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

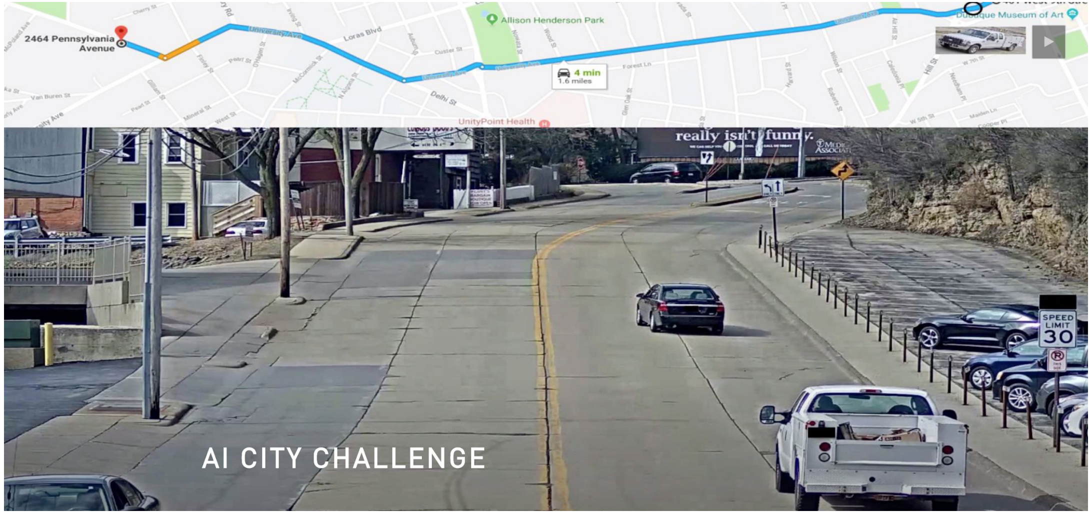
Week 4: Report (part2)

… continue from part1

Task 2: Multi-target single-camera (MTSC) tracking

Assess the quality of your best solution on SEQ3 in the CVPR 2022 AI City Challenge (Track 1):

“Participating teams will track vehicles across multiple cameras both at a single intersection and across multiple intersections spread out across a city. This helps traffic engineers understand journey times along entire corridors. The team with the highest accuracy in tracking vehicles that appear in multiple cameras will be declared the winner of this track. In the event that multiple teams perform equally well in this track, the algorithm needing the least amount of manual supervision will be chosen as the winner.”



Task 2.1: MTSC tracking - IDF1 (all teams)

Provide the results for your best technique

	IDF1 (SEQ 3)						
Camera	**c10**	**c11**	**c012**	**c013**	**c014**	**c015**	**Average**
Team 1	0.43	0.57	0.02	0.67	0.45	0.03	0.36
Team 2	0.46	0.19	0.44	0.18	0.49	0.03	0.30
Team 3	0.80	0.74	**0.66**	0.64	**0.76**	**0.43**	0.67
Team 4	0.24	0.47	0.56	0.37	0.45	0.02	0.35
Team 5	**0.81**	**0.82**	0.50	**0.76**	0.75	0.1	0.62
Team 6	0.25	0.06	0.01	0.47	0.39	0.004	0.14

Use implementation of IDF1/HOTA provided in TrackEval.

Task 2.1: MTSC tracking - HOTA (all teams)

Provide the results for your best technique

	HOTA (SEQ 3)						
Camera	**c10**	**c11**	**c012**	**c013**	**c014**	**c015**	**Average**
Team 1	0.43	0.36	0.04	0.53	0.45	0.01	0.30
Team 2	0.50	0.17	0.39	0.19	0.41	0.01	0.28
Team 3	0.76	0.63	0.54	0.48	0.57	0.31	0.54
Team 4	0.22	0.32	0.37	0.26	0.32	0.06	0.26
Team 5	0.65	0.61	0.36	0.59	0.60	0.16	0,5
Team 6	0.36	0.13	0.04	0.52	0.57	0.03	0.27

Use implementation of IDF1/HOTA provided in TrackEval.

Task 2.2: MTSC tracking - Discussion (Team X) [max 3 slides]

Task 2.2: MTSC tracking - Discussion (Team 1) [1/2]

Methodology

To obtain the previous results we do the following:

- Fine-tune **YOLOv8** with Seq01 and Seq04
  - We use cameras 03, 21, 26 and 40 for validation. The rest belong to the training subset.
- Generate the bounding boxes **predictions for all cameras in Seq03**
- Generate tracks using **tracking by Overlap** (we did not have time to compute the optical flow, although it is implemented :( )
- Evaluate the provided GT and the generated tracks, using **TrackEval**

Analysis on quantitative results results

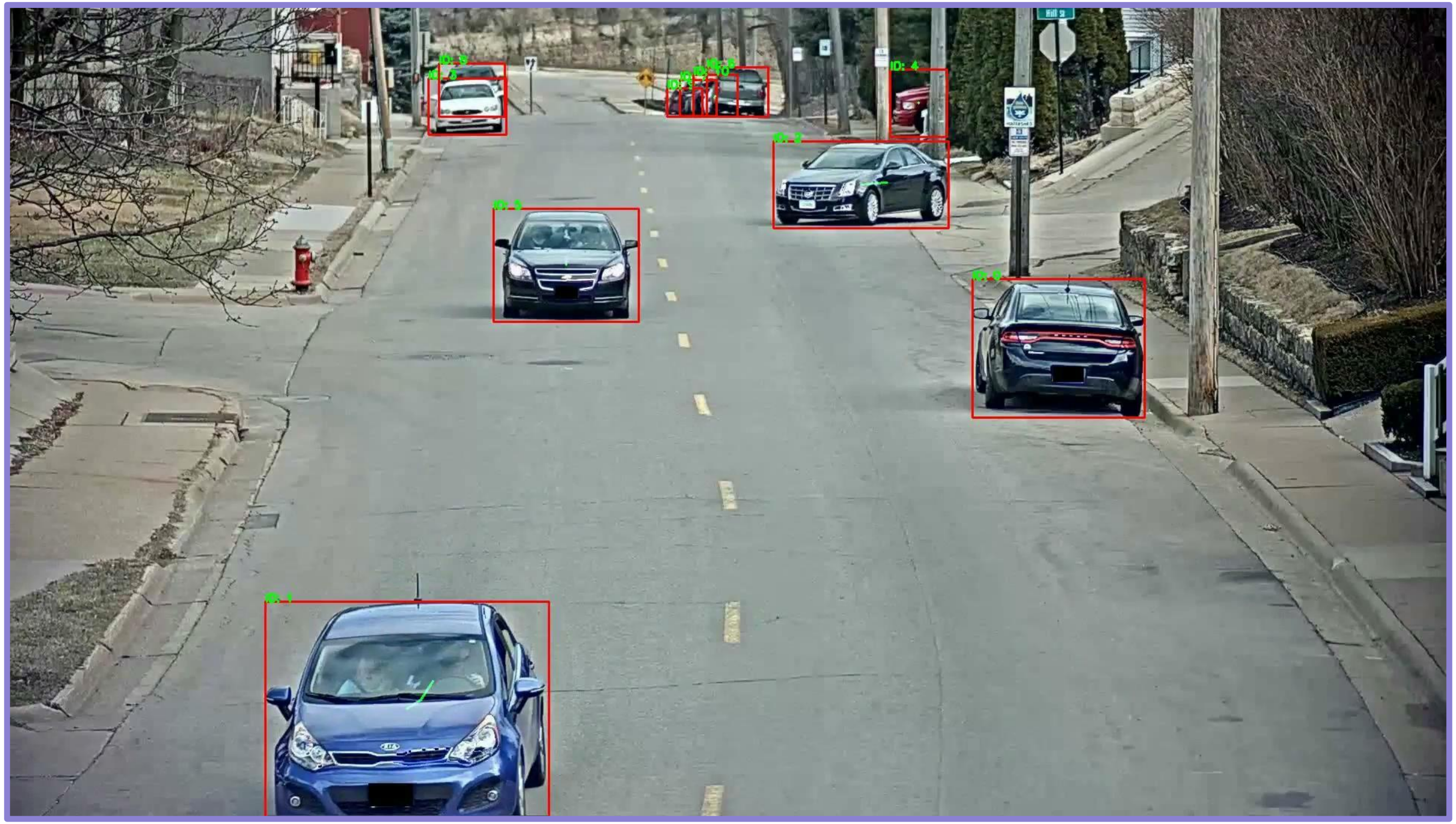
- The worse results were obtained on sequences with parked cars. Since, the GT did not consider them.
- The tracking metrics where low due to the difficulty to detect the proper bounding boxes.
- Postprocessing to the detected bounding boxes could be employed to remove parked cars
- Additionally, the ROI file provided in the annotations could delimit our predictions too

	Seq03						
Camera	**c010**	**c011**	**c012**	**c013**	**c014**	**c015**	**Average**
HOTA	0.43	0.36	**0.04**	0.53	0.45	0.01	0.30
IDF1	0.43	0.57	**0.02**	0.67	0.45	0.03	0.36

c015

c012

Task 2.2: MTSC tracking - Discussion (Team 1) [2/2]



- Again, the overlap tracker struggles on areas where we find parked cars or vehicle crossings

Task 2.2: MTSC tracking - Discussion (Team 2) [1/3]

Tracking method	Model	Optical Flow	HOTA	IDF1
Kalman	faster-finetune	-	**0,6758**	**0,6940**
Kalman	faster-finetune	Block Matching	0,4889	0,5009
Kalman	faster-finetune	Unimatch	0,4888	0,5008
Overlapping	faster-finetune	-	**0,6626**	**0,6387**
Overlapping	mask-rcnn	-	0,5158	0,4579
Overlapping	ssd512	-	0,3742	0,3377
Overlapping	yolo3	-	0,3812	0,3689

Against the odds, we could not find an approach of incorporating Optical Flow into the tracking in order to improve the results with respect to the Kalman filter (see Table Section 1.3 and summarized above). This also applies to SEQ03 video for al cameras.

As we will see in the following slides the main issue when Optical Flow is added to the tracking procedure, we add all the outliers coming from the OF.
In the sample video above we observe that OF approaches may present the following drawbacks:

1. Inaccurate optical flow due to occlusions, illumination changes, and camera motion, resulting in incorrect predictions of object motion.
2. Mismatched motion models between the Kalman filter linear motion model and the actual non-linear or complex motion patterns of objects in the video sequence.
3. Increased computational complexity and numerical errors due to additional computations required for incorporating OF, resulting in additional sources of error to the tracking.
4. Lack of robustness to noise, especially in regions with low textures or areas with rapid motion, introducing error in the tracking and decreasing performance.

Also, note that as unimatch is eager to consume the whole cluster's memory and as SEQ03 - C011, 012, 013 are slightly bigger images, there's no option but go straight to the Block Matching approach.

Task 2.2: MTSC tracking - Discussion (Team 2) [2/3]

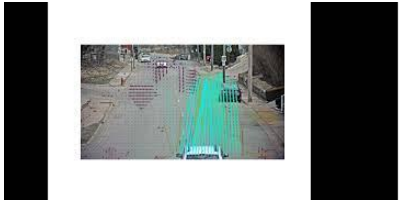
Although there is a huge improvement in terms of Optical Flow when unimatch is used instead of block matching, the improvement disappears on the tracking challenge.

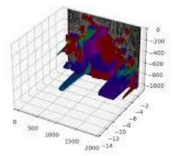
On the results we observe that although unimatch is way better; it presents lot of ticking false positives that we will analyze in the next slide. We suspect that this continuous presence of False Positives is due to the sensitivity of the method to small movements in the camera. As Block Matching matches by local similarities, it should be robust to this kind of similarity transformations.

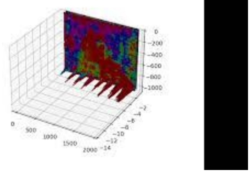
Block Matching

Unimatch

C010







The surface of the OF during the sequence tells us that while in Block Matching there's the possibility of 0 movement presence (denoted by transparency) in Unimatch there's a certain constant movement offset.

This makes us suspect of a continuous noise on the camera for both illumination and physical movement (wind or vibrations).

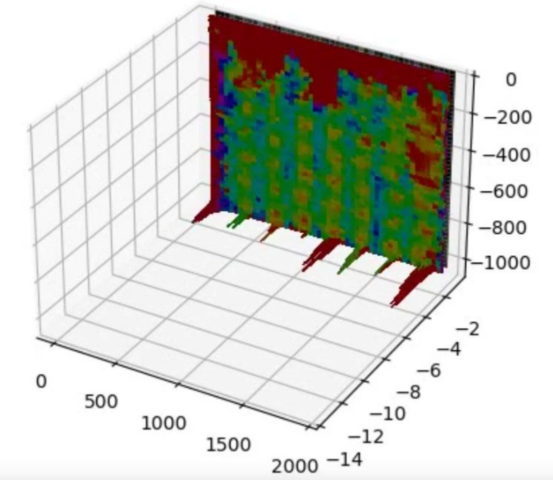
C013

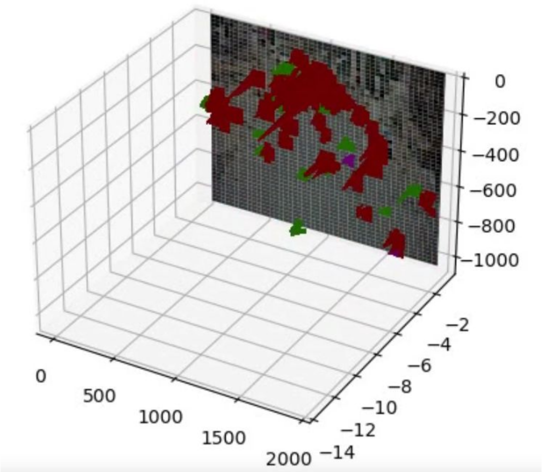
For C013, unimatch consumes all the memory of the cluster.
Which is an additional constraint on using a proper OF approach to the tracking algorithm-.

reflections on solid objects may also cause a huge failure in the tracking

Task 2.2: MTSC tracking - Discussion (Team 2) [3/3]

While in Block Matching there are static regions, in unimatch there's not. Kind of errors from unimatch compensate its benefits and, therefore, we find no difference on the tracking.

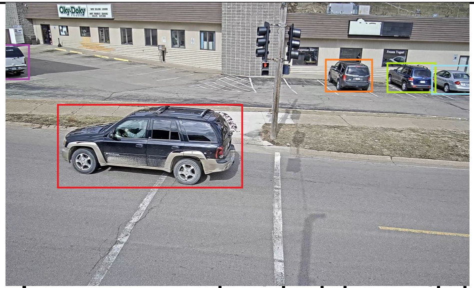




The presence of this kind of errors causes failure in the Tracking with OF due to mismatches and numerous False Positive motion vectors in the borders.

Most of the ticking false positives are matches with the pavement marks which appears on the borders constantly. So there's an additional problem with staticity.

Task 2.2: MTSC tracking - Discussion (Team 3) [1/3]



- We have used the Lucas-Kanade OpenCV implementation method because it gives us the best trade-off between computation time and accuracy. Due the fact that we have limited time and a large amount of experiments we decided that it was the best option.

- We started by using and iterating with the other two methods (Block Matching and Pyflow) but we found that the speed was extremely slow, so we decided to switch to Lucas-Kanade. Despite we used Lukas-Kanade method, we have struggled in order to finish all the experiments. The total computation time for all the experiments was 18h without taking into account the failed executions.

- To speed up the computation, we have used different CPUs to accelerate the process of obtaining results, and as we can see, the i9900K processor is the fastest, but the one with the best consumption in relation to computing capacity is the Apple M1.

CPU	i7 6700HQ	i9 9900K	Apple M1
Time 1 execution (min)	46	**23**	25
Power Consumption	45W	69W	**20W**

Task 2.2: MTSC tracking - Discussion (Team 3) [2/3]

For car detection, we have used YoloV3 and YoloV5.
In the comparison videos, it was observed that YOLOv5 tends to exhibit better stability and faster performance in comparison to YOLOv3. This means that YOLOv5 is more reliable and consistent in its object detection capabilities while also running at a quicker pace than its predecessor.

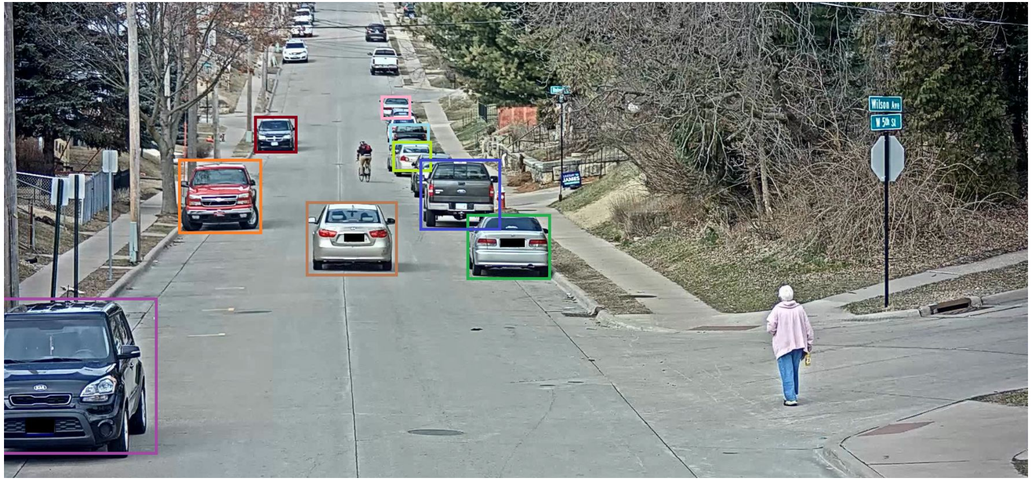
YoloV5 implementation

Cam 10	**IDF1**	**HOTA**
YOLOV3	0.76	0.69
YOLOV5	0.80	0.76

YoloV3 implementation

Task 2.2: MTSC tracking - Discussion (Team 3) [3/3]

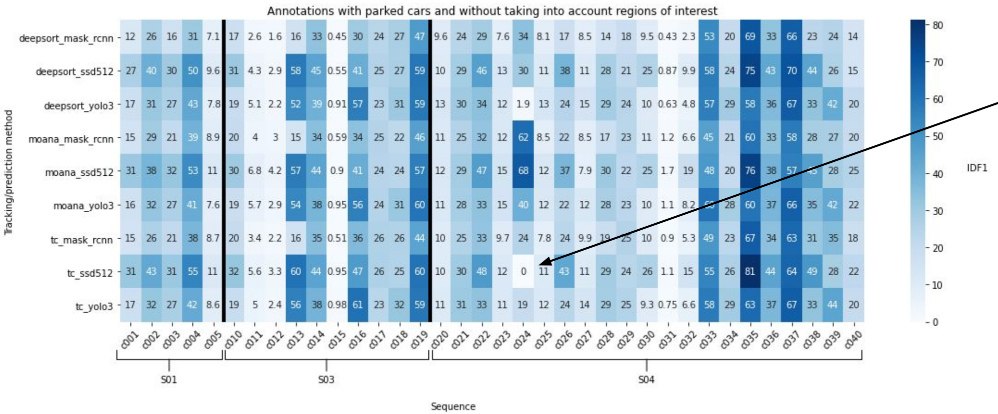
Cam 15 was found to have the lowest IDF1 and HOTA scores in comparison to the other cameras. This can be attributed to the presence of multiple parked cars that overlap and occluded by trees, making it difficult for the object detection algorithm to accurately track them. Additionally, the algorithm often struggles to correctly label cars that are situated far away from the camera. These factors contribute to the lower performance of Cam 15 in comparison to the other cameras.



Parked cars overlap in cam 15

Task 2.2: MTSC tracking - Discussion (Team 4) [1/3]

We start by looking at the given baseline predictions from the 2022 AICity challenge results

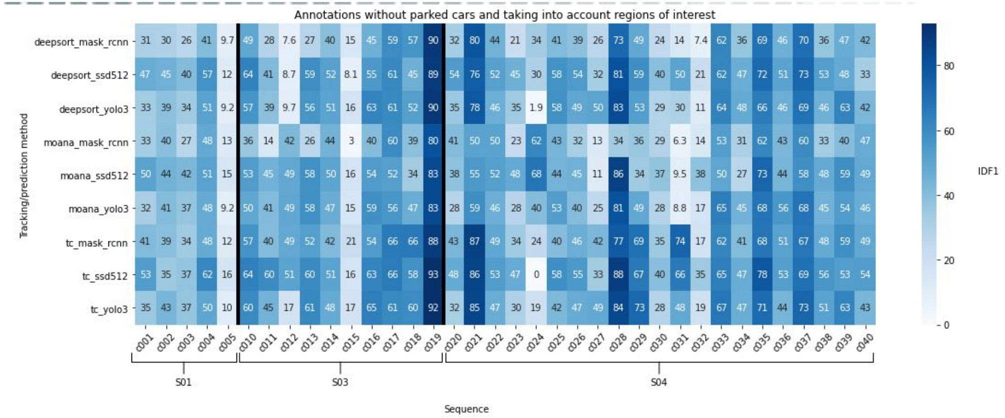


We can observe that some baselines do not have any prediction

mcv-m6-2023-team4 > datasets > AITrack > train > S04 > c024 > mtsc > mtsc_tc_ssd512.txt

1

Furthermore, IDF1 baseline results are really bad for almost all sequence cameras



Also, as expected, our postprocessing methods (explained on the next slide) that involve removing parked cars and taking into account the regions of interest improve the IDF1 of all the baseline methods

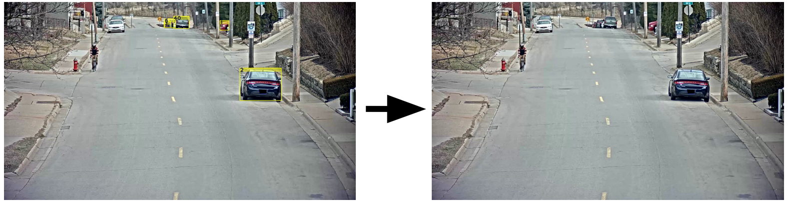
Task 2.2: MTSC tracking - Discussion (Team 4) [2/3]

From now on

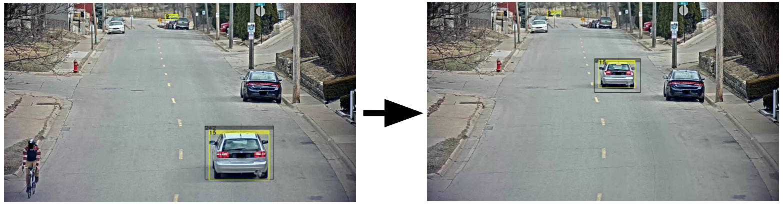
Groundtruth

Predictions

Postprocessing methods



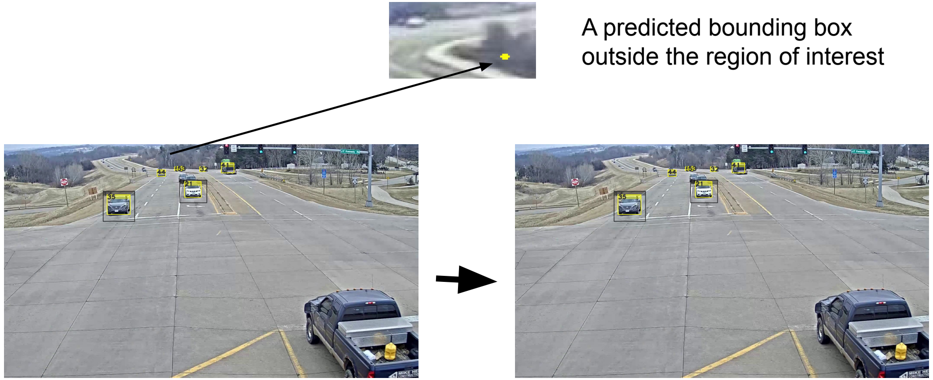
Objects that stay almost on the same position are removed



Objects with high speed remain

We remove parked cars from the predictions by using the speed and distance of the detected objects. We set a threshold speed and a threshold distance to choose which slow or not moving objects are removed

A predicted bounding box outside the region of interest



We remove predicted objects that are not inside the region of interest by checking if the bounding box center is inside or outside of the roi

Task 2.2: MTSC tracking - Discussion (Team 4) [3/3]

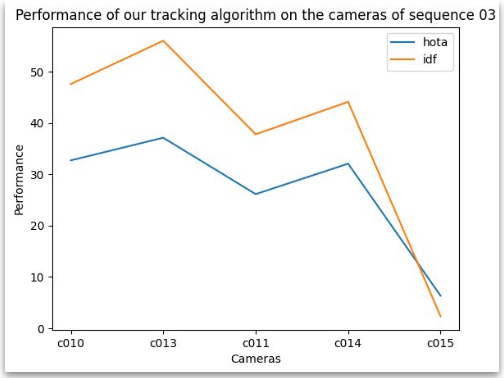
Since object groundtruth come from multiple cameras they are not suitable for training to improve the detectors (on predicting cars because some cars are not on groundtruth)

Different cameras of sequence #03

HOTA results + pipeline

Run pretrained object detection neural network (faster_rcnn finetuned from last week) on video → Run tracking algorithm (maximum overlap) on car detections → Postprocessing

A tracking algorithm pipeline



Some results for cameras of sequence 03. We can observe that our results are not quite good, perhaps due to the groundtruth txt files which do not involve all the cars

Task 2.2: MTSC tracking - Discussion (Team 5) [1/3]

- Two versions of the DETR object detector with two different tracking algorithms were tested. The first one used pretrained DETR weights, while the second one was fine-tuned on S01 and S04 sequences. Then, our method using optical flow (GMFlow for optical flow estimation) and SORT were applied to these two object detector outputs. Additionally, post-processing was performed to remove tracks of static objects (there were not in the ground-truth).
- To remove static objects, we set a threshold based on the mean of the standard deviation of the tracked object bounding box centers' mean coordinates.

In the next slide, all the results will be displayed.

Fine tune DETR + OF tracking

Pre Trained DETR + OF tracking

Fine tune DETR + SORT

Winner

Pre Trained DETR + SORT

Task 2.2: MTSC tracking - Discussion (Team 5) [2/3]

STATIC OBJECTS REMOVED

HOTA	C010	C011	C012	C013	C014	C015	AV
FT + OF	0.65	0.61	0.36	0.59	0.60	0.16	.49
PT + OF	0.36	0.19	0.23	0.19	0.33	0.06	.22
FT + SORT	**0.75**	**0.49**	**0.57**	**0.52**	**0.60**	**0.19**	**.52**
PT + SORT	0.38	0.24	0.31	0.28	0.35	0.07	.27

IDF1	C010	C011	C012	C013	C014	C015	AV
FT + OF	0.81	0.82	0.50	0.76	0.75	0.1	.617
PT + OF	0.58	0.30	0.28	0.26	0.46	0.02	.316
FT + SORT	**0.90**	**0.65**	**0.79**	**0.75**	**0.79**	**0.11**	**.665**
PT + SORT	0.57	0.30	0.33	0.35	0.48	0.03	.343

WSO

AV
.318
.07
.32
.08

AV
.361
.051
.352
.058

The effect of two factors on the results can be observed in the presented tables. The first factor is fine-tuning, which significantly improves the results. The second factor to consider is the elimination of static objects, which can be primarily attributed to the omission of static objects in the ground truth of the dataset, resulting in a decrement in the results' quality. Therefore, it is concluded that both fine-tuning and the elimination of static objects are important factors to consider when obtaining optimal results in tasks related to object detection + tracking. Our optical flow based method performed similarly to the SORT algorithm.

FT: FINE TUNED DETECTOR
PT: PRETRAINED DETECTOR
OF: OPTICAL FLOW TRACKING
WSO: WITH STATIC OBJECTS

Task 2.2: MTSC tracking - Discussion (Team 5) [3/3]

Fine tuned obj detector + static obj removed

Pretrained obj detector + static obj removed

Fine tuned DETR + Static objects removed + OF

Fine tuned DETR + Static objects considered + OF

We can observe that by fine-tuning the object detector, we can limit our focus to only vehicles, as only vehicles are present in the ground truth annotations of the sequences used for training. Similarly, during evaluation on sequence s03, which was not used during training, the results improved as the model is only considering vehicles.

As this task and the database focus on detecting moving objects, we initially considered detected objects that were stationary as static. Furthermore, the ground truth of the database only accounted for moving objects. Consequently, the results obtained without removing the static objects were considerably worse compared to the alternative approach. Therefore, we can conclude that removing static objects is vital in achieving optimal results for tasks related to object detection in motion, as seen in the last table where removal was applied.

OUR PIPELINE

Fine Tuning → DETR Obj Detection → Tracking Algorithm → Evaluation

DETR Obj Detection: Fine tuned / pretrained

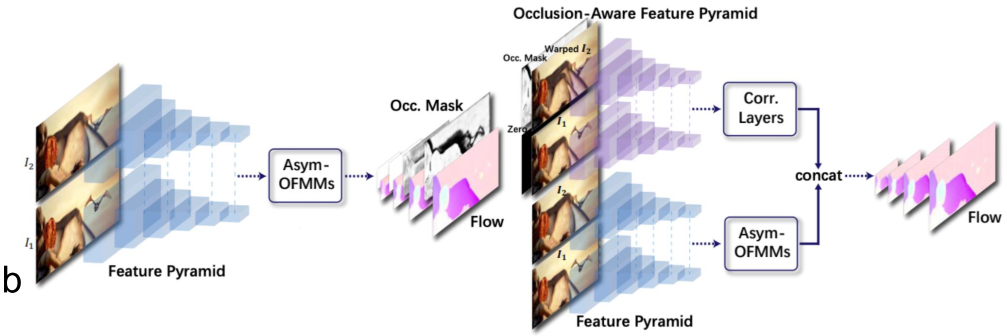
Tracking Algorithm: OFlow / SORT

Evaluation: TrackEval

Task 2.2: MTSC tracking - Discussion (Team 6) [1/3]

Methodology

- We are using **Yolo_V3** detections provided b **Dataset**.
- **Maskflownet** (Best algorithm in the previous tasks)
- **No training** is needed. GPU is required to reduce time.



Remember: MaskFlowNet is a two-stage framework designed for object segmentation and motion estimation. The first stage, MaskFlowNet-S, is an end-to-end network that handles initial estimations. The second stage consists of a cascaded network with dual pyramids, which refines the results from the first stage to improve accuracy and precision.

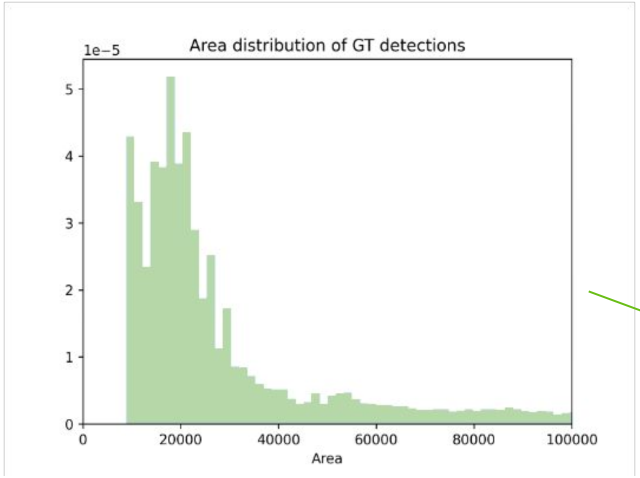
Hyperparameter

- Confidence Threshold: 0.6
- IoU Threshold: 0.5

Task 2.2: MTSC tracking - Discussion (Team 6) [2/3]

Why bad results?

	SEQ 03						
Camera	**c010**	**c011**	**c012**	**c013**	**c014**	**c015**	**Average**
HOTA	36.1	13.1	4.43	52.13	57.70	3.52	26.53
IDF1	25.4	5.58	0.95	46.94	39.26	0.35	13.84



- Ground truth data not consider small detections.
- Parked cars from detections were not removed.
- In detection data, a lot of more detections provided than GT.
- In last week, we provided detections with better performance than detections that are provided by these datasets. So we should use a custom detection algorithm for each video.

Task 2.2: MTSC tracking - Discussion (Team 6) [3/3]

Example Camera c015:
Bad Detections
Bad tracking

Not detected

id: 24 id:26

Task 2: MTSC tracking (feedback)

Team ID	Feedback
Team 1	Analysis of failure due to parked cars Finetuned Yolov8 + overlap
Team 2	OF does not help in MTSC. Are results provided for OF only? Maybe your best also could be reported? Any camera where your algorithm fails, like c15?
Team 3	Analysis of failure due to parked cars Good results
Team 4	Bonus point: Always good to check baselines first Remove object outside region of interest. Any camera where your algorithm fails, like c15?
Team 5	Finetuned DETR + SORT best results Removal of static objects. Why so bad results of c15? Good results
Team 6	IDF1 values do not match initial table, why? Comment on c15? why the bad tracking?

(optional) Task 3: MTSC tracking - Other sequences

If computation capability allows it, provide test results for SEQ 1 & SEQ 4, considering the other two as training data.

	Time (min.)	# cam.	# boxes	# IDs	Scene type	LOS
1	17.13	5	20,772	95	highway	A
2	13.52	4	20,956	145	highway	B
3	23.33	6	6,174	18	residential	A
4	17.97	25	17,302	71	residential	A
5	123.08	19	164,476	337	residential	B
total	195.03	40	229,680	666		

	Time (min.)	# cam.	# boxes	# IDs	Scene type	LOS
1	17.13	5	20,772	95	highway	A
2	13.52	4	20,956	145	highway	B
3	23.33	6	6,174	18	residential	A
4	17.97	25	17,302	71	residential	A
5	123.08	19	164,476	337	residential	B
total	195.03	40	229,680	666		

(optional) Task 3.1: MTSC tracking - SEQ1

Provide the results for your best technique

	IDF1 (SEQ 1)					
Camera	**c001**	**c002**	**c003**	**c004**	**c005**	**Average**
Team 1						
Team 2	0.23	0.21	0.23	0.37	0.22	0.25
Team 3						
Team 4	0.22	0.27	0.31	0.28	0.32	0.27
Team 5	68.78	63.42	54.93	56.7	64.5	61.7
Team 6	0.17	0.28	0.23	0.39	0.03	0.22

Use implementation of IDF1/HOTA provided in TrackEval.

(optional) Task 3.1: MTSC tracking - SEQ1

Provide the results for your best technique

	HOTA (SEQ 1)					
Camera	**c001**	**c002**	**c003**	**c004**	**c005**	**Average**
Team 1						
Team 2	0.20	0.22	0.25	0.29	0.19	0.23
Team 3						
Team 4	0.24	0.42	0.43	0.33	0.46	0.31
Team 5	55.1	53	45.8	46.9	49.1	50
Team 6	0.42	0.55	0.49	0.50	0.25	0.44

Use implementation of IDF1/HOTA provided in TrackEval.

(optional) Task 3.2: MTSC tracking - SEQ4

Provide the results for your best technique

	IDF1 (SEQ 4)												
Camera	**c16**	**c17**	**c18**	**c19**	**c20**	**c21**	**c22**	**c23**	**c24**	**c25**	**c26**	**c27**	**c28**
Team 1													
Team 2	0.33	0.27	0.27	0.43	0.11	0.23	0.19	0.25	0.09	0.10	0.24	0.17	0.20
Team 3													
Team 4	37.2	62.3	55.8	87.1	43.2	58.3	48.7	42.3	53.1	42.3	43.0	25.5	48.0
Team 5	84.5	73	37.2	91.2	54	95.7	92	63.4	49.2	44	86.4	59	54
Team 6	55.23	21.30	18.22	46.53	11.86	34.75	36.07	7.58	10.72	9.60	34.06	10.99	17.17

Use implementation of IDF1/HOTA provided in TrackEval.

(optional) Task 3.2: MTSC tracking - SEQ4

Provide the results for your best technique

	IDF1 (SEQ 4)												
Camera	**c29**	**c30**	**c31**	**c32**	**c33**	**c34**	**c35**	**c36**	**c37**	**c38**	**c39**	**c40**	**Average**
Team 1													
Team 2	0.29	0.07	0.24	0.04	0.31	0.24	0.19	0.25	0.33	0.25	0.32	0.22	0.24
Team 3													
Team 4	40.1	40.2	19.3	27.2	59.8	35.0	57.8	43.9	57.8	47.2	51.0	38.0	46.6
Team 5	71.1	73.9	54	71.5	60.7	51.1	67.4	45.4	65.8	64.3	73.3	63.9	68.1
Team 6	22.1	8.373	0.294	6.88	51.95	24.22	44.24	33.64	53.13	30.90	44.24	21.47	22.329

Use implementation of IDF1/HOTA provided in TrackEval.

(optional) Task 3.2: MTSC tracking - SEQ4

Provide the results for your best technique

	HOTA (SEQ 4)												
Camera	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	c26	c27	c28
Team 1													
Team 2	0.31	0.24	0.23	0.46	0.09	0.24	0.18	0.26	0.11	0.11	0.18	0.19	0.20
Team 3													
Team 4													
Team 5	70.3	58.7	21.2	77.1	37	81.4	77.9	46.3	31.6	27.7	67.6	46.6	48.1
Team 6													

Use implementation of IDF1/HOTA provided in TrackEval.

(optional) Task 3.2: MTSC tracking - SEQ4

Provide the results for your best technique

	HOTA (SEQ 4)												
Camera	**c29**	**c30**	**c31**	**c32**	**c33**	**c34**	**c35**	**c36**	**c37**	**c38**	**c39**	**c40**	**Average**
Team 1													
Team 2	0.27	0.05	0.28	0.05	0.33	0.23	0.15	0.29	0.31	0.24	0.29	0.19	0.24
Team 3													
Team 4													
Team 5	62.6	51.4	38.7	47.5	55.6	49	62.6	40	56.8	48.9	57.5	43.6	52.2
Team 6													

Use implementation of IDF1/HOTA provided in TrackEval.

(optional) Task 3.2: MTSC tracking - SEQ4 (move values)

Provide the results for your best technique

	IDF1 (SEQ 4)																						
Camera	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	c26	c27	c28	c29	c30	c31	c32	c33	c34	c35	c36	c37	c38
[Team 1]()																							
[Team 2]()	0.33	0.27	0.27	0.43	0.11	0.23	0.19	0.25	0.09	0.10	0.24	0.17	0.20	0.29	0.07	0.24	0.04	0.31	0.24	0.19	0.25	0.33	0.25
[Team 3]()																							
[Team 4]()	37.2	62.3	55.8	87.1	43.2	58.3	48.7	42.3	53.1	42.3	43.0	25.5	48.0	40.1	40.2	19.3	27.2	59.8	35.0	57.8	43.9	57.8	47.
[Team 5]()	84.5	73	37.2	91.2	54	95.7/	92	63.4	49.2	44	86.4	59	54	71.1	73.9	54	71.5	60.7	51.1	67.4	45.4	65.8	64.3
[Team 6]()	55.23	21.304	18.227	46.528	11.86	34.749	36.07	7.5805	10.714	9.6084	34.068	10.998	17.169	22.1	8.3726	0.29377	6.8783	51.951	24.216	44.246	33.639	53.136	30.901

Use implementation of IDF1/HOTA provided in [TrackEval]().

(optional) Task 3.2: MTSC tracking - SEQ4 (move values)

Provide the results for your best technique

	HOTA (SEQ 4)																						
Camera	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	c26	c27	c28	c29	c30	c31	c32	c33	c34	c35	c36	c37	c38
Team 1																							
Team 2	0.31	0.24	0.23	0.46	0.09	0.24	0.18	0.26	0.11	0.11	0.18	0.19	0.20	0.27	0.05	0.28	0.05	0.33	0.23	0.15	0.29	0.31	0.24
Team 3																							
Team 4	31.5	40.2	33.5	61.8	22.4	37.9	36.1	23.6	35.8	26.4	30.1	21.7	33.4	32.0	29.7	19.2	18.3	37.4	26.6	36.8	31.8	42.2	28.4
Team 5	70.3	58.7	21.2	77.1	37	81.4	77.9	46.3	31.6	27.7	67.6	46.6	48.1	62.6	51.4	38.7	47.5	55.6	49	62.6	40	56.8	48.9
Team 6	64.7 48	27.4 81	26.7 7	45.2 47	23.1 47	44.6 95	59.5 51	13.7 85	10.6 51	16.9 28	55.2 32	29.2 2	22.3 16	32.3 75	29.2 57	2.36 54	29.2 93	54.4 19	33.5 26	50.5 97	47.5 41	53.4 32	34.8 32

Use implementation of IDF1/HOTA provided in TrackEval.

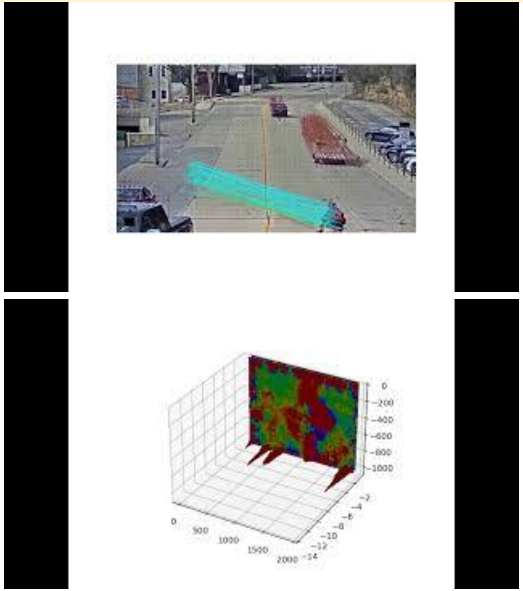
(optional) Task 3.3: MTSC tracking - Discussions (Team X) [m
2 pages]

(optional) Task 3.3: MTSC tracking - Discussions (Team 2) [1/1]

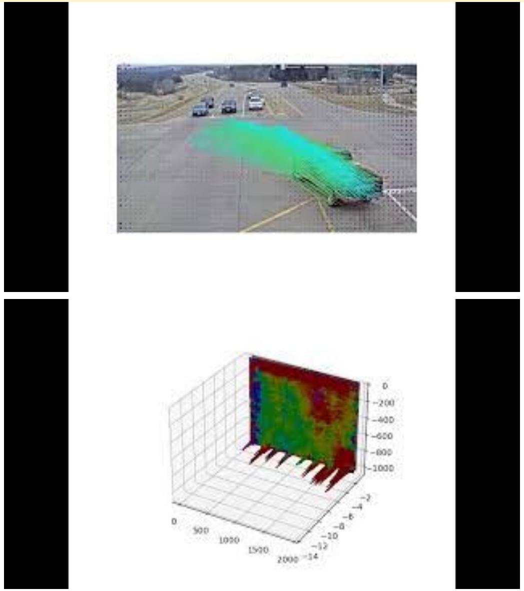
We utilize the tracking implementation with optical flow computed with Unimatch.
As detections, we use the ones provided alongside the dataset, extracted with a Faster R-CNN.

Similar to what was observed in the case of Seq. 3 in section 2, we can see that the outliers coming from the OF make it harder for the Kalman tracking, by introducing more complexity into the signal. In general, we observe the drawbacks that we found in that case, especially, the inaccurate OF due to occlusions and illumination changes, and the lack of robustness to noise.

For instance, in Seq03, camera c016, in the videos displaying the OF (top row), we observe in the initial frames how a car hides right behind another one, making the OF computation totally misleading.
In addition, the OF presents a constant, background noise (bottom video)s that introduces more randomness in the tracking process.



Unimatch Seq04 c016



Unimatch Seq01 c003

We observed that ground truth data for these additional sequences is quite different from the data utilized in previous weeks.

First, we observed that GT consider much **fewer** objects than are detected by regular object detectors such as a pretrained Faster R-CNN. This is because the GT **does not consider** small-sized objects, nor parked/static cars.

Additionally, we noticed that the **quality** of the provided **annotations** is lower than that of our own annotations (which we do not utilize here, due to restrictions in time). This has been already observed in Tracking, section 1.3.

With further work, we could expect results to improve: by removing small detected objects and parked cars, and by considering the interaction between multiple cameras filming the same scene.

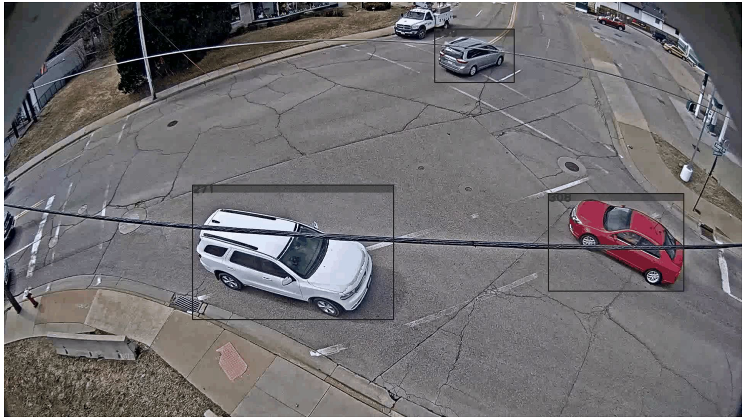
(optional) Task 3.3: MTSC tracking - Discussions (Team 4) [1/1]

We proceed to check one of the best and the worst results on different cameras of sequences 04 and 01

From now on

- Groundtruth
- Predictions

SEQ04 - Camera #c035



We can observe that this camera is really well positioned (objects are close to its view), which involves a good recognition of the objects and also, that all the groundtruth elements are easily recognizable

SEQ01 - Camera #c005

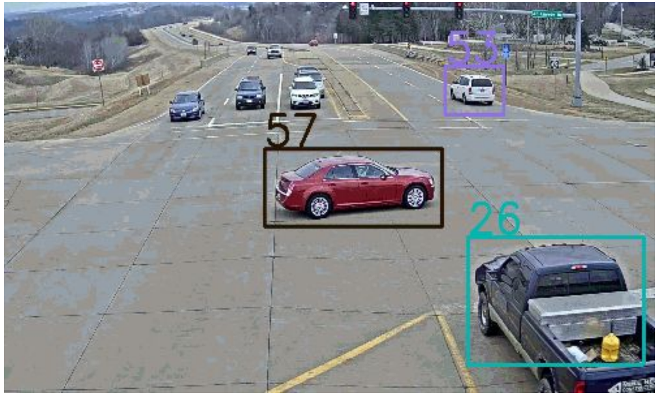
By contrast, in this camera we can observe that even if the tracker takes into account a lot of cars, the groundtruth is not correlated with it (since it is a single camera)

(optional) Task 3.3: MTSC tracking - Discussions (Team 5) [1/2]

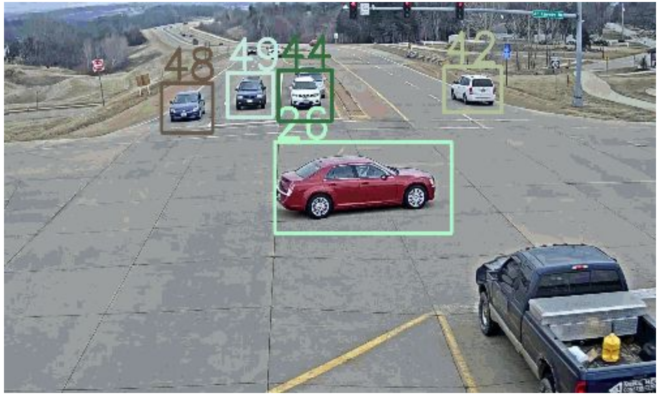
For each of the sequences, different images were used to fine-tune the DETR detector for 5 epochs:
- SEQ1: SEQ3 and SEQ4 were used for fine-tuning.
- SEQ4: SEQ1 and SEQ3 were used for fine-tuning.

- For tracking our algorithm that uses optical flow for tracking (explained in 1.3 slides) was used.
- For the optical flow prediction the GMFlow model was used.
- We post-processed our tracking, by removing the static objects tracked.

SEQ1: c003



Our results



GT

- Our post-processing method removed tracks that were not totally static because they had stopped for a considerable amount of time or the movement during the sequence was not big.
- Track with id 61 was assigned to the wrong car when they overlapped in our method.
- Ground truth (GT) only detects objects that are not occluded (id 26 is not detected).

(optional) Task 3.3: MTSC tracking - Discussions (Team 5) [2/2]

SEQ4: c018, worst results

Our results

GT

SEQ4: c022, best results

Our results

GT

- When there is no overlapping, our method works very well as shown in the c022 example.
- In other cases, it is very dependent on the accuracy of the predicted optical flow. If the optical flow is not accurate, the tracking may switch to the wrong object, as shown in the c018 and c003 examples.
- The post-processing method is also problematic because it is too restrictive and may remove tracks that move slightly, as shown in the c003 example.
- Similar results and issues were observed in all test sequences.
- The fine-tuning of the DETR detector was successful, as it did not detect objects that should not be tracked.

(optional) Task 3.3: MTSC tracking - Discussions (Team 6) [1/3]

	SEQ 01					
Camera	**c001**	**c002**	**c003**	**c004**	**c005**	**Average**
HOTA	42.46	54.642	48.802	50.25	24.868	43.963
IDF1	16.957	27.65	23.167	39.433	3.3253	21.539

(optional) Task 3.3: MTSC tracking - Discussions (Team 6) [2/3]

	SEQ 04							
Camera	**c016**	**c017**	**c018**	**c019**	**c020**	**c021**	**c022**	**c023**
HOTA	64.748	27.481	26.77	45.247	23.147	44.695	59.551	13.785
IDF1	55.23	21.304	18.227	46.528	11.86	34.749	36.07	7.5805

	SEQ 04							
Camera	**c024**	**c025**	**c026**	**c027**	**c028**	**c029**	**c030**	**c031**
HOTA	10.651	16.928	55.232	29.22	22.316	32.375	29.257	2.3654
IDF1	10.714	9.6084	34.068	10.998	17.169	22.1	8.3726	0.29377

(optional) Task 3.3: MTSC tracking - Discussions (Team 6) [2/3]

	SEQ 04									
Camera	**c032**	**c033**	**c034**	**c035**	**c036**	**c037**	**c038**	**c039**	**c040**	**Average**
HOTA	29.293	54.419	33.526	50.597	47.541	53.432	34.832	47.592	42.572	35.359
IDF1	6.8783	51.951	24.216	44.246	33.639	53.136	30.901	44.248	21.474	22.329

Worse results than Task 2.2.
Similar results between SEQ 01 and SEQ 04
Future Work:
Post-processing is necessary, which we have demonstrated before.
Deleting parked cars and small detections should improve the performance of our results.
Cross tracking between cameras with same ID for any object, should increase the metrics.

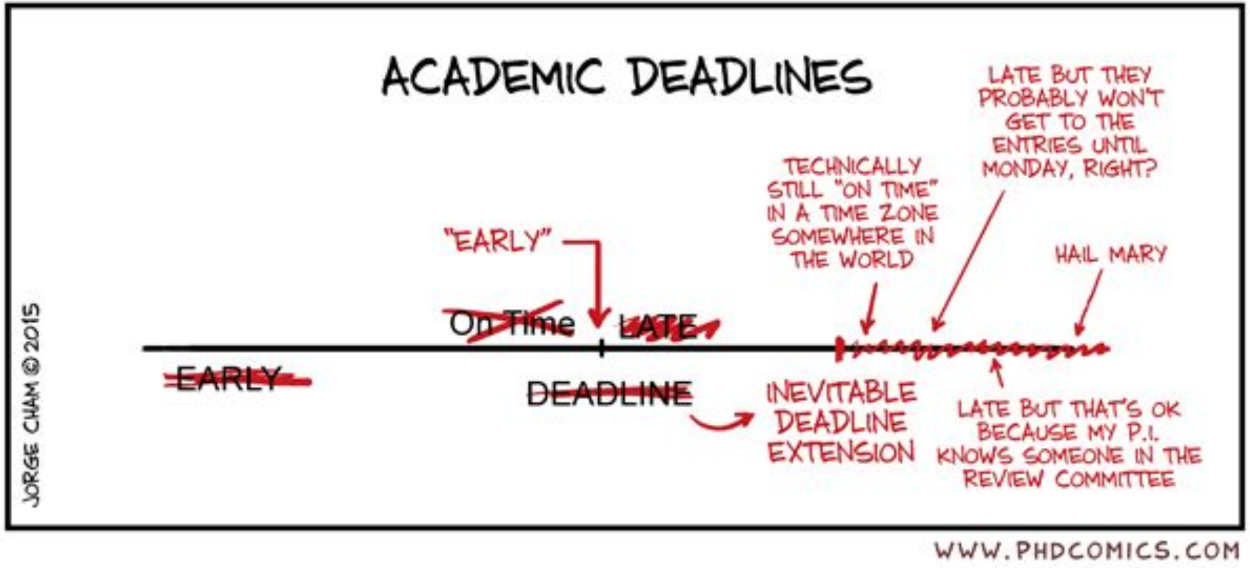
Task 3: MTSC tracking (feedback)

Team ID	Feedback
Team 1	
Team 2	Maybe consider using a tracking algorithm without OF.
Team 3	
Team 4	Like the idea of simplifying by checking best and worst cameras I don’t understand: “the ground truth is not correlated with it (since it is a single camera)”
Team 5	Good analysis and methodology
Team 6	Do not repeat results (already in the table).

Scoring rubric

Task		Weight
T1	Optical Flow	
T1.1	Optical Flow with Block Matching	2
T1.2	Off-the-shelf Optical Flow	2
T1.3	Object Tracking with Optical Flow	3
T2	Multi-target single-camera (MTSC) tracking	
T2.1	SEQ 3 - IDF1 / HOTA	1
T2.2	Discussions	2
T3	SEQ1 + SEQ4 + Discussions	+1

Deliverables



- Deadline: **Wednesday April 12th at 3pm**
- Deliverables:
  - Submit your report by editing these slides: task1 and task2/3
  - Provide feedback regarding the teamwork (email)