Accelerating Multi-Object Tracking in Edge Computing Environment with Time-Spatial Optimization

CBD 2021

Mengyang Liu[1,2] Anran Tang[1] Huitian Wang[1] Lin Shen[1] Yunhan Chang[1] Guangxing Cai[1]

Daheng Yin[1] Fang Dong[1] Wei Zhao[3,4]

[1]Southeast University
[2]Jiangsu University
[3]Comprehensive National Science Center
[4]Anhui University of Technology

Our paper: *Our code:*

Outline

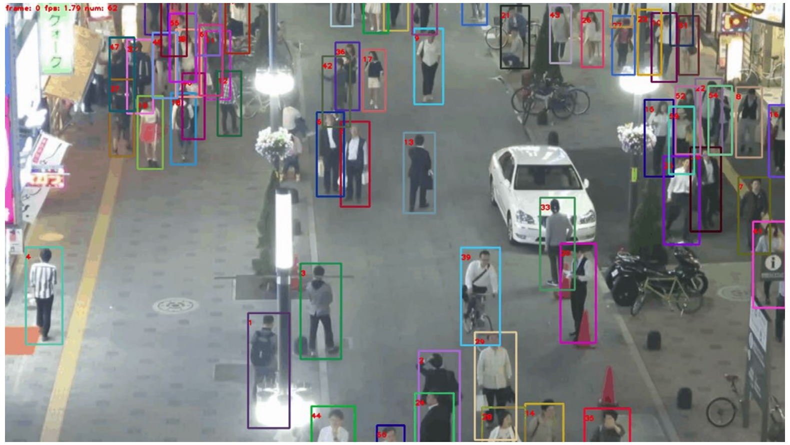
- Introduction
  - Multi-Object Tracking
  - Performance Issue of Deep Learning
  - Time-Spatial Processing Redundancy
- Methodology
  - System Overview
  - Video Frames Classification
  - Region Dividing
  - Image Partitioning and Scheduling
- Experiments
  - Progressive Results
- Future Work

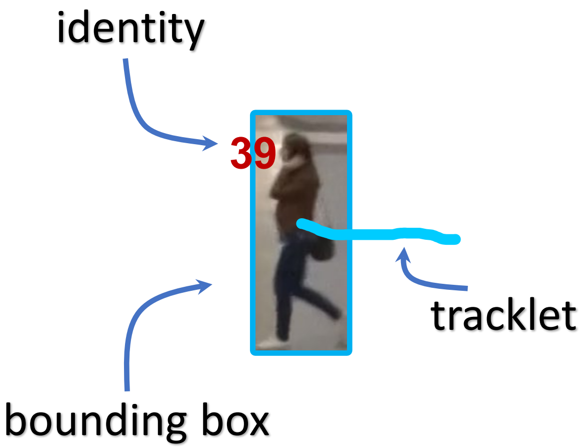
Outline

- Introduction
  - Multi-Object Tracking
  - Performance Issue of Deep Learning
  - Time-Spatial Processing Redundancy
- Methodology
  - System Overview
  - Video Frames Classification
  - Region Dividing
  - Image Partitioning and Scheduling
- Experiments
  - Progressive Results
- Future Work

Introduction – *Multi-Object Tracking*

What is multi-object tracking?





Applications:

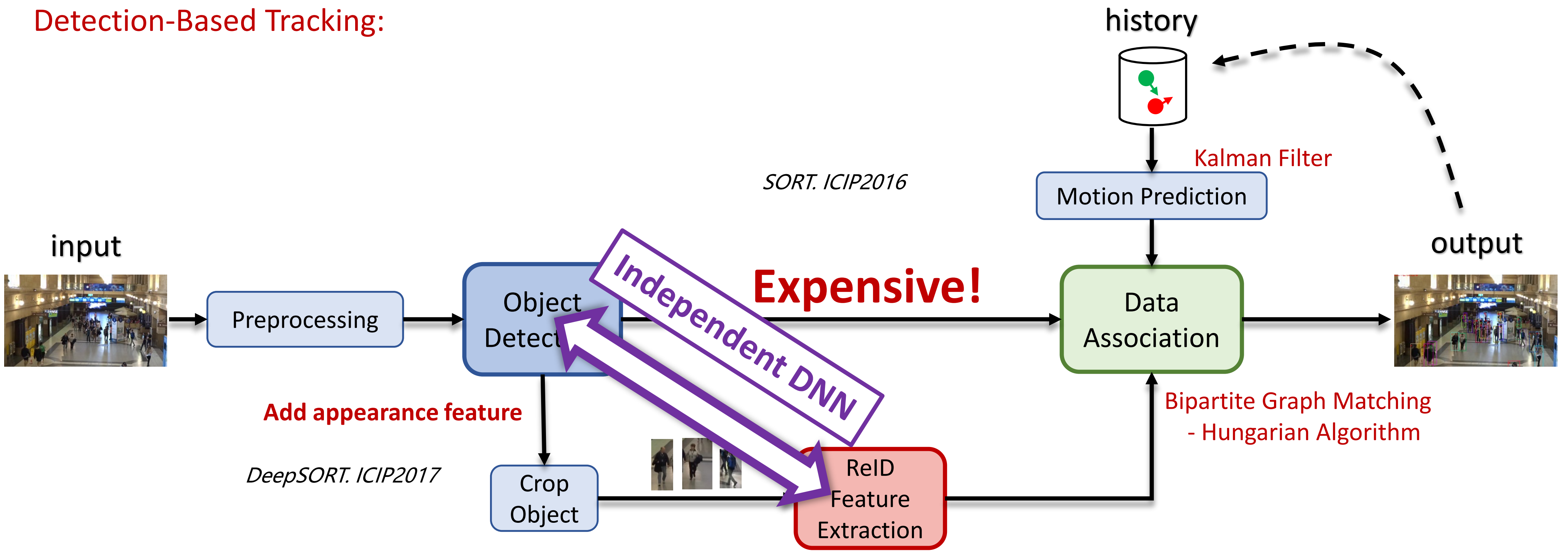
video surveillance

augmented/virtual reality

autonomous driving

Introduction – *Multi-Object Tracking*

DL-based MOT



Introduction – *Multi-Object Tracking*

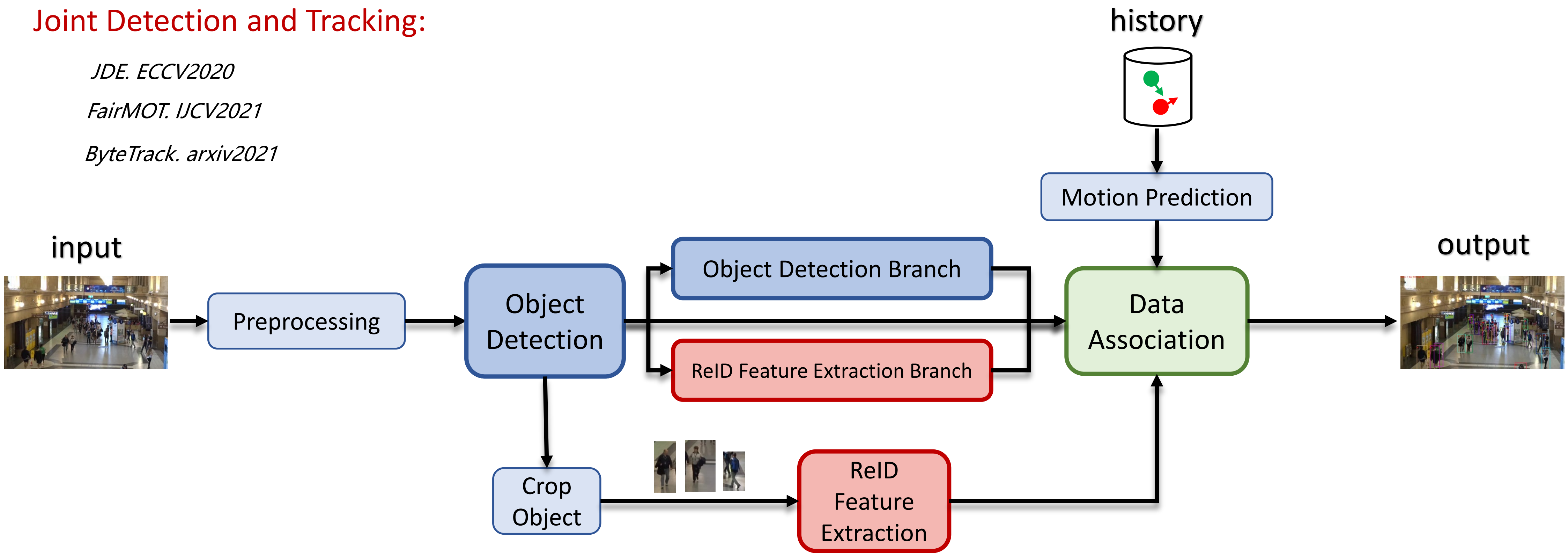
DL-based MOT

Joint Detection and Tracking:

JDE. ECCV2020

FairMOT. IJCV2021

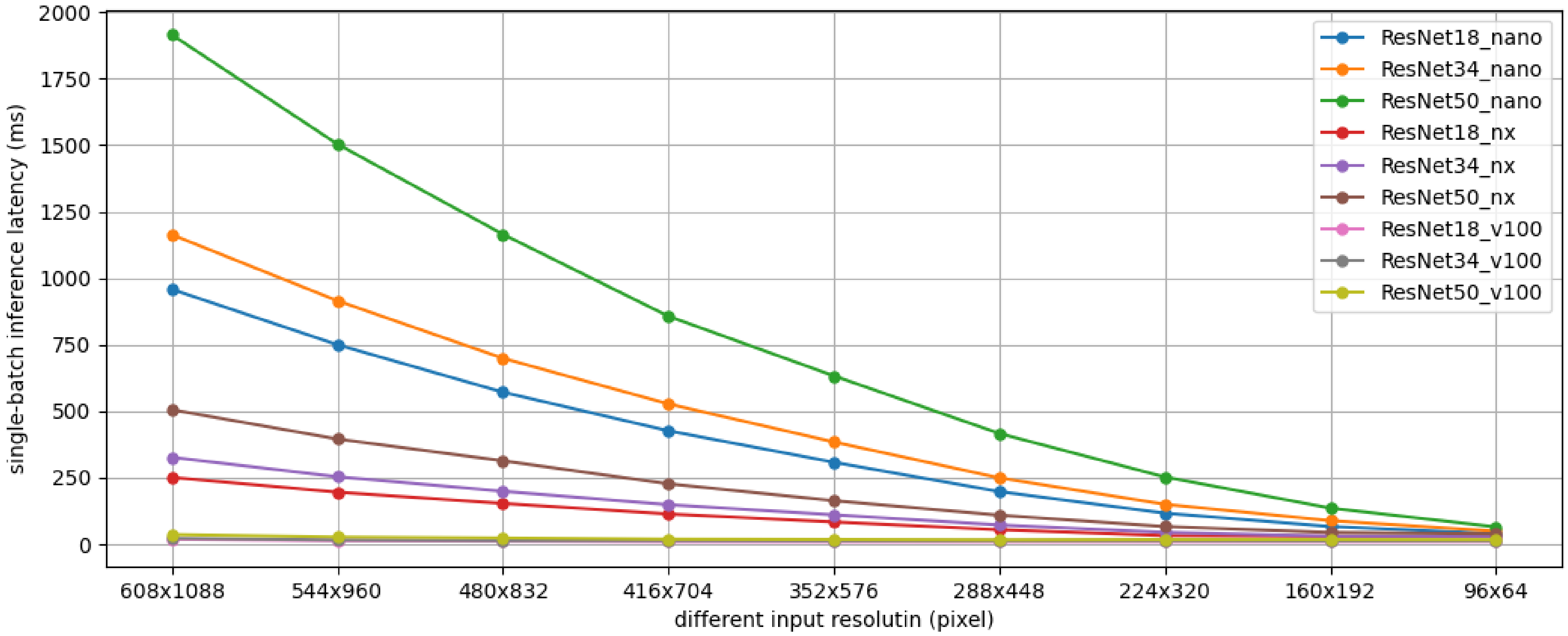
ByteTrack. arxiv2021

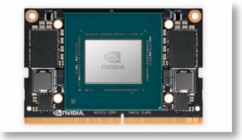


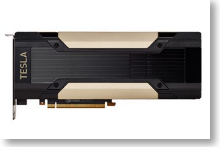
Introduction – *Performance Issue of Deep Learning*

Cost-performance paradox

High input resolution and computational NN can bring good accuracy.
But for guaranteeing the latency requirement, expensive computing platform is needed.



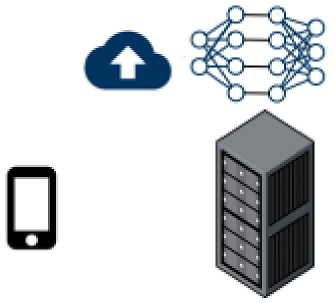




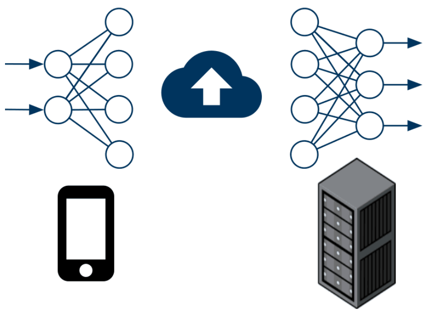
Introduction – *Performance Issue of Deep Learning*

Edge Computing

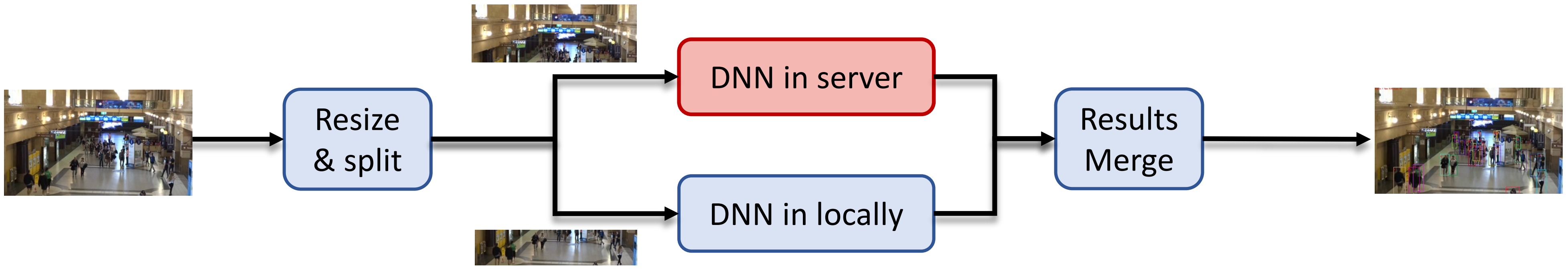
Integral offloading



Vertical split



Horizontal split



Introduction – *Time-Spatial Processing Redundancy*

Coarse-grained processing in a standard DNN pipeline

Redundancy!

All frames

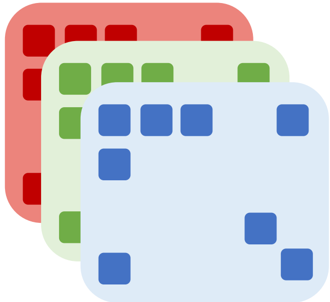
Static Pipeline

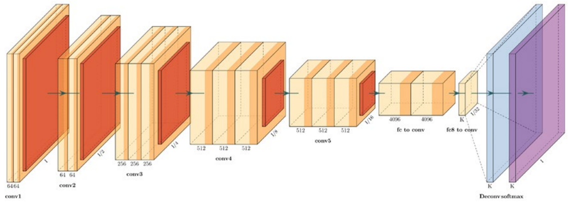
resize

Fixed size input

608 x 1088 x 3

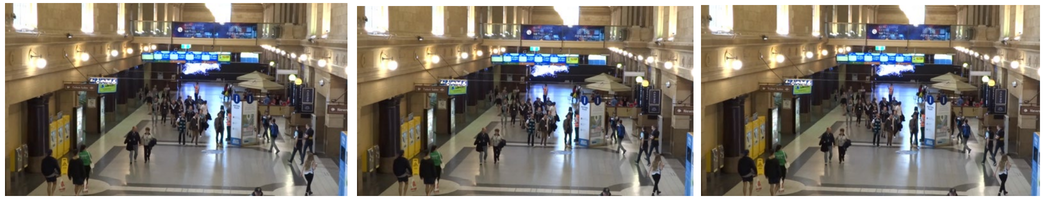
Tensor





Introduction — *Time-Spatial Processing Redundancy*

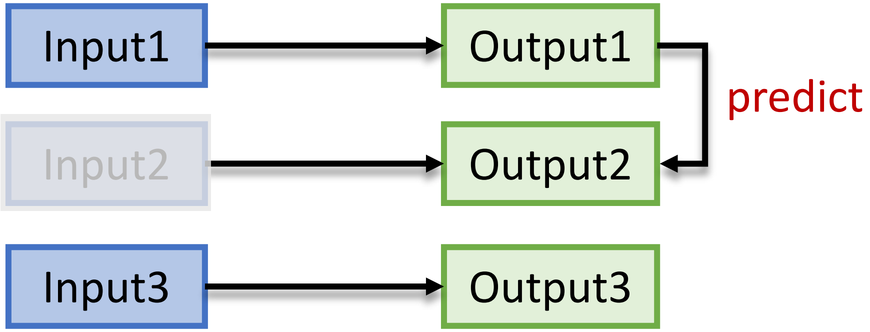
Time redundancy



similar moving state

- ☐ Fixed camera
- ☐ Low velocity moving objects
- ☐ Static environment

😃**We can only select few frames into DNN periodically.**



Introduction – *Time-Spatial Processing Redundancy*

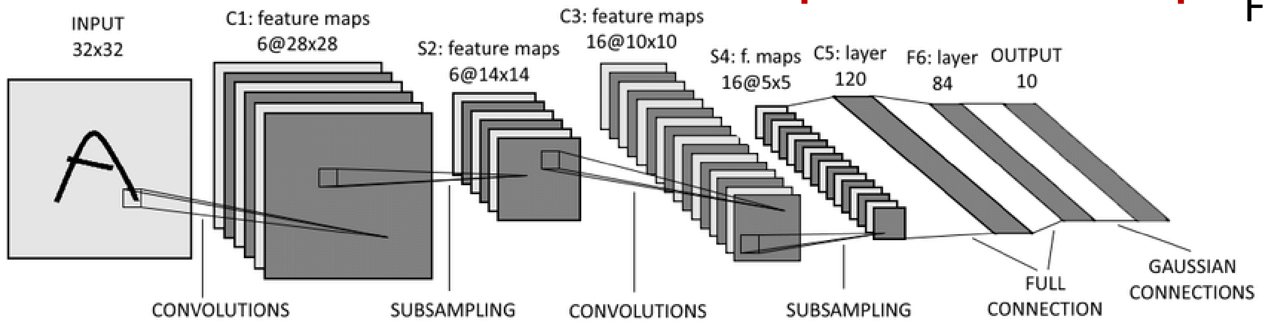
Spatial redundancy

- Background
- Low velocity moving objects
- Sparse objects

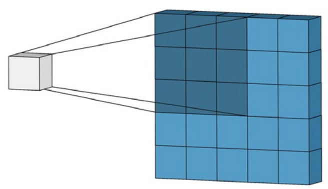
😃We can crop the image with valuable content and modify the network input size.

Modulation constraint:

Fixed input of FC layers



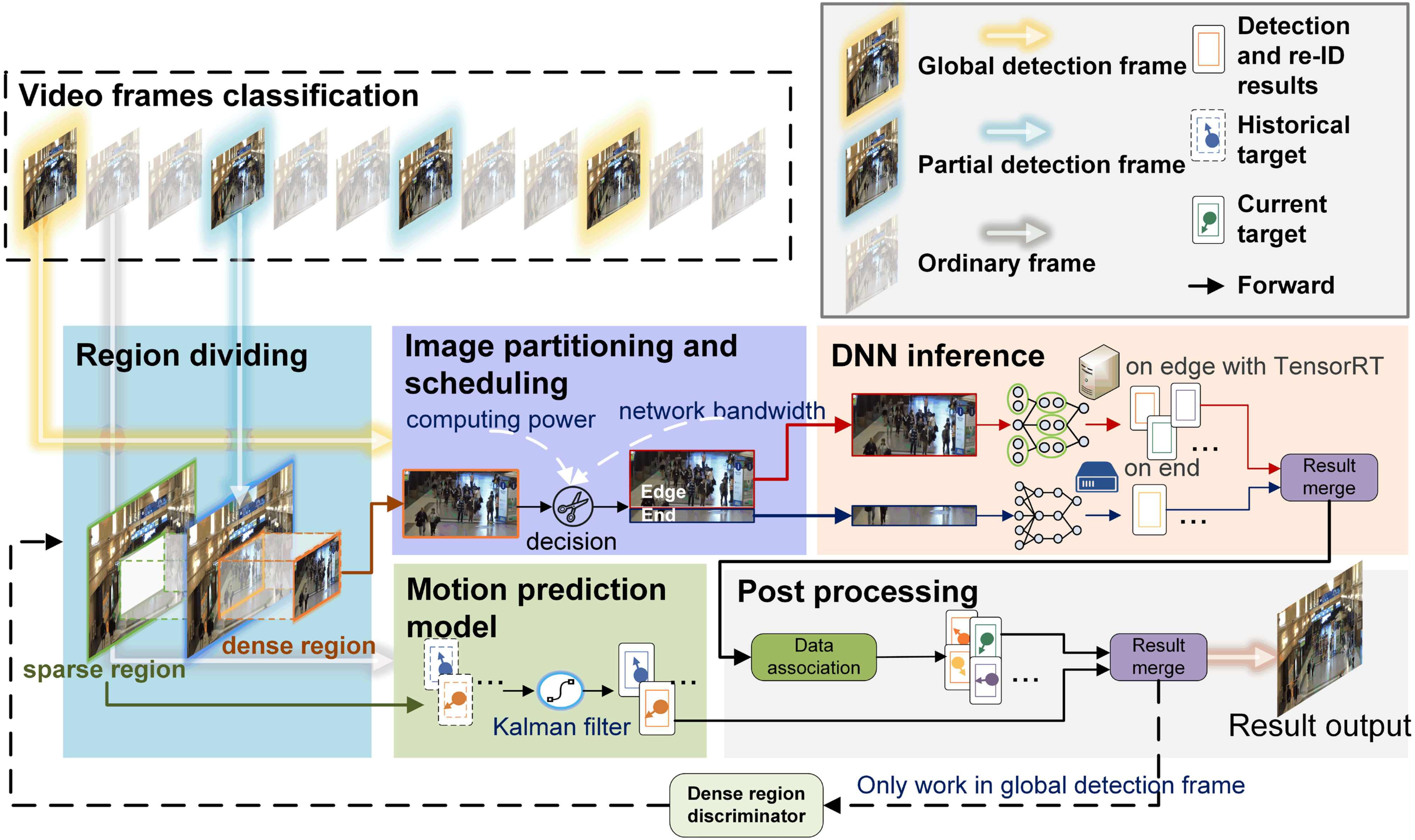
Fully convolution network don't have this fixed boundary.



Outline

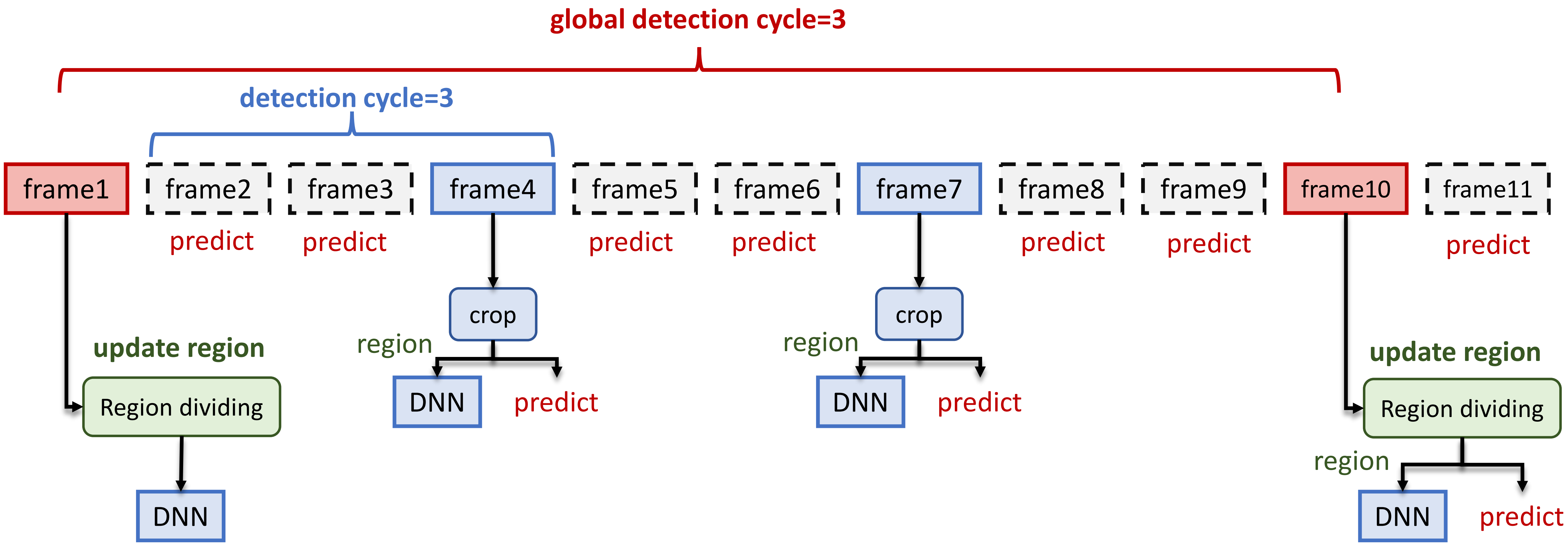
- Introduction
    - Multi-Object Tracking
    - Performance Issue of Deep Learning
    - Time-Spatial Processing Redundancy
- Methodology
    - System Overview
    - Video Frames Classification
    - Region Dividing
    - Image Partitioning and Scheduling
- Experiments
    - Progressive Results
- Future Work

Methodology – *System Overview*



Methodology – *Video Frames Classification*

Video frame classification



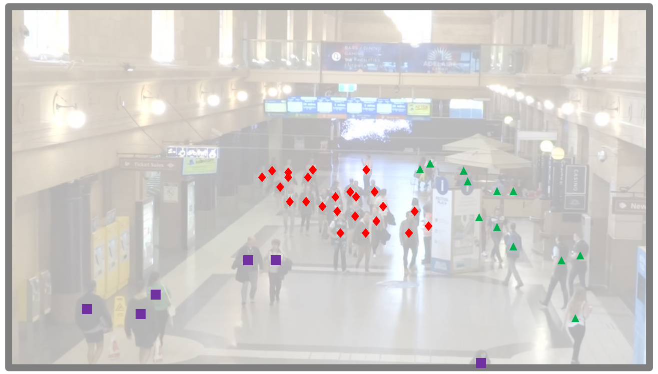
Three types of frames: ordinary, partial detection and global detection.

Methodology – *Region Dividing*

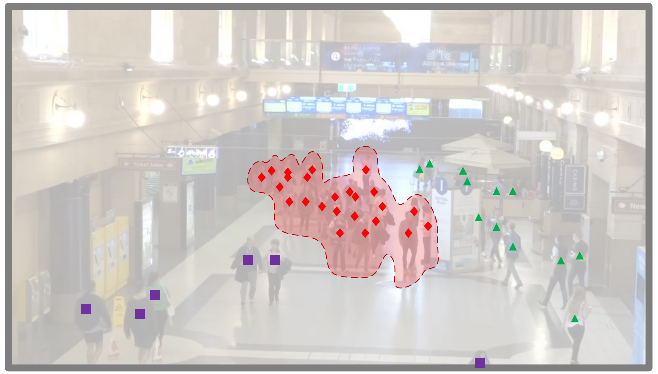
Region dividing

DBSCAN clustering based region dividing algorithm:

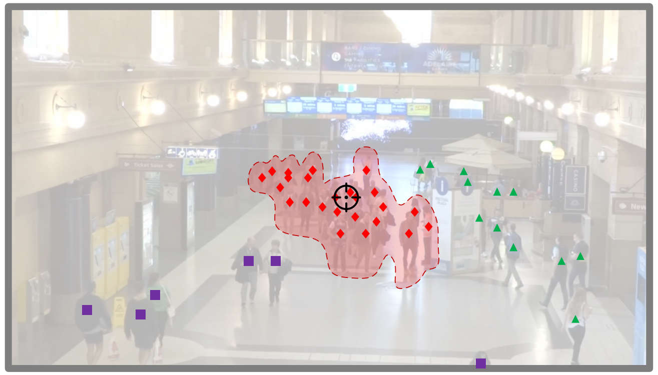
a) Detect



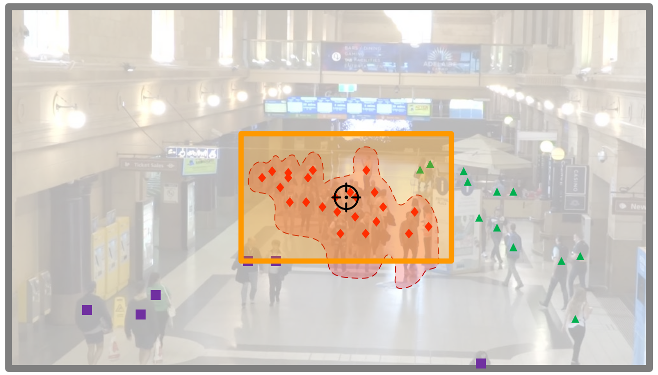
b) Cluster



c) Select top1 cluster



d) Calculate center point



e) Generate dense region box

object point

point in cluster1

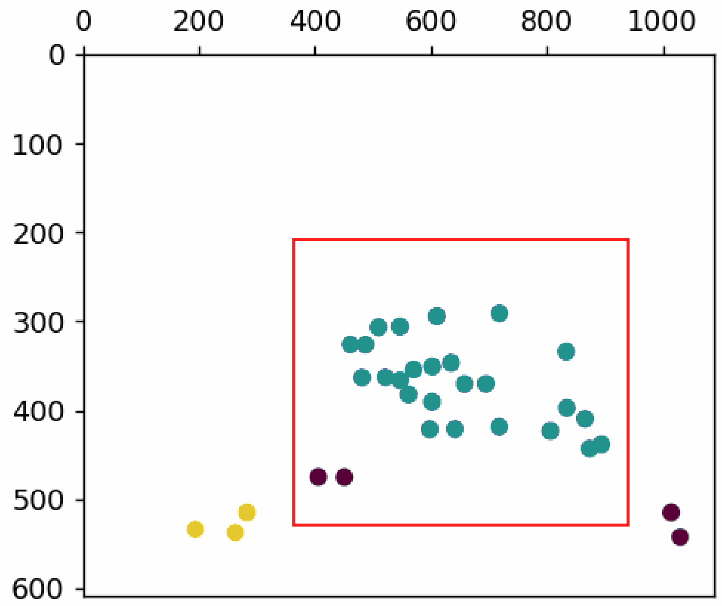
point in cluster2

outlier

Methodology – *Region Dividing*

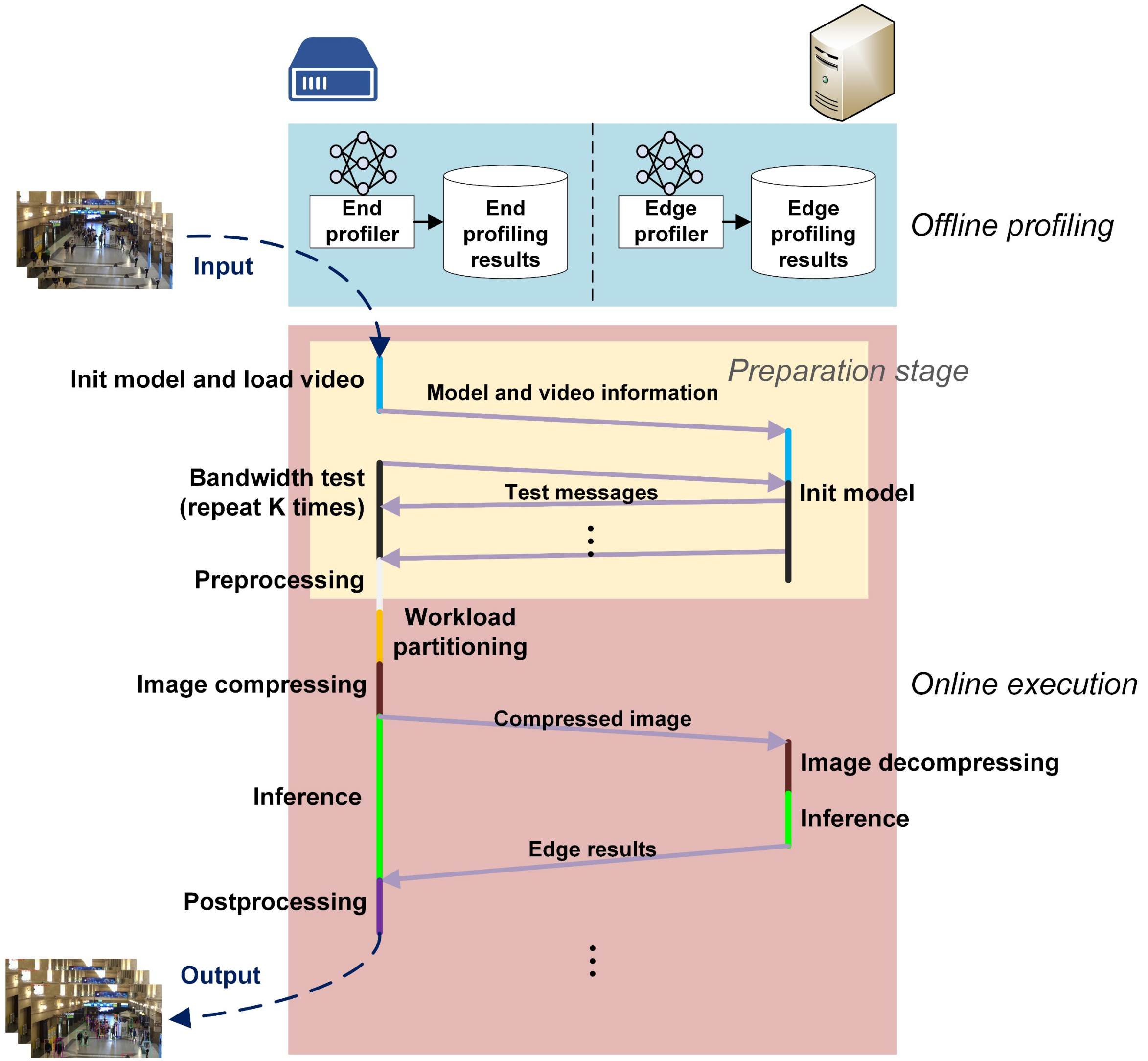
Region dividing

Show



Methodology – *Image Partitioning and Scheduling*

End-edge collaboration



workload ratio between the end and edge

$$
\begin{aligned}
\underset{pr}{\text{minimize}} \quad & \max[L_{send}(r_{end}) + P_{edge}(r_{edge}), \\
& L_{send}(r_{end}) + P_{end}(r_{end})] \\
s.t. \quad & r_{end} = r_{origin} \cdot pr \\
& r_{edge} = r_{origin} \cdot (1 - pr) \\
& B = \frac{1}{K} \sum_{j=i-K}^{i-1} B_j \\
& L_{send} = \frac{1}{p} \cdot r_{end} \cdot \frac{1}{B} \\
& i > K
\end{aligned}
$$

Outline

- Introduction
  - Multi-Object Tracking
  - Performance Issue of Deep Learning
  - Time-Spatial Processing Redundancy
- Methodology
  - System Overview
  - Video Frames Classification
  - Region Dividing
  - Image Partitioning and Scheduling
- Experiments
  - Progressive Results
- Future Work

Experiments – *Progressive Results*

Setup

Hardware

Edge

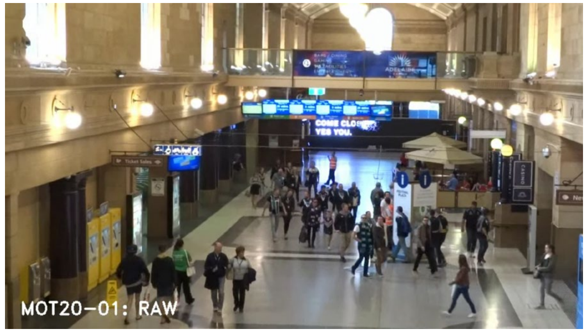
CPU	Intel Core i5-7300HQ
Memory	8GB
GPU	NVIDIA GTX 1050

End - NVIDIA Jetson Nano

CPU	Quad-core ARM A57
Memory	4GB
GPU	Maxwell 128 CUDA core

Dataset

MOT20-01



Metric

Accuracy: MOTA
Speed: FPS

Baseline method

FairMOT

Software Framework

PyTorch

TensorRT

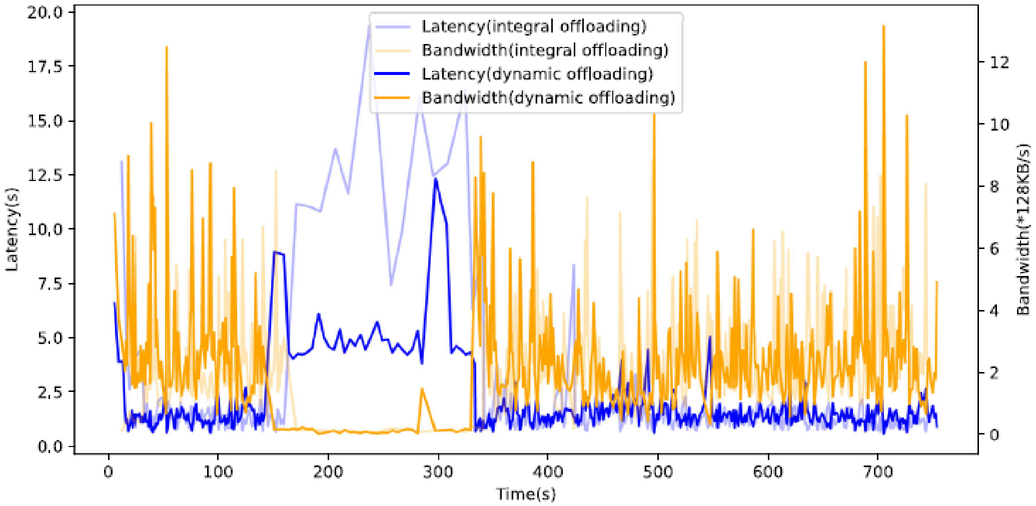
Experiments – *Progressive Results*

Performance

COMPARISON OF DIFFERENT ACCELERATION COMBINATIONS.

	fairmot-origin									after acceleration		
end-only	√	√	√	√								
time-spatial strategy		√	√	√		√	√	√		√	√	√
end-edge collaboration					√	√	√	√	√	√	√	√
TensorRT									√	√	√	√
global detection cycle		1	2	3		1	2	3		1	2	3
detection cycle		1	2	3		1	2	3		1	2	3
MOTA ↑	**61.5**	57	52.7	48.8	61	58.8	54.7	51.4	60.9	58.5	54.4	51.1
FPS ↑	0.26	0.66	1.03	1.37	1.15	3.03	4.68	7.09	1.77	4.58	7.7	**9.91**

Adaptability to Network Fluctuations



The system can achieve 17.6~38.1x speedup ratio, while with 3%~10.4% absolute tracking accuracy sacrifice.

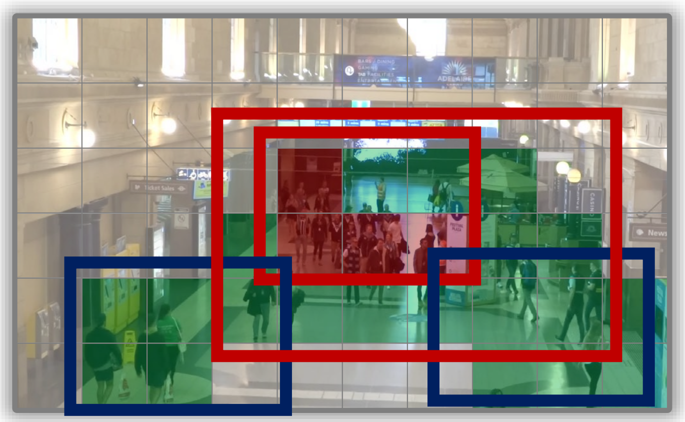
- Standard bandwidth: 1MBs
- Limited bandwidth: 50KBs

Adaptability achieved can save 2.5~4.5x of the overall processing delay.

Outline

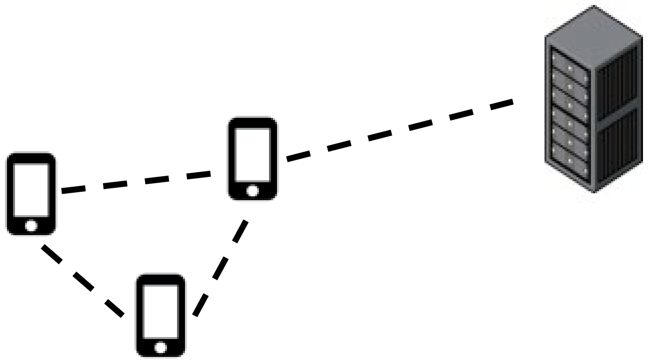
- Introduction
  - Multi-Object Tracking
  - Performance Issue of Deep Learning
  - Time-Spatial Processing Redundancy
- Methodology
  - System Overview
  - Video Frames Classification
  - Region Dividing
  - Image Partitioning and Scheduling
- Experiments
  - Progressive Results
- Future Work

Future Work



From single region box to multi region boxes.

More opportunities in edge and distributed computing.



Thank you!

myliu@seu.edu.cn