INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Université Fédérale Toulouse Midi-Pyrénées

Pedestrians mobility analysis

Mathilde Cornille - 4 AE-SE

INSA Toulouse, Department of Electrical and Computer Engineering

135 avenue de Rangueil, 31400 Toulouse

Internship - from 17/06/19 to 27/09/19

CONTEXT | OBJECTIVES | MATERIAL & DEPLOYMENT

ACTIA® FONDATION INSA TOULOUSE

ACTIA intelligent mobility chair at INSA, created in March 2019

Campus mobility analysis: Understanding pedestrians moves over the campus in order to arrange urban spaces in a smart way.

Analyse pedestrians density and flow over the campus.

- Provide a local video processing
- Send only metadata over internet using oneM2M standard
- Provide a simple web application to show the results

Choice of the material

Gateway : Nvidia Jetson Nano (Nvidia Maxwell GPU 128 Nvidia CUDA cores, Quad-Core ARM Cortex-A57 Processor, 4GB 64-bits Memory)

Camera : Raspberry Pi v.2

Large-scale deployment

Interoperability & scalability : New sensors can be added to measure other types of data.

Adaptability : Video processing is adaptable to the observation area during settings, so we can deploy other cameras using the same algorithm over the campus.

CONTRIBUTION

1 PEDESTRIANS DETECTION ON VIDEO FRAME

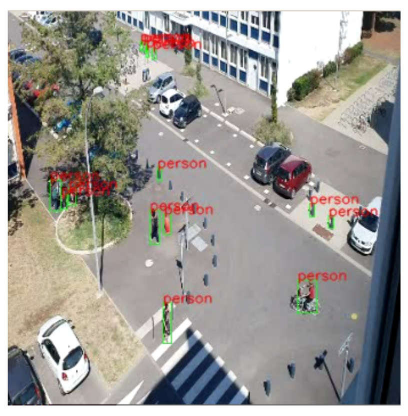
YOLOv3 (You Only Look Once) – OPENCV – Python

Principle: Computer vision CNN algorithm working by regression.

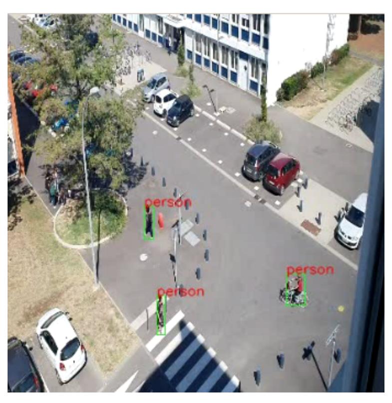
- Applying a single neural network on the full rescaled image (416x416).
- Predicting bounding boxes and object probabilities.
- Using Non-Maximal Suppression (NMS) technique to remove duplicated detections of a same object.

Characteristics: Unlike other algorithms it runs the input image only once through the Darknet deep neural network → Quickest computer vision algorithm for real-time detection with quite good level of accuracy but some false positives.

Solution: Taking into account the size of detected objects (area of pedestrian bounding box should be between 100 and 500 pixels).



False positives with YOLOv3 detection



Detection + false positives management

2 PEDESTRIANS TRACKING AND METADATA CALCULATION

Python

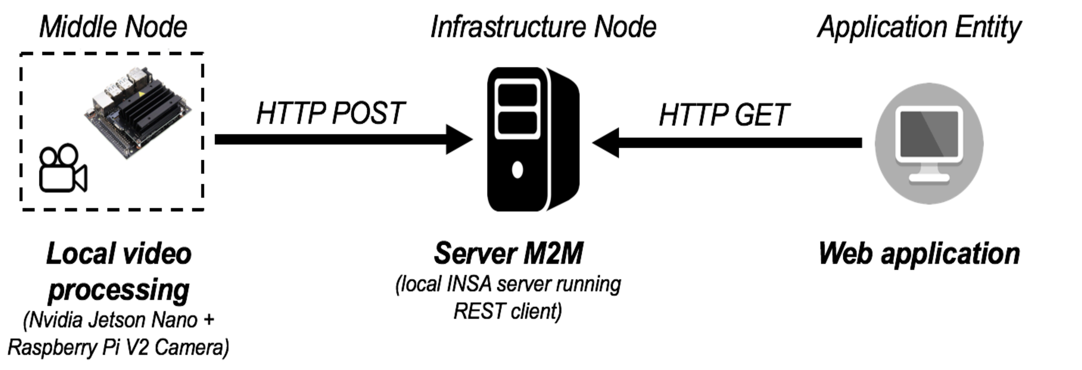
Flow calculation is done with **tracking of detected pedestrians**.

Settings of the camera : determine on first frame the limits from where people can enter and leave the area.

Calculation of the flow : count entrances and exits of tracked pedestrians when they cross one of the limits. Data is reset every twenty minutes.

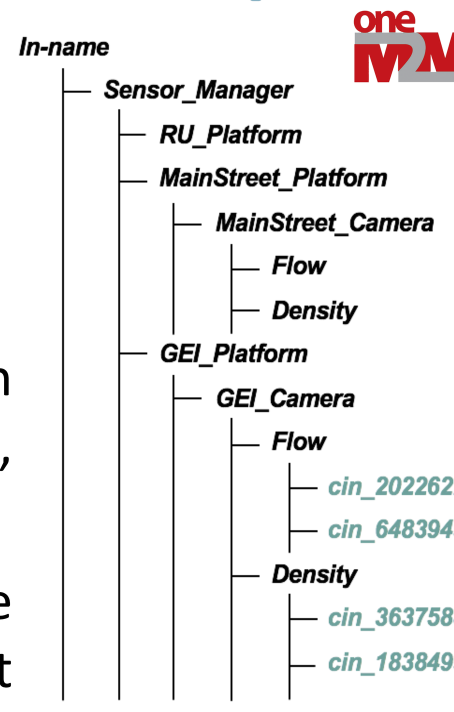
3 METADATA SENDING TO M2M SERVER

oneM2M standard – HTTP – C++ – Python



Density data : Content Instance (CIN) with number of pedestrians in the observation area, sent with HTTP POST each time its changes.

Flow data : Content Instance (CIN) with the number of entrances and exits for each limit determined during settings of the camera, sent with HTTP POST every twenty minutes.



Resource tree of data stored in oneM2M

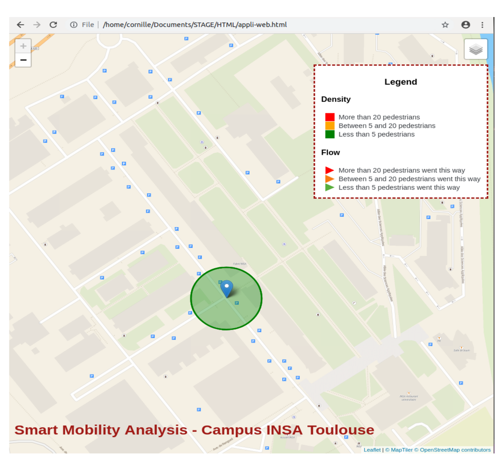
Interoperability and Scalability

4 RESULTS : WEB APPLICATION DEPLOYMENT

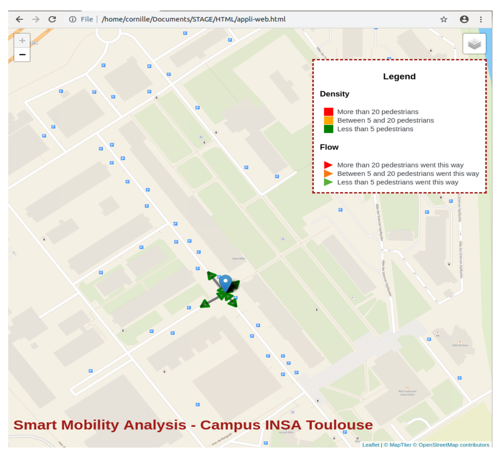
HTML5 – JavaScript – Leaflet API

Aim : Show density and flow information on a map with intuitive markers and colours (green, orange, red).

➢ Metadata collection from OM2M platform with HTTP GET when refreshing the webpage.



Density map layer



Flow map layer

CONCLUSION

Scientific and technical conclusions

Scientific results : Good expendable solution respecting privacy. GPU issues making analysis really slow.
Improvements: Run on Nvidia GPU for real-time analysis. Better representation of flow on the map.
New technical skills developed: OpenCV, video processing, web programming.

Personal conclusion

Great experience on **Internet of Things** with a concrete application.
Autonomy, initiative taking, co-working

*Internship tutored by **Thierry Monteil***