Master in
Computer Vision
Barcelona

Project
Module 6
Coordination

Video Surveillance for Road Traffic Monitoring

J. Ruiz-Hidalgo / X. Giró
j.ruiz@upc.edu / xavier.giro@upc.edu

Team 5 Final presentation

Eloi Bové, Jordi Burgués, Albert Jiménez, Arnau Roche

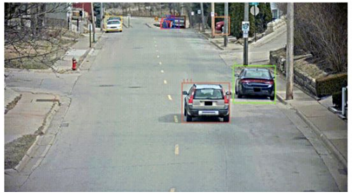
Master in
Computer Vision
Barcelona

UAB UOC UPC upf.

Table of contents

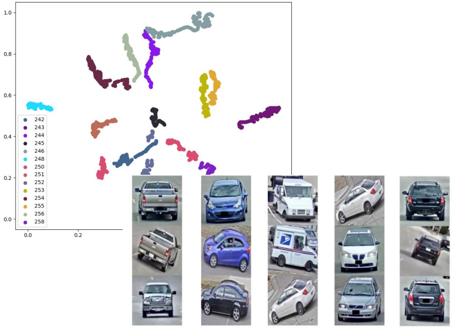
T5: Eloi Bové

1 - MTSC tracking



1.1 Methods
1.2 Results and discussion

2 - MTMC tracking



2.1 Tracking pipeline
2.2 Vehicle embeddings
2.3 Matching algorithm
2.4 Results and discussion

3 - Conclusions

3.1 Summary and conclusions

1.1 MTSC tracking - Methods

T5: Eloi Bové

Maximum Overlap Tracking

Track objects with an IoU greater than a certain threshold.

Week 3 results: **IDF1=0.774**, MOTA=0.764

Kalman Filter Tracking

Track objects based on a constant velocity model that also takes into account the object size and aspect ratio. Based on SORT [1]

Week 3 results: IDF1=0.714, **MOTA=0.821**

- Maximum overlap provided higher IDF1 in previous weeks. However, the **new ground truth** favors the Kalman filter approach.

- **Having a good single camera tracking** will be key for our multi camera tracking system.

[1] Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP) (pp. 3464-3468). IEEE

1.2 MTSC tracking - Results and discussion

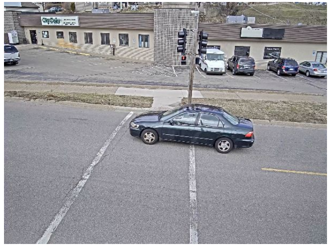
T5: Eloi Bové

Summary of the obtained results

	IDF1 (SEQ 3)							
Camera	**c010**	**c011**	**c012**	**c013**	**c014**	**c015**	**Average IDF1**	**Weighted IDF1 average**
Number of frames	1923	2031	2156	2185	2275	16		
SSD-512 + **Overlap**	0.3422	**0.3737**	0.0863	0.4489	0.3889	**1.0**	0.4400	0.3292
Yolo3 + **Overlap**	0.4361	0.2284	0.0442	0.2416	0.5091	0.5925	0.3420	0.2923
Mask RCNN + **Overlap**	0.4920	0.1810	0.0701	**0.6643**	0.5709	**1.0**	0.4964	0.3998
SSD-512 + **Kalman**	0.4326	0.3003	**0.1422**	0.3175	0.3260	**1.0**	0.4197	0.3023
Yolo3 + **Kalman**	0.6397	0.3101	0.0257	0.2170	**0.6477**	0.6666	0.4178	0.3660
Mask RCNN + **Kalman**	**0.6461**	0.3470	0.1097	0.6607	0.6029	**1.0**	**0.5611**	**0.4738**

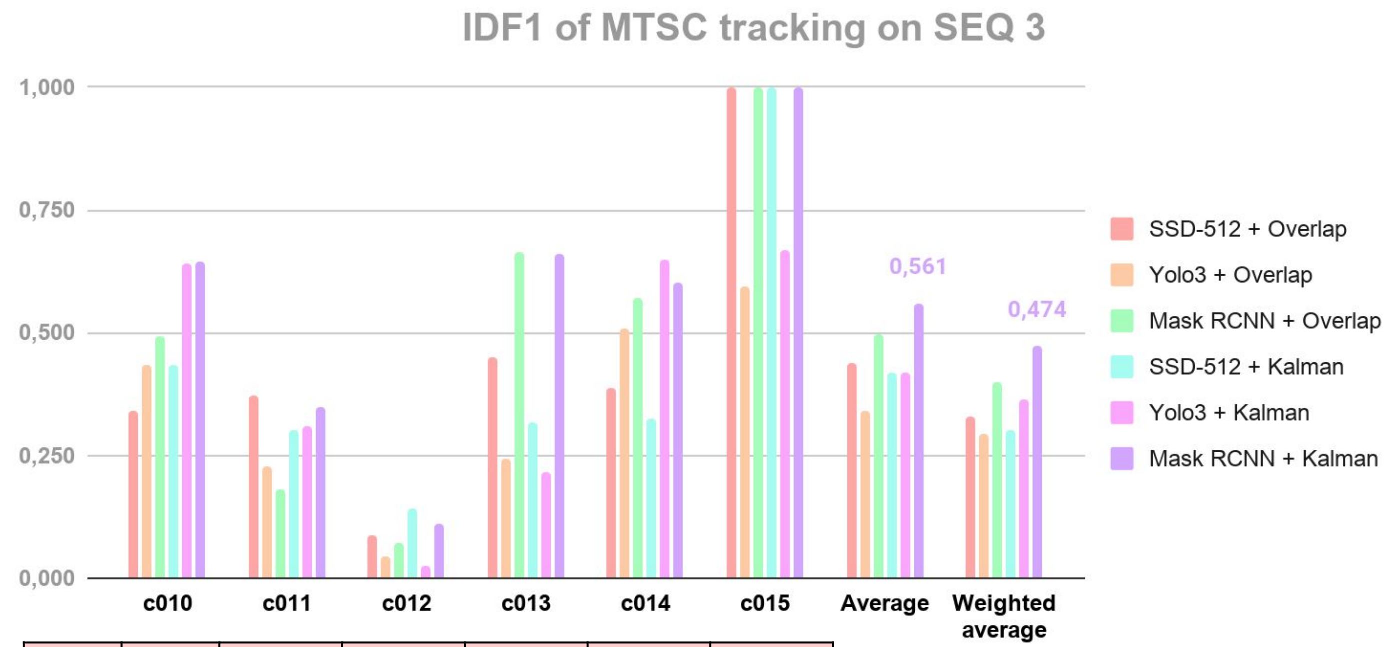
1.2 MTSC tracking - Results and discussion

T5: Eloi Bové



c012 perspective

c014 perspective



Num frames	1923	2031	2156	2185	2275	16

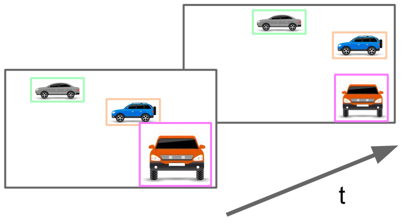
All in all, the best results are provided by the Kalman tracking with Mask RCNN detections, and this method will be used as the baseline for our Multi Camera Tracking system.

2.1 MTMC tracking - General Pipeline

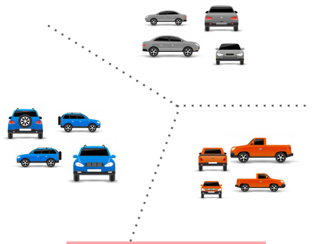
T5: Albert Jiménez

We implemented the following **pipeline** in order to track **multiple cars** across **multiple cameras**:

Tracking
Mask RCNN + Kalman

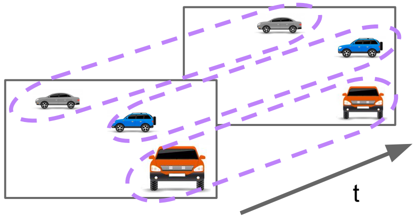


Embeddings
For each BB, to represent cars
512 dimensions



Trained triplet NN

Tracklets
BB evolution across frames **from the same camera**

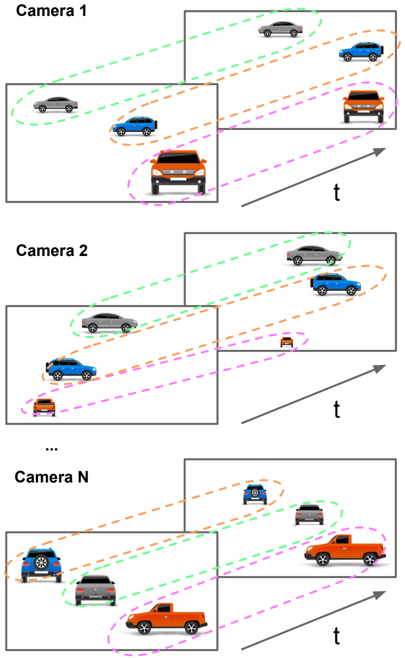


Embedding statistics
For each Tracklet

- ➔ Embedding mean
- ➔ Embedding variance

Distance between embedding statistics:
- ➔ Gaussian prob.
- ➔ Bhattacharyya
- ➔ Squared distance

Matching
Between tracklets **from different cameras** to assign a global ID



2.2 MTMC tracking - Vehicle Embeddings

We have performed metric learning to obtain meaningful embeddings to compare cars from different cameras in sequence 03. **Two strategies** to compute these embeddings:

- ➔ Inference on pre-trained model with the **VeRi dataset**[2]
- ➔ Using AI City data:
    - **Dataset creation** with our own car crops
    - Training of a Triplet Network from scratch using our dataset
    - Inference on our trained model

VeRi dataset

over 50k images,
776 vehicles,
20 cameras
24 hour recordings

Our dataset

over 44k images
184 vehicles
36 cameras
3-4 minutes recording

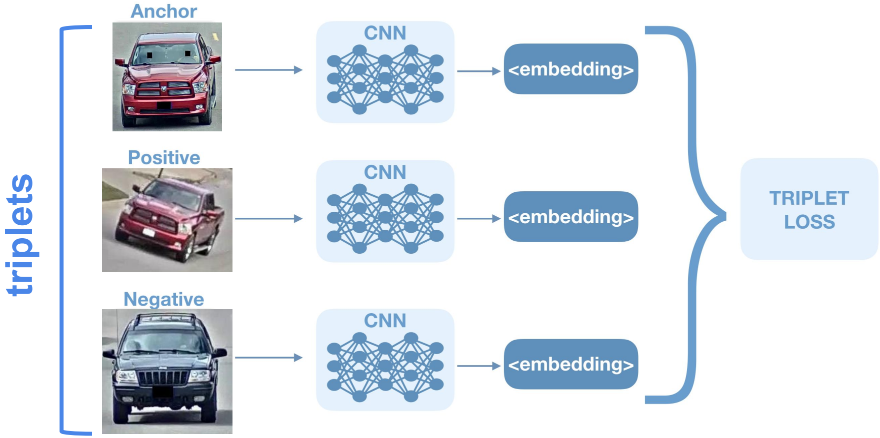
[2]: JDAI - CV: https://github.com/JDAI-CV/VeRidataset

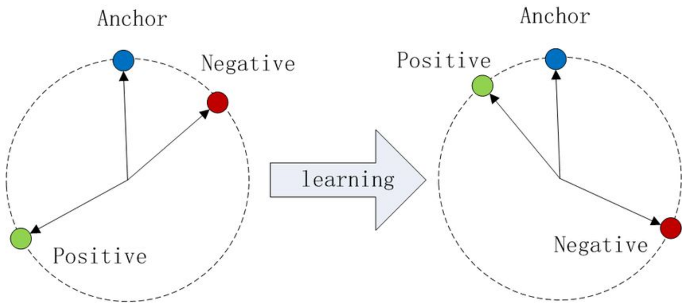
2.2 MTMC tracking - Vehicle Embeddings

T5: Jordi Burgués

TRIPLET NETWORK

- Both models used for embedding computation use a **Triplet Network**[3] with a **Triplet Loss**
- A triplet network uses an **anchor**, a **positive** and a **negative** sample as inputs (triplets)
- **Metric learning**: learn embeddings such that the positive sample is closer to the anchor than the negative sample is, by some margin value





TRIPLET LOSS

$$\mathcal{L} = max(d(a,p) - d(a,n) + margin, 0)$$

L2 distance between anchor and positive sample

[3]: Pranoy Radhakrishnan: https://github.com/pranoyr/vehicle-reid

2.2 MTMC tracking - Vehicle Embeddings

T5: Jordi Burgués

MODEL TRAINING

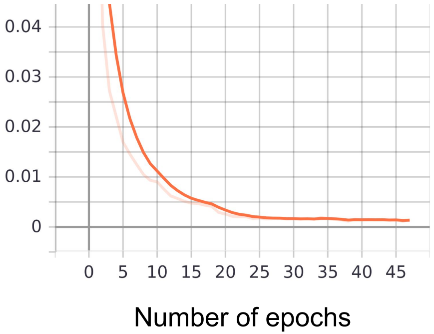
- Using the aforementioned Triplet Network we have performed metric learning from scratch by training a ResNet-18 architecture with our own dataset of cars (extracted from AI City):
  - S01 and S04 as training data → 37984 images
  - S03 as validation data → 6170 images

TRAINING PARAMS

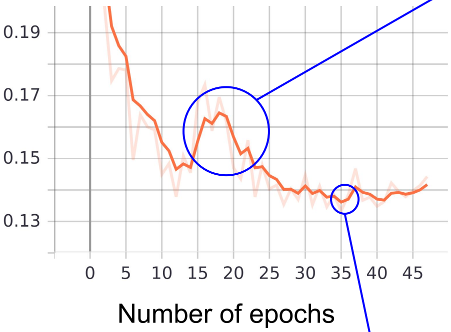
LR: 1e-3, with decay scheduler every 20 epochs
Epochs: 46
Batch size: 32
Optimizer: SGD
Triplet Loss

Training and validation images are resized to 224x224

TRAINING LOSS



VALIDATION LOSS



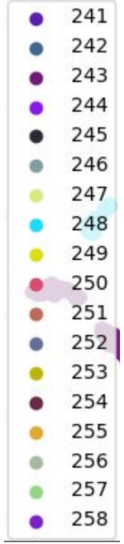
Learning rate decay helps in validation loss convergence

Embedding computation when the model yields minimum validation loss

TRAINING TIME
36 hours

2.2 MTMC tracking - Vehicle Embeddings

T5: Jordi Burgués

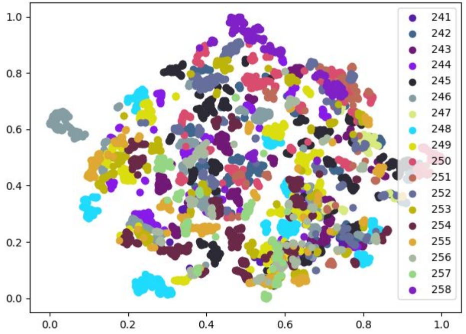


MODEL INFERENCE

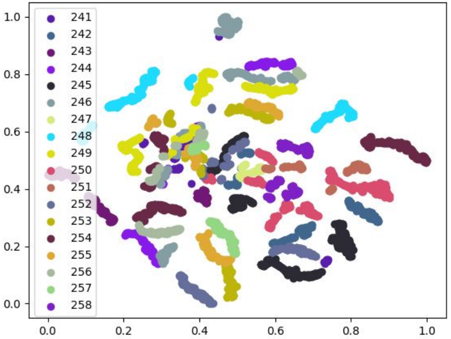
- With the pretrained model on VeRi[3] and our trained model on cars from AI City, we can visualize the embeddings inferred by the Triplet Network, which will be used as feature vectors for car re identification
- Every embedding has 512 dimensions → t-SNE to 2 dimensions for the sake of visualization

S03
All cameras

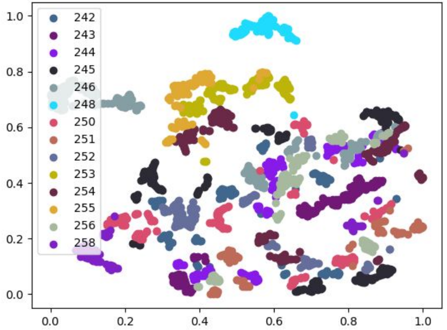
S03
Camera 10



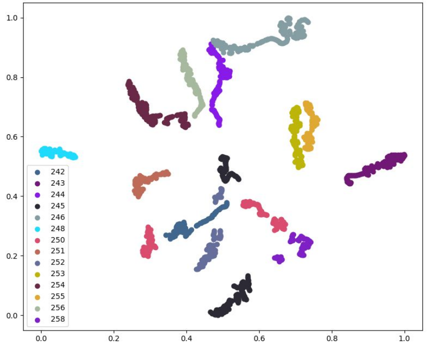
pretrained model with VeRi



our trained model



pretrained model with VeRi



our trained model

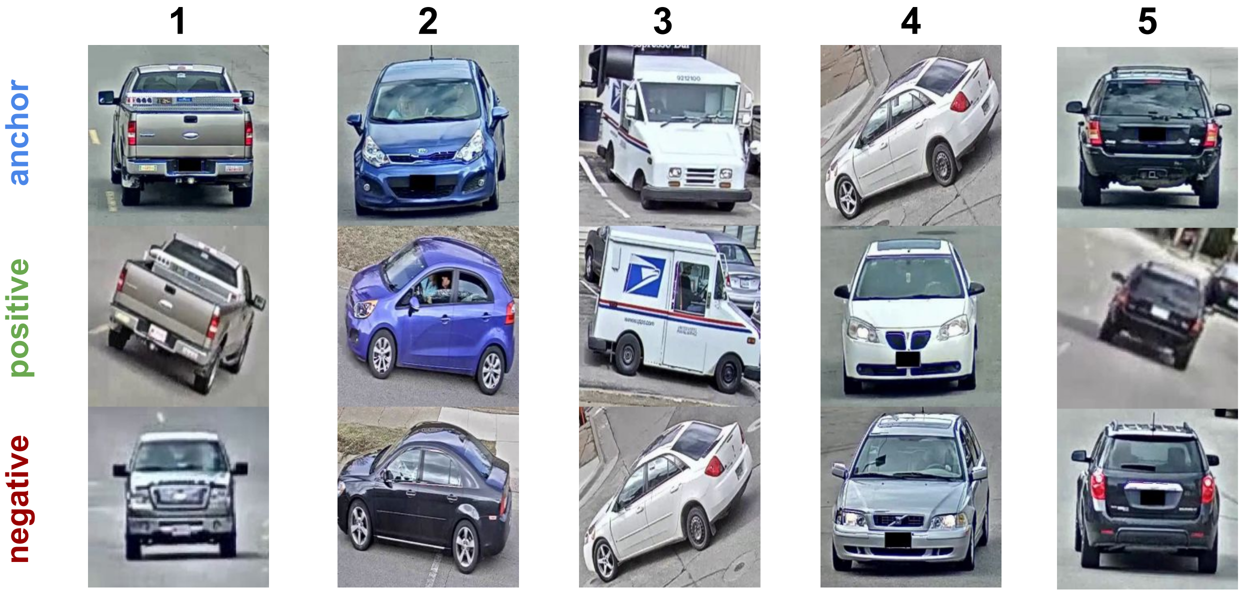
[3]: Pranoy Radhakrishnan: https://github.com/pranoyr/vehicle-reid

2.2 MTMC tracking - Vehicle Embeddings

T5: Jordi Burgués

MODEL INFERENCE

- By generating triplets of images, we can perform inference with our trained model and compute the anchor-positive, d(a,p), and anchor-negative, d(a,n), L2 distances for evaluation

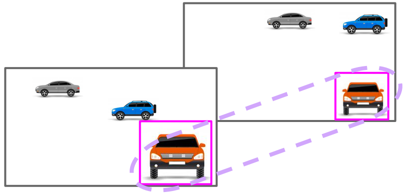


	1	2	3	4	5
d(a,p)	3.96	4.07	4.55	5.40	4.06
d(a,n)	5.47	6.39	5.91	5.78	3.27

2.3 MTMC tracking - Matching Algorithm

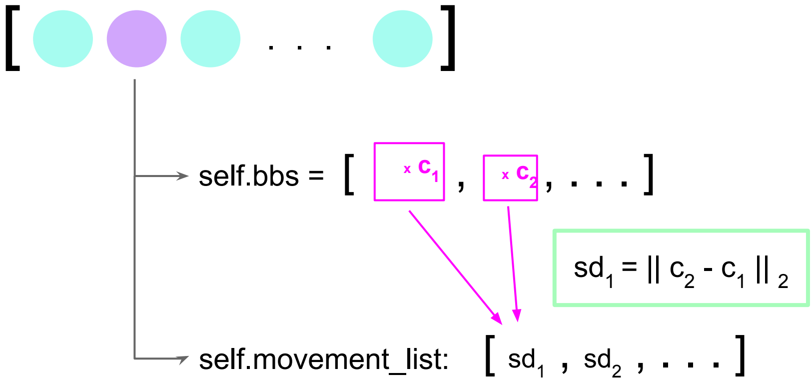
T5: Albert Jiménez

Before we actually match the **tracklets**, we filter out some to improve the results and make the car re-identification simpler. Particularly, we detect and **delete tracklets** corresponding to **parked cars**:



Tracking scenario

Tracklets:



Pseudo-code to **remove** tracklets of **parked cars**:

```
for tracklet in tracklets:
    if median(tracklet.movement_list) < 2.75:
        self.parked = True
        remove tracklet
```

c_{1}, c_{2} are the bounding box centers

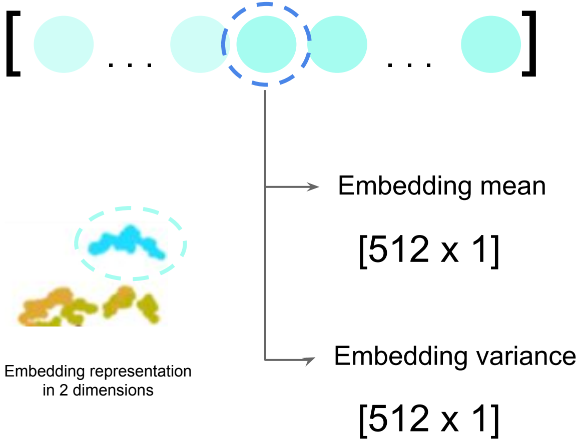
The vehicle movement is estimated by the squared difference of the BB centers.

2.3 MTMC tracking - Matching Algorithm

T5: Albert Jiménez

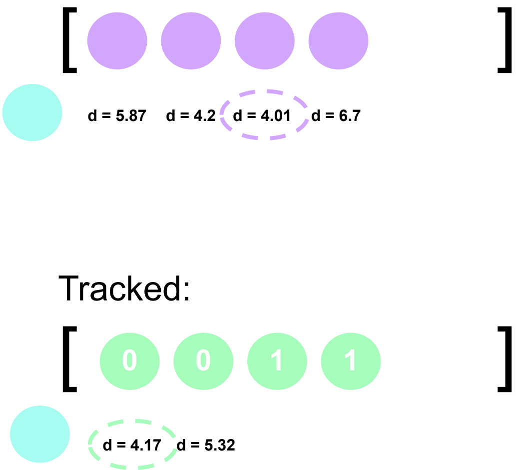
Tracklets are arranged in **order of appearance** and **matched**, if possible, with **previous tracklets** based on the **embedding statistics**.

Tracklets:



Embedding representation in 2 dimensions

Untracked:



3 distances
3 thresholds

Logprob
d>0.5

Bhattacharyya
(PCA 20 comp.)
d<23

Squared distance
d>4.62

A tracklet can be matched to another from the **untracked** or **tracked** list **indistinctly**. If it is matched to an **untracked tracklet**, a **new ID** is generated. At the end, **untracked tracklets** are **discarded**.

2.4 MTMC tracking - Results and discussion

T5: Arnau Roche

Both cars 5 and 6 are re-identified when they go from one camera to another one

2.4 MTMC tracking - Results and discussion

T5: Arnau Roche

Car 15 is correctly re-identified between 2 cameras, but a different ID is given for a third one

2.4 MTMC tracking - Results and discussion

T5: Arnau Roche

Similar white cars are confused: ID=20 is assigned to different cars and the same car has ID=20 and ID=24

2.4 MTMC tracking - Results and discussion

T5: Arnau Roche

Car is correctly identified with ID=32

MTMC tracking - Results and discussion

T5: Arnau Roche

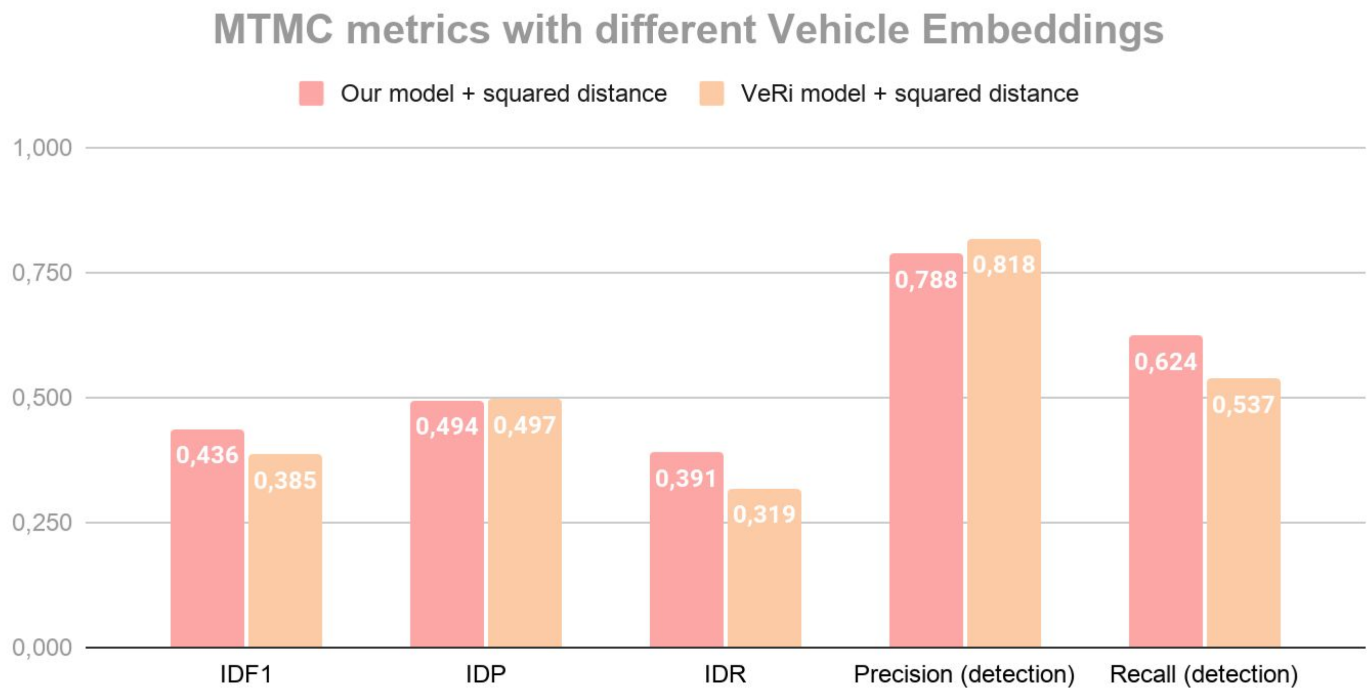
Summary of our results

Metric	IDF1	IDP	IDR	Precision (detection)	Recall (detection)
Our model + squared distance	0.436278	0.49365	0.390853	0.787792	0.623743
VeRi model + squared distance	0.385309	0.497345	0.319007	0.818249	0.536653

MTMC tracking - Results and discussion

T5: Arnau Roche

Summary of our results



MTMC tracking - Conclusions

T5: Arnau Roche

MTSC tracking

- **Mask RCNN** works better than SSD/Yolo3, and **Kalman filter** approach outperforms maximum overlap method.
- **Longitudinal car movements** are better tracked with our method than transversal ones, presumably because they appear more time on camera.
- The **GT** is **not suited** for **MTSC**: results are better than metrics show.

MTMC tracking

- Our **trained embeddings outperform** the **pre-trained VeRi** ones as they are more specific to our testing scenario.
- **Camera synchronization** is key to assist vehicle re-identification with **temporal constraints**.
- **Spatial constraints** could be applied, but they would be specific to each scenario.

MTSC and MTMC

- Both MTSC and MTMC results could be improved by making our tracking algorithm more robust to **missed detections**.
- **Parked car removal** improves tracking results and makes car re-identification easier.

Thank you!