Pedestrian Detection summary

G.Giannoulis, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 2.6.1

Artificial Intelligence &
Information Analysis Lab

Pedestrian Detection

- Handling Pedestrians in self-driving cars
  - Self-driving car architecture
  - Pedestrian's state estimator
  - Crosswalk state estimator
- CNN pedestrian detection and tracking
  - Pedestrian detection
  - Pedestrian Tracking
- Pedestrian Trajectory Estimation

Part 1: Handling Pedestrians in self-driving cars

Introduction

- The development of intelligent autonomous cars is of great interest.

- A particular and challenging problem is to handle pedestrians, for example.
  - Crossing or walking along the road.
  - In cases that a pedestrian is standing on the pavement near a crosswalk without intending to cross it.

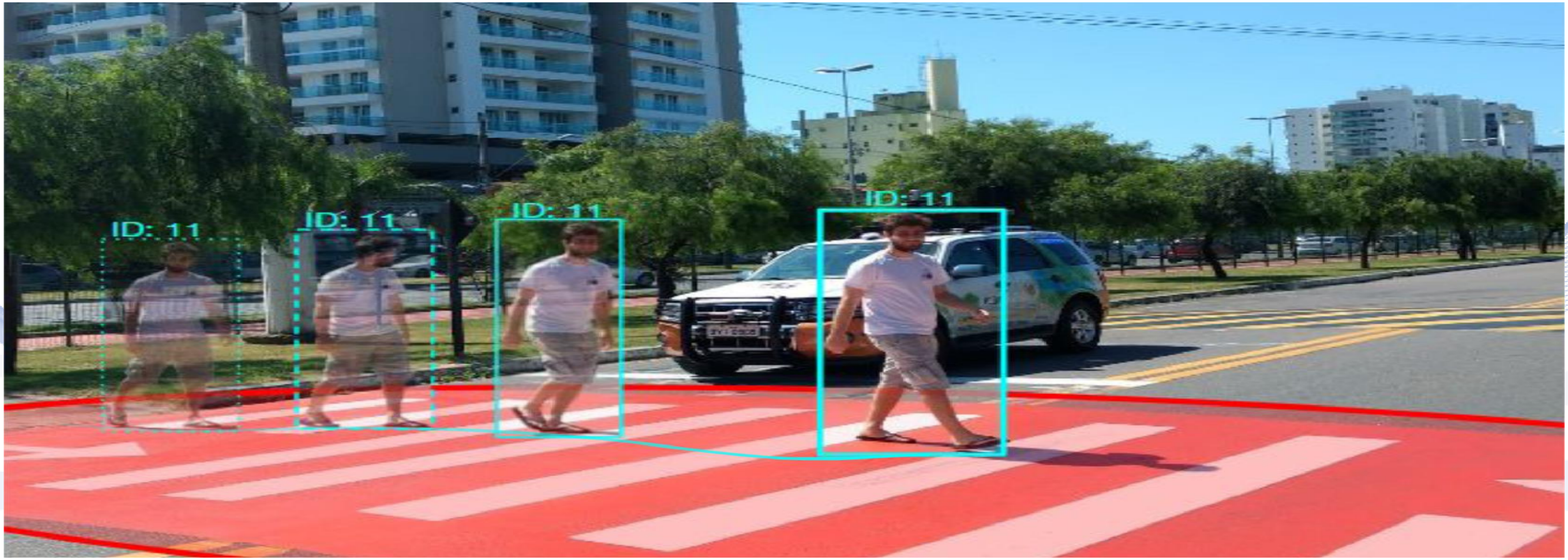
Introduction

- Self-driving cars rely on several sensors ,LiDAR, radar, or cameras and with the pedestrian’s position, velocity, and orientation can facilitate the behavior selection decision making.[1]

- This work improves on current pedestrian handling systems by incorporating a pedestrian tracking system, which aims to predict the near-future pedestrians’ states.[11]

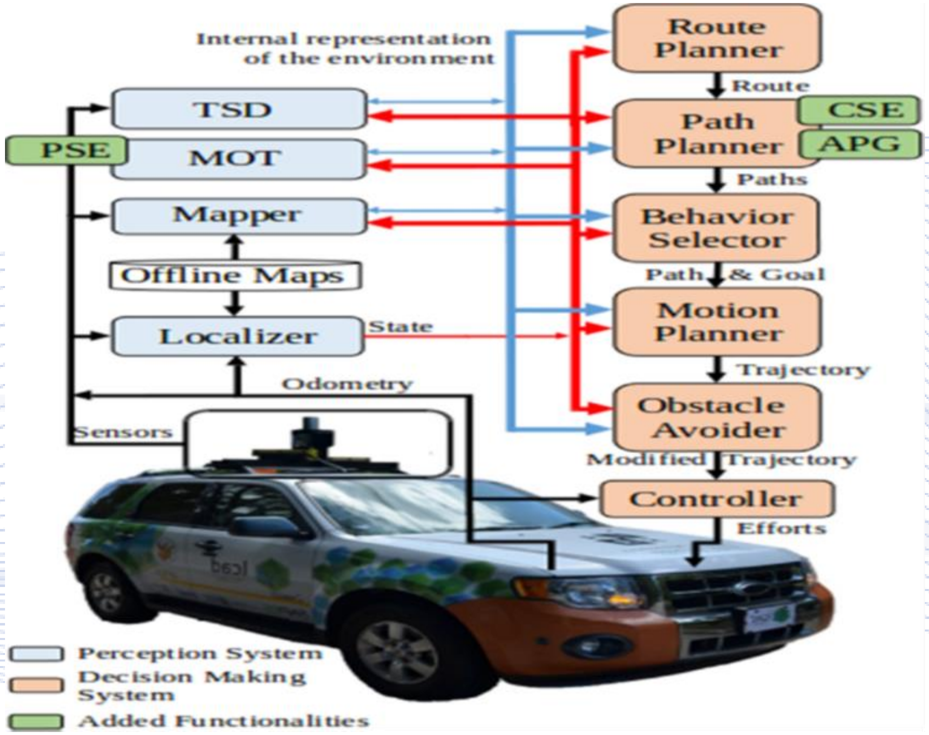
Introduction

- The self-driving car handling a pedestrian in a crosswalk.



Self-driving car architecture

- The perception and decision making systems are shown as a collection of modules of different colors.



The Perception System

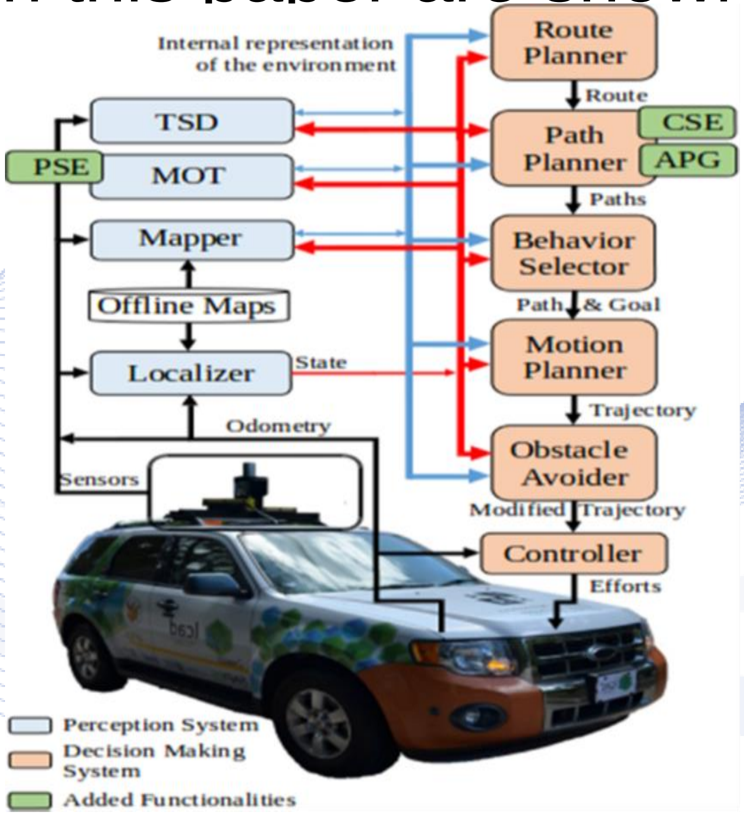
- The Mapper module constructs occupancy grid maps that represent obstacles of the environment, operates in [28]:
  - Offline mode :
    - Receives as input data from multiple sensors,odometer, LiDAR, IMU, and GNSS.
    - Estimates the self-driving-car’s states along the path and the values of the offline map cells.
  - Online mode:
    - Receives as input the offline map, the sensors’ data ,odometer, LiDARs, and IMU and the car’s state, and computes the online map.

The Decision-Making system

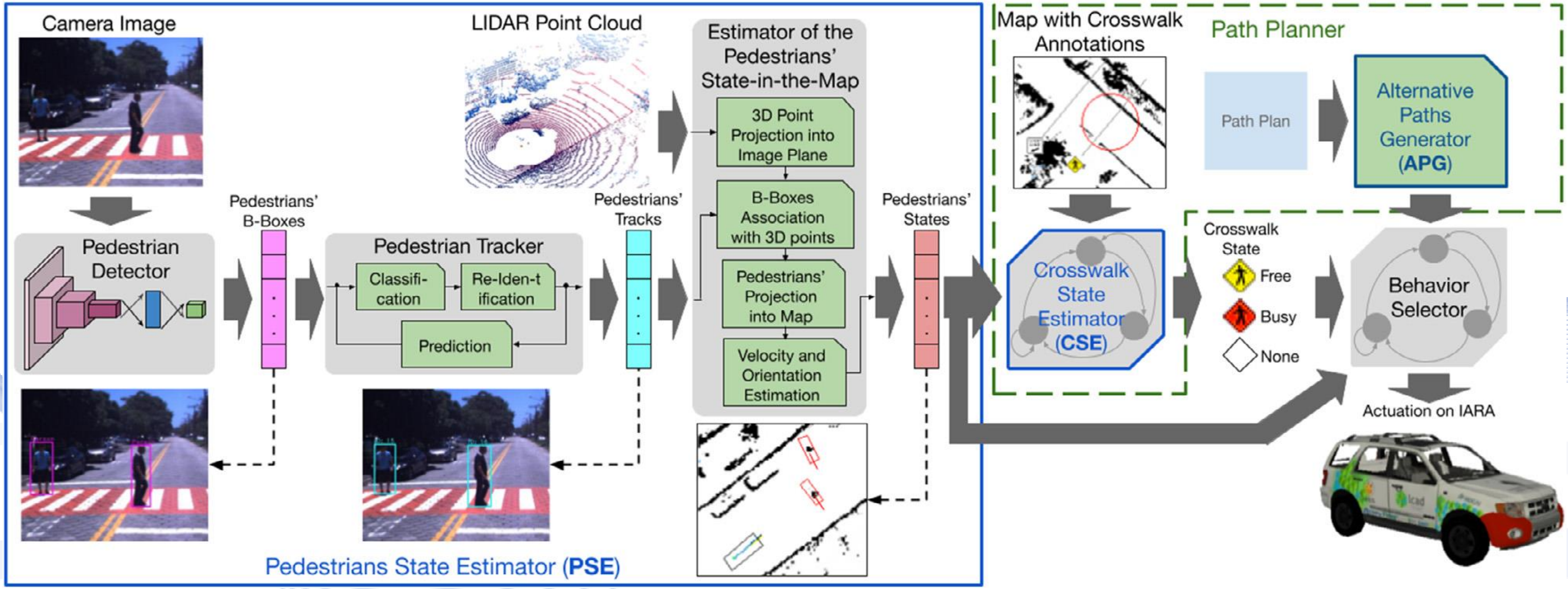
- The Path Planner creates a path from the current state of the self-driving car to a local goal state.
  - Receives the car's state and a Road Definition Data File (**RDDF**) file as input.
    - RDDF is composed of a sequence of car's states.
    - Extracts a path from the RDDF, which consists from the current state of the car to a goal state a few meters ahead.

Additions functions

- The added functionalities presented in this paper are shown in green:
  - **PSE,**is part of the MOT module and provides Pedestrians' States Estimator.
  - **CSE,**is part of the Path Planner module and provides the Crosswalk State Estiamtor.
  - **APG,** part of the Path Planner module and provides Alternative Paths.



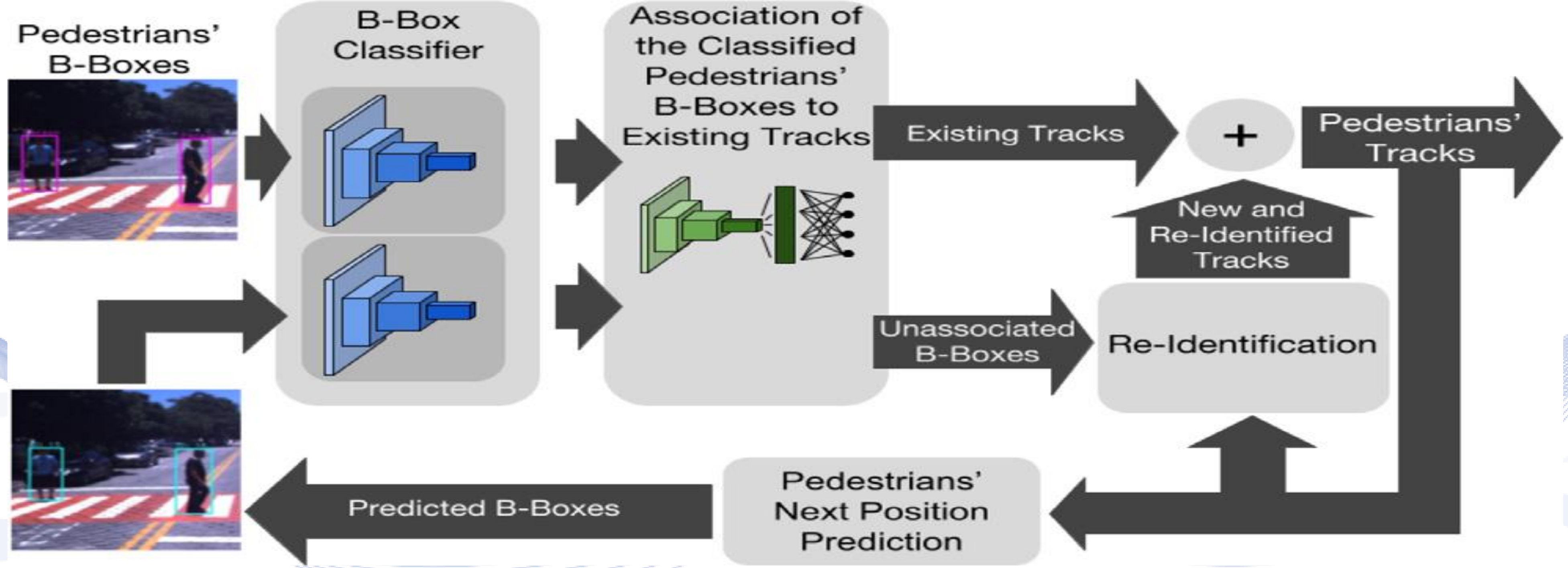
Pedestrian's state estimator



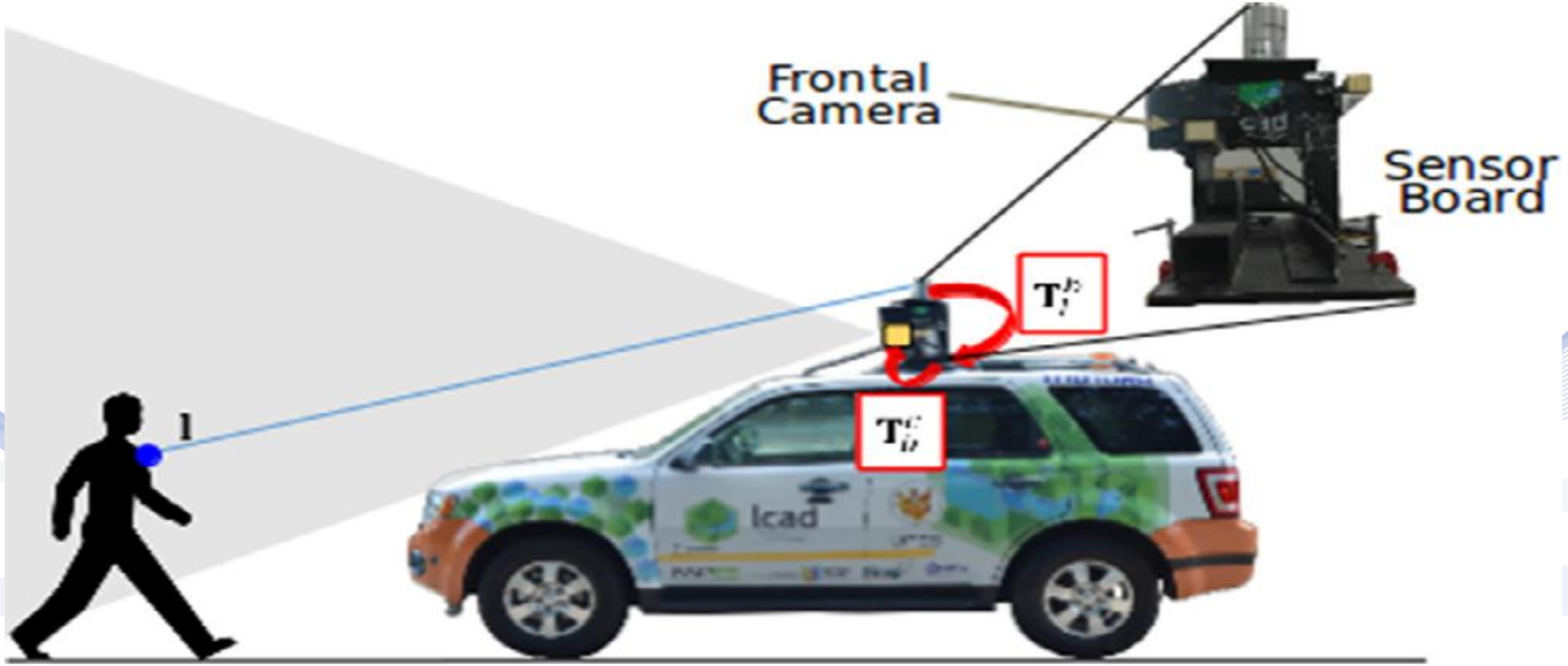
Artificial Intelligence & Information Analysis Lab

Pedestrian's state estimator

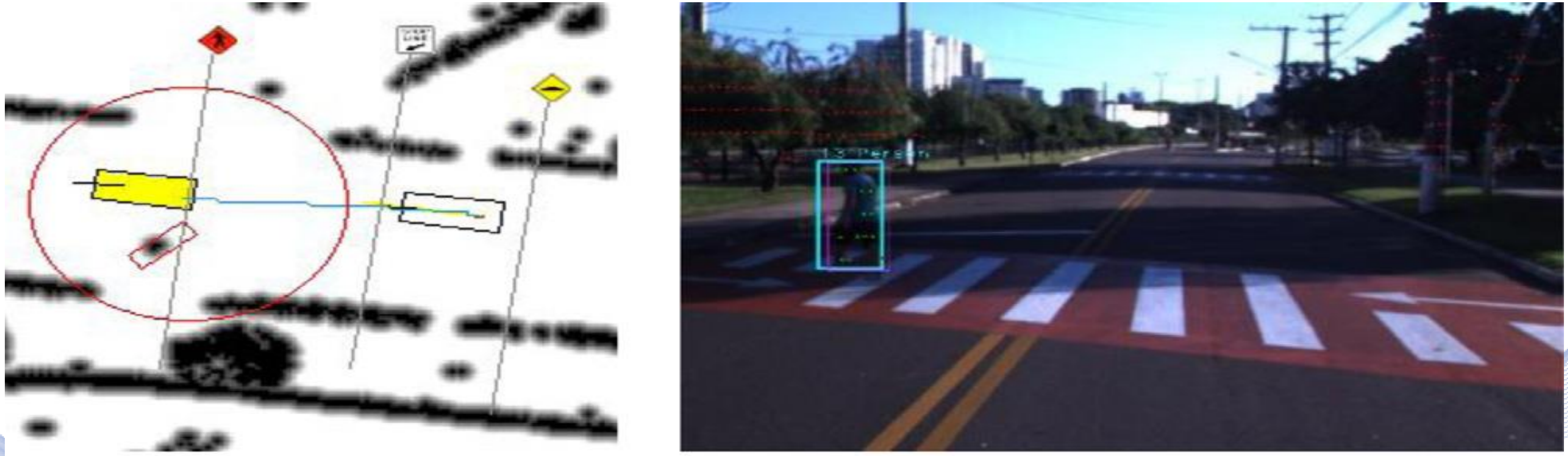
- Pedestrian tracking system.



Pedestrian's state estimator



Crosswalk state estimator



(a) (b)

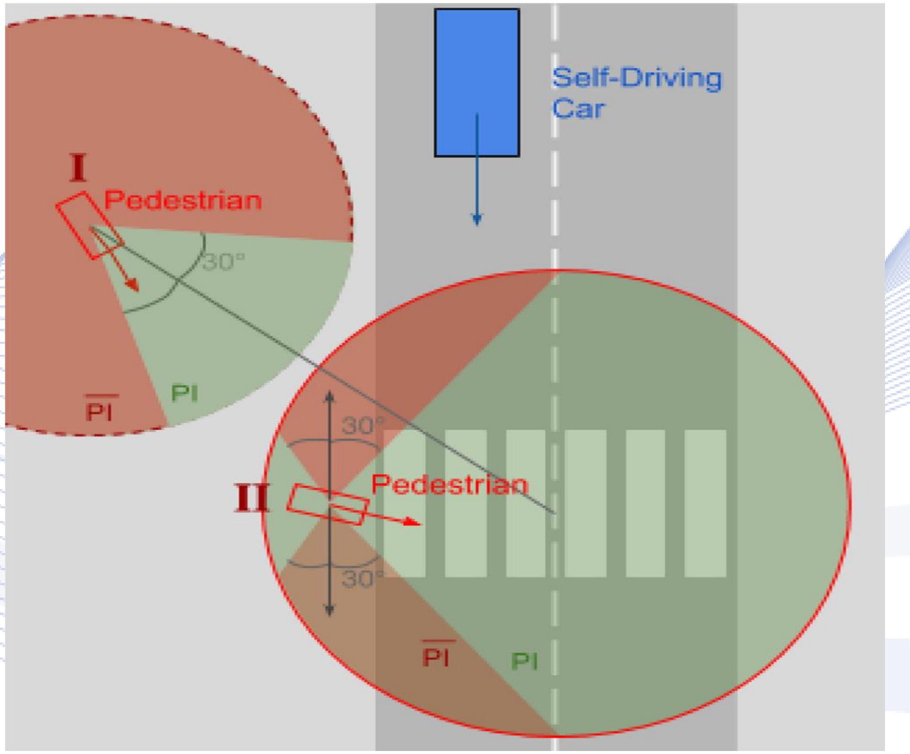
a) A crosswalk it is represented as the red circle.

b) The detection of the pedestrian in the map in the camera image.

Crosswalk state estimator

- Scenario I (dashed circle) when the pedestrian is outside the crossing area (solid circle):

  - PI is true if the movement orientation is within ±30 ◦ of the line that connects the pedestrian to the centre of the crosswalk.

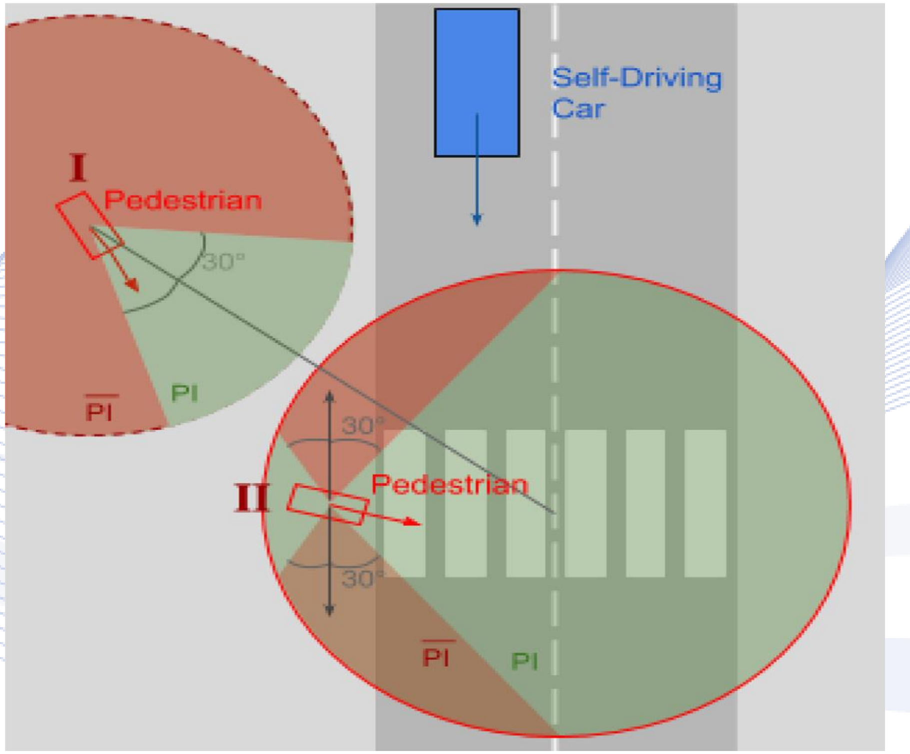


Artificial Intelligence & Information Analysis Lab

Crosswalk state estimator

- Scenario II when the pedestrian is inside the crossing area (solid circle):

  - PI is true if :(i) the pedestrian is in the car's path or (ii) if velocity is equal or greater than 0.3 m/s and the pedestrian is not moving in the approximate direction (±30 ◦) of the road,not moving parallel to the road.



Alternative Paths Generator

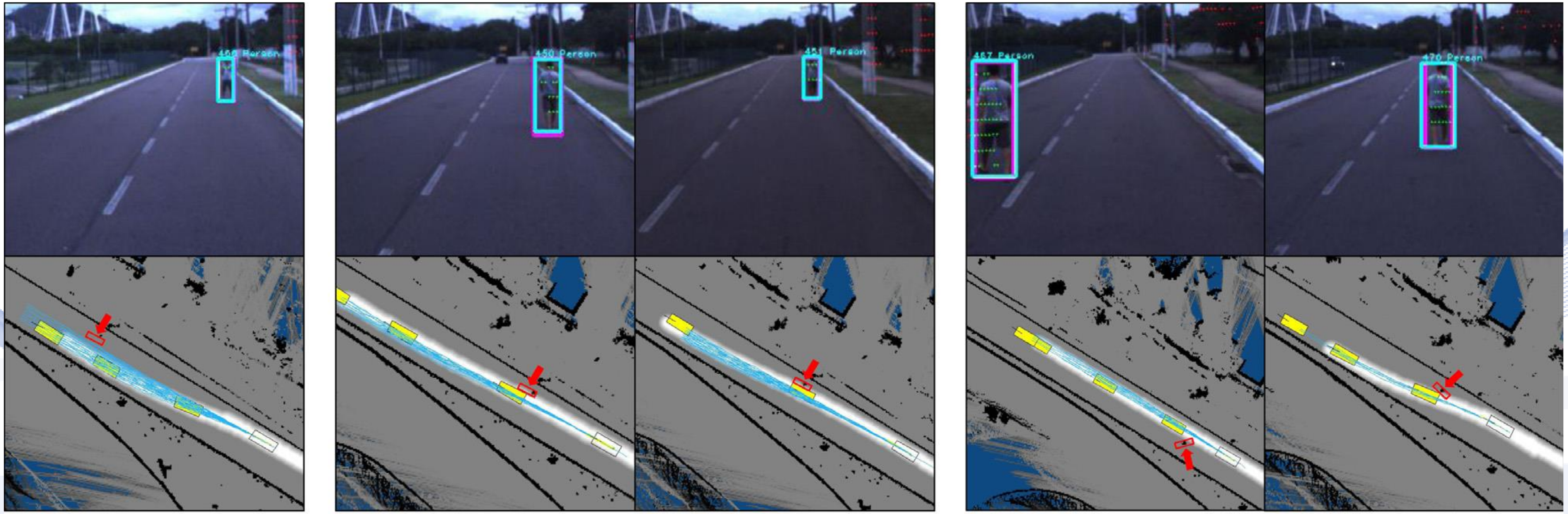
- (**APG**) Provides many alternatives of paths, are computed using Frenét Frames.[12]

  - If there are pedestrians in the road near the path of the car, it has to select another path that is at a safe distance of the pedestrians.

  - Frenét Frames in tracking theory for modeling the path generation as the problem of generating a sequence of moving frames using a frame as reference.

Results

- The real-world experiments near crosswalks.



(a) (b) (c)

Artificial Intelligence & Information Analysis Lab

Part 2: CNN pedestrian detection and tracking

Artificial Intelligence &
Information Analysis Lab

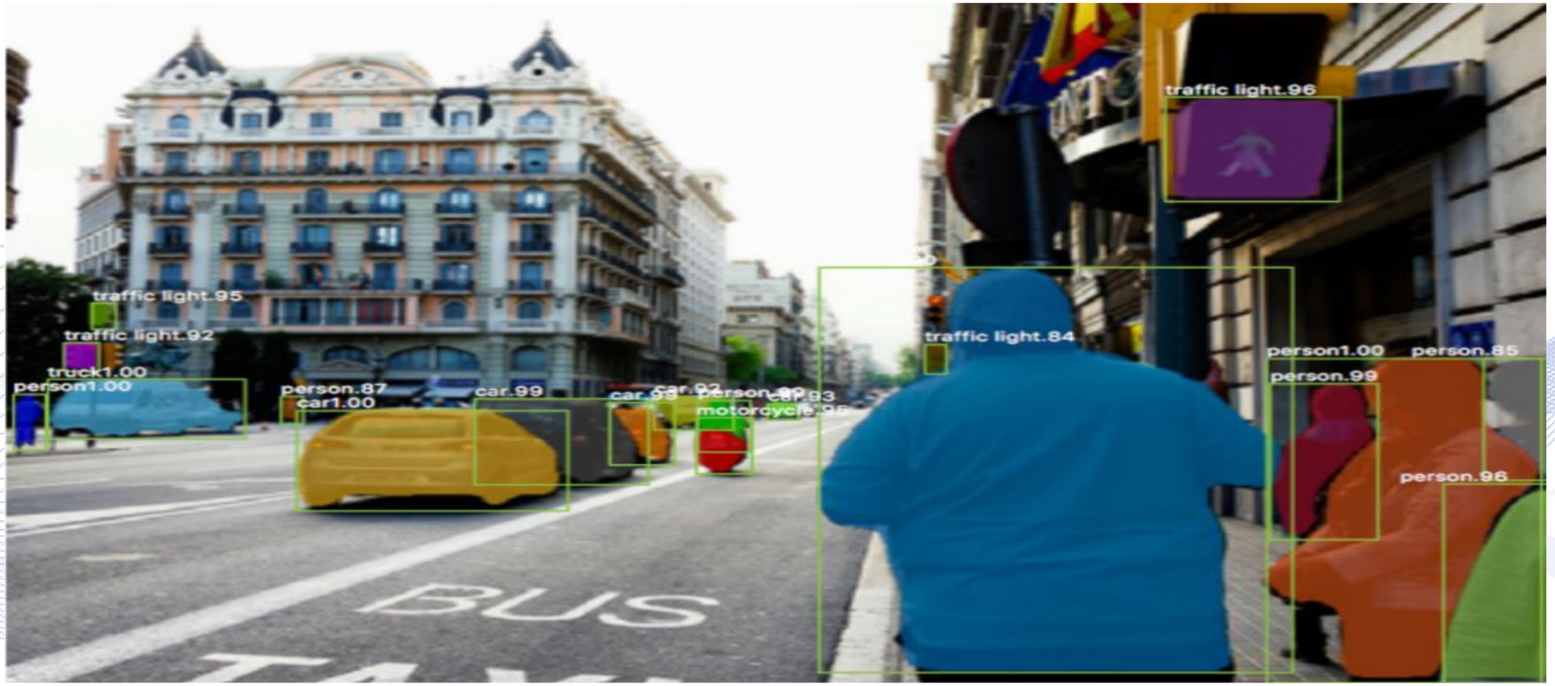
Faster R-CNN

- The Region Proposal Network works by passing a sliding window over the CNN feature map and at each window, outputting **k** potential bounding boxes and scores for how good each of those boxes is expected to be.

[1]

Mask R-CNN

- Faster R-CNN works so well for object detection,we extend it to also carry out pixel level segmentation [1,7].



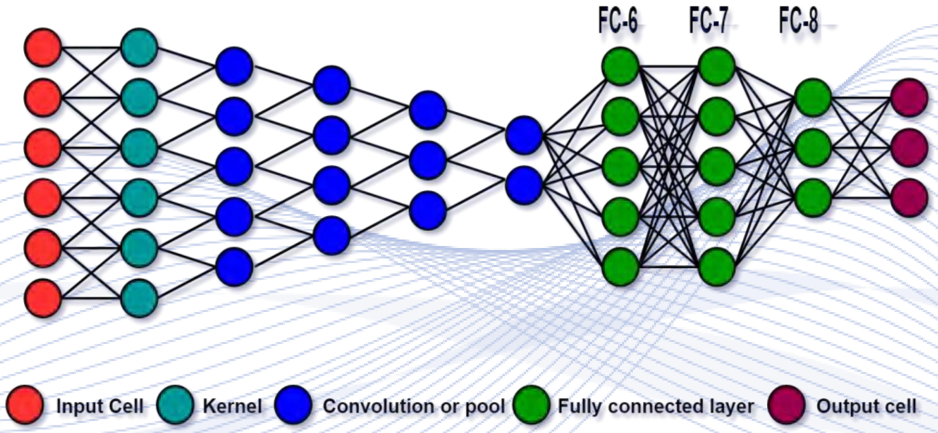
RoiAlign

- Combines the classifications and bounding boxes from Faster R-CNN to generate segmentations[1]:

[1]

Pedestrians Tracking

- The developed framework uses CNNs both to detect pedestrians within the frames and track across the frames.
- A simplied architecture of the deep CNN:



[1]

- FC-8 layer is the last layer before the classification, from which the features are extracted.

Evaluation of the detector and the tracker

- First 30 seconds of the video was used at a reduced frame rate of 8 frames per second:

(a) (b)

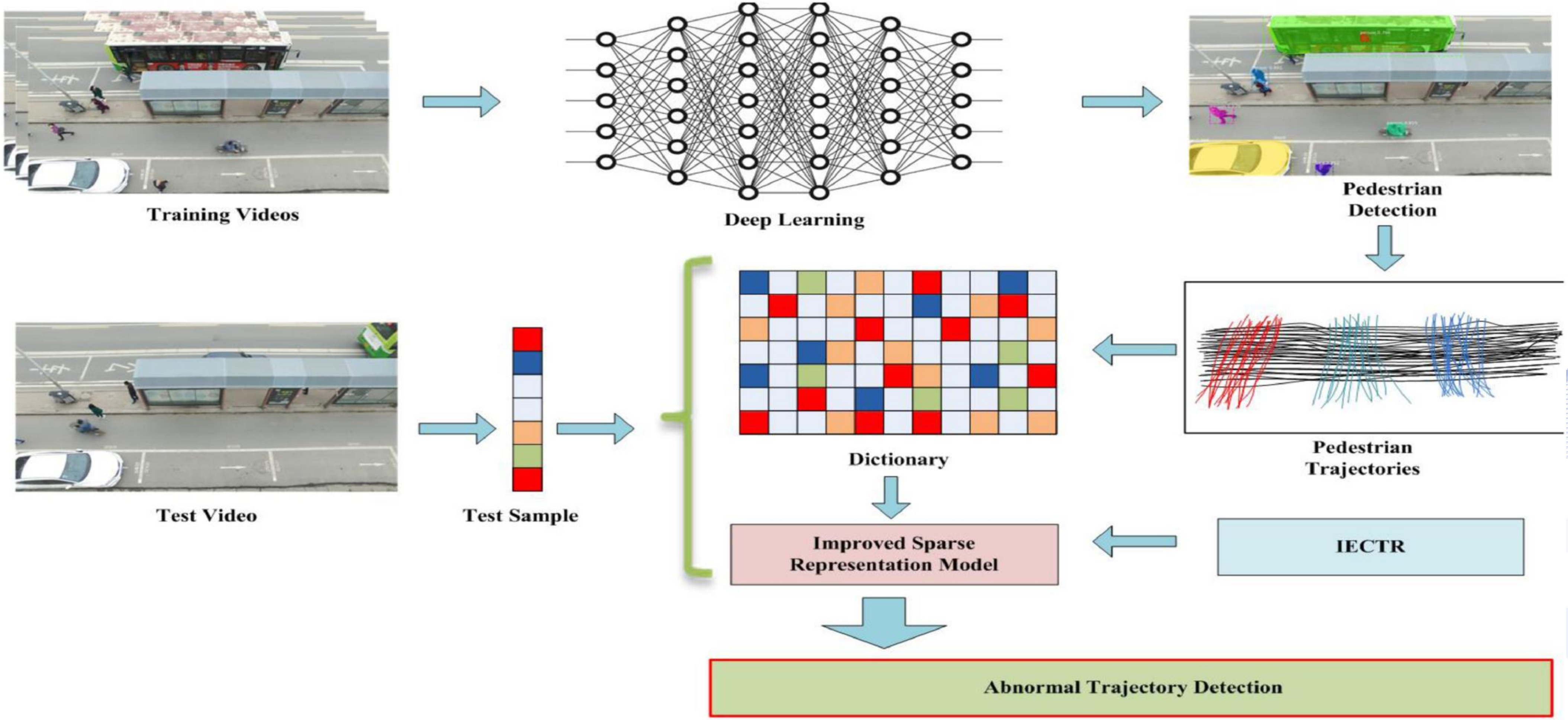
[1]

- Shows the detections in video sequence that are 8 frames.
- Shows the tracking results, the number denoting each pedestrian is generated randomly in the first frame.

Part 3: Pedestrian Trajectory Estimation

Artificial Intelligence & Information Analysis Lab

Experimental results



Experimental results

Pedestrian detection for Bus station dataset1.

Experimental results

(a) the normal trajectory , (b) the abnormal trajectory.

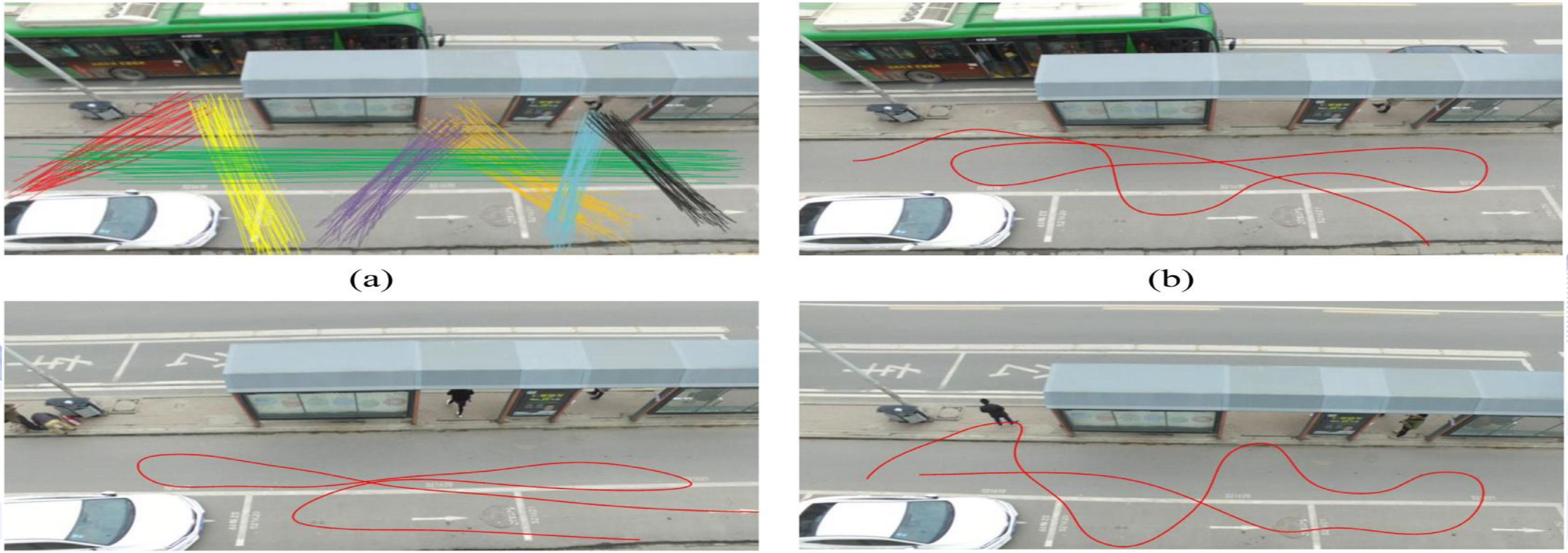
(a) (b)

Experimental results

- Pedestrian detection for Bus station dataset2

Experimental results

Normal and abnormal trajectory in Bus station dataset2.



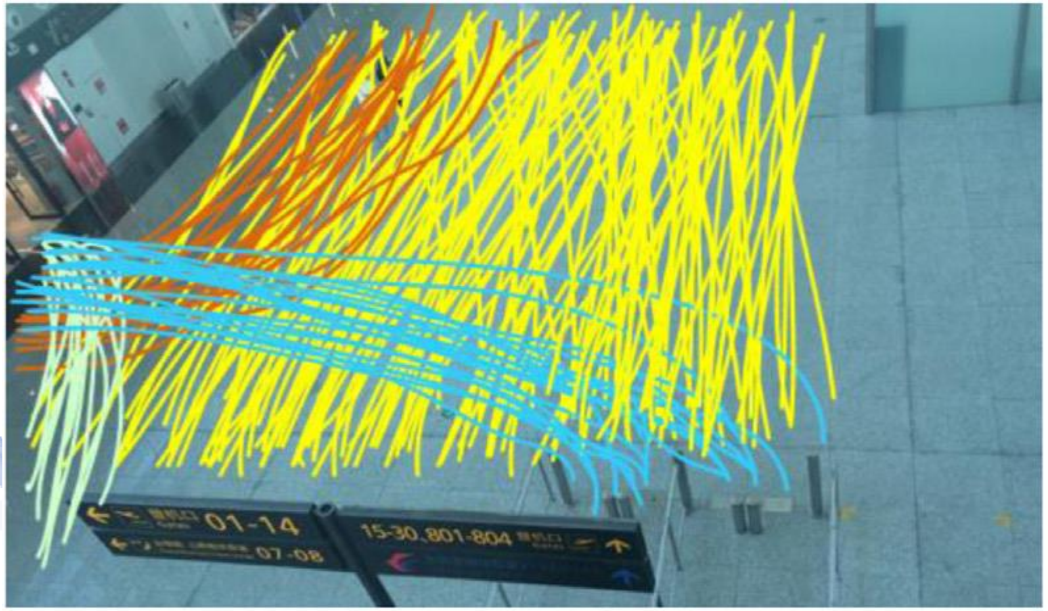
(a) (b) (c) (d)

Experimental results

- Pedestrian detection for Airport.

Experimental results

(a) normal trajectory



(a)

(b) the abnormal trajectory

(b)

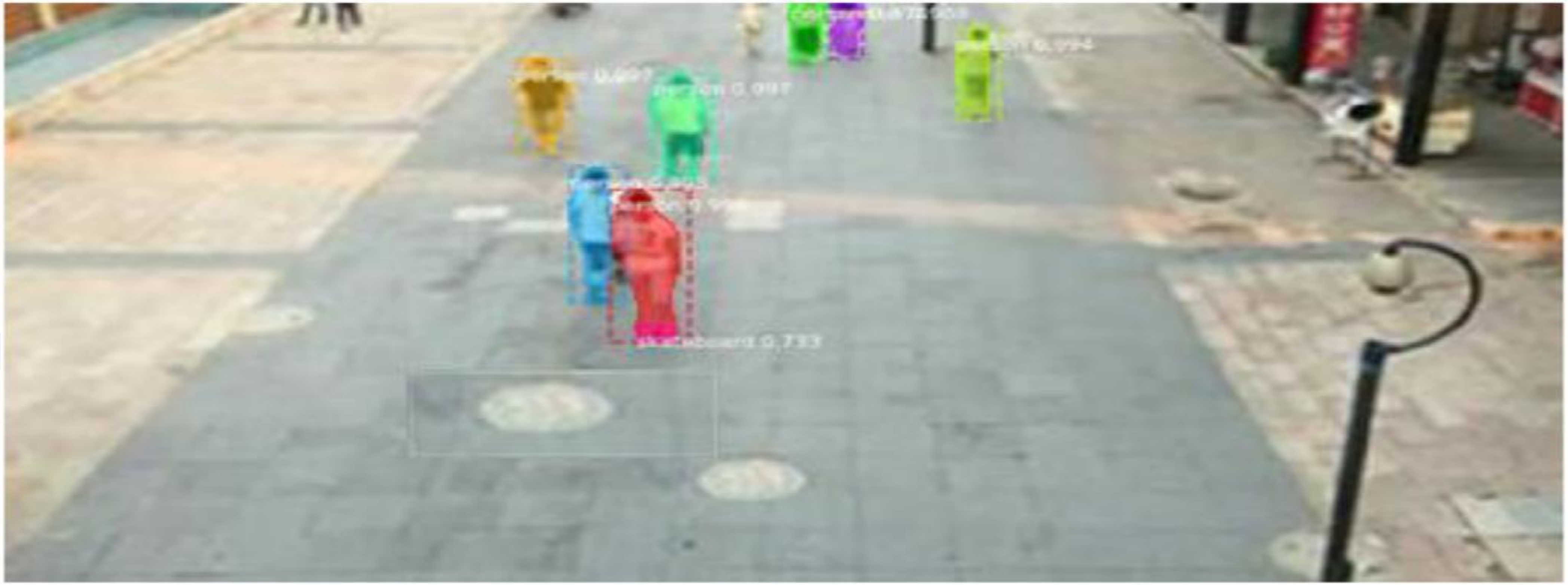
Experimental results

Table 6
Results of pedestrian trajectory identification on Airport dataset.

Methods	Accuracy of normal trajectory understanding				Accuracy of abnormal trajectory understanding	
	N1	N2	N3	N4	TPR	FPR
NBC	74.3%	77.6%	74.3%	80.3%	31.4%	1.6%
SVM	73.3%	84.2%	79.3%	77%	31.4%	1.6%
kNN	79.6%	82%	80%	81.3%	42.9%	1.5%
SR-l^1	81.6%	83.6%	81.3%	83%	82.8%	0.4%
SR-l^2	80.3%	82.3%	79.6%	80%	77.1%	0.6%
SR-$l^{0.5}$	85.6%	86.3%	88.3%	85%	85.7%	0.3%
Our method	89.6%	86.3%	89.3%	87.6%	88.6%	0.2%

Experimental results

- Pedestrian detection for Square dataset.



Experimental results

- Normal and abnormal trajectory classes in Square.

(a) (b) (c) (d)

Experimental results

Table 7
Results of pedestrian normal trajectory identification on Square dataset.

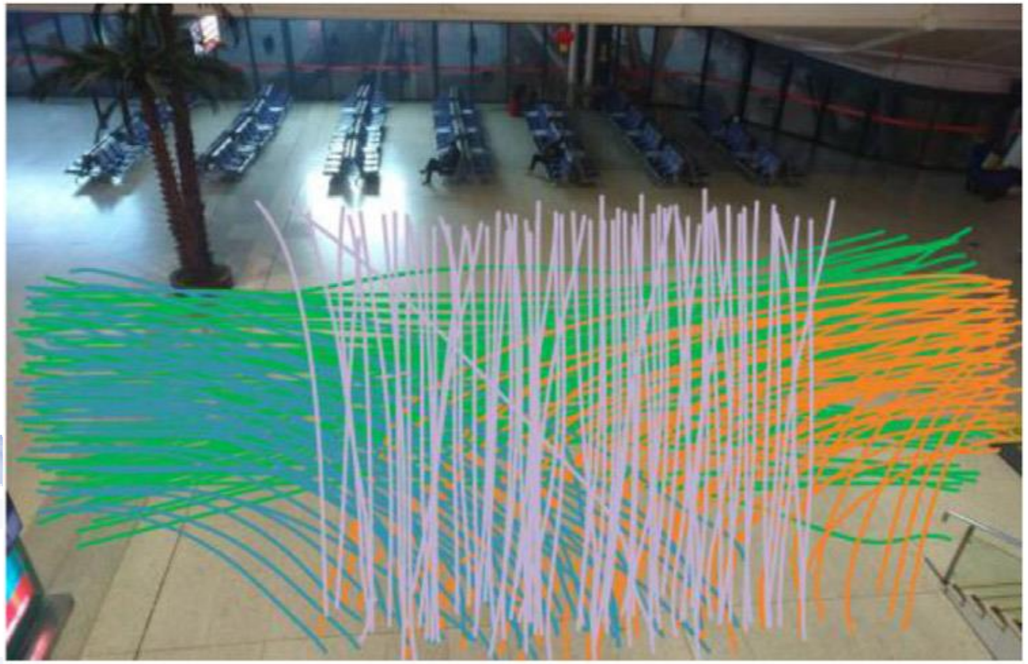
Methods	Accuracy of normal trajectory understanding			Accuracy of abnormal trajectory understanding	
	N1	N2	N3	TPR	FPR
NBC	77.5%	79.5%	78.8%	47.5%	1.6%
SVM	80.8%	77.8%	81.3%	42.5%	1.7%
kNN	82%	80.3%	79.3%	45%	1.6%
SR-l^1	81.5%	82.8%	82.3%	82.5%	0.4%
SR-l^2	79.8%	82%	80.8%	80%	0.6%
SR-$l^{0.5}$	83.7%	82.8%	80.7%	85%	0.3%
Our method	88%	88.3%	89.2%	90%	0.2%

Experimental results

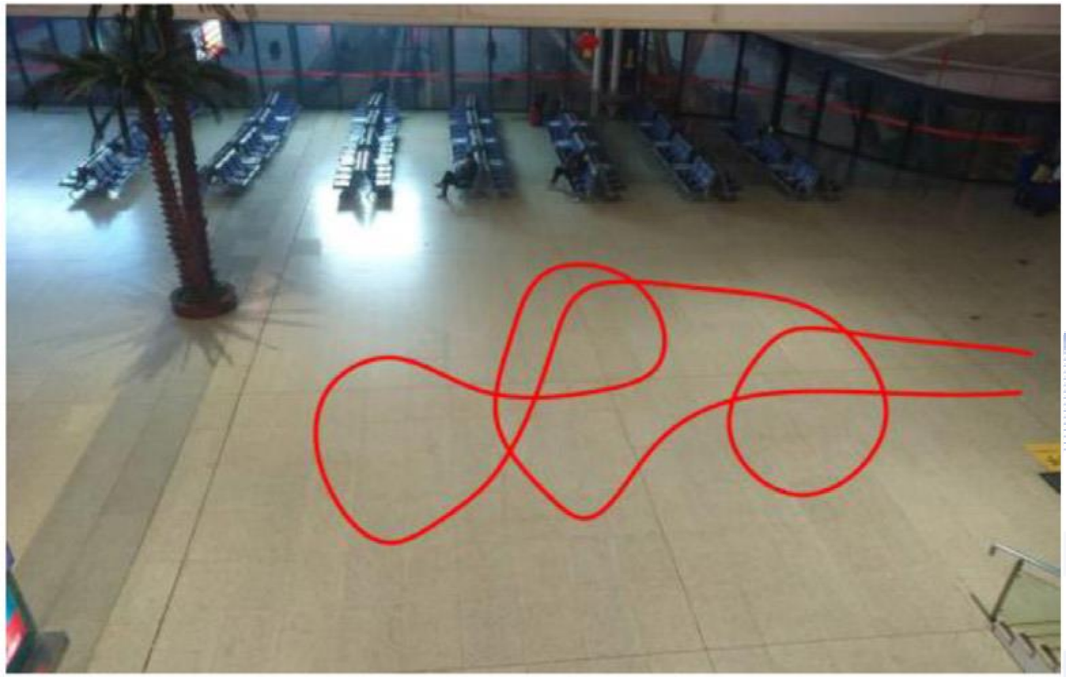
- Pedestrian detection for Railway station dataset

Experimental results

(a) Normal trajectory (b) Abnormal trajectory



(a)



(b)

Artificial Intelligence & Information Analysis Lab

Experimental results

Table 8
Results of pedestrian trajectory identification on Railway station dataset.

Methods	Accuracy of normal trajectory understanding				Accuracy of abnormal trajectory understanding	
	N1	N2	N3	N4	TPR	FPR
NBC	74.5%	71.9%	76%	76.9%	32.5%	1.4%
SVM	77.5%	74.3%	75.2%	79.2%	31.3%	1.5%
kNN	81%	78.1%	79.6%	83.1%	52.5%	1.2%
SR-l^1	76.5%	76%	82%	81.5%	73.8%	0.9%
SR-l^2	78%	76.8%	78%	80.8%	61.3%	1.2%
SR-$l^{0.5}$	80.5%	81.3%	83.2%	84.6%	75%	1.1%
Our method	82.5%	84.4%	87.2%	83.1%	78.8%	0.9%

Bibliography

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2000] I. Pitas, Digital Image Processing Algorithms and Applications, J. Wiley, 2000.
[SZE2011] R.Szelinski, “ Computer Vision ” , Springer 2011
[HAR2003] Hartley R, Zisserman A. , “ Multiple view geometry in computer vision” . Cambridge university press; 2003.
[DAV2017] Davies, E. Roy. “Computer vision: principles, algorithms, applications, learning ”. Academic Press, 2017
[TRU1998] Trucco E, Verri A. “Introductory techniques for 3-D computer vision”, Prentice Hall, 1998.
[PIT2017] I. Pitas, “Digital video processing and analysis ” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television ” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, 3D Image Processing Algorithms, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr