Video Surveillance for road traffic monitoring

Final Presentation - Team 1

› Aditya Rana, Carmen García,
Juan Chaves & Germán Barquero ‹

I. Multi-target single-camera tracking (MTSC)

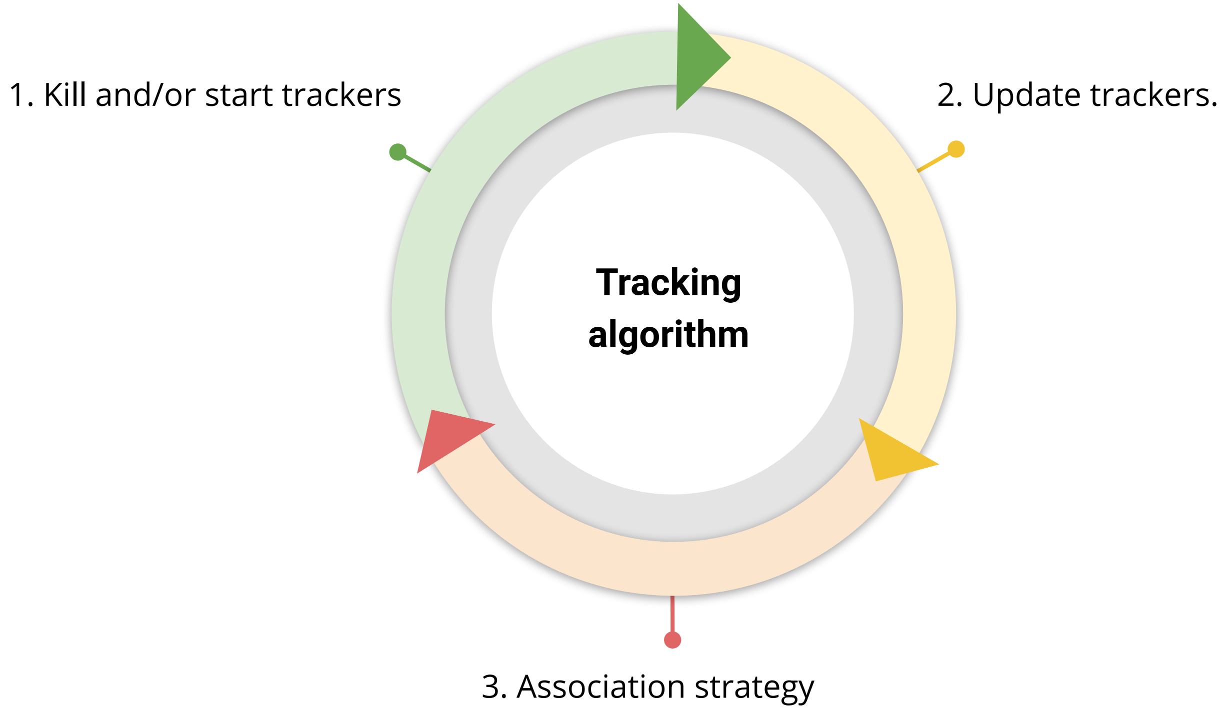
I - MTSC tracking - Scenario

Goal: assigning a unique IDs to the same object along the whole video sequence.

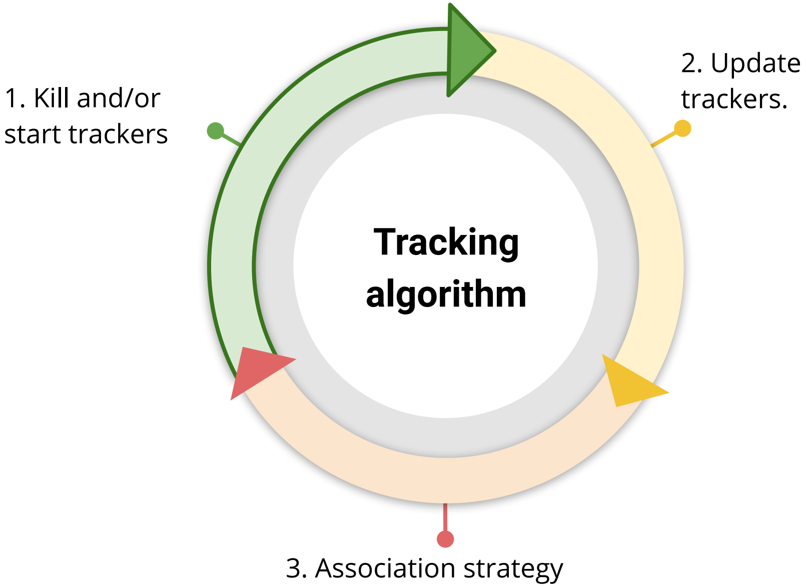
Tracking algorithms

All-in-one	⟷	Detection-based
Classic	⟷	Deep learning
Online	⟷	Offline
Short-term	⟷	Long-term

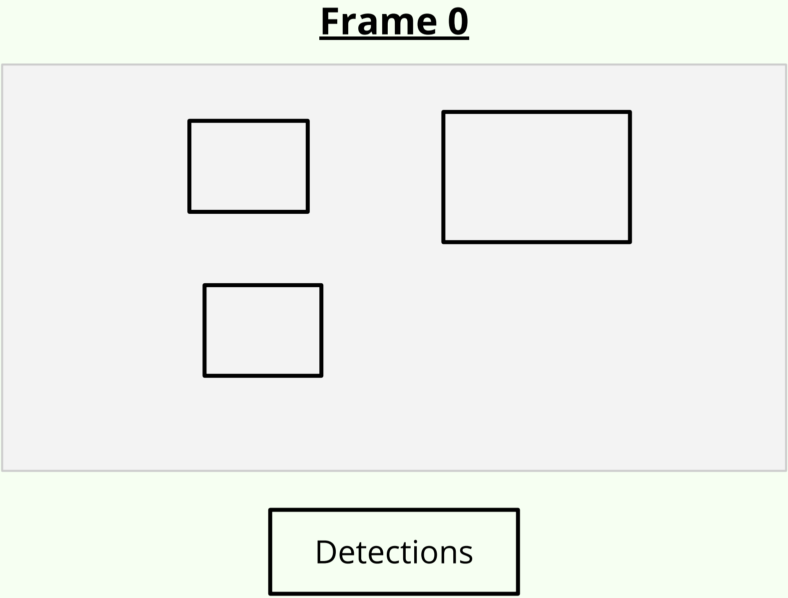
I - MTSC tracking - Our strategy



I - MTSC tracking - Our strategy

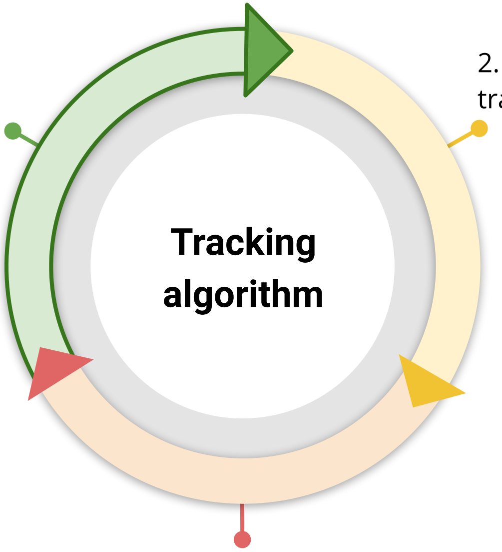


1. Kill and/or start trackers

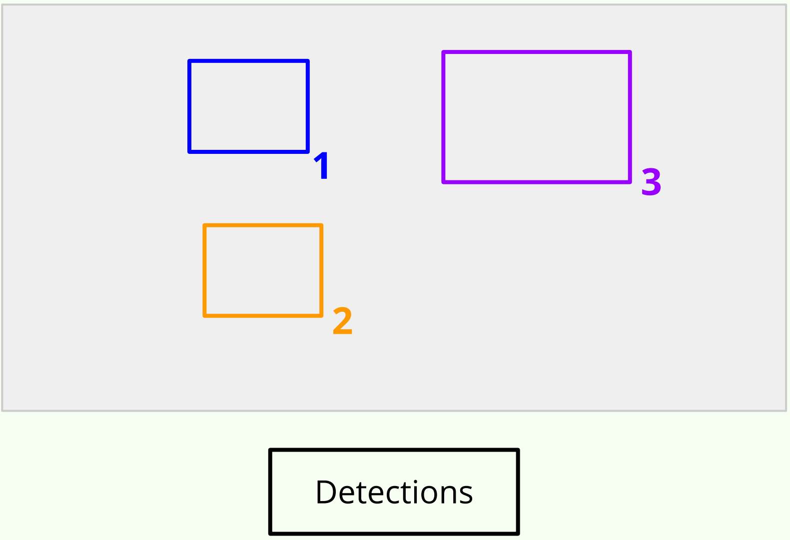


I - MTSC tracking - Our strategy

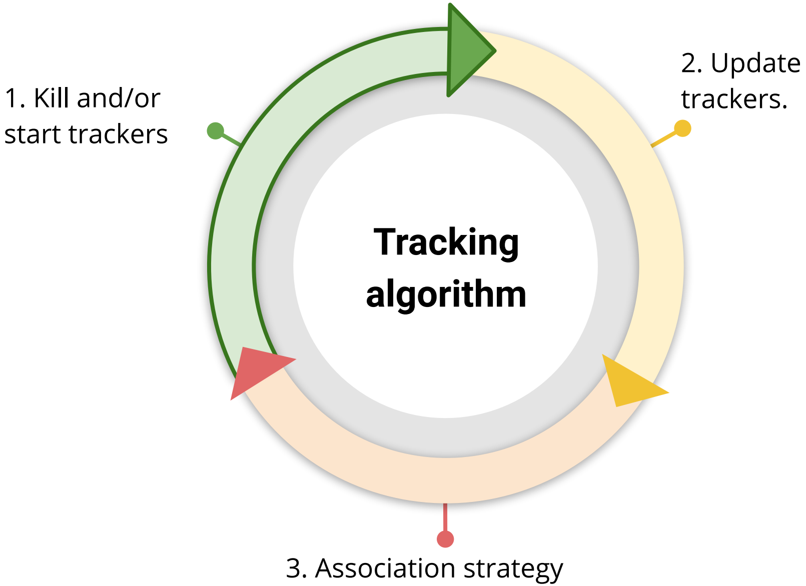
1. Kill and/or start trackers



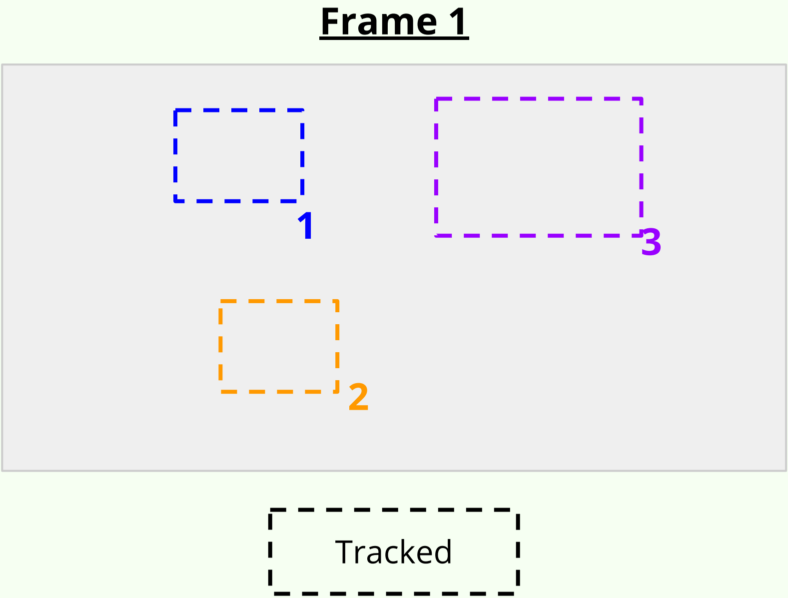
Frame 0



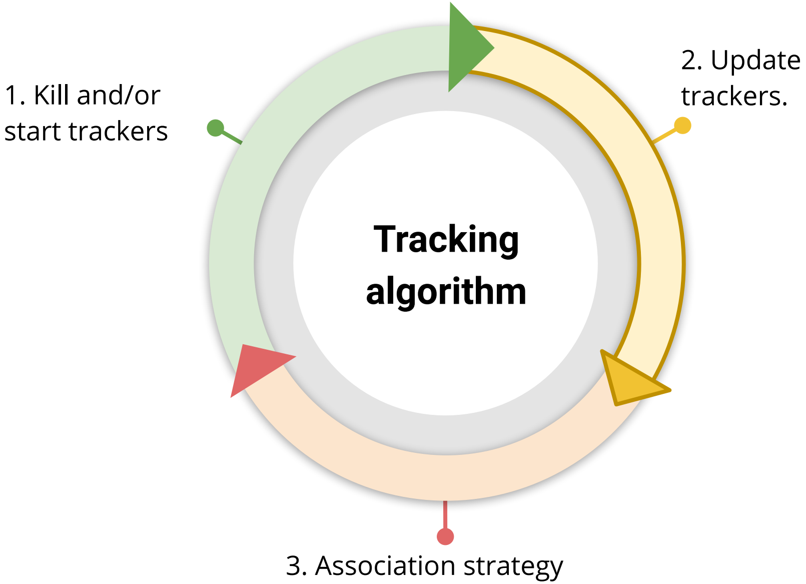
I - MTSC tracking - Our strategy



1. Kill and/or start trackers

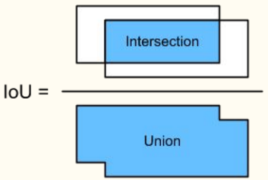


I - MTSC tracking - Our strategy

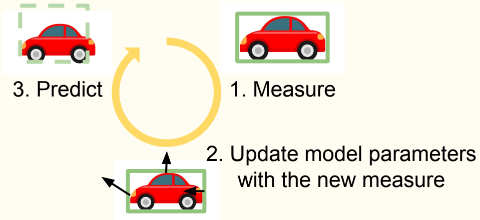


2. Trackers update

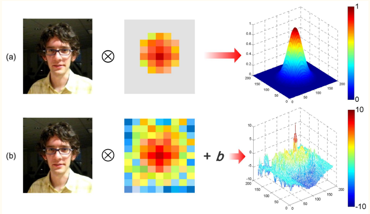
IOU



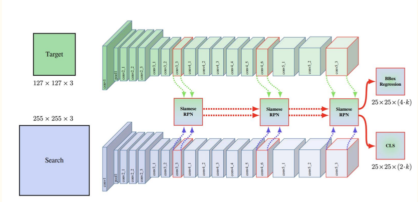
Kalman



KCF[1]



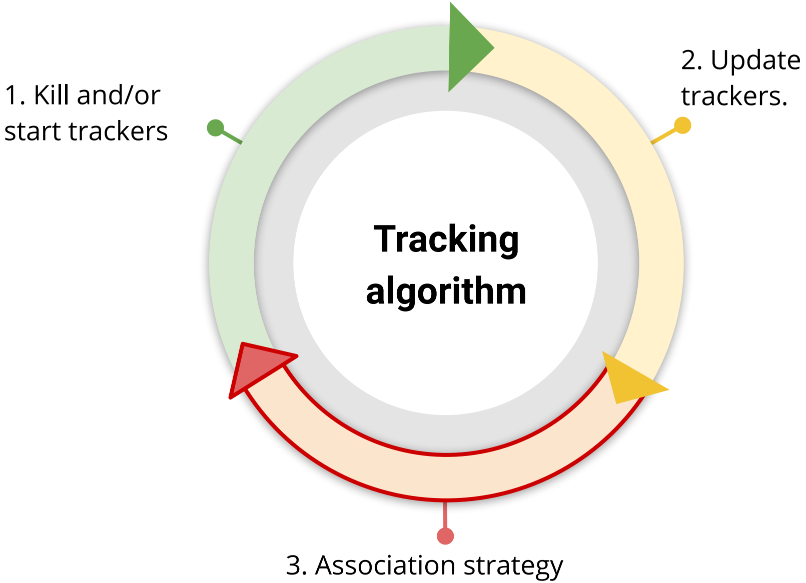
SiamRPN++[2]



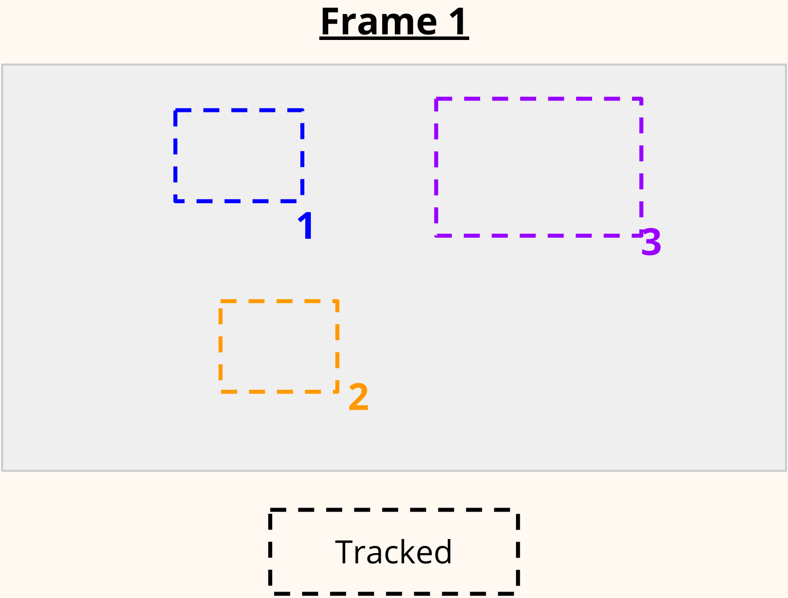
[1] Henriques, João F., et al. "High-speed tracking with kernelized correlation filters".

[2] Li, Bo, et al. "Siamrpn++: Evolution of siamese visual tracking with very deep networks".

I - MTSC tracking - Our strategy

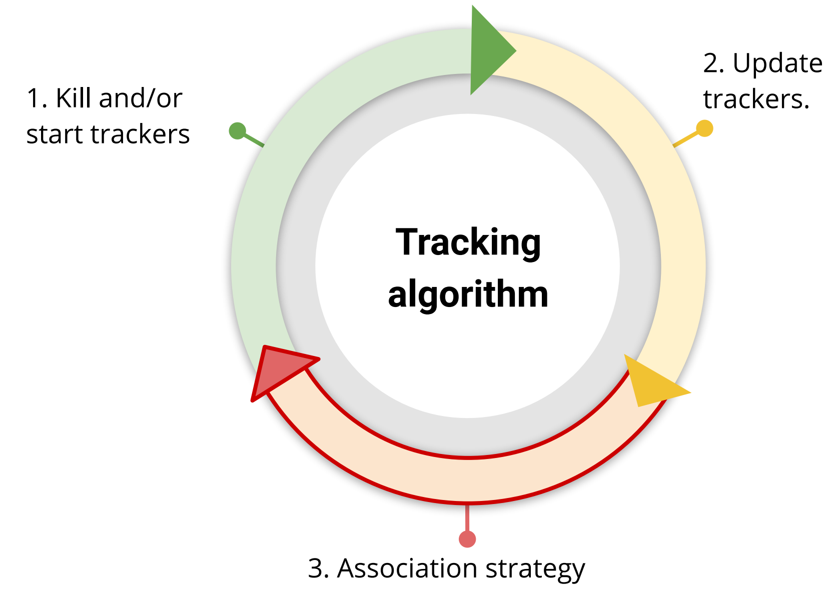


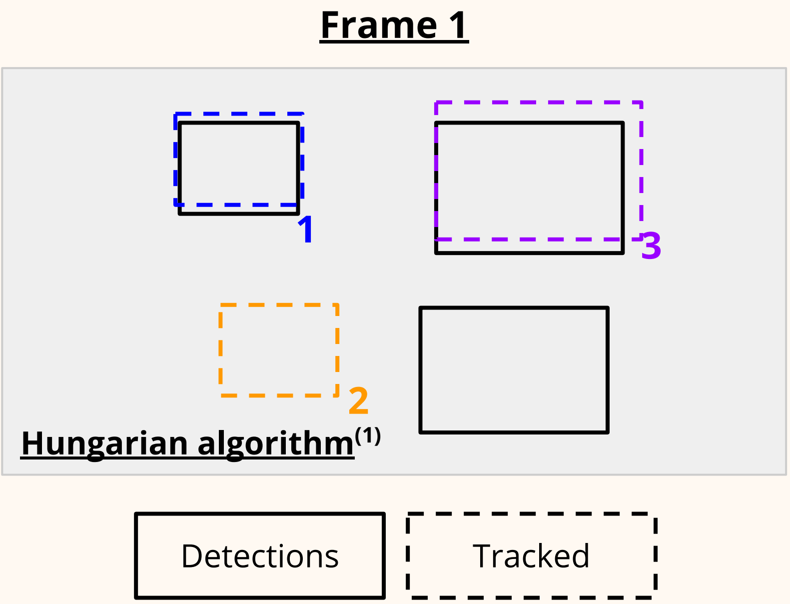
3. Association strategy



I - MTSC tracking - Our strategy

3. Association strategy

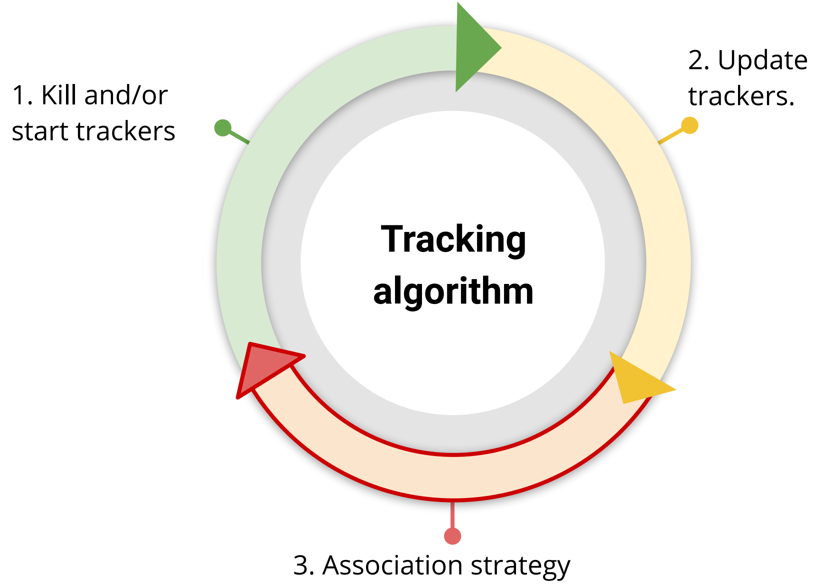


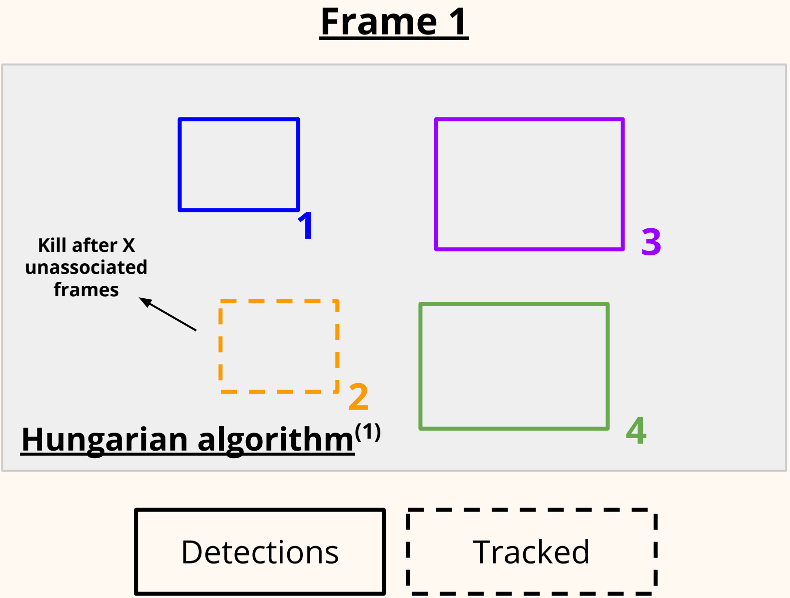


[1] Munkres, James. "Algorithms for the assignment and transportation problems."

I - MTSC tracking - Our strategy

3. Association strategy





[1] Munkres, James. "Algorithms for the assignment and transportation problems."

I - MTSC tracking

➔ Detections used were extracted with the SSD512[(1)] detector.

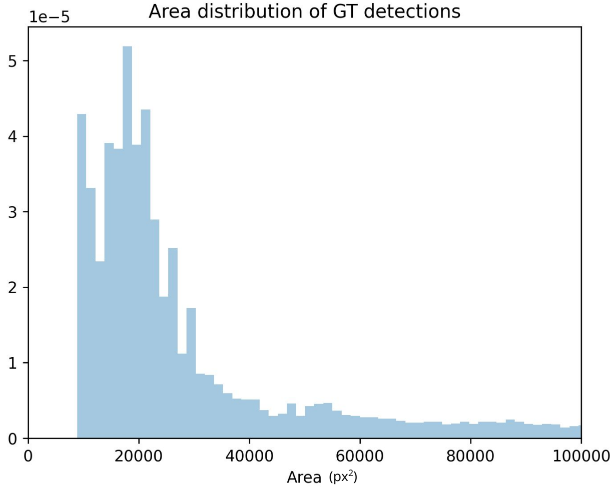
➔ Baselines: SSD512 + DeepSort[(2)]

		IDF1 (SEQ 3)						
Camera		**c10**	**c11**	**c012**	**c013**	**c014**	**c015**	**Average**
BASELINE		30.9	4.3	2.8	58.1	44.6	0.3	23.5
SSD	IOU	34.5	16.9	**4.1**	64.6	45.6	**1.4**	27.9
	Kalman	34.1	15.8	**4.1**	60.2	43.0	**1.4**	26.4
	KCF	**35.2**	17.1	3.9	64.6	**46.6**	**1.4**	**28.1**
	SiamRPN++	34.9	**17.5**	**4.1**	**64.8**	44.7	**1.4**	27.9

[1] Liu, Wei, et al. "Ssd: Single shot multibox detector." *European conference on computer vision*. Springer, Cham, 2016.
[2] Wojke, Nicolai, and Alex Bewley. "Deep cosine metric learning for person re-identification." WACV, 2018.

I - MTSC tracking - Post-processing 1

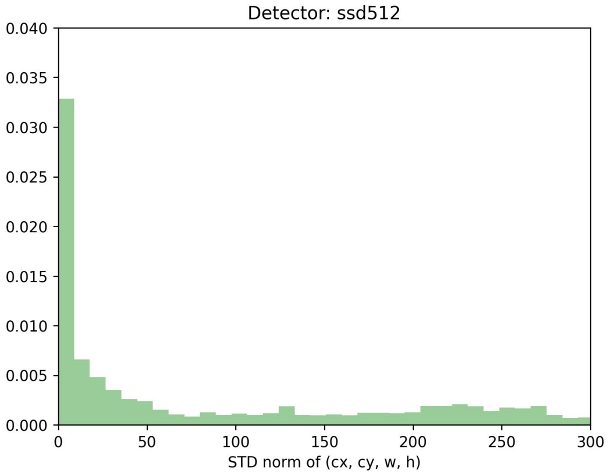
- ➔ Small detections were filtered.



GT clearly does not consider small detections

I - MTSC tracking - Post-processing 2

➔ Parked cars were removed.



I - MTSC tracking - Results

➔ Post-processing dramatically increased our IDF1 values.

		IDF1 (SEQ 3)							
Camera		**c10**	**c11**	**c012**	**c013**	**c014**	**c015**	**Average**	**Increase**
BASELINE		84.0	52.4	8.6	**67.5**	**72.0**	6.9	48.6	**+25.1**
SSD	IOU	80.4	66.3	**10.9**	63.6	66.3	**11.5**	49.8	+21.9
	Kalman	79.5	64.0	**10.9**	60.7	62.7	**11.5**	48.2	+21.8
	KCF	80.7	67.1	10.2	63.6	66.9	**11.5**	**50.0**	+21.9
	SiamRPN++	**81.4**	**68.6**	10.7	63.5	63.6	**11.5**	49.9	+22.0

I - MTSC tracking - Results

➔ Post-processing dramatically increased our IDF1 values.

		IDF1 (SEQ 1)					
Camera		**c01**	**c02**	**c03**	**c04**	**c05**	**Average**
BASELINE		54.0	62.8	52.4	69.1	22.0	52.1
SSD	IOU	55.6	**76.1**	51.8	**72.4**	**22.9**	**<u>55.8</u>**
	Kalman	**56.4**	72.9	**52.5**	72.1	22.5	55.3
	KCF	54.3	75.2	50.6	69.1	21.7	54.2
	SiamRPN++	55.6	72.7	50.5	69.1	21.7	53.9

I - MTSC tracking - Results

➔ Post-processing dramatically increased our IDF1 values.

		IDF1 (SEQ 4)								
Camera		**c16**	**c17**	**c18**	**c19**	**c20**	**c21**	**c22**	**c23**	**c24**
BASELINE		63.5	46.5	59.1	**89.2**	**75.3**	70.4	66.6	**69.4**	34.7
SSD	IOU	57.7	**61.2**	54.7	42.2	72.9	**82.1**	62.9	61.7	56.1
	Kalman	57.8	48.7	56.0	42.3	51.3	79.7	62.9	57.2	**57.1**
	KCF	**63.8**	43.5	57.0	41.9	72.7	79.4	**66.1**	62.4	56.6
	SiamRPN++	61.2	53.9	**66.2**	58.2	45.5	80.5	65.9	62.4	55.6

I - MTSC tracking - Results

➔ Post-processing dramatically increased our IDF1 values.

		IDF1 (SEQ 4)								
Camera		**c25**	**c26**	**c27**	**c28**	**c29**	**c30**	**c31**	**c32**	**c33**
BASELINE		**76.3**	**70.8**	33.9	**55.0**	59.9	**53.0**	40.6	29.4	**72.9**
SSD	IOU	54.1	65.8	32.6	34.3	63.3	45.7	40.0	**42.2**	63.9
	Kalman	41.9	66.4	37.7	33.7	**65.3**	46.1	**40.8**	38.5	63.8
	KCF	54.4	65.6	38.9	46.0	62.1	45.8	39.7	**42.2**	70.9
	SiamRPN++	49.3	65.1	**39.2**	47.8	**65.3**	44.4	39.7	41.3	63.4

I - MTSC tracking - Results

➔ Post-processing dramatically increased our IDF1 values.

		IDF1 (SEQ 4)							
Camera		**c34**	**c35**	**c36**	**c37**	**c38**	**c39**	**c40**	**Average**
BASELINE		**57.2**	72.5	**62.2**	**73.6**	**69.9**	**64.6**	50.9	**61.2**
SSD	IOU	45.1	71.6	53.6	60.5	59.9	60.3	**70.5**	56.3
	Kalman	45.4	71.6	54.9	63.1	59.3	54.8	69.7	54.8
	KCF	53.5	**74.7**	56.1	60.7	58.2	57.5	51.7	57.1
	SiamRPN++	54.4	71.1	57.1	66.2	58.9	55.3	69.9	57.5

I - MTSC tracking - Analysis

		IDF1 (SEQ 4)
Camera		**c28**
BASELINE		**55.0**
SSD	IOU	34.3
	Kalman	33.7
	KCF	46.0
	SiamRPN++	47.8

➔ Detections at *frame=19*

19,	-1,	352.421,	121.505,	136.740,	58.375,	0.539
19,	-1,	193.297,	121.174,	125.303,	57.130,	0.458
19,	-1,	1196.477,	317.899,	551.855,	228.320,	0.412
19,	-1,	186.036,	133.312,	81.675,	47.166,	0.280
19,	-1,	283.902,	120.508,	65.168,	44.273,	0.228
19,	-1,	1200.248,	306.825,	554.370,	244.190,	0.856

IOU and Kalman are less precise so the **identity switches a lot**

I - MTSC tracking - Analysis

		IDF1 (SEQ 4)
Camera		**c40**
BASELINE		50.9
SSD	IOU	**70.5**
	Kalman	69.7
	KCF	51.7
	SiamRPN++	69.9

➔ Very fast car size changes may decrease the performance of correlation filters.

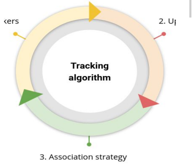
I - MTSC tracking - Conclusions

- ➔ IOU and Kalman trackers work well in constrained environments.
- ➔ The performance boost provided by DL trackers can not be appreciated.
- ➔ IDF1 values are similar for all algorithms due to noisy detections.

II. Multi-target multi-camera tracking (MTMC)

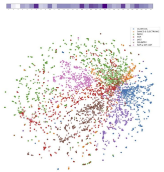
II - Multi-target multi-camera (MTMC) tracking

MTSC → **Get Track Features** → **Match Tracklets**



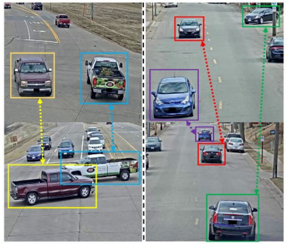
MTSC

Generate tracks for detected objects in all camera sequences



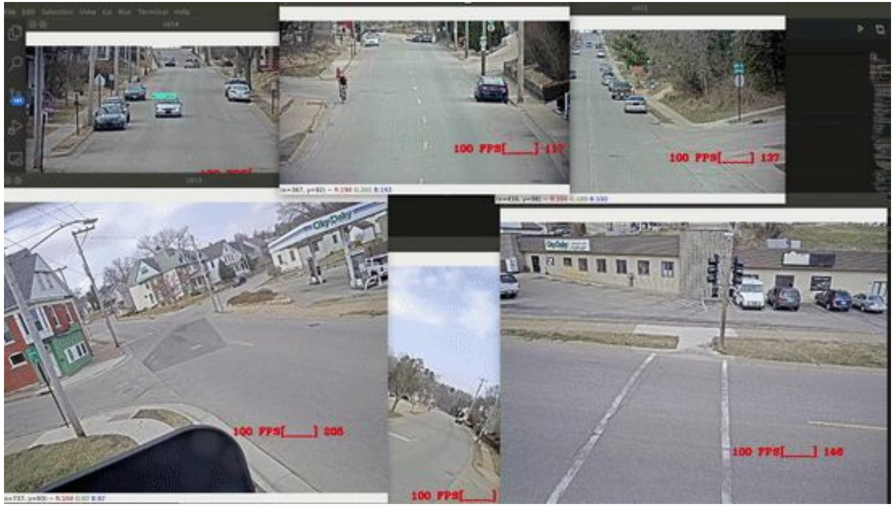
Track Features

Generate an embedding for each tracklet using classical or deep learning techniques



Match Tracklets

Generate an embedding for each track using classical or deep learning techniques



II - Multi-target multi-camera (MTMC) tracking

MTMC: **Algorithm***

We start from the **MTSC** we obtained for each camera and we define a **adjacency matrix** per sequence.

Neighbours: cameras on which a car can appear in a reasonable time

*for this part we assume a perfect single camera tracking (gt) to focus in evaluate this multicamera task

II - Multi-target multi-camera (MTMC) tracking

MTMC: **Algorithm***

We start from the **MTSC** we obtained for each camera and we define a **adjacency matrix** per sequence.

$$\begin{array}{cc} & \text{cameras} \\ \text{cameras} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Neighbours: cameras on which a car can appear in a reasonable time

*for this part we assume a perfect single camera tracking (gt) to focus in evaluate this multicamera task

II - Multi-target multi-camera (MTMC) tracking

MTMC: **Algorithm***

We start from the **MTSC** we obtained for each camera and we define a **adjacency matrix** per sequence.

We **synchronize** our search in time between cameras

$$\begin{array}{cc} & \text{cameras} \\ \text{cameras} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Active frames

Neighbours: cameras on which a car can appear in a reasonable time

*for this part we assume a perfect single camera tracking (gt) to focus in evaluate this multicamera task

II - Multi-target multi-camera (MTMC) tracking

MTMC: **Algorithm***

We start from the **MTSC** we obtained for each camera and we define a **adjacency matrix** per sequence.

We **synchronize** our search in time between cameras

$$
\begin{array}{cc}
 & \text{cameras} \\
\text{cameras} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}
\end{array}
$$

Neighbours: cameras on which a car can appear in a reasonable time

Sync with different **time stamps**

*for this part we assume a perfect single camera tracking (gt) to focus in evaluate this multicamera task

II - Multi-target multi-camera (MTMC) tracking

MTMC: **Algorithm***

We start from the **MTSC** we obtained for each camera and we define a **adjacency matrix** per sequence.

We **synchronize** our search in time between cameras

$$\begin{array}{cc} & \text{cameras} \\ \text{cameras} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Neighbours: cameras on which a car can appear in a reasonable time

Temporal window

- Only consider **tracks within** a user defined **window of *n* seconds** around current track

*for this part we assume a perfect single camera tracking (gt) to focus in evaluate this multicamera task

II - Multi-target multi-camera (MTMC) tracking

MTMC: Algorithm*

We start from the **MTSC** we obtained for each camera and we define a **adjacency matrix** per sequence.

We **synchronize** our search in time between cameras

We look for a possible **match** in the **spatial and time windows**

$$\begin{array}{cc} & \text{cameras} \\ \text{cameras} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Neighbours: cameras on which a car can appear in a reasonable time

Active frames

Sync with different **time stamps**

Temporal window

This **reduces** a lot the **number of comparisons** (especially for seq 4)

Query

Candidates

*for this part we assume a perfect single camera tracking (gt) to focus in evaluate this multicamera task

II - Multi-target multi-camera (MTMC) tracking

MTMC: **Algorithm***

Query

Candidates

Set of **ROI images** (bbox containing the car) for the query in the query camera

Set of **ROI images** for each candidate in the camera we are looking for correspondences

Track Matching

- 1 on 1 matching algorithm
- Some of the features employed:
  - RGB Concat Histogram
  - 3D RGB Histogram
  - VGG16 (1)
  - TransReID (2)

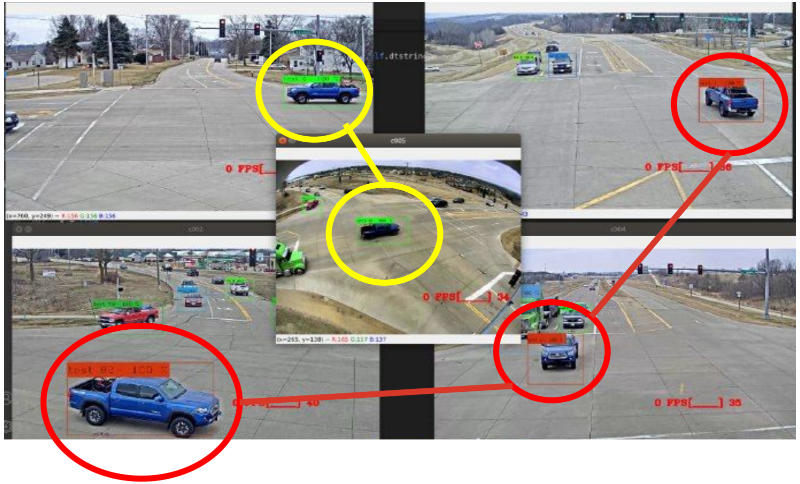
Match ID

Confidence

[1] Simonyan et al. "Very deep convolutional networks for large-scale image recognition." 2014
[2] He, Shuting, et al. "Transreid: Transformer-based object re-identification." 2021

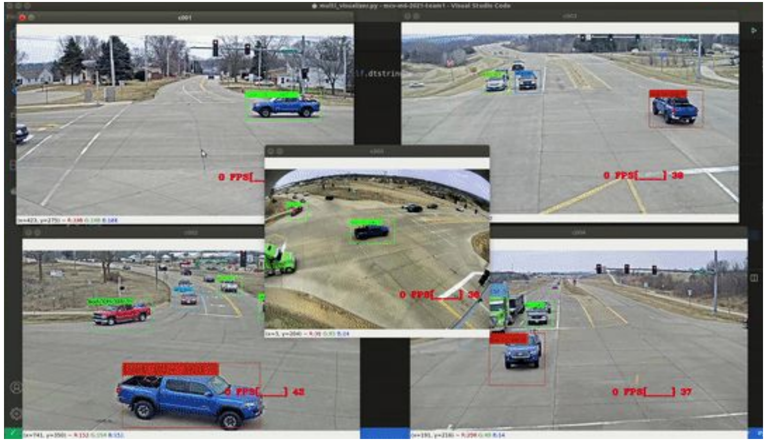
II - Multi-target multi-camera (MTMC) tracking

MTMC: **Qualitative results on S01**



- Matched in 3 and 2 cameras, but separately

- Car visible in all images



II - MTMC tracking - Color Histograms

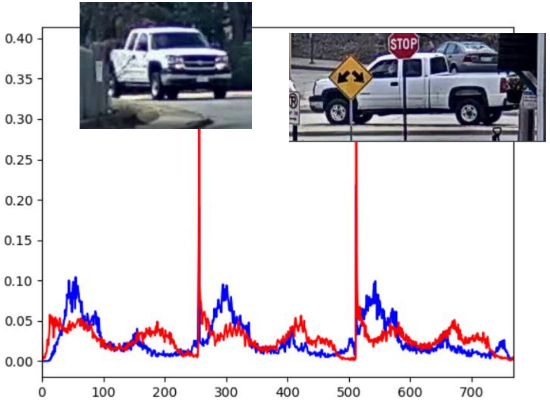
Features explored: **Histograms**

	IDF1		
SEQUENCE	**S01**	**S03**	**S04**
1D RGB Histogram	56.6	**54.2**	**45.8**
3D RGB Histogram	**58.1**	46.6	45.3

New Baseline

The boxes contain a **lot of background** around the car. Without vehicle segmentation it will greatly affect the histogram.

Even for the **same vehicle** the histogram changes a lot depending of the background and position of the car



Advantages:

- Easy to implement
- No training
- Fast

Drawbacks:

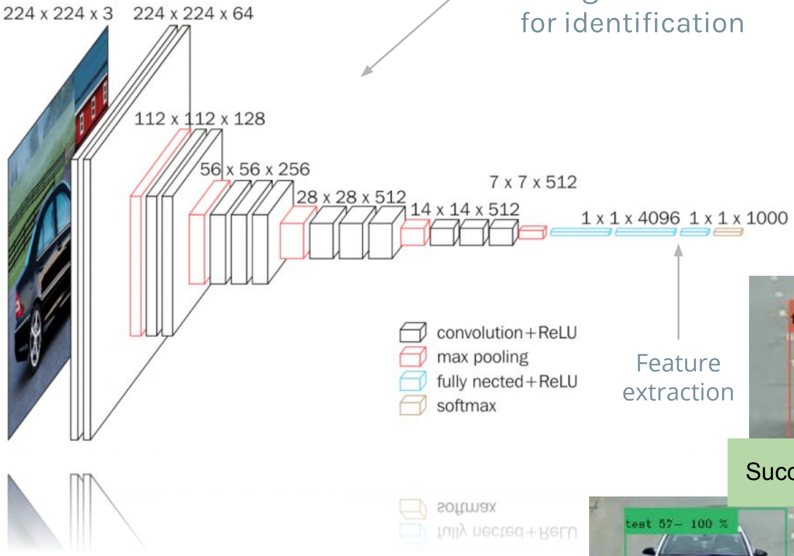
- Not very robust

II - MTMC tracking - CNNs

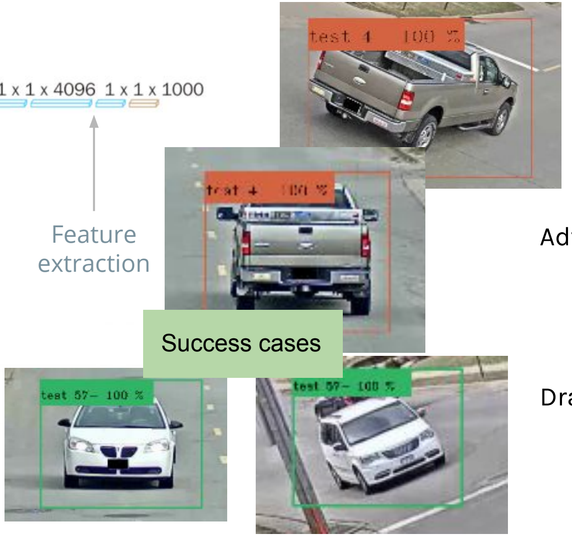
	IDF1		
SEQUENCE	**S01**	**S03**	**S04**
BASELINE	56.6	54.2	**45.8**
VGG_16_hell	51.8	55.7	45.5
VGG_16_l2	-	**58.4**	-

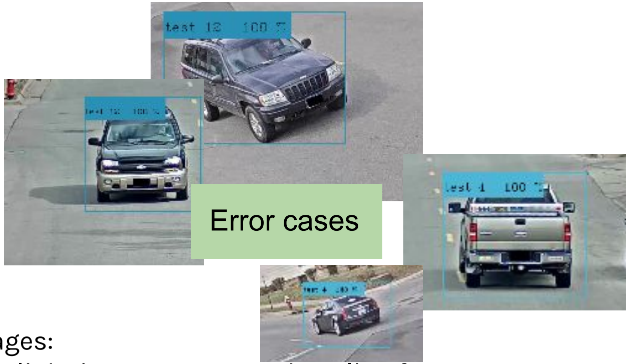
Features explored: **CNNs** (VGG16)

Learned features, though not necessarily for identification



Feature extraction





Advantages:

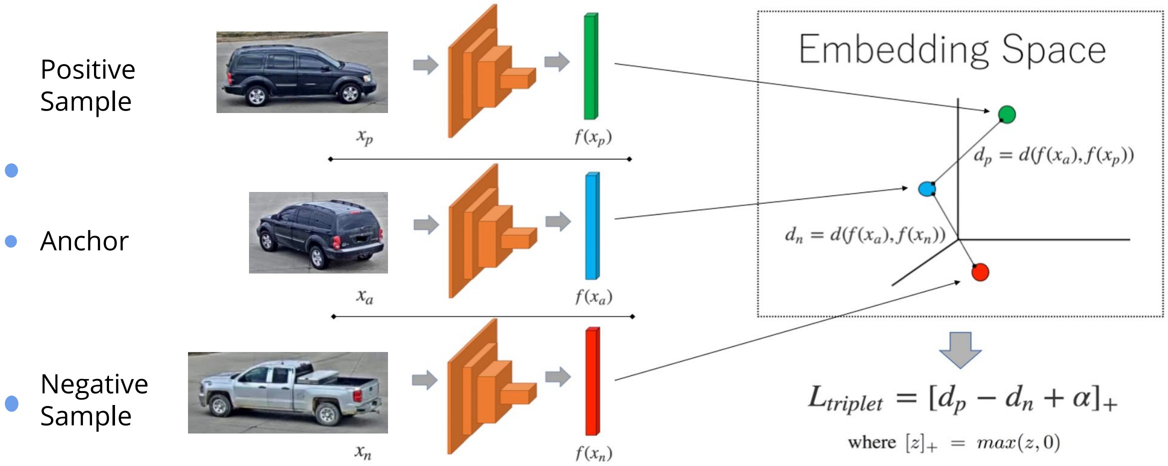
- Slight improvement on baseline for some sequences
- Trainable (fine tuning)

Drawbacks:

- Longer computation time
- Not trained with a focus on data separation (see Metric learning)

II - MTMC Tracking - Metric Learning

Objective: learn embeddings such that the anchor is closer to the positive example than it is to the negative example by some margin value.



Backbone Model: MobileNetV3 pre-trained on ImageNet as our

Training data: Sequences 01 and 04

Optimizer: Adam with lr=1e-4

Embedding Size:256

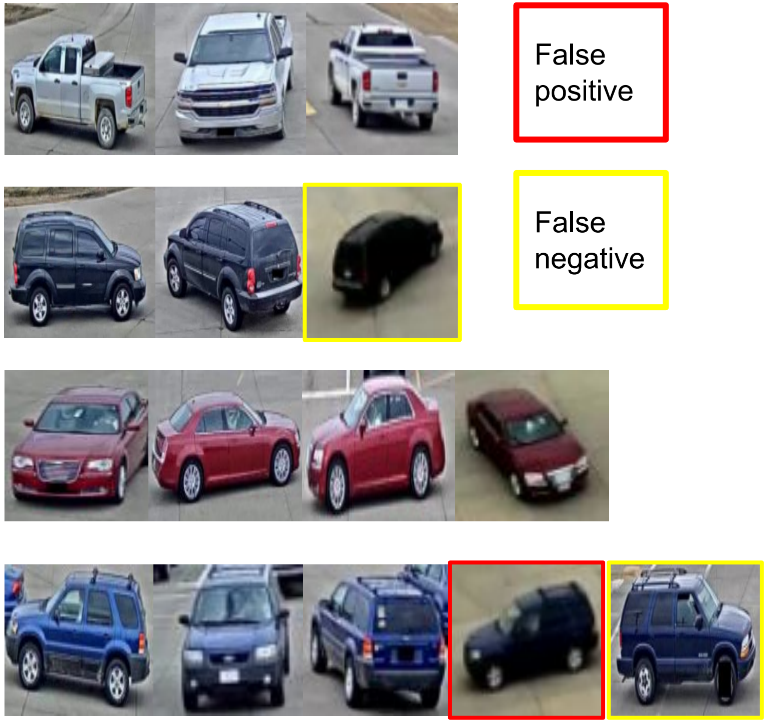
Strategy: Online Mining Triplet Loss[1]

Triplets are generated on the fly for each batch using hardest negative sample for each positive pair.

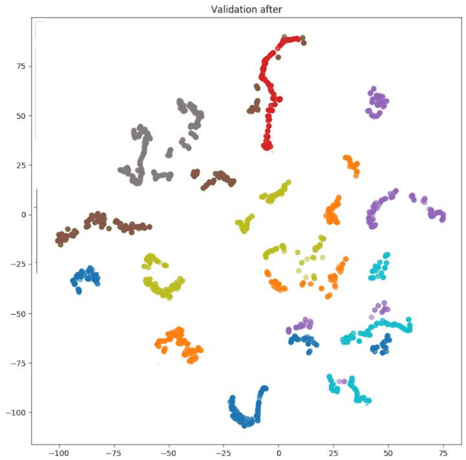
The training scripts were adapted from Adam Bielski's metric learning library.

1. FaceNet: A Unified Embedding for Face Recognition and Clustering, https://arxiv.org/abs/1503.03832

II - MTMC Tracking - Metric Learning



BackBone	Retrieval Results	
	Rank 1	Rank 5
MobileNetV3	63.1	75.8

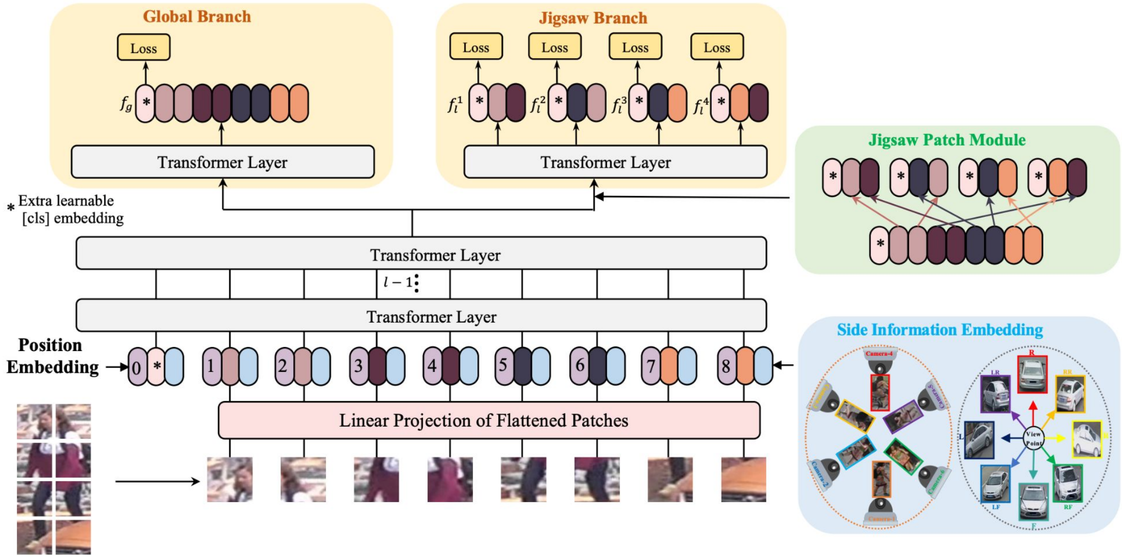


We also try to visualize the embeddings assuming that validation set has only 10 classes. We add a fully-connected layer with the number of classes and train the network for classification with softmax and cross-entropy and then we extract 2 dimensional embeddings from penultimate layer.

Embeddings generate using the tutorial [here]

II - MTMC tracking - Transformers

Features explored: **TransReId**[1]



	IDF1		
SEQUENCE	**S01**	**S03**	**S04**
BASELINE	56.6	54.2	45.8
Simplified	56.7	48.0	46.1
Intensive (all patches)	--	48.0	--

```
(base) group01@etse-75-51:~/M6/mcv-m6-2021-team1/week5$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             21898      mlow    taskC  group01  R    6:24:09      1 etse-75-51
             21897     mhigh    taskC  group01  R    6:24:41      1 etse-75-51
```

Re-arranges patch embeddings (shift and shuffle)

Cameras and viewpoints information

[1] He, Shuting, et al. "Transreid: Transformer-based object re-identification." 2021

II - MTMC tracking - Overall results

Metric	IDF1	IDP	IDR	Feature
SEQ 1	58.1	58.3	58.1	*3D RGB Hell*
SEQ 3	58.4	59.4	58.4	*VGG16 l2*
SEQ 4	45.8	45.8	45.8	*3D RGB Hell*
AVERAGE	54.1	54.5	54.1	

II - MTMC Tracking - Conclusions

- ➔ Re-identification is a much harder task than tracking.
- ➔ Object detection and tracking, like AP and IDF1, **metrics are tricky**
- ➔ Different camera configurations makes MTMC much harder (distortions, mismatches, different resolutions…)
- ➔ Future Work would be to exploit Transformers and Graph based architectures for tracking multiple objects