CuttinEdge

The underlying technology

Lucas Saban
ENSAE Paris, Institut Polytechnique de Paris
MVA, ENS Paris-Saclay

July 25, 2022

Problem

We seek to identify and count the number of customers in a hair salon
Our constraints :

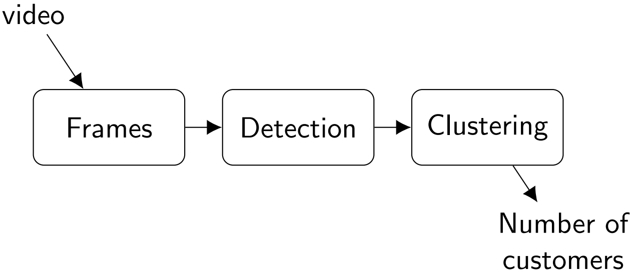
- The unpredictable events (People waiting in chair, child with parents, movements of hairdressers)
- Visual obstructions (hairdressers blocking the view, etc.)
- The less salon-dependant parameters the better, for scalability
- We rely on video feeds, the less hardware the better

Overview

1. **Overall Pipeline**
2. **Human Detection**
3. **Pattern Recognition with** `c4der`

Overall Pipeline

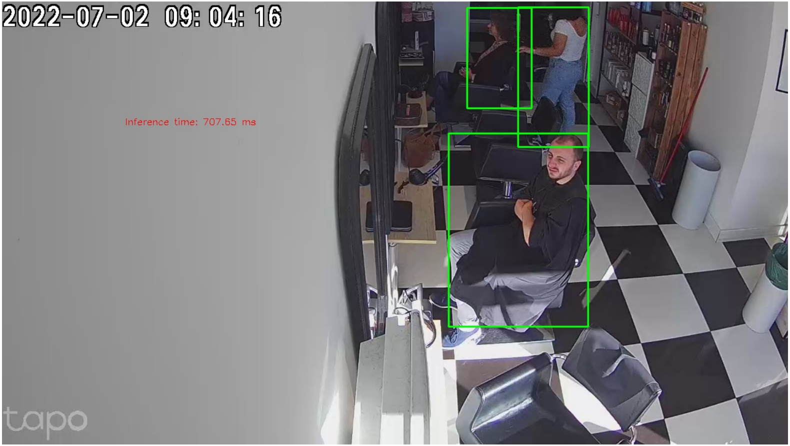
Global Pipeline



Human Detection

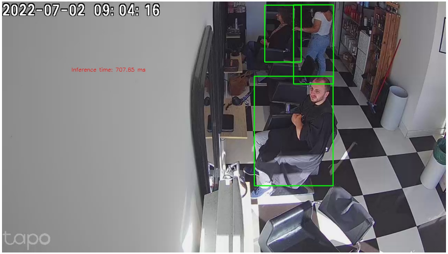
pre-detection

post-detection



data fetched

From the frame, we extract temporally indexed vectors for each person detected



In particular, we get :

- The bounding boxes centroid, width and length
- the timestamp

How ?

We use a convolutionnal neural network using the YoloV4[1] architecture.
The model is pre-trained on COCO[2] Dataset (>300K images)

Why YoloV4 ?

- You Only Look Once : single stage object detector $\rightarrow$ fast detection
- Relatively lightweight $\rightarrow$ easier to distribute
- Accurate enough
- Compatible with opencv[3] built-in neural network implementation

[1] Bochkovskiy et al.
[2] Lin et al. [2014]
[3] Bradski

Pattern Recognition with `c4der`

First glance

If we plot the centroids found with respect to time :

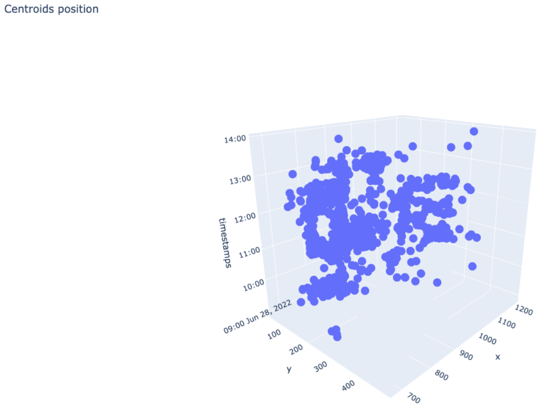


Figure: raw_points.html

Zoom in

Some patterns seems to appear, it is not random noise.

If we can regroup points together[4] and get rid of the noise[5], we can retrieve the underlying distribution.

[4] e.g. count customers

[5] +1 robustness

Clustering

Goal of clustering

Create clusters that minimizes intra-cluster differences while maximizing inter-cluster differences.

Yet, we have more constraints compared to usual clustering setups :

1. We have 1 temporal , 2 spatial and k other dimensions[6].
2. Variations along the Y axis are more meaningful to distinguish clusters than along the X axis [7]
3. We need adaptability because clusters don't share the same intra-variance because of perspective[8]
4. We need to distinguish a customer waiting from a customer being processed.

[6] width, lengths of bounding and any transformation of those
[7] Seats can be separated with horizontal lines, not vertical
[8] 2D representation of 3D space

Proposed solution

c4der

c4der for Clustering for Dynamical Event Recognition is a clustering algorithm that answers the attention points from the last slide.

Based on the DBSCAN/ST_DBSCAN algorithms[9], it is :

- Agnostic w.r.t. data distribution (no convexity or compacity hypothesis)
- compatible with non-spatial data

[9] https://www.youtube.com/watch?v=RDZUdRSDOok

`c4der`

To reach our constraints, new features were added.

Thus, compared to DBSCAN, `c4der` :

- is semi-stochastic using gaussian-based kernels for neighborhoods.
- uses tweaked Mahalanobis-distance for spatial distances
- detects seats and pre-clusterizes points according to them
- filters clusters by timespan and intra-variance

c4der **in practice**

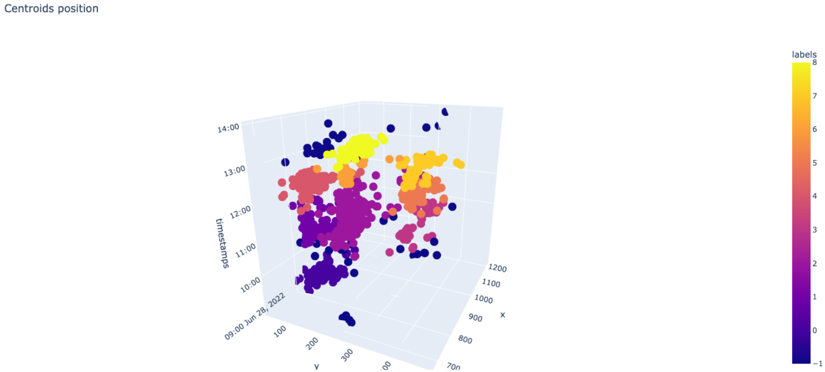


Figure: clusters.html

Post-processing

The clusters left are the customers we sought to identify.
A -1 label corresponds to noise.
With little post-processing, we can extract from the previous graph :

```
                    ################## c4der ##################

[c4der]       Cluster 0 rejected => [size = 5] [TimeSpan = 0:02:40] [Variance = 13.043772460450237]
[c4der]        Cluster 1 rejected => [size = 6] [TimeSpan = 0:02:00] [Variance = 4.667561298175124]
[c4der]       Cluster 11 rejected => [size = 6] [TimeSpan = 0:08:20] [Variance = 12.266711956438133]
                           Nombre de clients détectés :9
                                 09:09 -->  09:37
                                 10:35 -->  11:38
                                 10:48 -->  12:18
                                 11:10 -->  11:25
                                 11:46 -->  12:34
                                 11:53 -->  12:09
                                 12:29 -->  12:54
                                 12:33 -->  12:54
                                 12:57 -->  13:20
                              Execution time : 967 ms

                     ############################################
```

By having a macro point of view, the process is more **robust** to noise and unpredictable events.
Coupled with human verification in case of fraud suspicion, it can handle most of the situations **accurately**.

Summary

- The solution proposed is a **two-staged pipeline**.
- The first detection step is handled by a **pretrained neural network** called YoloV4.
- The analysis and clustering step uses a **robust clustering algorithm**, `c4der` that can identify and filter clusters using a macro approach.
- Compared to a sequential approach, the overall pipeline is
  1. more robust to noise and unpredictable events,
  2. requires less salon-dependent setup
  3. requires less computational ressources
  4. original, thus not likely to be used by any potential competitor.

References

D. Birant and A. Kut. ST-DBSCAN: An algorithm for clustering spatial–temporal data. 60(1):208–221. ISSN 0169-023X. doi: 10.1016/j.datak.2006.01.013.

A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. YOLOv4: Optimal speed and accuracy of object detection. URL http://arxiv.org/abs/2004.10934.

G. Bradski. The OpenCV library. 25(11). ISSN 1044-789X. URL https://elibrary.ru/item.asp?id=4934581.

M. Ester, H.-P. Kriegel, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. page 6.

T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2014. URL https://arxiv.org/abs/1405.0312.

The End