澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

DEPARTMENT OF
COMPUTER AND INFORMATION SCIENCE

Privacy-sensitive Objects Pixelation for Live Video Streaming

Chi-Man Pun

Introduction

Privacy-sensitive Objects Pixelation for Live Video Streaming

↓

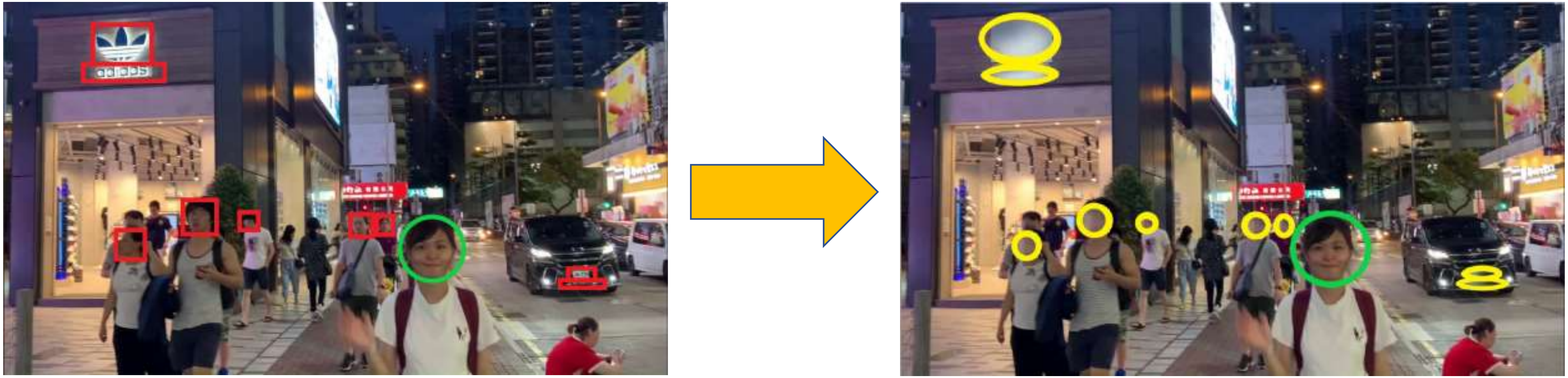
Privacy-sensitive Objects + **Pixelation** + **Live Video Streaming**

Problem Scope (Privacy-sensitive Objects)

- Objects expose/leak privacy-sensitive information
  - Faces, phone number, car plates, erotic images, trademarks, etc.
  - **But also depends on the scenes**.

Problem Scope (Pixelation)

- The process of allocating mosaics on sensitive objects.



Problem Scope (Live Video Streaming)

- Broadcast real-life scenes instantly through phone camera and mobile networks (outdoor live).
- Live platforms: YouTube LIVESTREAM, facebook LIVE Streaming, Instagram LIVE, TikTok
- Live videos vs. Conventional videos:
  - Video resolution after multiple compressions.
  - Single or a few shots.
  - **Camera shakes**.

Problem Scope (Live Video Streaming)

Problem Scope(Direct Solution)

Related Methods

- Only offline trackers
  - YouTube Creator Studio
  - Microsoft Azure
  - Other, mainly Adobe Photoshop

- Tracking-by-Detection Structure is adopted by above all.

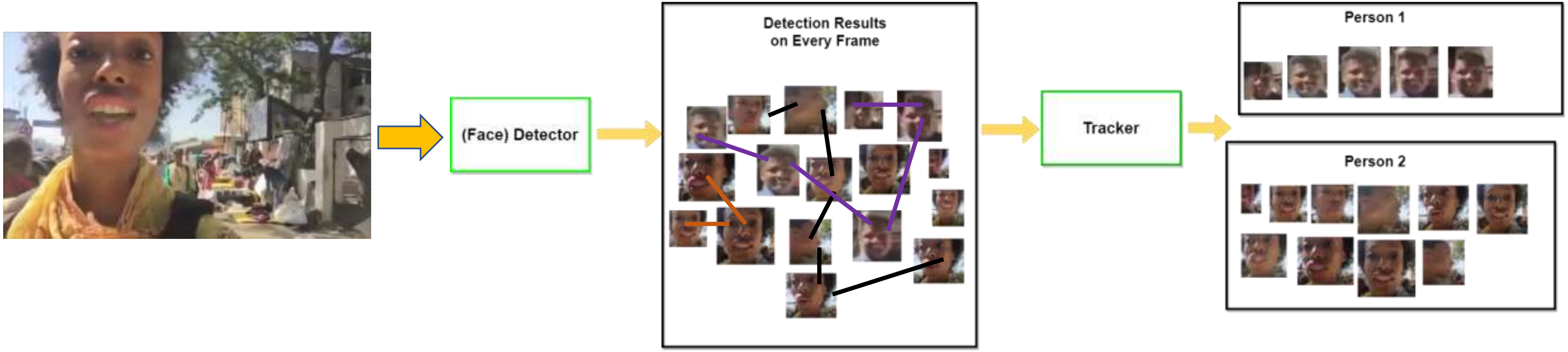
Related Methods (Tracking-by-Detection)

- Tracking heavily relies on detection.
  1. Adaptive Tracker: Particle Filter, Correlation Filter, TDL, etc.
     - Prior: The bounding box (position) of every objects in their **initial** frame.
     - Method: Tracking through gradually adapting filters or online learning.
     - Pros: Finding the most discriminative features for an object.
  2. Tracking-by-Detection: PoI tracker, IoU tracker, Eco, etc.
     - Prior: Objects detection results on **every** frame → tracklets.
     - Method: Associate the detection through learning (appearance, color, motion, etc., features).
     - Pros: More accurate.

Related Methods (Tracking-by-Detection)

- For pixelation tasks:
  - Tacking-by-Detection assembles a privacy-sensitive object detector ahead of a tacker.
  - Contains the detector and tracker independently.
- The privacy-sensitive object detector generate tracklets for tackers.
- Can we migrate the online Tracking-by-Detection algorithms for use?

Related Methods (Tracking-by-Detection)



Related Methods (Privacy-sensitive Objects detector)

- In b tection em se the det

Related Methods (Privacy-sensitive Objects detector)

- Image-based, end-to-end objects detectors are not accurate at all!
  - Lacks comprehension for video context
    - Cannot specify what are the sensitive objects in a certain scene and vice versa
  - Inadequate training samples regarding videos
    - Massive false positives and false negatives under live frames
    - →solve by feeding live video frames as training data?

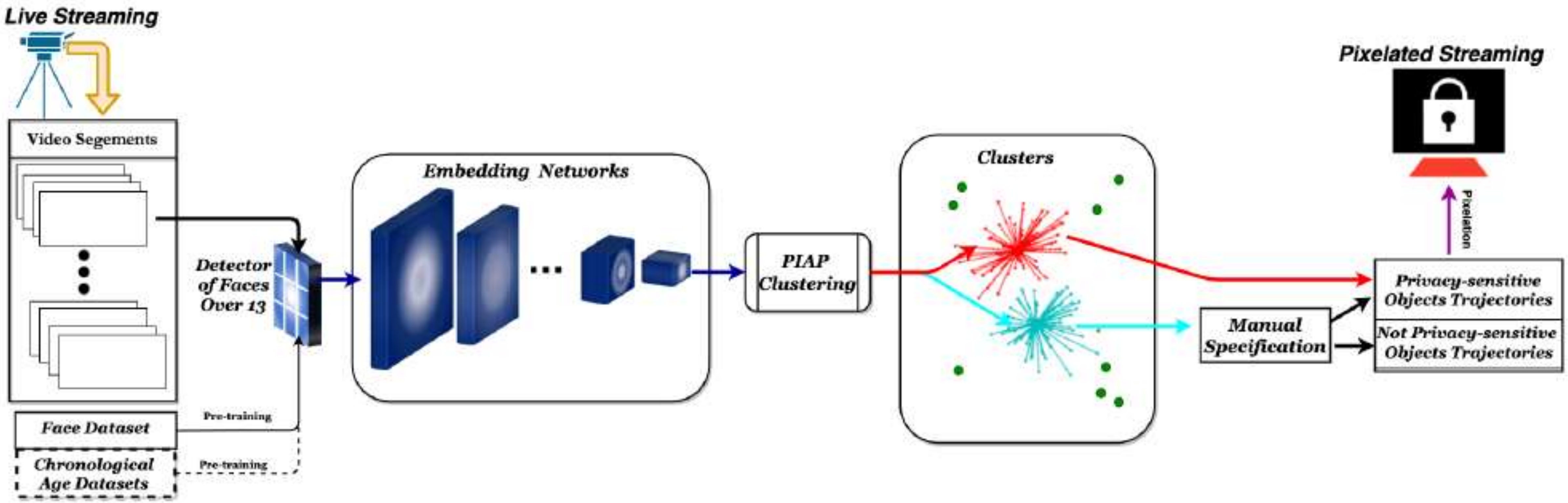
Related Methods (over-pixelation)

- Trackers insist on generating puzzling and unnecessary mosaics for un-identifiable privacy-sensitive objects.

Proposed Method

- So we propose a brand-new framework of Privacy-sensitive Objects Pixelation (PsOP) to address:
  - Detection inaccuracies (both context comprehension and detection errors)
  - Over-pixelation

PsOP (Framework)



PsOP (DPO and IPO)

- Detection Comprehension:
  - Indiscriminating Pixelation Objects
    - Contexts-irrelevant, including erotic images, trademarks, phone numbers or car plate numbers, etc.
    - Pixelation on detection
  - Discriminating Pixelation Objects
    - Context relevant, including faces and texts
    - Pixelation settings shall be discriminately imposed on every instance of DPOs
      - Embedding networks
  - IPO is prioritized if IPO and DPO overlapped
    - Phone number is also a kind of text

PsOP (IPO)

- Detected and then smoothed
  - Relatively easy and more accurate in detection
  - No need for discriminate pixelation → smoothing
- For smoothing, we build a buffer section at the beginning of the live to accumulate some time and enable Gaussian smoothing.

PsOP (DPO)

- Ground on detection → avoid over-pixelation
- False negative in Detection:
  - Lower detection threshold → more false positives
  - Slight smoothing
- False positives in Detection:
  - Embedded into higher level abstraction
  - Positioned Incremental Affinity Propagation (PIAP) Clustering (unsupervised learning strategy)

PsOP (PIAP)

- Affinity Propagation (AP):
  - Fred and Deck, 2008
  - Finding exemplars that suitable for all data nodes
  - **Affinity**: Similarities between data nodes
  - **Propagation**: Message passing between data nodes to reach consensus

$$R(i,k) \leftarrow S(i,k) - \max_{k', s.t. k' \neq k} \{A(i,k') + S(i,k')\}$$

$$A(k,k) \leftarrow \sum_{i', s.t. i' \neq k} \max\{0, R(i',k)\}$$

$$A(i,k) \leftarrow \min\{0, R(k,k) + \sum_{i', i' \notin \{i,k\}} \max\{0, R(i',k)\}\}$$

$$C(i,k) \leftarrow R(i,k) + A(i,k)$$

PsOP (PIAP)

- The problems shall be handled during clustering
  1. High dimensional features after embedding (FC, DBSCAN, AP)
  2. Ill-defined cluster number (AP)
  3. Unbalanced sample size (AP)

  ---

  4. Noise (PAP)
  5. Time-costs (IAP)

- **PIAP inherits the ability of handling 1-3 from AP, and we endow AP with noises-resistance and time-saving merits through positioned affinities and incremental clustering.**

PsOP (PIAP)

Algorithm 1 Positioned Incremental Affinity Propagation

Input: $R_{t-t'}, A_{t-t'}, C_{t-t'}, Z_t$

Output: R_t, A_t, C_t

while (t-t') not contain the end of a live-streaming **do**
  Compute similarity matrix for every z_t^{pq} according to (5).
  if the first video segment of a live-stream **then**
    Assign zeros to all responsibilities and availabilities.
  else
    Assign responsibilities and availabilities for all $z_{t-t'}^{pq}$ by extending $R_{t-t'}$ to R_t, $A_{t-t'}$ to A_t according to equation (8), (9), (10) and (11).
  end if
  Message-passing according to equation (1), (6) and (3) till run δ; then, exclude outliers according to (7).
  Continue message-passing till convergence; compute exemplars and clustering results C_t as equation (4).
end while

Experiments

- Dataset
  - Collected 20 videos from the live platforms with various categories.
  - Dense annotation on the objects.
  - Divided into sub-datasets according to resolution and complexity.

Dataset	Quantity of videos	Category	Resolution	People occurred	Frames	Privacy-sensitive objects Labels*	IPOs Labels*	DPOs Lables*
HS	4	a,b,c,d	720p/1080p	$\gg 2$	4133	27337	6255	21082
LS	8	a,b,c,d	360p/480p	$\gg 2$	4680	23869	3975	19894
LN	4	a,b,c,d	360p/480p	≤ 2	5692	16994	7946	9048
HN	4	a,b,c,d	720p	≤ 2	4867	16061	10805	5256

PsOP (Experiments)

- Parameters
  - Buffering Section
    - User defined setting, log effect on the results, higher the better but will decrease quickly. Not set to a very low value (like 60 frames).
  - Superparameteres
    - Consistent effects cross videos, so we find the optimal value in our test. But not a huge deal if just set to a meaningful value.

PsOP (Experiments)

- Metrics
  1. Borrowed from tracking for basic accuracy and precision evaluation.
     - Sensitive Objects Pixelation Precision: SOPP.

$$SOPP = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t}$$

     - Sensitive Objects Pixelation Accuracy: SOPA.

$$SOPA = 1 - \frac{\sum_t (m_t + fp_t + mm_t)}{\sum_t g_t}$$

  2. Customized for pixelation including over-pixelation and mosaics. drifting
     - Most (consecutively, correctly) Pixelated Frame Length: MP.
     - Over Pixelation Ration: OPR.

$$OPR = \sum_t (fp_t) / \sum_t c_t$$

PsOP (Experiments)

Method	SOPA↑ (*HS*)	SOPA↑ (*LS*)	SOPA↑ (*HN*)	SOPA↑ (*LN*)	SOPP↑ (*HS*)	SOPP↑ (*LS*)	SOPP↑ (*HN*)	SOPP↑ (*LN*)	Entire Dataset MP↑ (frames)	Entire Dataset OPR↓
YouTube [12]	0.45	0.42	0.56	0.49	0.53	0.47	0.77	0.63	238	0.56
Azure [16]	0.43	0.47	0.54	0.53	0.50	0.53	0.70	0.68	203	0.54
KCF [13]	0.35	0.32	0.38	0.31	0.41	0.40	0.44	0.40	113	0.64
ECO [5]	0.27	0.28	0.34	0.31	0.37	0.39	0.41	0.40	148	0.59
PsOP	**0.63**	0.60	**0.71**	**0.66**	**0.80**	0.77	**0.86**	**0.85**	**362**	**0.34**
PsOP with Advanced CNNs*	**0.63**	**0.62**	**0.71**	0.65	**0.80**	**0.79**	**0.86**	**0.85**	349	**0.34**

Advanced CNNs* substitutes the base model with PrimidBox [25]+ArcFace [6], and Tesserat [22]+BERT [7]

PsOP (Experiments)

Dataset	Method	Purity↑	Cluster nubmer (Clustered/Truth)	Time(s)↓
HS	AP*	0.61	59/47	1.82
	PAP	0.83	44/47	1.80
	PIAP	0.83	44/47	0.07
LS	AP*	0.52	117/91	3.05
	PAP	0.84	80/91	3.05
	PIAP	0.83	80/91	0.13
HN	AP*	0.66	19/16	0.80
	PAP	0.89	16/16	0.76
	PIAP	0.89	16/16	0.04
LN	AP*	0.70	23/17	0.80
	PAP	0.82	16/17	0.80
	PIAP	0.81	16/17	0.04

PsOP (Experiments)

- Raw
- YouTube
- KCF
- PsOP

PsOP (Experiments)

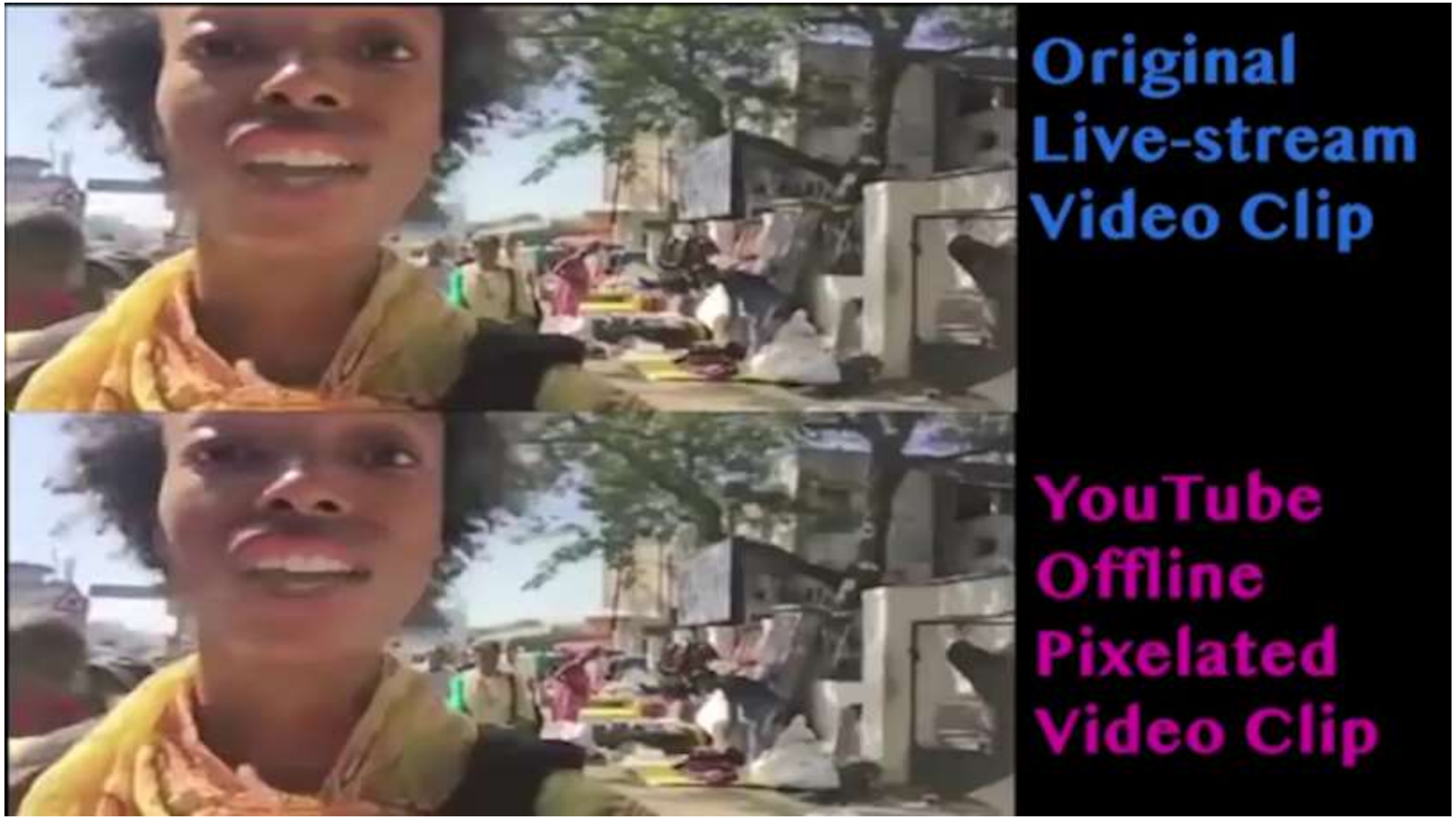
- YouTube
- PsOP

PsOP (Experiments)

- YouTube

- PsOP

PsOP (Experiments)



Conclusion

- A novel framework of Privacy-sensitive Objects Pixelation (PsOP) is proposed.
- Future works:
  - PsOP is still not very robust to low-resolution scenes.
  - PsOP is not efficient enough if there are too many privacy-sensitive objects in the frames (30+ more).

Publications

- J. Zhou, C.-M. Pun and Y. Tong, "Privacy-sensitive Objects Pixelation for Live Video Streaming," Proceedings of the 28th ACM International Conference on Multimedia (MM), 2020.
- J. Zhou and C.-M. Pun, "Personal Privacy Protection via Irrelevant Faces Tracking and Pixelation in Video Live Streaming," IEEE Transactions on Information Forensics and Security (TIFS), In Press, 2020.

Q&A

- Thank You!