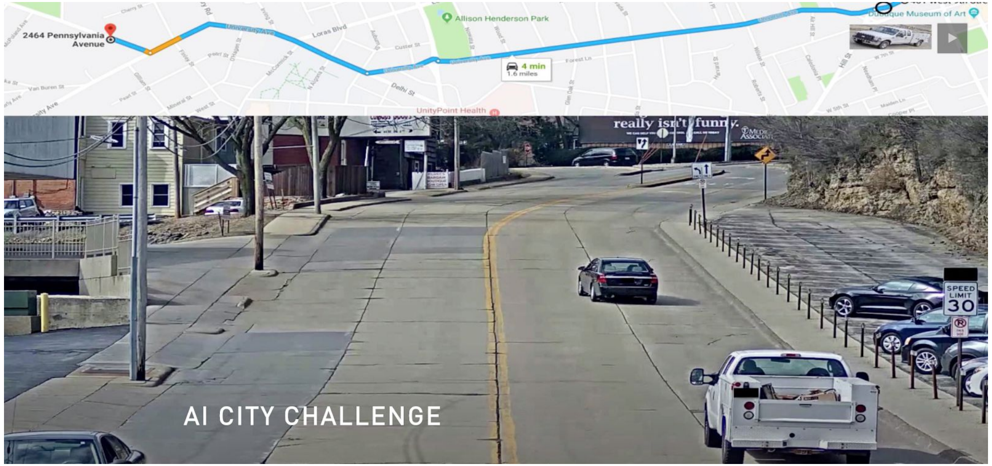
Video Surveillance for Road Traffic Monitoring

Master in Computer Vision Barcelona

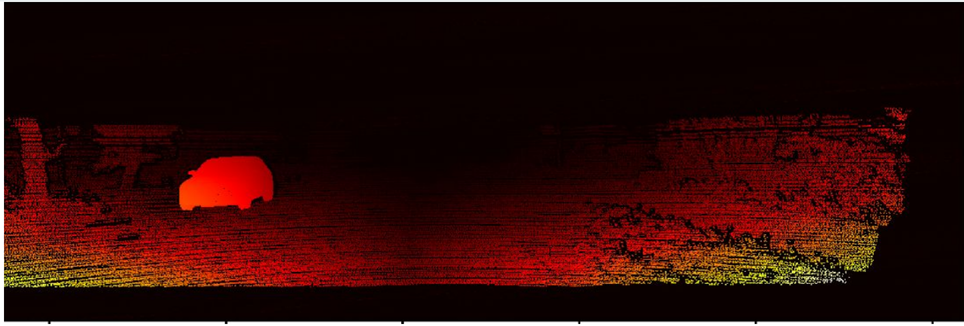
Marcos Melgar Segovia
Daniel Rodríguez Estévez
Pau Torras Coloma

AI CITY CHALLENGE

- Challenge focus on several traffic tracking:
  - car tracking
  - driver’s actions capture
  - automatic check-out
  - ...
- We will develop a traffic monitoring application during 5 weeks.



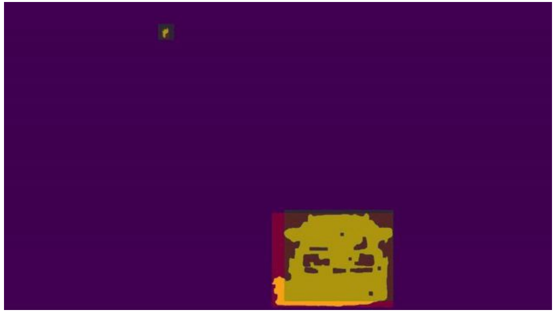
Week 1

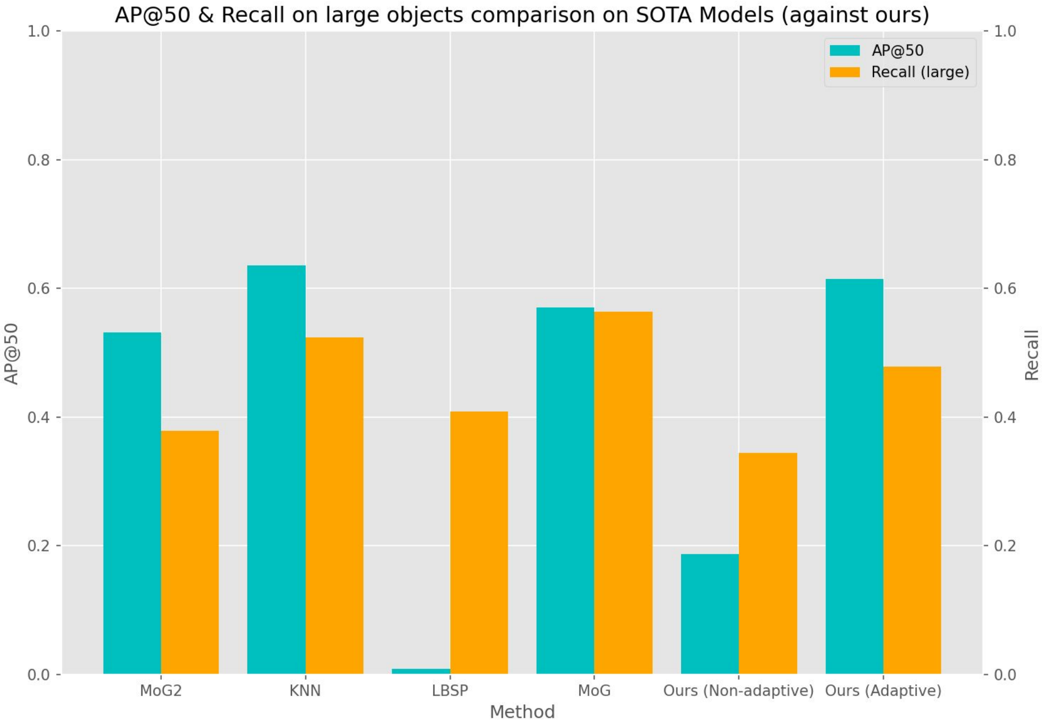


- Evaluation metrics implementation for detections
- Optical flow:
  - Evaluation metrics
  - Visualization

Week 2

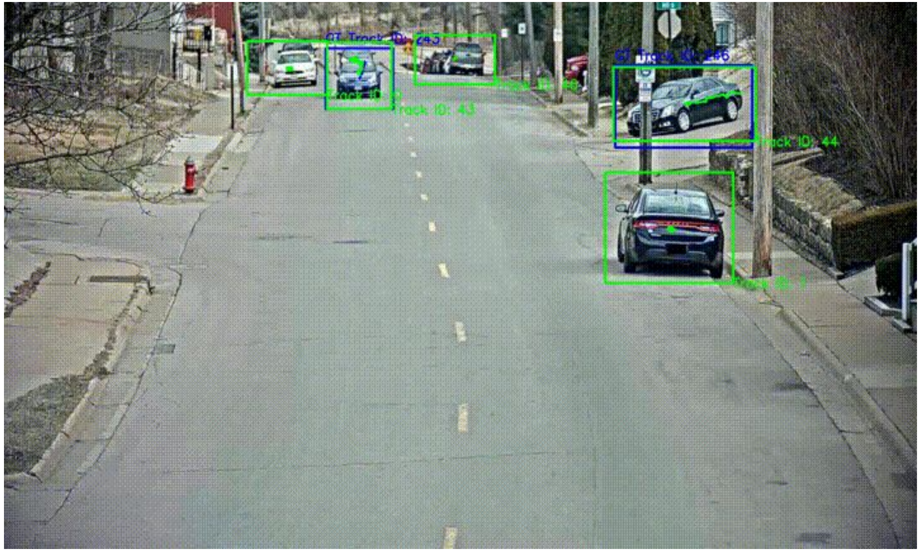
- Background extraction grayscale
    - Adaptive
    - Non-adaptive
- Background extraction color





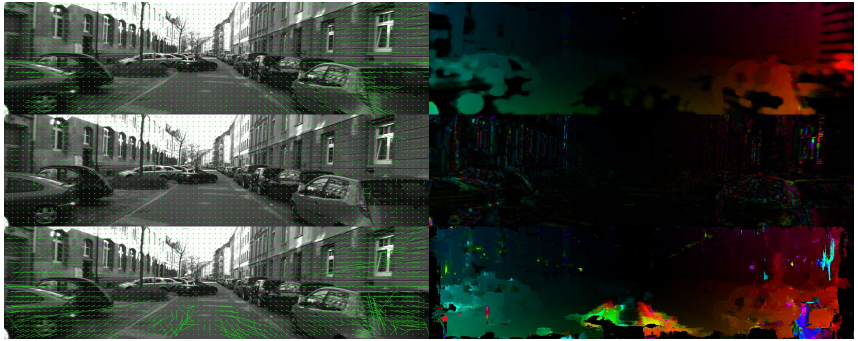
Week 3

- Object detection
  - Off-the-shelf
  - Fine tuning
- Object tracking
  - Maximum overlap
  - Kalman Filter



Fine tuning	RetinaNet + FPN	Faster-RCNN + FPN	Faster-RCNN + DC5	Faster-RCNN + C4
AP50	95.81	95.63	95.87	94.96
AP75	87.46	78.5	87.57	85.45
AP-bike	71.38	58.18	64.74	61.54
AP-car	86.55	87.36	84.68	86.21

Week 4



- Optical flow
    - Using block matching
    - State of the art methods
- Car tracking techniques in AI Cities sequences
- Incorporating Optical Flow into tracking

Week 5: CVPR 2020 AI City Challenge

Task 1: Multi-target single camera tracking.

For every vehicle, assign a unique individual ID.

Deal with parked cars, occlusions, pedestrians, …

Task 2: Multi-target multi-camera tracking

Assign a unique ID for every car that appears on different viewpoints.

Cresults_path

Week 5: CVPR 2020 AI City Challenge

Dataset:

- 40 cameras in real-world traffic surveillance environment.
- 195.03 minutes of videos.
- 666 vehicles.
- 5 different scenarios.

T1. Multi-target single-camera (MTSC) tracking

IDF1 (SEQ 3)			Camera							
Detector	**Tracking**	**Purge**	c010	c011	c012	c013	c014	c015	Average	Weighted avg
faster_c4	MaxOverlap	No	0.552	0.746	0.349	0.895	0.522	0.327	0.565	0.572
faster_c4		Yes	**0.966**	0.913	0.119	**0.926**	0.825	**1**	**0.792**	0.779
faster_dc5	MaxOverlap	No	0.574	0.448	0.372	0.867	0.499	0.228	0.498	0.507
faster_dc5		Yes	0.956	**0.916**	0.145	0.901	**0.885**	0.723	0.754	0.749
faster_fpn	MaxOverlap	No	0.499	0.471	0.376	0.826	0.517	0.241	0.488	0.497
faster_fpn		Yes	0.963	0.881	0	0.897	0.867	**1**	0.768	0.754
retina_fpn	MaxOverlap	No	0.163	0.135	0.095	0.203	0.178	0.06	0.139	0.141
retina_fpn		Yes	0.466	0.435	0.381	0.668	0.371	0.773	0.516	0.509
faster_fpn	MaxOverlap+OF	Yes	0.122	-	-	-	-	-	0.122	0.019
AdaptBckgExt	MaxOverlap	Yes	0.326	0.372	0.122	0.497	0.321	0.736	0.396	0.385
faster_c4	Kalman	No	0.686	0.747	0.425	0.582	0.664	0.343	0.575	0.578
faster_dc5	Kalman	No	0.577	0.566	0.446	0.63	0.55	0.363	0.522	0.526
faster_fpn	Kalman	No	0.599	0.706	0.44	0.606	0.646	0.395	0.565	0.569
retina_fpn	Kalman	No	0.689	0.396	**0.54**	0.526	0.654	0.757	0.594	0.587

T1. Multi-target single-camera (MTSC) tracking

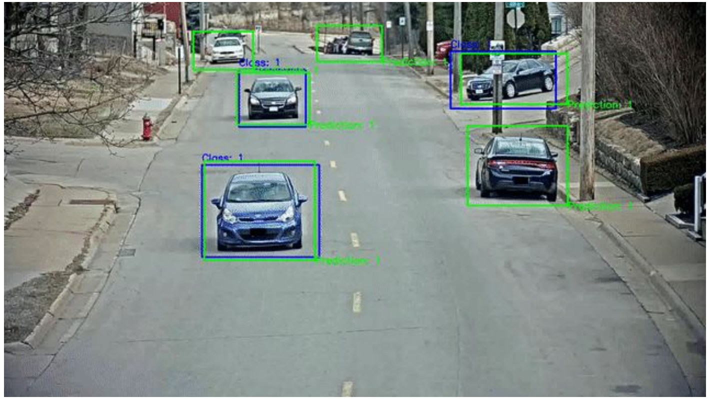
Purging allow us to increase upt to 40% the performance of the system.

Purge

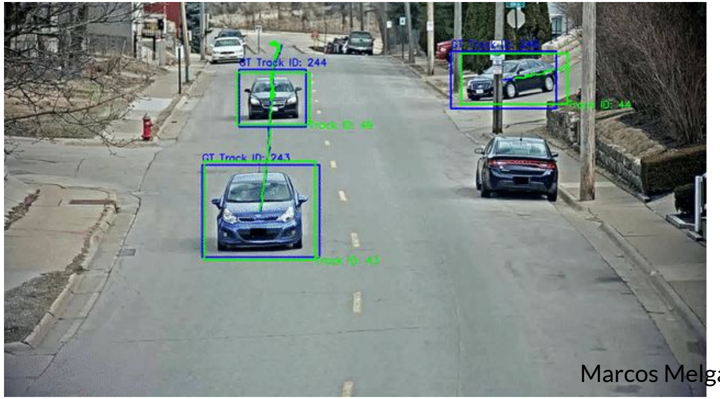
- Removes **parked cars**: tracks with displacement under 50 pixels.
- Removes tracks that did not stay **more than 20 frames** in the scene.
    - downgrades c12 results.

T1. Multi-target single-camera (MTSC) tracking

Before purge

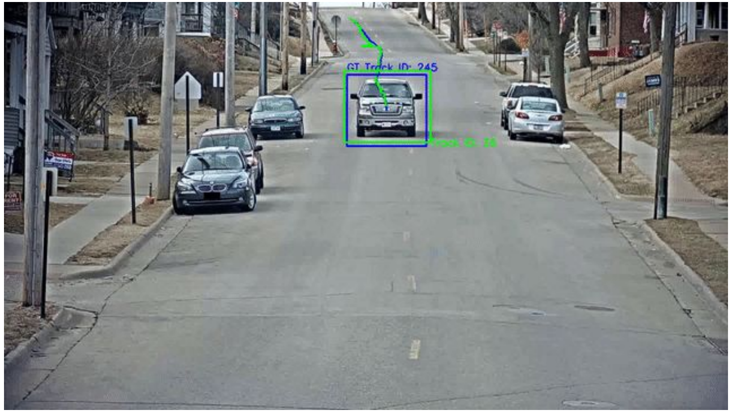


After purge



T1. Multi-target single-camera (MTSC) tracking

	Average (IDF1)
faster_c4 maxoverlap + purge	0.792
faster_fpn maxoverlap + purge	0.768



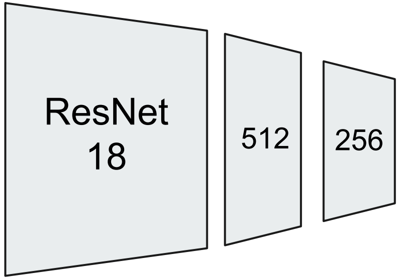
- Best results: MaxOverlap + purged (removes unwanted boxes by stability and length).

- Cam 012 downgrades our results.

T2. Multi-target multi-camera (MTMC) tracking

- Pytorch Metric Learning [GitHub]
- Triplet network
- Resnet18 + 2 linear layers
- Triplet margin loss (0.2)
- Semi hard mining
- Exponential lr decay
- Gradient clipping
- Hyperparameters:
  - lr = 0.0003
  - batch_size = 40
  - epochs = 100
- Train with S01 & S04, test for S03

PML
PyTorch Metric Learning



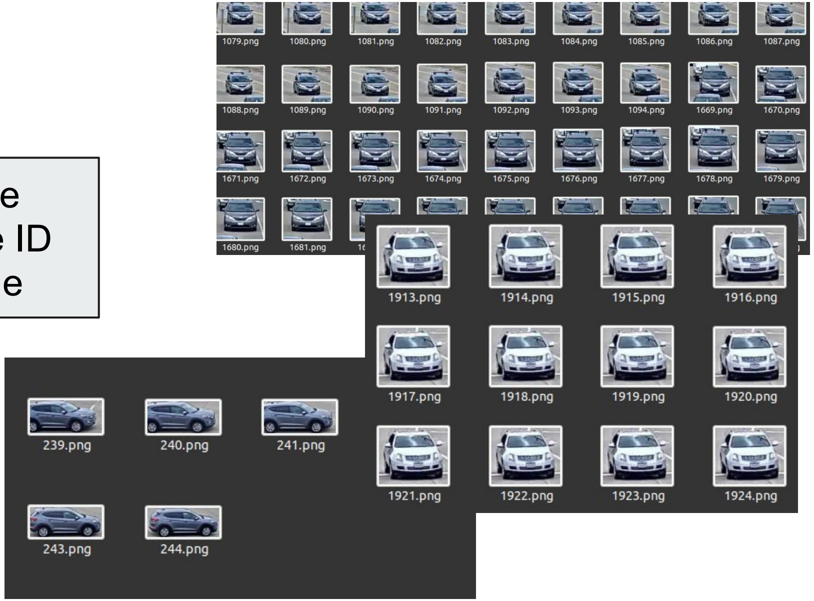
T2. Multi-target multi-camera (MTMC) tracking

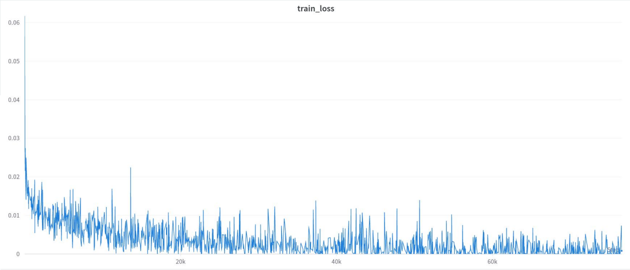
Dataset creation → Metric Learning training → ReID → Multi-camera evaluation

T2. Multi-target multi-camera (MTMC) tracking

Dataset creation

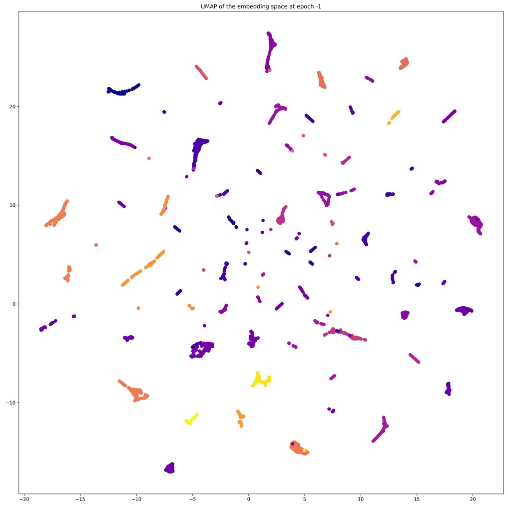
T1 tracking extraction → for every frame
for every unique ID
store car image



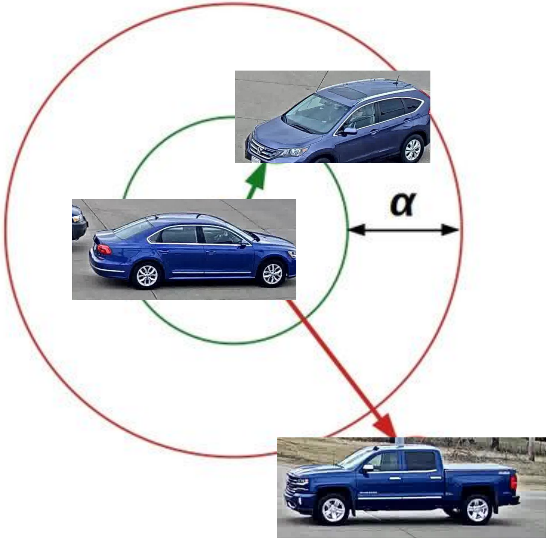


T2. Multi-target multi-camera (MTMC) tracking

Metric Learning training



Precision	0.9612
Recall	0.2264
F1	0.3665



T2. Multi-target multi-camera (MTMC) tracking

ReID

Once the model is trained -> compute embeddings -> same ID’s to embeddings with distance < threshold

- Comparing average of all features
- Comparing only with the largest size image crops
- Some folders shared the same ID’s number for different cameras
    - PML understood that they are the same car -> Assign unique id folders

T2. Multi-target multi-camera (MTMC) tracking

		Metrics				
Sequence	Model	IDF1	IDP	IDR	Precision	Recall
S01	Faster + FPN	0.490	0.461	0.532	0.777	0.909
	Faster + DC5	0.465	0.406	0.556	0.681	0.947
	Faster + C4	0.455	0.450	0.470	0.772	0.821
	Retina + FPN	–	–	–	–	–
S03	Faster + FPN	0.687	0.679	0.700	0.900	0.934
	Faster + DC5	0.674	0.668	0.681	0.898	0.917
	Faster + C4	0.679	0.651	0.713	0.862	0.948
	Retina + FPN	0.362	0.549	0.273	0.875	0.444
S04	Faster + FPN	0.453	0.501	0.417	0.912	0.764
	Faster + DC5	0.458	0.465	0.452	0.883	0.858
	Faster + C4	0.445	0.522	0.388	0.936	0.700
	Retina + FPN	0.311	0.411	0.254	0.872	0.552

T2. Multi-target multi-camera (MTMC) tracking

Example of correct pair

Example of incorrect pair

T2. Multi-target multi-camera (MTMC) tracking

- Metric Learning seems to work – results are coherent. However, **thresholds are difficult to set.**
- ReID not strong enough – More data?
- **Alternatives:**
  - Online embedding of Faster RCNN features
  - Plan B: Some averaging of high level features
- Working with metric learning is hard – Many runs, difficult to test results

Conclusions

- Single camera tracking give good results.
- Some image crops are really small and noisy.
- ReID task consumed a lot of time.
- Purge helped us increment the tracking performance, but did not help for environments like cam012.
- We had to rely on data augmentation in order to train the Metric Learning.
- Pre-training metric learning over other datasets (VehicleID, VeRi) would have helped us to not rely on data augmentation.
- Transformers for tracking as future work.