M6 Project Overview

Video Surveillance for Road Traffic Monitoring

Team 5:

Ignacio Galve Ceamanos

iceamanos1998@gmail.com

Brian Guang Jun Du

briandu8@gmail.com

Eric Henriksson Martí

eckehenrikssonmarti@gmail.com

José Manuel López Camuñas

joseplcam@gmail.com

Today’s menu

Reminder of **objectives** and **dataset**

Multi-target single camera tracking (**MTSC**)

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

Takeaways

CVPR 2022 AI City Challenge

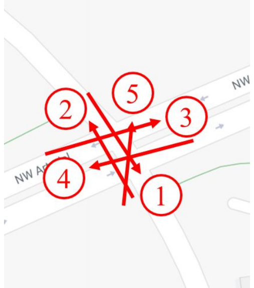
Objective

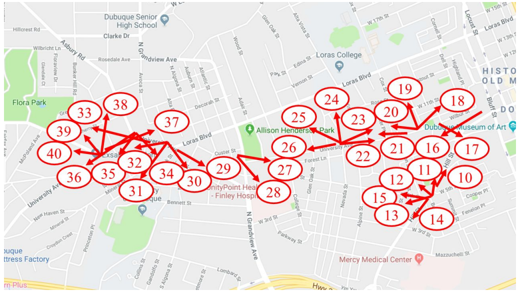
- **Keep track** of and **differentiate** between **moving vehicles** appearing in sequences taken from static cameras
- **Establish correspondences** between identified tracks **across different cameras**

Dataset

Road footage from 3 of the sequences:

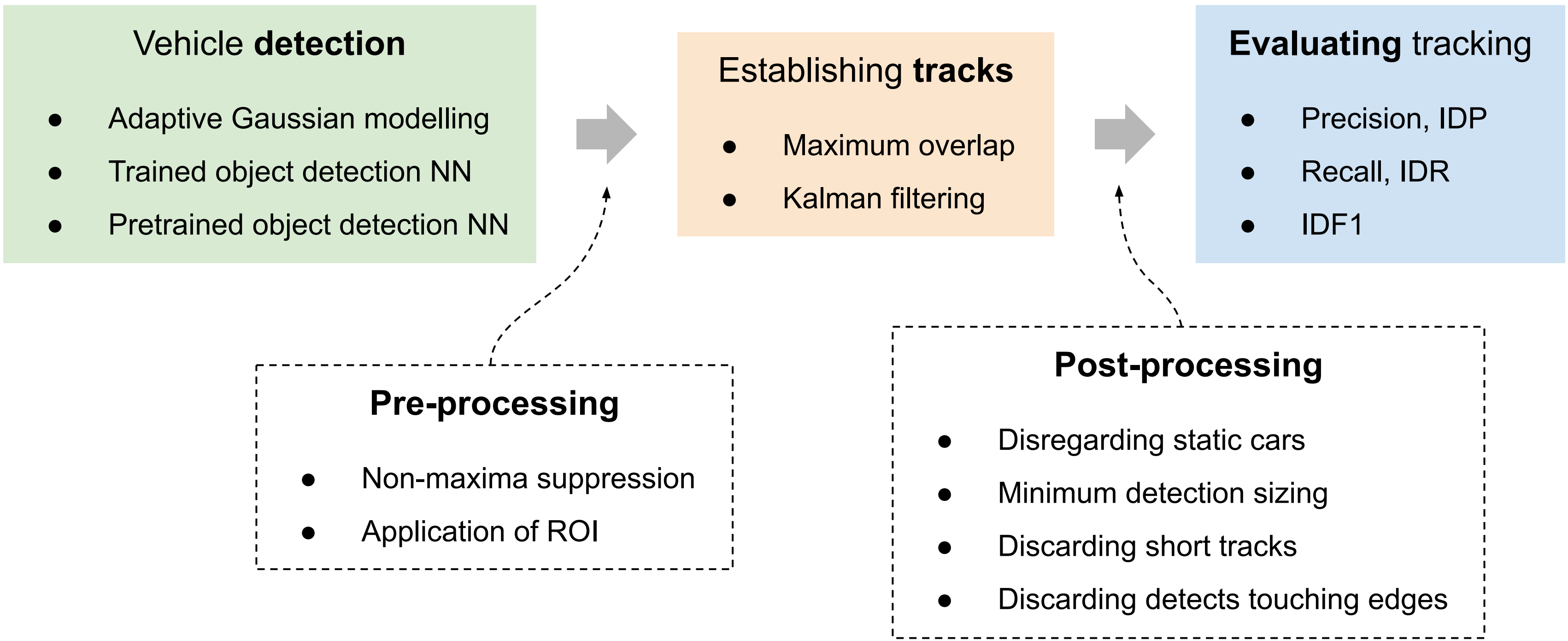
Seq.	Time [min.]	# Cams.	# IDs
1	17.13	5	95
3	23.33	6	18
4	17.97	25	71





T1.1: Multi-target single-camera (MTSC) tracking

Process



Vehicle **detection**

- Adaptive Gaussian modelling
- Trained object detection NN
- Pretrained object detection NN

Establishing **tracks**

- Maximum overlap
- Kalman filtering

Evaluating tracking

- Precision, IDP
- Recall, IDR
- IDF1

Pre-processing

- Non-maxima suppression
- Application of ROI

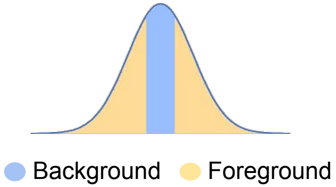
Post-processing

- Disregarding static cars
- Minimum detection sizing
- Discarding short tracks
- Discarding detects touching edges

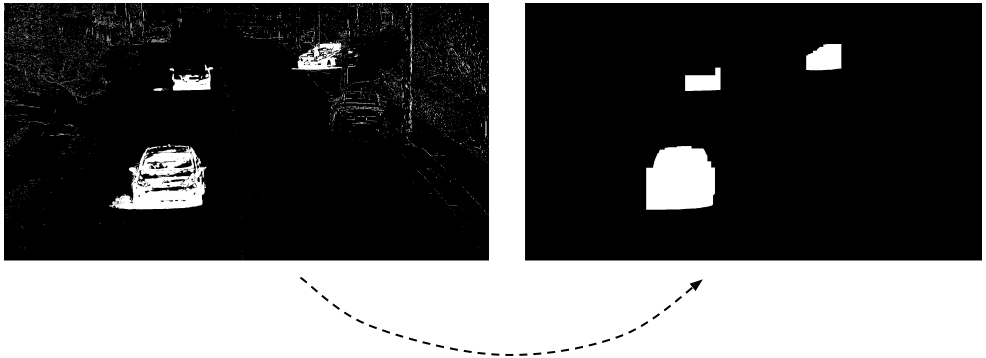
T1.1: Multi-target single-camera (MTSC) tracking

Vehicle detection a) Adaptive Gaussian modelling

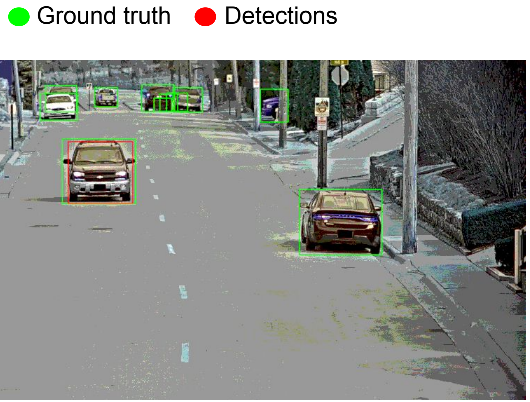
1) “Modelling” background based on first 25% of frames
2) Adapt mean and variance on the go based on background pixels
3) Post-processing and final detections



Background/foreground distinction



Morphological operations



Detections based on foreground

T1.1: Multi-target single-camera (MTSC) tracking

Vehicle detection

b) NN trained on AI City Challenge dataset

Object detection model

RetinaNet

Pre-trained on COCO
(ResNet-101-FPN backbone)

Detectron2

Training

Data

Sequences 1 and 4
(1/30 of frames)

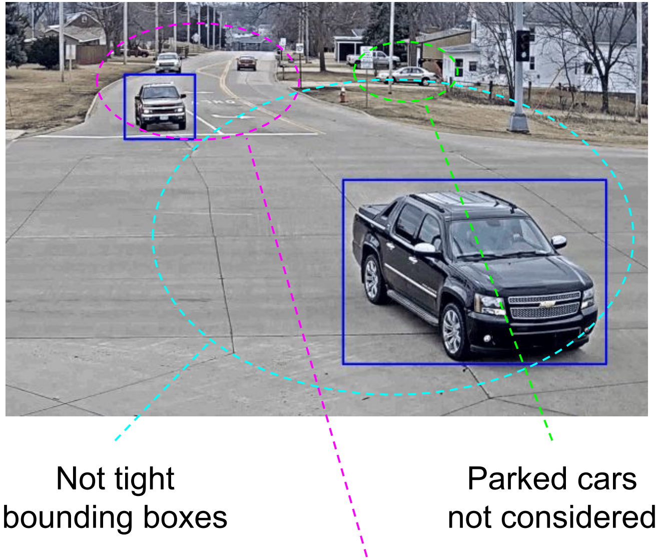
Conditions

Images per batch: 2
Learning rate: 1e-3
Maximum iterations: 3000
Batch size per image: 512

Inference

Sequence 3

“Problematic” ground truth



Not tight bounding boxes

Parked cars not considered

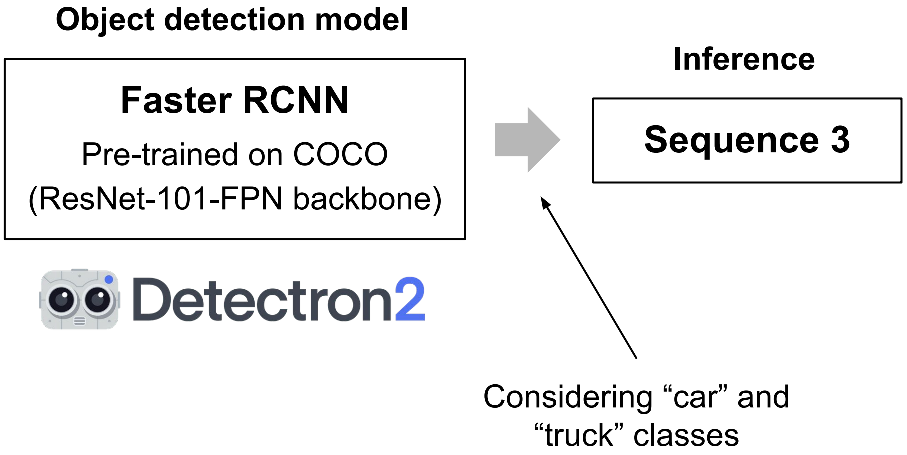
Unclear criteria regarding far away cars

Eric Henriksson - Team 5

T1.1: Multi-target single-camera (MTSC) tracking

Vehicle detection

c) NN pre-trained on COCO



Object detection model

Faster RCNN

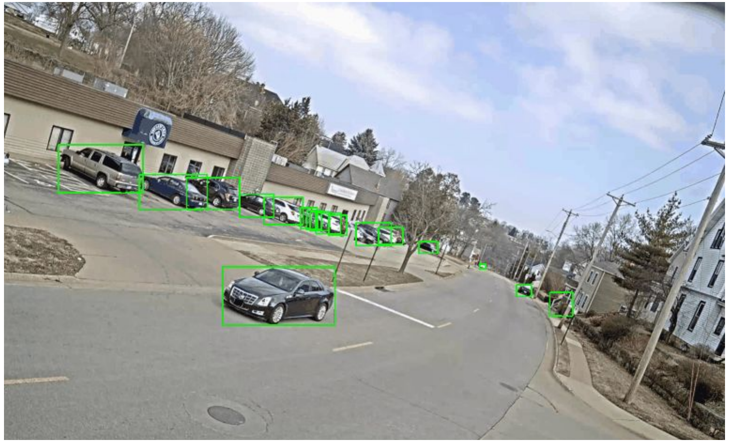
Pre-trained on COCO
(ResNet-101-FPN backbone)

Detectron2

Inference

Sequence 3

Considering “car” and “truck” classes



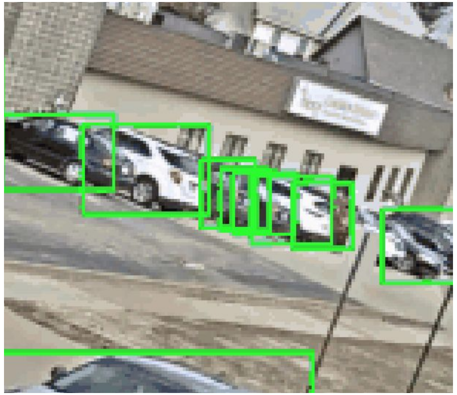
Successful at identifying most vehicles, including parked and far away ones.

T1.1: Multi-target single-camera (MTSC) tracking

Pre-processing

Non-Maxima Suppression

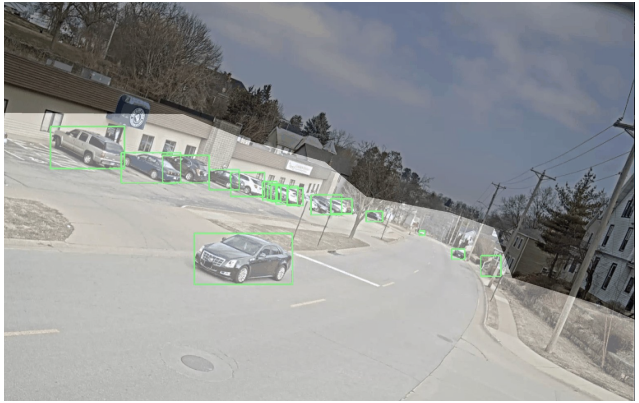
Dealing with cluttered detections



If IoU between detection boxes > 0.8 ➡ Only keep one with highest confidence

Application of ROI

Ignoring detections outside region of interest



Discard detections that have their centre in the zero-value area of the ROI

T1.1: Multi-target single-camera (MTSC) tracking

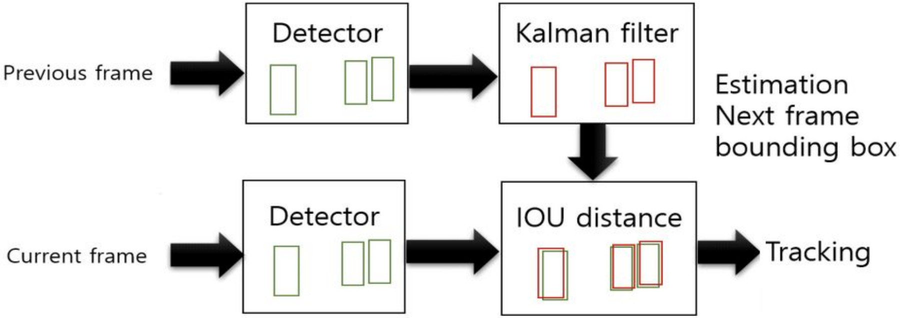
Establishing tracks

a) Maximum overlap

- Determine track IDs by establishing matches between box detections in consecutive frames through the evaluation of **maximum IoU**

b) Kalman filter (SORT)

- Consideration of the **predicted movement** of detections when establishing matches



T1.1: Multi-target single-camera (MTSC) tracking

Post-processing

Disregarding static cars

Checking movement of track centres during seq.

Discarding detects. close to edge

Dealing with cluttered detections

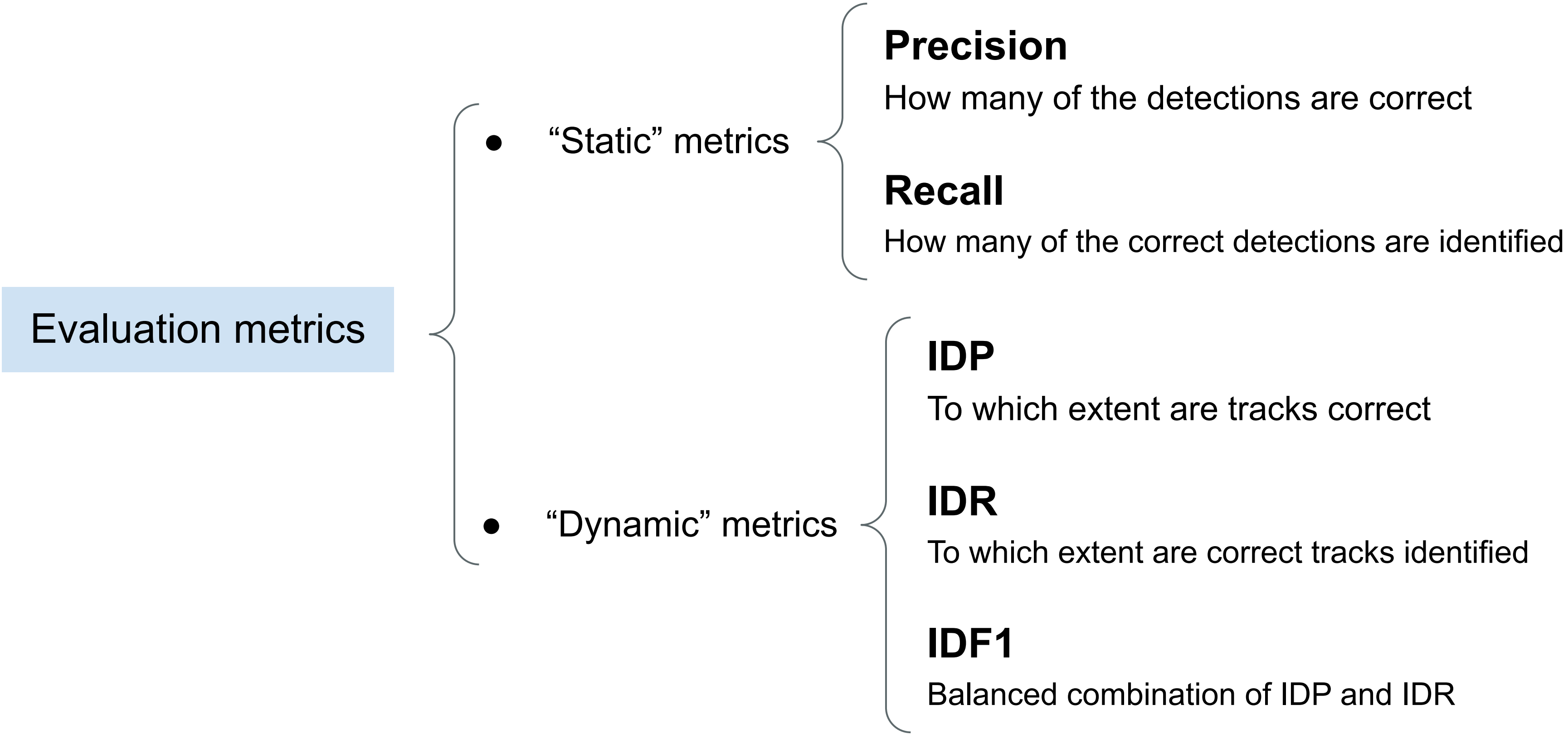
Discarding short tracks

Removing tracks that last for
less than 5 frames

Minimum detection sizing

Ignoring detections smaller than 0.7 of the
minimum detection box in the ground truth

T1.1: Multi-target single-camera (MTSC) tracking



Evaluation metrics

- "Static" metrics
  - **Precision**
    How many of the detections are correct
  - **Recall**
    How many of the correct detections are identified
- "Dynamic" metrics
  - **IDP**
    To which extent are tracks correct
  - **IDR**
    To which extent are correct tracks identified
  - **IDF1**
    Balanced combination of IDP and IDR

T1.1: Multi-target single-camera (MTSC) tracking

Overview of results

			IDF1 (SEQ 3)						
			Camera						Average
Detection method	**Tracking method**	**Post processing**	**c010**	**c011**	**c012**	**c013**	**c014**	**c015**	
Adaptive Gaussian modelling	Max. overlap	No	0.414	0.272	0.073	0.244	0.437	0.018	0.243
		Yes	0.333	0.494	0.095	0.279	0.444	0.444	0.348
	Kalman filter	No	0.402	0.272	0.068	0.222	0.446	0.022	0.239
		Yes	0.348	0.492	0.088	0.282	0.522	0.522	0.376
RetinaNet trained on AI city dataset	Max. overlap	No	0.041	0.011	0.002	0.005	0.030	0.003	0.015
		Yes	0.342	0.232	0.250	0.242	0.302	0.005	0.228
	Kalman filter	No	0.049	0.028	0.002	0.009	0.037	0.003	0.021
		Yes	0.360	0.238	0.255	0.366	0.345	0.006	0.262
Faster RCNN pretrained on COCO	Max. overlap	No	0.199	0.046	0.018	0.140	0.252	0.001	0.109
		Yes	0.754	0.337	0.667	0.869	0.545	0.127	0.550
	Kalman filter	No	0.397	0.049	0.018	0.146	0.255	0.001	0.144
		Yes	**0.768**	**0.472**	**0.824**	**0.767**	**0.742**	**0.129**	**0.617**

T1.1: Multi-target single-camera (MTSC) tracking

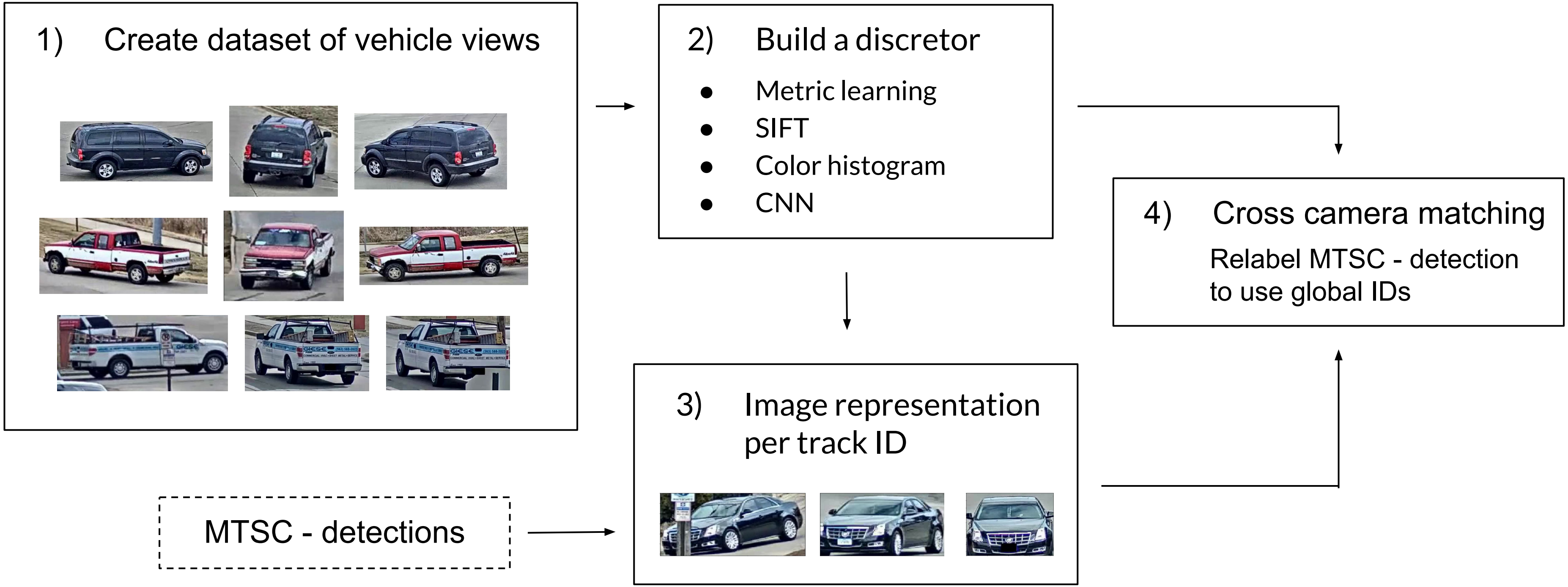
Overview of results - Additional sequences

			IDF1 (SEQ 1)					
			Camera					Average
Detection method	Tracking method	Post processing	c001	c002	c003	c004	c005	
Faster RCNN pretrained on COCO	Kalman filter	Yes	0.727	0.613	0.723	0.675	0.610	0.670

IDF1 (SEQ 4)														
Camera														Average
c016	c017	c018	c019	c020	c021	c022	c023	c024	c025	c026	c027	c028	c029	
0.658	0.566	0.744	0.958	0.694	0.832	0.874	0.730	0.566	0.543	0.635	0.798	0.611	0.684	0.707

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

Process



1) Create dataset of vehicle views

2) Build a discretor

- Metric learning
- SIFT
- Color histogram
- CNN

3) Image representation per track ID

4) Cross camera matching

Relabel MTSC - detection to use global IDs

MTSC - detections

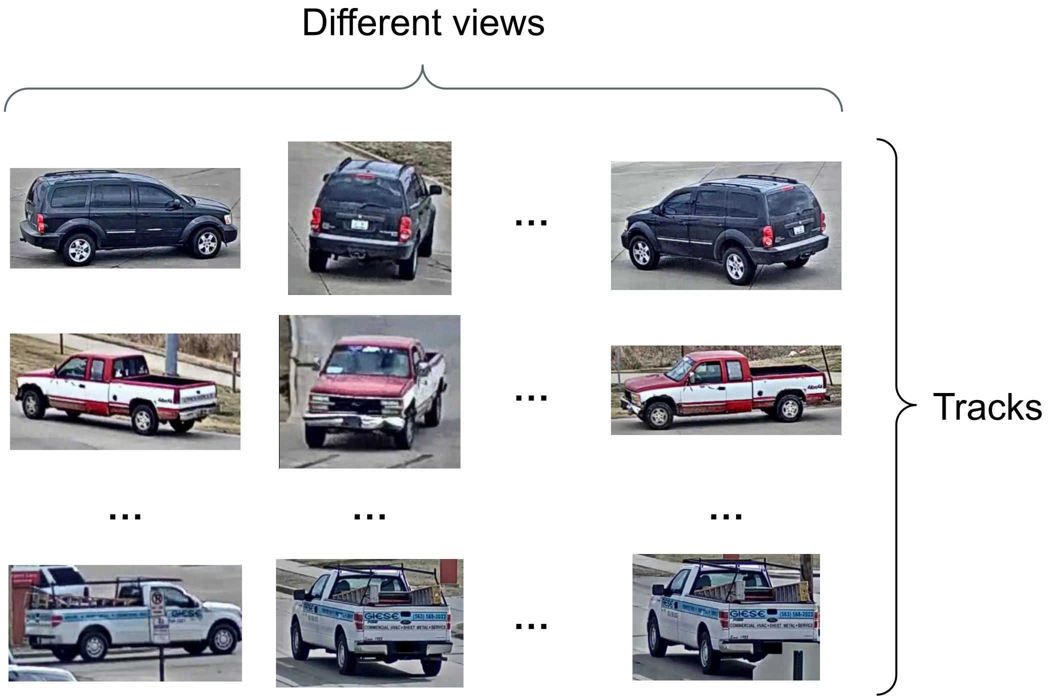
T2: Multi-target multi-camera (MTMC) tracking

Creating custom vehicle dataset

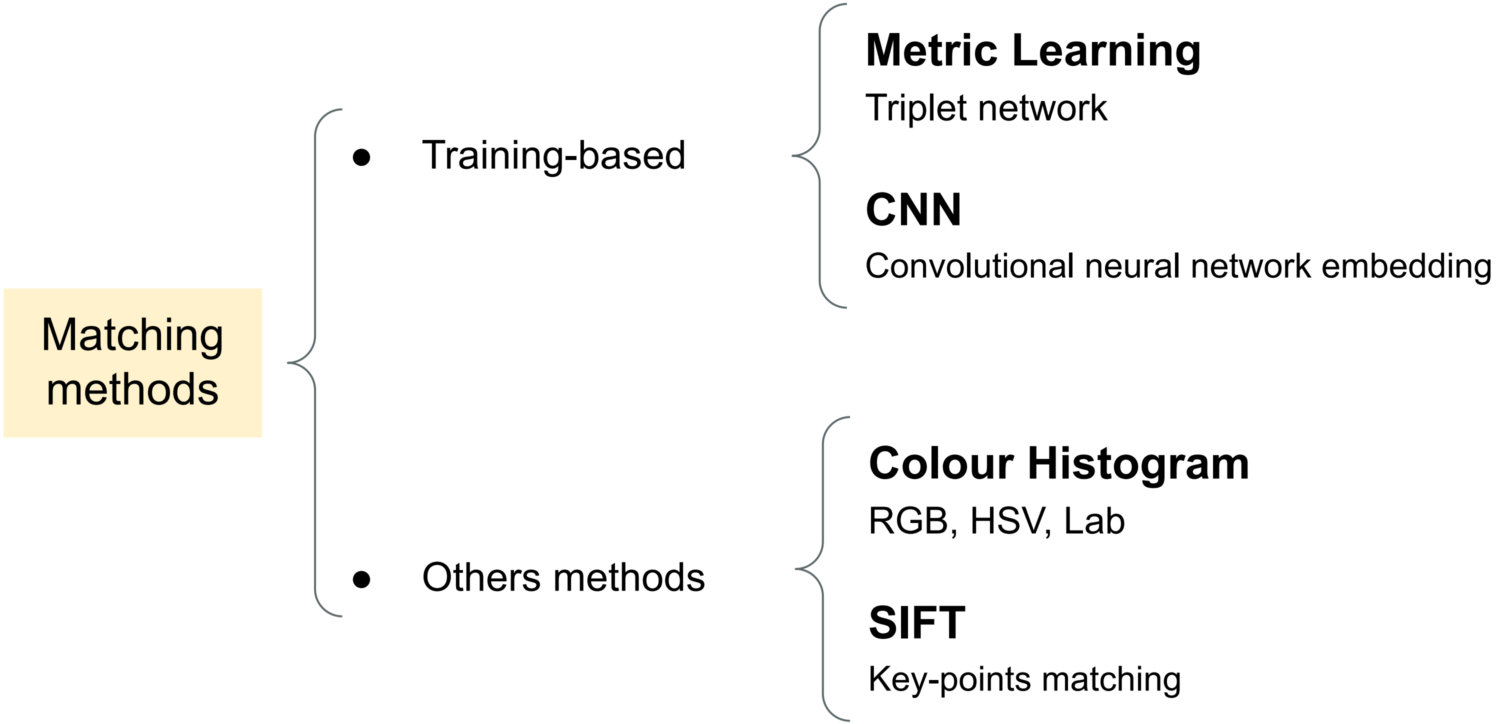
Custom dataset generated from cropping the video frames with **ground truth** detections

Each unique **track ID** is used as a **class**

3 sequences
24 cameras
132 vehicles



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



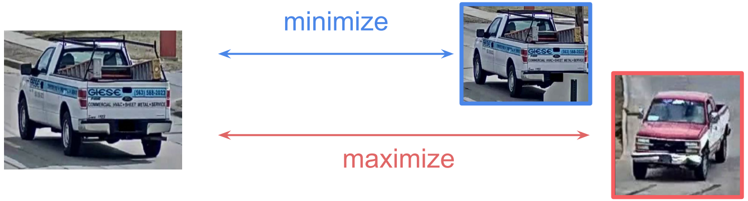
Matching methods

- Training-based
  - **Metric Learning**
    Triplet network
  - **CNN**
    Convolutional neural network embedding
- Others methods
  - **Colour Histogram**
    RGB, HSV, Lab
  - **SIFT**
    Key-points matching

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

Matching methods a) Metric Learning - Triplet network (ResNet18)

Triplet network - **ResNet**18

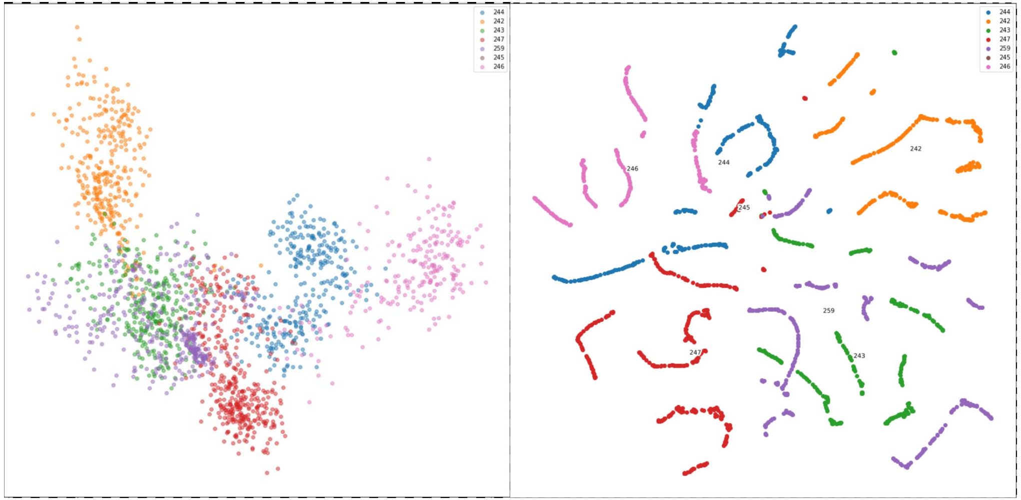


Example of a difficult case

259

243

Visualizing learnt representations on the **test** set



PCA **UMAP**

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

Matching methods a) Metric Learning - Triplet network (ResNet18)

Training config.

Batch size: 32

Learning rate: 1e-3

Epochs: 20

Optimizer: SGD

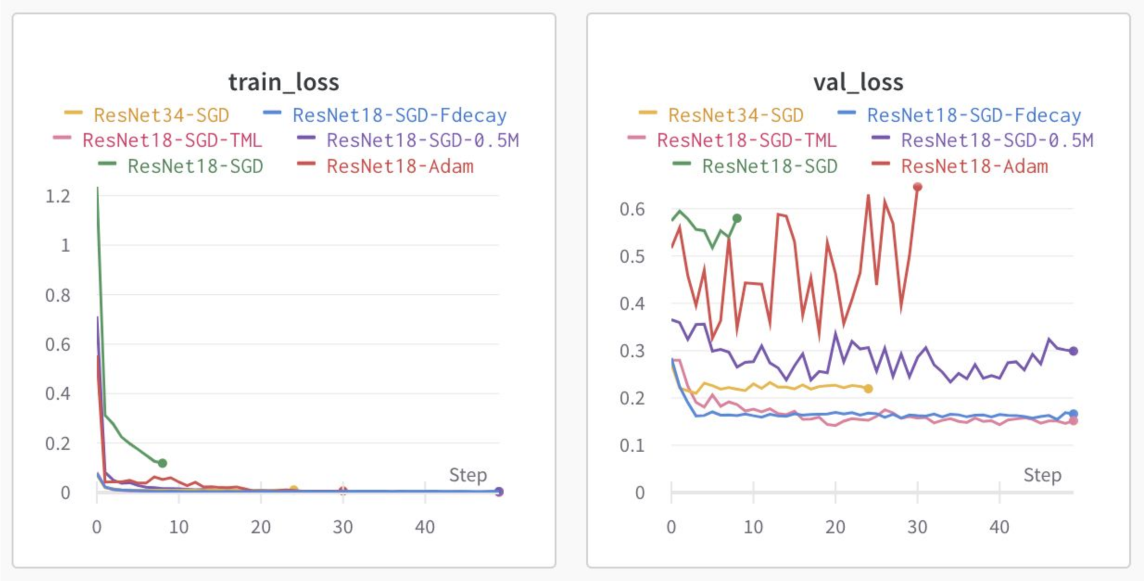
Lr scheduler: gamma-0.1, step 3

Loss margin: 0.5

Loss: Triplet margin lose

Distance Metric

Hist. comparison: Hellinger

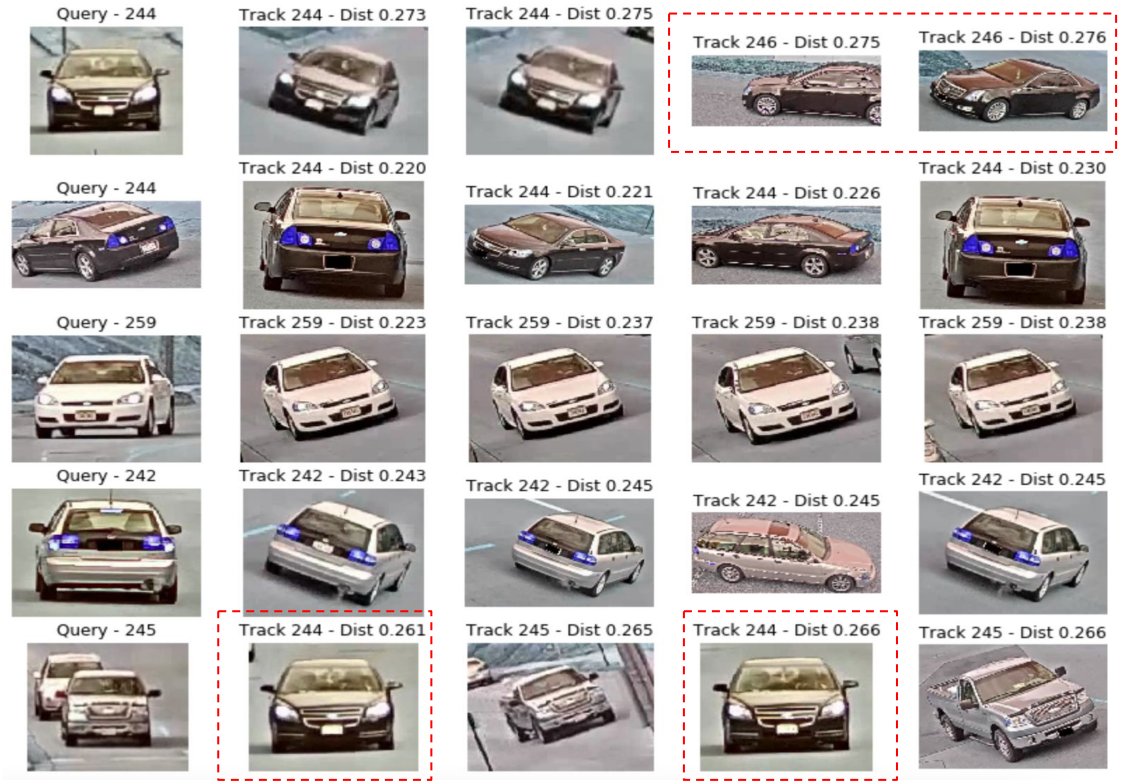


Ignacio Galve - Team 5

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

The hardest problem is check whether a distance between car embeddings is a TP or not

Set a **threshold** computed by **averaging the distances of TP** retrievals at 1



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

Matching methods a) CNN - ResNet18

- Pretrained ResNet18 from torchvision (on COCO dataset) fine-tuned with our train dataset
  - 94% accuracy for train set
- Extracted Features from the last FC layer (512 features)

Training config.

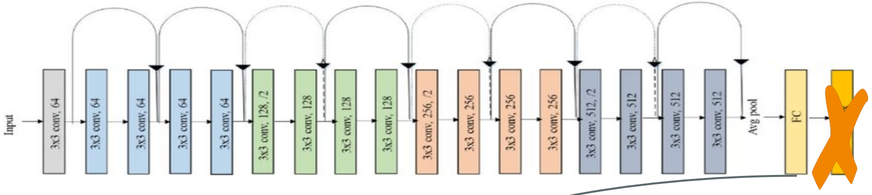
Batch size: 8

Learning rate: 1e-3

Epochs: 5

Optimizer: SGD

Momentum: 0.9



Embedding Model

0.000	0.006	-0.013	...	-0.013

Extracted Embedding/Features (512 features)

Feature comparison: Euclidean

Distance Metric

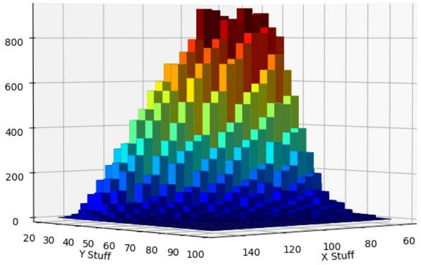
Brian Du - Team 5

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

Matching methods b) 3D Color histogram

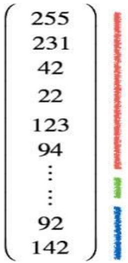
- Extract the 3D (3 channels) colour histogram of each image, normalize it and then flatten into a single vector
- Tested on different colour spaces: RGB, HSV and LAB
- HSV performs slightly better than other colour spaces

3D histogram



Source

Flatten and Normalize



Source

Distance Metric

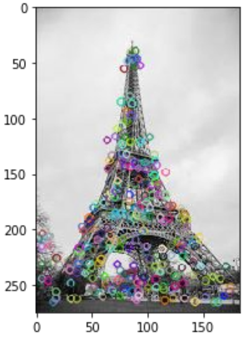
Hist. comparison: Hellinger

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

Matching methods c) SIFT features

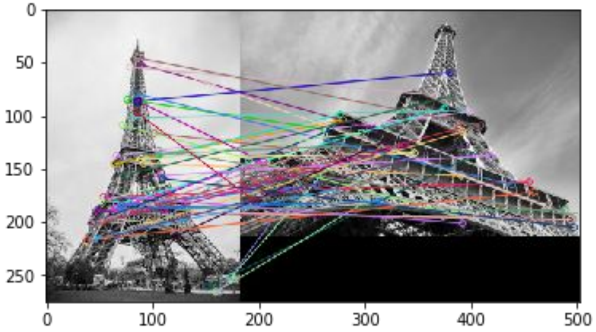
- Extract keypoints and descriptors per image with SIFT
- Apply ratio test to discard missmatched points
- Use number of matching points as distance metric (1/num. matches)

Features Extraction



Source

Matching Keypoints



Source

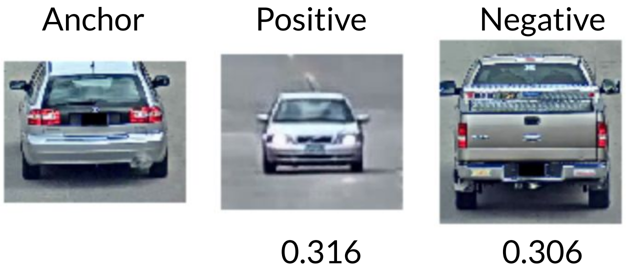
Distance Metric

1 / Number of matches

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

Matching pre-processing

1. It is important to select a **"good" frame** for matching, therefore, we create a selection of **image representation** for each track.
2. ReId tracks with closest anchor track if better than threshold

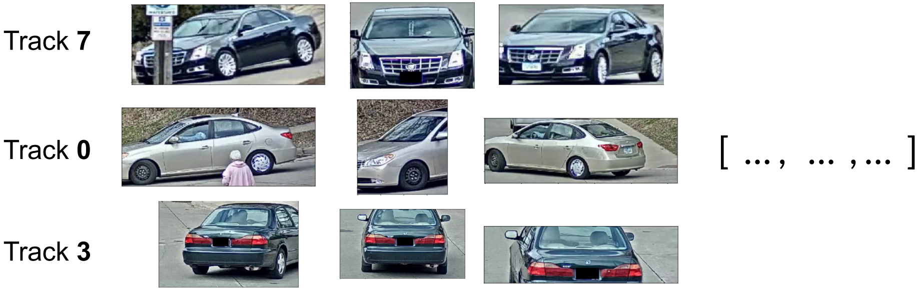


Metric learning inference with **Hellinger** distance

MTSC - detection

Metric Learning

A selection of **image representation**



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

Cross camera matching / Relabeling

1. Compare image representations between the first two cameras.
2. Get track with minimum distance.
3. Check if it's smaller than the threshold of new tracks.
    - a. **Smaller**: assign anchor track to current detection.
    - b. **Bigger**: new global track, add to anchor list.
4. Repeat until all cameras in sequential order have been checked.
5. Remove tracks that only appears in one camera

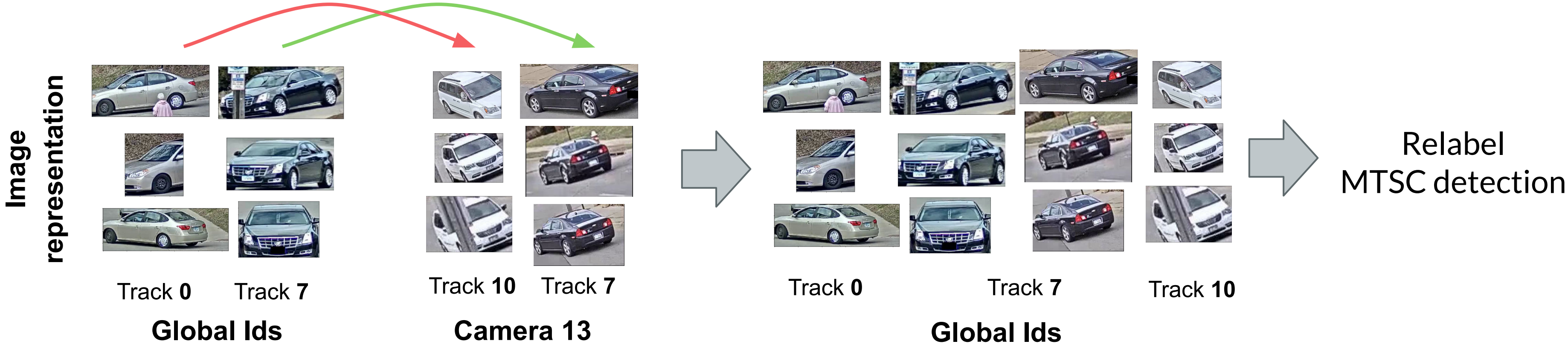


Image representation

Track **0** Track **7**

Global Ids

Track **10** Track **7**

Camera 13

Track **0** Track **7** Track **10**

Global Ids

Relabel MTSC detection

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

Overview of results

		SEQ 3				
Method		**IDF1**	**IDP**	**IDR**	**Precision (detection)**	**Recall (detection)**
Baseline		0.271	0.217	0.357	0.356	0.584
Metric Learning		**0.385**	**0.392**	**0.385**	**0.770**	**0.767**
Colour Histogram	RGB	0.365	0.370	0.345	0.770	0.767
	HSV	0.371	0.372	0.371	0.770	0.767
	Lab	0.370	0.372	0.368	0.770	0.767
SIFT		0.364	0.375	0.363	0.770	0.767
CNN		0.325	0.444	0.263	0.778	0.483

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

Overview of results

		SEQ 1				
Method		**IDF1**	**IDP**	**IDR**	**Precision (detection)**	**Recall (detection)**
Baseline		0.382	0.401	0.461	0.712	0.887
Metric Learning		**0.470**	**0.462**	**0.523**	**0.725**	**0.887**
Colour Histogram	HSV	0.456	0.414	0.506	0.725	0.887
SIFT						
CNN		0.452	0.524	0.502	0.726	0.888

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

Overview of results

		SEQ 4				
Method		**IDF1**	**IDP**	**IDR**	**Precision (detection)**	**Recall (detection)**
Baseline		0.404	0.408	0.399	0.791	0.774
Metric Learning		0.431	0.435	0.426	0.792	0.775
Colour Histogram	HSV	0.423	0.428	0.419	0.792	0.775
SIFT		**0.447**	**0.437**	**0.471**	**0.799**	**0.784**
CNN		0.409	0.414	0.405	0.791	0.774

Key takeaways

MTSC tracking

- Several methods can be used to detect dynamic objects in video footage
- Kalman helps generate more robust tracks than maximum overlap
- Pre and post-processing are key, especially for non learning-based methods
- When using training-based methods, tracking performance greatly depends on the quality of the ground truth

MTMC tracking

- Several learning and non learning-based methods can be used to match detections in tracks
- Set a new track threshold has been the hardest step, preventing us from getting better results
- Timing-related constraints could help improve MTMC tracking

Thank you!