A Cloud-Fog Architecture for Video Analytics on Large Scale Camera Networks Using Semantic Scene Analysis

Kunal Jain, Kishan Sairam Adapa, Kunwar Grover, Ravi Kiran Sarvadevabhatla, Suresh Purini

International Institute of Information Technology, Hyderabad

Table of Content

CCTV Cameras

Uses

- Security of public (malls) and private (apartments) spaces
- Traffic Monitoring
- Others: industrial process control, environmental monitoring etc.

Helps in maintaining a visual record of activities

COMMUNICATIONS OF THE ACM

Search

HOME | CURRENT ISSUE | NEWS | BLOGS | OPINION | RESEARCH | PRACTICE | CAREERS | ARCHIVE | VIDEOS

Home / News / A World With a Billion Cameras Watching You Is Just... / Full Text

ACM TECHNEWS

A World With a Billion Cameras Watching You Is Just Around the Corner

By The Wall Street Journal
December 9, 2019
Comments

VIEW AS: SHARE:

Industry researcher IHS Markit expects the number of cameras used for surveillance to rise above 1 billion by the end of 2021, marking an almost 30% increase from the 770 million such cameras in use today.

China will continue to account for more than 50% of the total, but fast-growing, populous nations such as India, Brazil, and Indonesia also will help to drive growth in the sector.

The global security camera industry has been spurred by developments in image quality and artificial intelligence (AI), technologies that allow better and faster facial recognition and video analytics.

SIGN IN for Full Access

User Name

Password

» Forgot Password?

» Create an ACM Web Account

SIGN IN

MORE NEWS & OPINIONS

CISA Director Calls on Tech Companies to Build Security into Products
Axios

Long-Term Mentoring for Computer Science Researchers

Problem Perspective

- One CCTV camera @20 fps, 1080 pixel resolution, generates 72 MB of data per hour
- 1000 such cameras generate 72 GB per hour, 2 TB in a day
- 50000 cameras generates 100 TB in a day

How do we answer a simple query such as

"Track a Red Toyota Sedan in an area X"?

Real Time Vehicle Tracking

Information Retrieval Tasks or Query Types

Event Detection

- Events that happen actively but don't require real-time processing
- Ex. jumping of traffic lights, not wearing seat belt

Outlier Detection

- Implicit queries with near real-time processing
- Ex. traffic jams and accident detection

Tracking and Pursuit

- Tracking a vehicle through a road network or analysing the possible paths taken
- Can be offline or active real-time

Data Analytics

- Perform analytics such as rush hour analysis, busy roads, etc
- Study correlations such as traffic density and accidents, AQI, etc

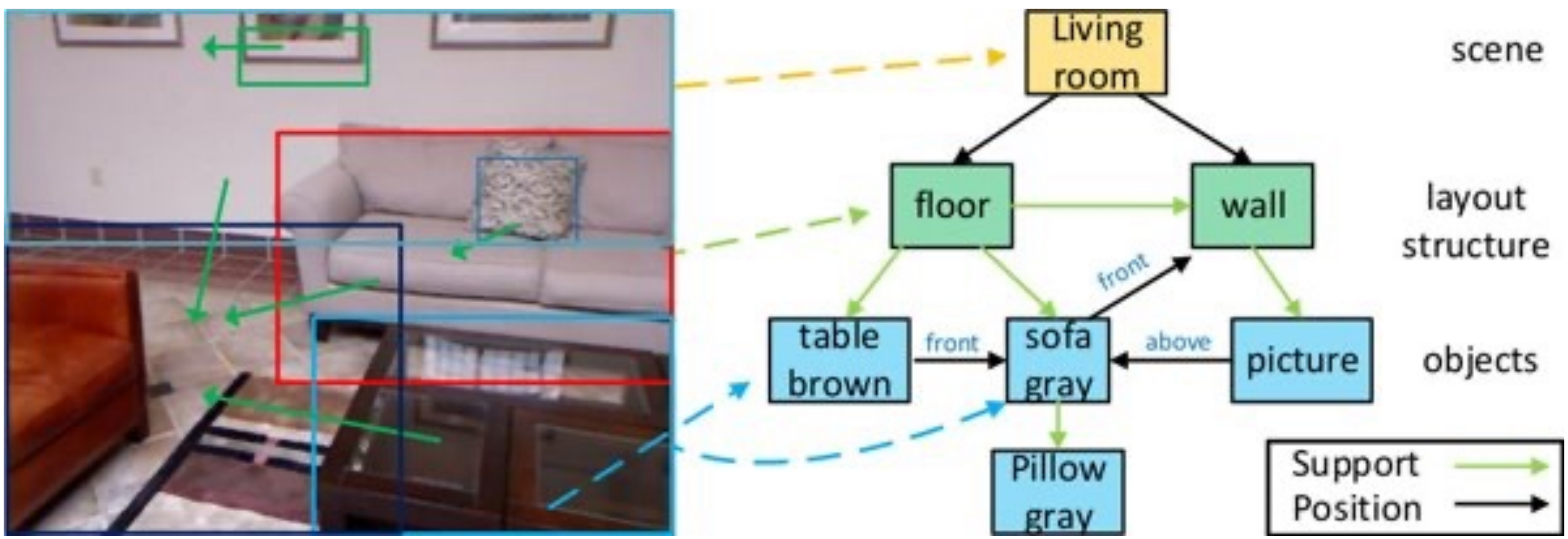
Our Approach : Overview

Puts together two ideas

- Semantic Scene Analysis
- Cloud – Fog Architecture

Semantic Scene Analysis

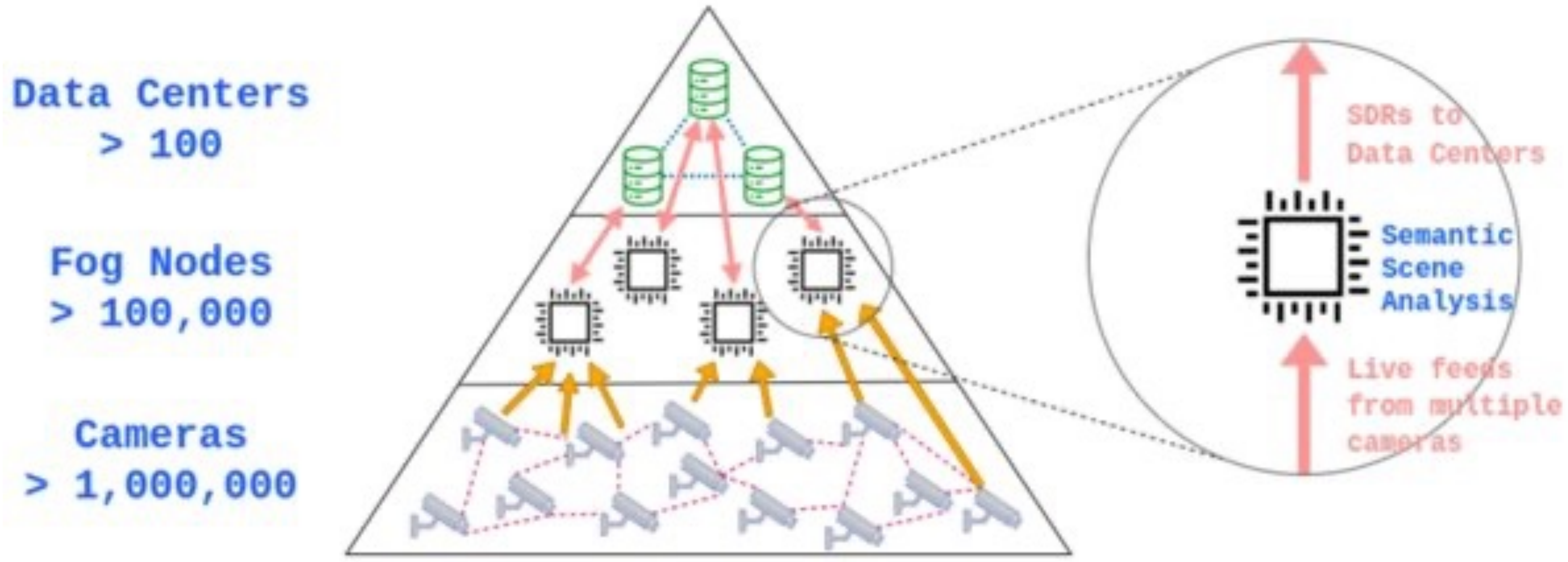
- Generate a textual description of a scene
- Establish relationships between recognizable objects
- Store objects as Scene Description Records (SDRs)



Semantic Scene Analysis : Example

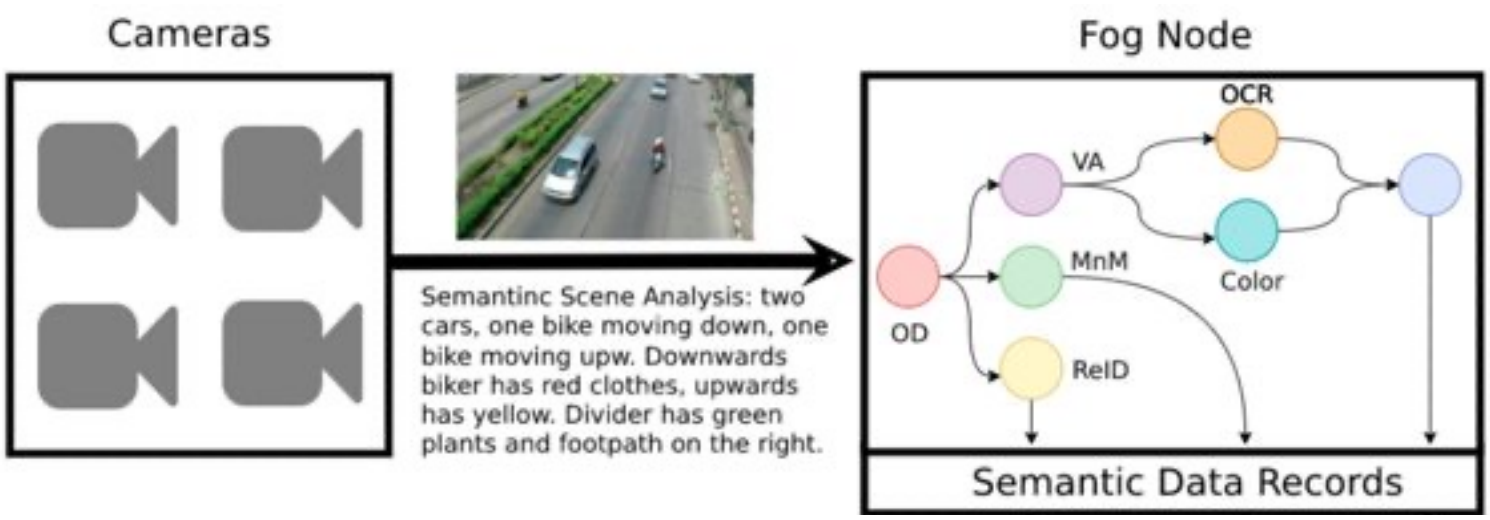
- "Black car moving right"
- "White Honda City waiting at a red light"
- "Maroon sedan at location X between time A and B"
- "Cars similar to given"

Cloud – Fog Architecture



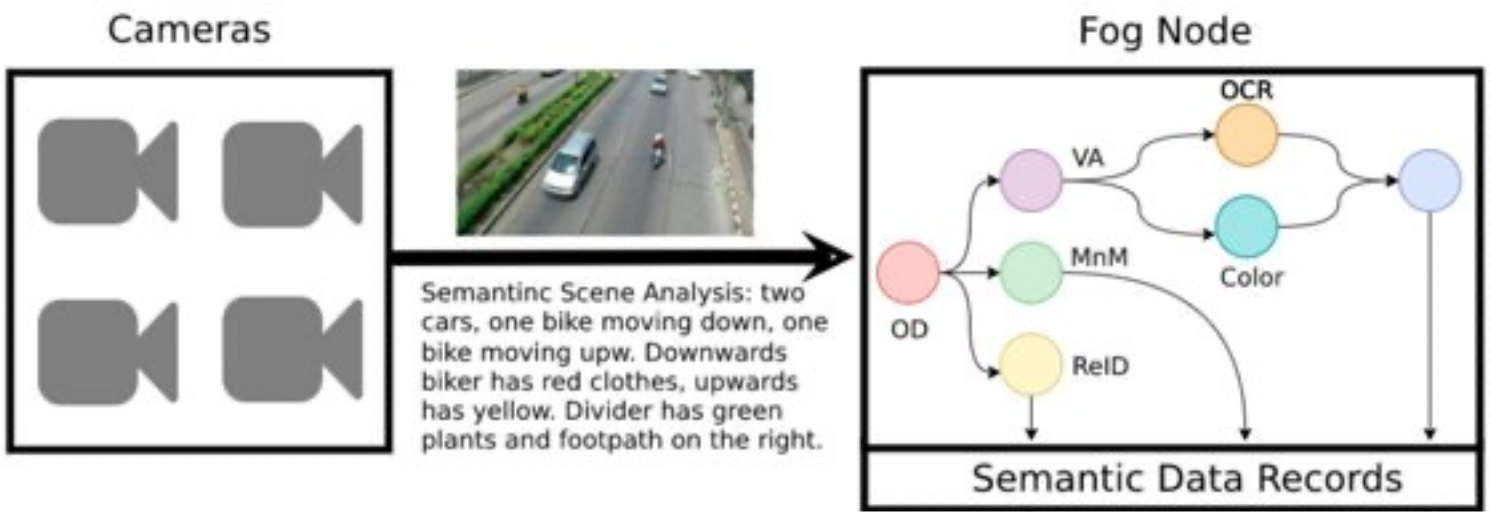
SSA on Fog Nodes

- Can add and remove nodes in the DAG on the fly
- New pipelines can subscribe to components they share with pre-existing deployment
- Tested with reidentification networks, make-and-model, color detection and number plate identification models on a 6GB GPU and 4GB CPU



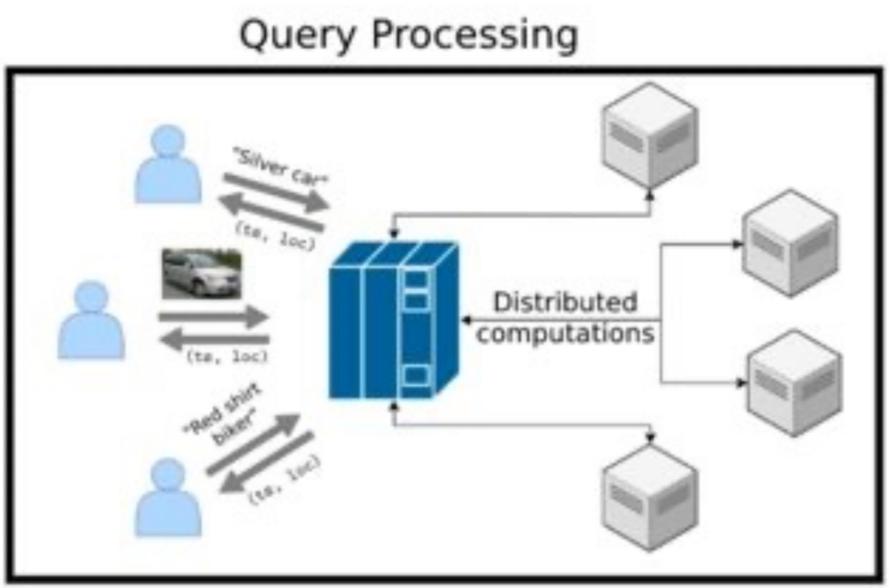
Communication Queues on Fog Nodes

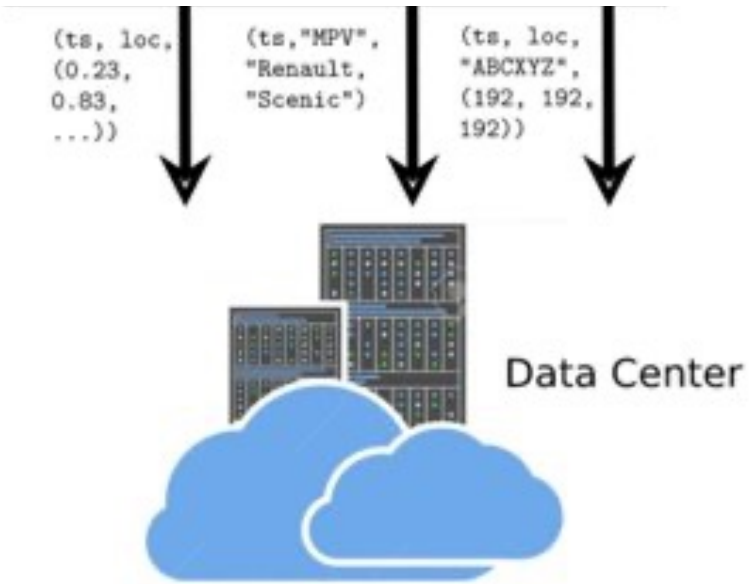
- Communication between DAG nodes using custom queues
- Publisher-subscriber model
- Subscription to nodes is quick
- Low drop rate, repetitions
- Very low overhead
- Allows reasonable number of subscribers to read parallelly



Data Storage on Data Centers

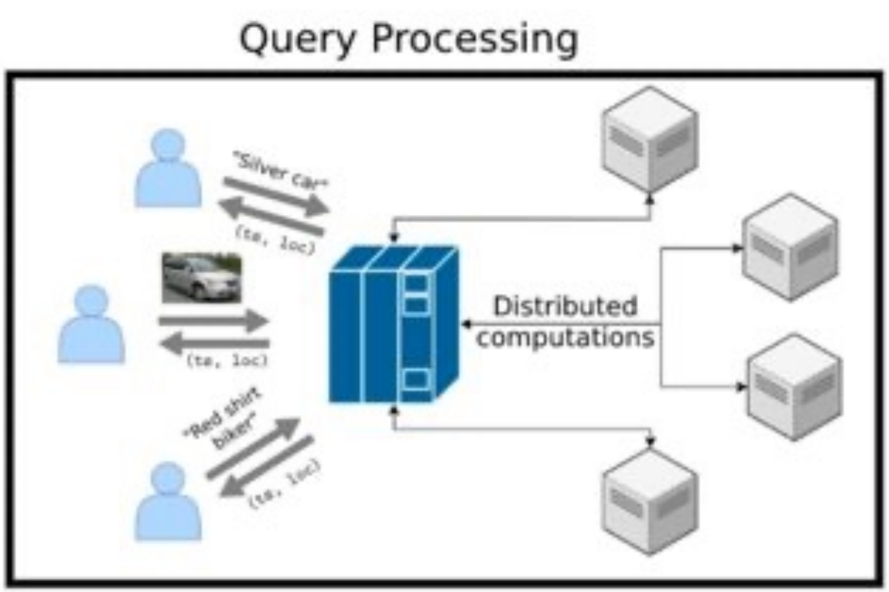
- Distributed NoSQL database with spatio-temporal indexing
- Allows for horizontal scalability as new pipelines can add new columns
- Data centers can communicate with each other to answer queries
- 98% reduction in data size with SDRs

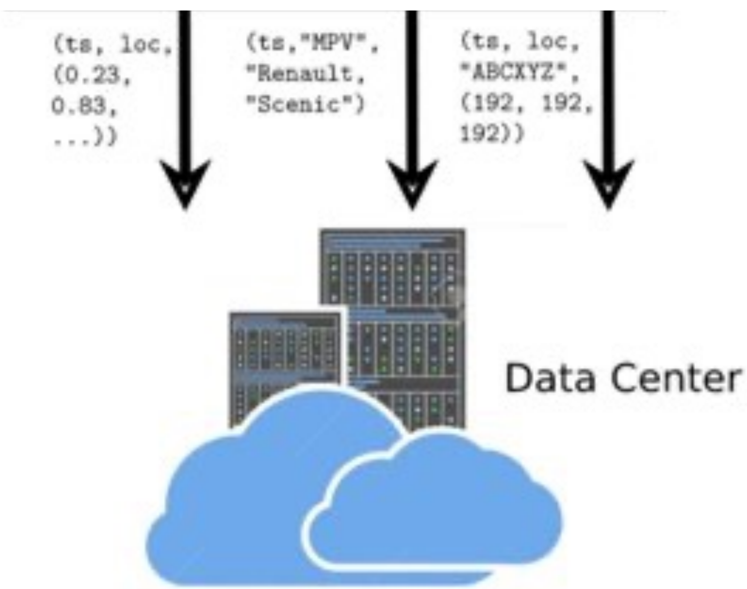




Query Processing on Data Centers

- Use Apache Spark to write queries
- Integrates well with our choice of database
- Allows arbitrary complexity for new kinds of queries





Implementation Details

Architectural Component	Support
Ingestion Nodes	Kafka
Distributed Deel Learning Inference	Ray
Queues	NATS Jetstream
Data Storage	Accumulo
Spatio-Temporal Indexing	Geomesa
Query Processing	Spark

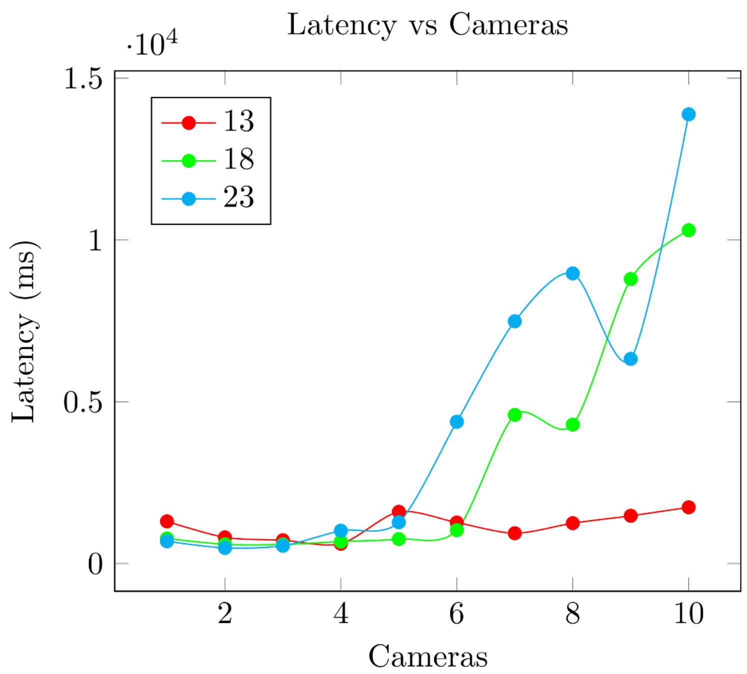
Experimental Methodology

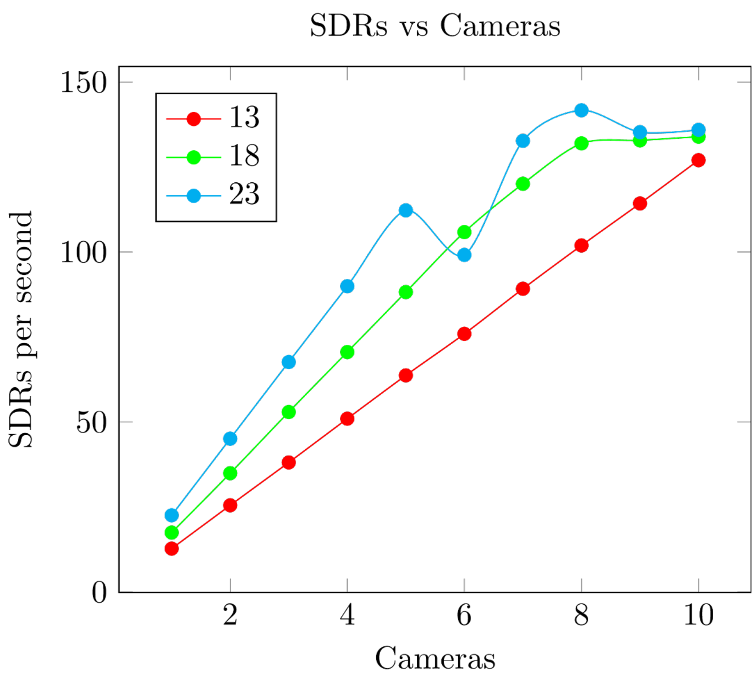
Properties of the system

- All fog nodes work independently of each other
- So do all the data centers
- Network bandwidth can limit performance
- Need to ensure fog nodes and data centers perform well individually and both can ingest from respective sources
- End-to-end testing and practical application as well

Experimental Results

- Latency and throughput as number of cameras increase on a fog node

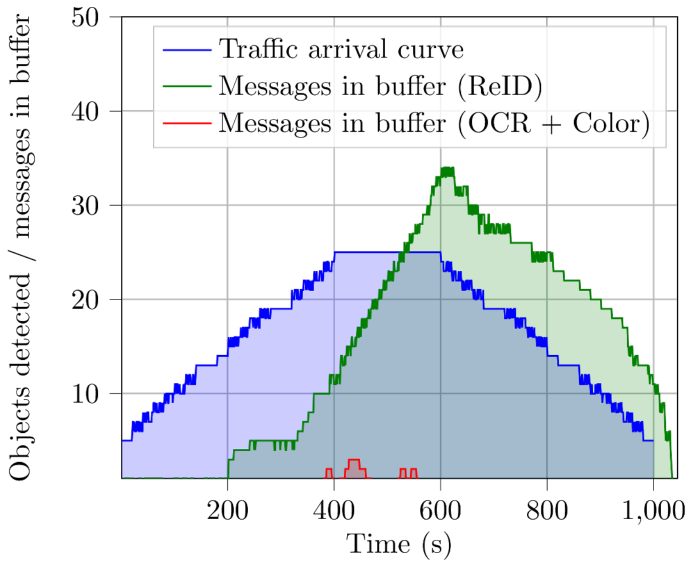




Experimental Results

- Buffer queue as traffic pressure increases
- Ingestion latency at data center
- End-to-end deployment

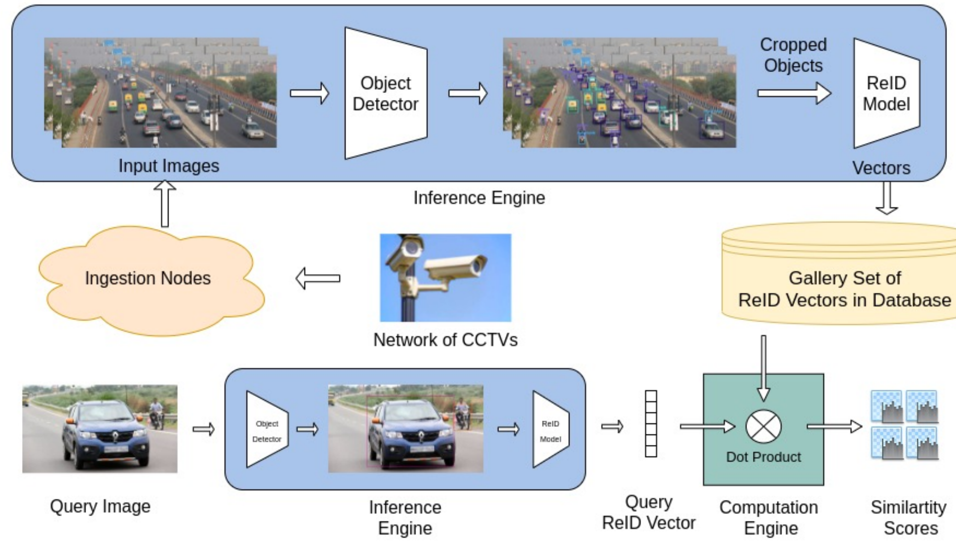
Fog Nodes	Latency (ms)
10	476.45
20	476.81
30	504.31
40	523.30
50	547.23
100	577.60

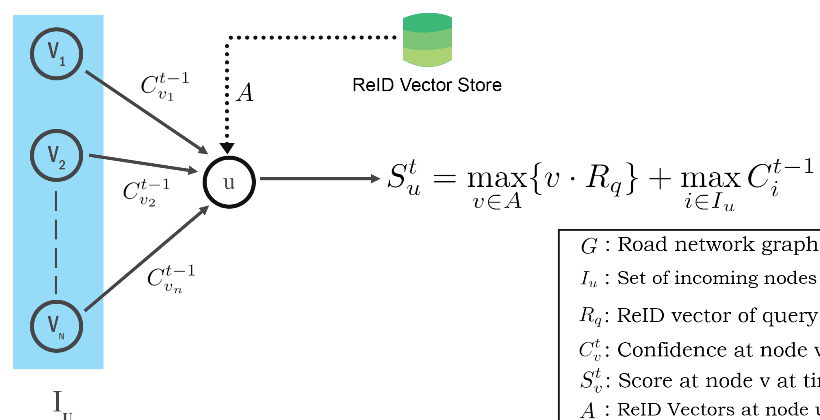


#Nodes	Processing Time (ms)	Insertion Time (ms)
1	1571	393
2	1522	404
3	1575	412
4	1546	422
5	1571	434
6	1554	442

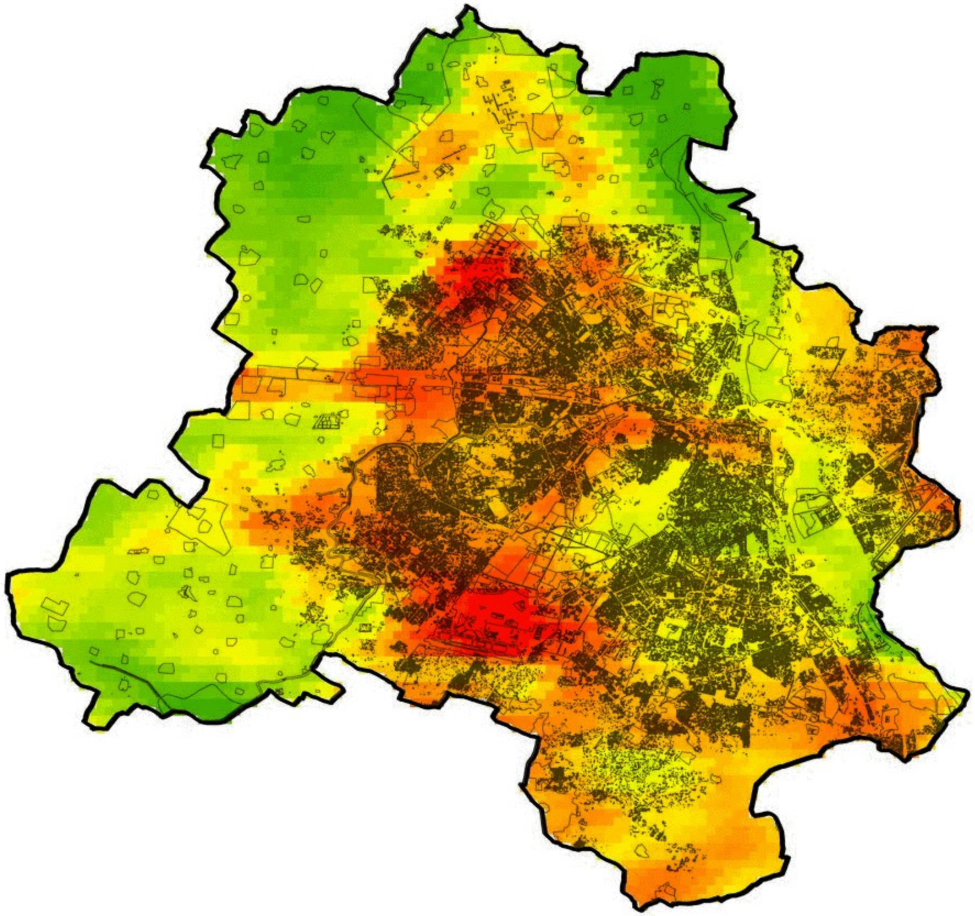
Application : Vehicle Pursuit

- Tracking a vehicle through a city with the help of CCTV cameras
- AI City Challenge Dataset
- Iterative algorithms with incrementing time steps
- Maintain a set of top-k matches at each iteration
- Concurrent insertion and queries
- Real time!

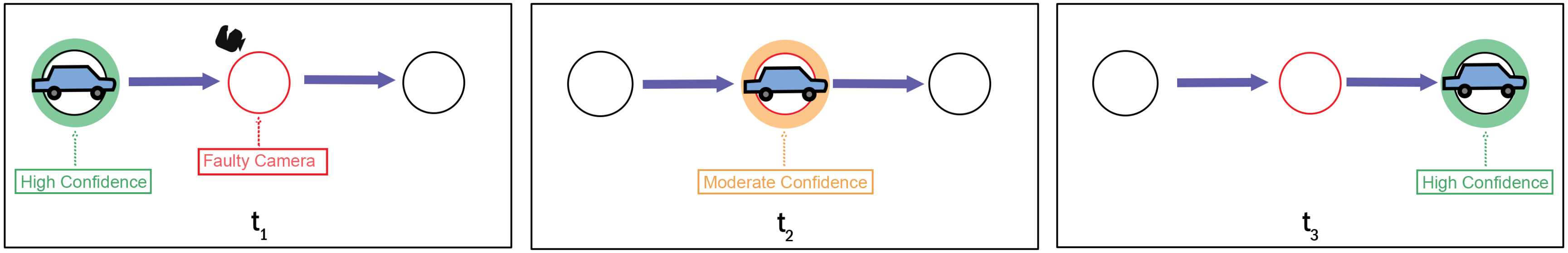


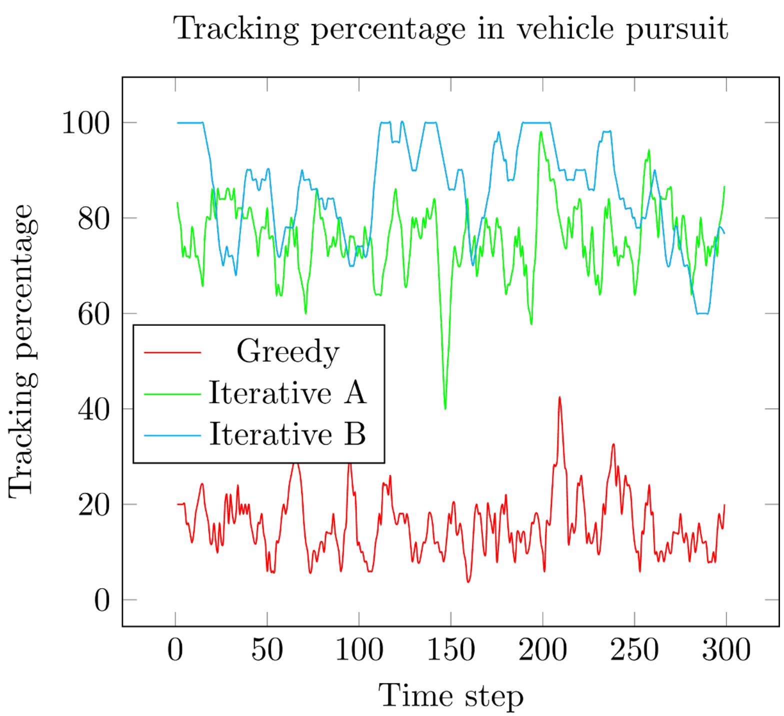


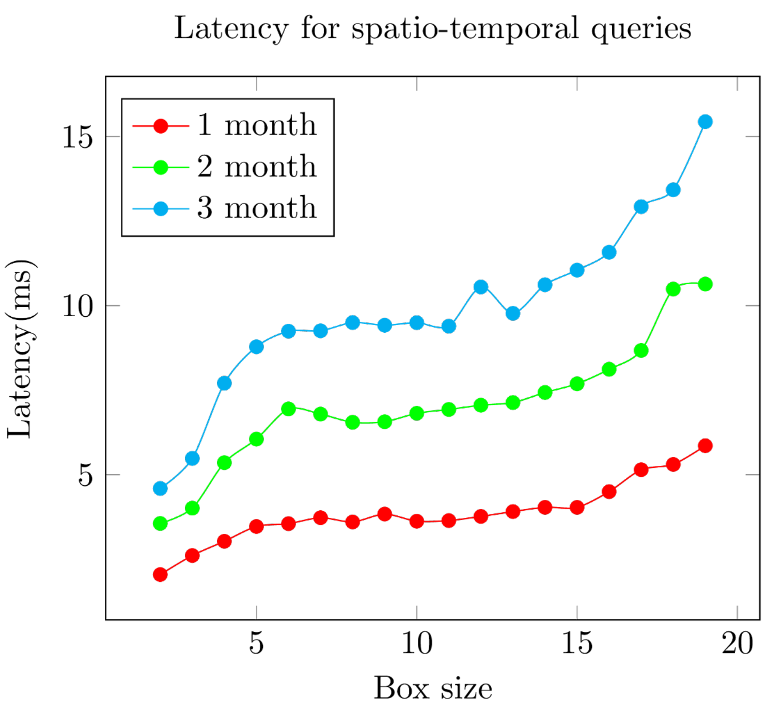
G : Road network graph $(u \in G)$
I_u : Set of incoming nodes to node u
R_q: ReID vector of query image
C_v^t: Confidence at node v at time t
S_v^t: Score at node v at time t
A : ReID Vectors at node u at time t
k : Size of maximum likelihood set



Application : Vehicle Pursuit







Future Works

- Applications to other camera networks
- Large scale deployment
- Privacy awareness
- Bounds on tracking algorithm

THANK YOU