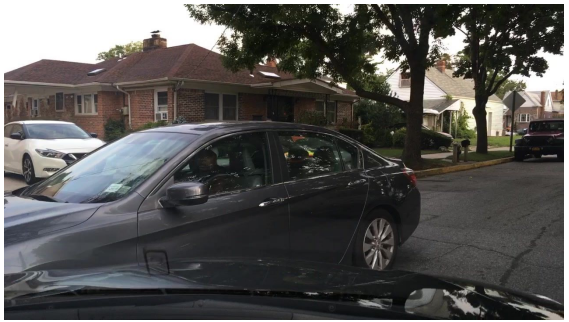
MIRIS: Fast Object Track Queries in Video

Favyen Bastani, Songtao He, Arjun Balasingam, Karthik Gopalakrishnan,
Mohammad Alizadeh, Hari Balakrishnan, Michael Cafarella, Tim Kraska, Sam Madden

MIT CSAIL

Traffic Cameras

Dashcams



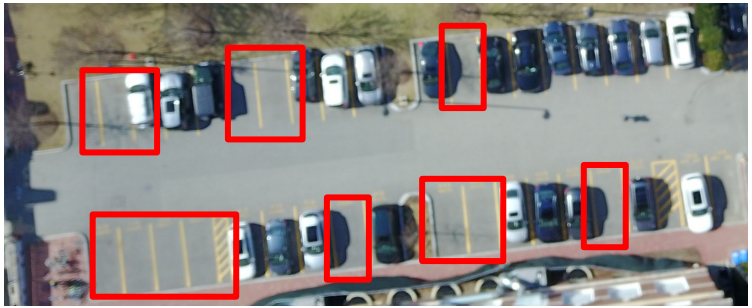
Miscellaneous

Video Analytics

Debugging Autonomous Vehicle Software

Traffic Planning

Finding Interesting Events



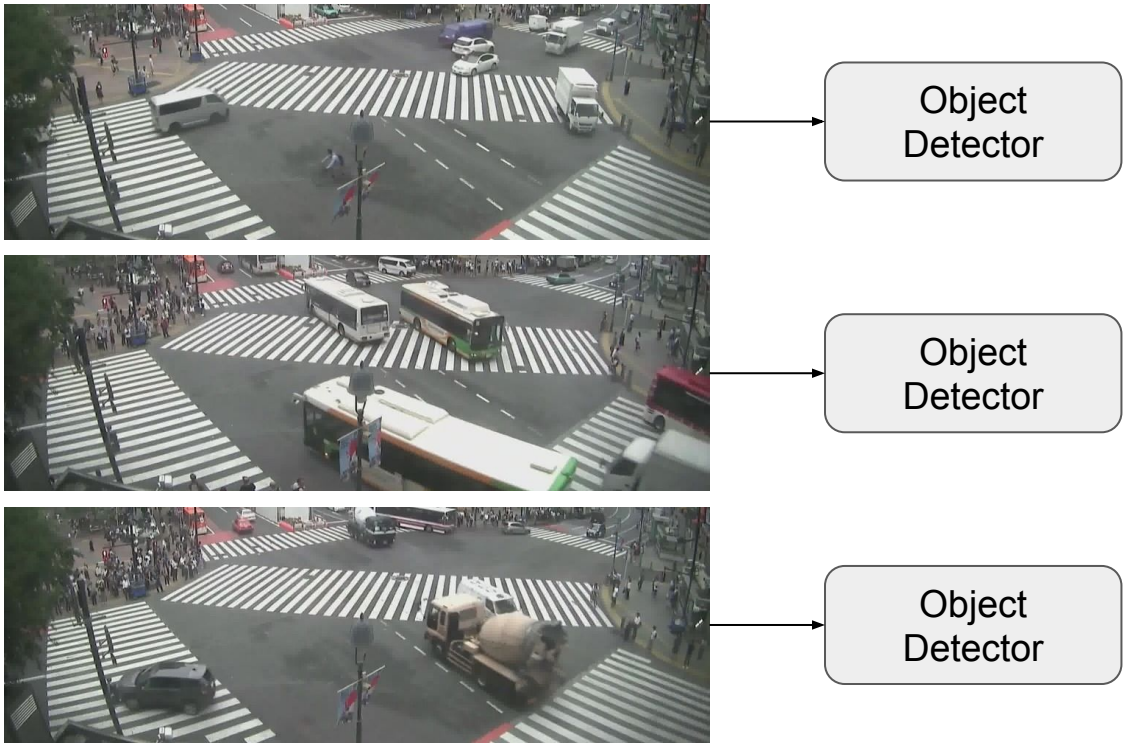
Real-Time Mapping

Prior Work [1, 2, 3]

Select video frames with three buses

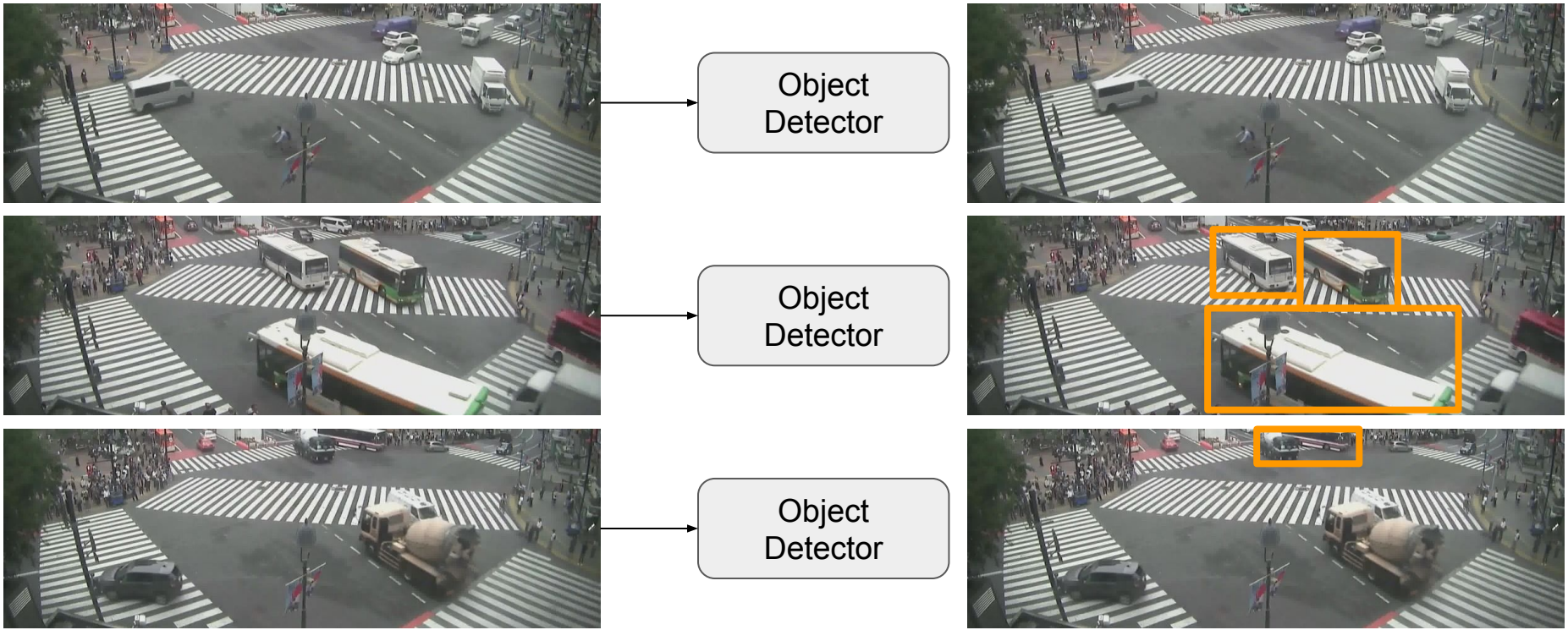
[1] **NoScope**: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[2] Accelerating Machine Learning Inference with **Probabilistic Predicates**. Yao Lu et al. SIGMOD 2018.
[3] **Blazelt**: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.

Prior Work [1, 2, 3]



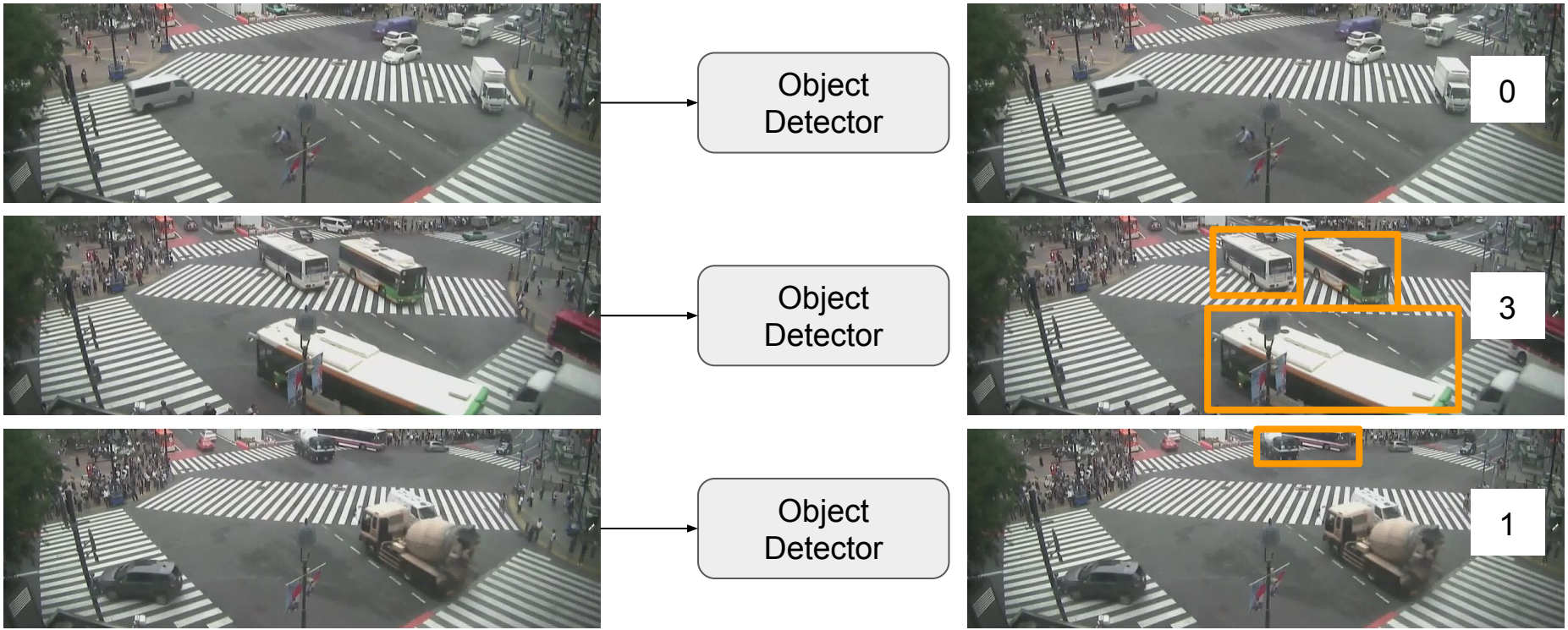
[1] **NoScope**: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[2] Accelerating Machine Learning Inference with **Probabilistic Predicates**. Yao Lu et al. SIGMOD 2018.
[3] **Blazelt**: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.

Prior Work [1, 2, 3]



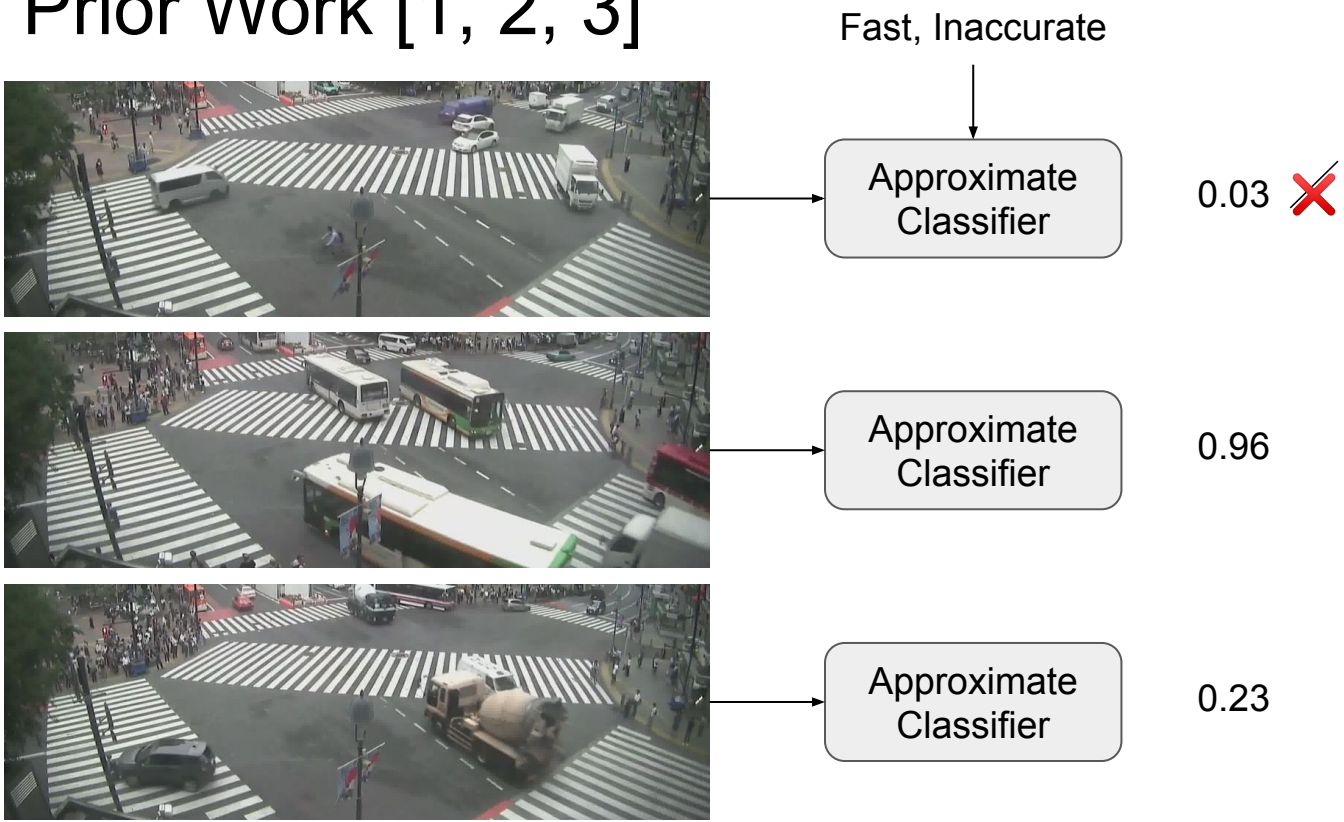
[1] **NoScope**: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[2] Accelerating Machine Learning Inference with **Probabilistic Predicates**. Yao Lu et al. SIGMOD 2018.
[3] **Blazelt**: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.

Prior Work [1, 2, 3]



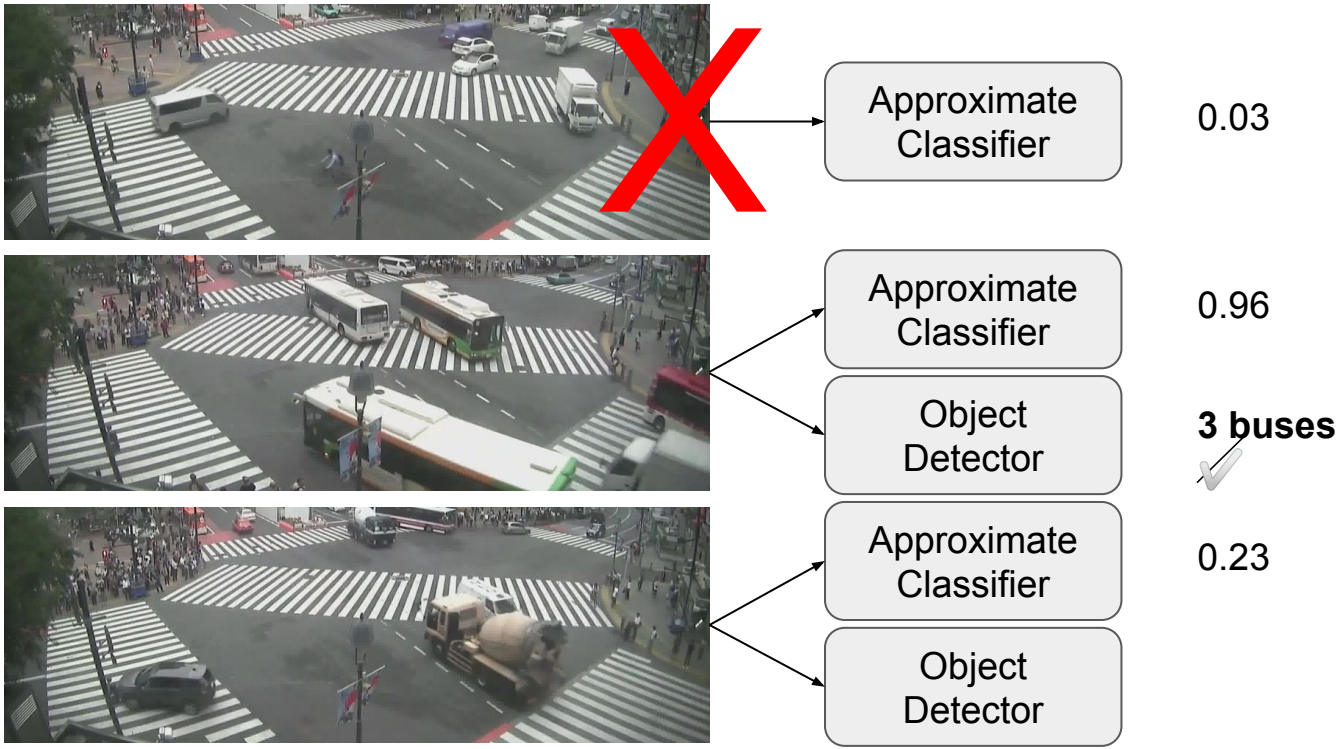
[1] **NoScope**: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[2] Accelerating Machine Learning Inference with **Probabilistic Predicates**. Yao Lu et al. SIGMOD 2018.
[3] **Blazelt**: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.

Prior Work [1, 2, 3]



[1] **NoScope**: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[2] Accelerating Machine Learning Inference with **Probabilistic Predicates**. Yao Lu et al. SIGMOD 2018.
[3] **Blazelt**: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.

Prior Work [1, 2, 3]



[1] **NoScope**: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[2] Accelerating Machine Learning Inference with **Probabilistic Predicates**. Yao Lu et al. SIGMOD 2018.
[3] **Blazelt**: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.

Prior Work [1, 2, 3]

Query: Select video frames with three buses

Approximate Classifier → 0.03

Approximate Classifier → 0.96

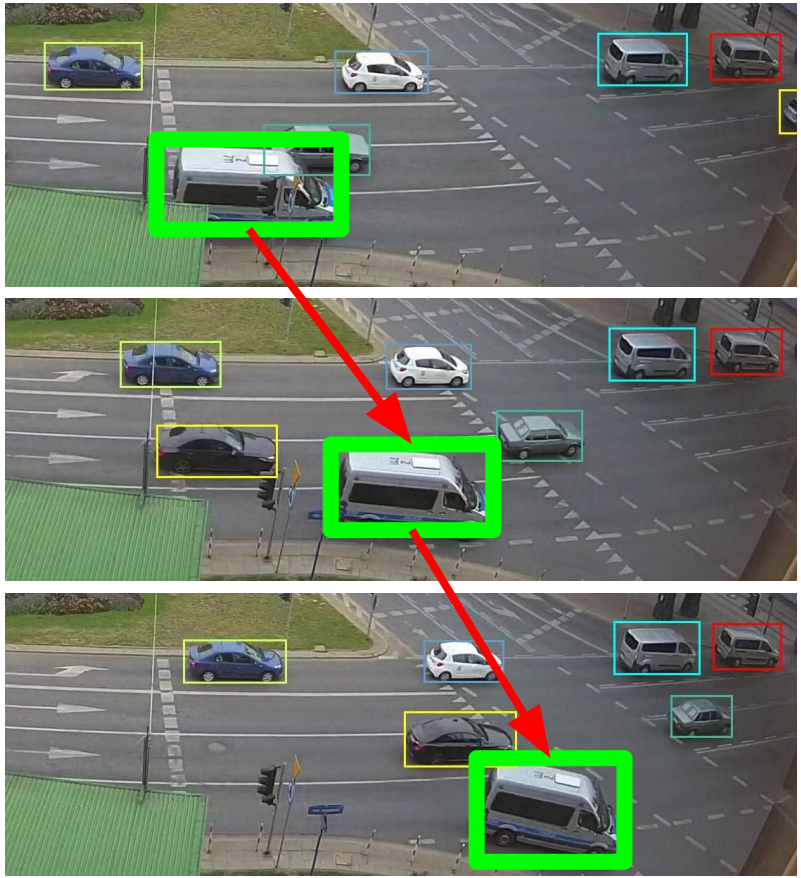
Object Detector → **3 buses** ✓

Approximate Classifier → 0.23

Object Detector → **Only 1 bus** ✗

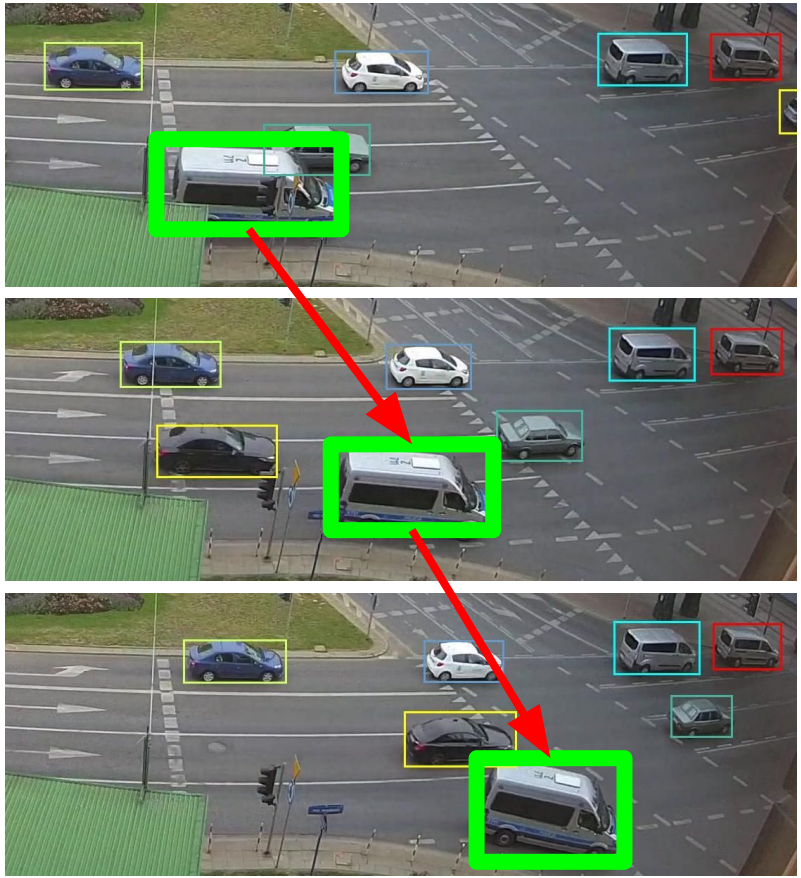
[1] **NoScope**: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[2] Accelerating Machine Learning Inference with **Probabilistic Predicates**. Yao Lu et al. SIGMOD 2018.
[3] **Blazelt**: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.

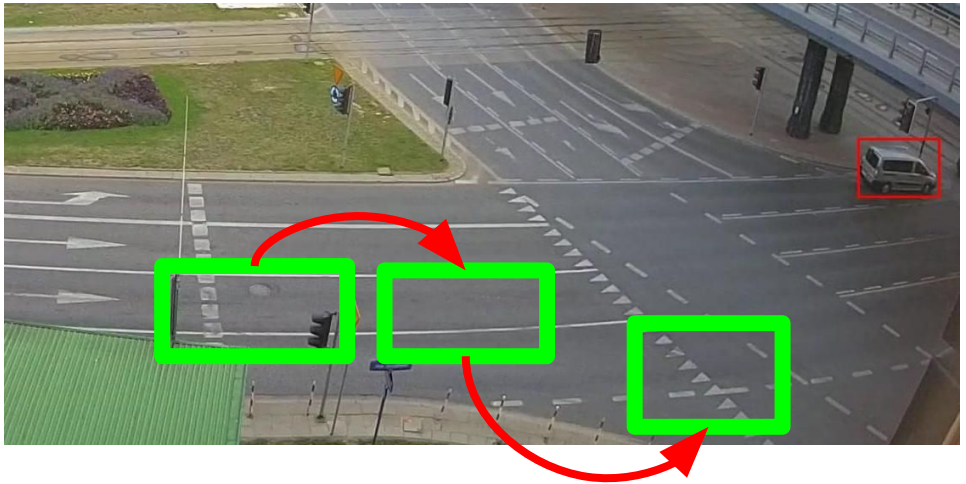
Object Track Queries



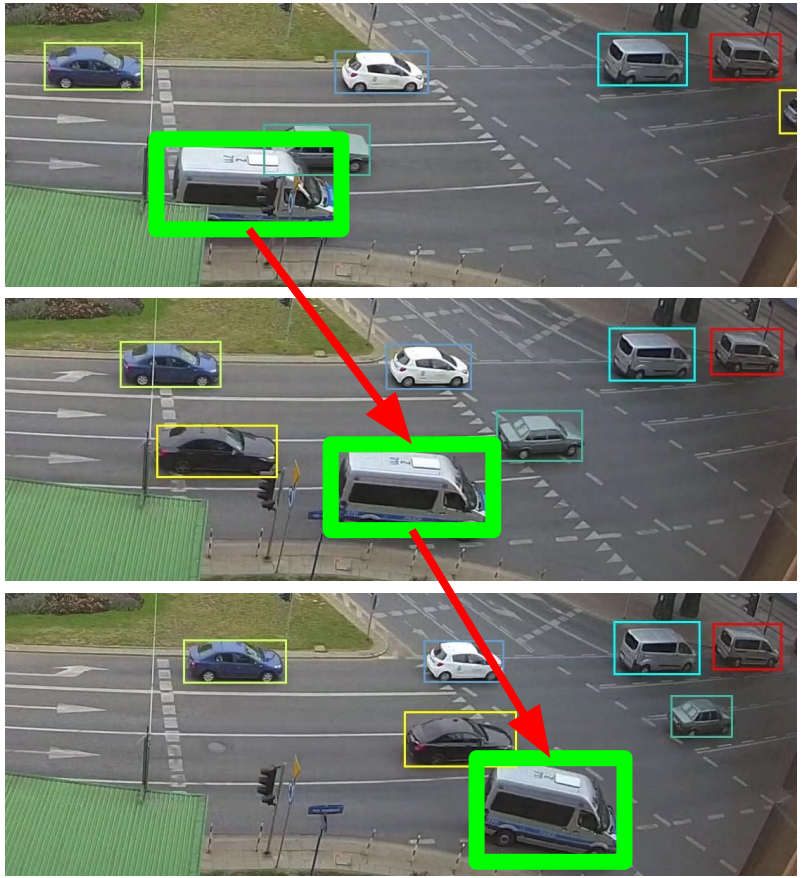
$$<(t_1, x_1, y_1, w_1, h_1), \ldots , (t_n, x_n, y_n, w_n, h_n)>$$

Object Track Queries

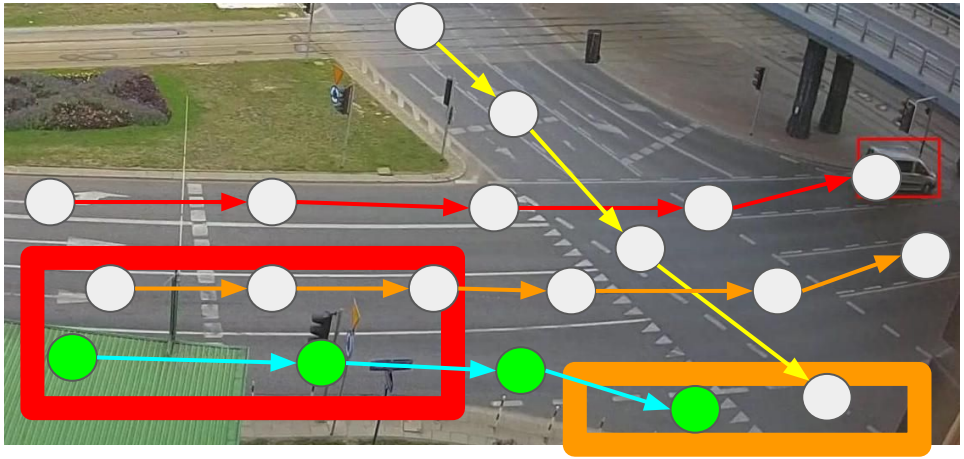




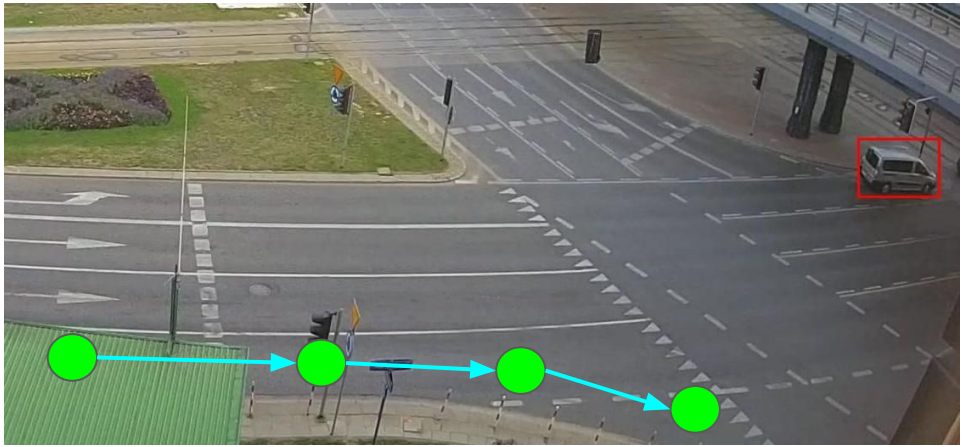
Object Track Queries



Object Track Queries



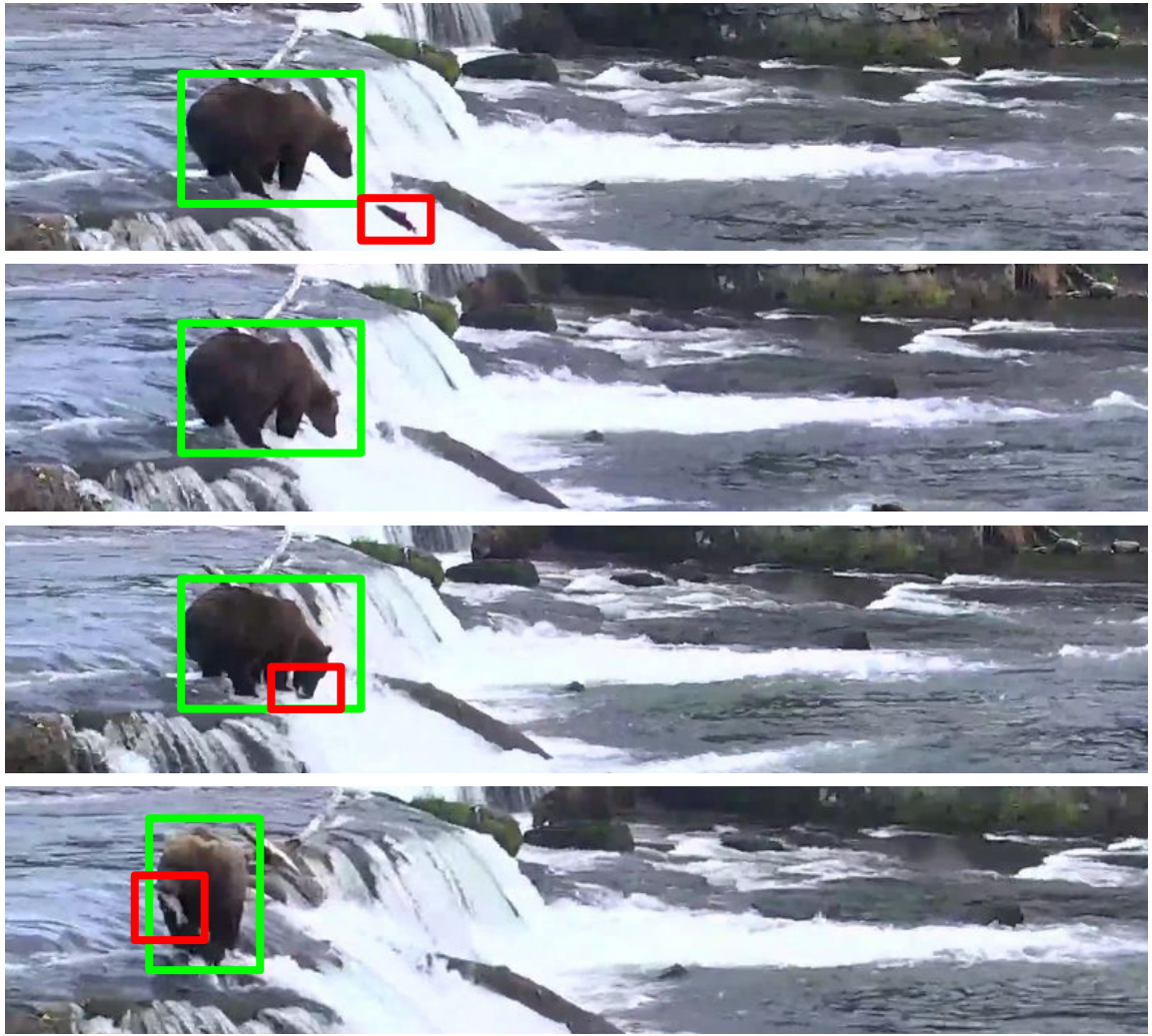
Object Track Queries



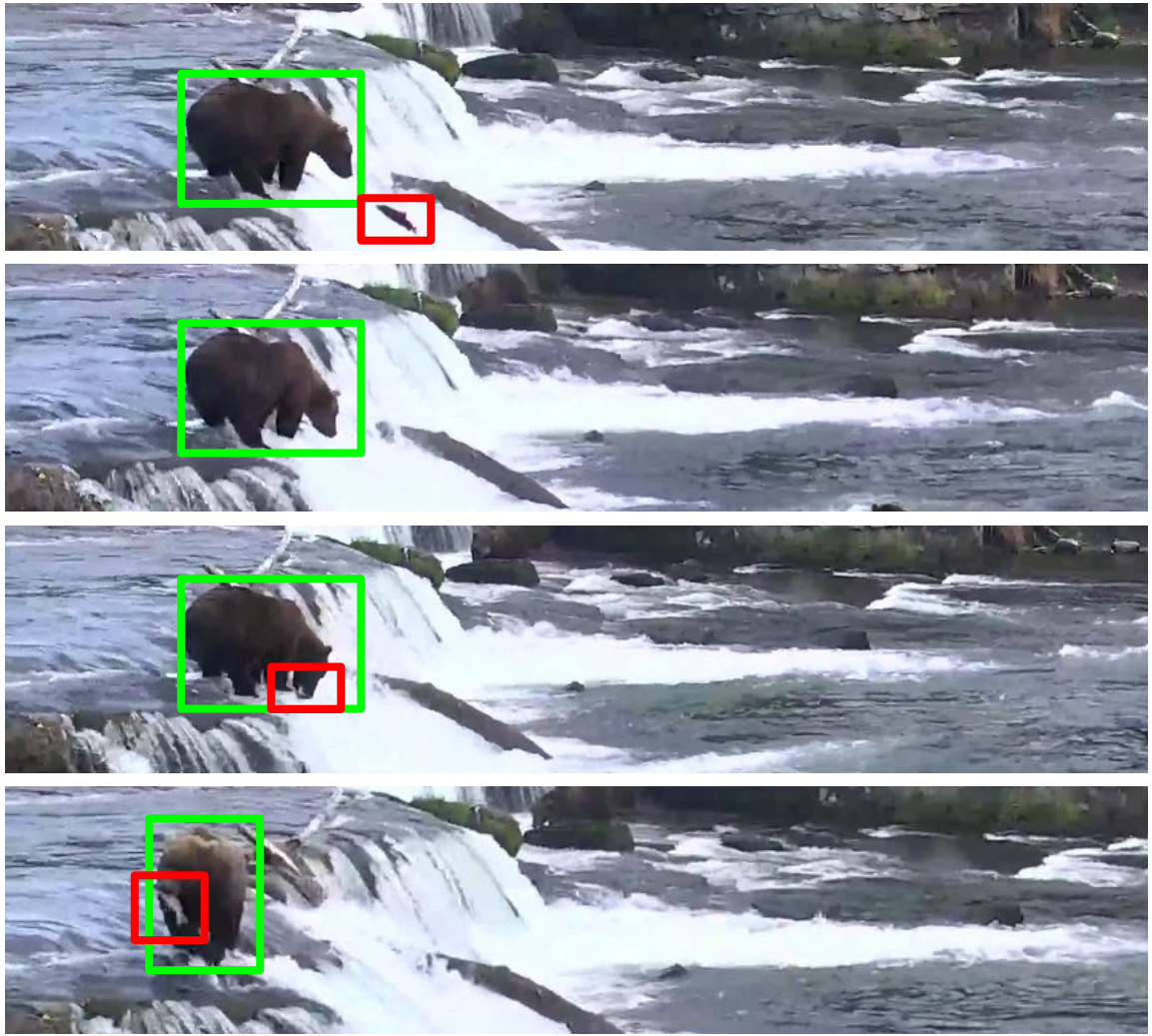
Find cars that rapidly decelerate

Find cars that rapidly decelerate

Given track *A*: select *A* if there is a 1 sec interval *I* such that, if v_1 is *A*'s velocity in first half of *I*, and v_2 is velocity in second half, then v_1 - v_2 exceeds a threshold.

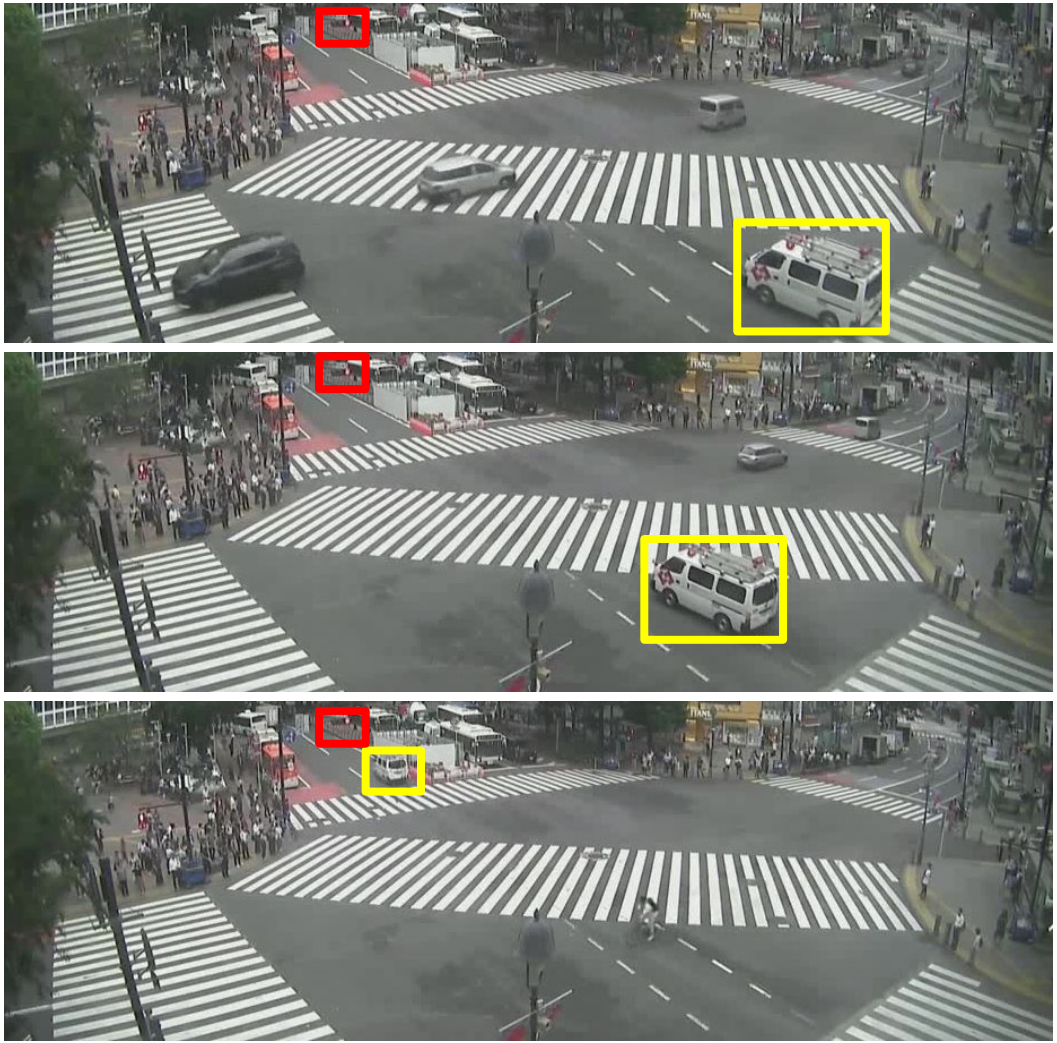


Find bears catching salmon

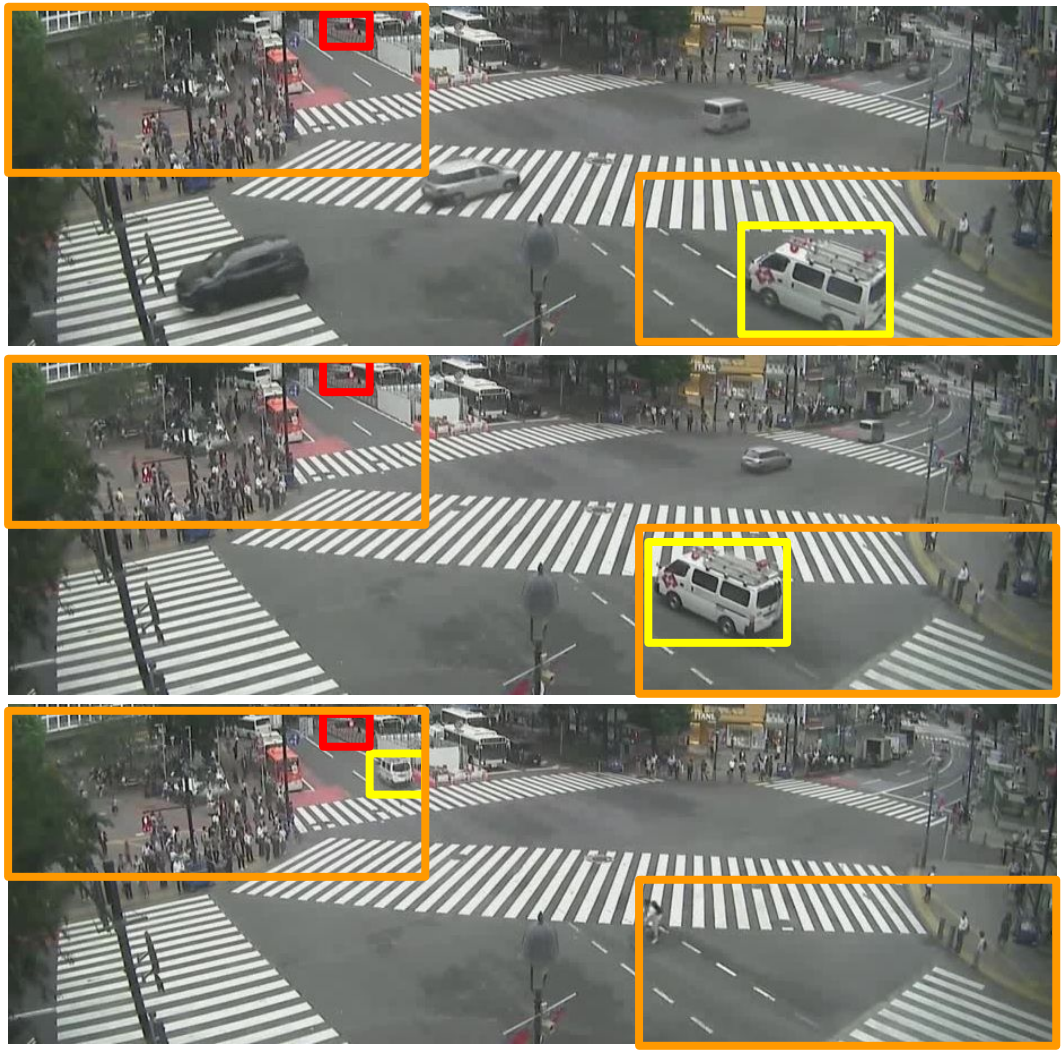


Find bears catching salmon

Given bear *A* and salmon *B*: select (*A*, *B*) if *A* and *B* intersect for at least two seconds.



Find cars that run a red light



Find cars that run a red light

Given car *A* and red light *B*: select (*A*, *B*) if *A* starts in bottom-right and ends in top-left, and the interval of *A* is contained in the interval of *B*.






Object Detector

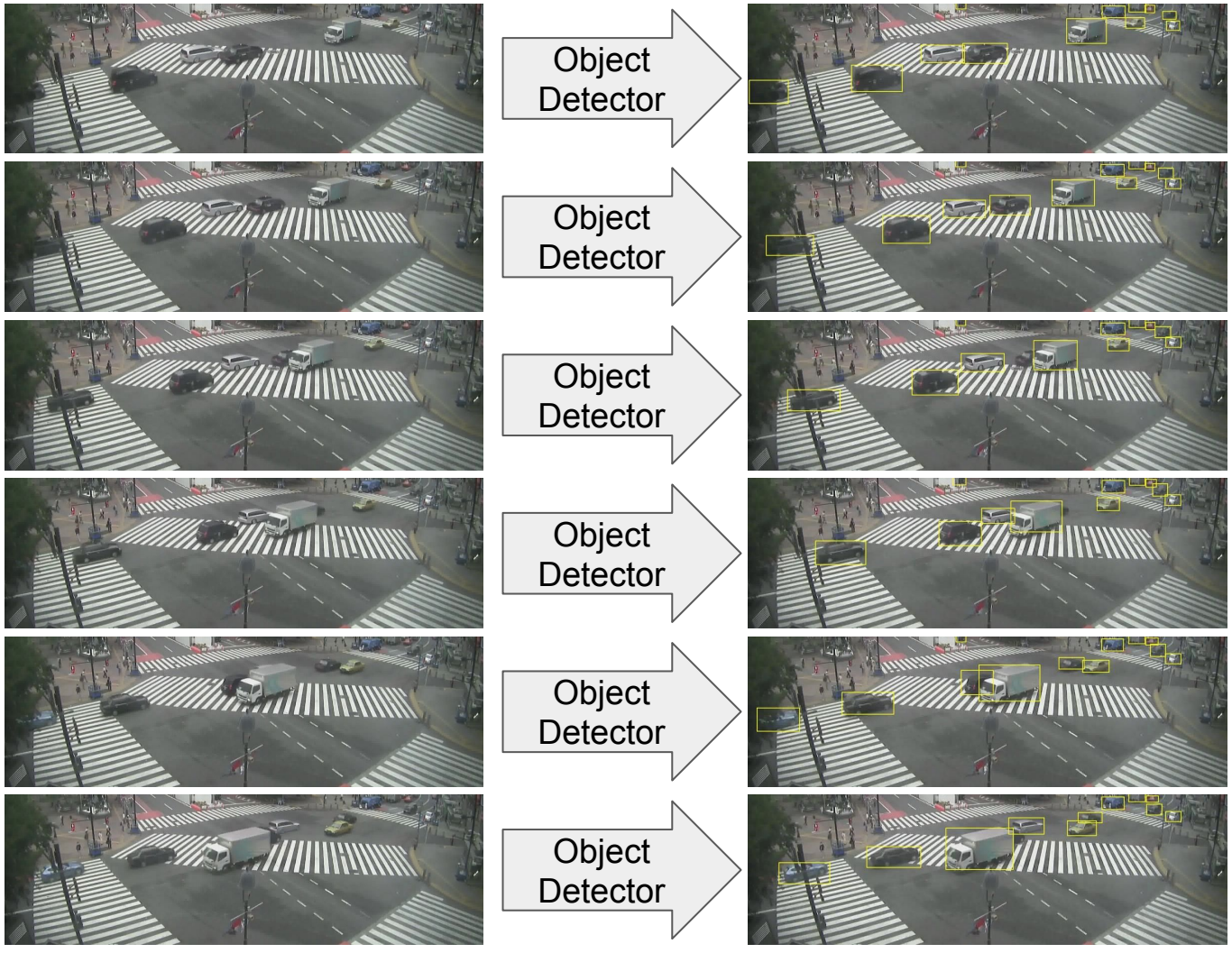
Object Detector

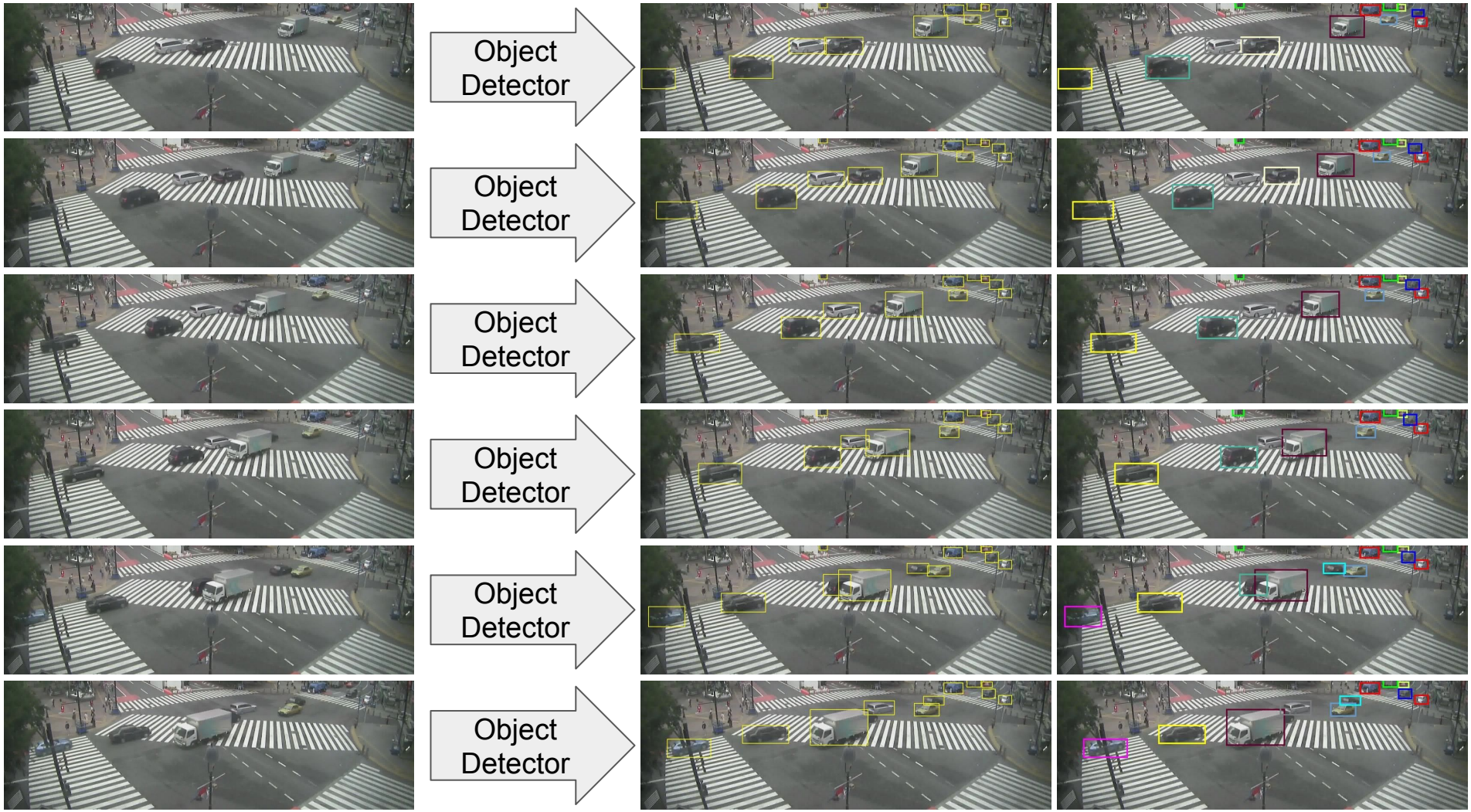
Object Detector

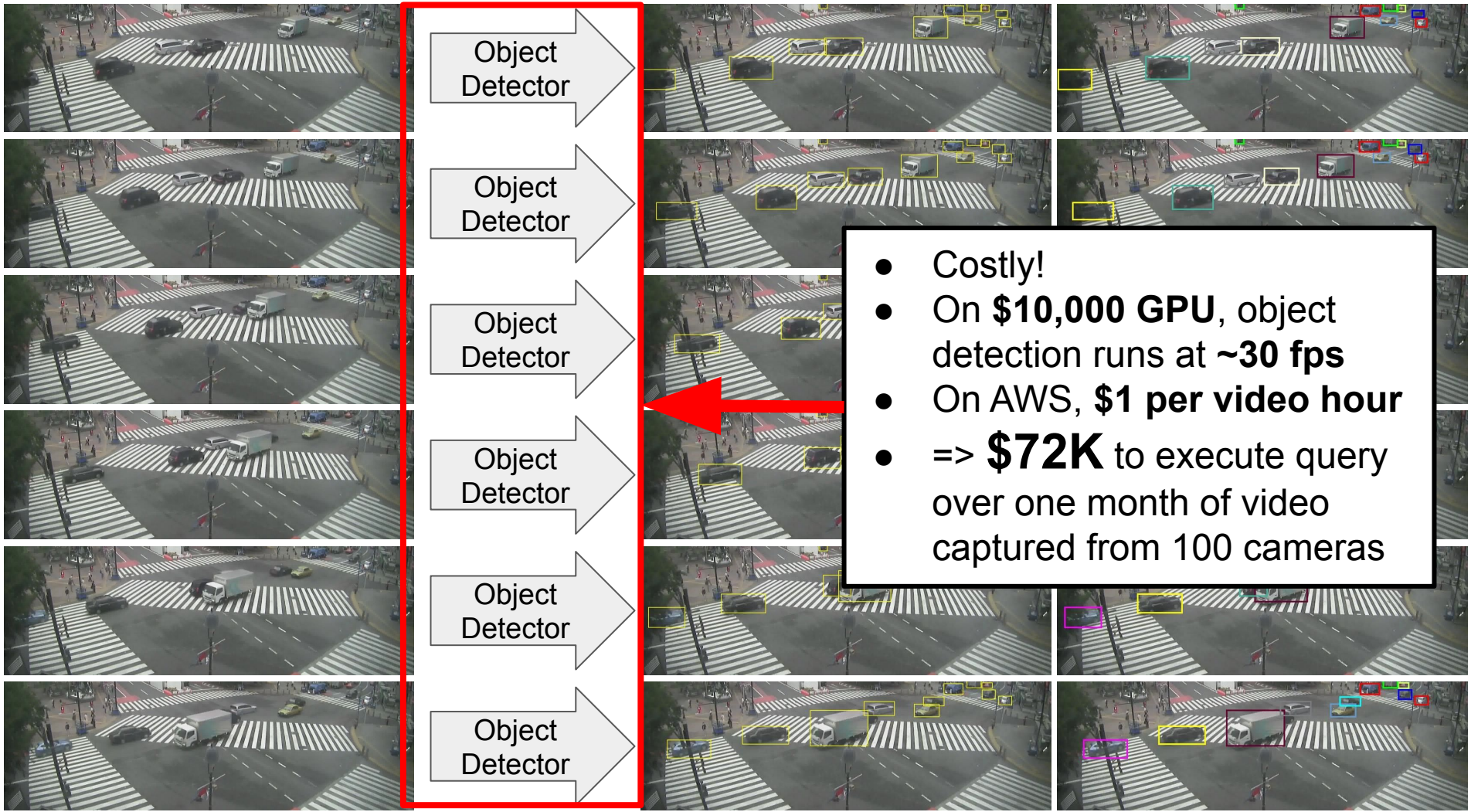
Object Detector

Object Detector

Object Detector


Object Detector
Object Detector
Object Detector
Object Detector
Object Detector
Object Detector

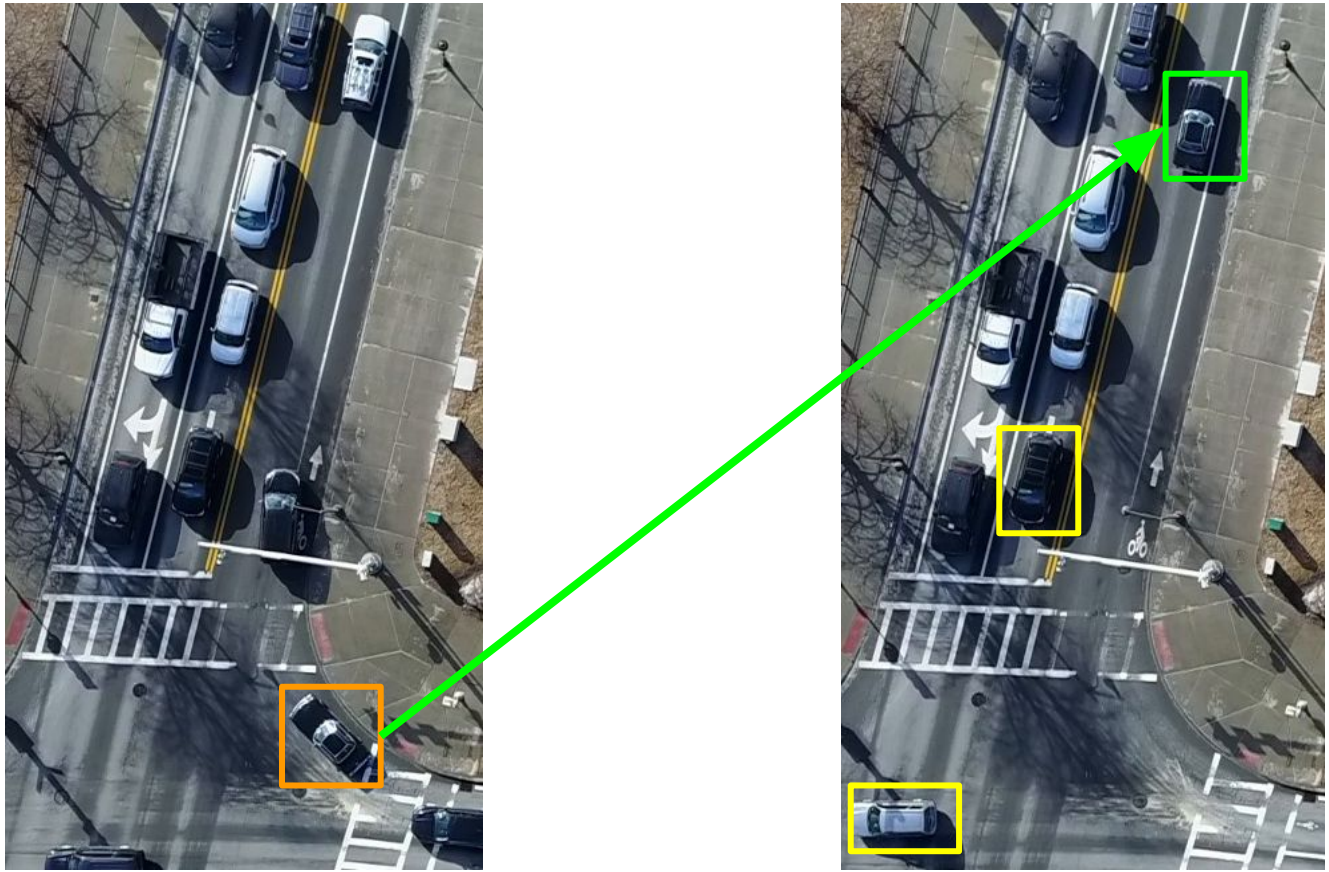

Object Detector
Object Detector
Object Detector
Object Detector
Object Detector
Object Detector


Object Detector
Object Detector
Object Detector
Object Detector
Object Detector
Object Detector
Costly!
On **$10,000 GPU**, object detection runs at **~30 fps**
On AWS, **$1 per video hour**
=> **$72K** to execute query over one month of video captured from 100 cameras

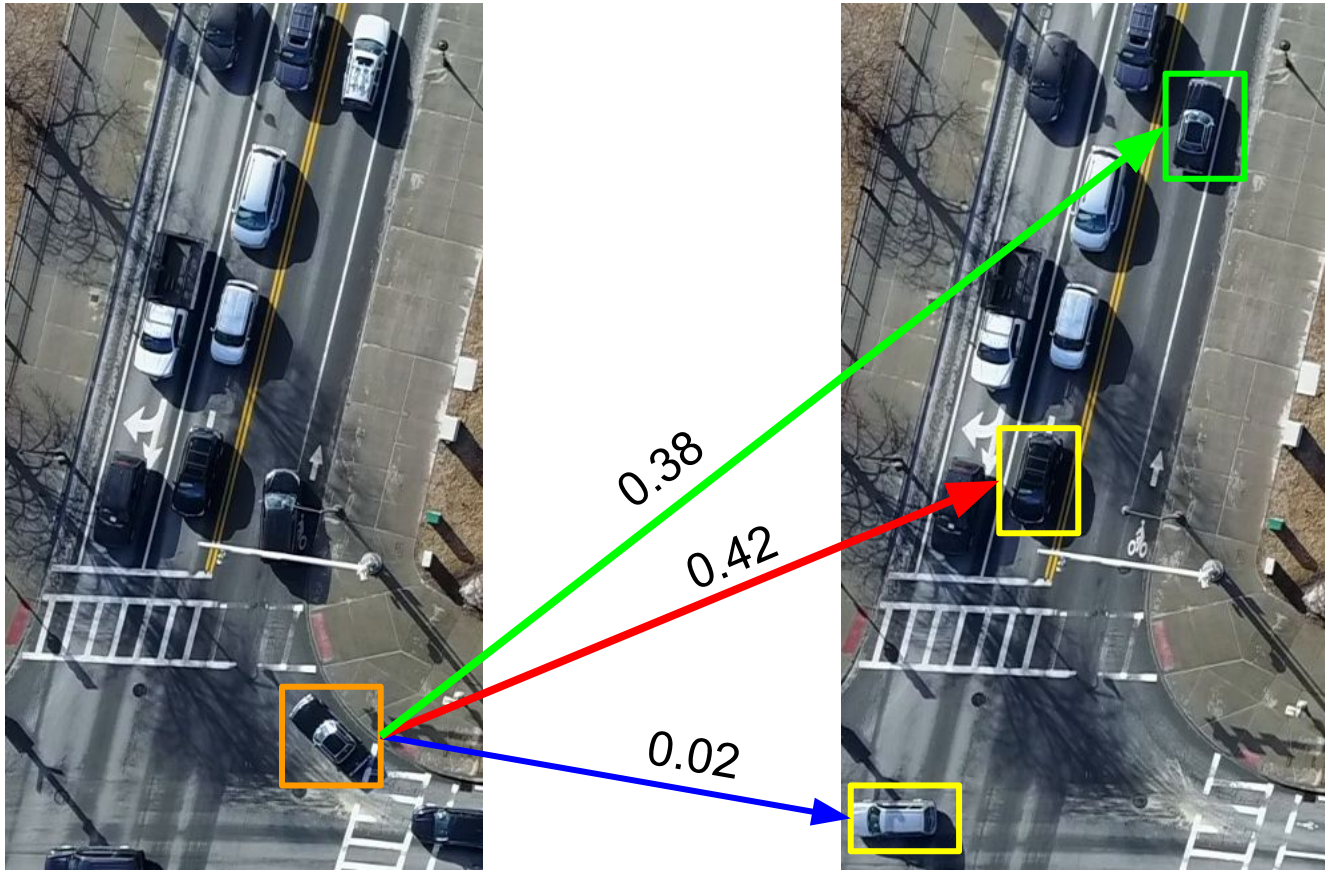
Object Detector

Object Detector

Low-Framerate Tracking: Matching Errors



Low-Framerate Tracking: Matching Errors



Low-Framerate Tracking: Predicate Errors

Predicate Errors

Predicate Errors

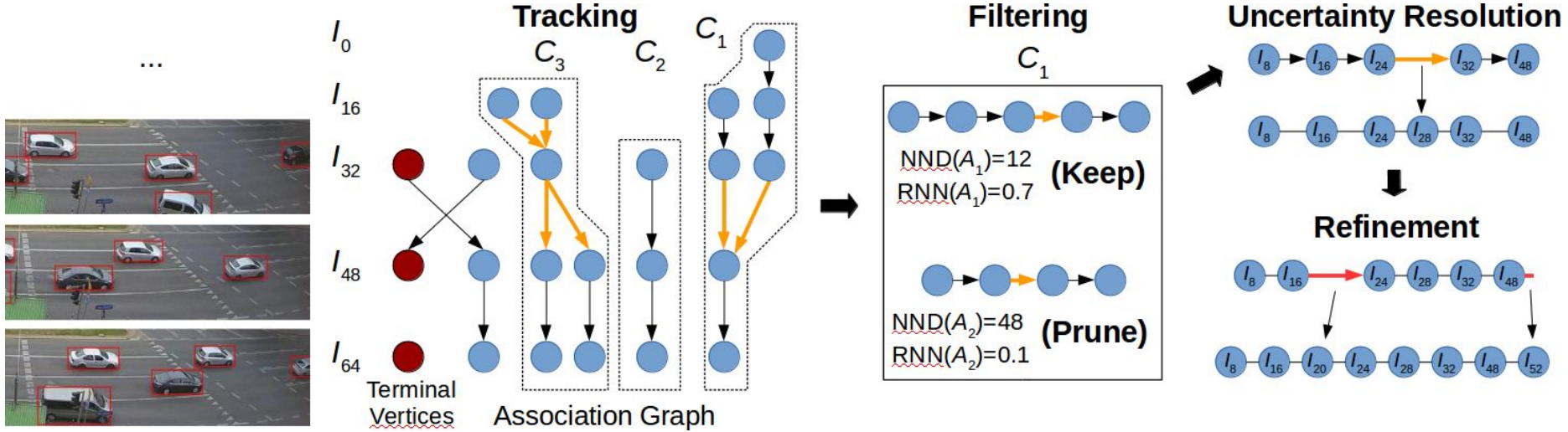
Predicate Errors

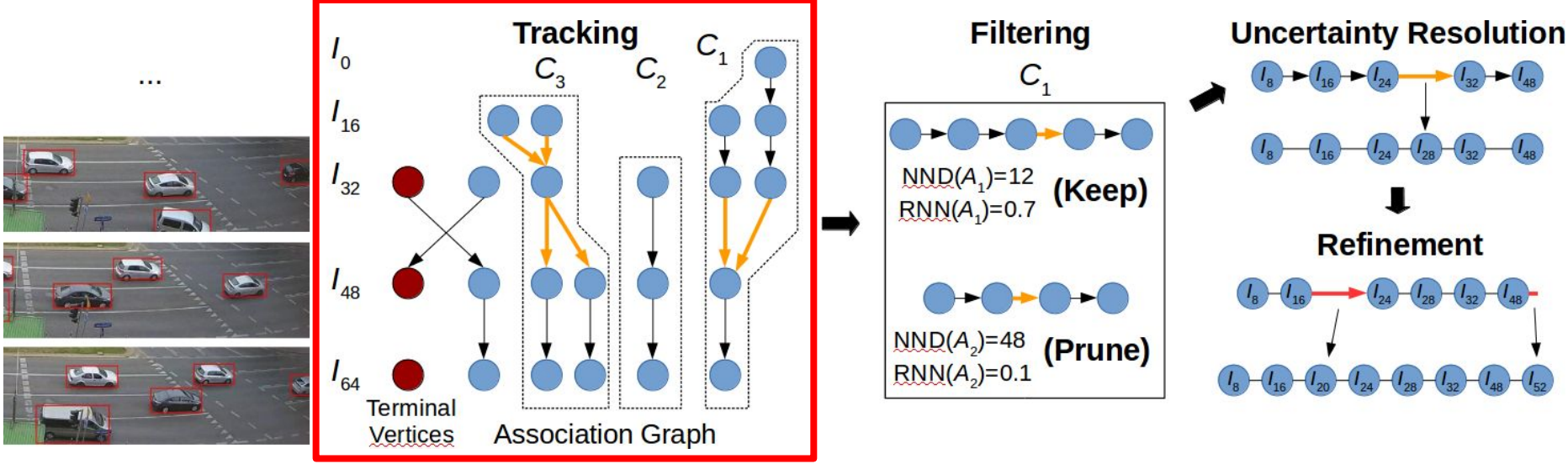
Predicate Errors

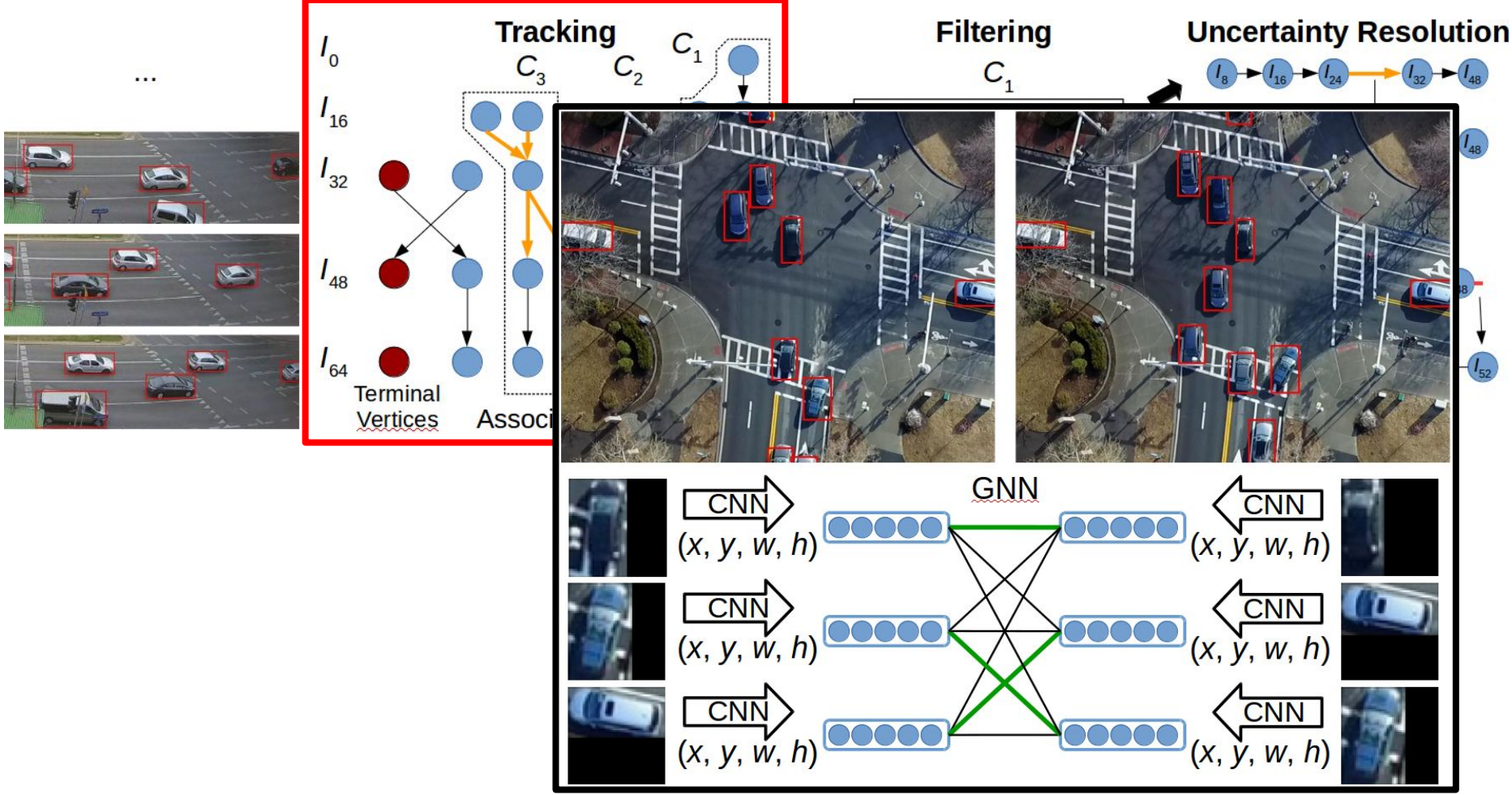
MIRIS: Fast Object Track Queries over Video

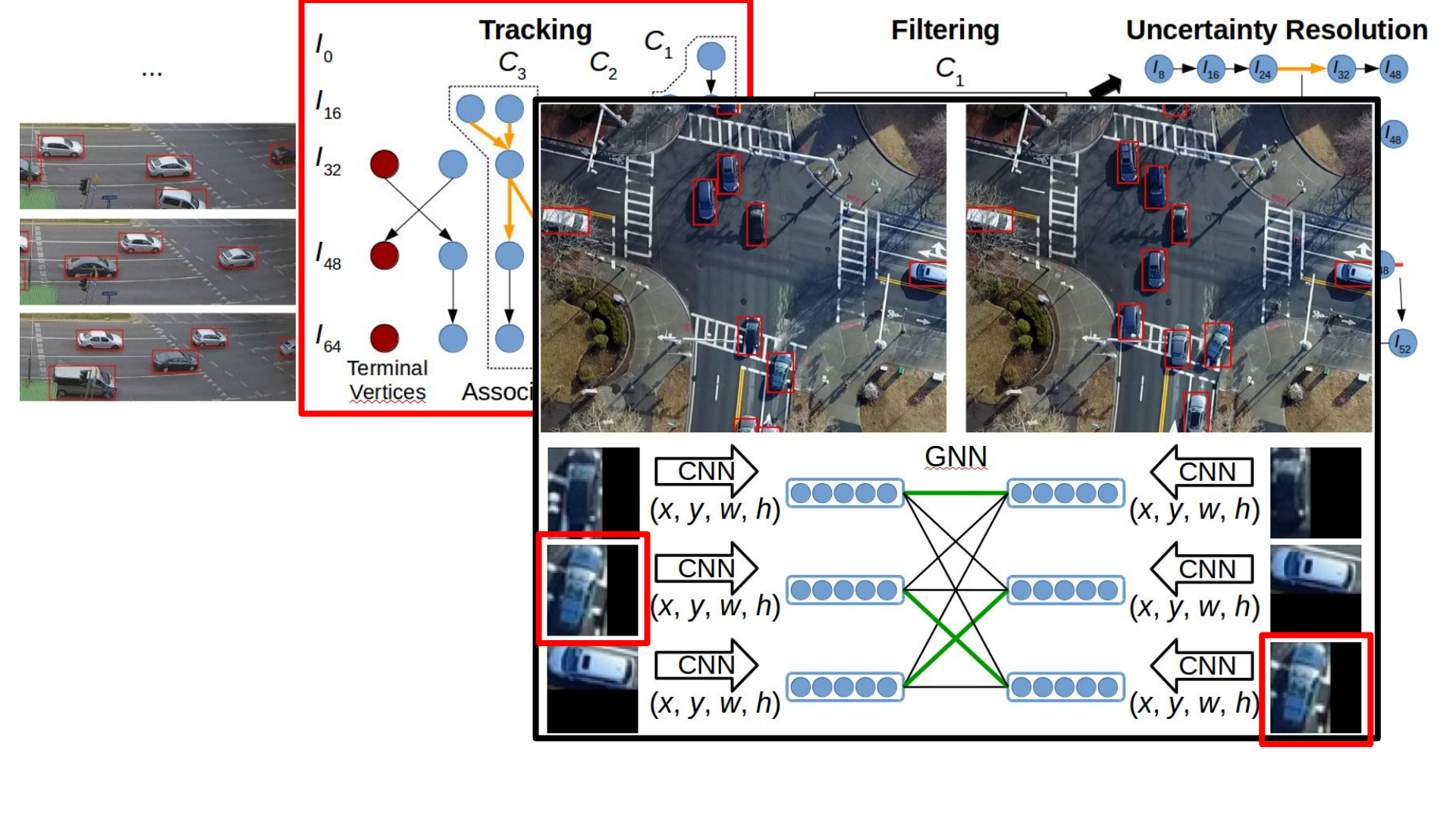
Key ideas:

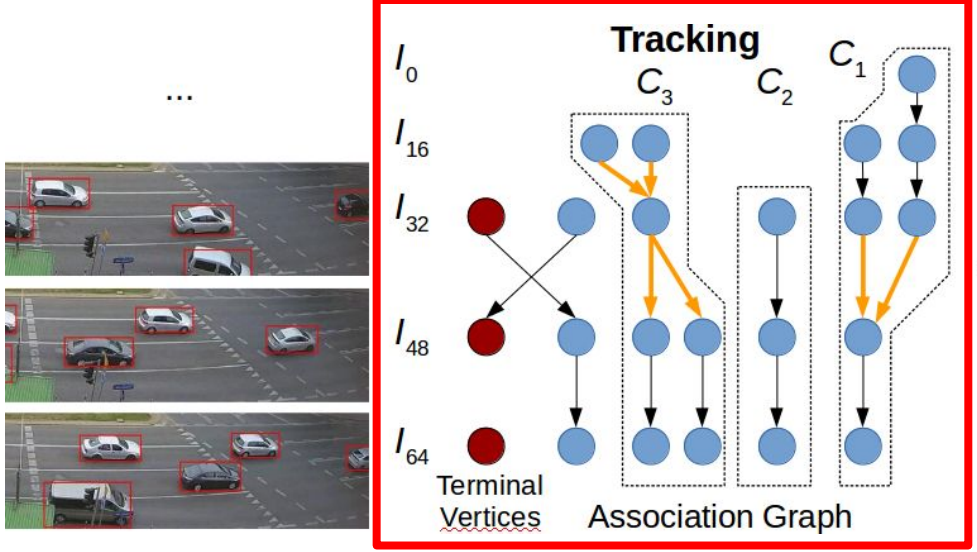
- Track at low framerate; but may need to re-visit some intermediate frames
- Query Planning + Object Tracking
  - Parameterizable query-driven object tracking method
  - Query planner to select the parameters using AQP techniques

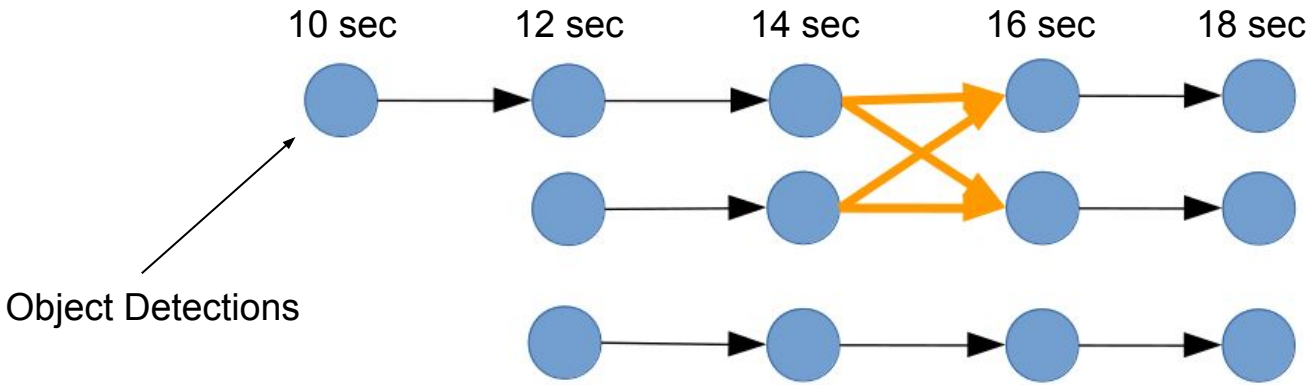


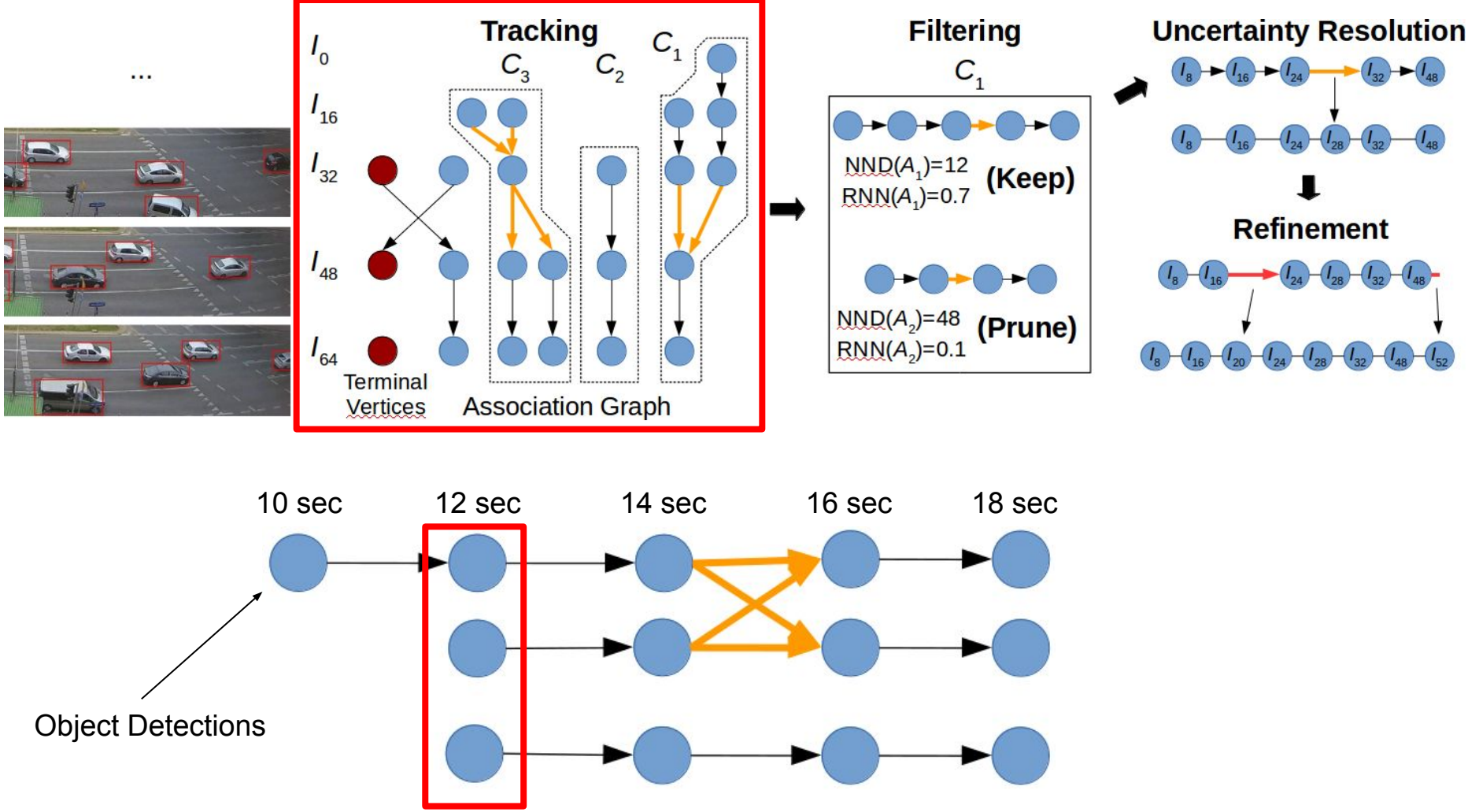

...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
I_8 I_{16} I_{24} I_{32} I_{48}
I_8 I_{16} I_{24} I_{28} I_{32} I_{48}
Refinement
I_8 I_{16} I_{24} I_{28} I_{32} I_{48}
I_8 I_{16} I_{20} I_{24} I_{28} I_{32} I_{48} I_{52}

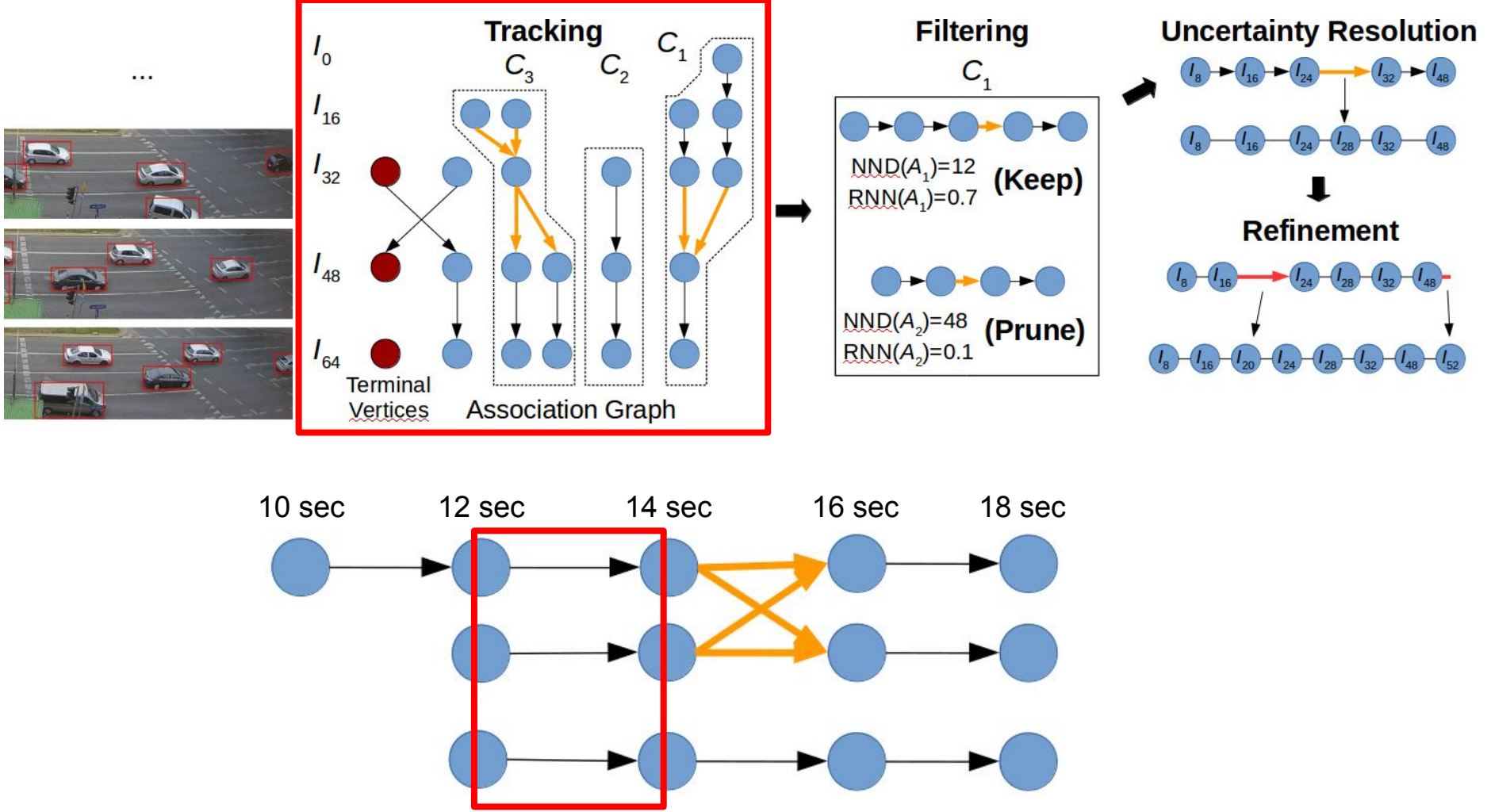

...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_1
C_3
C_2
Terminal Vertices
Filtering
C_1
Uncertainty Resolution
I_8
I_{16}
I_{24}
I_{32}
I_{48}
I_{48}
I_{52}
GNN
CNN
(x, y, w, h)
CNN
(x, y, w, h)
CNN
(x, y, w, h)
CNN
(x, y, w, h)
CNN
(x, y, w, h)
CNN
(x, y, w, h)

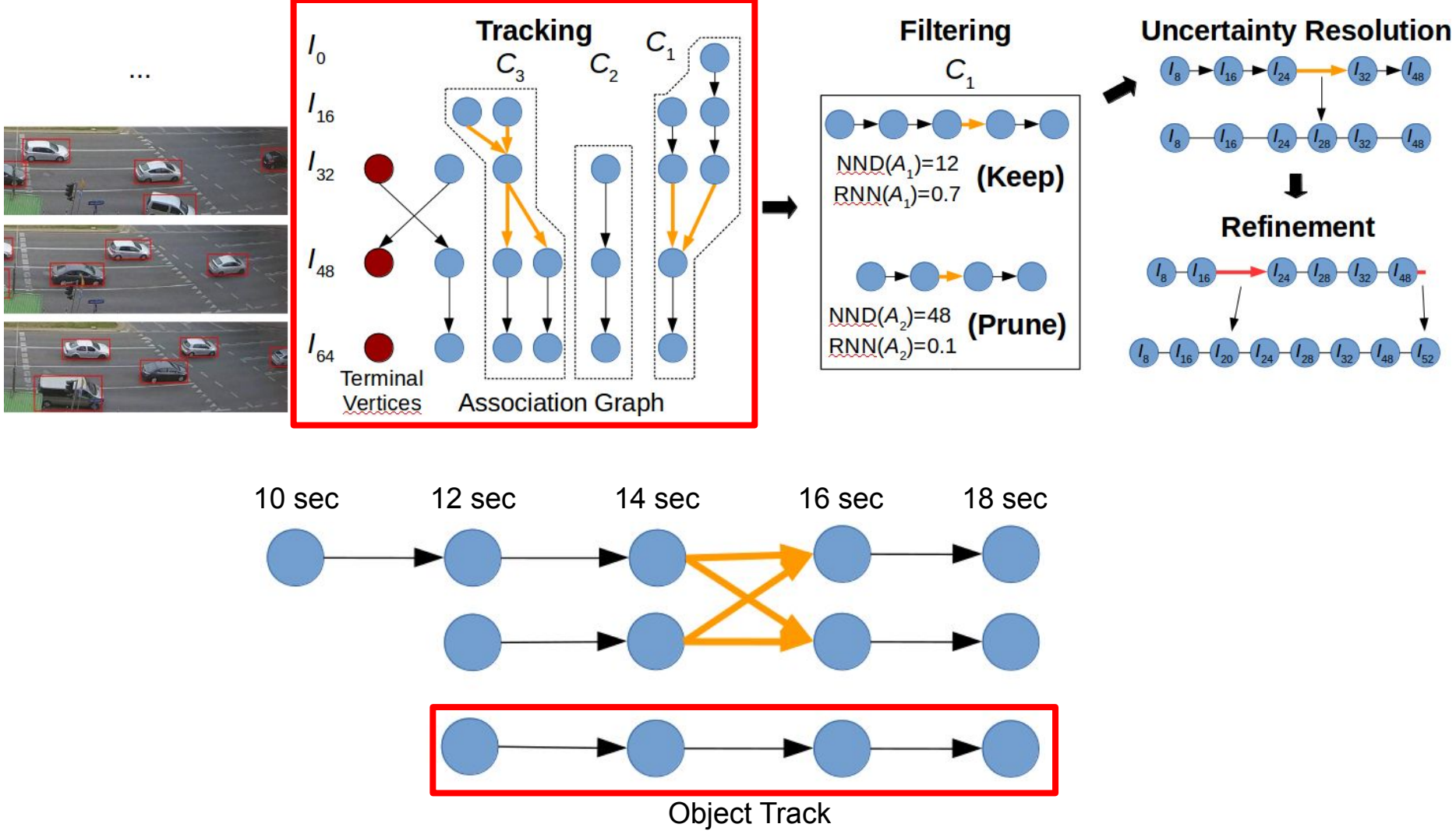

...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_1
C_2
C_3
Terminal Vertices
Filtering
C_1
Uncertainty Resolution
I_8
I_{16}
I_{24}
I_{32}
I_{48}
I_{52}
GNN
CNN
(x, y, w, h)

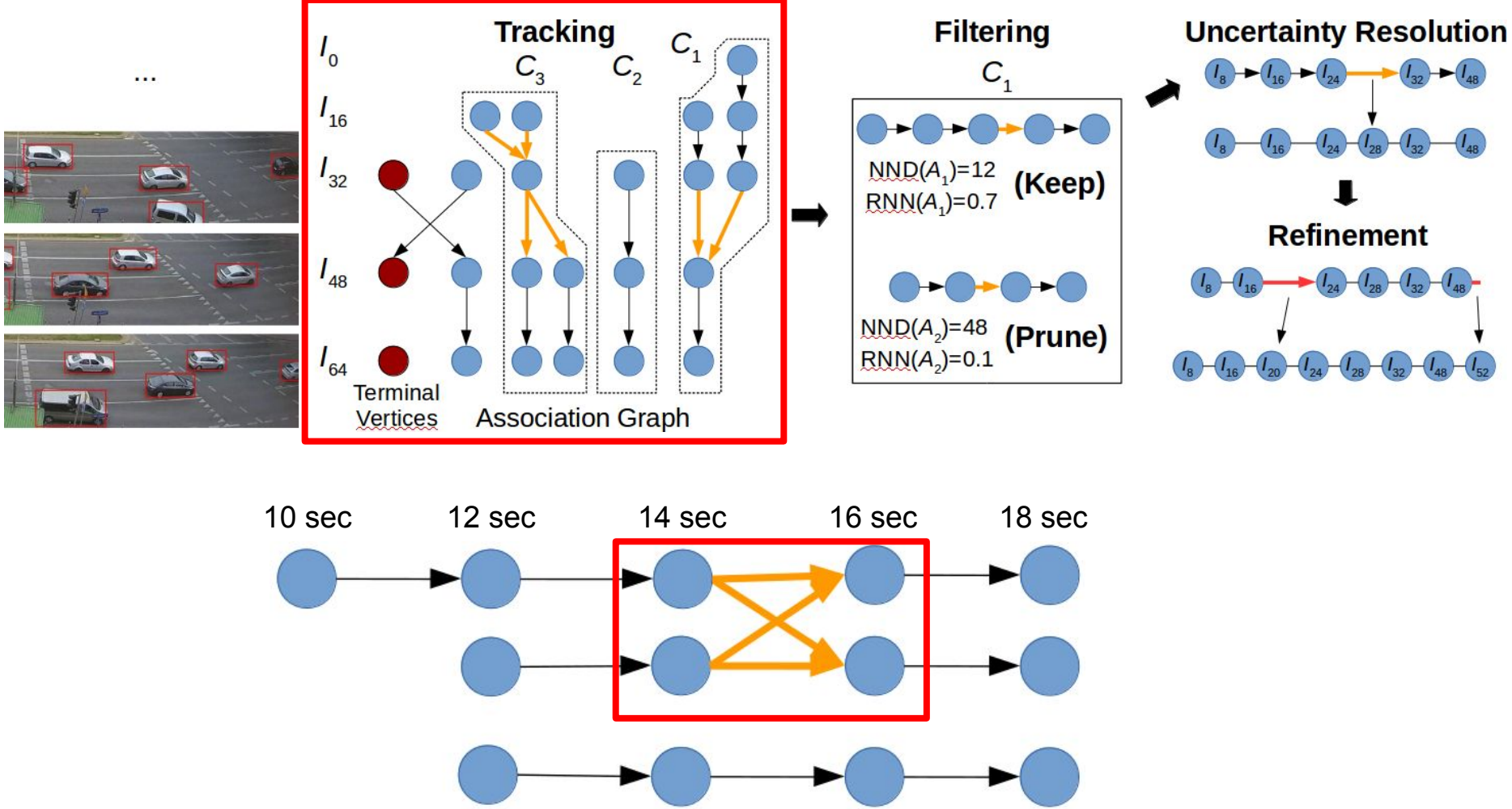

...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
Refinement


10 sec
12 sec
14 sec
16 sec
18 sec
Object Detections


...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
Refinement
10 sec
12 sec
14 sec
16 sec
18 sec
Object Detections

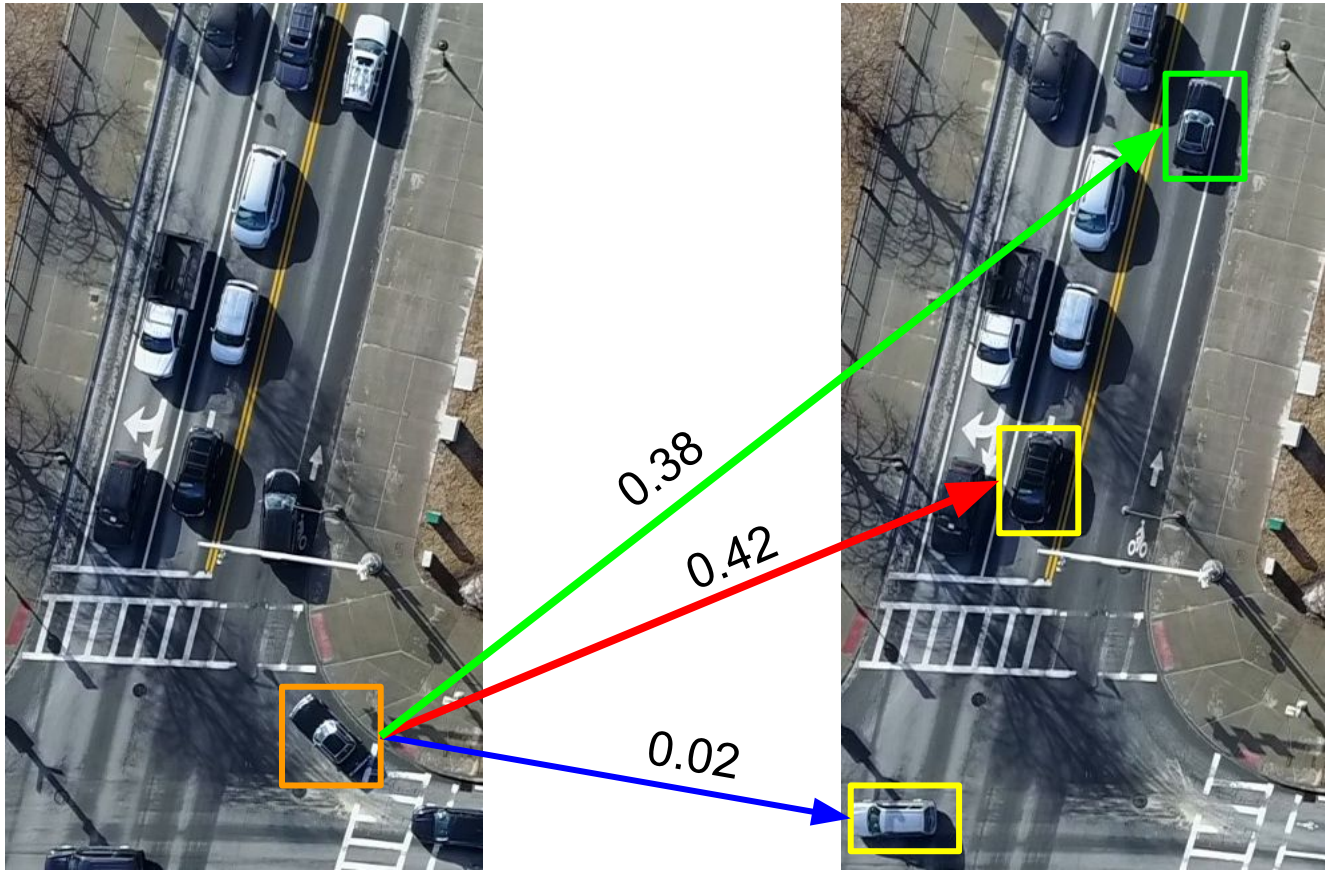

...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
Refinement
I_8 I_{16} I_{20} I_{24} I_{28} I_{32} I_{48} I_{52}
10 sec
12 sec
14 sec
16 sec
18 sec


...
Tracking
l_0
l_{16}
l_{32}
l_{48}
l_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
l_8 l_{16} l_{24} l_{32} l_{48}
l_8 l_{16} l_{24} l_{28} l_{32} l_{48}
Refinement
l_8 l_{16} l_{24} l_{28} l_{32} l_{48}
l_8 l_{16} l_{20} l_{24} l_{28} l_{32} l_{48} l_{52}
10 sec
12 sec
14 sec
16 sec
18 sec
Object Track

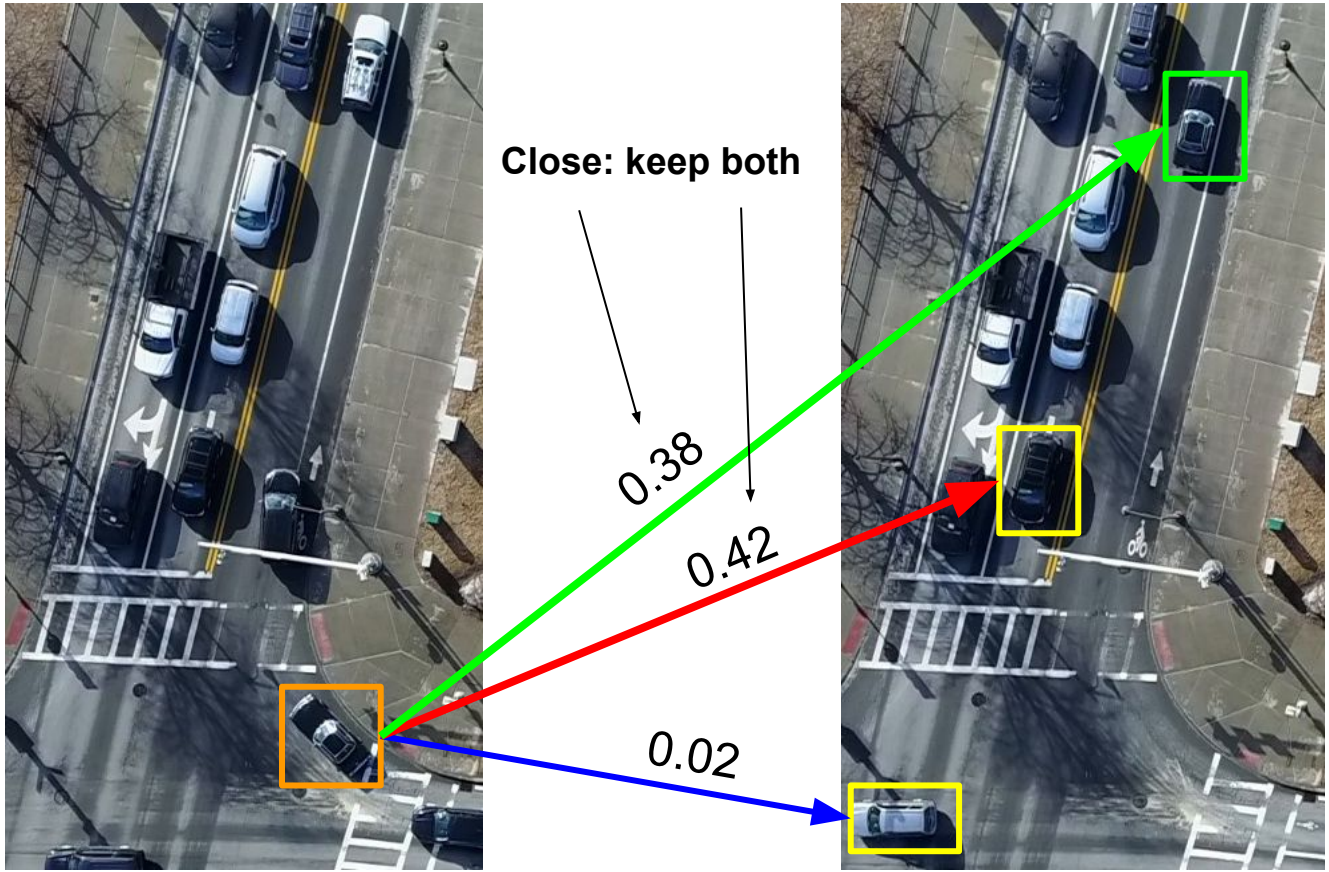

...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
Refinement
I_8 I_{16} I_{20} I_{24} I_{28} I_{32} I_{48} I_{52}

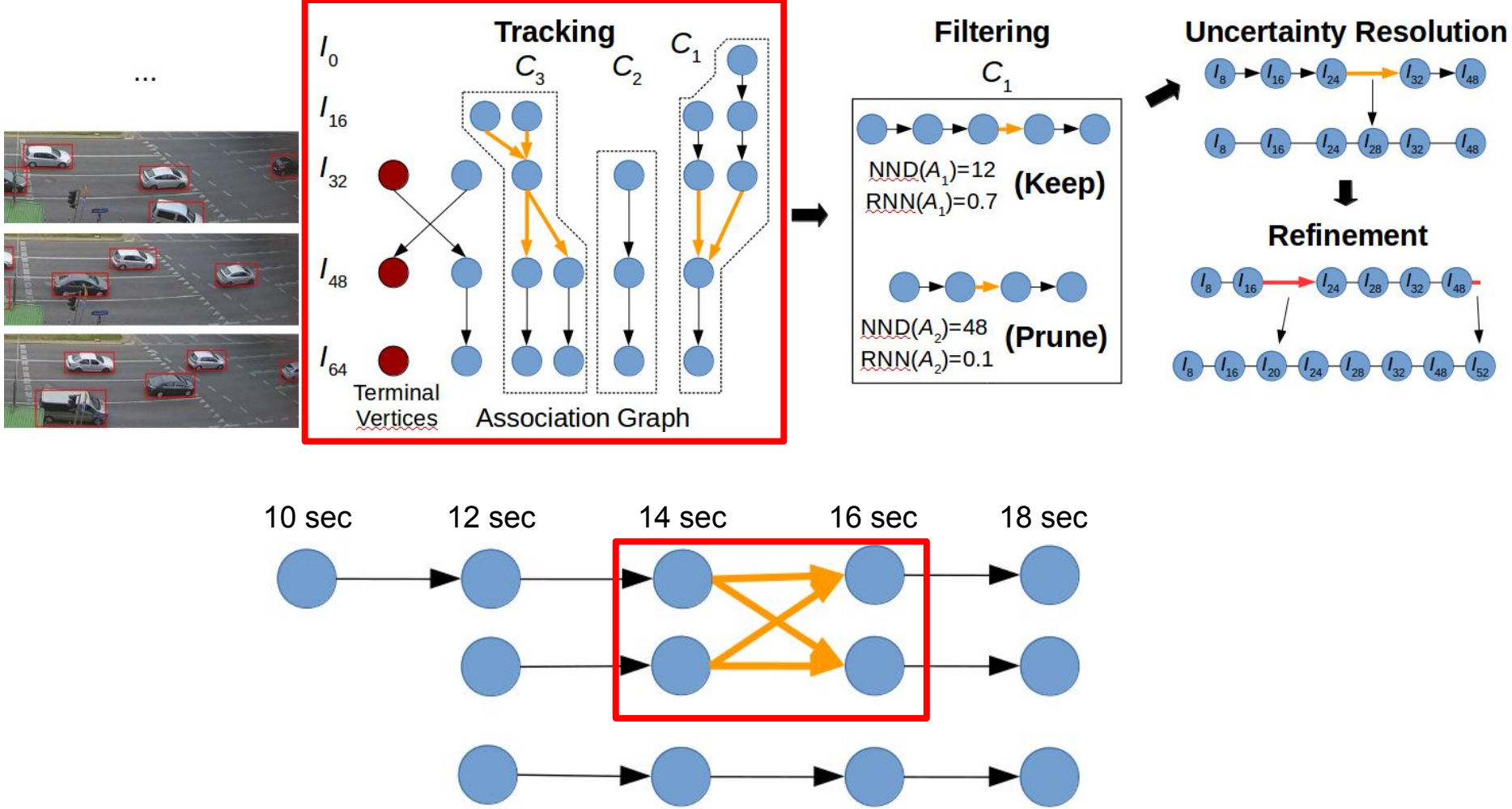
10 sec
12 sec
14 sec
16 sec
18 sec

Low-Framerate Tracking: Matching Errors

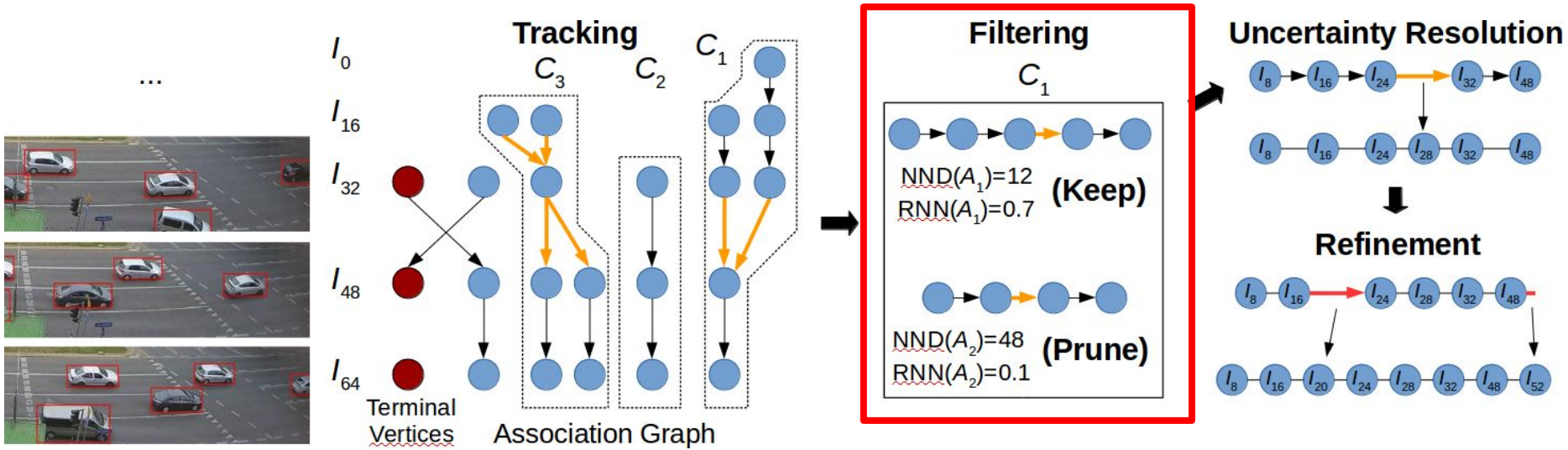


Low-Framerate Tracking: Matching Errors



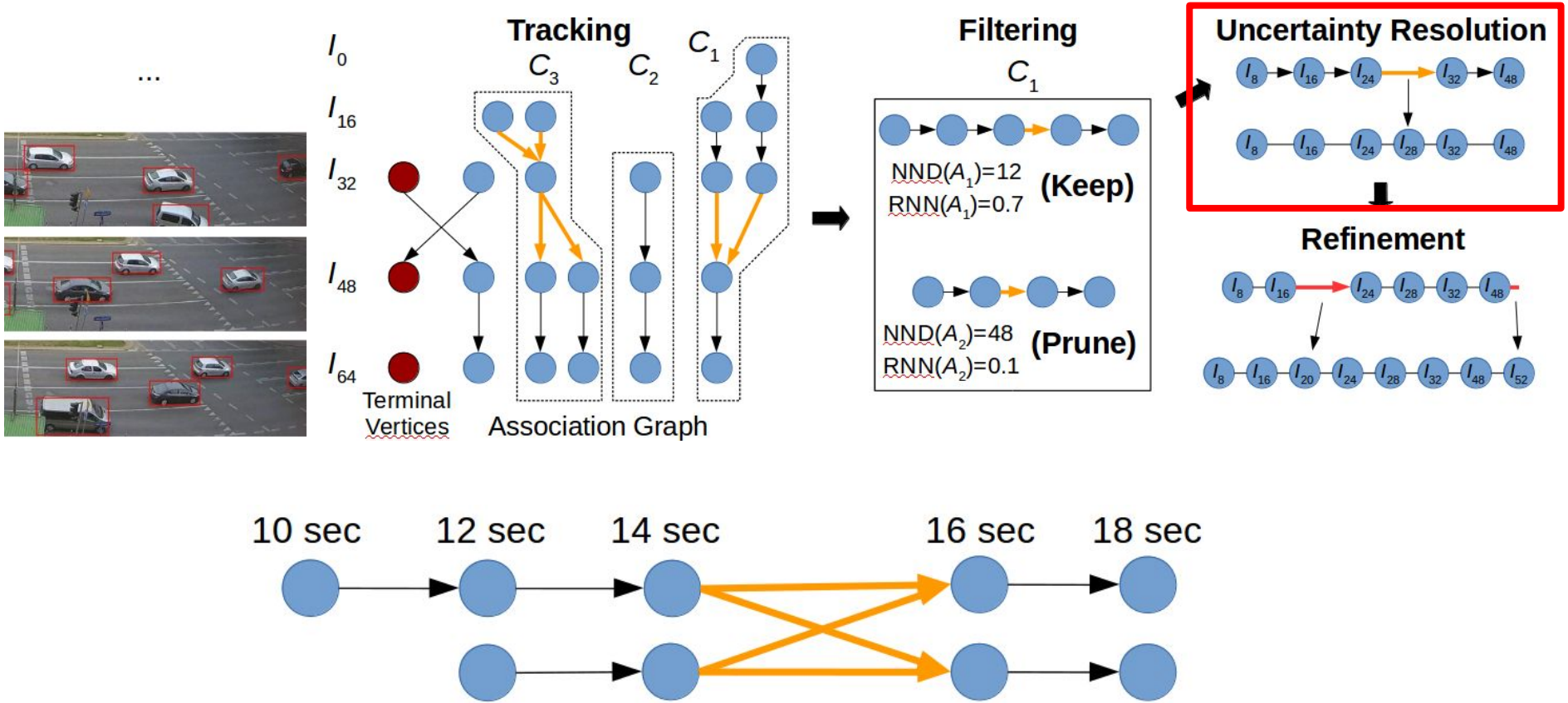

...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
Refinement
I_8 I_{16} I_{20} I_{24} I_{28} I_{32} I_{48} I_{52}

10 sec
12 sec
14 sec
16 sec
18 sec

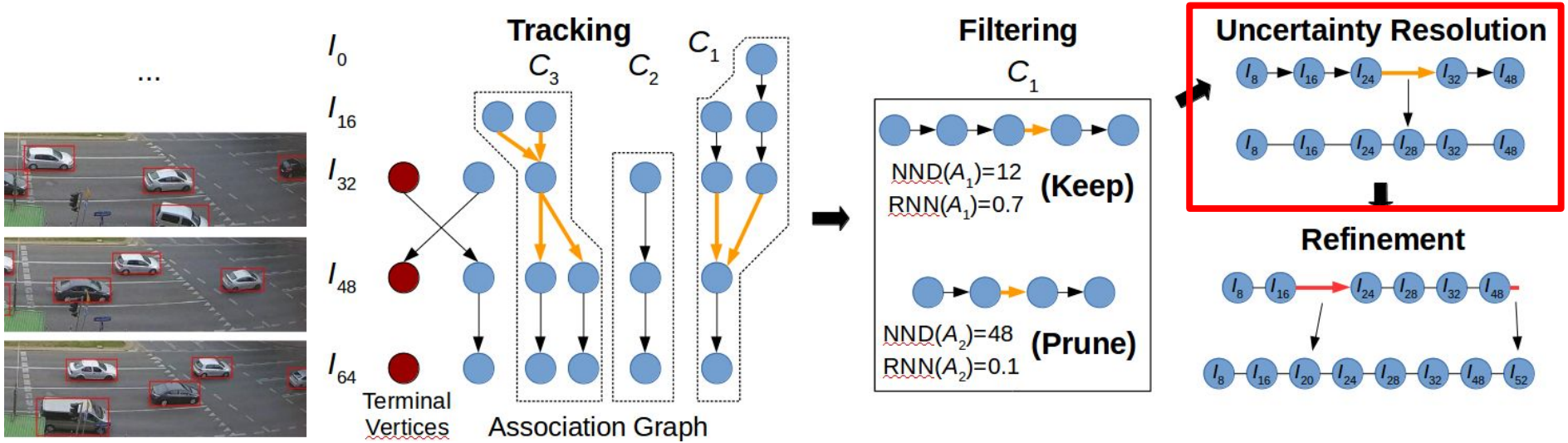


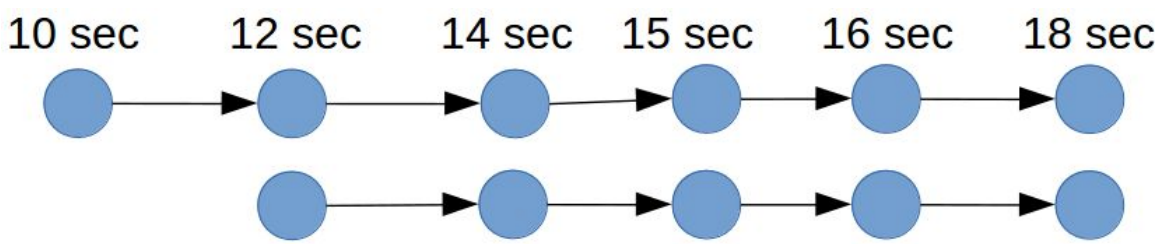
Filtering

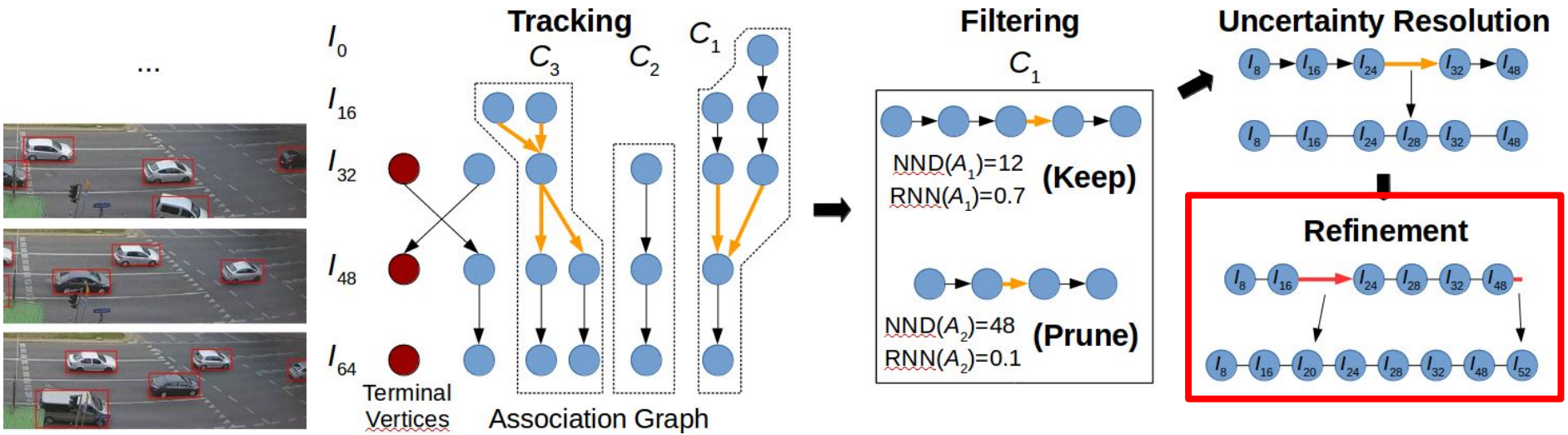
- Remove groups of paths that we are sure do not satisfy the predicate
- Several filtering methods for planner to choose from: nearest-neighbor, RNN

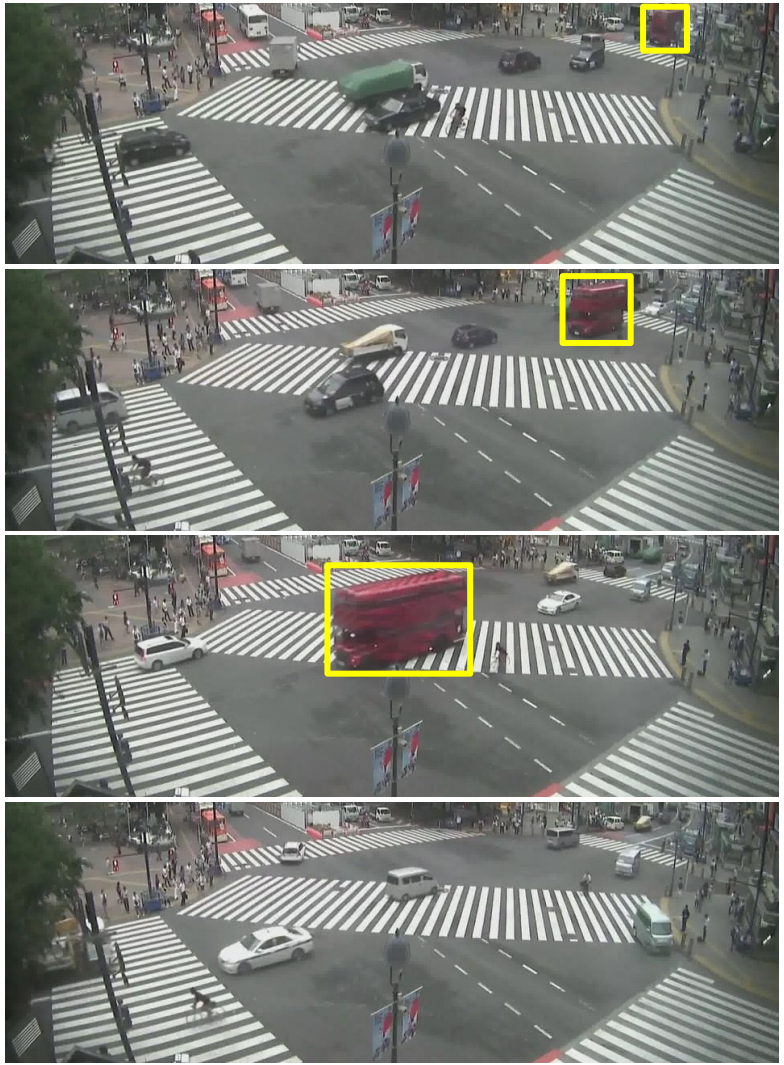

...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
I_8 I_{16} I_{24} I_{32} I_{48}
I_8 I_{16} I_{24} I_{28} I_{32} I_{48}
Refinement
I_8 I_{16} I_{24} I_{28} I_{32} I_{48}
I_8 I_{16} I_{20} I_{24} I_{28} I_{32} I_{48} I_{52}

10 sec
12 sec
14 sec
16 sec
18 sec

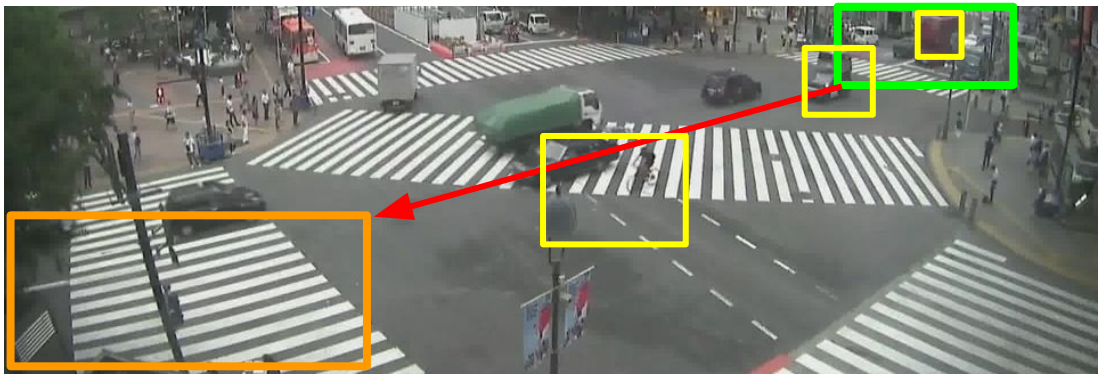

...
I_0
I_{16}
I_{32}
I_{48}
I_{64}
Terminal Vertices
Tracking
C_3
C_2
C_1
Association Graph
Filtering
C_1
NND(A_1)=12
RNN(A_1)=0.7
(Keep)
NND(A_2)=48
RNN(A_2)=0.1
(Prune)
Uncertainty Resolution
I_8 I_{16} I_{24} I_{32} I_{48}
I_8 I_{16} I_{24} I_{28} I_{32} I_{48}
Refinement
I_8 I_{16} I_{24} I_{28} I_{32} I_{48}
I_8 I_{16} I_{20} I_{24} I_{28} I_{32} I_{48} I_{52}


10 sec
12 sec
14 sec
15 sec
16 sec
18 sec


...
Tracking
I_0
I_{16}
I_{32}
I_{48}
I_{64}
C_3
C_2
C_1
Terminal Vertices
Association Graph
Filtering
C_1
$NND(A_1)=12$
$RNN(A_1)=0.7$
(Keep)
$NND(A_2)=48$
$RNN(A_2)=0.1$
(Prune)
Uncertainty Resolution
I_8 I_{16} I_{24} I_{32} I_{48}
I_8 I_{16} I_{24} I_{28} I_{32} I_{48}
Refinement
I_8 I_{16} I_{24} I_{28} I_{32} I_{48}
I_8 I_{16} I_{20} I_{24} I_{28} I_{32} I_{48} I_{52}



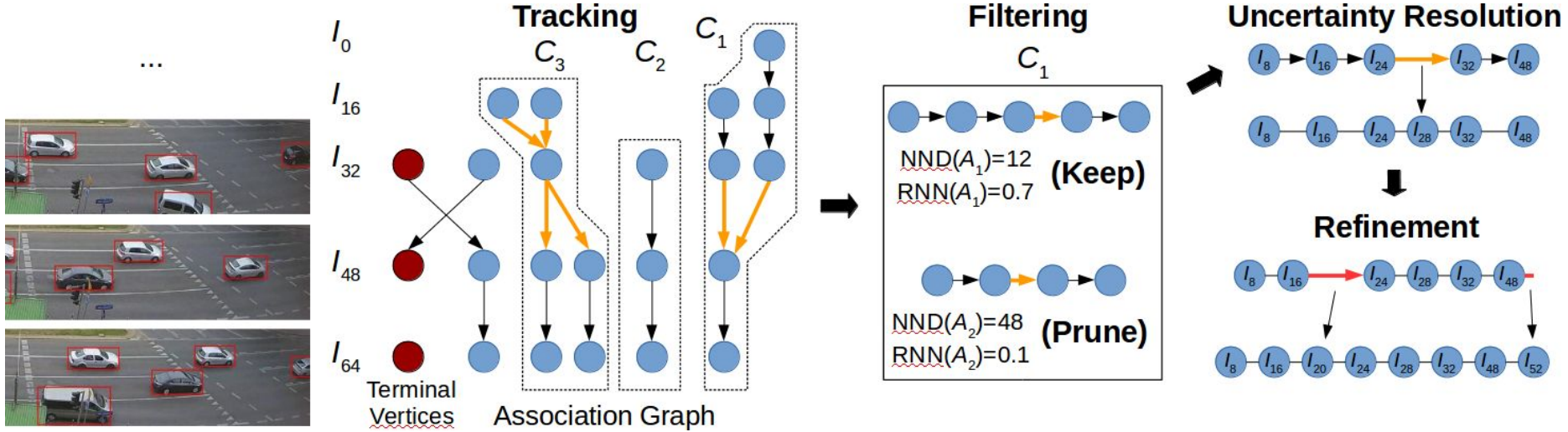
Refinement:
Address Predicate Errors



MIRIS: Fast Object Track Queries over Video

Key ideas:

- Track at low framerate; but may need to re-visit some intermediate frames
- Query Planning + Object Tracking
  - Parameterizable query-driven object tracking method
  - Query planner to select the parameters using AQP techniques



Query Planning

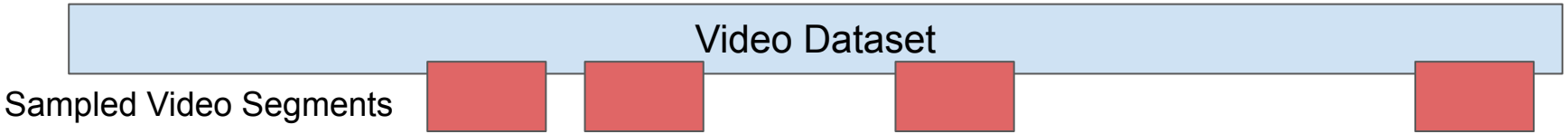
Select tracks satisfying *P*, with 99% accuracy.

Query Planning

Select tracks satisfying *P*, **with 99% accuracy**.

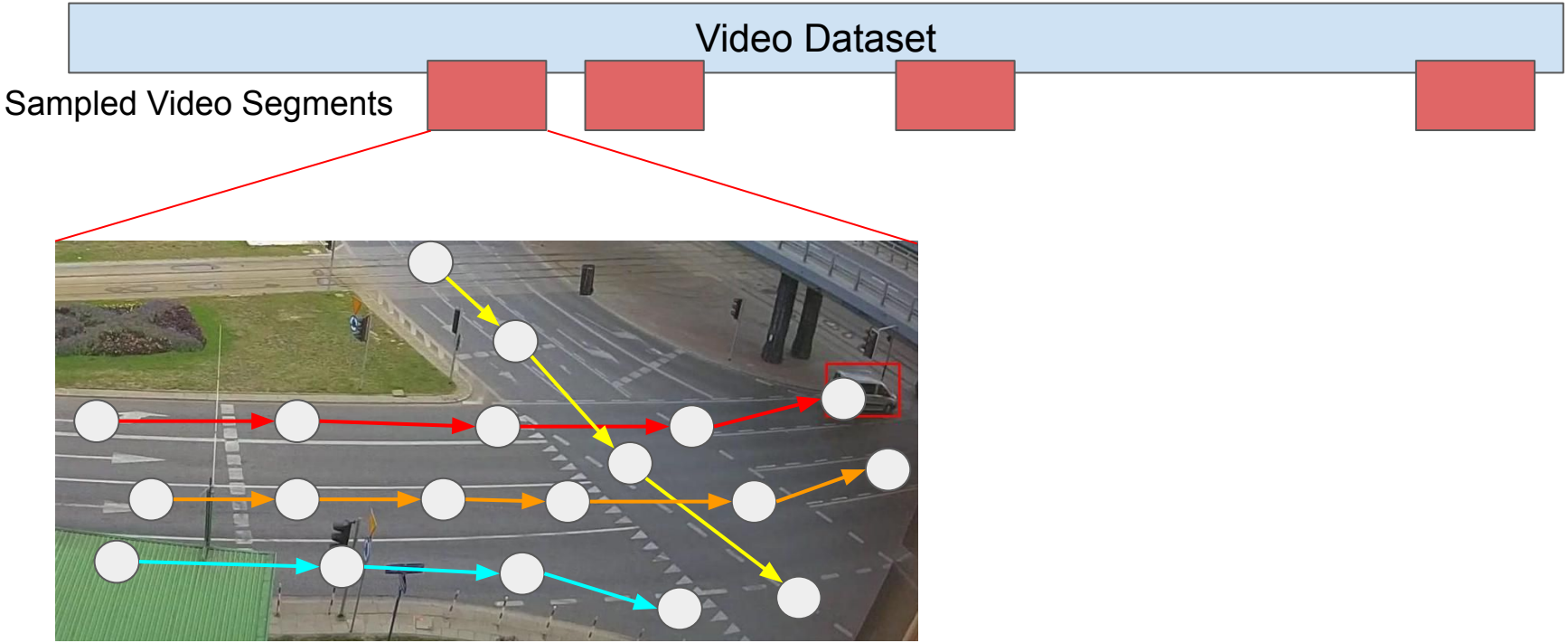
Query Planning

Select tracks satisfying *P*, with 99% accuracy.



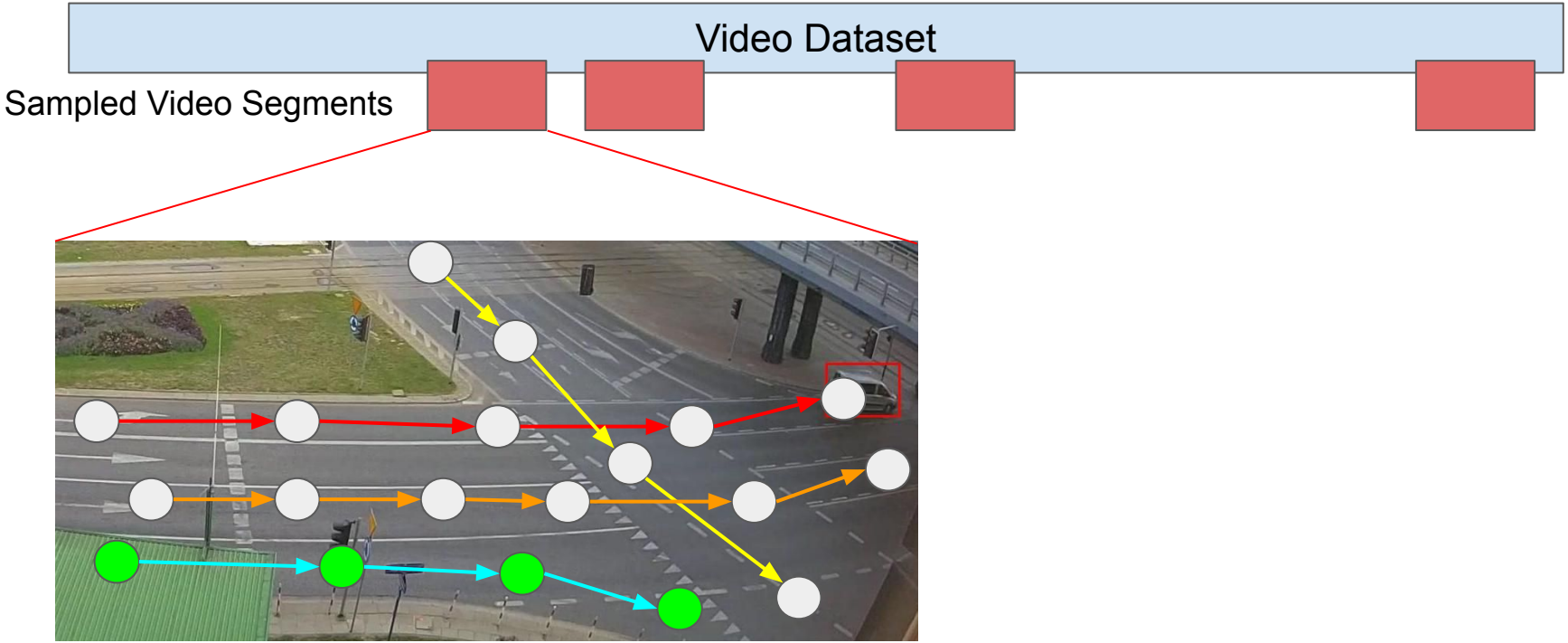
Query Planning

Select tracks satisfying *P*, with 99% accuracy.



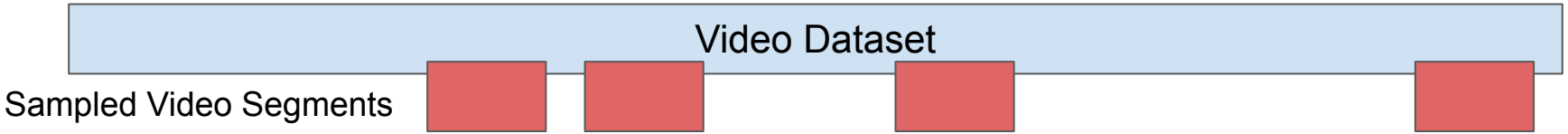
Query Planning

Select tracks satisfying *P*, with 99% accuracy.



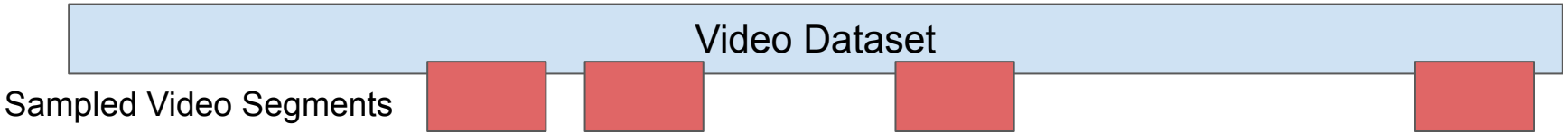
Query Planning

Select tracks satisfying *P*, with 99% accuracy.



Query Planning

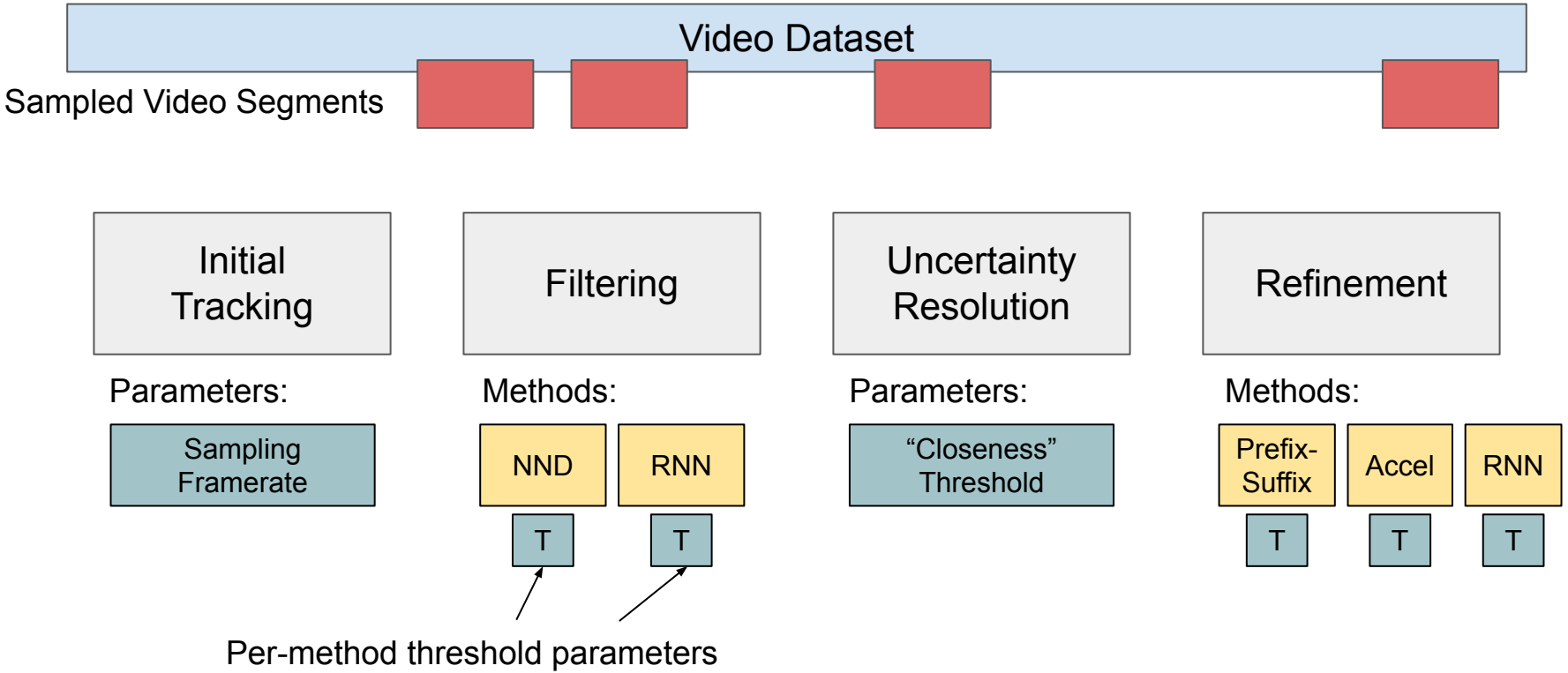
Select tracks satisfying *P*, with 99% accuracy.



Initial Tracking	Filtering	Uncertainty Resolution	Refinement
Parameters:		Parameters:	
Sampling Framerate		“Closeness” Threshold	

Query Planning

Select tracks satisfying *P*, with 99% accuracy.



Evaluation: 9 Queries over 5 Video Sources

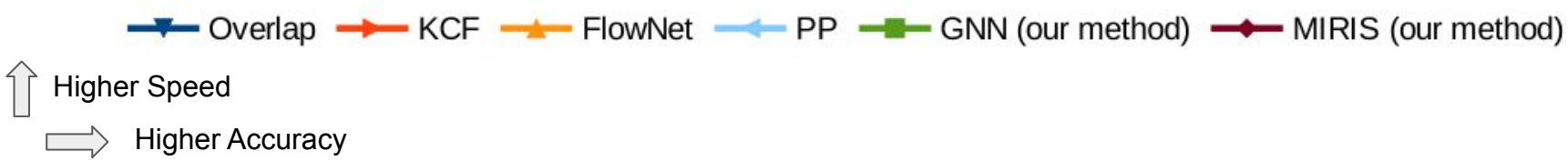
Diverse range of video sources:

- UAV: video captured by UAV over traffic junction
- Tokyo, Warsaw: video captured by fixed traffic camera
- Resort: video of a pedestrian walkway
- BDD: dashcam video

Four baselines:

- Overlap-based tracking [1]
- Kernel correlation filters (KCF) [2]
- FlowNet [3]
- Probabilistic predicates [4, 5, 6]

[1] Simple Online and Realtime Tracking. Alex Bewley et al. ICIP 2016.
[2] High-Speed Tracking with Kernelized Correlation Filters. Joao Henriques et al. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.
[3] FlowNet: Learning Optical Flow with Convolutional Networks. Alexey Dosovitskiy et al. ICCV 2015.
[4] NoScope: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[5] Accelerating Machine Learning Inference with Probabilistic Predicates. Yao Lu et al. SIGMOD 2018.
[6] BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.

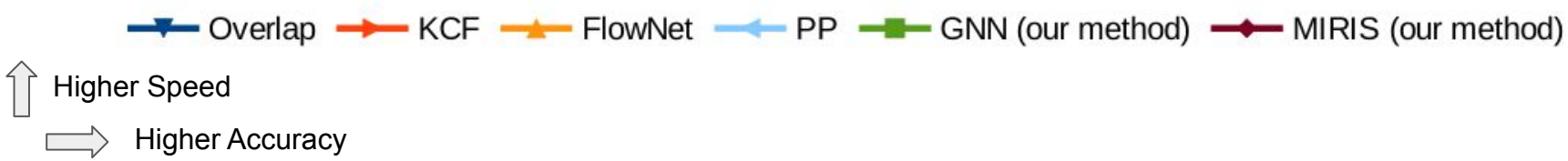


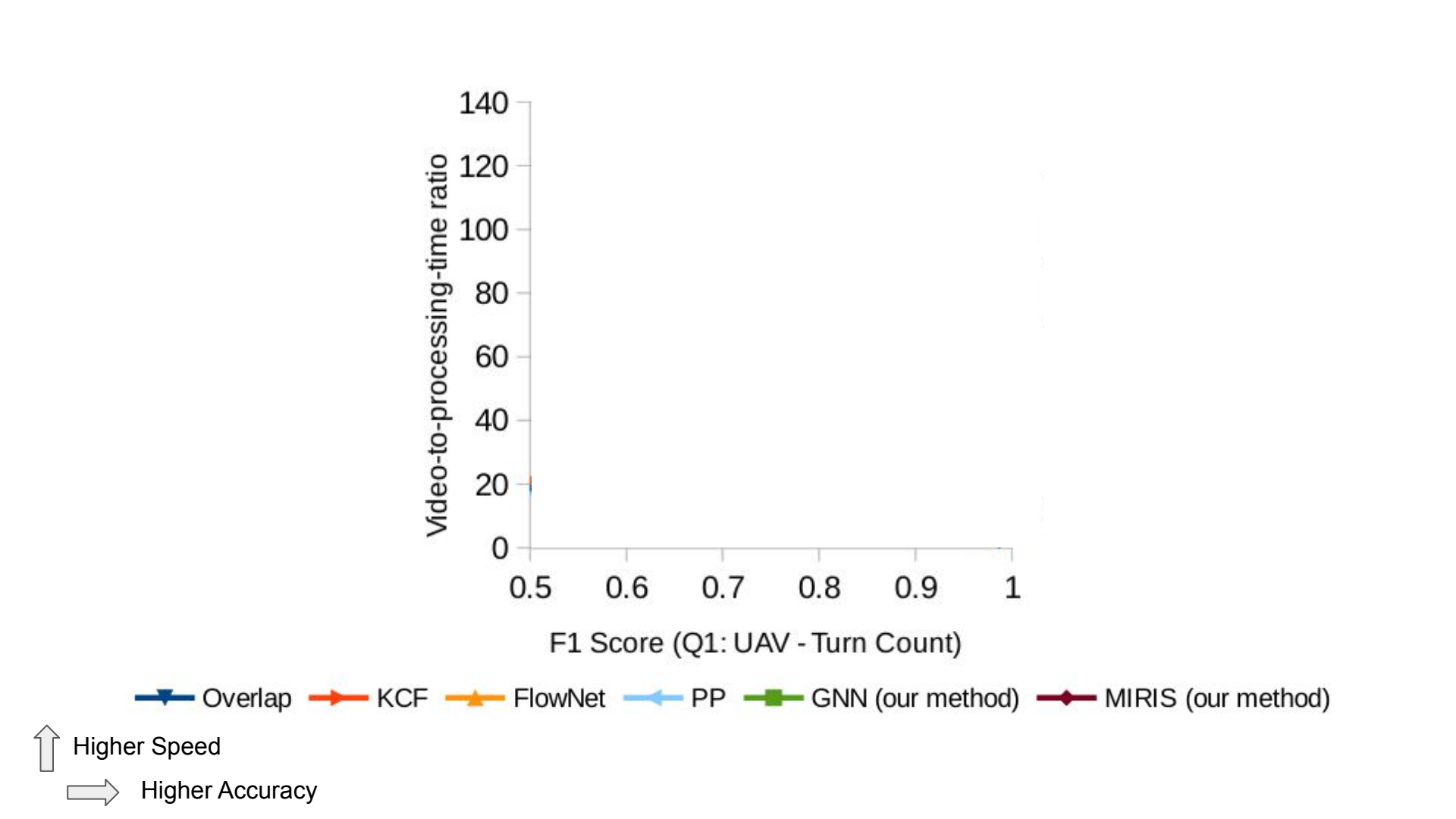
Four baselines:

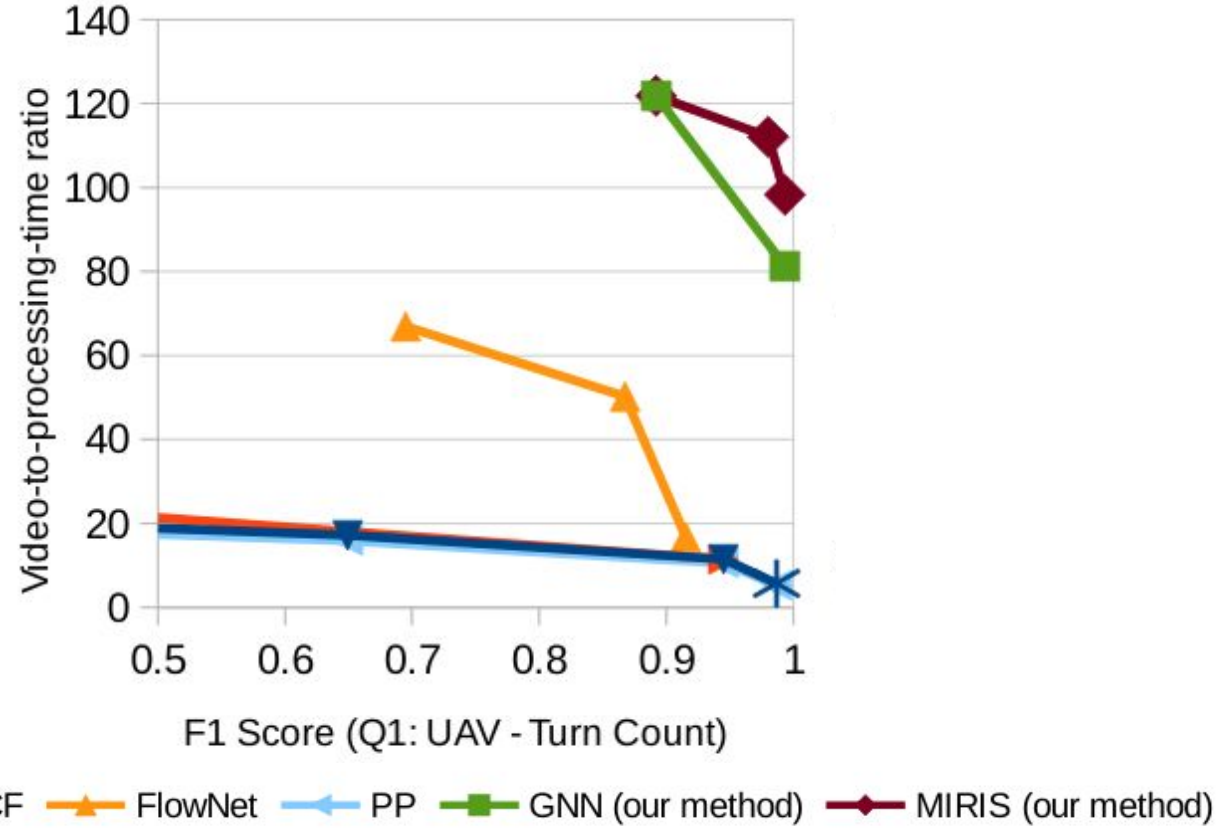
- Overlap-based tracking [1]
- Kernel correlation filters (KCF) [2]
- FlowNet [3]
- Probabilistic predicates [4, 5, 6]

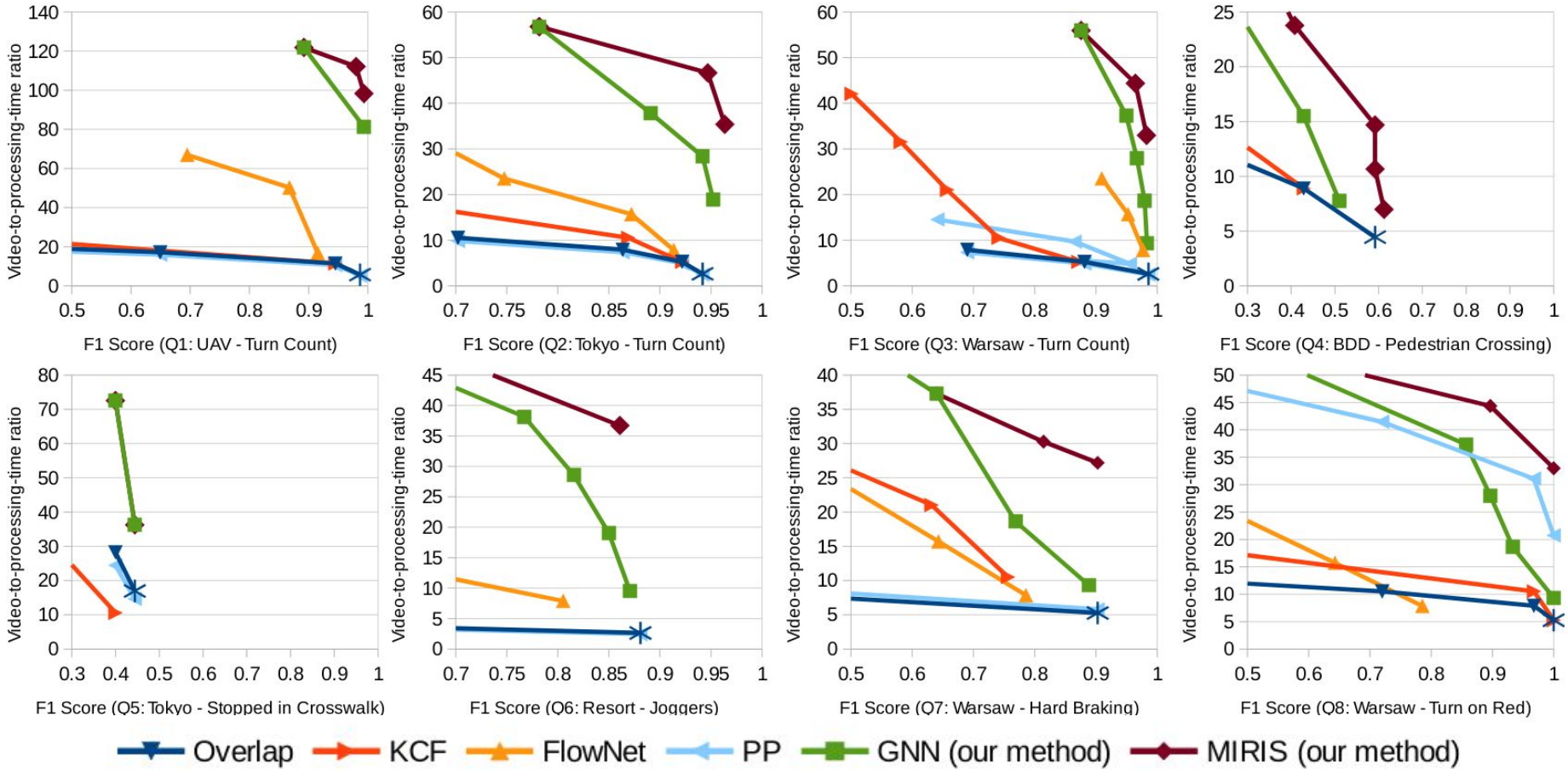
GNN: apply our tracker model without filtering, uncertainty resolution, and refinement

[1] Simple Online and Realtime Tracking. Alex Bewley et al. ICIP 2016.
[2] High-Speed Tracking with Kernelized Correlation Filters. Joao Henriques et al. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014.
[3] FlowNet: Learning Optical Flow with Convolutional Networks. Alexey Dosovitskiy et al. ICCV 2015.
[4] NoScope: Optimizing Neural Network Queries over Video at Scale. Daniel Kang et al. VLDB 2017.
[5] Accelerating Machine Learning Inference with Probabilistic Predicates. Yao Lu et al. SIGMOD 2018.
[6] BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. Daniel Kang et al. VLDB 2020.




140
120
100
80
60
40
20
0
Video-to-processing-time ratio
0.5
0.6
0.7
0.8
0.9
1
F1 Score (Q1: UAV - Turn Count)
Overlap
KCF
FlowNet
PP
GNN (our method)
MIRIS (our method)
Higher Speed
Higher Accuracy


140
120
100
80
60
40
20
0
Video-to-processing-time ratio
0.5
0.6
0.7
0.8
0.9
1
F1 Score (Q1: UAV - Turn Count)
Overlap
KCF
FlowNet
PP
GNN (our method)
MIRIS (our method)
Higher Speed
Higher Accuracy


Video-to-processing-time ratio
F1 Score (Q1: UAV - Turn Count)
F1 Score (Q2: Tokyo - Turn Count)
F1 Score (Q3: Warsaw - Turn Count)
F1 Score (Q4: BDD - Pedestrian Crossing)
F1 Score (Q5: Tokyo - Stopped in Crosswalk)
F1 Score (Q6: Resort - Joggers)
F1 Score (Q7: Warsaw - Hard Braking)
F1 Score (Q8: Warsaw - Turn on Red)
Overlap
KCF
FlowNet
PP
GNN (our method)
MIRIS (our method)
Higher Speed
Higher Accuracy

Conclusion

- MIRIS is an approach for efficiently executing object track queries on large video datasets
- Provides a 9x average speedup (at the highest accuracy levels)
- Code: https://favyen.com/miris/