Video Analytic based Alerting in Public Safety Video Systems

University of Houston

Shishir Shah

Pranav Mantini

Department of Computer Science

City of Houston

Julie Stroup

Public Safety Video Program Manager

Mayor's Office of Public Safety & Homeland Security

Tony Weldon

Public Safety Video System Manager

Convergint Technologies

PSCR 2019 - Chicago

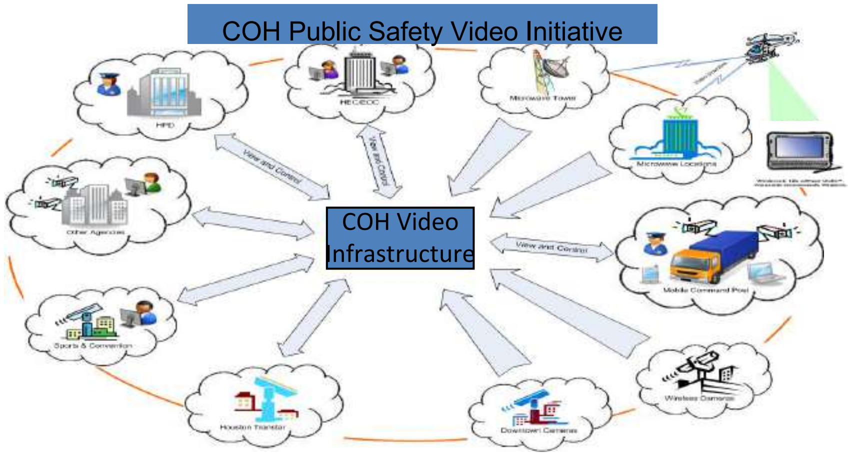
DISCLAIMER

This presentation was produced by guest speaker(s) and presented at the National Institute of Standards and Technology's 2019 Public Safety Broadband Stakeholder Meeting. The contents of this presentation do not necessarily reflect the views or policies of the National Institute of Standards and Technology or the U.S. Government.

Posted with permission

City of Houston - Public Safety Video

- 965+ city-owned/operated Public Space cameras
    - Homeland Security/Terrorism Nexus
    - 80% wireless infrastructure – 100 % PTZ
- Direct access to regional video partners - 500+ public space video streams
- User Workstations at 30+ locations – Command Centers & Patrol
- Users - Local, regional and federal agencies; major stadiums and parks
- Mobile and airborne
- Dedicated Support Team
- Infrastructure from many vendors
- NIST Grant – COH Public Safety Partner to UH
- Sister City Relationships

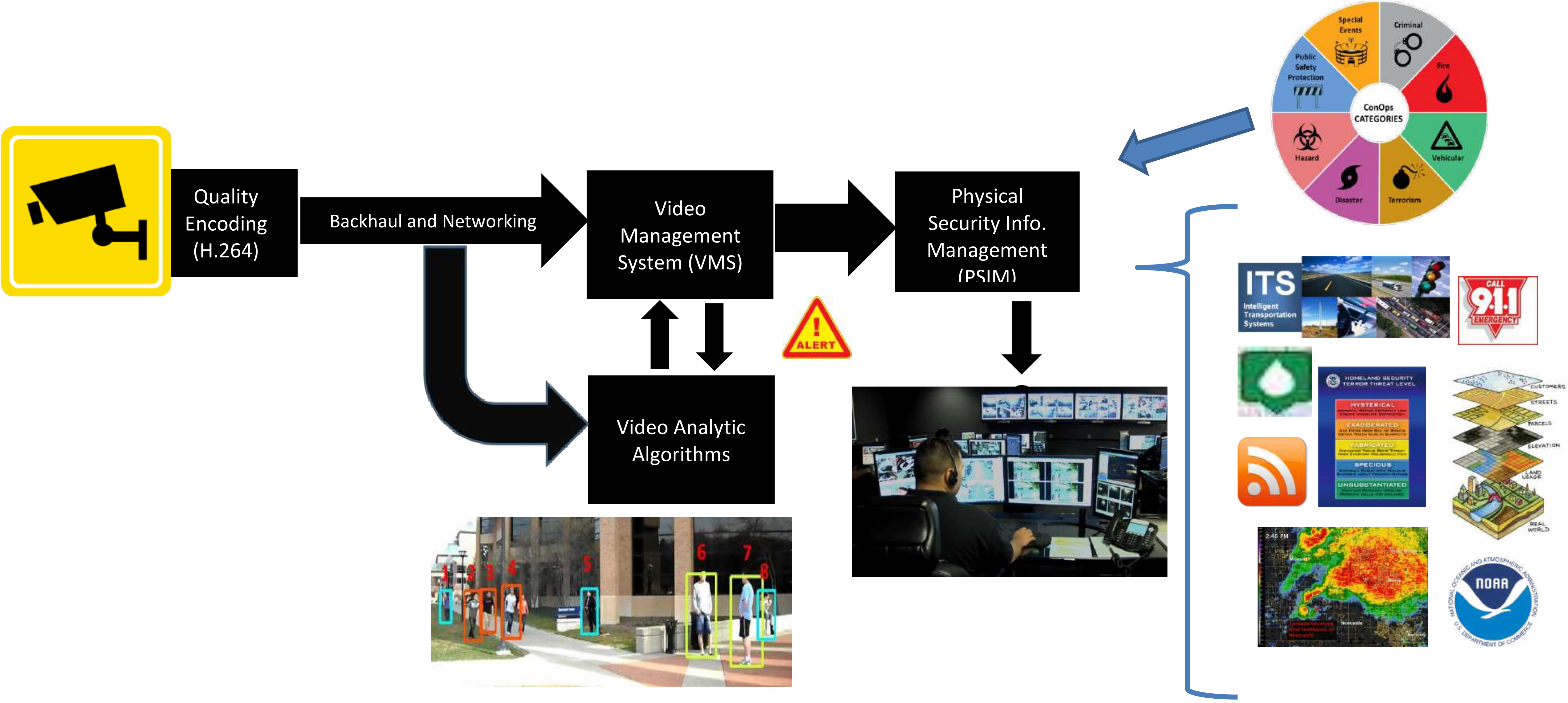


Project Focus and Structural Changes

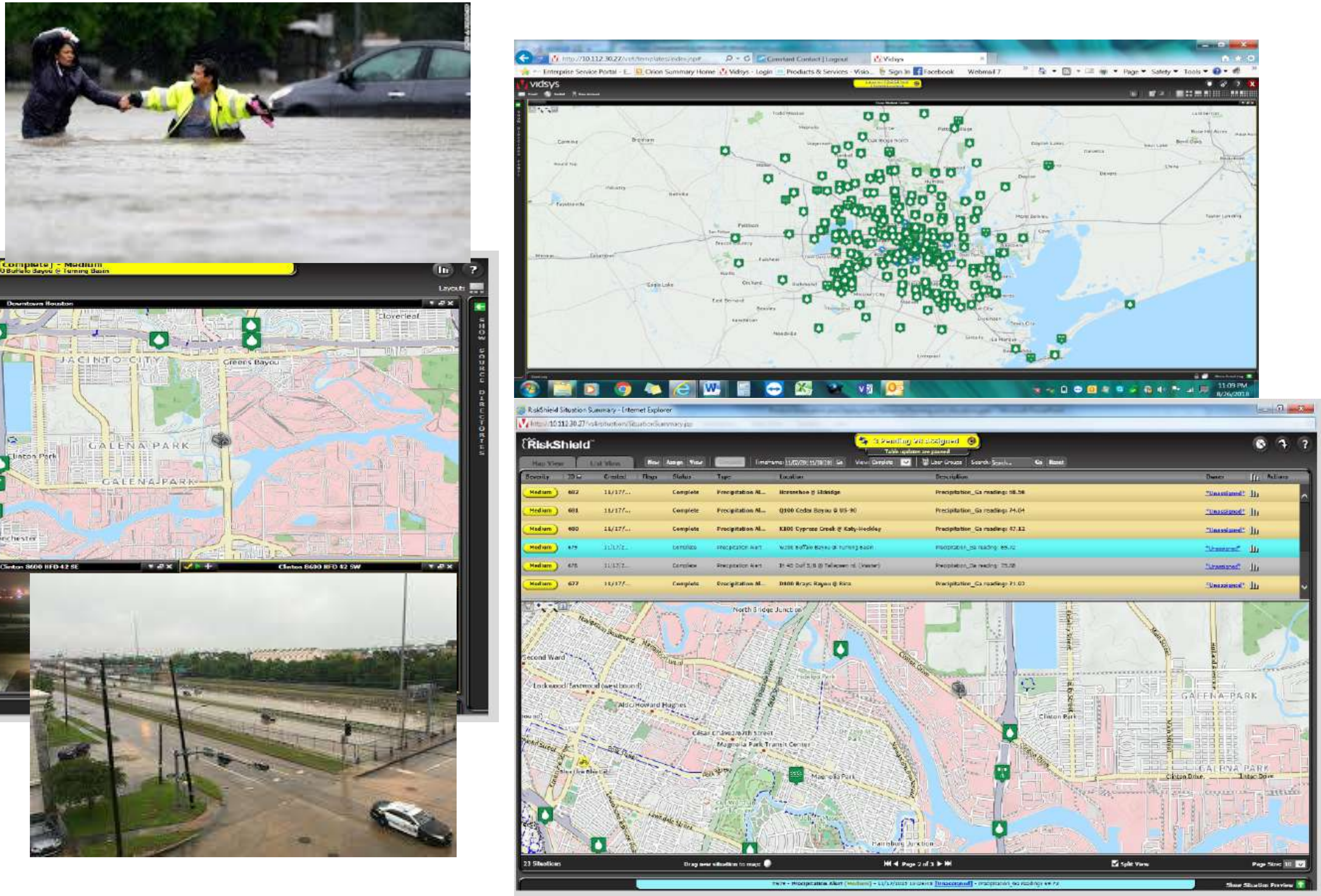
How will our efforts differ from what is already in the marketplace?
How to deliver ROI for Public Safety organizations as it relates to video analytics?

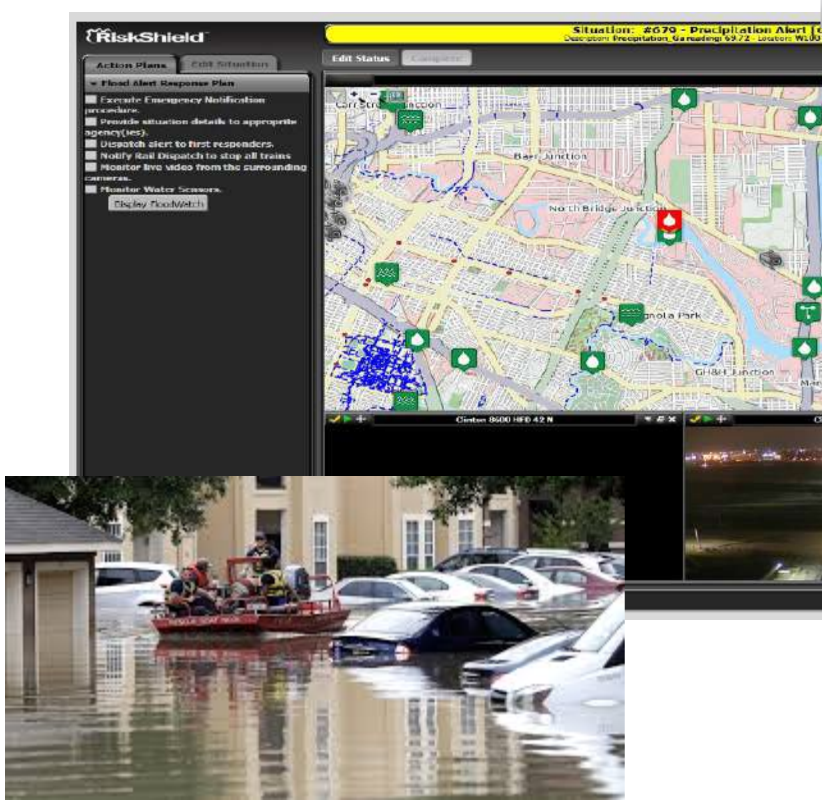
- Mid-sized City situations common to most
- Reduce Cycle Time through Workflow and Automation
  - ✓ Awareness of possible incident beginning
  - ✓ Notification of correct personnel to react
  - ✓ Facilitation of a faster trigger of a standard operating procedure
- Bundling of analytics of varying complexity vs. more complex analytics
- Integration of data sources –use-case relevant, customized format.
- Video / Data Feeds/ Sensor alerts
- OEMs and Fusion Centers as proxy for user groups

Video Analytics Workflow Automation Ecosystem



Use of Flood Use Case & Flood Sensor Integration as a building block to develop analytics automation

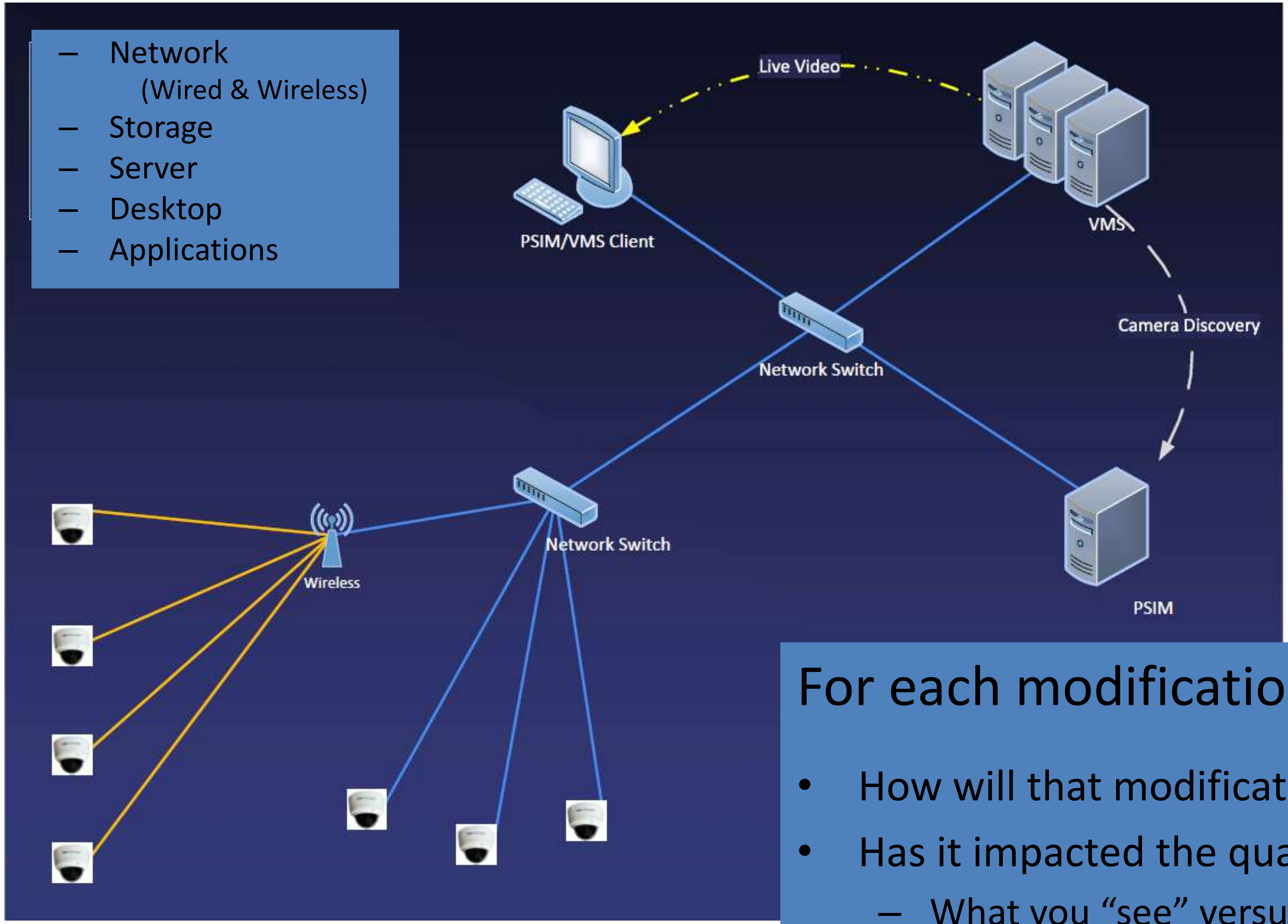




Evolutionary Use of NIBRS Crime Codes, GIS Mapping, PSIM and Analytics to aid in the Criminal Analysis Process and Incident Detection

The Video Microcosm & Impacts on Quality

- Network
  (Wired & Wireless)
- Storage
- Server
- Desktop
- Applications



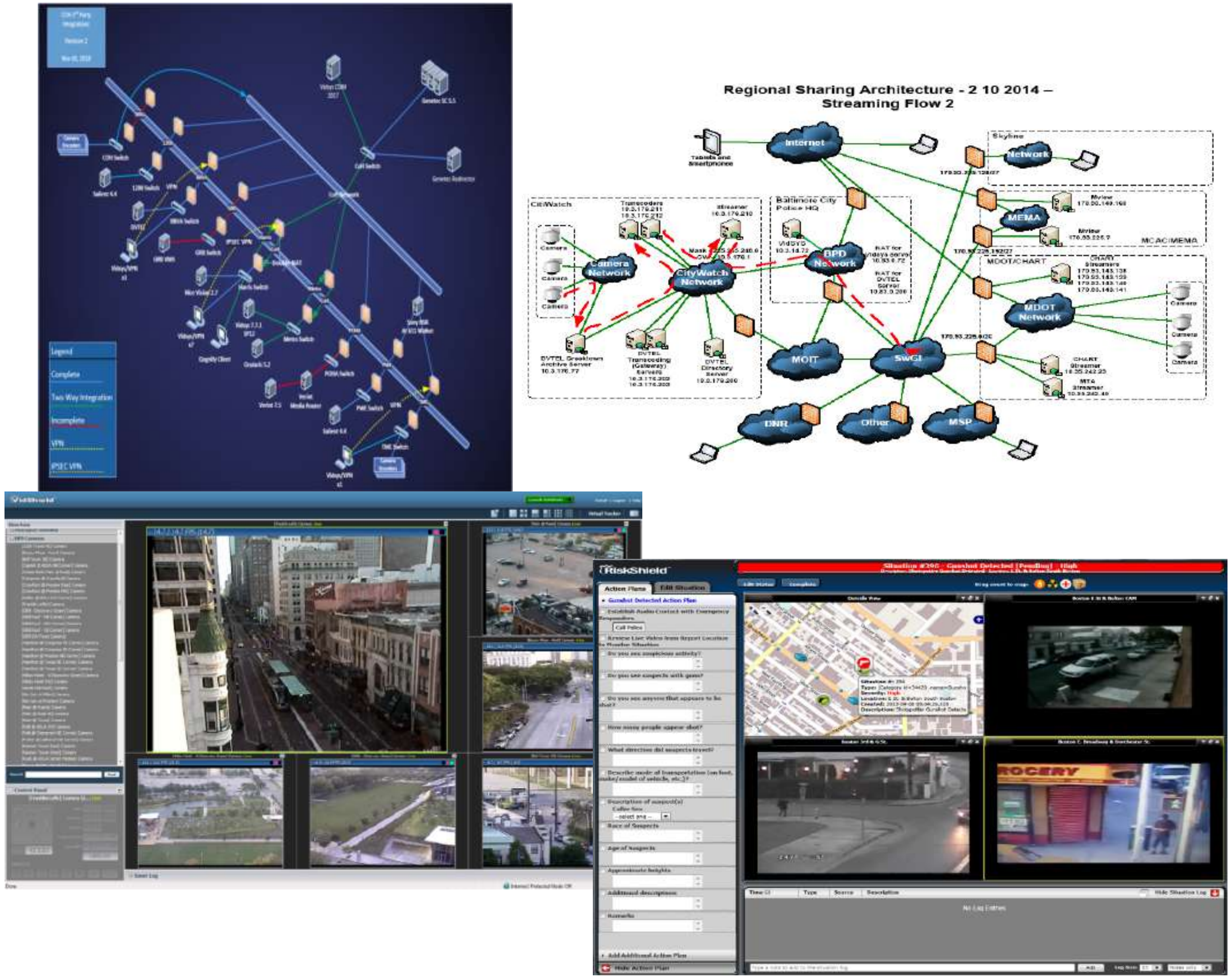
Infrastructure from many vendors

- Genetec VMS with Vidsys PSIM
- Cisco wired network with 3rd party lease lines
- Cambium, Siklu, Radwin, Bridgewave
- VMware / HP server environment
- Dell Compellent Storage
- Veeam / Exagrid Backup

For each modification made to a system component:

- How will that modification impact my Microcosm?
- Has it impacted the quality of video and analytics results?
  - What you “see” versus recommended hardware and system specs

Public Safety Video System Network Architectures are Complex and of Varying Design



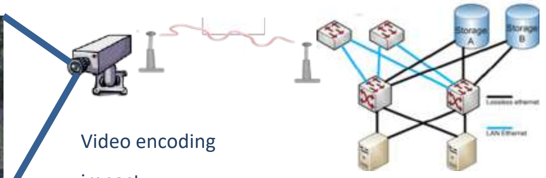
Houston and Baltimore Lessons Learned

- Camera Configuration:
  - If cameras are set to high resolution and frame rate while bandwidth is capped very low, the compression of video required to transmit with the cap in place results in poor video quality.
- Transmission – Wired, Wireless:
  - In the case of cameras using wireless connectivity, the configuration of the radios, cameras and fiber network must be configured correctly to avoid contention and loss of video data packets.
- Automatic Touring vs. Fixed Cameras:
  - Cameras on continuous tour provide a larger field of view but reduces video compression efficiency and requires a higher bit-rate for transmission.
  - Without adequate bandwidth, the video quality is further reduced and visibly affected.

Challenges for Video Analytic based Alerting

- Infrastructure variabilities and its impact on video quality (visual vs. automated video analytics)

UCLA Courtyard, Amer *et al., 2012*



Video encoding impact

Communication Channel & Network Infrastructure impact:
- Packet loss
- Network Congestion etc.

- Video analytic performance variabilities due to video quality

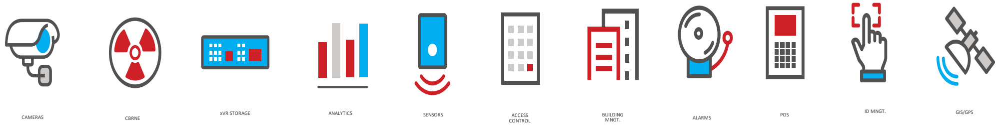
Vandalism
Graffiti
Vandalism
…

Unusual Motion
Stalled vehicle
Wrong direction
…

Crowd Analysis
Crowd dispersal
Crowd count
…

- Enabling video analytics for more accessible/actionable response

RULES	GEOSPATIAL	ROUTING	DISPATCH
Filtering Correlation	Spacial Recognition Mapping + Visualization	Quality Service Compression	Command + Control Communications



Common Video Analytic Use-Cases

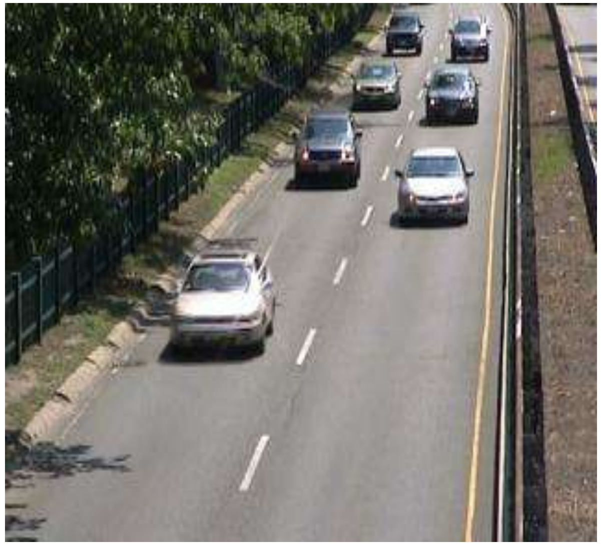
- **Unusual motion:** Automatically identify unusual motion of individual objects. Examples: Car driving in a wrong lane, loitering around cars, etc.
- **Unusual object location/motion:** Automatically identify objects in unusual location. Examples: baggage left behind, person on exposed rooftop, etc.
- **Camera compromised:** Automatically detect events leading to degradation in surveillance camera functionality. Examples: obstructed cameras, changed camera position, out-of-focus view, etc. (Initial development complete)
- **Unusual change in scene:** Automatically detect large changes in the scene. Examples: Vandalism, graffiti, etc.
- **Crowd density:** Automatically detect abrupt changes in crowd density. Examples: detecting large crowds and gatherings at unusual location, and time.
- **Crowd motion:** Automatically detect unusual crowd movement. Examples: crowd panic, riot, etc.
- …

Video Analytic Tasks and Algorithms for Common Use-Cases

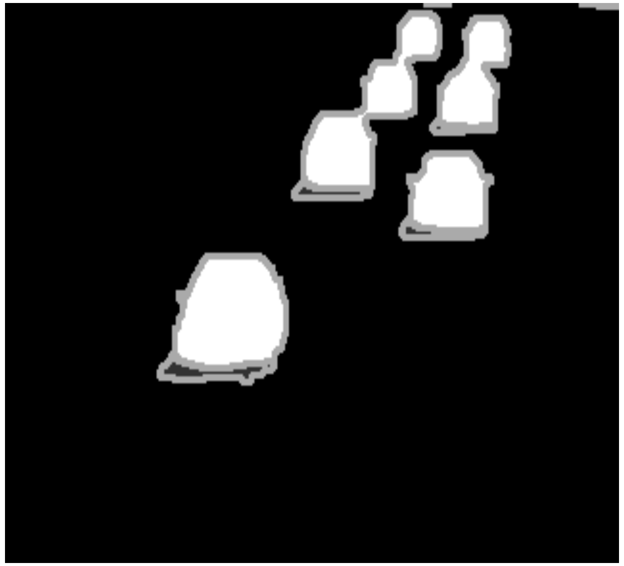
Use-Case	Video Analysis Step 1	Video Analysis Step 2
Unusual Motion (Perimeter Crossing, Car driving in wrong lane, …)	Task: Identify if there is movement in the scene Algorithm: Background Modeling and Subtraction	Task: Detection presence of known/unknown object type within region of detection motion Algorithm: Object Detection
Unusual Object Motion/Location (Abnormal object interactions, …)	Task: Identify objects and their motion in the scene Algorithm: Object Detection and Motion Computation	Task: Detect dominant motion patterns Algorithm: Clustering
Camera Compromised (Performance management, Automated logging/reporting of camera failures, …)	Task: Identify cameras that are not functioning within normal operating parameters without requiring manual review Algorithm: Image Comparison	Task: Detect the type of camera failure so appropriate requirements for repair can be identified Algorithm: Classification
…		

- Many tasks require analysis of individual pixels or groups of pixels in images
- Common algorithms involve Background Subtraction and Object Detection

Background Modeling



Background Modeling



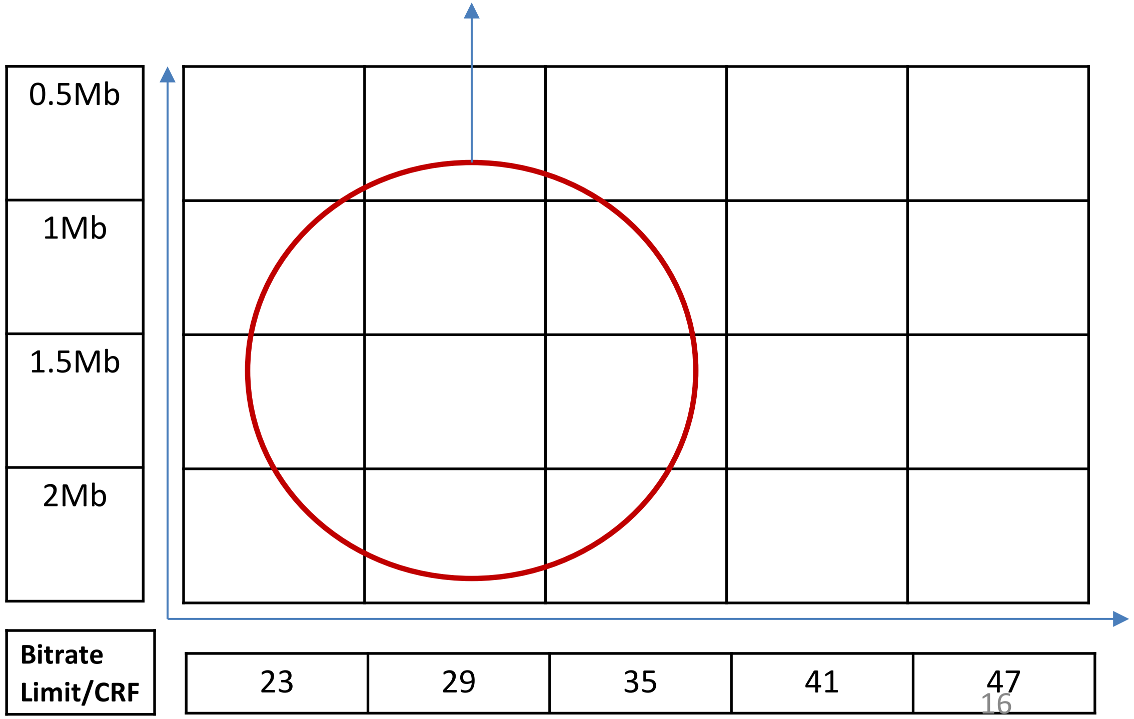
Dataset

Resolution: 1080P
Total: 1 Baseline, 20 Encoded Variants
630 videos (30 videos in original set)
4,987,059 frames (237,479 frames in original set)

What *settings are good* for use with *video analytics*?

(or)

Given a *CRF and upper limit on bitrate*, what is the *loss in performance*?



CRF- Constant rate factor

Quality Metrics for Videos

- The effect of the compression parameters on the video is dependent on the scene elements
  - If the scene has high frequency content, this can result in a loss in spatial content
  - If the scene has fast moving elements, this can result in a loss in spatial content
- It is difficult to establish a relation between compression parameters and the performance of the algorithm

Compression Parameters $\longleftrightarrow$ Algorithm Performance

Quality Metric for Videos

- We introduce a third component,
  - A quality metric that is more indicative of the performance of the algorithm.
  - Now we establish a two-fold relationship to predict performance

Compression Parameters ⟷ Video Quality ⟷ Algorithm Performance

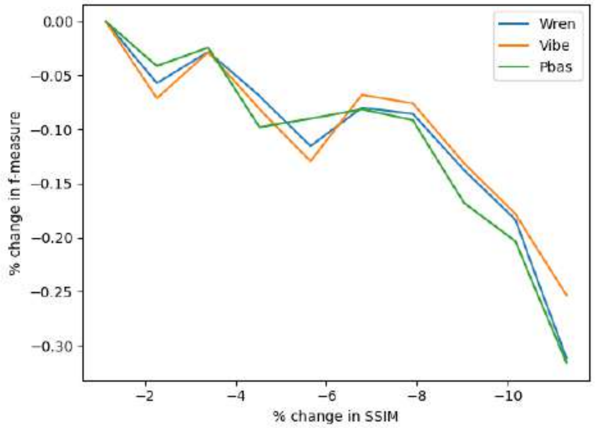
Given a *CRF and upper limit on bitrate*, what is the *loss in performance*?

(instead)

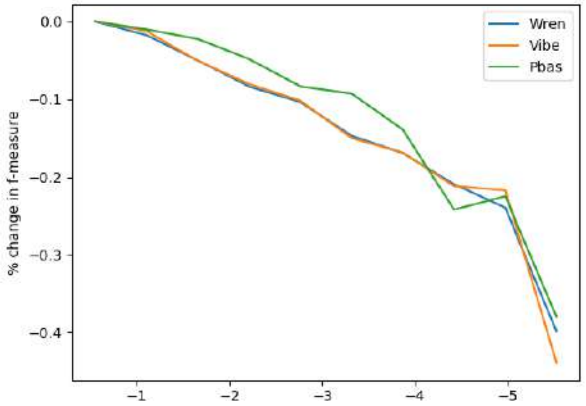
Given a *CRF, upper limit on bitrate*, and some loss in video quality, what is *loss in performance*

Relation between Quality and Performance

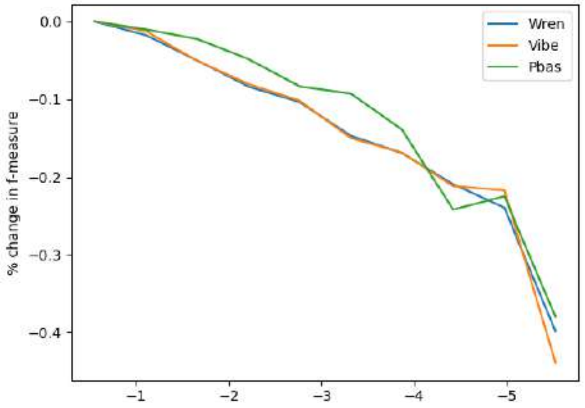
SSIM

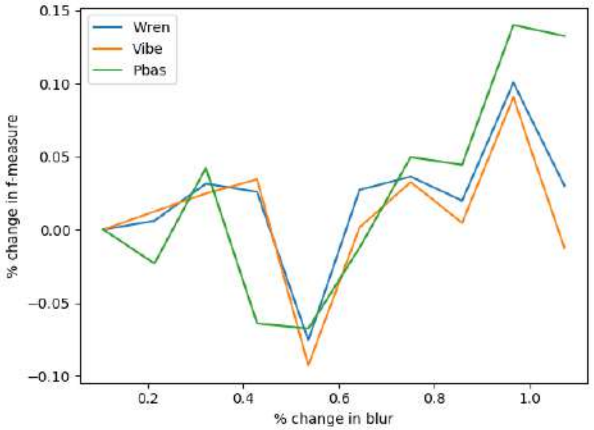


PSNR

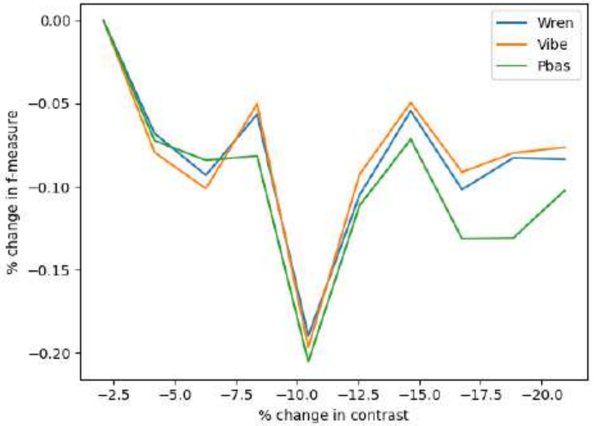


Blockiness

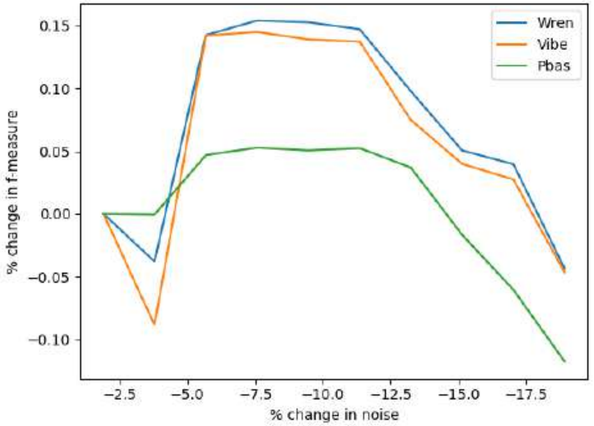




Blur



Contrast



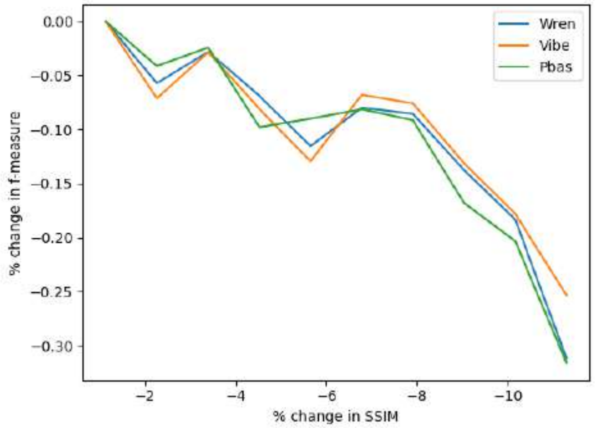
Noise

X-Axis: change in quality metric,
Y-Axis: Change in performance (BG)

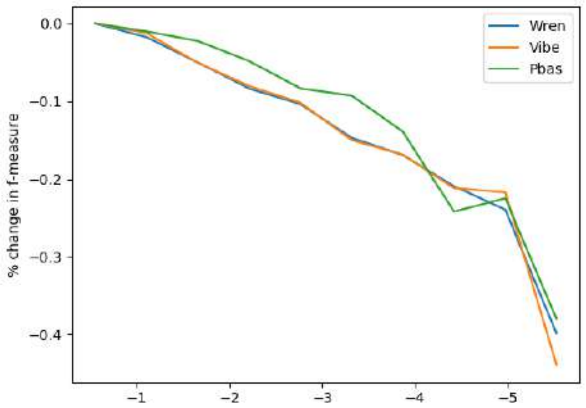
X-Axis: change in quality metric,
Y-Axis: Change in performance (BG)

Relation between Quality and Performance

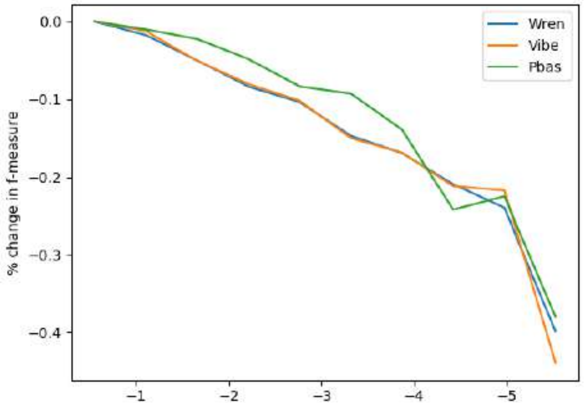
SSIM

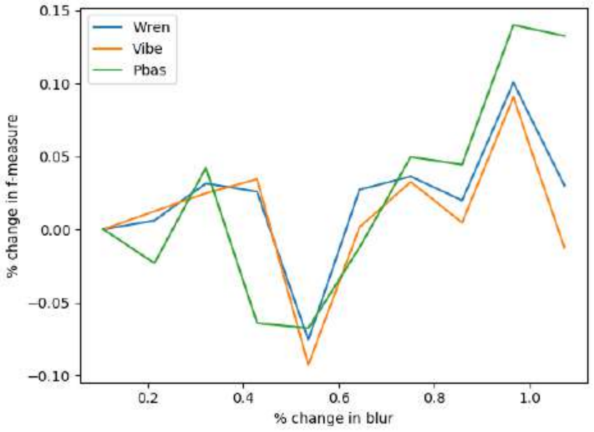


PSNR

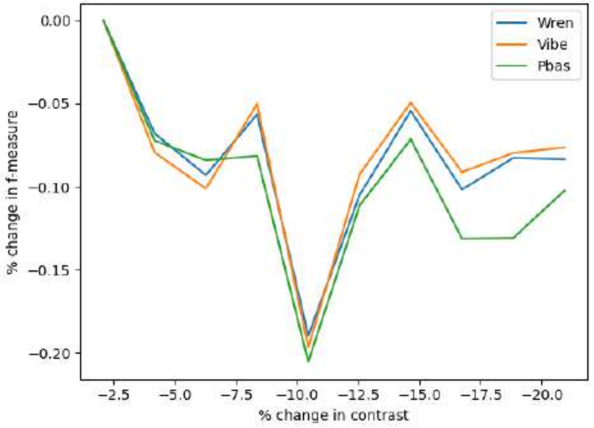


Blockiness

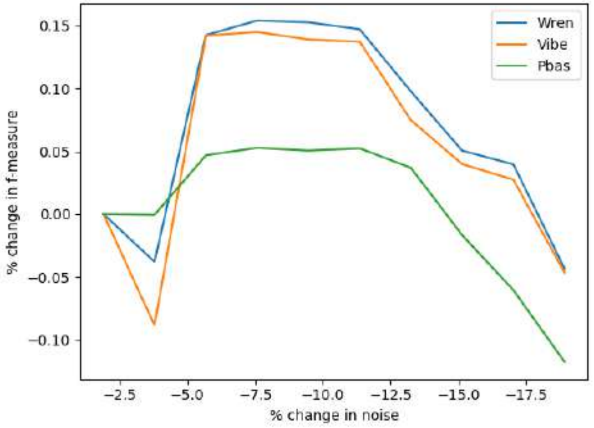




Blur



Contrast

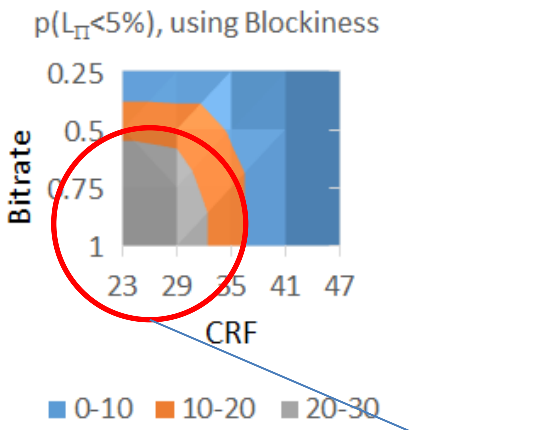


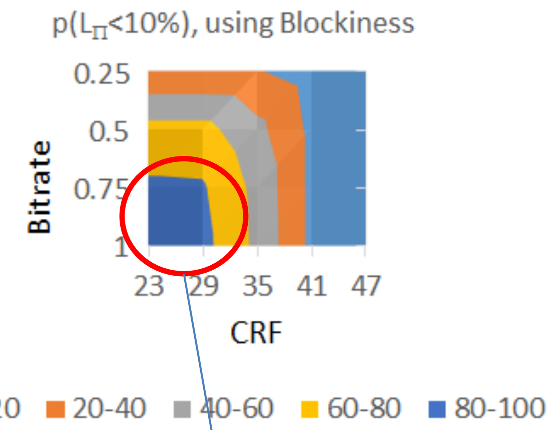
Noise

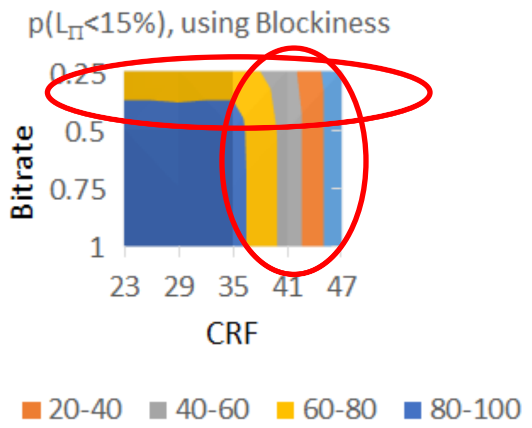
X-Axis: change in quality metric,
Y-Axis: Change in performance (BG)

PSNR and Blockiness show a direct relation with performance

Results



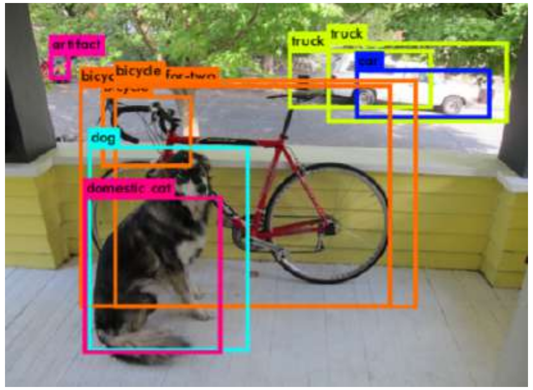




1. Using compression value (23-35, 2MB-1MB), there is an 70-80% chance that the loss in performance will be greater than 5%
2. However, at lower compression values (23, 29; 2Mb,1.5MB), there is an 80-100% change that the loss is bounded by 10%
3. Using compression values (> 35; < 1MB), there is high probability that the loss in performance in going to be greater than 15%

Object Detection

Object Detection



Two-Stage Detectors
Faster R-CNN (Ren *et al.* [1])
R-FCN (Dai *et al.* [2])
Mask R-CNN (He *et al.* [3])

Single-Stage Detectors
SSD (Liu *et al.* [4])
YOLO (Redmon and Farhadi [5])
RetinaNet (Lin *et al.* [6])

[1] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS.
[2] Dai, J., Li, Y., He, K., and Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. In NIPS.
[3] He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask r-cnn. In ICCV.
[4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In ECCV.
[5] Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
[6] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2017b). Focal loss for dense object detection. In ICCV.

Performance Drop

Can these state-of-the-art object detectors make reliable predictions when they encounter compression artifacts in their inputs?

	Bitrate 2Mb/s				Bitrate 1.5 Mb/s				Bitrate 1Mb/s			
	CRF-29	CRF-35	CRF-41	CRF-47	CRF-29	CRF-35	CRF-41	CRF-47	CRF-29	CRF-35	CRF-41	CRF-47
Faster R-CNN	31.5%	38.2%	54.0%	78.2%	33.3%	38.3%	54.2%	78.2%	38.7%	41.3%	54.2%	78.4%
SSD512	**16.8%**	25.5%	42.2%	69.3%	19.7%	25.4%	42.4%	69.5%	23.7%	27.4%	43.0%	70.2%
YOLOv3	17.9%	**22.6%**	**33.9%**	**55.4%**	**19.5%**	**23.0%**	**34.0%**	**55.4%**	**23.0%**	**24.6%**	**33.9%**	**55.6%**
RetinaNet	21.8%	29.1%	49.0%	77.7%	24.2%	29.7%	49.1%	77.8%	29.3%	32.8%	48.9%	78.1%

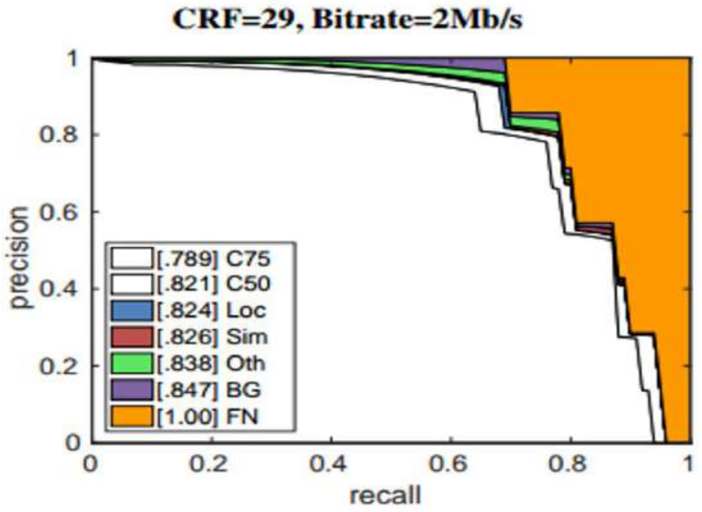
- All detectors are very sensitive to compression artifacts
- They make unreliable predictions when they encounter quality distortions in their inputs due to an inability to generalize from their sharp training sets

Analysis of False Positives

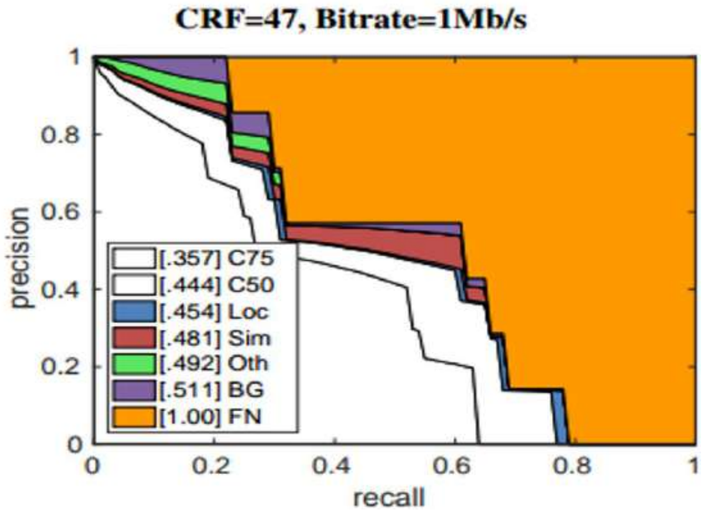
How much of the performance degradation is due to compression artifacts?

There are different types of false positives [7]:

- **Localization error (Loc)**
- **Confusion with similar objects (Sim)**: We consider two categories to be semantically similar if they are both within one of these sets: {person}, {car, bicycle, truck, bus}, {handbag, backpack}
- **Confusion with other objects (Oth)**
- **Confusion with background (BG)**



Lowest compression



Highest compression

- Detection errors are dominated by false negatives (detections with low confidence score or missing detections) due to compression artifacts
- At higher compression, the detection performance is reduced by roughly 50%

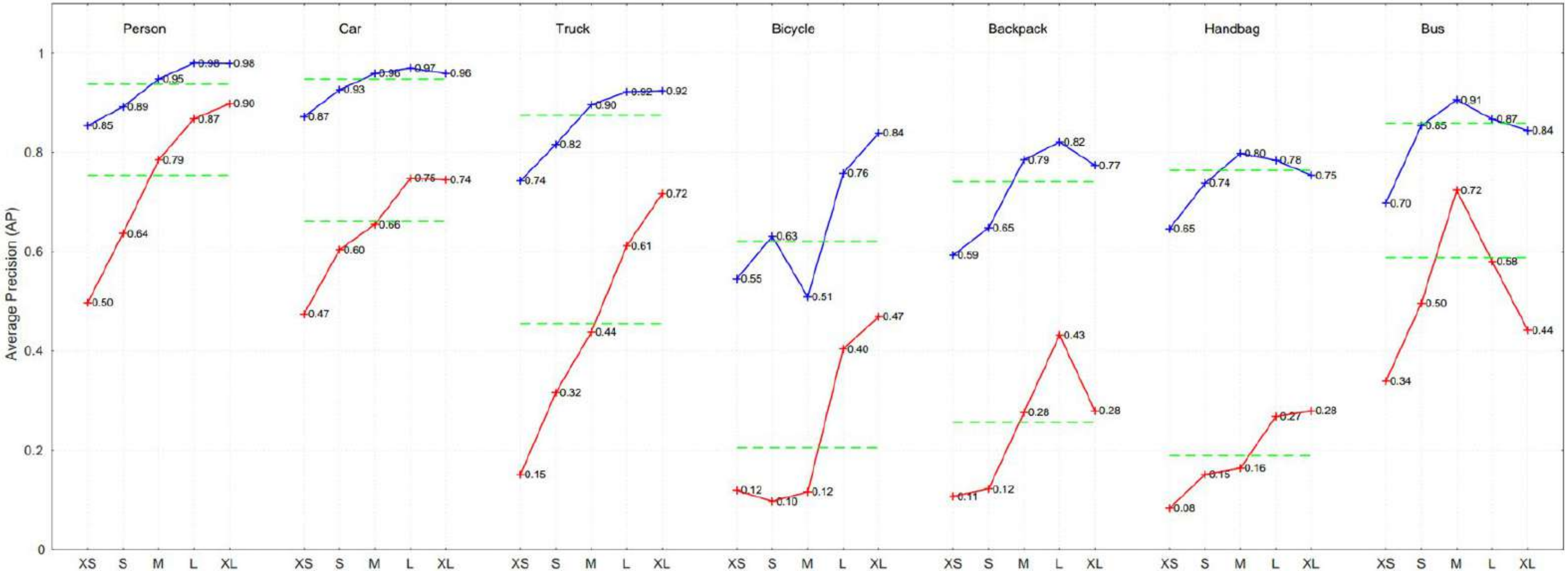
[7] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS.

Sensitivity to Object Size

How sensitive is YOLO to object characteristics under compression artifacts?

- Object size is measured as the pixel area of the bounding box. We assign each object to a size category, depending on the object's percentile size within its object category:
  - **Extra-small** (XS: bottom 10%)
  - **Small** (S: next 20%)
  - **Medium** (M: next 40%)
  - **Large** (L: next 20%)
  - **Extra-large** (XL: next 10%)

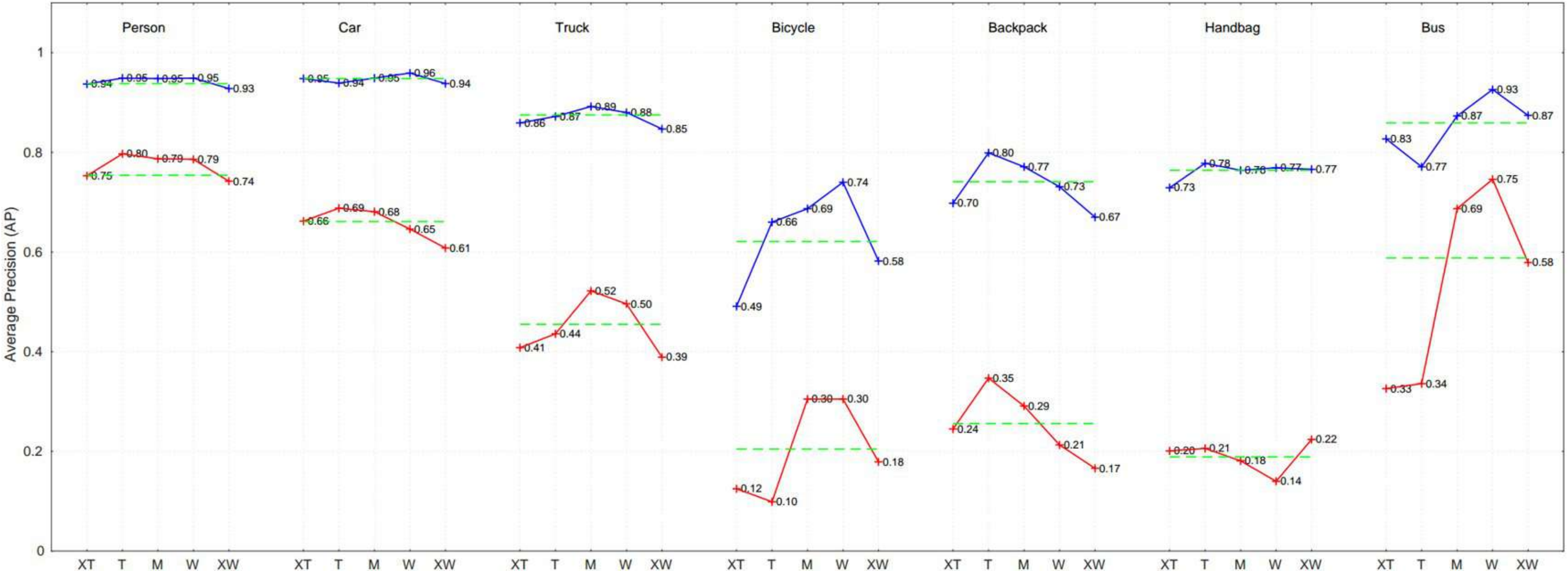
+ AP for lowest compression (CRF=29, Bitrate=2MB/s)
+ AP for highest compression (CRF=47, Bitrate=1Mb/s)
-- Overall AP



Sensitivity to Aspect Ratio

- Aspect ratio is defined as object width divided by object height, computed from the ground-truth bounding box. Similar to object size, objects are categorized into:
  - **Extra-tall** (XT: bottom 10%)
  - **Tall** (T: next 20%)
  - **Medium** (M: next 40%)
  - **Wide** (W: next 20%)
  - **Extra-wide** (XW: next 10%)

+ AP for lowest compression (CRF=29, Bitrate=2MB/s)
+ AP for highest compression (CRF=47, Bitrate=1Mb/s)
-- Overall AP



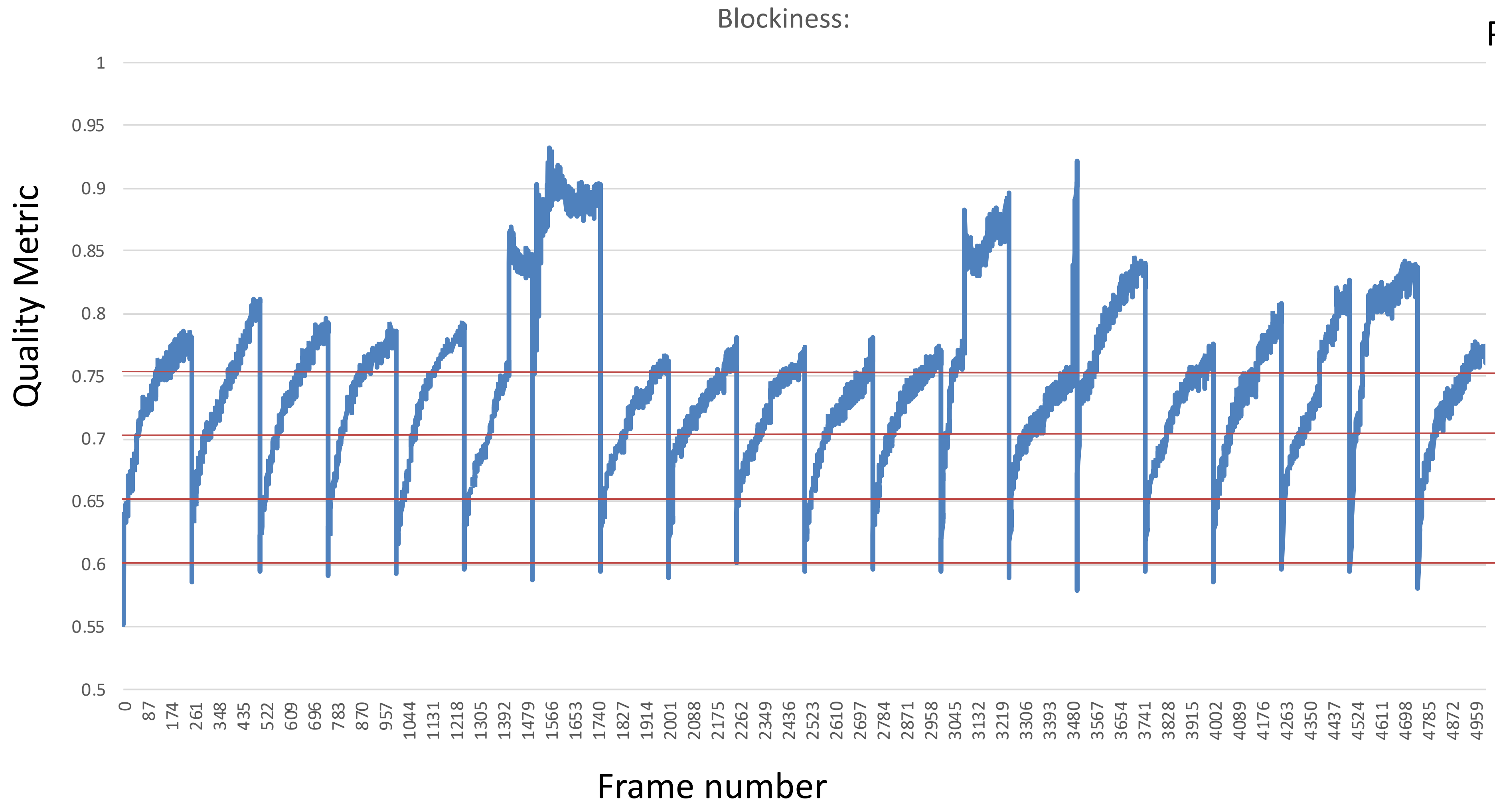
Overall Impact

- Detection errors are dominated by false negatives (detections with low confidence score or missing detections) due to compression artifacts

- The detection of small objects is poor due to high compression as performance drops below 0.15

- YOLO object detector is less sensitive to aspect ratio at both compression levels and tends to recognize objects better at their more natural orientations

Selective Analysis to Improve Object Detection



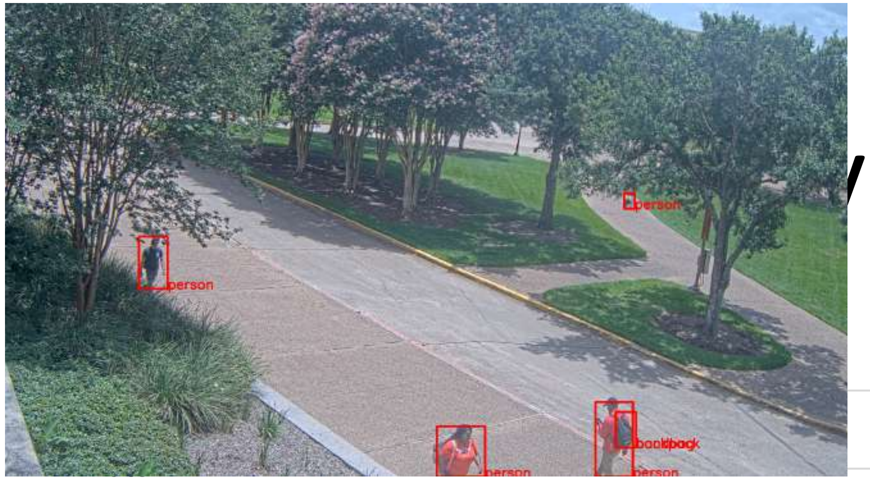
Performance (mAP)	Change
46.06	**+34.1%**
38.35	**+11.6%**
35.94	**+4.6%**
34.25	

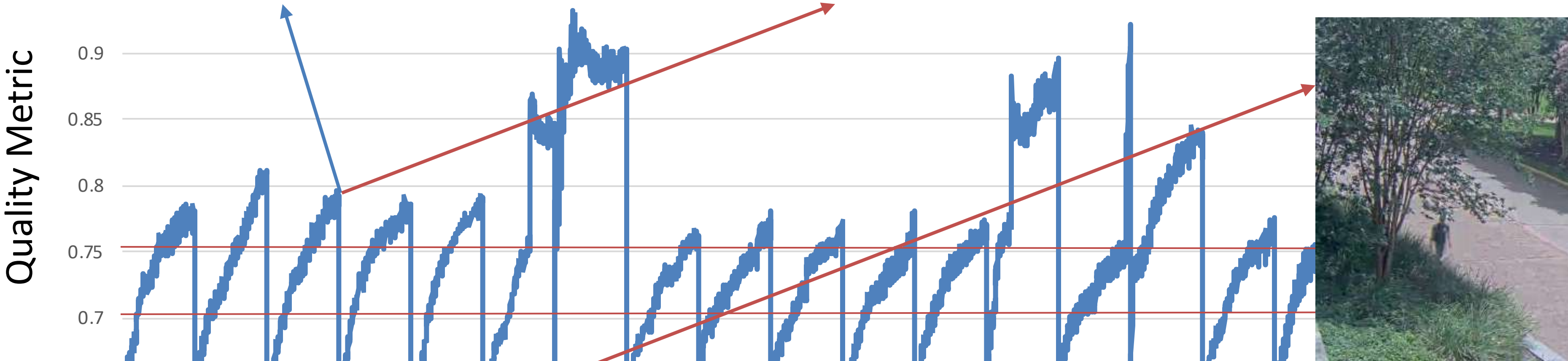
ysis t Detection

Blockir

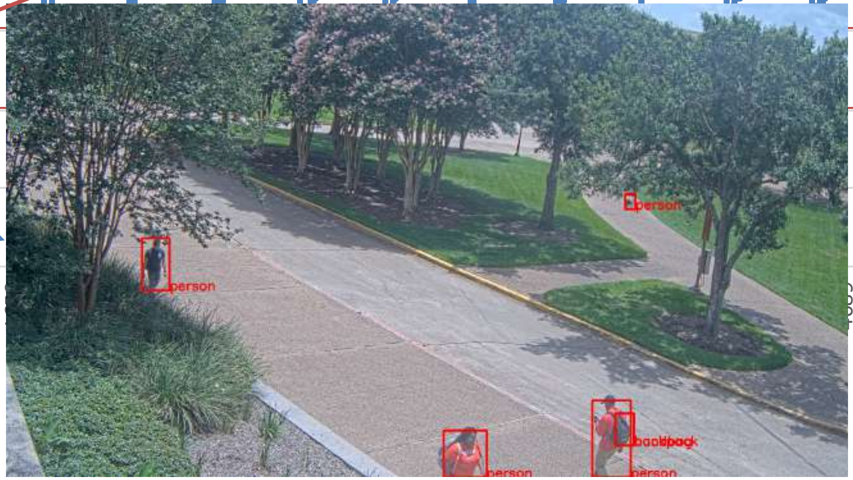
Performance (mAP)

Change





34.25



Use-Case: Camera Tampering

- An unauthorized/unintended alteration in the viewpoint of a surveillance camera is called *tampering*.

Tampering due to natural phenomena

Intentional Tampering

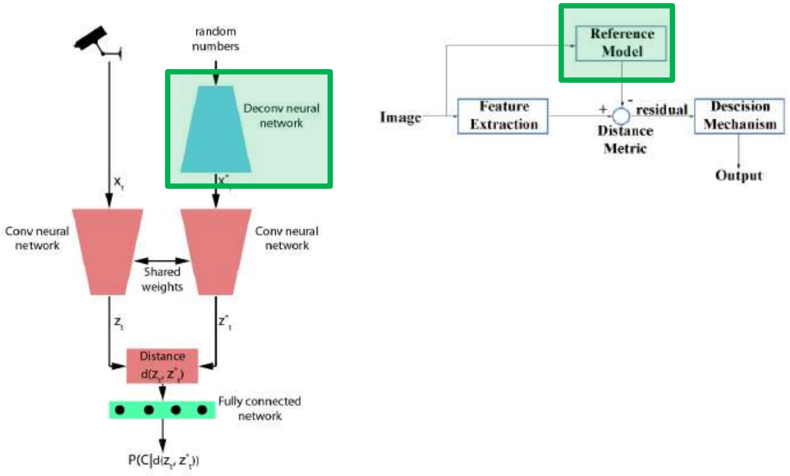
Intentional Tampering

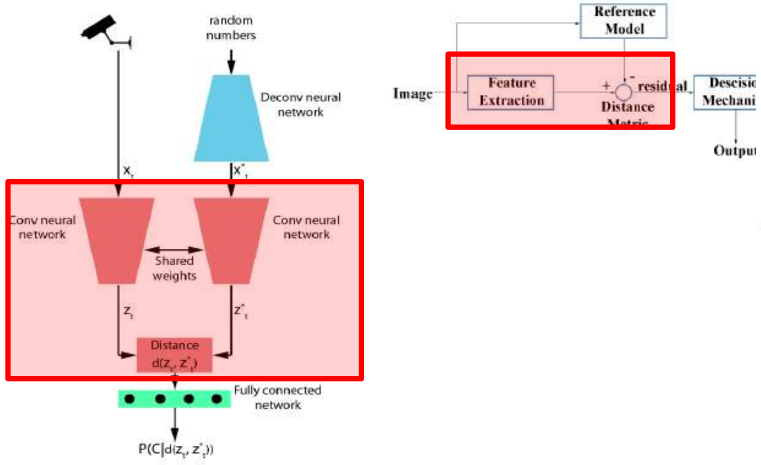
Camera Tampering

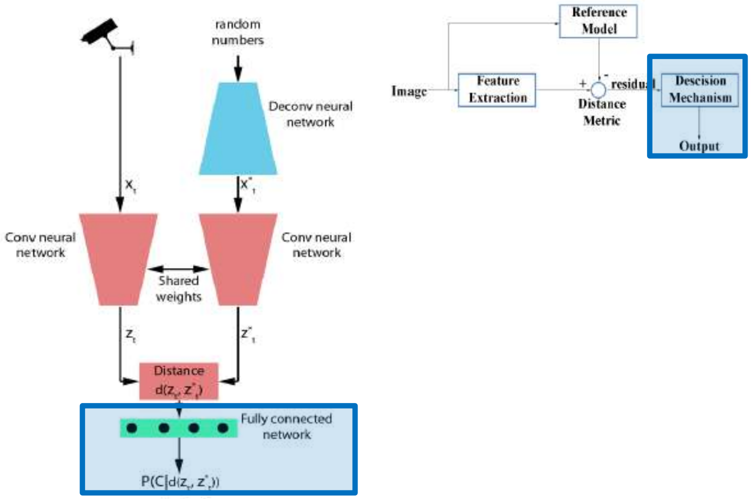
- An unauthorized/unintended alteration in the viewpoint of a surveillance camera is called tampering
- Reporting/Logging of such events is important to continuously ascertain the health of the video system
- Tamper can be classified as

Covered	*Defocused*	*Moved*
1. Lens accumulating dust 2. Perpetrator spray painting a camera	1. Lens losing focus 2. Lens accumulating moisture	1. Camera shifting its view-point 2. Perpetrator changing the view of the camera

Integrated Decision Making for Tamper Classification







Generative Adversarial Network as a Reference Model: A deconvolutional neural network that generates images that represents surveillance camera under normal operating condition

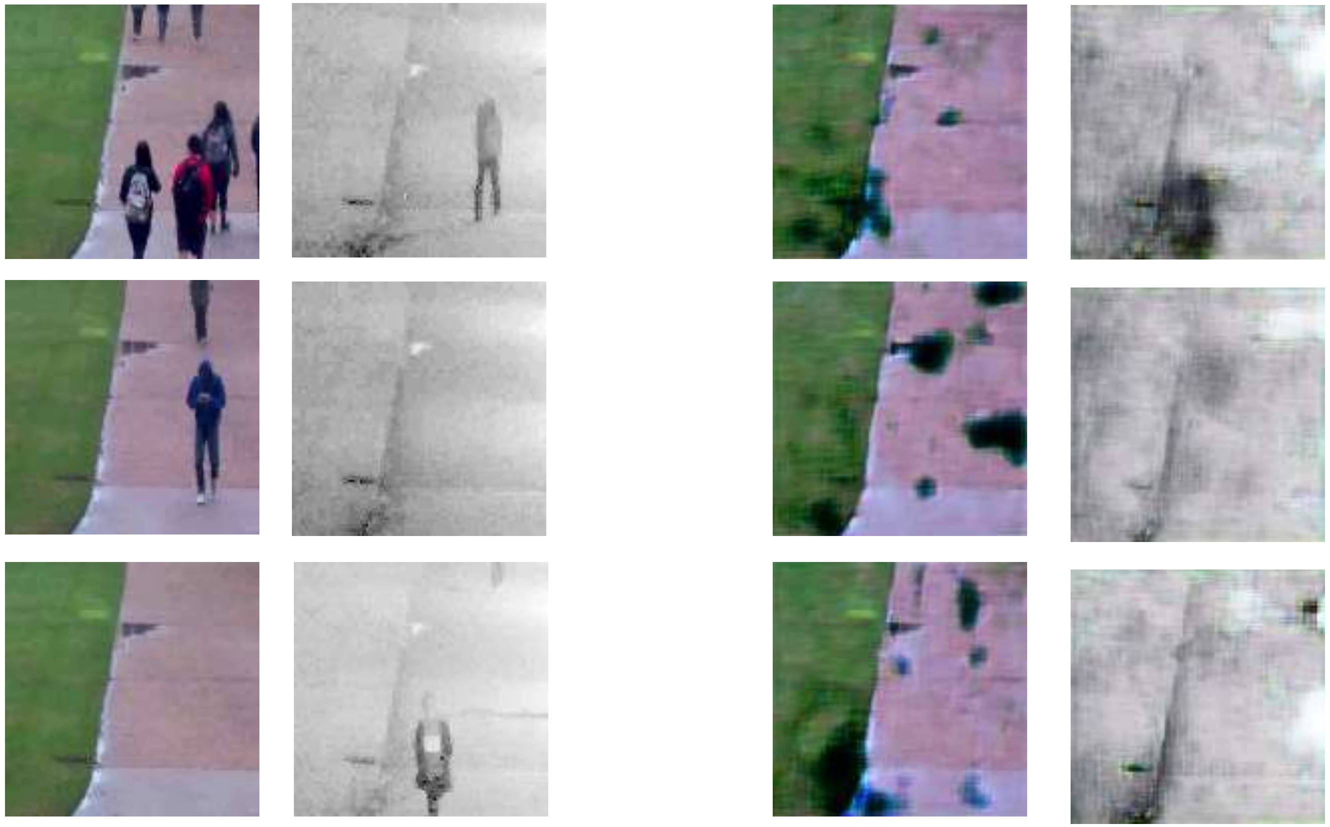
Siamese network as a Feature Extractor: A pair of convolutional neural network (CNN) with shared weights. A feature extractor for the generated and test images

Residual: Distance between the features

Classifier for Detection Mechanism: A fully connected neural network to classify the image as tampered or normal, based on the distance between them.

Generated Images

Original Image   Generated Image

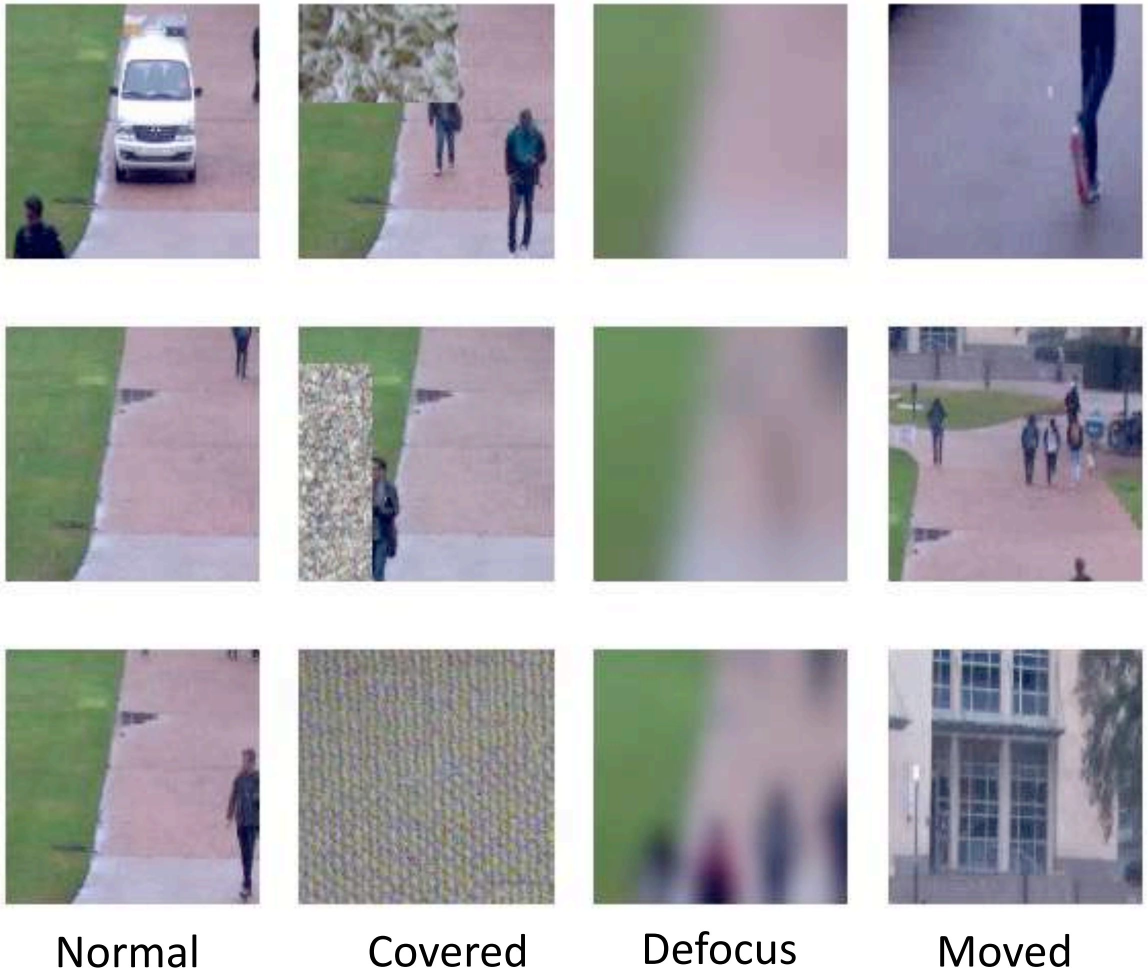


Day   Night   Day   Night

Generated images from two separate GANS for day and night images.

Dataset

- Video from surveillance camera (24 hours at 3 fps)
- Over 250K images (65K are tampered; approximately 21K of each tampering)
- The data are synthetic
  - Tampering is induced using image processing techniques
  - Two parameters are varied to induce tampers: extent, and rate



Results

	Method	TP	FP	TN	FN	TPR	FPR	Acc
Covered	(Lee et al., 2014)	20213	189383	4405	1567	0.928	0.977	0.114
	(Mantini and Shah, 2017)	14704	7720	186068	7076	0.675	0.039	0.931
	(Proposed)	21599	22286	171474	181	**0.991**	0.115	0.895
	(Proposed$_2$)	27889	5233	166712	1980	0.933	**0.030**	**0.964**
Defocus	(Lee et al., 2014)	21502	189383	4405	278	0.987	0.977	0.120
	(Mantini and Shah, 2017)	18929	7720	186068	2851	0.869	0.039	0.950
	(Proposed)	21508	22286	171474	272	**0.987**	0.115	0.895
	(Proposed$_2$)	26892	5233	166712	2449	0.916	**0.030**	**0.961**
Moved	(Lee et al., 2014)	20306	189383	4405	1474	0.932	0.977	0.114
	(Mantini and Shah, 2017)	2966	7720	186068	18814	0.136	0.039	0.876
	(Proposed)	21757	22286	171474	23	**0.998**	0.115	0.896
	(Proposed$_2$)	25567	5233	166712	2373	0.915	**0.030**	**0.961**
Overall	(Lee et al., 2014)	62021	189383	4405	3319	0.949	0.977	0.256
	(Mantini and Shah, 2017)	36599	7720	186068	28741	0.560	0.039	0.859
	(Proposed)	64864	22286	171474	476	**0.992**	0.115	0.912
	(Proposed$_2$)	80348	5233	166712	6802	0.921	**0.030**	**0.953**

Proposed$_2$ = Proposed + Temporal Smoothing

Results

	Method	TP	FP	TN	FN	TPR	FPR	Acc
Covered	(Lee et al., 2014)	20213	189383	4405	1567	0.928	0.977	0.114
	(Mantini and Shah, 2017)	14704	7720	186068	7076	0.675	0.039	0.931
	(Proposed)	21599	22286	171474	181	**0.991**	0.115	0.895
	(Proposed$_2$)	27889	5233	166712	1980	0.933	**0.030**	**0.964**
Defocus	(Lee et al., 2014)	21502	189383	4405	278	0.987	0.977	0.120
	(Mantini and Shah, 2017)	18929	7720	186068	2851	0.869	0.039	0.950
	(Proposed)	21508	22286	171474	272	**0.987**	0.115	0.895
	(Proposed$_2$)	26892	5233	166712	2449	0.916	**0.030**	**0.961**
Moved	(Lee et al., 2014)	20306	189383	4405	1474	0.932	0.977	0.114
	(Mantini and Shah, 2017)	2966	7720	186068	18814	0.136	0.039	0.876
	(Proposed)	21757	22286	171474	23	**0.998**	0.115	0.896
	(Proposed$_2$)	25567	5233	166712	2373	0.915	**0.030**	**0.961**
Overall	(Lee et al., 2014)	62021	189383	4405	3319	0.949	0.977	0.256
	(Mantini and Shah, 2017)	36599	7720	186068	28741	0.560	0.039	0.859
	(Proposed)	64864	22286	171474	476	**0.992**	0.115	0.912
	(Proposed$_2$)	80348	5233	166712	6802	0.921	**0.030**	**0.953**

Overall Accuracy:
Proposed$_2$(95%) > Proposed(91%) > Mantini & Shah(85%) > Lee *et al.* (25%)

Results

	Method	TP	FP	TN	FN	TPR	FPR	Acc
Covered	(Lee et al., 2014)	20213	189383	4405	1567	0.928	0.977	0.114
	(Mantini and Shah, 2017)	14704	7720	186068	7076	0.675	0.039	0.931
	(Proposed)	21599	22286	171474	181	**0.991**	0.115	0.895
	(Proposed$_2$)	27889	5233	166712	1980	0.933	**0.030**	**0.964**
Defocus	(Lee et al., 2014)	21502	189383	4405	278	0.987	0.977	0.120
	(Mantini and Shah, 2017)	18929	7720	186068	2851	0.869	0.039	0.950
	(Proposed)	21508	22286	171474	272	**0.987**	0.115	0.895
	(Proposed$_2$)	26892	5233	166712	2449	0.916	**0.030**	**0.961**
Moved	(Lee et al., 2014)	20306	189383	4405	1474	0.932	0.977	0.114
	(Mantini and Shah, 2017)	2966	7720	186068	18814	0.136	0.039	0.876
	(Proposed)	21757	22286	171474	23	**0.998**	0.115	0.896
	(Proposed$_2$)	25567	5233	166712	2373	0.915	**0.030**	**0.961**
Overall	(Lee et al., 2014)	62021	189383	4405	3319	0.949	0.977	0.256
	(Mantini and Shah, 2017)	36599	7720	186068	28741	0.560	0.039	0.859
	(Proposed)	64864	22286	171474	476	**0.992**	0.115	0.912
	(Proposed$_2$)	80348	5233	166712	6802	0.921	**0.030**	**0.953**

TPR: Proposed is highly capable of detecting tampering
Covered: 99%, Defocus: 98%, Moved: 99%; **Overall: 99%**

Results

	Method	TP	FP	TN	FN	TPR	FPR	Acc
Covered	(Lee et al., 2014)	20213	189383	4405	1567	0.928	0.977	0.114
	(Mantini and Shah, 2017)	14704	7720	186068	7076	0.675	0.039	0.931
	(Proposed)	21599	22286	171474	181	**0.991**	0.115	0.895
	(Proposed_2)	27889	5233	166712	1980	0.933	**0.030**	**0.964**
Defocus	(Lee et al., 2014)	21502	189383	4405	278	0.987	0.977	0.120
	(Mantini and Shah, 2017)	18929	7720	186068	2851	0.869	0.039	0.950
	(Proposed)	21508	22286	171474	272	**0.987**	0.115	0.895
	(Proposed_2)	26892	5233	166712	2449	0.916	**0.030**	**0.961**
Moved	(Lee et al., 2014)	20306	189383	4405	1474	0.932	0.977	0.114
	(Mantini and Shah, 2017)	2966	7720	186068	18814	0.136	0.039	0.876
	(Proposed)	21757	22286	171474	23	**0.998**	0.115	0.896
	(Proposed_2)	25567	5233	166712	2373	0.915	**0.030**	**0.961**
Overall	(Lee et al., 2014)	62021	189383	4405	3319	0.949	0.977	0.256
	(Mantini and Shah, 2017)	36599	7720	186068	28741	0.560	0.039	0.859
	(Proposed)	64864	22286	171474	476	**0.992**	0.115	0.912
	(Proposed_2)	80348	5233	166712	6802	0.921	**0.030**	**0.953**

TPR: Proposed produces more false positives than Mantini & Shah
Because proposed does not use any temporal analysis to suppress false alarms, unlike Mantini& Shah, which uses Kalman fitlering

Results

	Method	TP	FP	TN	FN	TPR	FPR	Acc
Covered	(Lee et al., 2014)	20213	189383	4405	1567	0.928	0.977	0.114
	(Mantini and Shah, 2017)	14704	7720	186068	7076	0.675	0.039	0.931
	(Proposed)	21599	22286	171474	181	**0.991**	0.115	0.895
	(Proposed_2)	27889	5233	166712	1980	0.933	**0.030**	**0.964**
Defocus	(Lee et al., 2014)	21502	189383	4405	278	0.987	0.977	0.120
	(Mantini and Shah, 2017)	18929	7720	186068	2851	0.869	0.039	0.950
	(Proposed)	21508	22286	171474	272	**0.987**	0.115	0.895
	(Proposed_2)	26892	5233	166712	2449	0.916	**0.030**	**0.961**
Moved	(Lee et al., 2014)	20306	189383	4405	1474	0.932	0.977	0.114
	(Mantini and Shah, 2017)	2966	7720	186068	18814	0.136	0.039	0.876
	(Proposed)	21757	22286	171474	23	**0.998**	0.115	0.896
	(Proposed_2)	25567	5233	166712	2373	0.915	**0.030**	**0.961**
Overall	(Lee et al., 2014)	62021	189383	4405	3319	0.949	0.977	0.256
	(Mantini and Shah, 2017)	36599	7720	186068	28741	0.560	0.039	0.859
	(Proposed)	64864	22286	171474	476	**0.992**	0.115	0.912
	(Proposed_2)	80348	5233	166712	6802	0.921	**0.030**	**0.953**

Proposed_2 is augmented with simple temporal analysis method, produces lowest false positives

Results (Daytime)

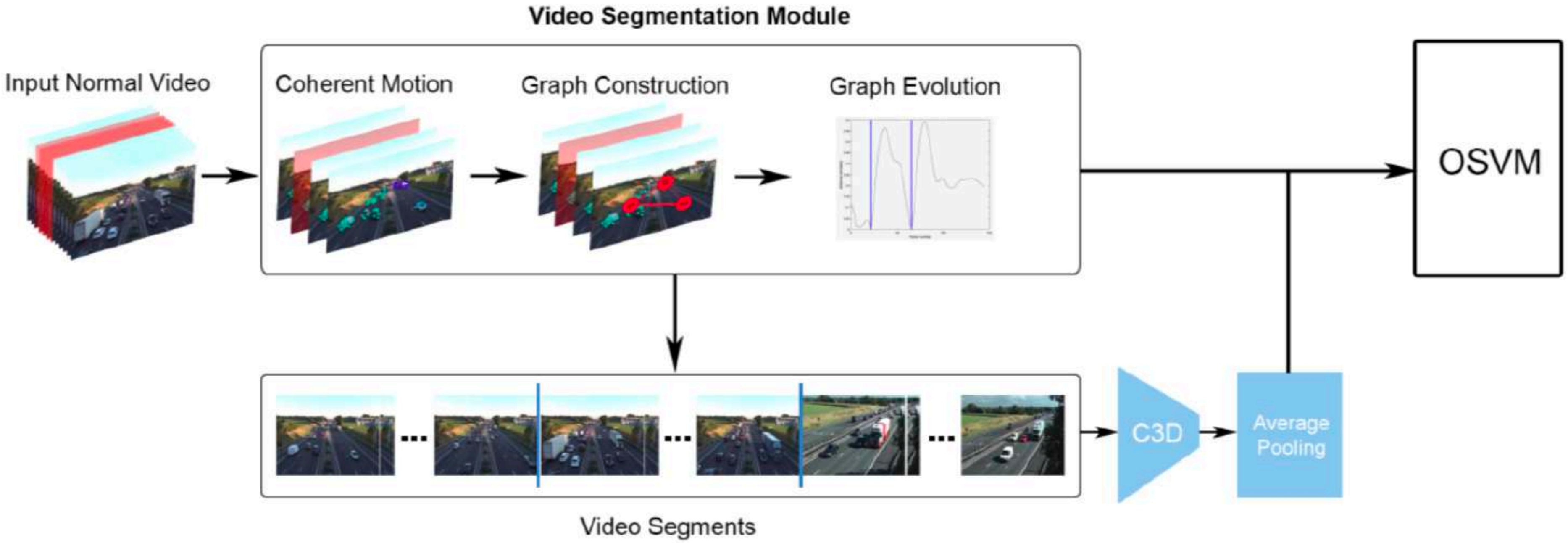
Covered

Defocus

Moved

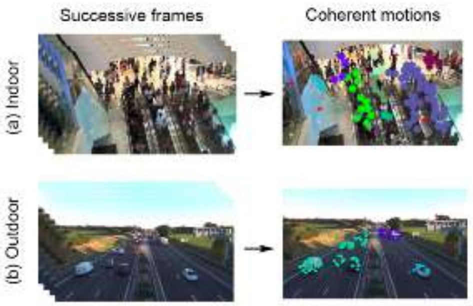
GT – Groundtruth class
PR – Predicted class
Classes - {0: normal; 1: covered; 2: defocus; 3: moved}

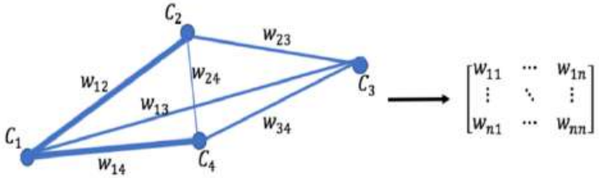
Use-Case: Unusual Object Motion

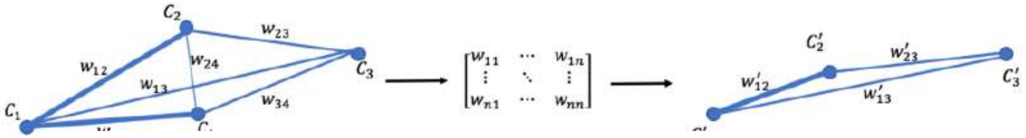


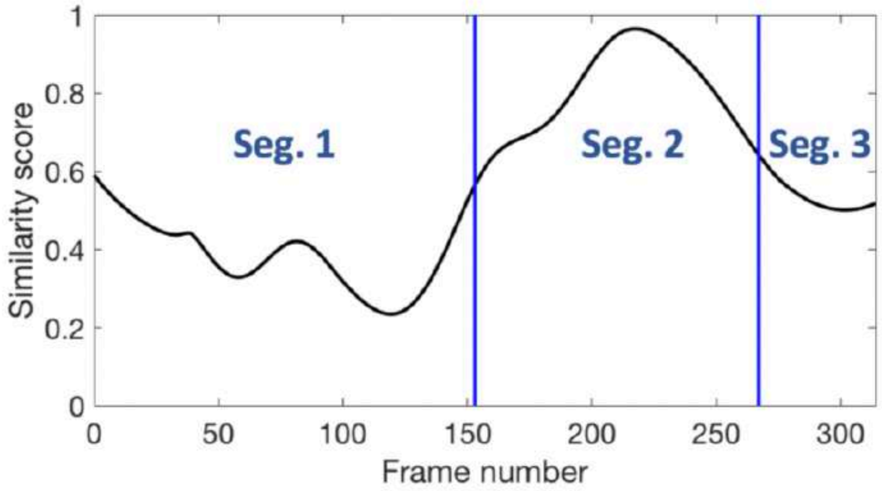
Overview

- Detect Objects/Motion and Track
- Coherent Motion Detection
- Graph Representation of Motion
- Dominant Motion Extraction
- Graph Change Detection





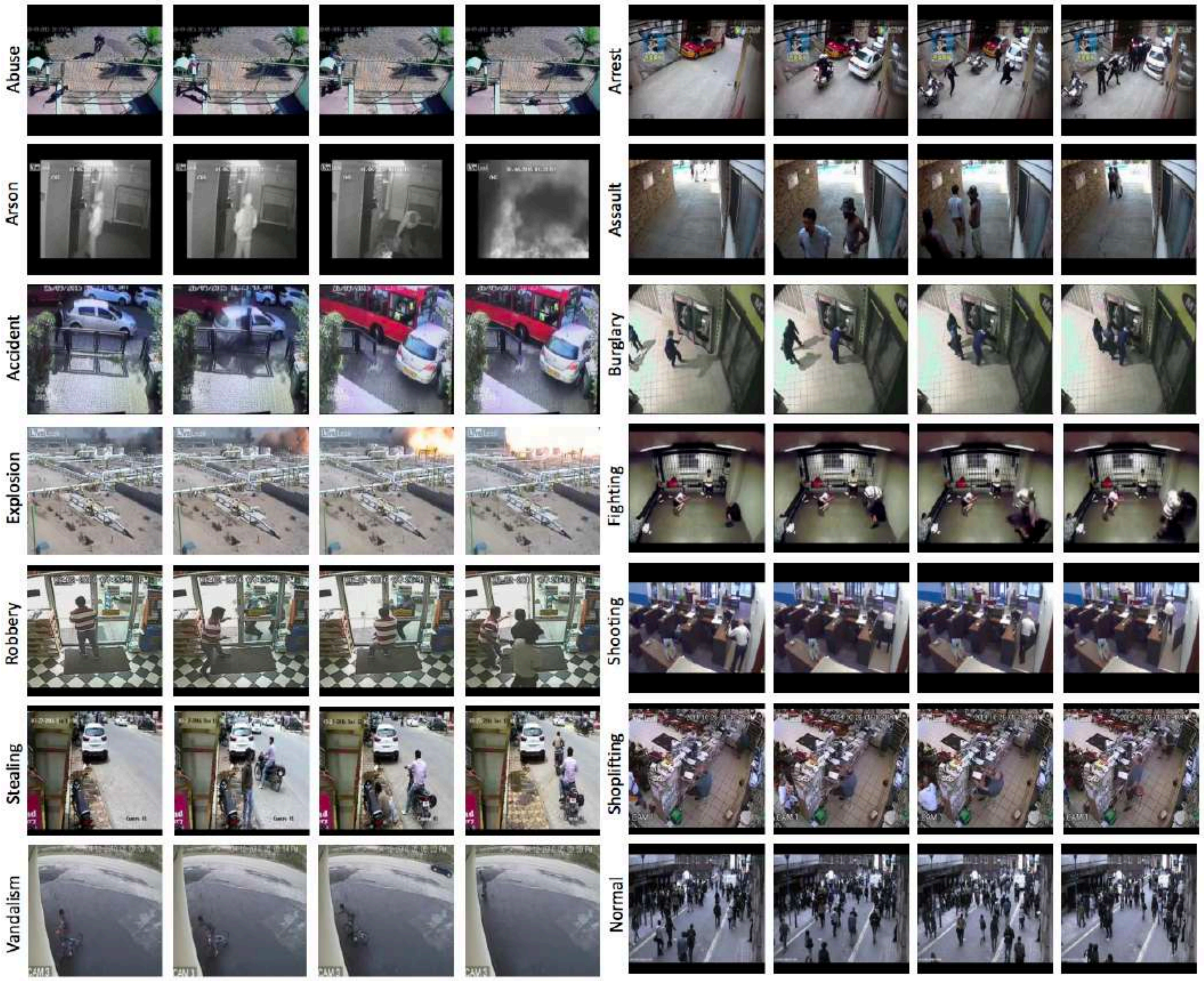




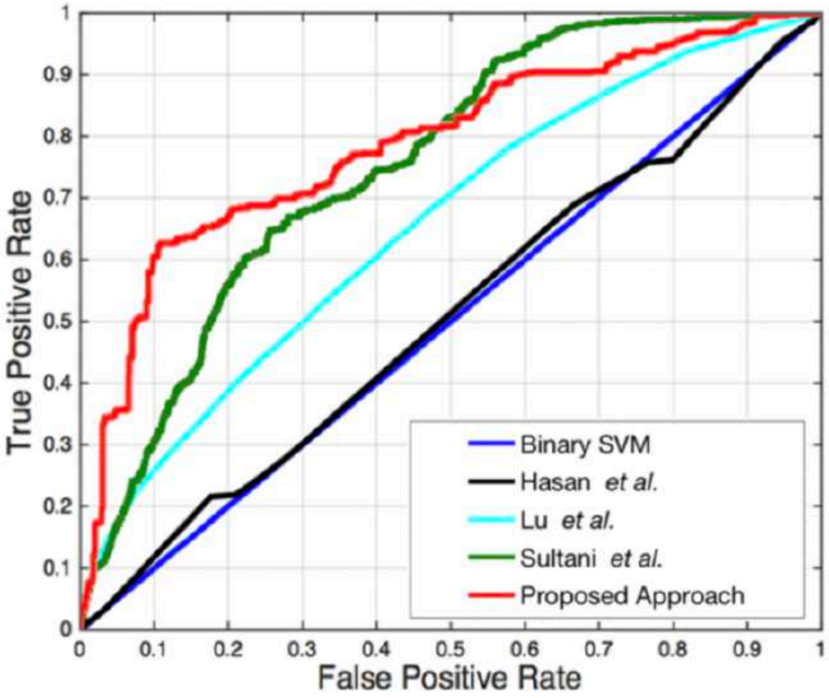
Dataset

- UCF Crime dataset
- 13 real-world anomalies

W. Sultani *el al*. Real-world anomaly detection in surveillance videos. *2018 CVPR*



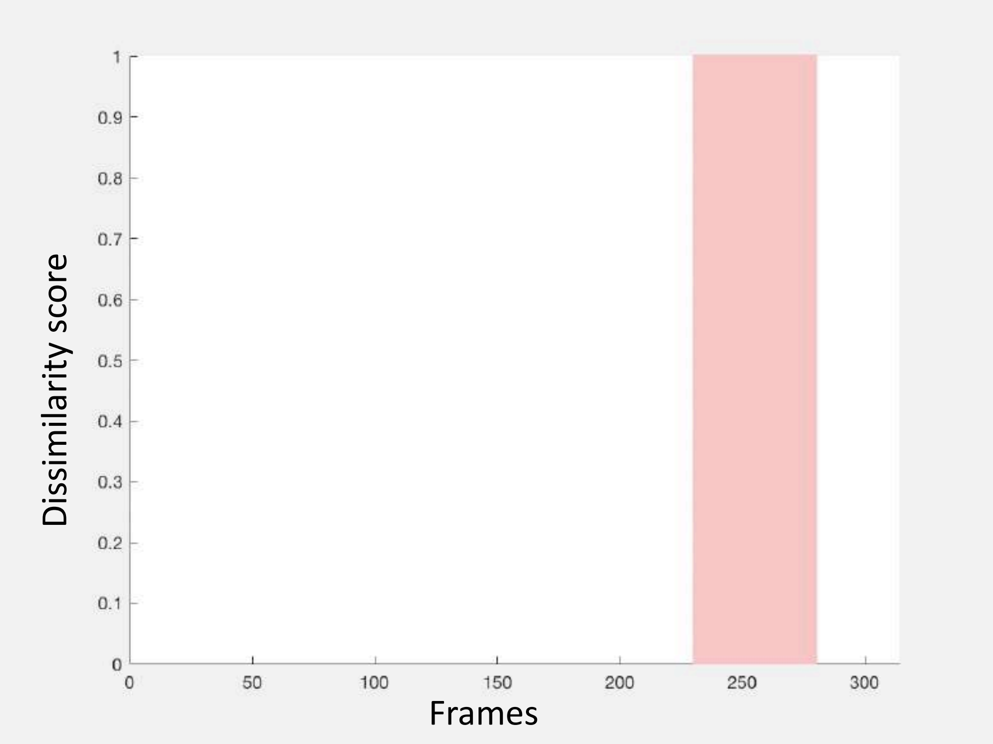
Experimental Results – Anomaly Detection



Method	AUC
Binary SVM	50.0
Hasan *et al.*	50.6
Lu *et al.*	65.5
Sultani *et al.*	75.4
Proposed method	**78.3**

Method	FAR
Hasan *et al.*	27.2
Lu *et al.*	3.1
Sultani *et al.*	1.9
Proposed method	**1.0**

Qualitative Results

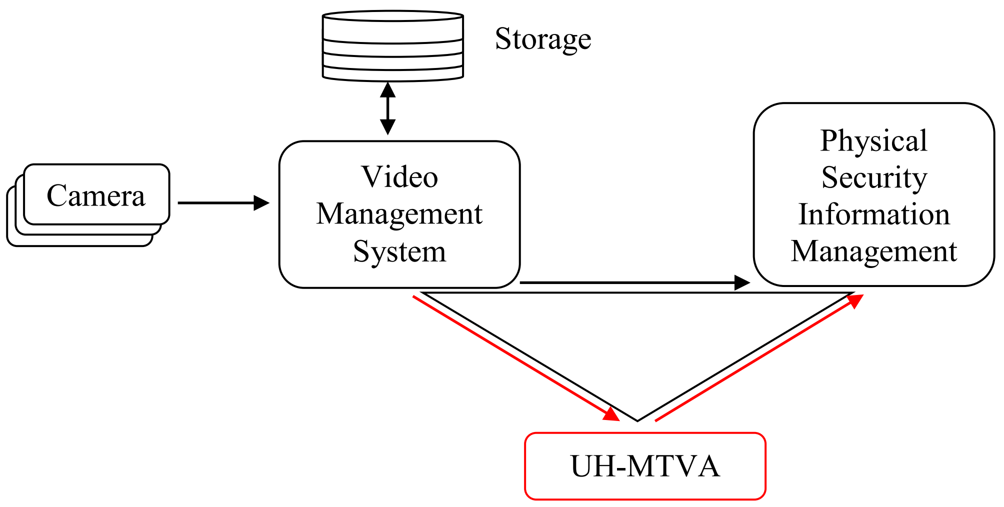


Next Steps

Dataflow

1. Capture the video from the VMS
2. Inject it into the analytics engine developed by the UH team (UH MTVA).
3. Generate various alerts based on events such as compromised camera, unusual motion, etc.
4. Inject alerts into the PSIM

A SOP can be instantiated as response to alerts.



Integrated Workflow Environment at UH

Specification:

1. Genetec as a VMS
2. Vidsys Risk shield as a PSIM
3. Cloud based analytic engine

THANK YOU!

#PSCR2019

Come back for the

Next Session