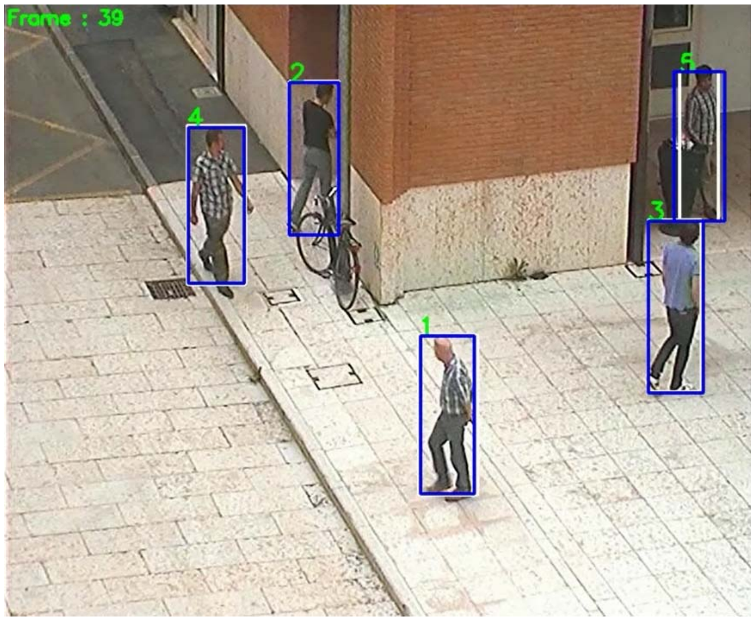
PEDESTRIAN ANALYZING

Pedestrian analyzing is an application able to track pedestrian in a video, and then analyze their activity. It also creates motion heat map for user reference. It is easy to use this application via REST API and user interface (UI).



Features

- Pedestrian tracking
- Activities analyzing
- Motion heat map
- REST API and WEB GUI
- Deployment and scalable

Pedestrian tracking

Pedestrian tracking is the most important part of this application. It follows tracking-by-detection paradigm with the following steps:

- Detection stage: Object detection algorithm analyzes each input frame to localize pedestrians by bounding boxes (detections).
    - Regarding to model's accuracy and FPS, YOLOv3 model was chosen for this application.
    - Although standard YOLOv3 model is good enough for the application, ***re-trained YOLOv3 model*** (with customized dataset of pedestrians) was implemented in this stage to improve detection accuracy, thanks to Darknet framework. Re-train process should follow instructions in Darknet project with some notes:
        - Split labeled data into train and test set, and create files: train.txt and test.txt.
        - Update configuration files (yolov3.cfg, obj.names) as instruction.
        - Change number of batch if encountering memory issues during training process.

- Feature extraction: Feature extraction algorithm analyzes and extracts pedestrian features from the detections
    - ***Re-trained RESNET50 model***, reference to “Person re_ID baseline Pytorch”, was implemented to extract pedestrian features.
- Tracking stage: tracking algorithm uses features, motion predictions to compute a similarity (cosine) distance/score between pairs of detections and trackers, then associates detections and trackers belonging to the same target by assigning the same ID.
    - Deep SORT is the tracking algorithm of this application.
    - Deep SORT algorithm was modified to make it more robust in assigning ID to trackers, and able to do ***pedestrian re-identification (re-ID)***: assign the same ID to the same pedestrians who disappeared in some video frames (occlusion, out of camera range, etc.) and reappeared.
        - Pedestrian re-ID ***improves the ID switching issue*** in MOT (Multi-Object Tracking).

Activity analyzing

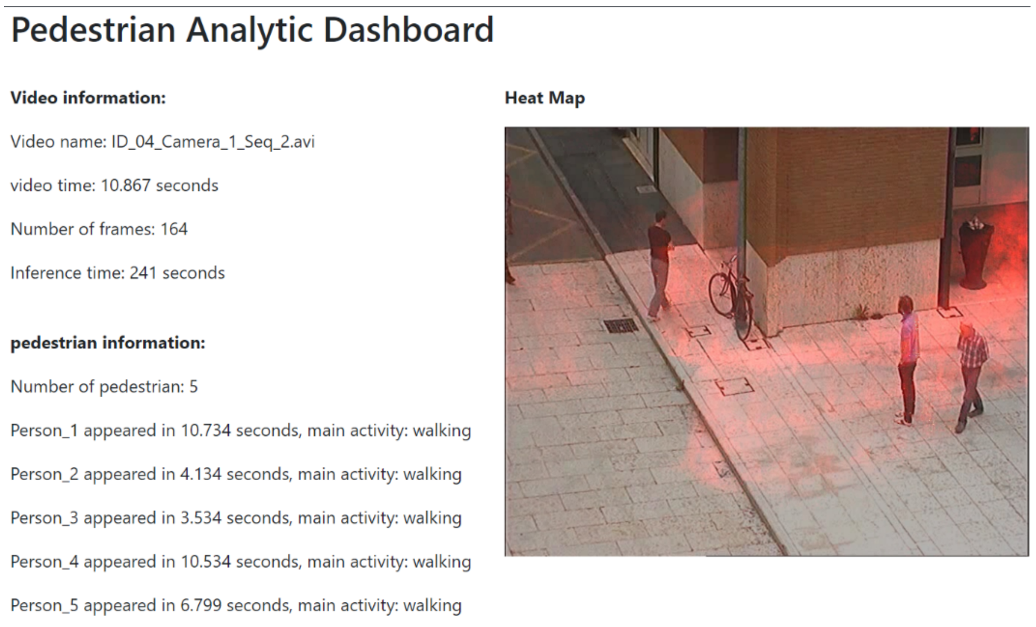
Activities are generated by analyzing the movement and change of the tracked bounding boxes. The activities of a pedestrian can be recognized as: sitting, standing, slow walking, walking and running. In this application, the thresholds differentiate these activities are based on some experiments. It should be customized in specific application as user’s definition.

Motion heatmap

Motion heatmap was created to see movement pattern. It is useful in some specific application and/or for further investigation.

REST API and GUI

User is able to access the application via WEB GUI: Pedestrian analytic UI allows user to upload video for tracking and analyzing pedestrian activities, and dashboard UI shows the summary results and motion heatmap.


Pedestrian Analytic Dashboard

Video information:

Video name: ID_04_Camera_1_Seq_2.avi

video time: 10.867 seconds

Number of frames: 164

Inference time: 241 seconds

pedestrian information:

Number of pedestrian: 5

Person_1 appeared in 10.734 seconds, main activity: walking

Person_2 appeared in 4.134 seconds, main activity: walking

Person_3 appeared in 3.534 seconds, main activity: walking

Person_4 appeared in 10.534 seconds, main activity: walking

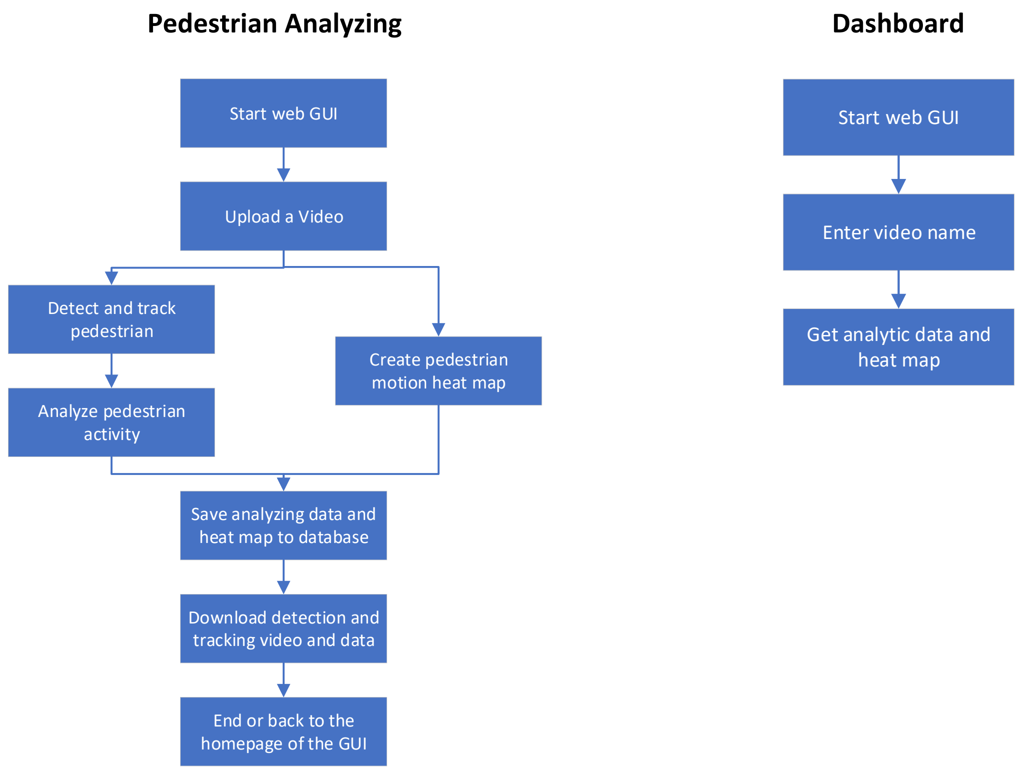
Person_5 appeared in 6.799 seconds, main activity: walking

Heat Map

Deployment and scalability

Containerized (docker container) application makes it easy to deploy and scale.

Application Workflow:


Pedestrian Analyzing
Start web GUI
Upload a Video
Detect and track pedestrian
Analyze pedestrian activity
Create pedestrian motion heat map
Save analyzing data and heat map to database
Download detection and tracking video and data
End or back to the homepage of the GUI
Dashboard
Start web GUI
Enter video name
Get analytic data and heat map

Requirements

- Numpy
- Pillow
- Tensorflow ==1 .14.0
- keras
- Opencv-python
- Opencv-contrib-python
- Flask
- Scipy
- Scikit-learn == 0.19.2
- Psycopg2
- POSTGRESQL > 9.5

How to use

1. Download detection model and feature extraction model and save to directory /tracking/model_data

2. Build docker images (only in first time)

```
docker-compose Build
```

3. Run docker container

```
docker-compose up
```

4. Run application: open web browser and go to

   `localhost:5000` for uploading video and performing pedestrian analyzing
   `localhost:4040` for viewing the dashboard of analyzed video

5. Stop docker container

```
docker-compose down
```

Note for further improvement:

- Detection accuracy affects to pedestrian ID switch. At the implementation time, YOLOv3 may be the good choice for both real time and accuracy, other detection models should be explored (especially, YOLOv4 and YOLOv5 just released in after April 2020).
- Explore other person re-ID algorithms to address ID switch issue.
- Modify Deep SORT algorithm for multi camera object tracking (MCMT).

References

1. Darknet project: https://github.com/AlexeyAB/darknet
2. YOLOv3 implementation in Keras: https://github.com/qqwweee/keras-yolo3
3. Deep Sort algorithm: https://github.com/nwojke/deep_sort
4. Baseline Person re-ID Pytorch: https://github.com/layumi/Person_reID_baseline_pytorch
5. 3DPeS dataset: https://aimagelab.ing.unimore.it/imagelab/page.asp?IdPage=16