database.py
feature_comparison.py
feature_extract.py
features.py
helpers.py
main.py
metadata.json
obj_detection.py
objs.py
old_todos.txt
output.py
requirements.txt
target_obj.jpg
test.py
```

TARGET OBJECT



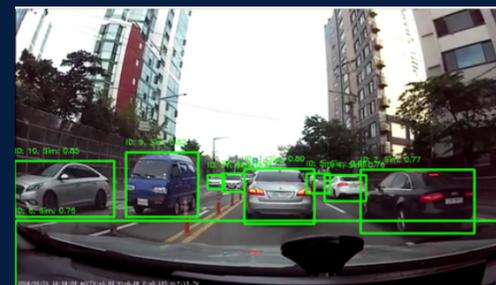
```
[[0.10, 0.07, 0.92, 0.919],  
 [0.84, 0.03, 0.44, 0.16 ],  
 [0.77, 0.79, 0.214, 0.86],  
 [0.70, 0.56, 0.11, 0.22]]
```

features



objects: id, coordinates, class_id, confidence
features: id, features

VIDEO PROCESSING



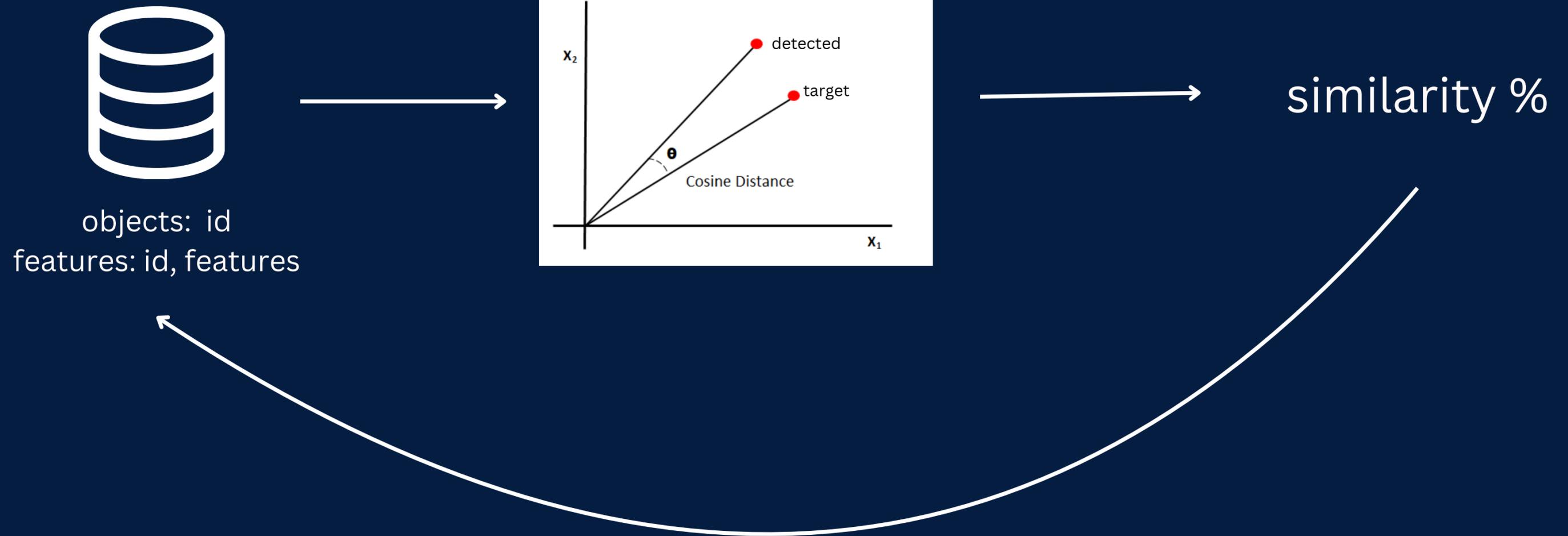
[[0.10, 0.07, 0.92, 0.919],
[0.84, 0.03, 0.44, 0.16],
[0.77, 0.79, 0.214, 0.86],
[0.70, 0.56, 0.11, 0.22]]

features for every object



objects: id, coordinates, class_id, confidence, frame_id
features: id, features

COMPARISON



OUTPUT



top 3 result ids:

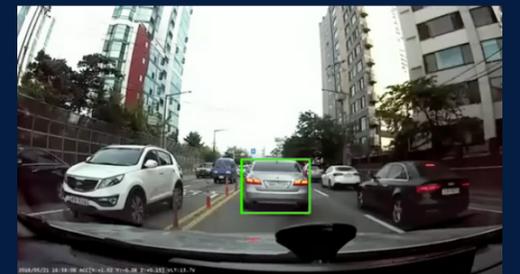
id: 2

id: 4

id: 9



output_2.mp4



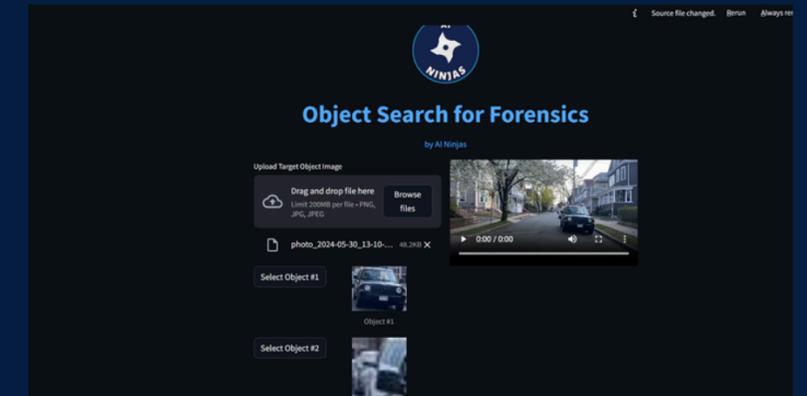
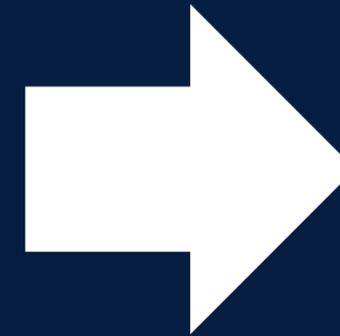
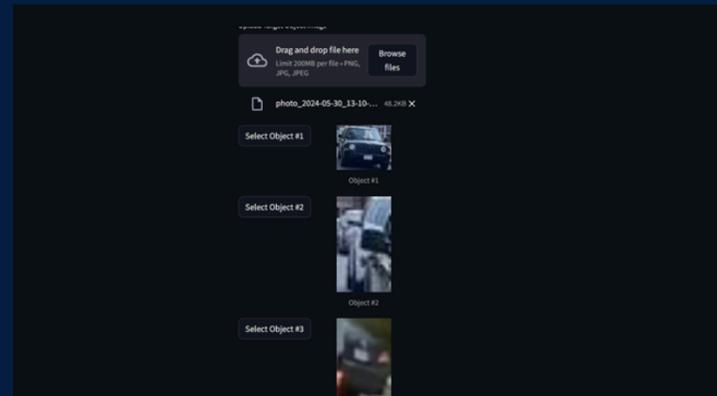
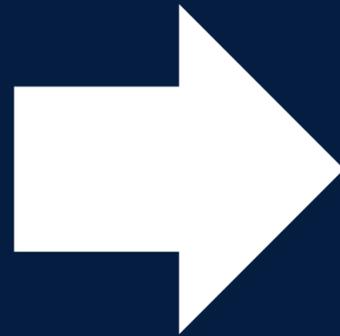
output_4.mp4



output_9.mp4

objects: id, similarity, frame_id

UI



User uploads the target object image and the search video.

Chooses the object among the ones detected on the target image.

After the set up is complete, the process runs, and the outputs appear.

RESULTS

Our system:



- Locates suspects and vehicles in videos at least twice as fast as other methods
- Saves investigators countless hours of manual review.
- Enables law enforcement to focus on solving crimes, not watching videos.
- Potential to significantly reduce crime rates through faster response times.

DEMO



Link:

<https://www.youtube.com/watch?v=QRwVBhQMG24>

NEXT SOLUTIONS

Time constraint: Several target images and videos; add more forensics classes to Yolo;

Resources: Improve feature extraction accuracy;

Limited webpage -> software application; error-handling

Modify the code for other use cases:

- Drones, Web search etc.



Q&A



Github repository of the project:
<https://github.com/Ikromov247/forensicsUI>