DiVE: Differential Video Encoding for Online Edge-assisted Video Analytics on Mobile Agents

Jiangang Shen[1], Hongzi Zhu[1], Liang Zhang[2], Yunzhe Li[1], Shan Chang[2], Jie Wu[3], and Minyi Guo[1]

[1]Shanghai Jiao Tong University, China
[2]Donghua University, China
[3]Cloud Computing Research Institute China Telecom, China

Background

- Offloading video analytic task to edge server for mobile agents can be of great interest.

Mobile Agents are resource-constrained.

Online video analytics is important.

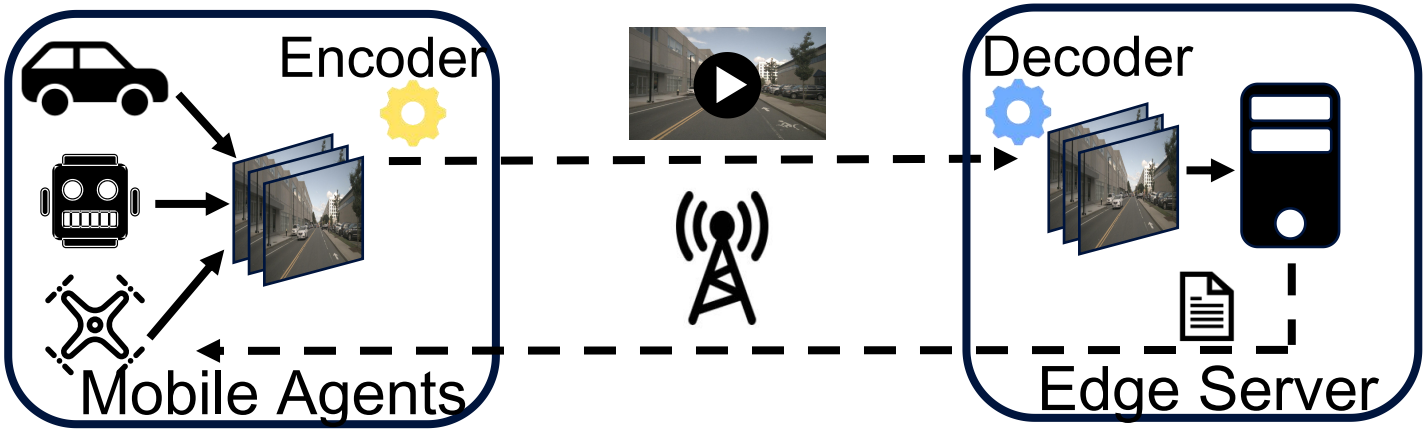
Existing work

- Key frame filtering at camera side.
  - Perform poorly in highly dynamic scenarios!
- Video compressing based on server feedback.
  - Depend on guidance from the edge, leading to huge latency!
- Local compressed model assisted video analytics.
  - Require significant computing power!

System Model & Design Goals

System Model

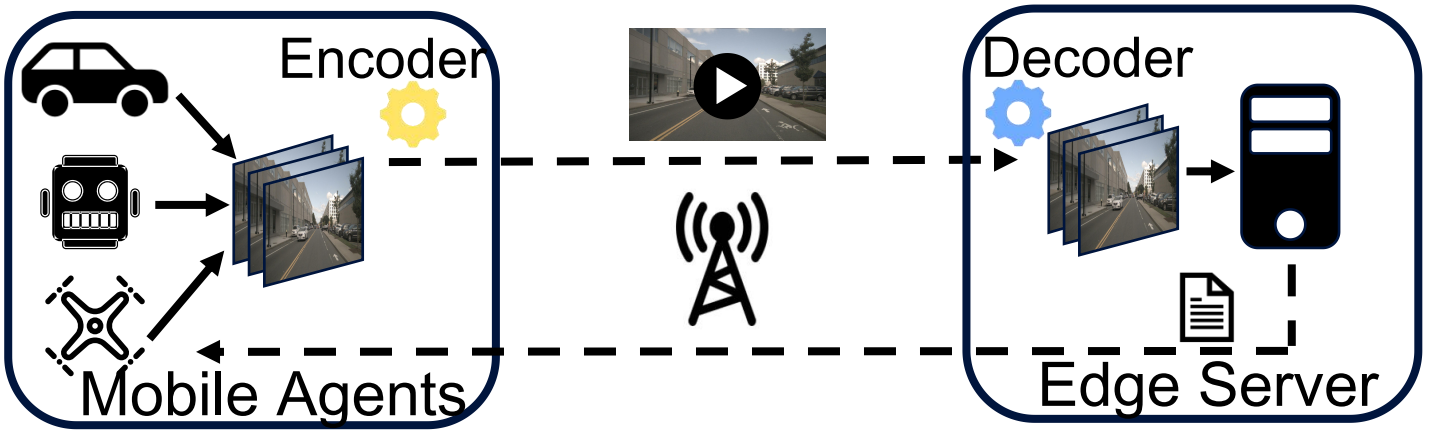
- Mobile Agents:
  - Demanding video analytics like object detection.
  - Basic computility for video encoding such as H.264.
- Edge Server:
  - Computility for video decoding and online model inference.
  - Returning inference results to mobile agents.



System Model & Design Goals

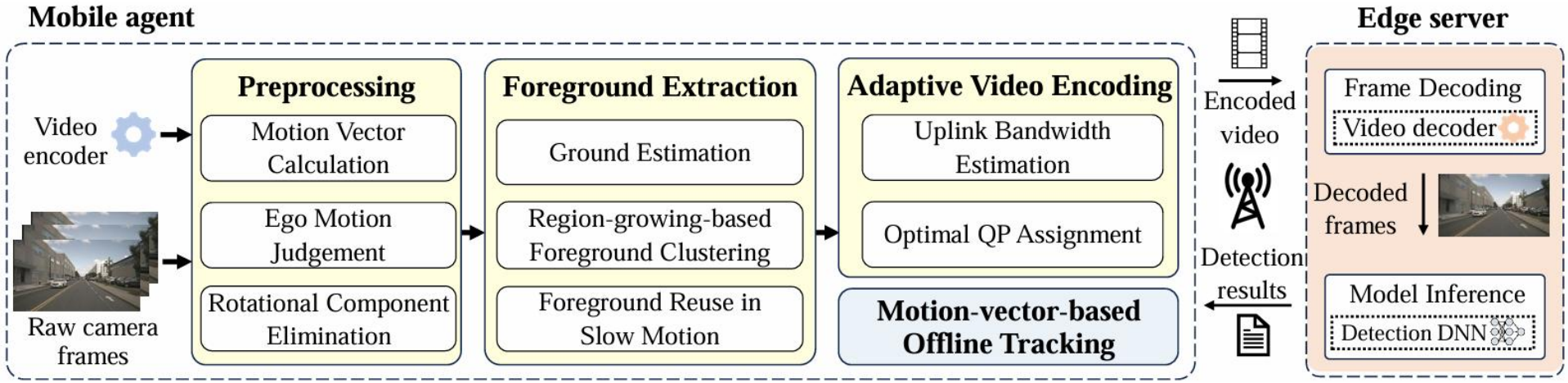
Design Goals

- High Inference Accuracy.
- Low End-to-end Latency.
- Limited computational power Requirement.



Core Idea of DiVE

- Mobile agent leverages motion vectors output by the video codec to identify foreground objects in each frame.
- Mobile agent adaptively chooses different compression ratios between the foreground and the rest of a frame based on current uplink bandwidth.



Arhcitecture of DiVE

Challenges

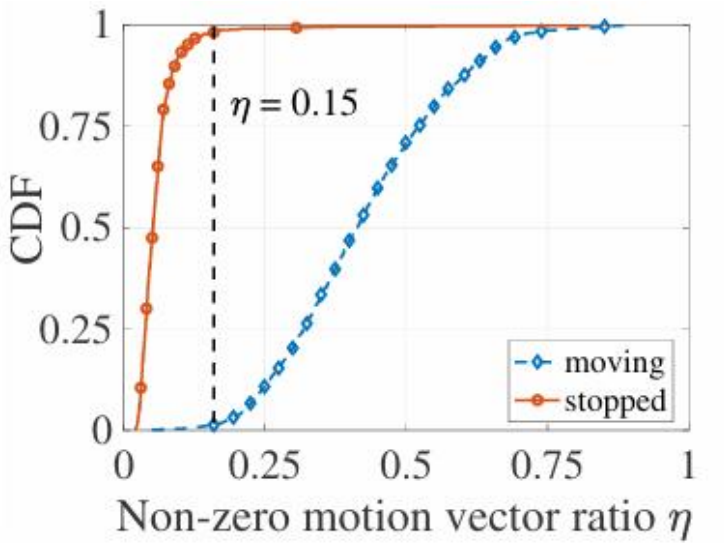
Challenge 1：Motion vectors are very coarse and vulnerable to image noise and rotation of the camera.

Challenge 2：Foreground objects may have distinct motion and different image textures.

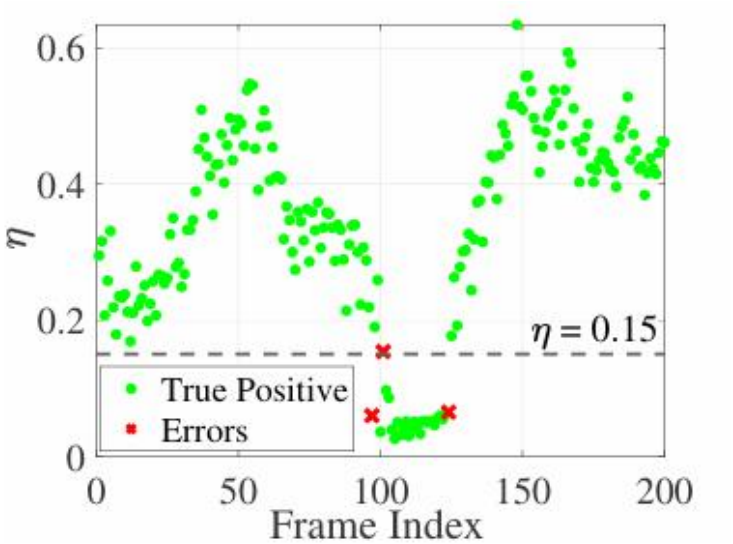
Challenge 1: Process coarse motion vectors

Ego Motion Judgement

- Judge ego motion state with non-zero motion vector ratio η.
- Ego agent is moving if η > threshold, otherwise ego agent is static.



(a) CDFs of η for stopped ego vehicles and moving ego vehicles



(b) Motion estimation on an example driving video clip

Challenge 1: Process coarse motion vectors

Rotational Component Elimination

- When ego agent is turning, the motion vector along x-axis and y-axis can be written as:

$$vx_q = \frac{\Delta Z x_q^t}{Z_Q} - \Delta\phi_y f + \frac{\Delta\phi_x x_q^t y_q^t}{f} - \frac{\Delta\phi_y {x_q^t}^2}{f},$$

$$vy_q = \frac{\Delta Z y_q^t}{Z_Q} + \Delta\phi_x f - \frac{\Delta\phi_y x_q^t y_q^t}{f} + \frac{\Delta\phi_x {y_q^t}^2}{f}.$$

Challenge 1: Process coarse motion vectors

Rotational Component Elimination

- When ego agent is turning, the motion vector along x-axis and y-axis can be written as:

$$vx_q = \frac{\Delta Z x_q^t}{Z_Q} - \Delta\phi_y f + \frac{\Delta\phi_x x_q^t y_q^t}{f} - \frac{\Delta\phi_y x_q^{t\,2}}{f},$$

$$vy_q = \frac{\Delta Z y_q^t}{Z_Q} + \Delta\phi_x f - \frac{\Delta\phi_y x_q^t y_q^t}{f} + \frac{\Delta\phi_x y_q^{t\,2}}{f}.$$

translation components

rotation components

Challenge 1: Process coarse motion vectors

Rotational Component Elimination

- When ego agent is turning, the motion vector along x-axis and y-axis can be written as:

$$vx_q = \frac{\Delta Z x_q^t}{Z_Q} - \Delta\phi_y f + \frac{\Delta\phi_x x_q^t y_q^t}{f} - \frac{\Delta\phi_y {x_q^t}^2}{f},$$

$$vy_q = \frac{\Delta Z y_q^t}{Z_Q} + \Delta\phi_x f - \frac{\Delta\phi_y x_q^t y_q^t}{f} + \frac{\Delta\phi_x {y_q^t}^2}{f}.$$

- Eliminate $\frac{\Delta Z}{Z}$ from the equations above:

$$x_q^t f \Delta\phi_x + y_q^t f \Delta\phi_y = y_q^t vx_q - x_q^t vy_q,$$

Challenge 1: Process coarse motion vectors

Rotational Component Elimination

- When ego agent is turning, the motion vector along x-axis and y-axis can be written as:

$$vx_q = \frac{\Delta Z x_q^t}{Z_Q} - \Delta\phi_y f + \frac{\Delta\phi_x x_q^t y_q^t}{f} - \frac{\Delta\phi_y x_q^{t\,2}}{f},$$

$$vy_q = \frac{\Delta Z y_q^t}{Z_Q} + \Delta\phi_x f - \frac{\Delta\phi_y x_q^t y_q^t}{f} + \frac{\Delta\phi_x y_q^{t\,2}}{f}.$$

- Eliminate $\frac{\Delta Z}{Z}$ from the equations above:

$$x_q^t f \Delta\phi_x + y_q^t f \Delta\phi_y = y_q^t vx_q - x_q^t vy_q.$$

Linear equation about $\Delta\emptyset_x$ and $\Delta\emptyset_y$, can be solved with 2 or more motion vectors.

Challenge 1: Process coarse motion vectors

Rotational Component Elimination

- How to choose motion vectors to efficiently solve the linear equations?

$$x_q^t f \Delta\phi_x + y_q^t f \Delta\phi_y = y_q^t v x_q - x_q^t v y_q,$$

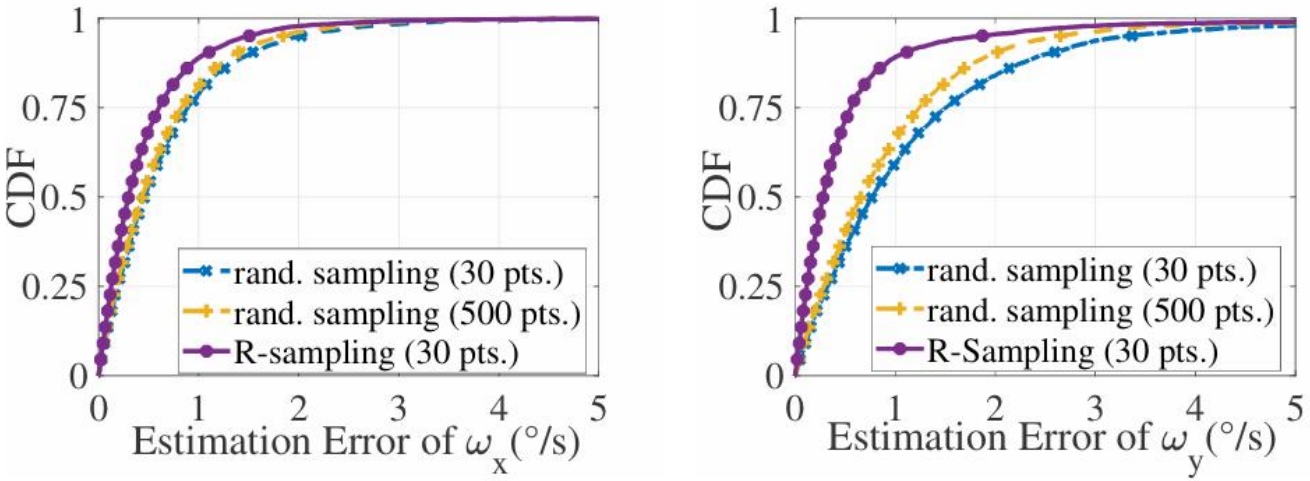
Challenge 1: Process coarse motion vectors

Rotational Component Elimination

- How to choose motion vectors to efficiently solve the linear equations?

$$x_q^t f \Delta\phi_x + y_q^t f \Delta\phi_y = y_q^t v x_q - x_q^t v y_q,$$

- *R*-sampling: choosing *k* motion vectors nearest to Focus of Expansion (FOE).



(a) Estimation error of ω_x (b) Estimation error of ω_y

Challenge 2: Extract foregrounds

- Key Observations:
  - All foreground objects stand on the ground.
  - Motion vectors on the same object are similar.

- Core idea:
  - First utilize a unique feature of motion vectors to estimate the ground region in each frame.
  - Then identify objects by clustering similar motion vectors starting from the ground region.

Challenge 2: Extract foregrounds

Ground Estimation

- Normalize the preprocessed motion vectors:

$$v\_norm = \frac{\Delta Z}{fY}$$

Challenge 2: Extract foregrounds

Ground Estimation

- Normalize the preprocessed motion vectors:

$$v\_norm = \frac{\Delta Z}{fY}$$

Magnitude of the normalized motion vector is only related to height of object point ***Y***!

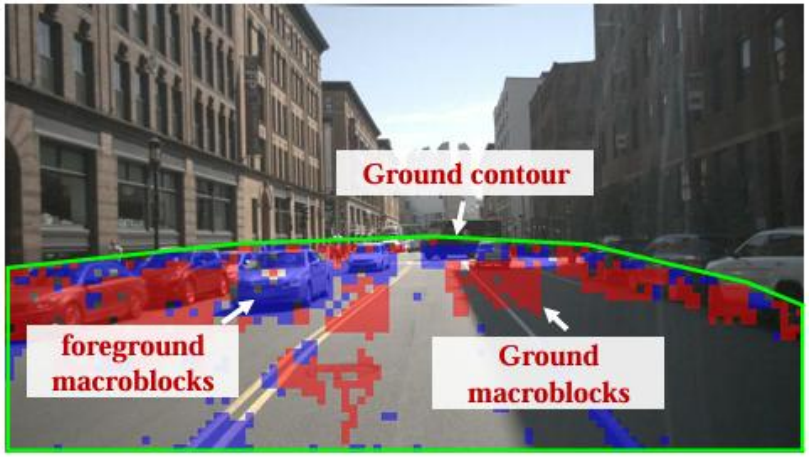
Challenge 2: Extract foregrounds

Ground Estimation

- Normalize the preprocessed motion vectors:

$$v\_norm = \frac{\Delta Z}{fY}$$

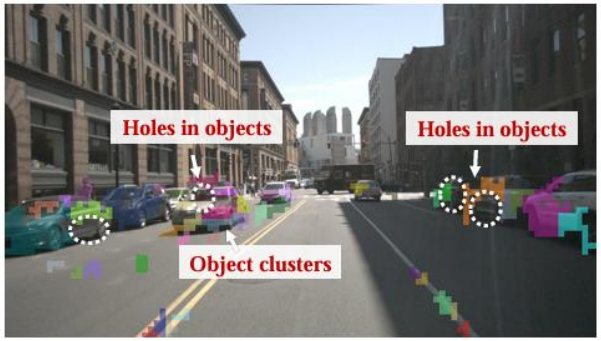
- Motion vectors with the smallest normalized magnitude belongs to ground, statistic method is utilized for calculating the threshold.



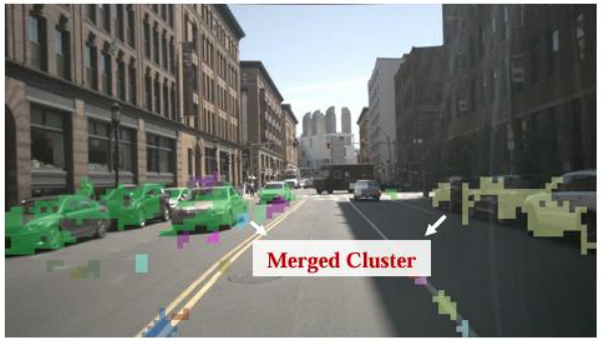
(a) Estimated ground with foreground seed macroblocks

Challenge 2: Extract foregrounds

- Region-growing-basd Foreground Clustering
  - Non-ground macroblocks inside ground contour are foreground seeds.
  - BFS-based clustering are utilized to cluster similar motion vectors.
  - Object clusters with similar motion vectors are merged as final foregrounds.



(b) Region-growing-based object clustering with holes in identified objects



(c) Similar foreground objects can be merged into one region

(d) Final foreground obtained by taking convex hull of foreground regions

Adaptive video encoding

- Uplink bandwidth is estimated with slicing window of 2ms length.

- Optimal QP Assignment
  - Observation: Larger extracted foregrounds are more likely to cover more real foregrounds.
  - QP offset value δ=c*size(FG)

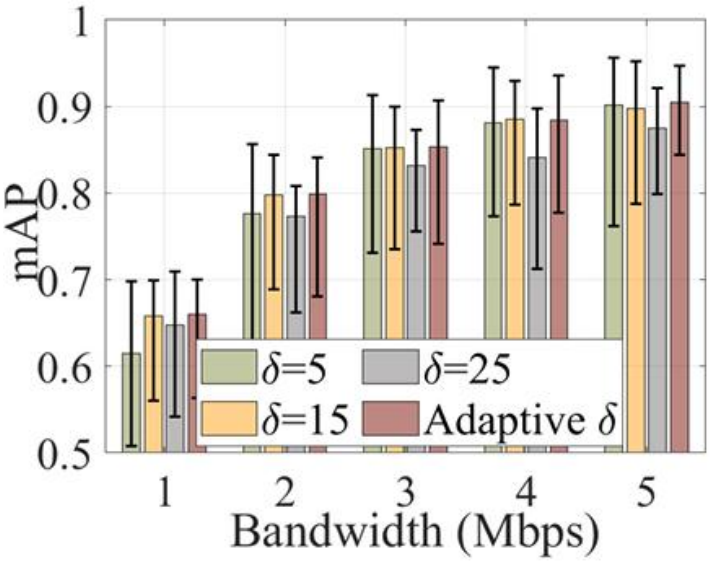
- Motion-vector-based offline tracking is enabled when uplink is interrupted.

Performance Evaluation

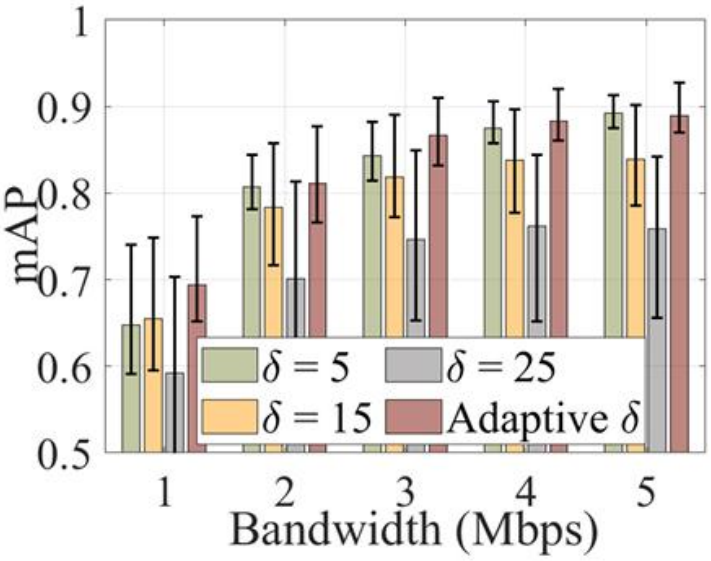
- Settings
  - Datasets: nuScenes, RobotCar
  - Video analytics task: object detection (Car + Pedestrian)
- Metrics
  - Average Precision (AP)
  - Response Time
- Compared Methods
  - O^3 (INFOCOM' 21)
  - EAAR (MobiCOM' 19)
  - DDS (SIGCOMM' 20)

Effectiveness of Optimal QP Assignment

- Vary δ from 5 to 25 with an interval of 10, along with adaptive δ in DiVE.
- Adaptive δ achieves the highest mAP under most settings.



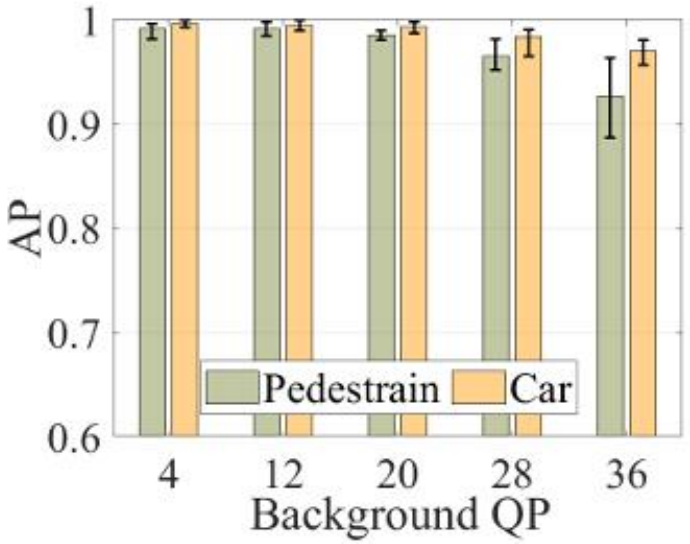
(a) mAP on RobotCar



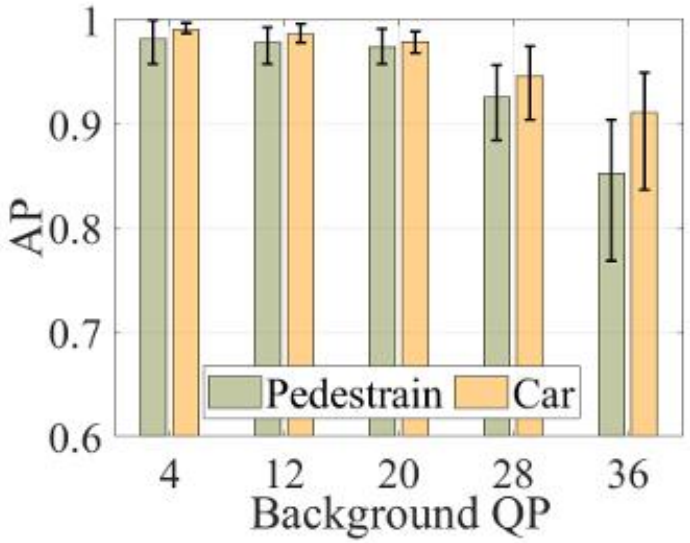
(b) mAP on nuScenes

Effectiveness of Foreground Extraction

- Fix the QP value of foregrounds to 0, vary QP value of backgrounds from 4 to 36 with an interval of 8.
- AP of the pedestrian and car is larger than 0.97 on both datasets when QP=20.



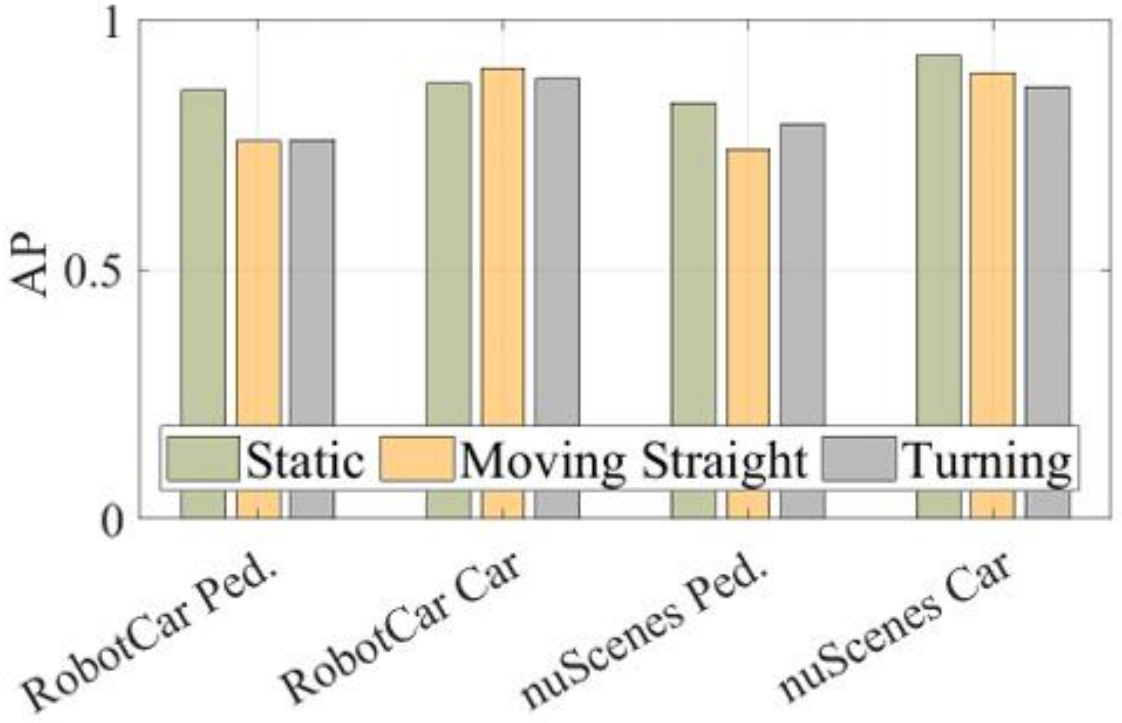
(a) AP on RobotCar



(b) AP on nuScenes

Impact of Different Motion State

- Manually divide RobotCar and nuScenes into static, moving straight and turning.
- DiVE consistently achieves car detection AP exceeding 0.8 across both datasets.



Impact of Different Motion State

- Samples of foregrounds extracted by DiVE.

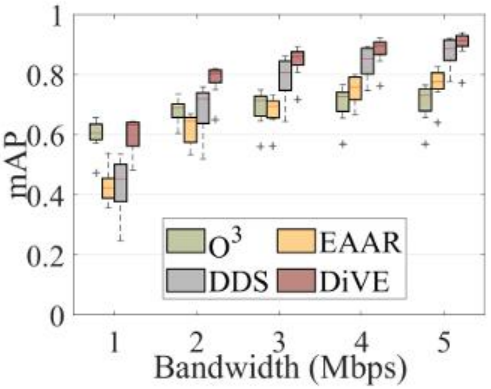
(a) Extracted foregrounds when vehicle is moving straight.

(b) Extracted foregrounds when vehicle is turning.

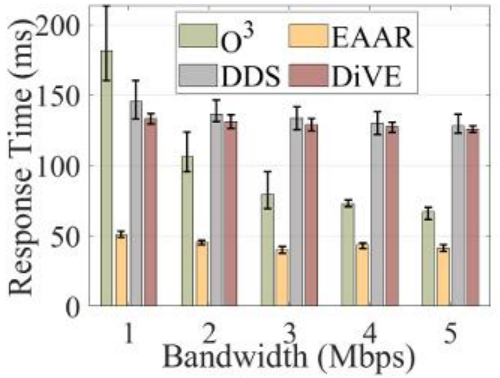
(c) Extracted foregrounds when vehicle is static.

End-to-end Performance Evaluation

- Varying netwok bandwidth from 1Mbps to 5Mbps with an interval of 1Mbps.

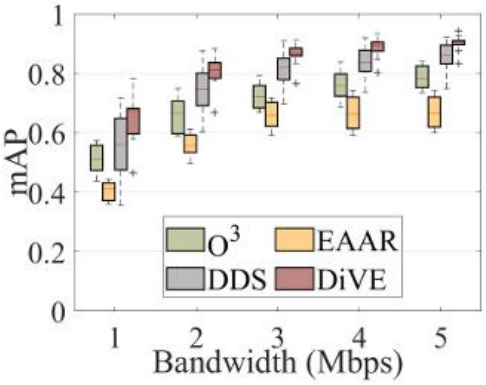


(a) mAP on RobotCar

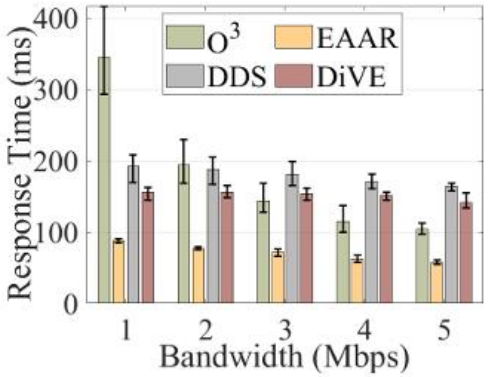


(b) Response Time on RobotCar

- DiVE improves mAP by 39.1% and reduce response time by 19.1% at most compared to DDS.



(a) mAP on nuScenes



(b) Response Time on nuScenes

Conclusion

- We propose a light-weight video analytics system designed for mobile agents called DiVE.
- In DiVE, mobile agent extracts interested foregrounds based on low-cost motion vectors for differential encoding.
- We conduct extensive experiments to demonstrate the efficiency of DiVE.

Thanks!

Q & A