Introduction

A video-processing framework combines stream processing, batch processing, and deep learning to realize deep intelligence that can help reveal knowledge hidden in video data. Evaluations of performance, scalability, and fault tolerance showed the framework's effectiveness.

Introduction(1/3)

- Application such as smart transportation and security surveillance collect vast amounts of video data daily. Surveillance videos are the biggest source of big data.

Introduction(2/3)

- Video analysis research has extensively explored such topics as feature extraction and video summary.

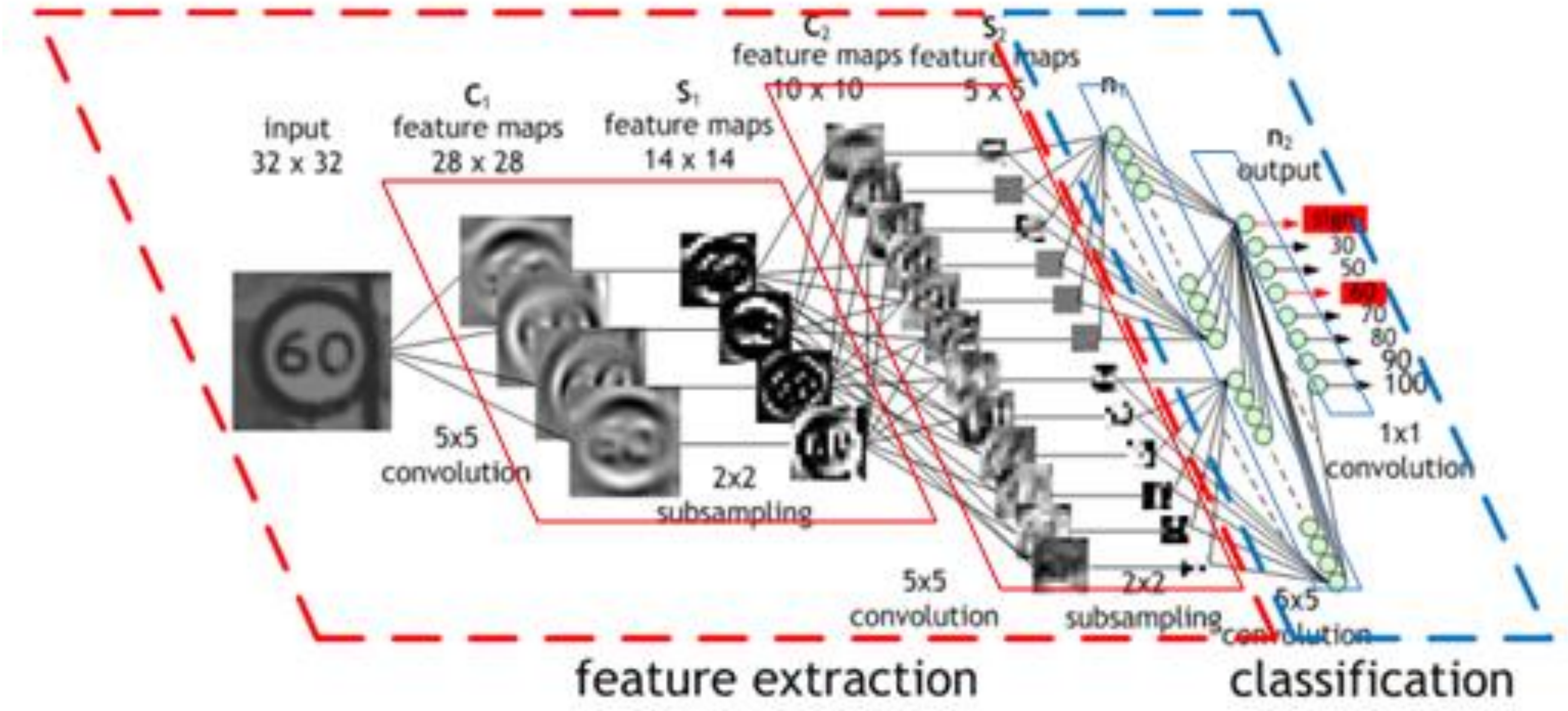
- Recently, video- and image-processing research has been deeply influenced by the success of deep learning initiated by Geoffrey Hinton and Ruslan Salakhutdinov.

Introduction(3/3)

- Owing to the volume, velocity, and complexities of video data, some video-processing research has employed the latest computing technologies, such as cloud computing, using offline processing or stream processing.

What is CNN?

What is CNN?


input
32 x 32
C_1
feature maps
28 x 28
S_1
feature maps
14 x 14
C_2
feature maps
10 x 10
S_2
feature maps
5 x 5
n_1
n_2
output
30
50
70
80
90
100
5x5
convolution
2x2
subsampling
5x5
convolution
2x2
subsampling
6x5
convolution
1x1
convolution
feature extraction
classification

Five key quality attributes

Performance

Availability

Scalability

Modifiability

Portability and usability

Performance

- There are two types of performance requirements.

- The first is batch processing of large volumes of historical video data .

- The second type is for real–time processing situations which is performance critical.

Availability

- The framework must be reliable to allow hardware failures. It must provide dependable services to users.

Scalability

- The cloud must be able to scale up, with the ability to add video-sensing devices and processing nodes dynamically.

Modifiability

- Because big data processing techniques are evolving quickly, the framework must allow easy changes of its functionalities so that future modifications are easy.

Portability and usability

- The framework must also provide a unified way to manage video equipment and video data. It must also support different Linux-style OSs.

- **(Other important quality attributes exist—for example, safety and testability— but aren't part of this article's focus.)**

software architecture styles

Service-oriented architecture(**SOA**)

Publish-subscribe

MapReduce

Shared Data

Layered architecture

Service-oriented architecture

- Owing to the complexities of video data processing and the various technologies involved, the framework must employ an SOA. This will enable modifiability, and functionality upgrades won't affect users.An SOA can also help achieve portability.

Publish–subscribe

- To decouple event consumers and providers, a publish–subscribe mechanism should handle events while monitoring the framework’s running status. This will help achieve the framework’s availability such that users can detect malfunction states and initiate appropriate responses.

MapReduce

- The framework must analyze enormous volumes of video data. MapReduce is a large-scale distributed-computing paradigm based on the divide-and-conquer strategy. It can efficiently process distributed large datasets in parallel. So, it can help meet performance, scalability, and availability requirements.

Shared Data

- Different processing components—for example, background subtraction and video summary—will use video data. So, the framework uses the Shared Data pattern to achieve sharing of video data, which can help improve performance.

Layered architecture

- Framework functionalities might evolve separately, and the processing of big video data can also be separated into different types and might evolve independently. So, the framework uses a layered architecture to separate the different concerns. This can support modifiability, portability, and usability.

Mapping architecture styles to quality attributes

Architecture style	Supported quality attributes
Service-oriented architecture	Portability, modifiability
Publish–subscribe	Availability, modifiability
MapReduce	Performance, scalability, availability
Shared Data	Performance, availability
Layered architecture	Modifiability, portability, usability

Architecture

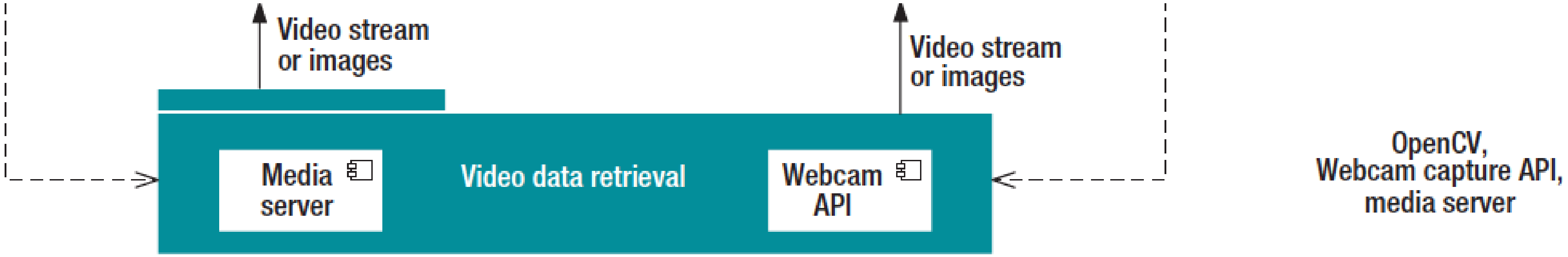
Bottom layer

Data-processing layer

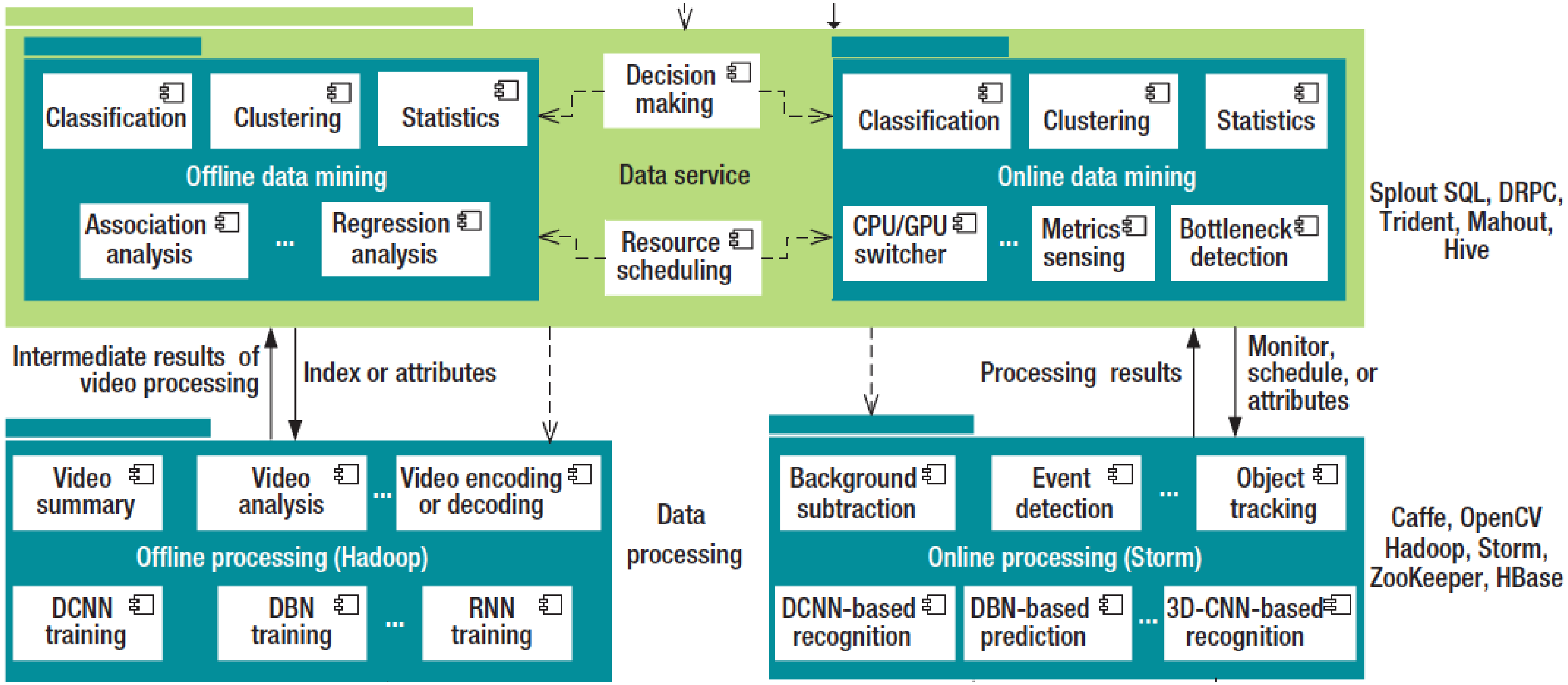
Top Layer

Bottom layer

- The bottom layer is the data retrieval layer. It’s a generalization of video data collection,in which a media server and webcam API receive different types of video data from various cameras.



Data-processing layer(1/5)



Data-processing layer(2/5)

- The data-processing layer comprises the offline-processing package, based on Apache Hadoop.

- The online-processing package, based on Apache Storm stream processing(http://storm.apache.org).

Data-processing layer(3/5)

- Some video-processing tasks—for example, video summary and encoding and decoding of large amounts of video data—occur during offline processing.

- The online-processing package handles some lightweight video-processing tasks—for example, background subtraction. On the basis of the offline neural-network training results, this package achieves real-time recognition, including the recognition of events, objects, and behaviors.

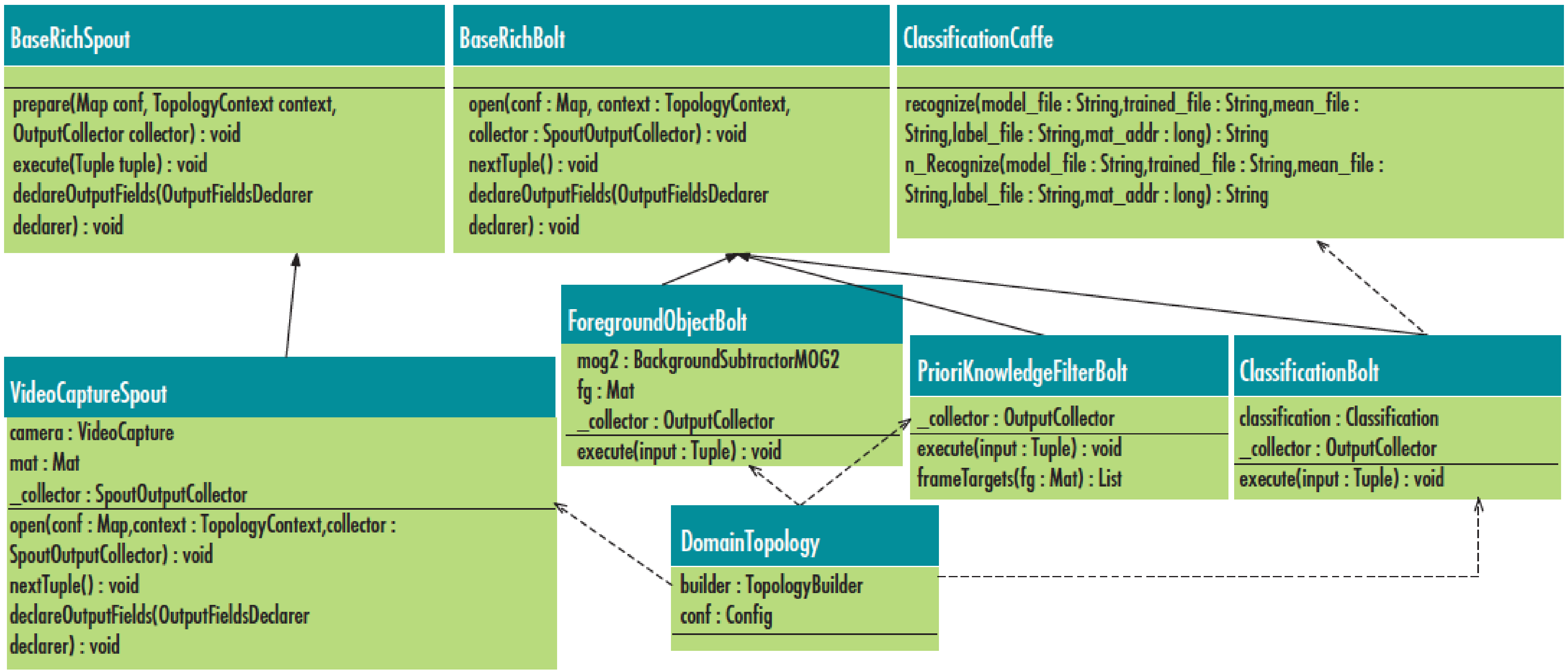
Data-processing layer(4/5)-
The key classes for online processing

- ForegroundObjectBolt-removes the background.

- PrioriKnowledgeFilterBolt-locates the moving targets and filters out the object of classification according to the prior knowledge.

- ClassificationBolt-classifies the moving targets.

Data-processing layer(5/5)-Class diagram



Top Layer

- The top layer is the domain service layer, which developers can use to create different domain applications such as smart transportation and smart campuses.

The configuration of the experiment's cloud infrastructure.

experiment's cloud infrastructure

DCNN training performance

Camera's Scalability

Fault Tolerance

The configuration of the experiment's cloud infrastructure.

Nine IBM 3650 servers(64 Gbytes of RAM, 24 cores, and 12 Tbytes of storage) deployed as a cloud. Each node ran the Ubuntu 14.04 server and a modified version of the Caffe deep-learning tool.

	Node								
	1	2	3	4	5	6	7	8	9
Roles	Name- Node Nimbus DRPC server Resource- Manager	DataNode Supervisor DRPC server Node- Manager	DataNode Supervisor DRPC server Node- Manager	DataNode Supervisor DRPC server Node- Manager	DataNode Supervisor DRPC server Splout SQL Node- Manager	DataNode Supervisor DRPC server Node- Manager	DataNode Supervisor ZooKeeper DRPC server Node- Manager	DataNode Supervisor ZooKeeper DRPC server Node- Manager	DataNode Supervisor ZooKeeper DRPC server Node- Manager Webserver

DCNN training performance

- As the image size grew and the network became more complex,the recognition rate increased.However, the training and recognition time also grew. Considering the accuracy and real-time-performance trade-off, Network 1 was a practical choice.

Network	Image size (pixels)	No. of layers	Training time	Recognition time (ms)	Accuracy (%)
1	32 × 32	14	1 h, 1 min, 30 s	6.82	92.36
2	224 × 224	20	4 h, 57 min, 43 s	959.76	95.85

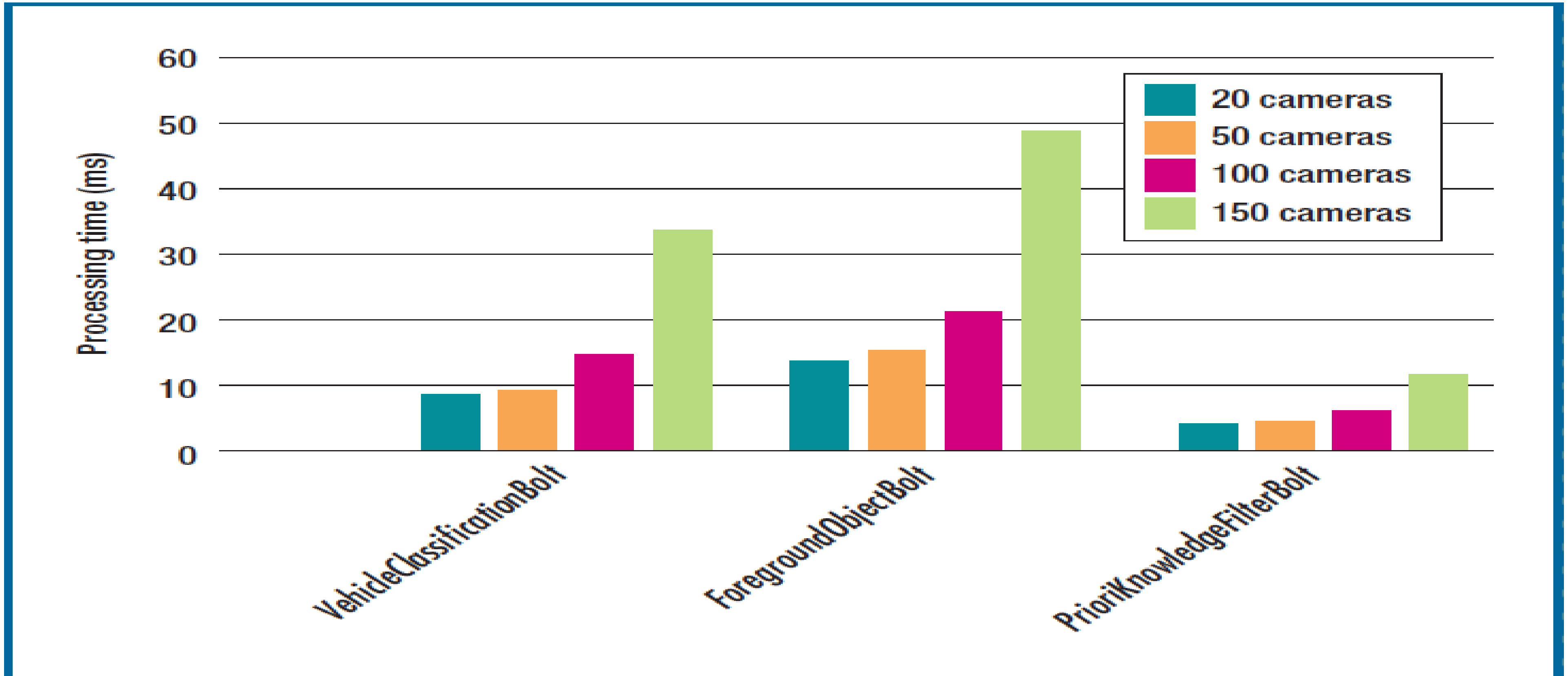
Example: How many cars had passed the cameras since 9 o’clock?

- Overall, it took approximately 43 ms (11.13 + 3.93 + 7.79 + 10.00 +10.00) from video frame acquisition to target recognition.

Run	Online-processing class*		
	ForegroundObjectBolt	PrioriKnowledgeFilterBolt	VehicleClassificationBolt
1	10.724	4.069	7.862
2	11.541	3.754	8.000
3	11.118	3.980	7.510
Average	11.130	3.930	7.790

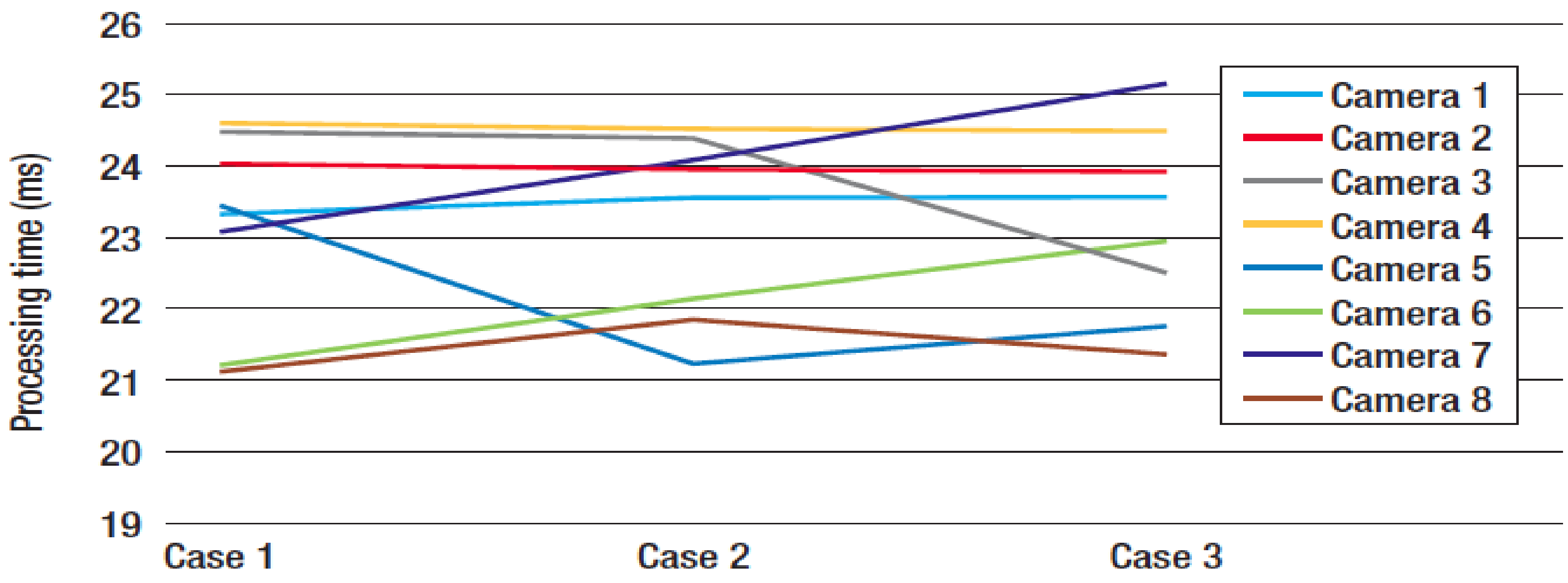
Camera's Scalability

- The test bed's performance scaled as we added cameras.



Fault Tolerance

- Case1 used seven supervisors.
- Case 2 used six supervisors.
- Case 3 used only four supervisors.



Conclusion

Conclusion

Challenge

Conclusion

- These evaluations also showed that our framework was effective and efficient for continuous real-time video processing.
- The SOA is enacted with clearly designed interface contracts and loosely couples components in different layers.
- Publish-subscribe decouples Storm generated events and event consumers that monitor the cluster status to optimize scheduling.
- MapReduce achieves parallel processing of video data, and we've employed it for implementations of some algorithms.
- The Shared Data pattern shares video data among processing components spanning multiple layers, implemented with the SOA.
- Finally,the layered architecture helps separate different data-processing concerns.

Challenge

- Integrating the SOA in a layered architecture posed two challenges:
- how to enable different features to interact harmoniously in different layers through well-defined interface contracts.
- how to use shared data to provide effective data partitioning for MapReduce.


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuję
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
спасибо
рахмат
Баярлалаа
mersi
kia ora
barka
welalin
tack
spas
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
хвала
asante
manana
obrigada
tenki
murakoze
chokrane
faafetai lava
vinaka
blagodaram
спасибі
kiitos
dankie
nanni
nandri
dhanyavad
maururu
köszönöm
hvala
gràçie
bayarlalaa
enkosi
akun
dankon
ačiū
djiere dieuf
tau
дякую
mamnun
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
sobodi
děkuji
sagolun
arigatô
takk
dakujem
trugarez
ありがとう
tanemirt
grazie
rahmet
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
ευχαριστώ
diolch
dhanyavadagalu
shukriya
mercé
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie