DeepLight: Robust & Unobtrusive Real-time Screen-Camera Communication for Real-World Displays

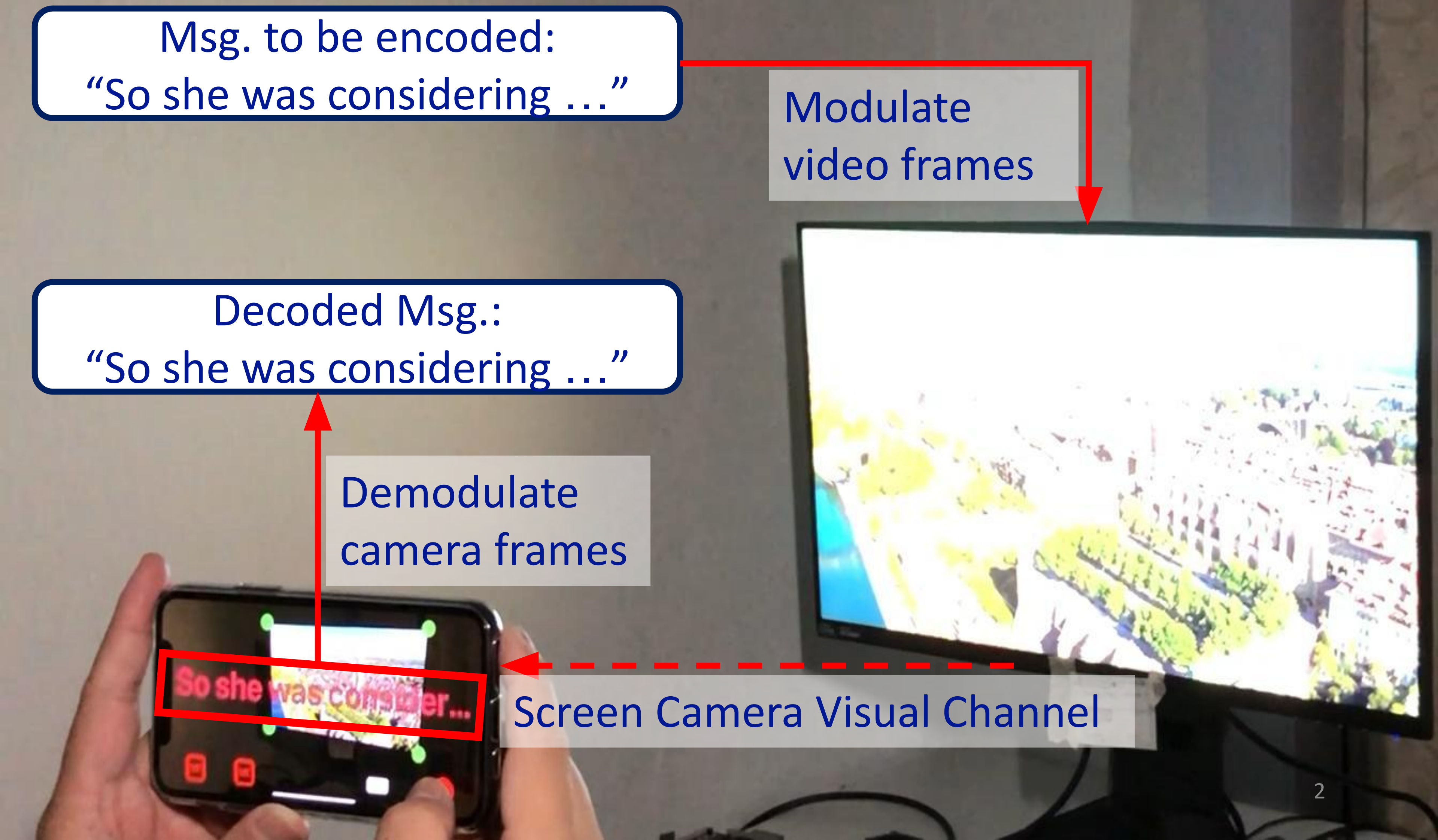
Vu Tran (University of Oxford, Singapore Management University)

Gihan Jayatilaka (University of Peradeniya)

Ashwin Ashok (Georgia State University)

Archan Misra (Singapore Management University)

Screen-Camera Communication: Background



Msg. to be encoded:
“So she was considering …”

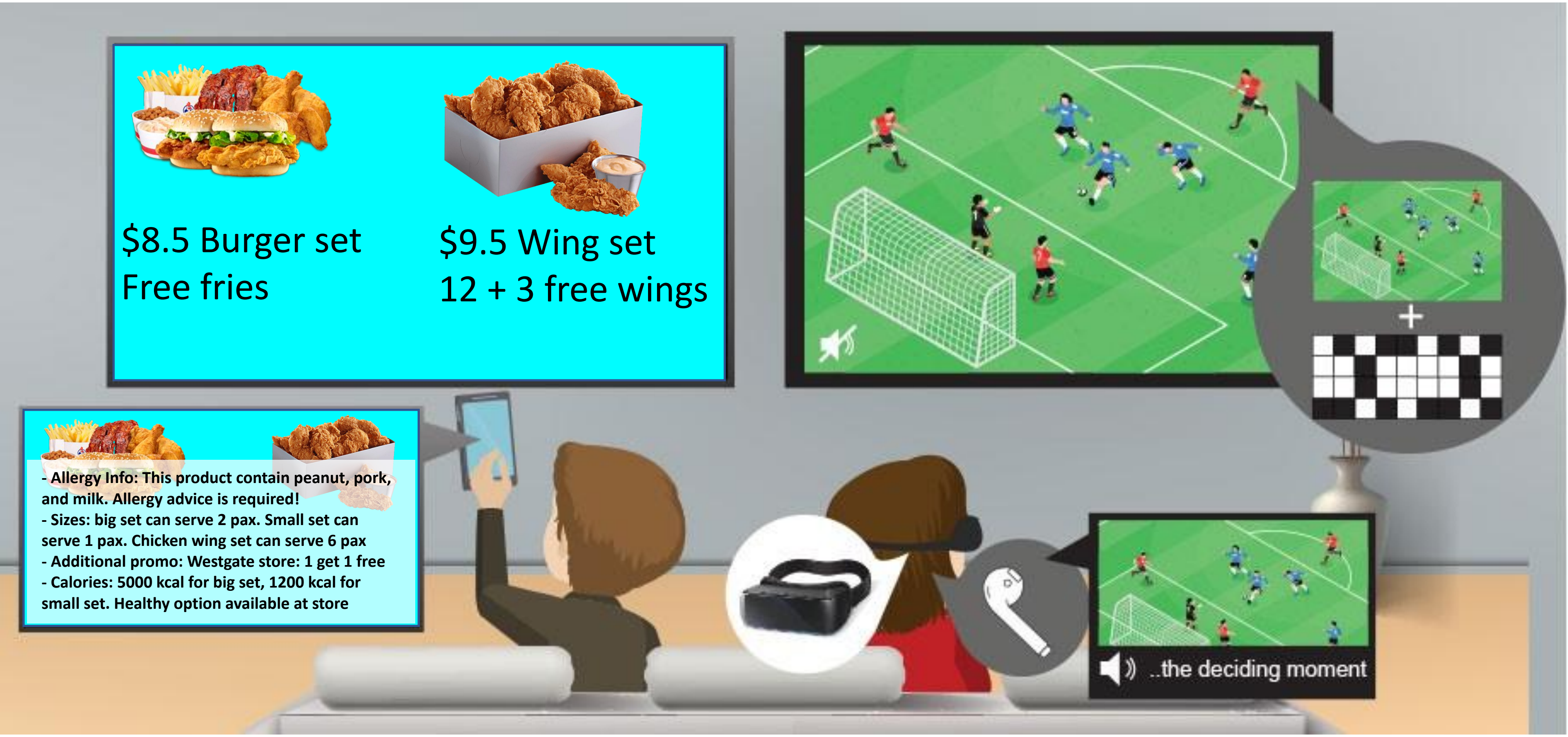
Modulate video frames

Decoded Msg.:
“So she was considering …”

Demodulate camera frames

Screen Camera Visual Channel

SCC use cases

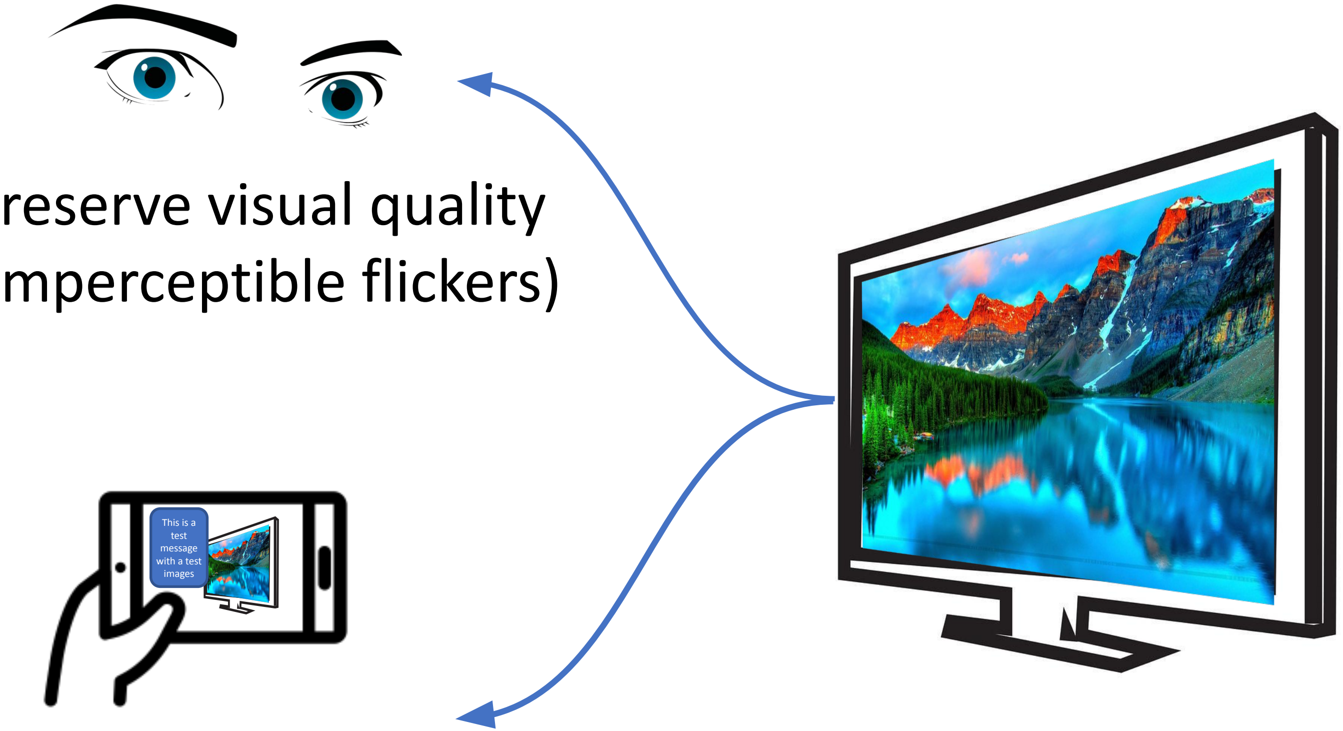


- Allergy Info: This product contain peanut, pork, and milk. Allergy advice is required!
- Sizes: big set can serve 2 pax. Small set can serve 1 pax. Chicken wing set can serve 6 pax
- Additional promo: Westgate store: 1 get 1 free
- Calories: 5000 kcal for big set, 1200 kcal for small set. Healthy option available at store

Augmented text, instructions, audio, … on public screens

- Only users, who want to, receive hidden information.
- Avoid annoying other users

Two key objectives

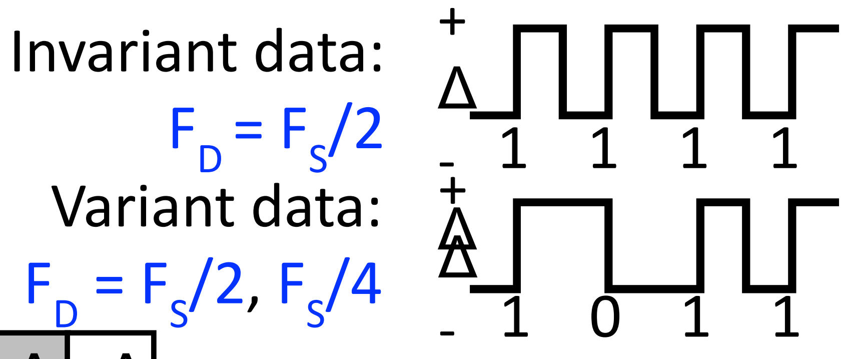


Preserve visual quality
(imperceptible flickers)

Achieve high goodput
(low error rate)

State-of-the-art Encoder: How hidden data is embedded?

- Frequency modulation
- Manchester coding

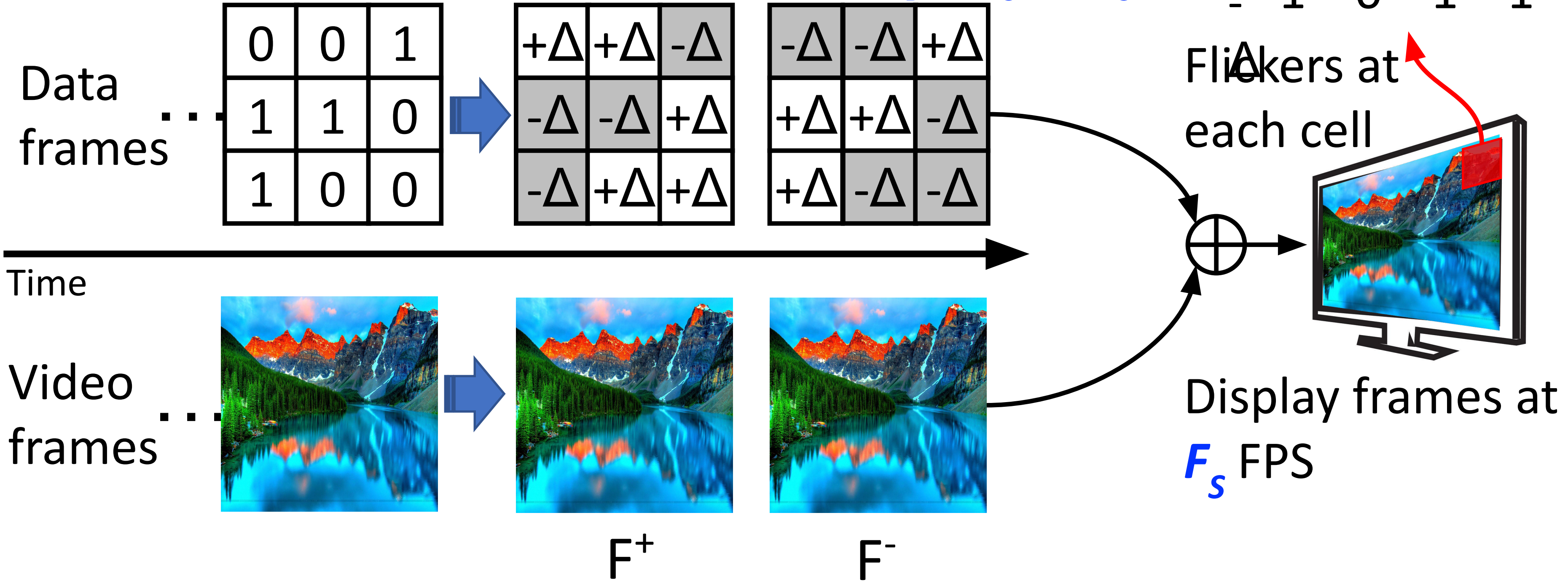


Invariant data:

$F_D = F_S/2$

Variant data:

$F_D = F_S/2, F_S/4$



0	0	1
1	1	0
1	0	0

+Δ	+Δ	-Δ
-Δ	-Δ	+Δ
-Δ	+Δ	+Δ

-Δ	-Δ	+Δ
+Δ	+Δ	-Δ
+Δ	-Δ	-Δ

Flickers at each cell

Display frames at F_S FPS

State-of-the-art Encoder: Suppress flickers with high display rates

$F_D < 50\text{Hz}$: perceptible

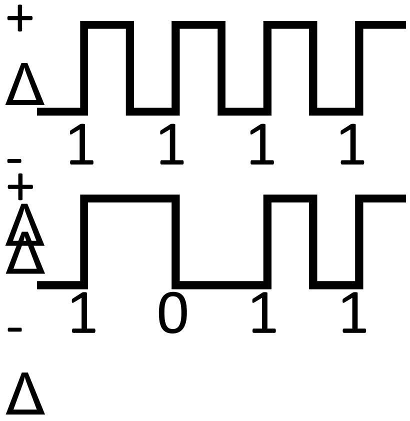
$F_D > 50\text{Hz}$: imperceptible

Invariant data:

$F_D = F_S/2$

Variant data:

$F_D = F_S/2$ or $F_S/4$

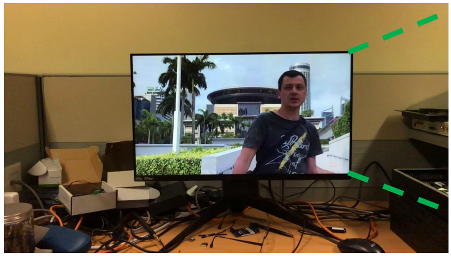


To achieve imperceptible flickers:

- $F_S > 100$ FPS for invariant data
- $F_S > 200$ FPS for variant data

How to support imperceptibility at common frame rates (e.g., 30, 60FPS) ?

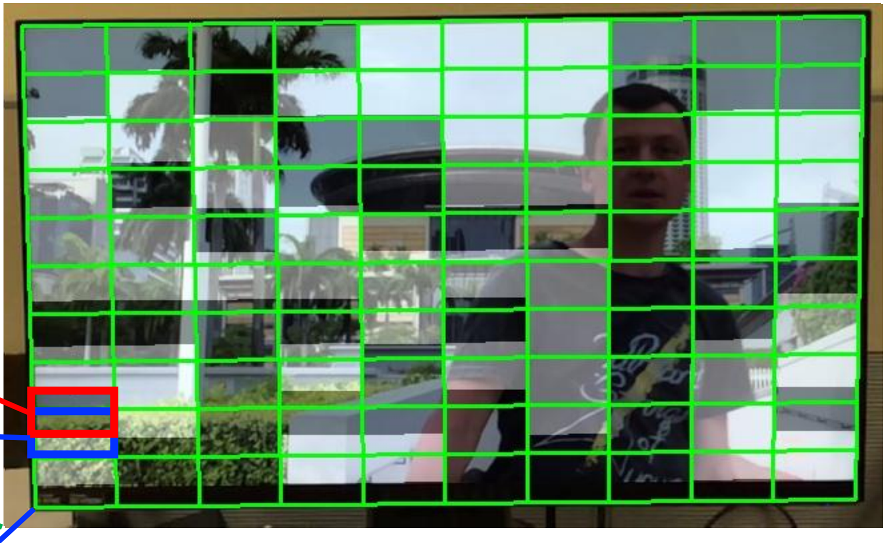
State-of-the-art Decoder: Grid splitting



Camera Frame

Tx Cell

Rx Cell



Modulated frame recorded at receiver side
(Overly high amplitude for visibility)

Mistakenly include lower edge
(IOU = 95%)

Severe interference!

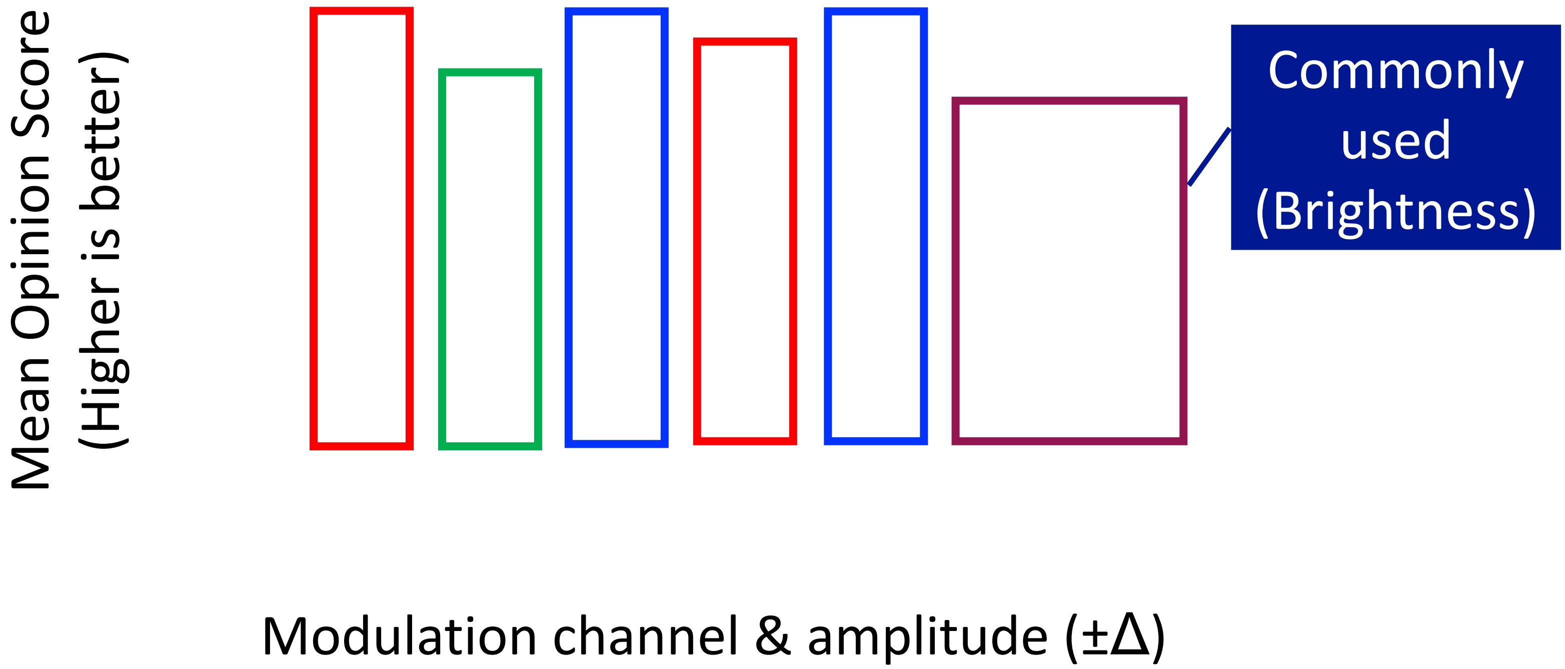
How to support robust decoding with imperfect screen extraction?

DeepLight contributions

- Blue channel modulation for imperceptibility at common frame rates (60 FPS)
- A holistic decoding method using convolutional neural network; support imperfect screen extraction
- A hybrid screen extraction method for practically high screen extraction accuracy

Imperceptibility at **common (60FPS)** display rates: **Blue** light

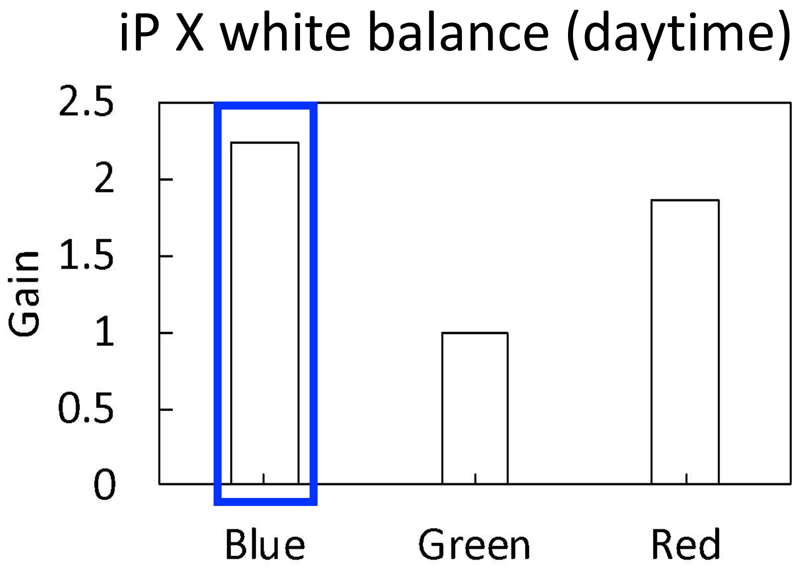
Human eyes are known to be less sensitive to Blue color



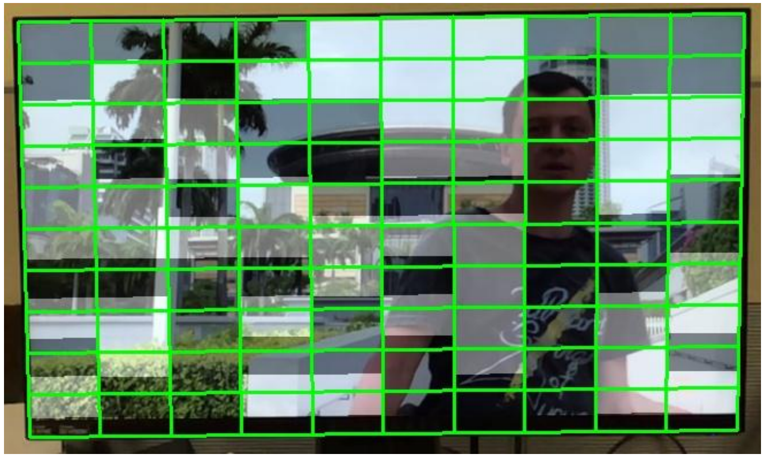
Commonly used (Brightness)

- **Green**, **Brightness**: low MOS even with the lowest amplitude (±1)
- **Red**: Low visual quality with higher amplitude (±2)
- **Blue**: High visual quality even with higher amplitude (±2)

Cope with noise



More sensor noise (Blue)

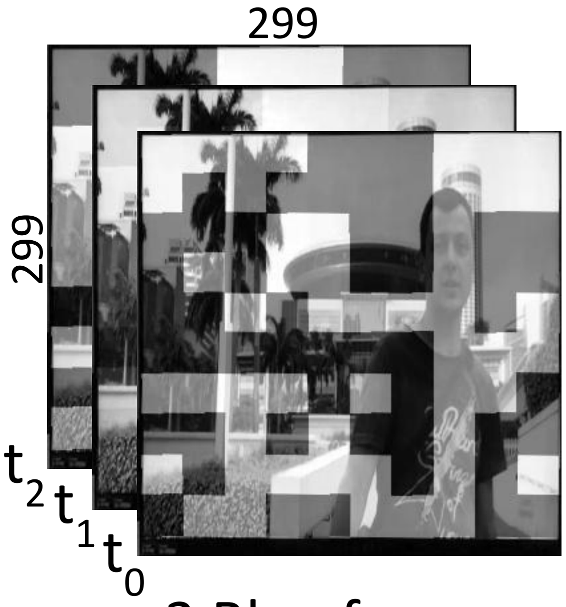


Cross channel noise
(Imperfect screen extraction)

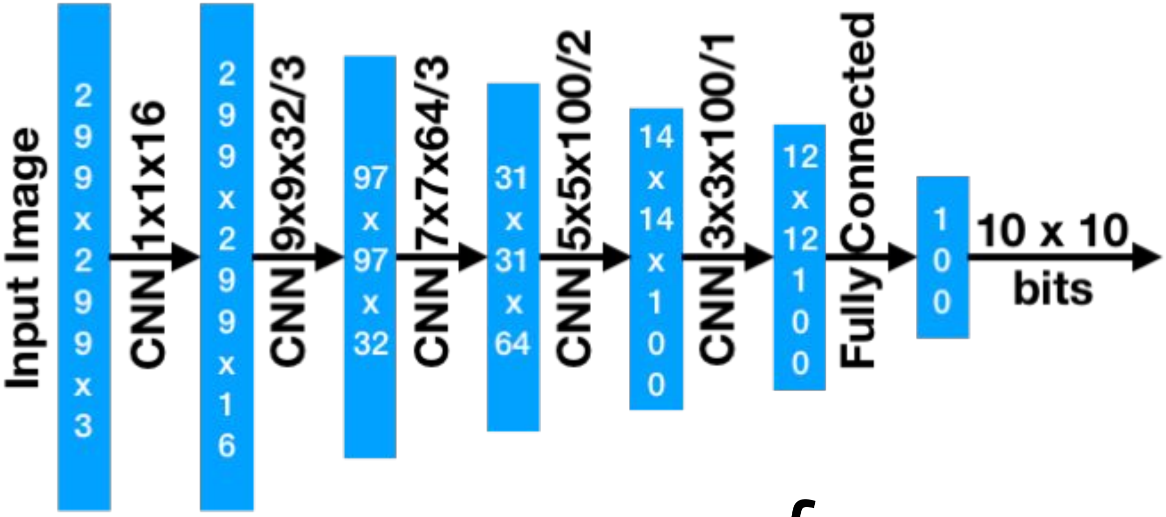
Increase mod. amplitude Δ? ✗ Decrease visual quality

- Learning-based decoding instead of hard thresholding
- Avoid grid-splitting

DeepLight Holistic Decoder



3 Blue frames



Learn function *output* = f (F_k^p, F_k^t, F_k^n)

- Each bit is inferred using the entire "screen", not just a cell
- Learn temporal relation (Manchester coding)

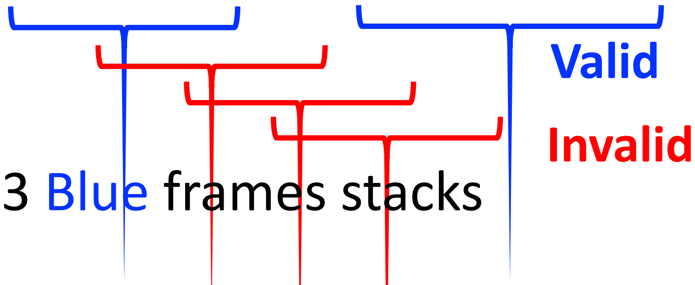
$$F_k^p, F_k^t, F_k^n, F_k^t, F_{k+1}^p, F_{k+1}^t, \ldots$$

Assume $F_{camera} = 2F_{display}$

F_k^s : Camera frame corresponding to Manchester pair *k*
frame type s (*p*ositive, *t*ransition , *n*egative)

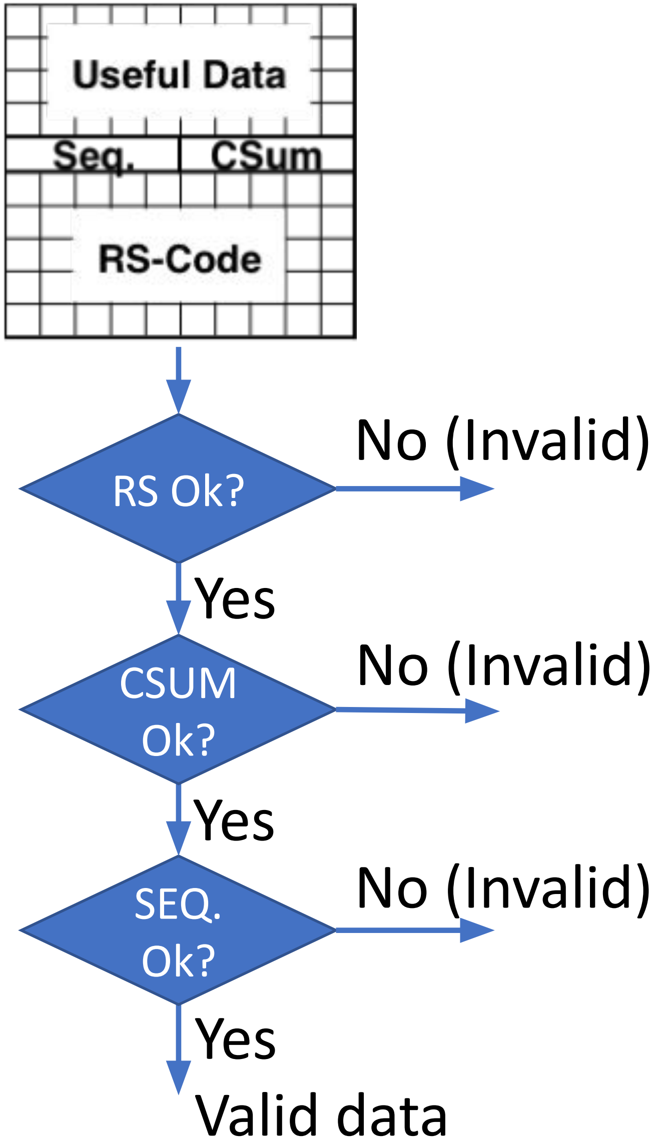
Filtering out invalid frames

$F_k^p, F_k^t, F_k^n, F_k^t, F_{k+1}^p, F_{k+1}^t, \cdots$

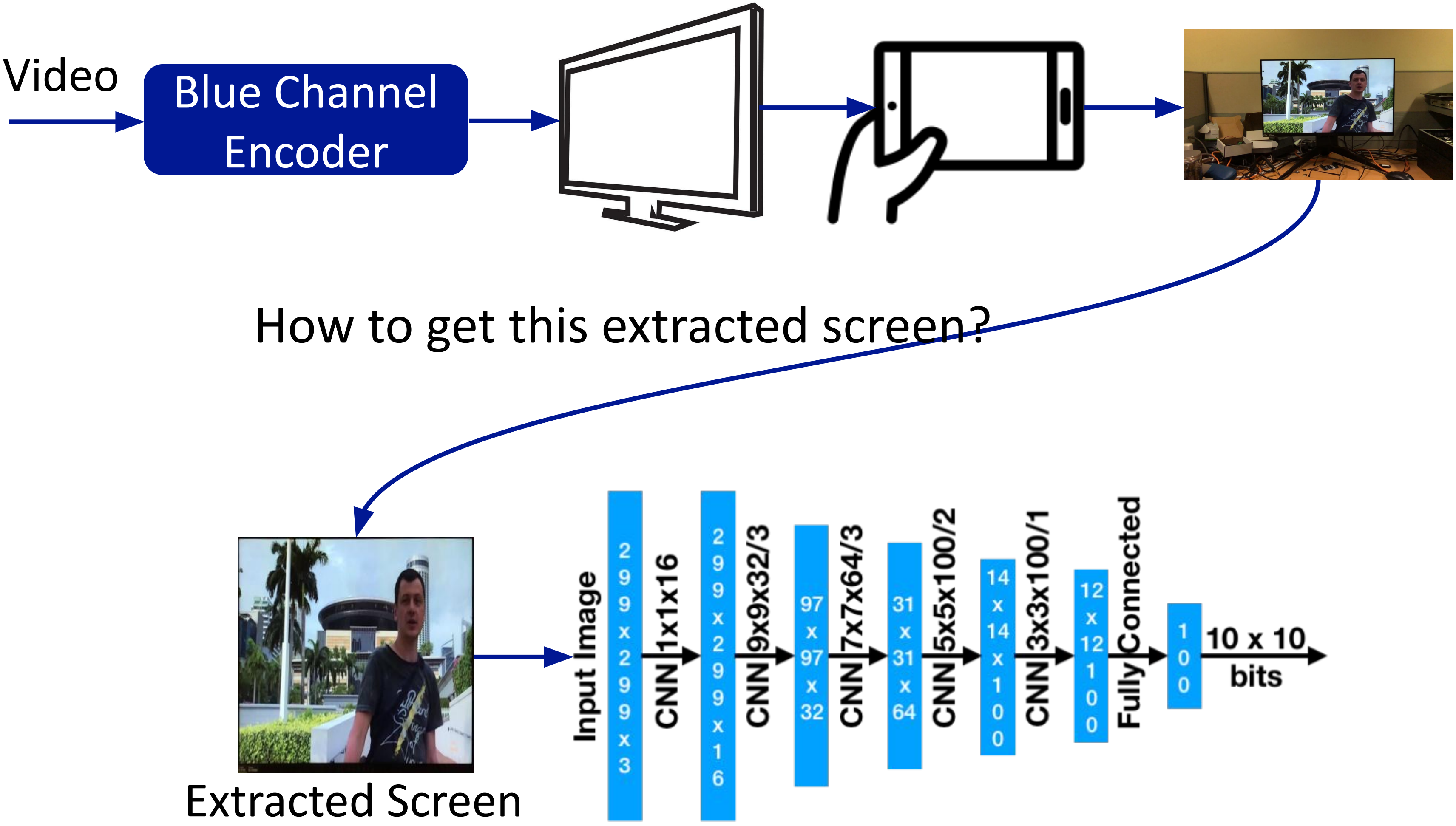


3 Blue frames stacks

- Apply structured data: RS-Coding, Checksum, Sequence number
- Detect invalid frames in a cascading manner



DeepLight screen detection



Video

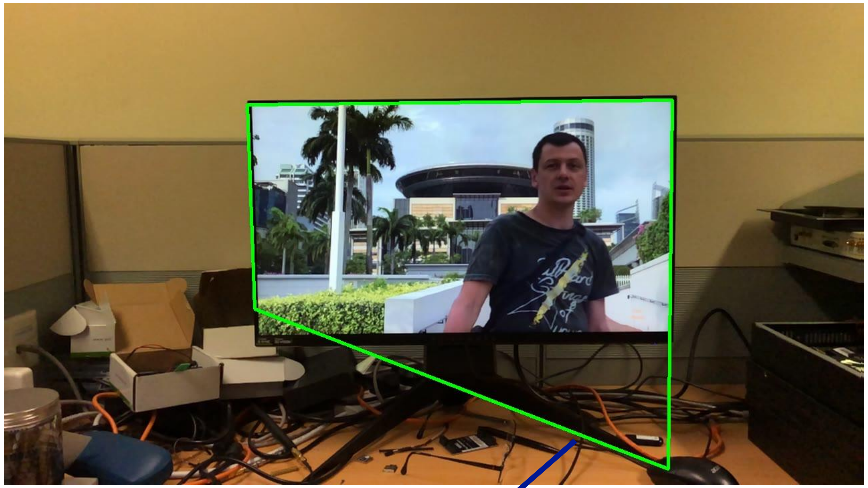
Blue Channel Encoder

How to get this extracted screen?

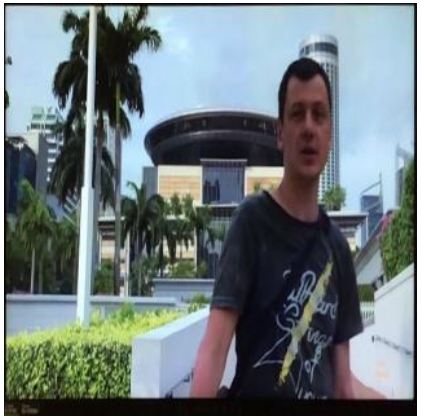
Extracted Screen

DeepLight screen detection

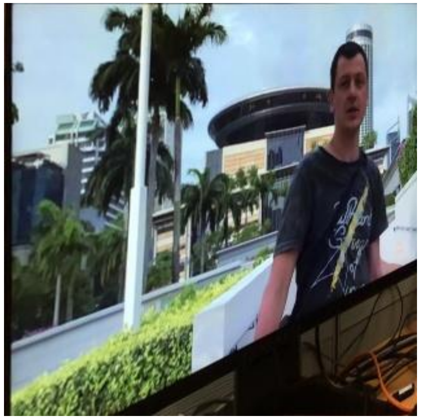
“practically” accurate screen extraction is still necessary



Canny + Hough Transform:
Tricked by nearby “line” textures



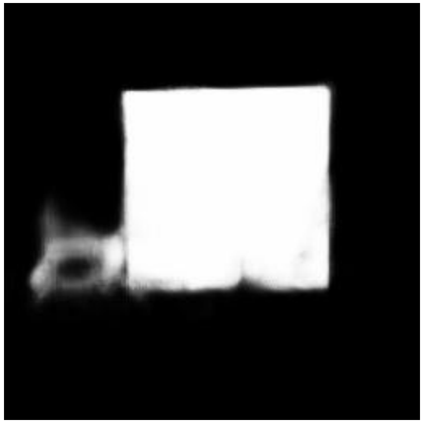
Expected



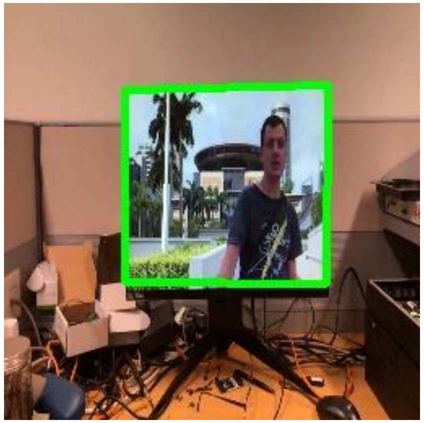
Reality

DeepLight screen detection

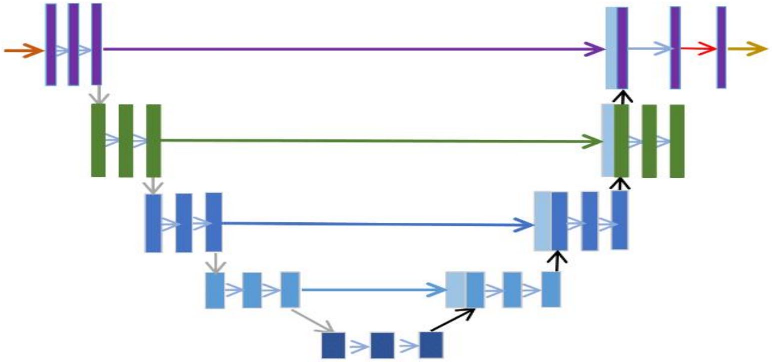
1. U-net based segmentation:
Filter out “non-screen” areas



2. Contour analysis



3. Perspective Transform



U-Net [MICCAI 2015]

Imperfect, but DeepLight decoder can deal with imperfection

DeepLight Evaluation

Mean Opinion Score (**MOS**):

- 1: Very unpleasant
- 2: It's bad
- 3: It could be better
- 4: It's good
- 5: Cannot differentiate from the original video

Performance metrics:

- Raw throughput: $(1\text{-BER})*F_S*D$
- Throughput: $(1\text{-FER})*F_S*D$
- Goodput: $(1\text{-FER})*F_S*U$

Note: Raw throughput is not informative if BER is high

Default settings:

- We used a 25” monitor
- Display rate: 60 FPS
- Grid size: 10x10

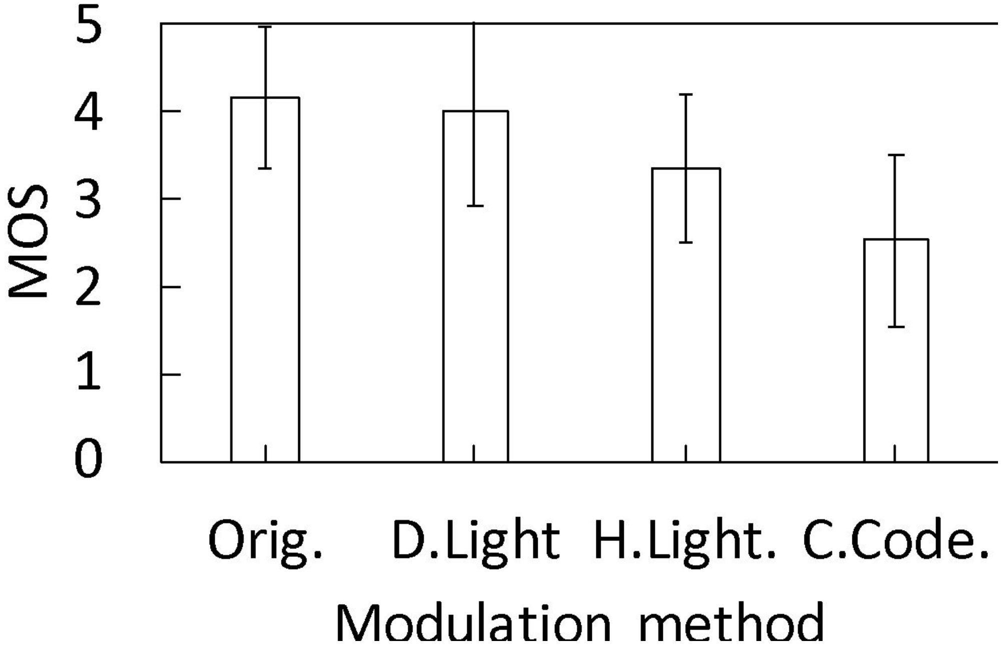
BER: Bit error Rate; **FER**: Frame error rate (only recoverable frames)

D: Number of bits in a frame; **U**: Number of useful bits in a frame; $\mathbf{F_S}$: Display frame rate

DeepLight preserves visual quality

17 participants:

- 10 males, 18 to 32 years old
- 1 astigmatism, 3 farsighted, 7 shortsighted
- Each person watches 6 video clips X 4 versions



DeepLight outperforms others in term of MOS at 60FPS

D.Light: DeepLight, **H.Light**: HiLight, **C.Code**: ChromaCode

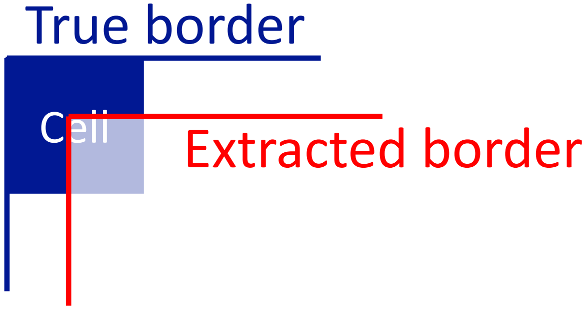
DeepLight performance with fixed camera

0% screen extraction error

> 0% screen extraction error (d=1.5m)

> 1.0 Kbps even at 2.0m

Example 40% SHF error:
64% cell area loss



True border

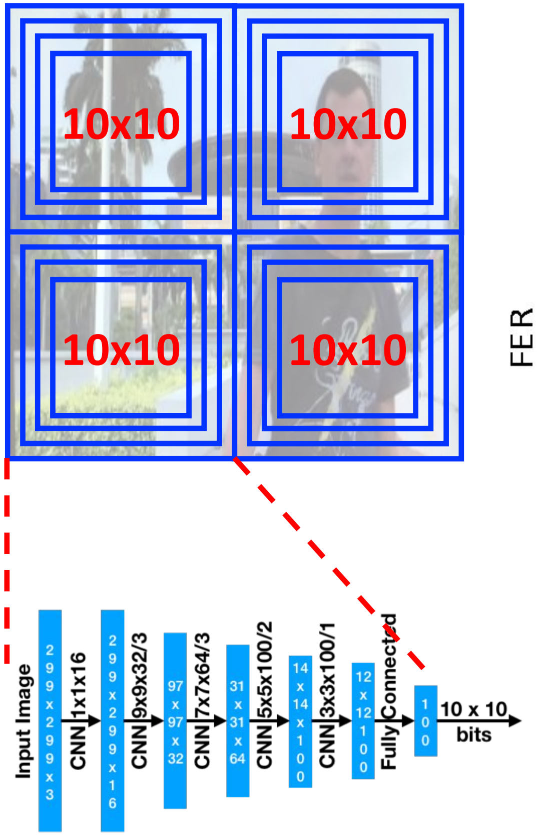
Cell

Extracted border

SHF: Shift, **EXP**: Expand, **SHR**: Shrink, **ROT**: Rotate

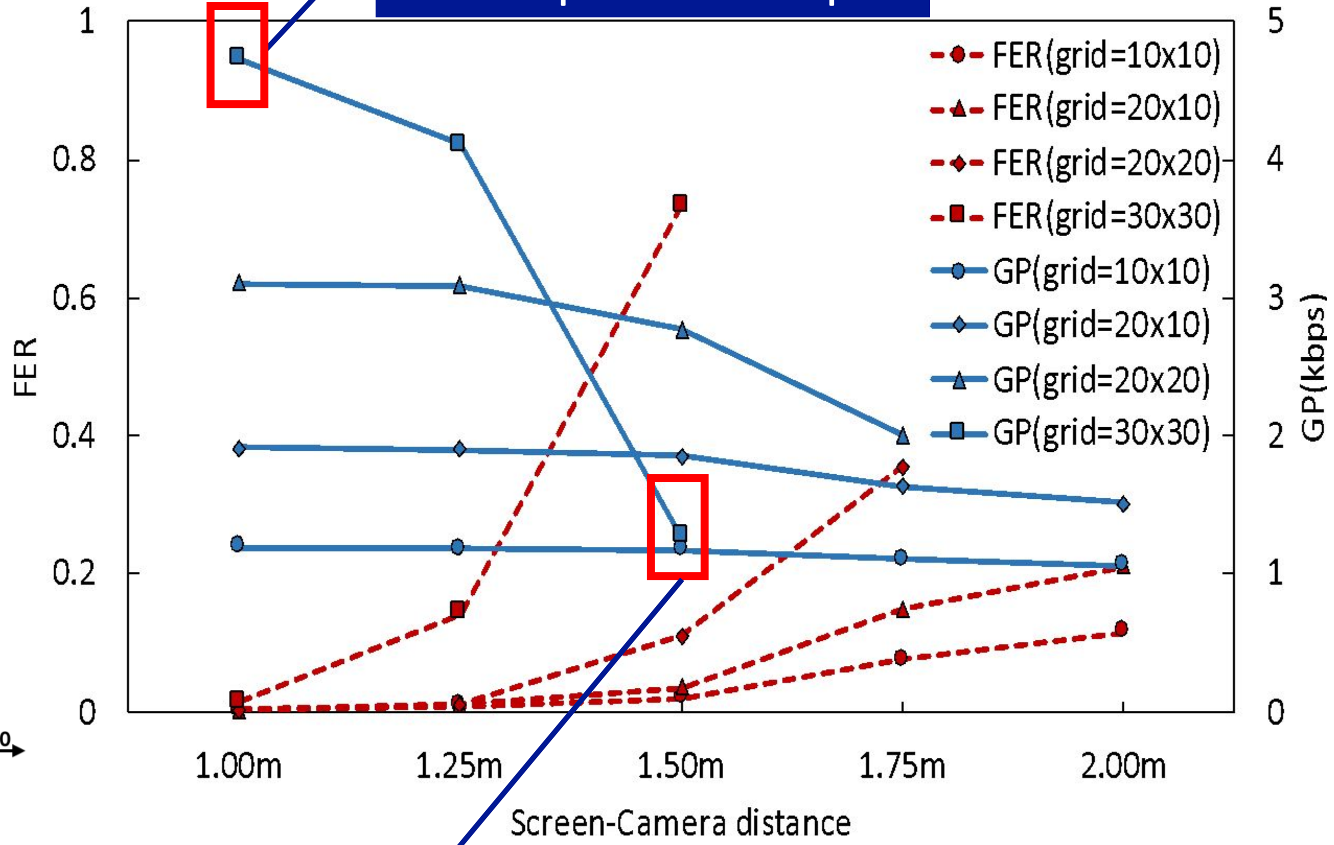
Support larger grid size

Grid = 20 x 20



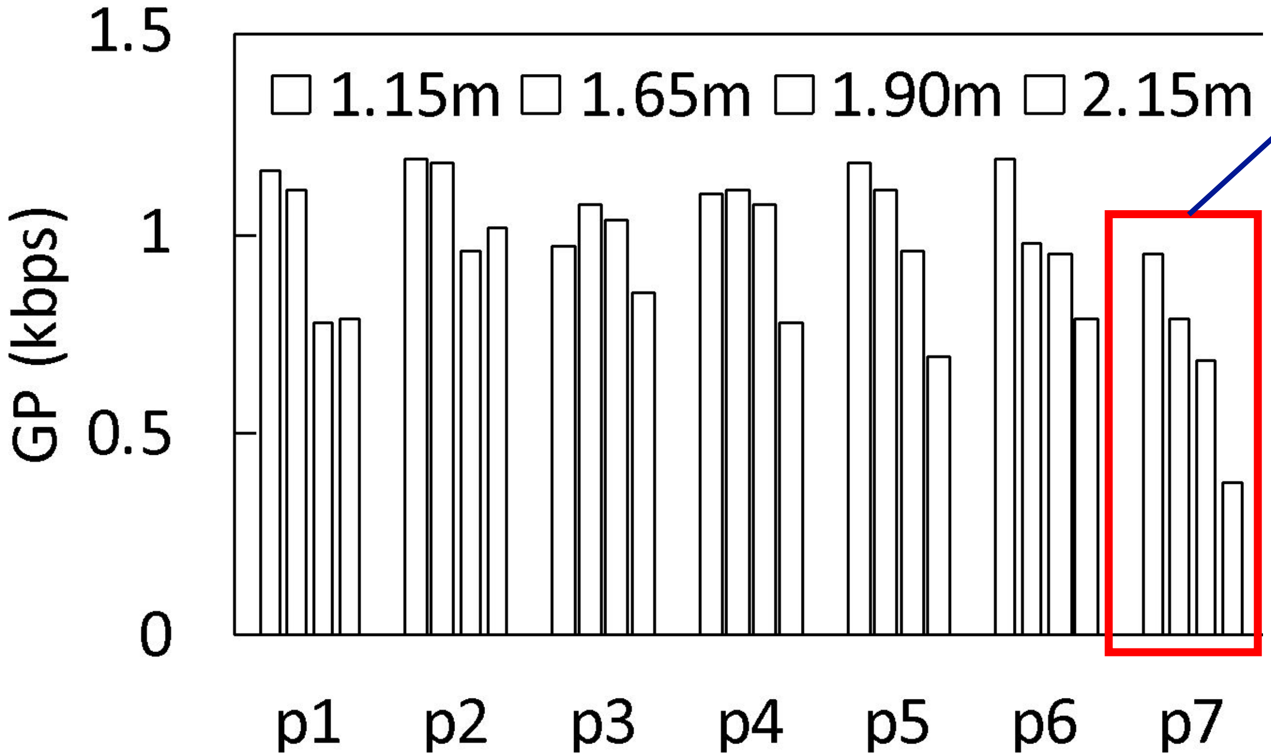
Canonical 10x10 model

Throughput = 26kbps
Goodput = 4.7kbps



cell size (appears in camera) ~= 4x4 pixels
Higher resolution might help

DeepLight performance with hand-held camera

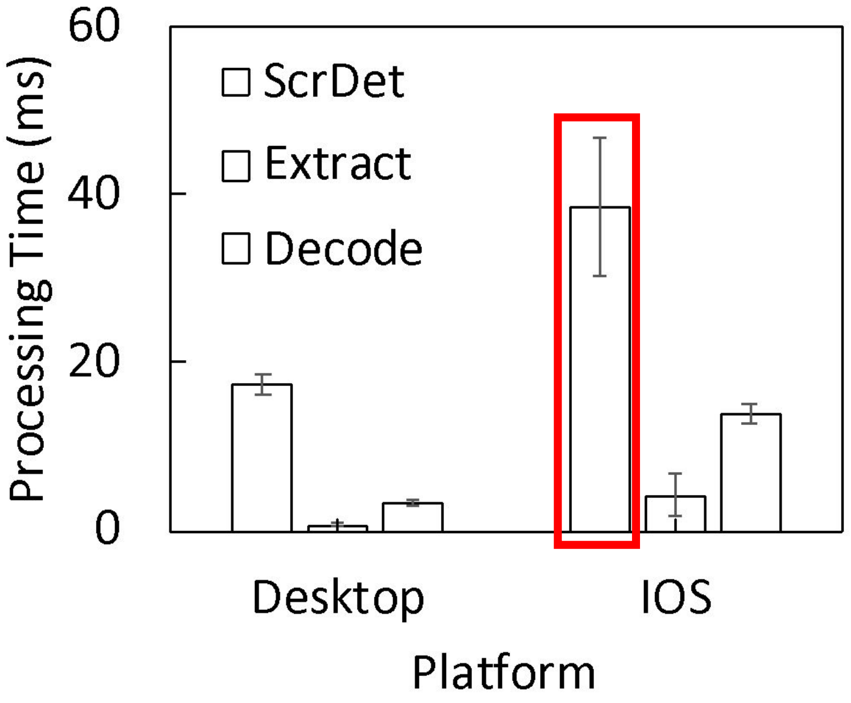


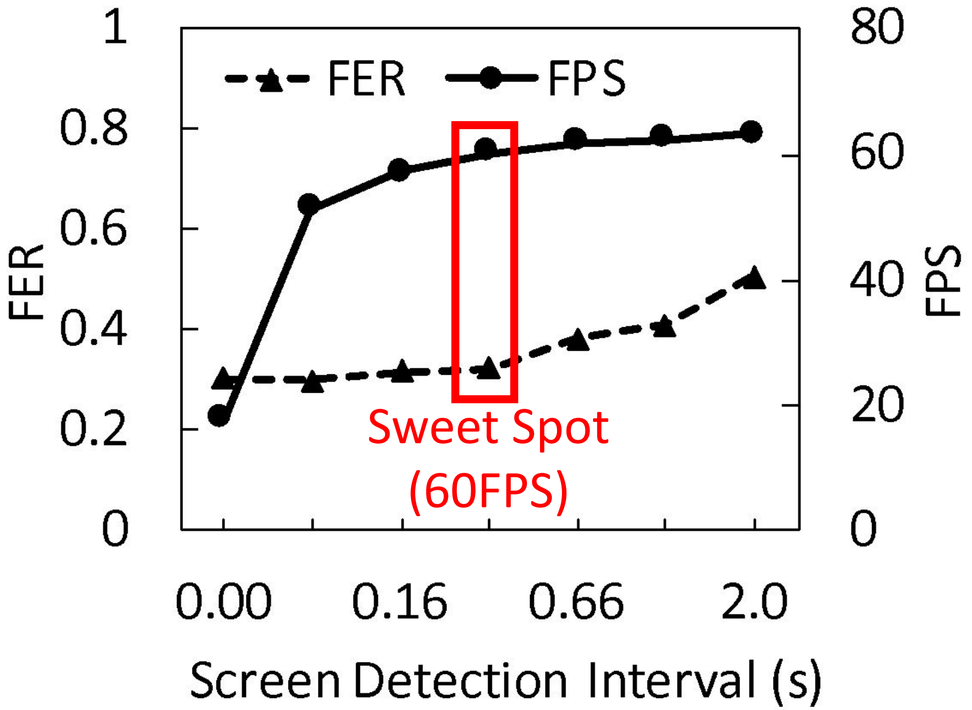
Severe hand tremors

- 7 participants, seated on a chair
- Instructed **not** to lean their arms on the chair arms

Goodput > 0.9kbps at 1.9m for most of the participants

DeepLight on Smartphone

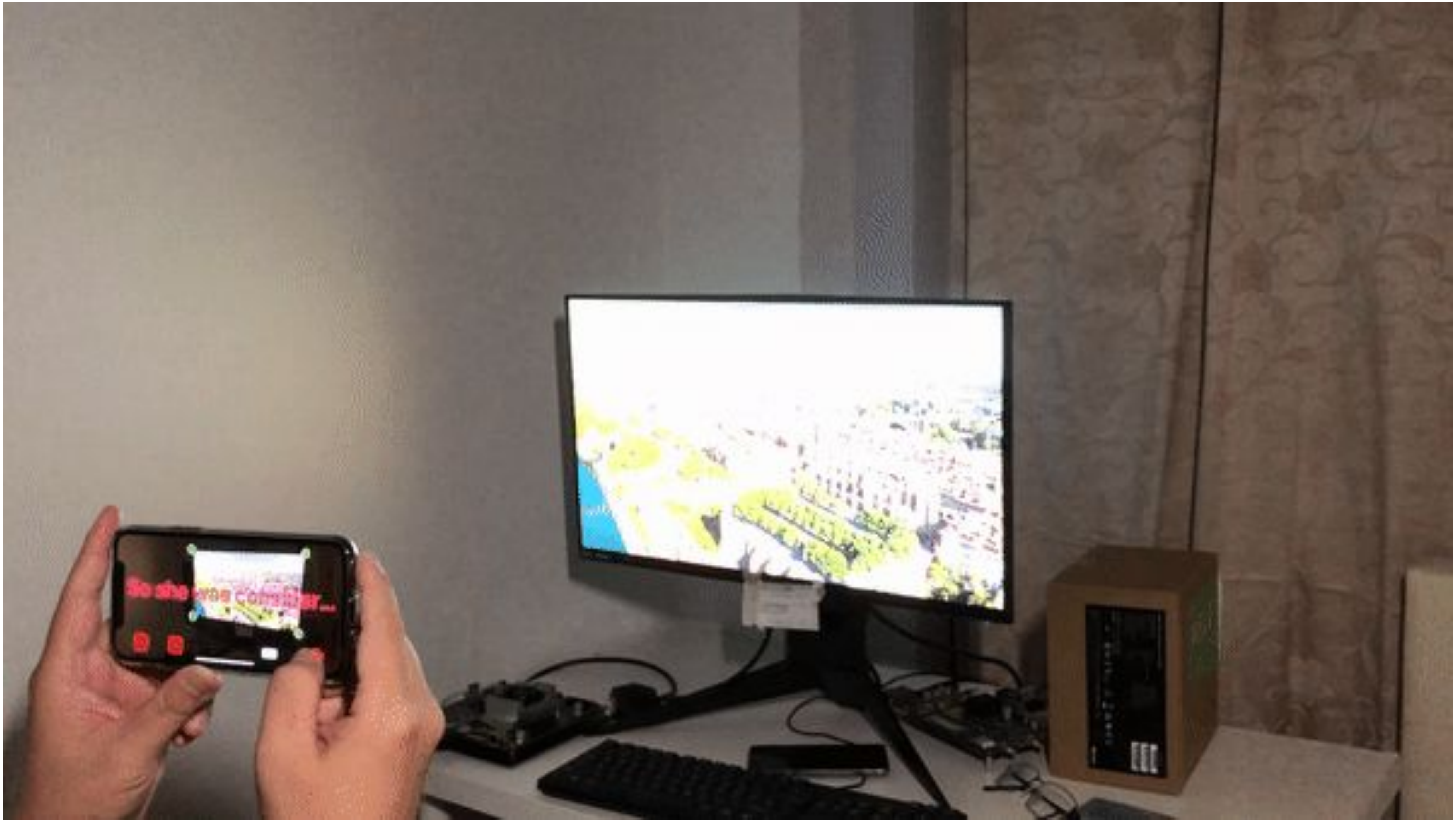




- We do not need screen detection for every camera frame (8.3ms)

Experiment with a **walking** user (more motion artifacts)

DeepLight captioning app.



- Press to process the latest 32 frames in buffer
- Detect screen in the first frame only

Comparison with previous works

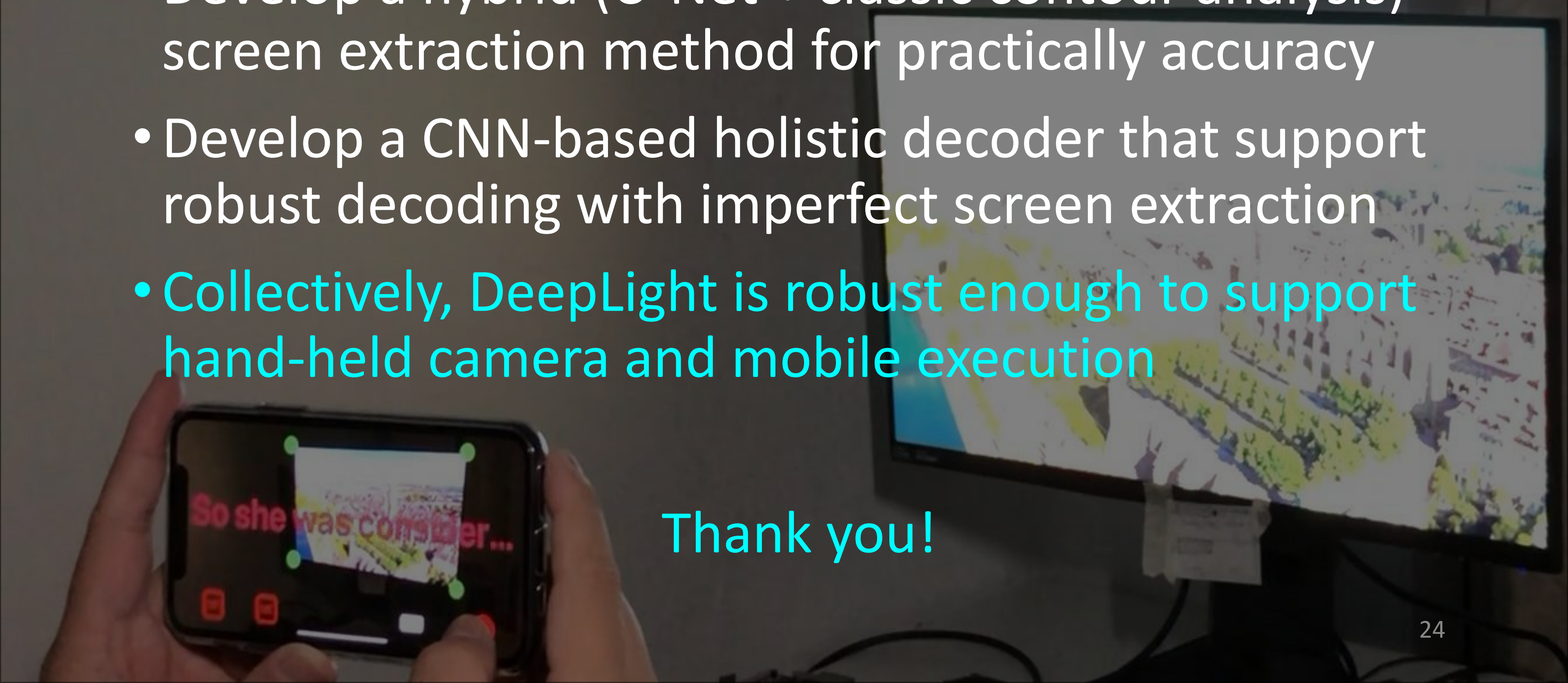
Work	Require SCR locator	Visual Quality	Throughput (kbps)	Goodput (kbps)	Processing time on mobile phone (ms)
DeepLight	No	Very high	26.6	4.7	16.6 (iPhone 11 Pro)
ChromaCode (2018) [5]	Yes (Black & White lines)	Low	N.A. (220 -- Raw throughput)	0.5 – 1.5	500 (Pixel2)
TextureCode (2016) [8]	N.R (Offline extracted)	N.A.	11.25	N.A.	N.A. (Offline)
Inframe++ (2015) [3]	Yes (QRCode locator)	N.A.	9	N.A.	200 (Core-i5 CPU + FirePro V3900 GPU)
Hilight (2015) [4]	Yes (OFF/ON screen)	High	4.6	N.A.	5 (iPhone 5)

Some values are borrowed from [5] and [8], normalized to 60FPS

DeepLight code: https://github.com/LARC-CMU-SMU/deeplight

Summary

- Apply Blue channel modulation for imperceptibility at common frame rate (60FPS)
- Develop a hybrid (U-Net + classic contour analysis) screen extraction method for practically accuracy
- Develop a CNN-based holistic decoder that support robust decoding with imperfect screen extraction
- Collectively, DeepLight is robust enough to support hand-held camera and mobile execution



Thank you!

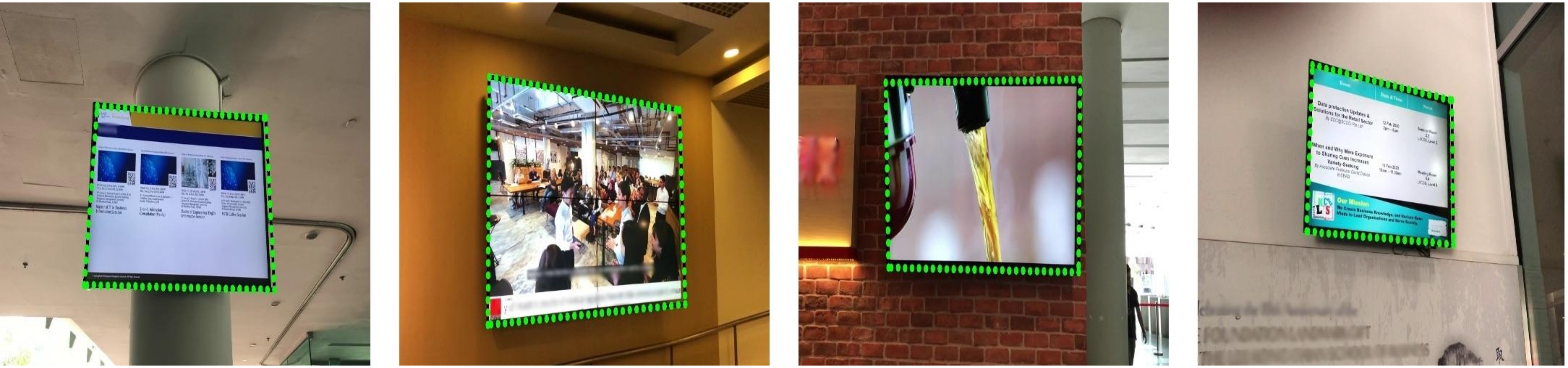
More details …

Training LightNet

Generate random error (-50%, 50%)

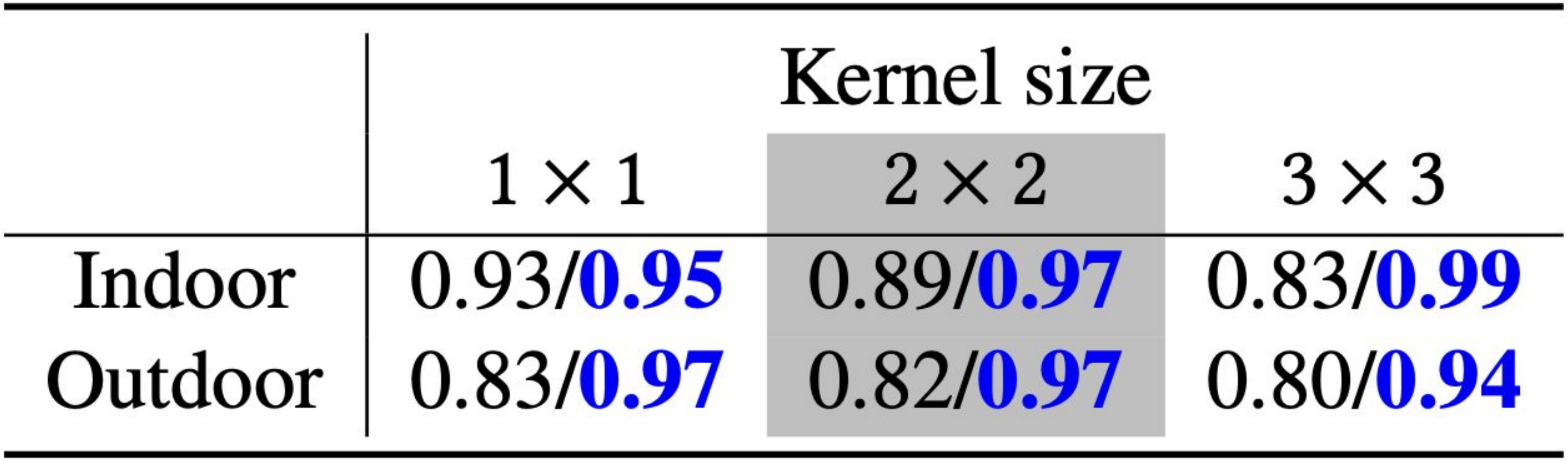
- Stack 3 consecutive Blue frames to form a sample
- Sample $S_k = \{B_k^p, B_k^t, B_k^n\}$
- 22500 fixed camera samples
- 25200 hand-held camera samples

Training ScreenNet



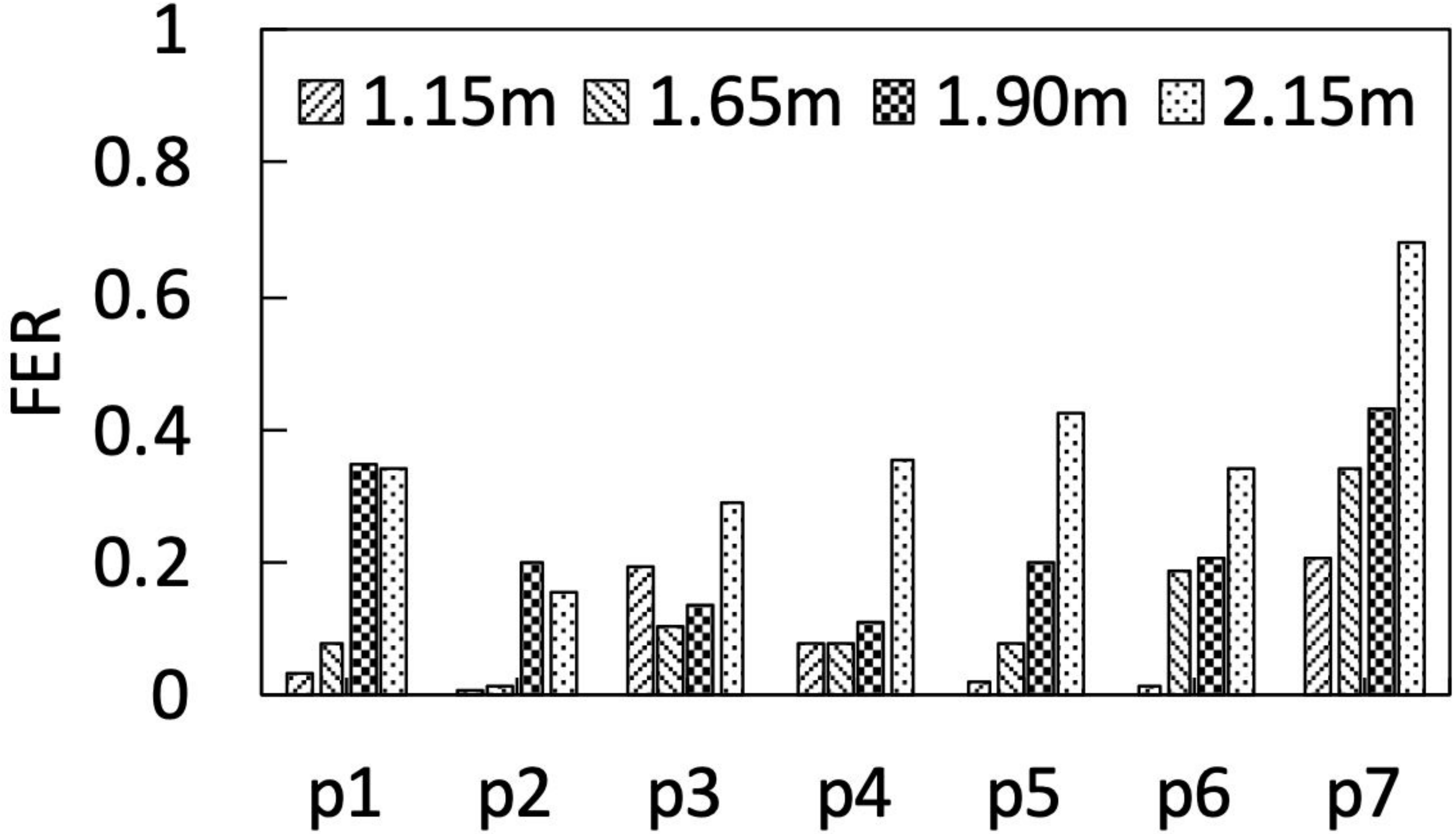
- Collect screens from Google search
- Take photos of screens at different places
- ~800 images + data augmentation (rotation, displacement, scale, …)

ScreenNet performance

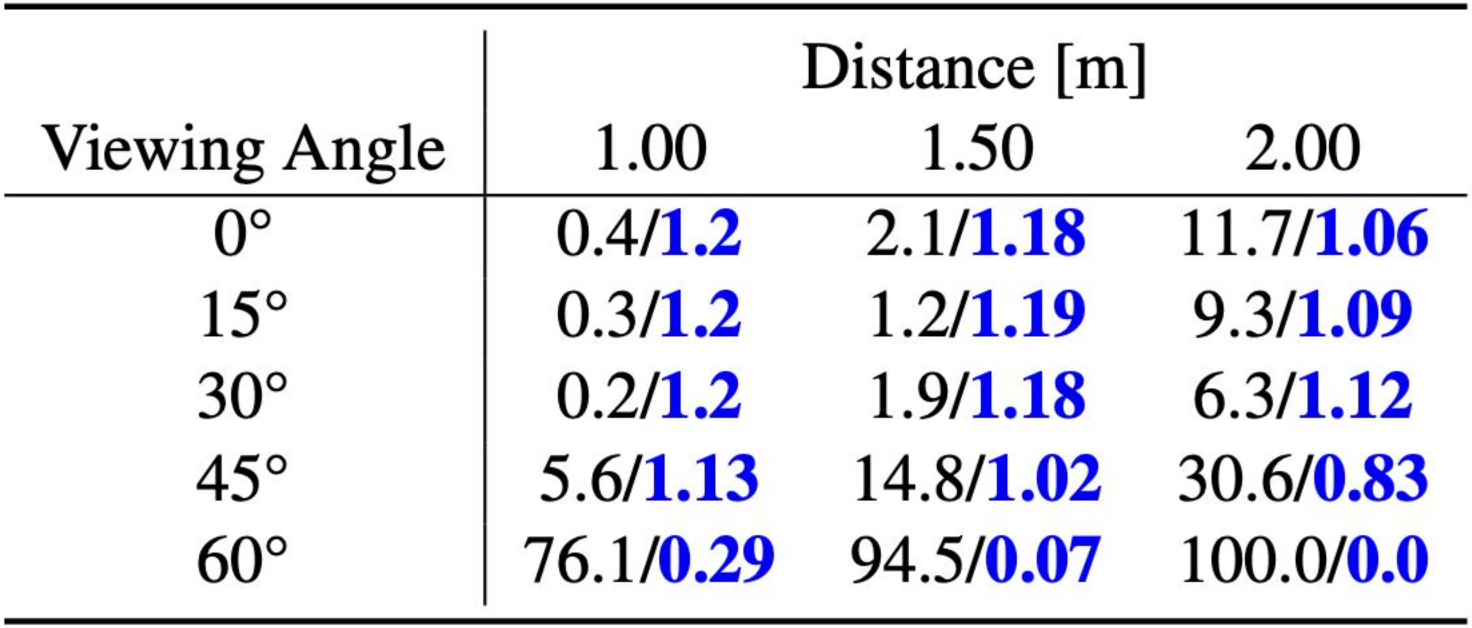


	Kernel size		
	1 × 1	2 × 2	3 × 3
Indoor	0.93/**0.95**	0.89/**0.97**	0.83/**0.99**
Outdoor	0.83/**0.97**	0.82/**0.97**	0.80/**0.94**

Hand-held performance

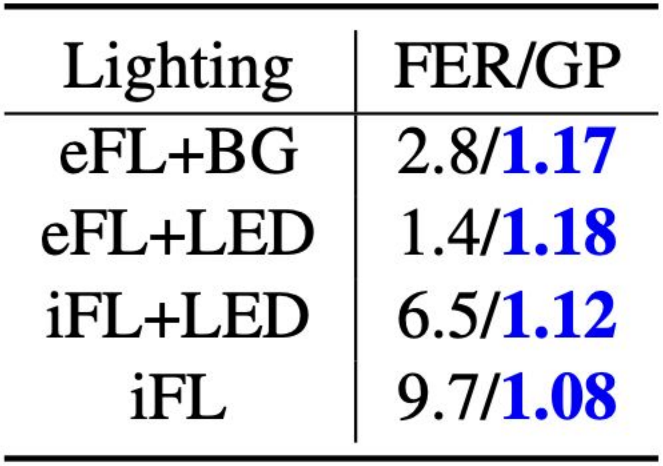


DeepLight vs. Viewing angle



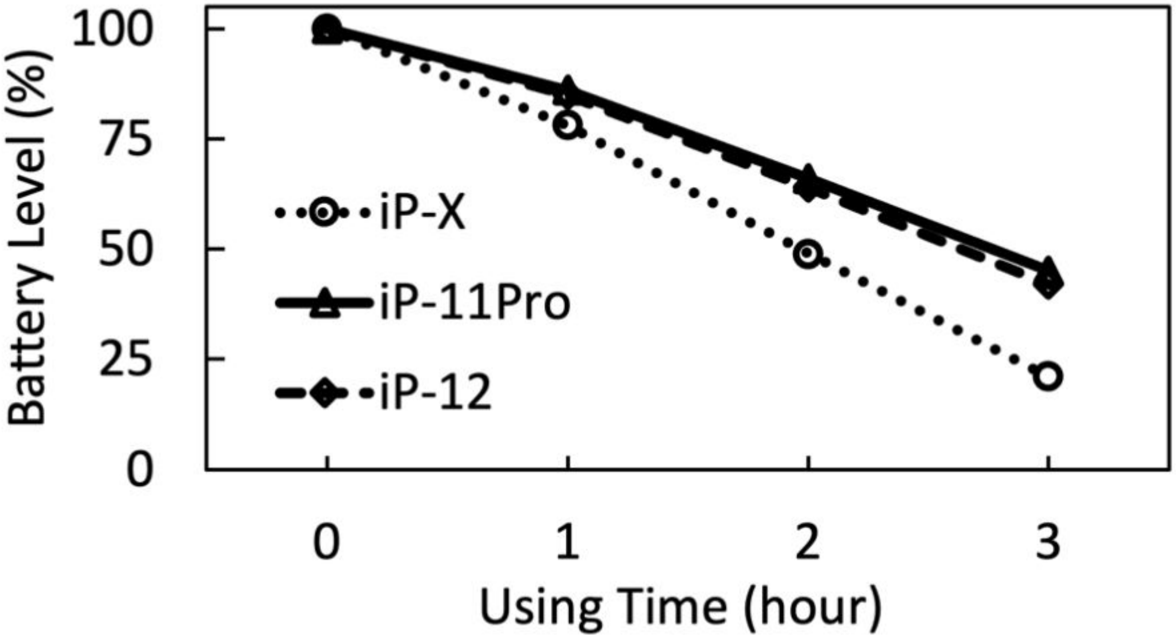
	Distance [m]		
Viewing Angle	1.00	1.50	2.00
0°	0.4/**1.2**	2.1/**1.18**	11.7/**1.06**
15°	0.3/**1.2**	1.2/**1.19**	9.3/**1.09**
30°	0.2/**1.2**	1.9/**1.18**	6.3/**1.12**
45°	5.6/**1.13**	14.8/**1.02**	30.6/**0.83**
60°	76.1/**0.29**	94.5/**0.07**	100.0/**0.0**

DeepLight vs. Ambient lighting



Lighting	FER/GP
eFL+BG	2.8/**1.17**
eFL+LED	1.4/**1.18**
iFL+LED	6.5/**1.12**
iFL	9.7/**1.08**

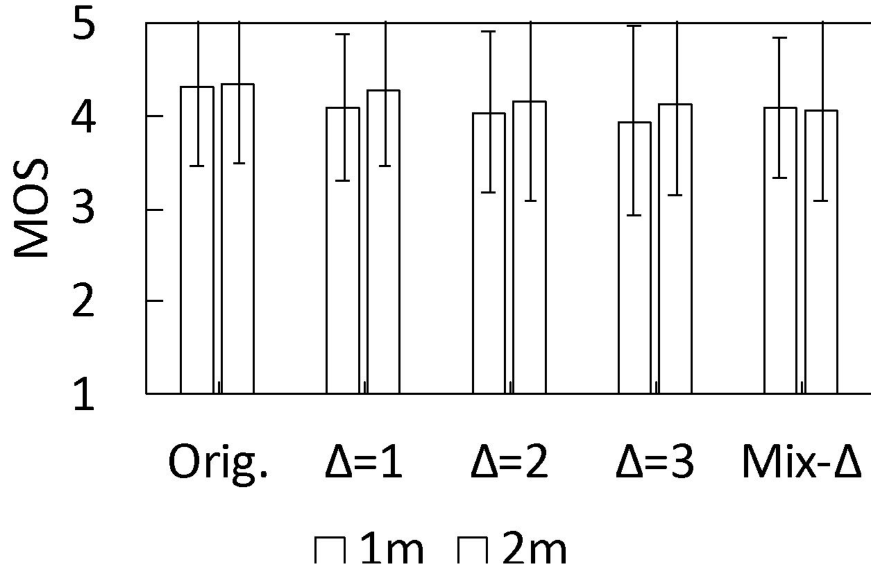
Energy consumption

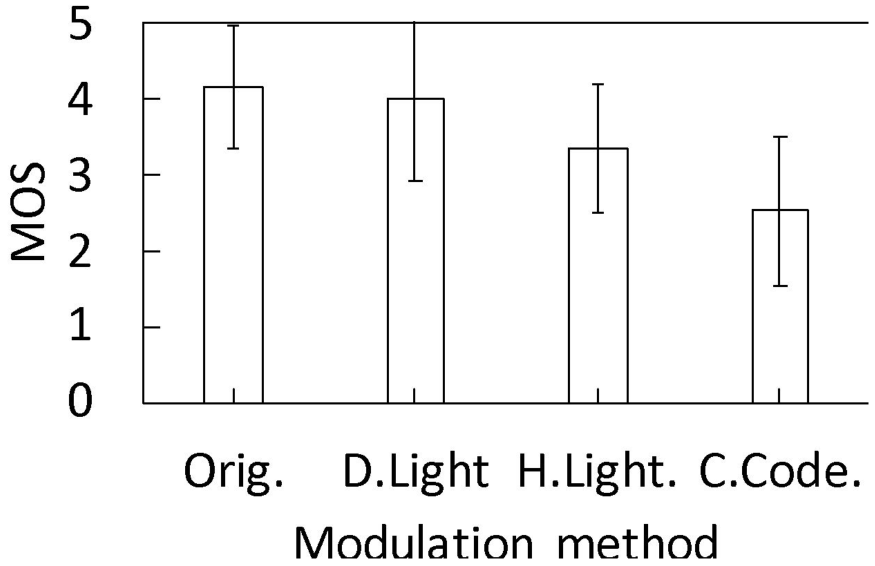


DeepLight preserves visual quality

17 participants:

- 10 males, 18 to 32 years old
- 1 astigmatism, 3 farsighted, 7 shortsighted





$$\text{Mixed-}\Delta\text{:}\begin{cases}\Delta = 3\text{ ; avg. cell value} < 30\\ \Delta = 2\text{ ; otherwise}\end{cases}$$

DeepLight outperforms others in term of MOS at 60FPS

H.Light: HiLight, C.Code: ChromaCode