EdgeCompression: An Integrated Framework for Compressive Imaging Processing on CAVs

Sidi Lu[1], Xin Yuan[2], Weisong Shi[1]

[1]Wayne State University, [2]Nokia Bell Labs

Introduction: take CAVs as an example

- **Fast high-resolution video processing**
- **Conventional approaches**
  - Reduce frame rates
  - Reduce the frame size

→ losing informative high-speed info and/or small objects in frames

- **Compressive Imaging (CI) camera**

Optical-domain compressed images (measurements)

  - Complicated algorithms to retrieve the desired signal
  - High energy consumption
- **Goal: AI + CI + CAVs**
  - Accelerating accurate video analysis
  - Decreasing energy consumption

Key Concepts

- Vehicle-EdgeServer-Cloud *closed-loop* framework (EdgeCompression)

- Experiments with four public datasets
  - Detection accuracy of measurements generated by the CI camera
    - Close to the accuracy on reconstructed videos
    - Comparable to the true value

→ CI camera is important

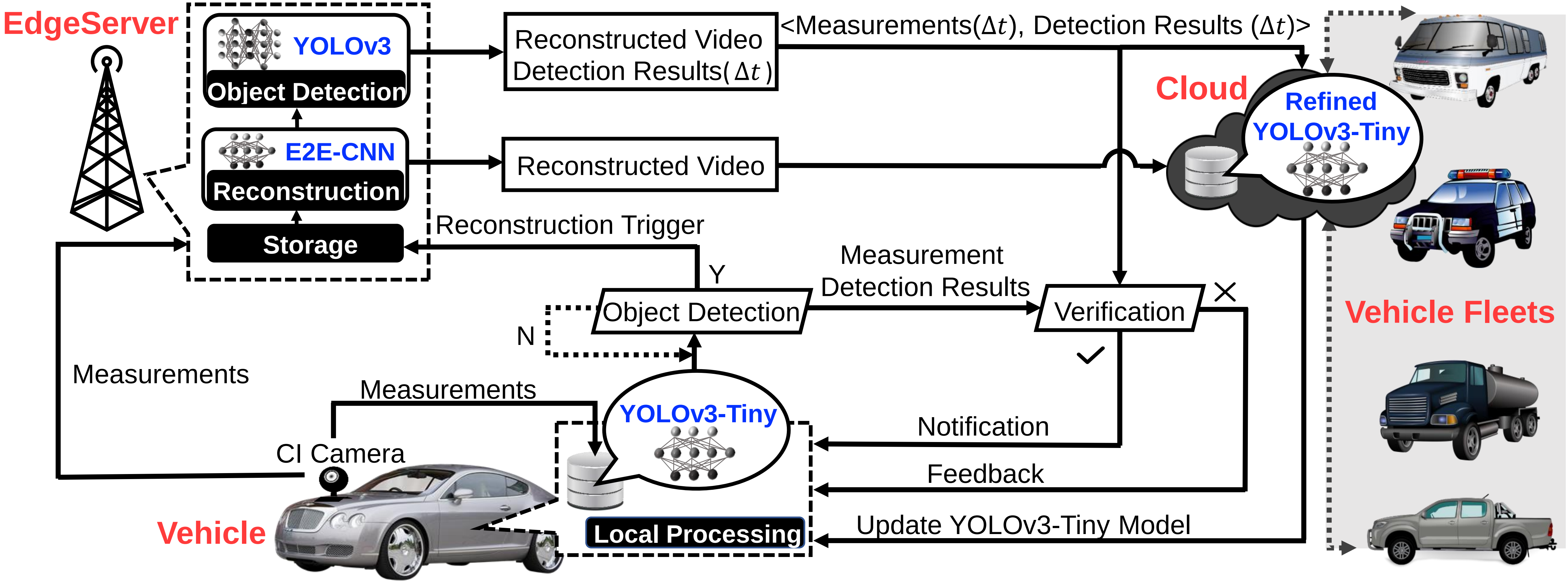
- Framework and methodology can be applied to regular datasets of the traditional camera

The proposed approach is generic

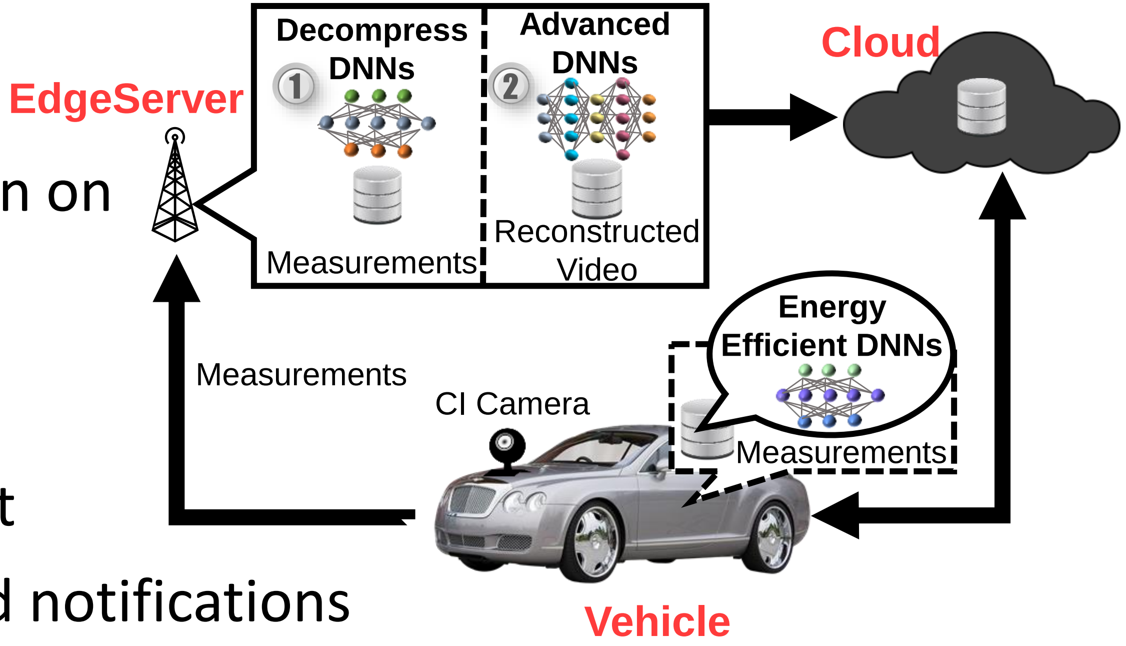
Outline

- EdgeCompression Framework
- Experiment Setup
- Results and Main Observations
- Conclusion

Vehicle-EdgeServer-Cloud Framework



Different Roles in EdgeCompression Framework



1) Vehicles:

- *Energy-efficient network:* make timely computation on measurements.

2) EdgeServer (more computational resources):

- *Reconstruct* high-speed data with a triggered event
- *Verify* the detection results of the Vehicle and send notifications

3) Cloud:

- *Aggregates* all useful information
- *Refine* the energy-efficient network on the Vehicle
- *Big data analysis*: traffic control and path planning

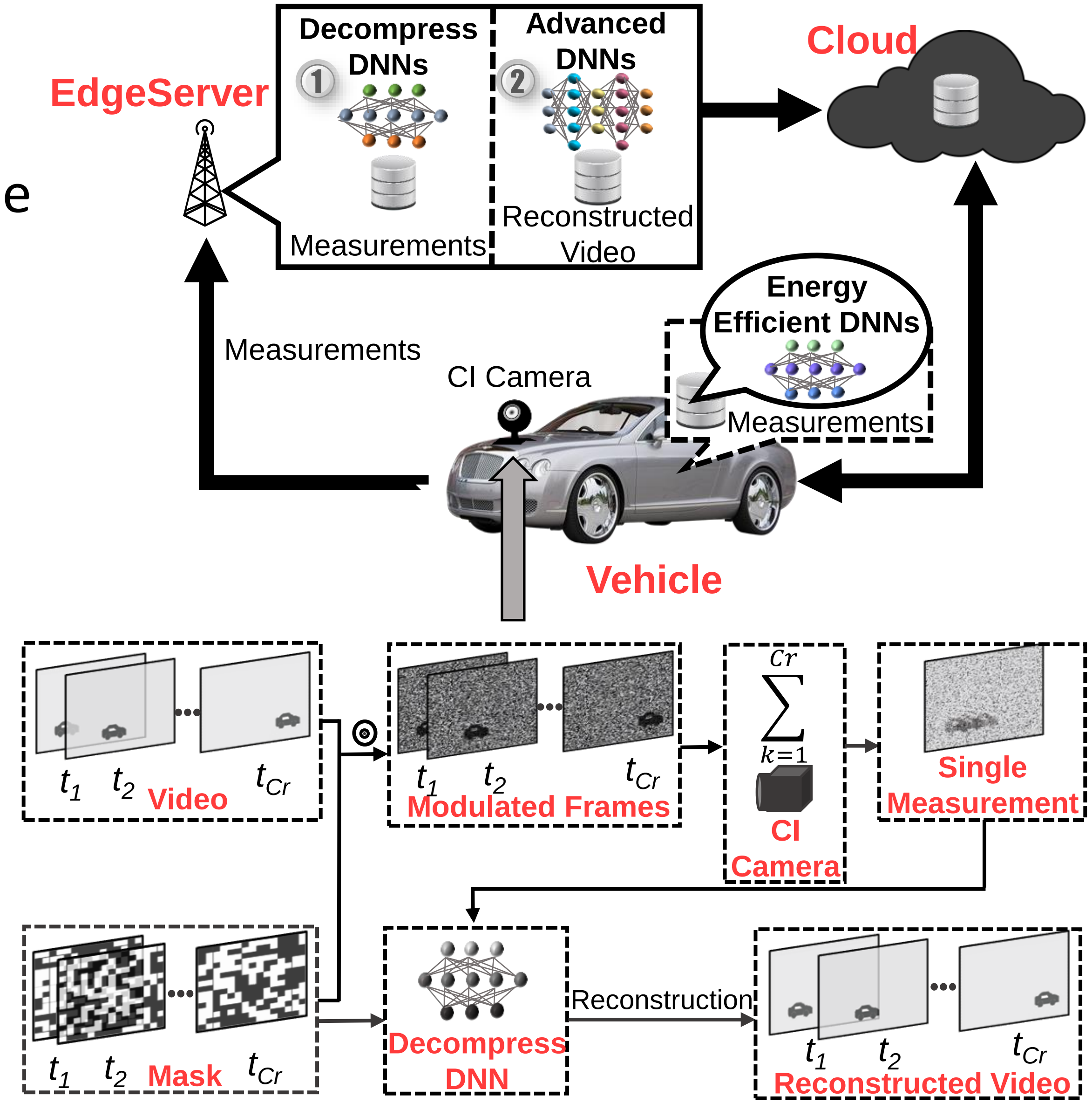
Using video CI as an example

Video Compressive Imaging

1) Video are modulated at a higher speed than the capture rate of the camera
2) Modulated frames are then compressed into a single measurement
3) Multiple frames can be reconstructed from every single measurement.

- $\odot$: element-wise product

  C_r: compression ratio



Framework Advantages

1) Save memory and bandwidth:

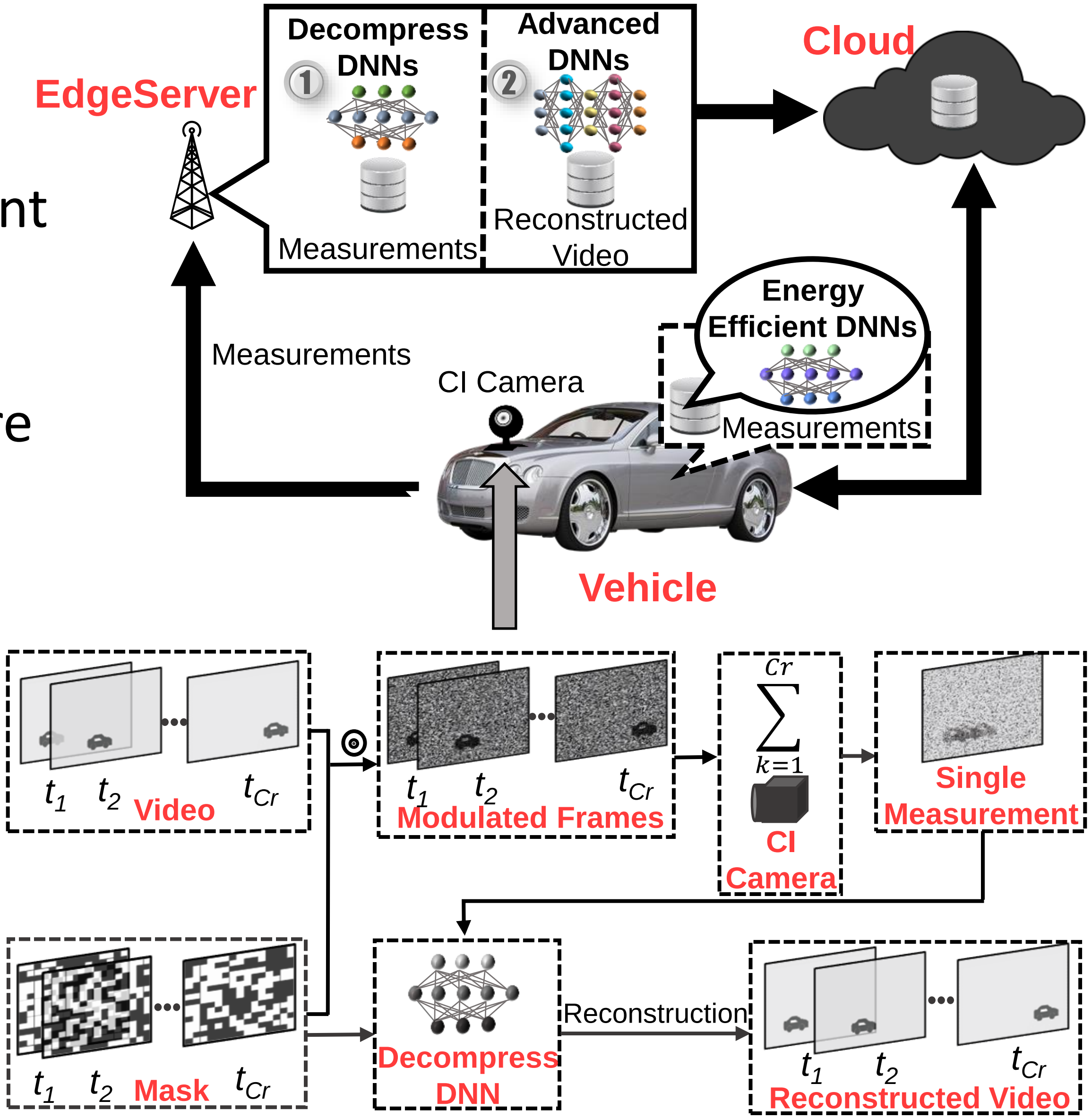
Every C_r frames collapsed to a single measurement

2) Save the computation:

The measurements captured by the CI cameras are already compressed

3) Save the energy:

- *Vehicle*: Detection directly on the measurements w/o reconstruction
- *EdgeServer*: only reconstruct video when the trigger is on



Dataset Selection

(a) AAU RainSnow Dataset

(b) BDD100K Dataset

(c) PDTV Dataset

(d) DynTex Dataset

- https://www.kaggle.com/aalborguniversity/aau-rainsnow
- https://bdd-data.berkeley.edu/
- http://www.tft.lth.se/english/research/video-analysis/co-operation/publicdataset/
- http://dyntex.univ-lr.fr/database.html

Hardware Setup

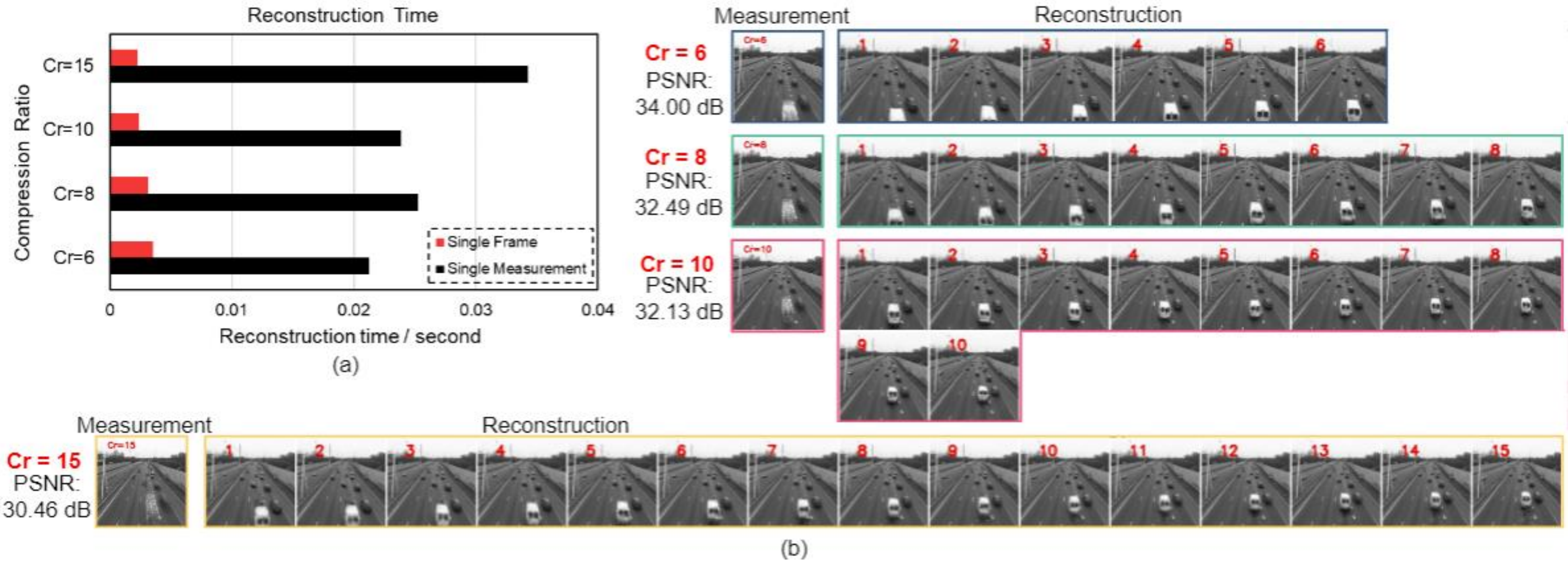
Intel Fog Reference Design (FRD)

NVIDIA GPU Workstation

We assume that a CAV is equipped with an Intel FRD, and the NVIDIA GPU Workstation is working as the EdgeServer.

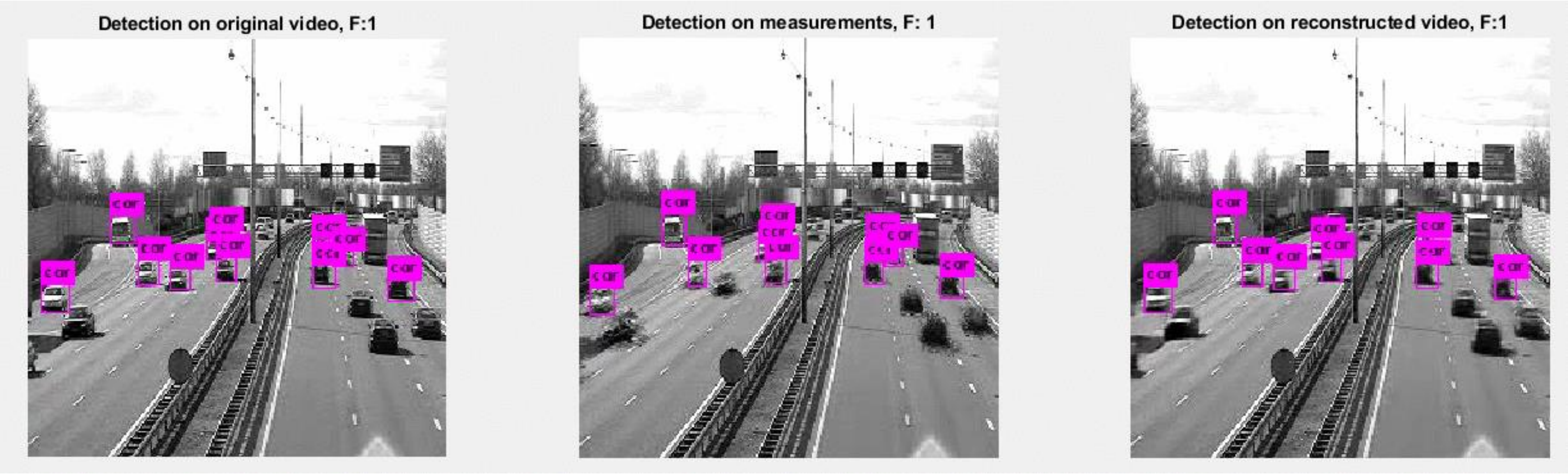
	Intel FRD	NVIDIA GPU Workstation
CPU	Intel Xeon E3-1275 v5	Intel Xeon E5-2690 v4
GPU	NONE	4 × 11 GB GeForce RTX 2080 Ti
Frequency	3.6 GHz	2.6 GHz
Core	4	14
Memory	32 GB	64 GB
OS	Ubuntu 16.04.6 LTS	Ubuntu 16.04.6 LTS

Reconstruction Results



PSNR: signal-to-noise ratio between two images

Detection on the Measurements Directly

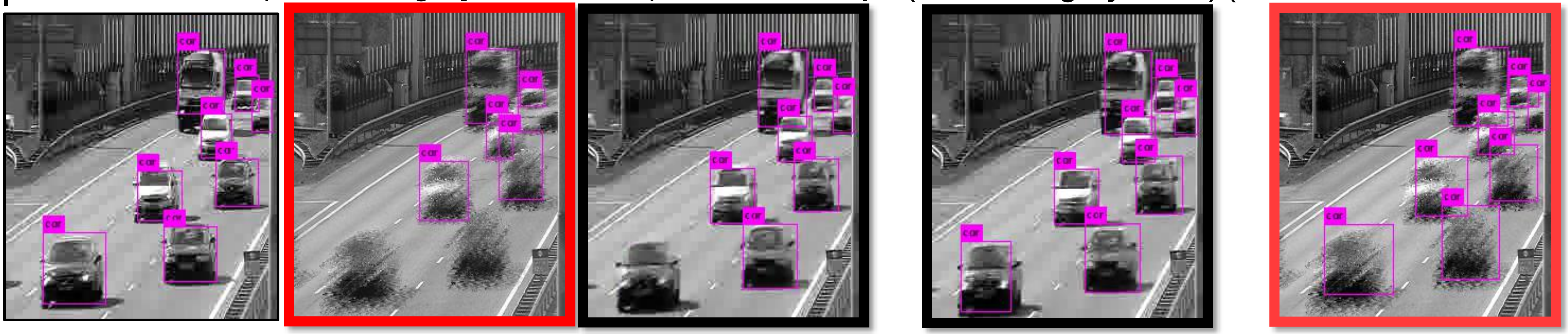


Detection Results

YOLOv3-Tiny
(trained on **grey-scale video**)

YOLOv3
(trained on **grey-scale**)

YOLOv3-Tiny
(trained on **measurements**)



Grey-Scale Video (a)

Measurement (b)

Reconstructed Video (c)

Reconstructed Video (d)

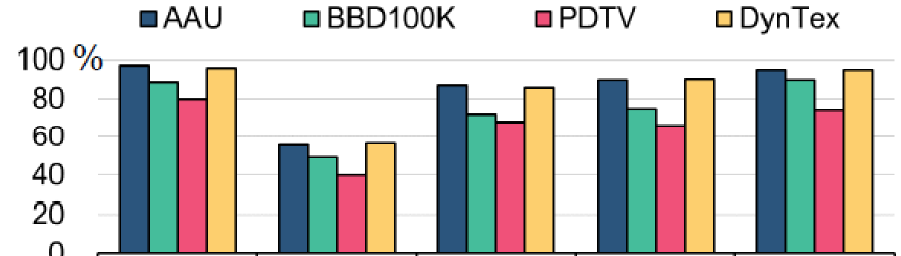
Measurement (e)

Upgrade model

Change training dataset of the model

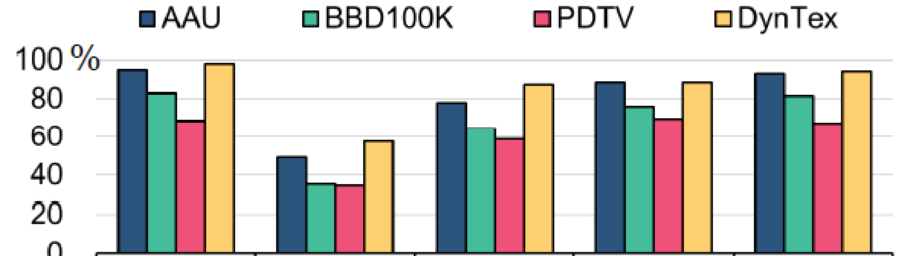
Experiment results in terms of mAP

Cr = 6



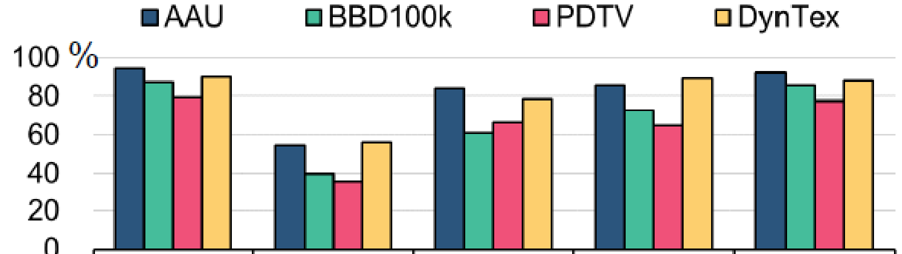
	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	97.14	55.33	87.02	89.71	95.16
BBD100K	88.49	49.25	71.71	74.61	89.63
PDTV	79.91	39.83	67.52	66.17	74.16
DynTex	96.13	56.15	85.78	90.03	95.23

Cr = 8



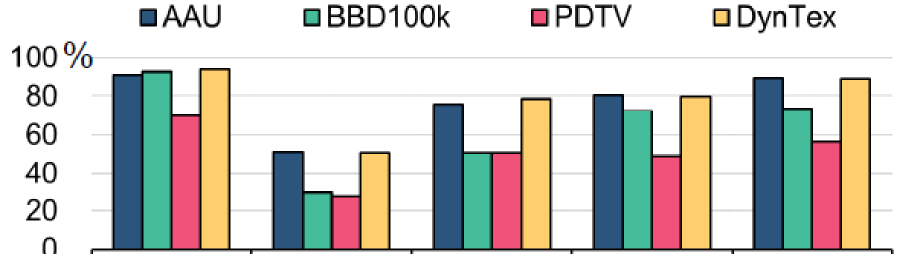
	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	95.11	49.35	77.80	88.60	93.17
BBD100K	82.92	35.18	64.22	75.67	81.43
PDTV	68.56	34.67	58.54	69.32	67.16
DynTex	98.12	57.31	87.39	88.62	94.17

Cr = 10



	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	94.20	54.58	83.95	85.64	92.28
BBD100k	87.49	39.87	61.01	72.48	85.71
PDTV	79.00	35.82	66.45	64.60	77.39
DynTex	90.10	56.12	78.45	89.40	88.26

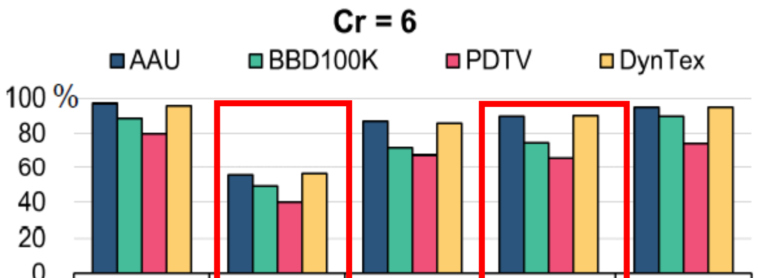
Cr = 15



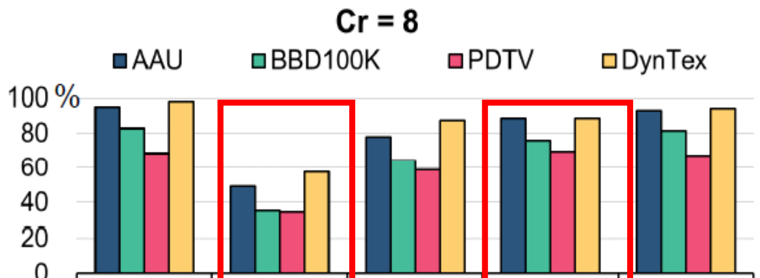
	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	91.18	50.85	75.76	80.54	89.32
BBD100k	92.47	29.71	50.30	72.07	73.30
PDTV	69.97	27.73	50.63	48.84	56.79
DynTex	93.88	50.33	78.22	79.34	89.03

- G: grey-scale video
- M: measurements
- R: reconstructed video.
- mAP refers to the mean Average Precision.

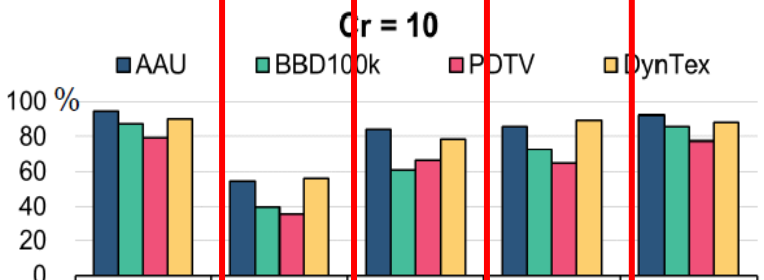
Observation #1: TinyG-M *vs.* TinyM-M



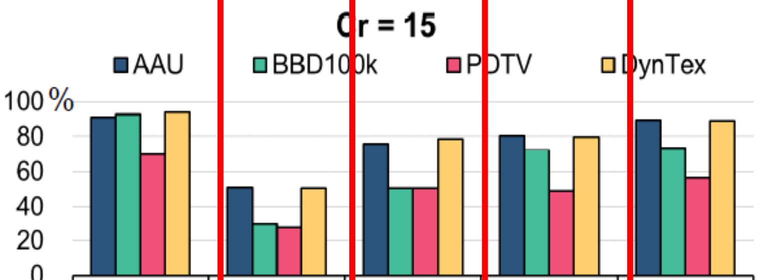
	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	97.14	55.33	87.02	89.71	95.16
BBD100K	88.49	49.25	71.71	74.61	89.63
PDTV	79.91	39.83	67.52	66.17	74.16
DynTex	96.13	56.15	85.78	90.03	95.23



	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	95.11	49.35	77.80	88.60	93.17
BBD100K	82.92	35.18	64.22	75.67	81.43
PDTV	68.56	34.67	58.54	69.32	67.16
DynTex	98.12	57.31	87.39	88.62	94.17



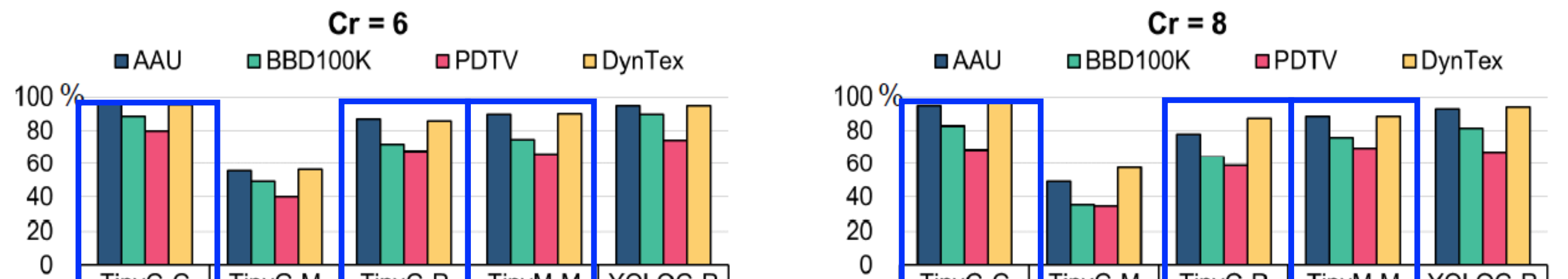
	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	94.20	54.58	83.95	85.64	92.28
BBD100k	87.49	39.87	61.01	72.48	85.71
PDTV	79.00	35.82	66.45	64.60	77.39
DynTex	90.10	56.12	78.45	89.40	88.26



	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	91.18	50.85	75.76	80.54	89.32
BBD100k	92.47	29.71	50.30	72.07	73.30
PDTV	69.97	27.73	50.63	48.84	56.79
DynTex	93.88	50.33	78.22	79.34	89.03

- The mAP score of TinyM-M is significantly larger (almost double) than that of TinyG-M.
- Training the model specifically on the raw measurement is necessary for the accurate detection.

Observation #2: TinyM-M, TinyG-R, and TinyG-G



Cr = 6

	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	97.14	55.33	87.02	89.71	95.16
BBD100K	88.49	49.25	71.71	74.61	89.63
PDTV	79.91	39.83	67.52	66.17	74.16
DynTex	96.13	56.15	85.78	90.03	95.23

Cr = 8

	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	95.11	49.35	77.80	88.60	93.17
BBD100K	82.92	35.18	64.22	75.67	81.43
PDTV	68.56	34.67	58.54	69.32	67.16
DynTex	98.12	57.31	87.39	88.62	94.17

Cr = 10

	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	94.20	54.58	83.95	85.64	92.28
BBD100k	87.49	39.87	61.01	72.48	85.71
PDTV	79.00	35.82	66.45	64.60	77.39
DynTex	90.10	56.12	78.45	89.40	88.26

Cr = 15

	TinyG-G	TinyG-M	TinyG-R	TinyM-M	YOLOG-R
AAU	91.18	50.85	75.76	80.54	89.32
BBD100k	92.47	29.71	50.30	72.07	73.30
PDTV	69.97	27.73	50.63	48.84	56.79
DynTex	93.88	50.33	78.22	79.34	89.03

Vehicle detection results from the measurements (TinyM-M) achieve:

1) Comparable detection results to the reconstructed video (TinyG-R)
2) Close to the detection results from the ground truth video (TinyG-G) across all compression ratios.

Do not need to reconstruct the high-quality data in real-time, and we can still use CI cameras for real applications in CAVs

Conclusion

- Vehicle-EdgeServer-Cloud *closed-loop* framework
- This is the first work that provides an alternative method to achieve fast object detection *on measurements.*
- Our code is hosted at https://www.thecarlab.org/outcomes/software

Q & A

lu.sidi@wayne.edu

https://www.thecarlab.org/