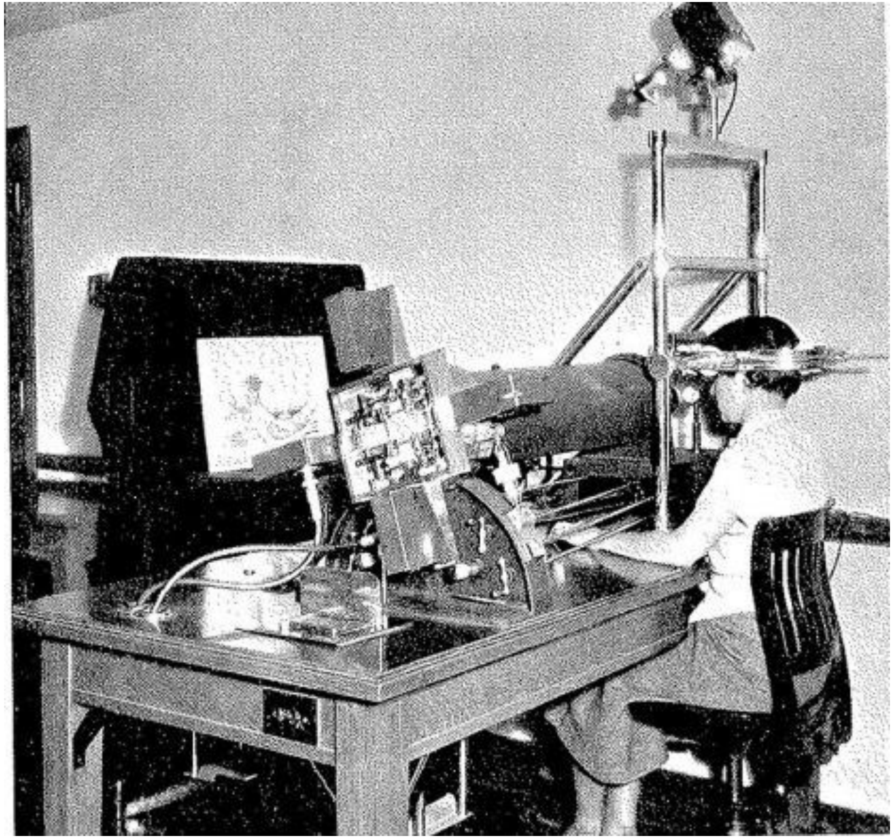
RGBDGaze: Gaze Tracking on Smartphones with RGB and Depth Data

Riku Arakawa

(adviser: Mayank Goel)

Background: gaze tracking technology

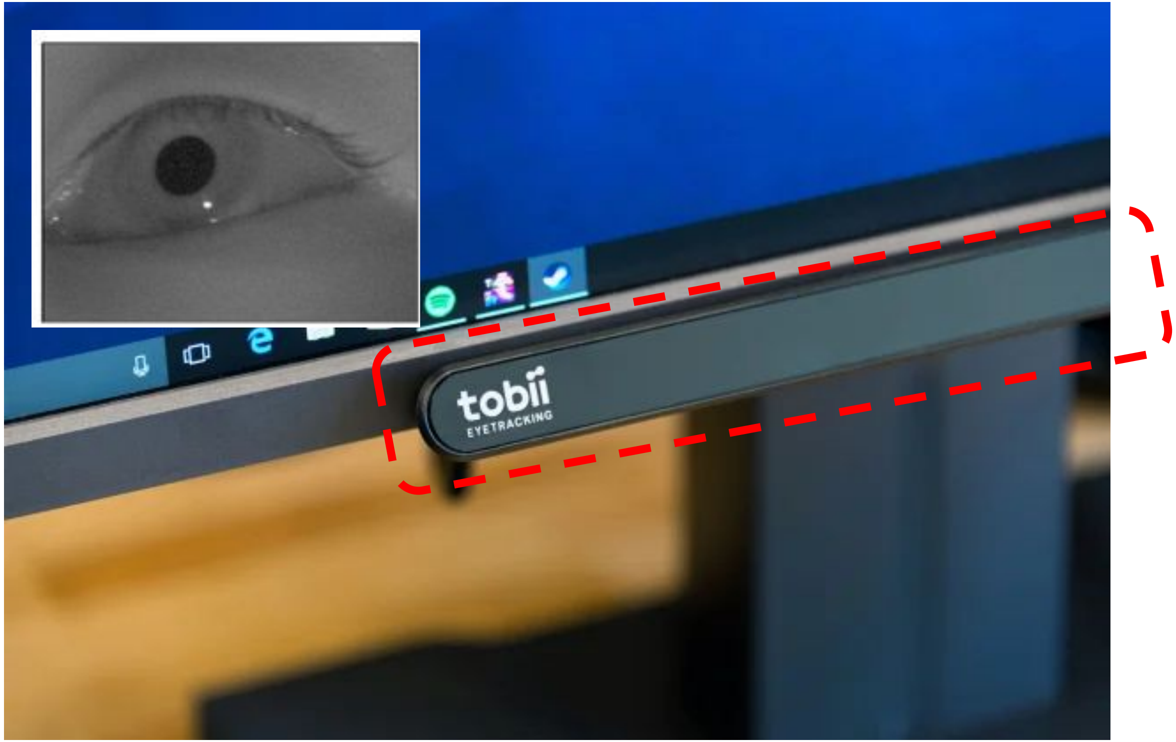
1930s
Psychological study



Background: gaze tracking technology

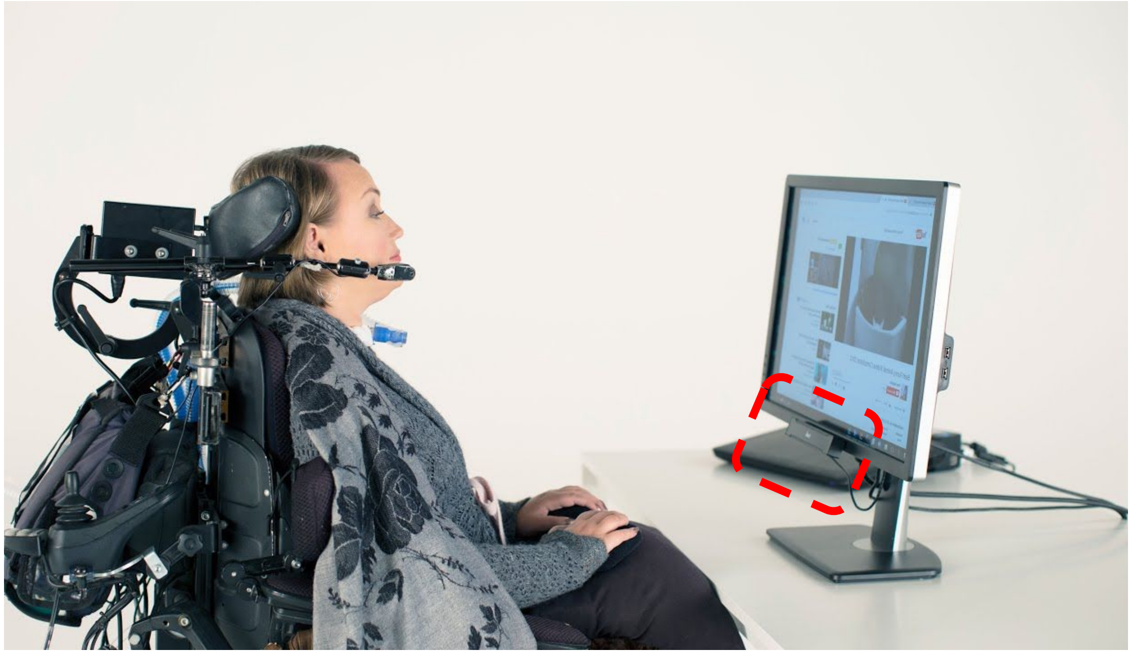
2017
Tobii Eye Tracker

Infrared sensor
< 1cm error



Background: gaze tracking application

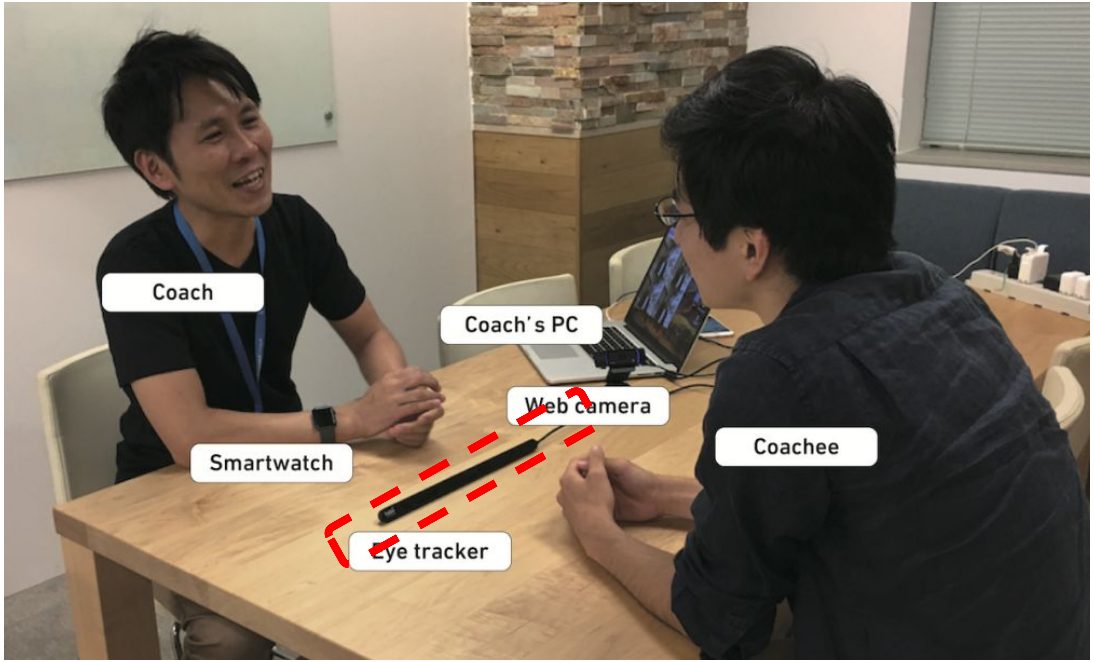
Accessible technology



Background: gaze tracking application

Human behavior analysis

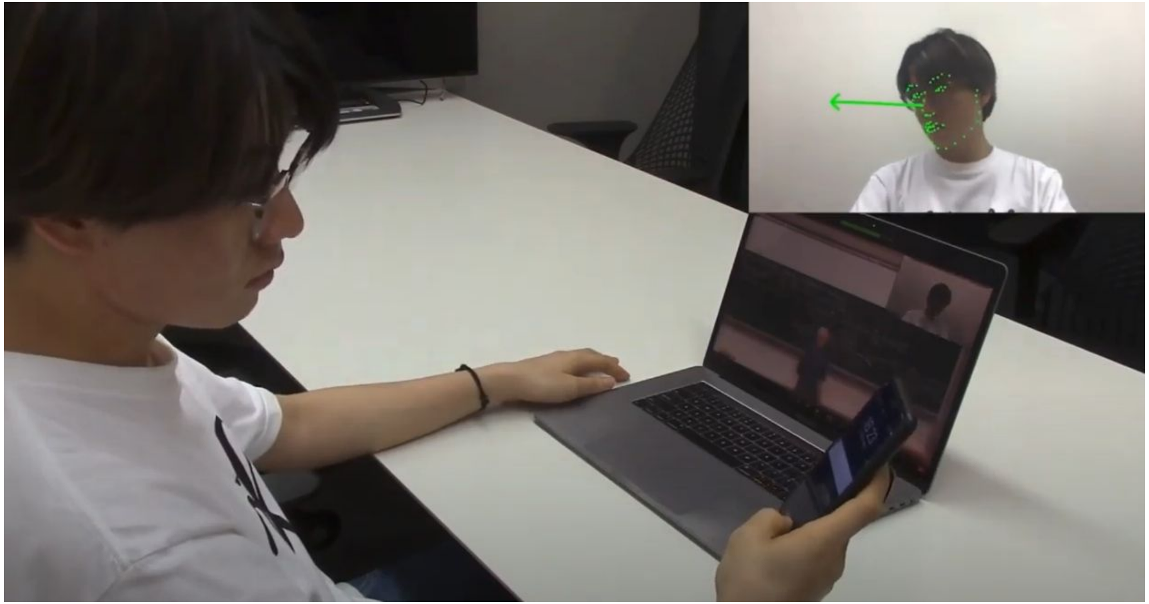
Arakawa and Yakura [CHI’19]



Background: gaze tracking application

Attention-based interaction

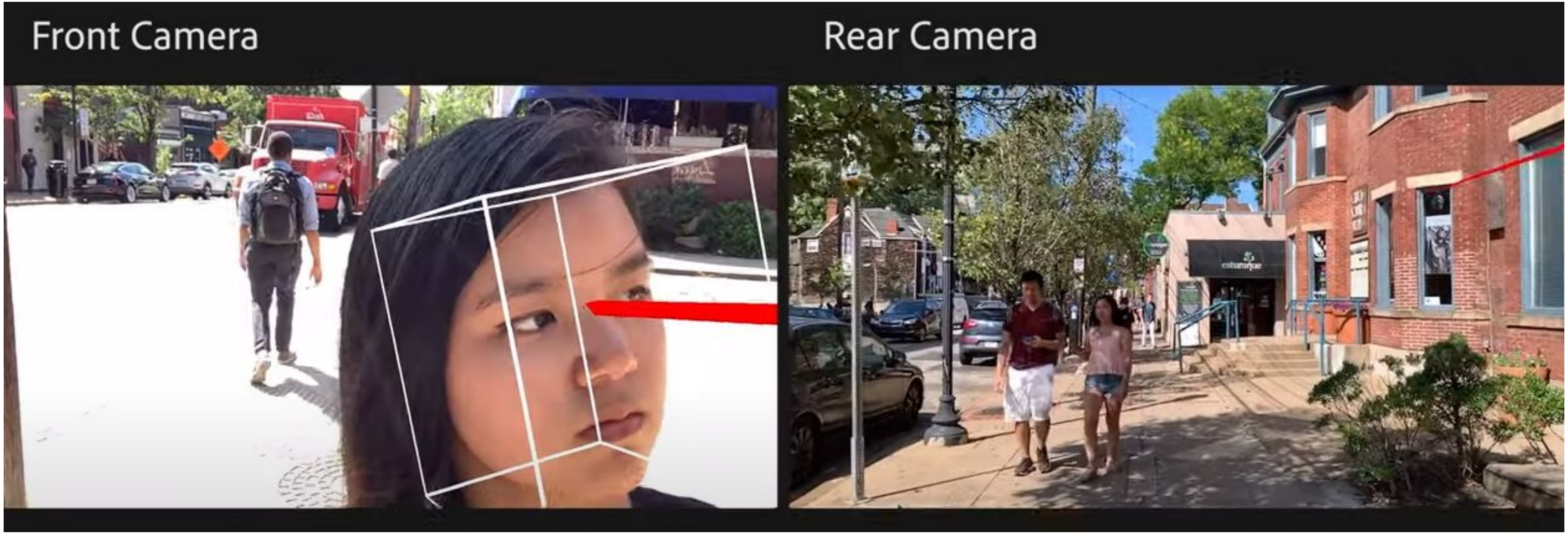
Arakawa and Yakura [CHI'21]



Background: potentials of mobile gaze tracking

- Mobile gaze tracking can support more everyday situations.
    - Inability to use touchscreen with encumbered hands
    - Social interactions using smartphones
    - Novel multimodal interactions between users and smartphones

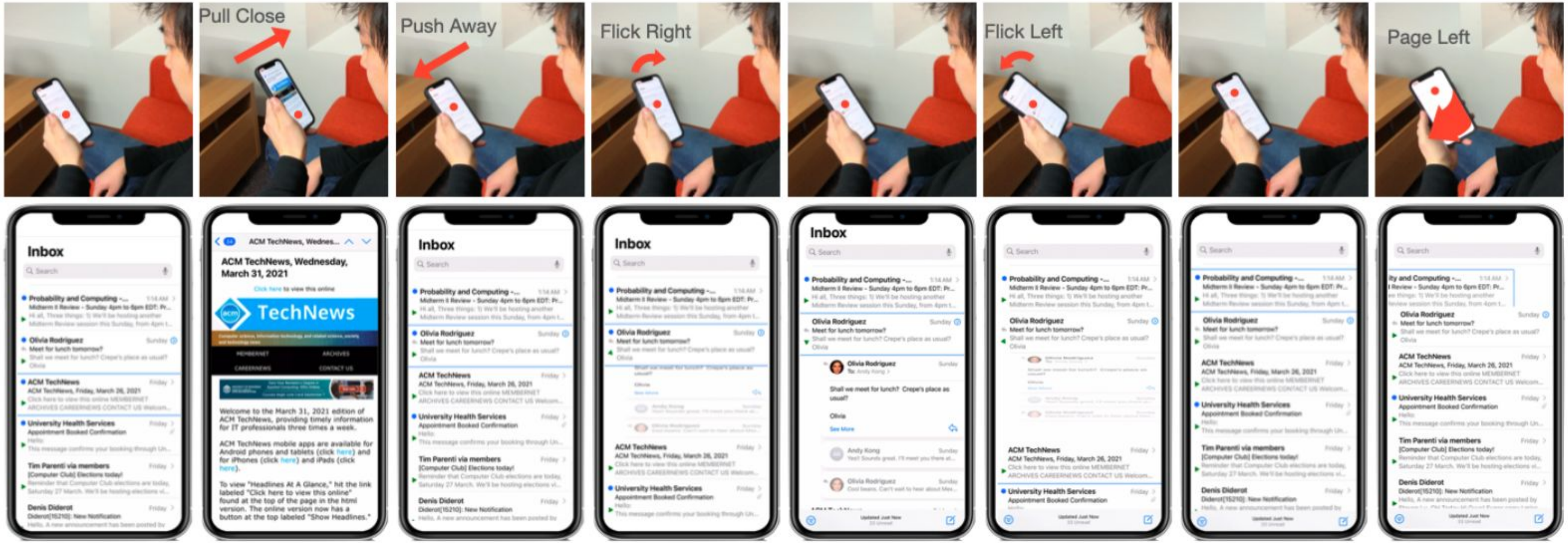
Background: mobile gaze tracking applications in research



Enhance voice assistant with gaze

Mayer et al. [CHI'21]

Background: mobile gaze tracking applications in research



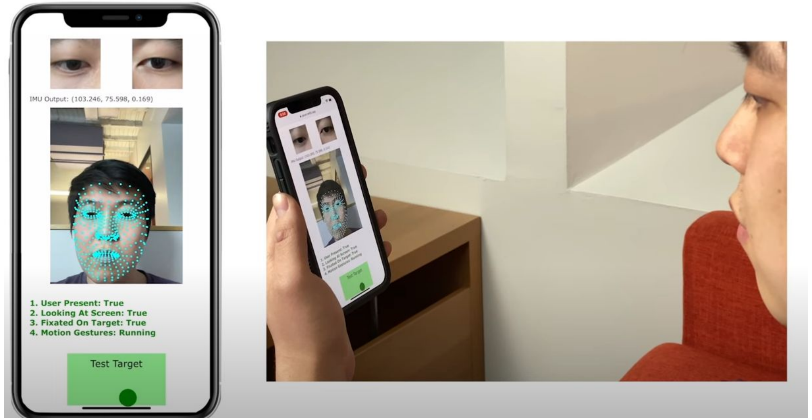
Combined use of gaze and hand gestures

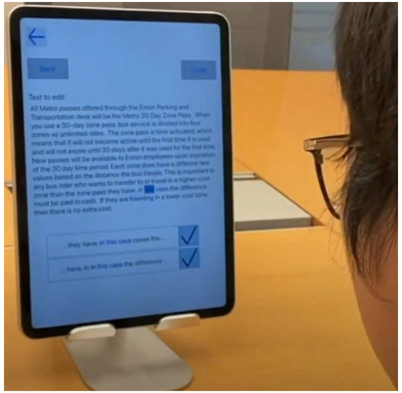
Kong et al. [ICMI'21]

Background: why is mobile gaze tracking not common?

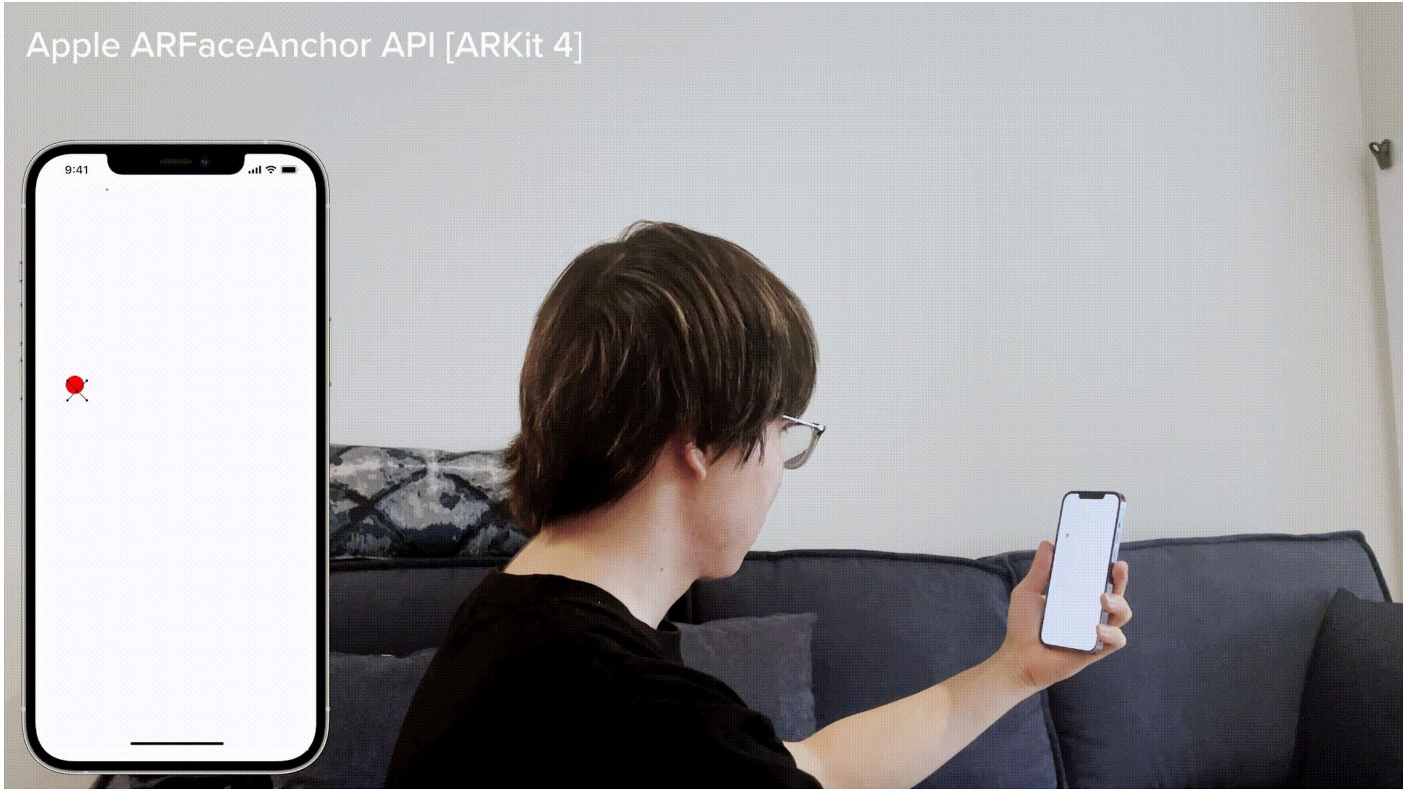
These mobile gaze tracking works require

- Per-user calibration
- Constrained scenarios



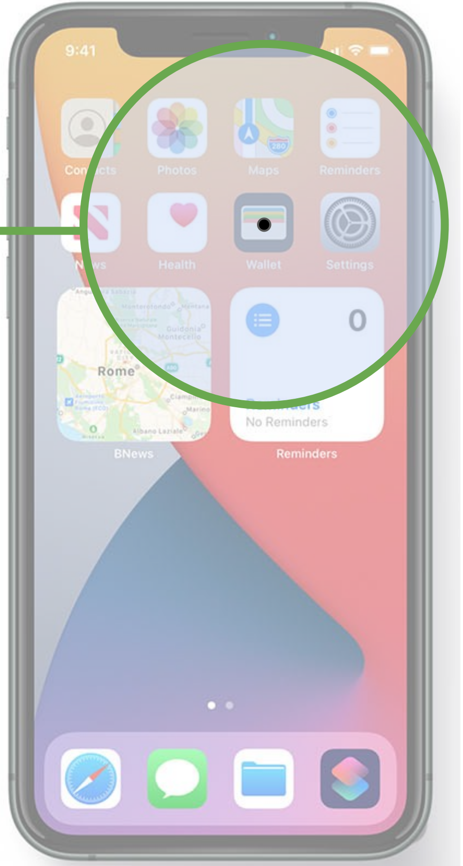


Background: insufficient accuracy in mobile gaze tracking



Background: insufficient accuracy in mobile gaze tracking

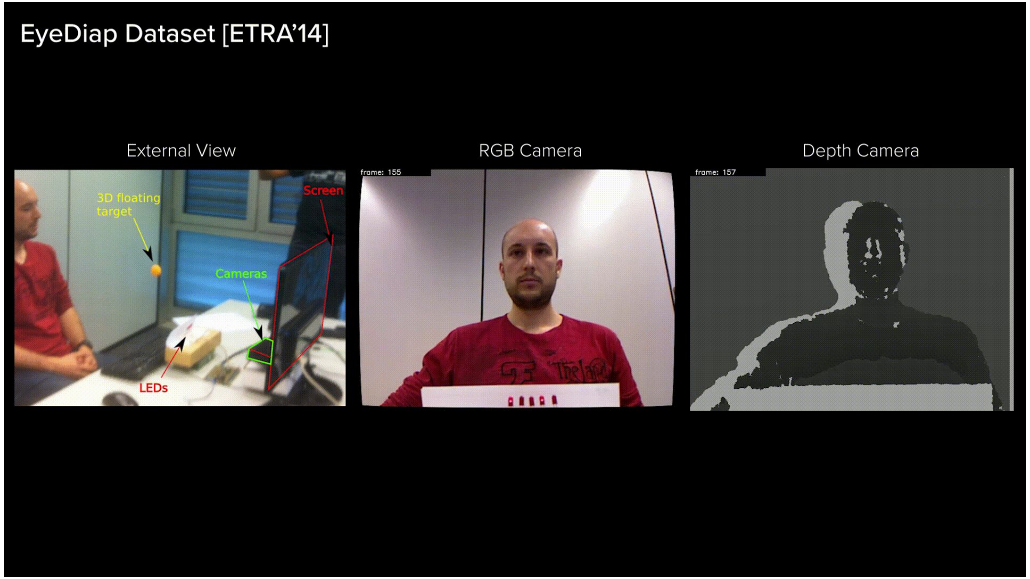
ARKit prediction range



What’s needed for mobile gaze tracking?

- High accuracy
- Calibration-free
- Capable of working in unconstrained situations

Related Work: effectiveness of depth channel



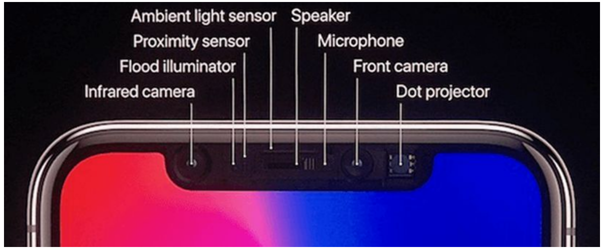
The addition of the depth channel decreased the error by roughly 18%.
Depth provides

- Precise head orientation
- Distance from the screen to the head

Can we leverage depth sensor on recent mobile devices?

Depth cameras on mobiles

iPhone X



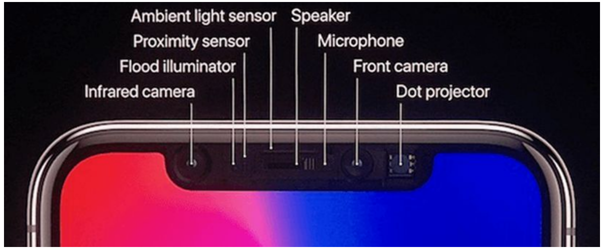
Google Pixel 4

Can we leverage depth sensor on recent mobile devices?

Depth cameras on mobiles

Desktop-grade depth cameras

iPhone X



Google Pixel 4

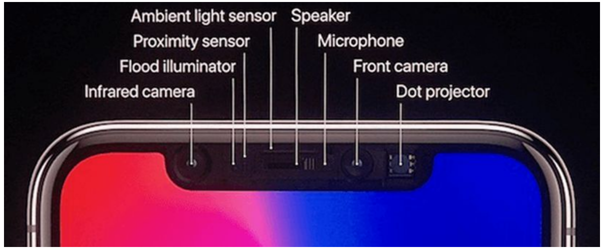
Kinect 2

Can we leverage depth sensor on recent mobile devices?

Depth cameras on mobiles

Desktop-grade depth cameras

iPhone X



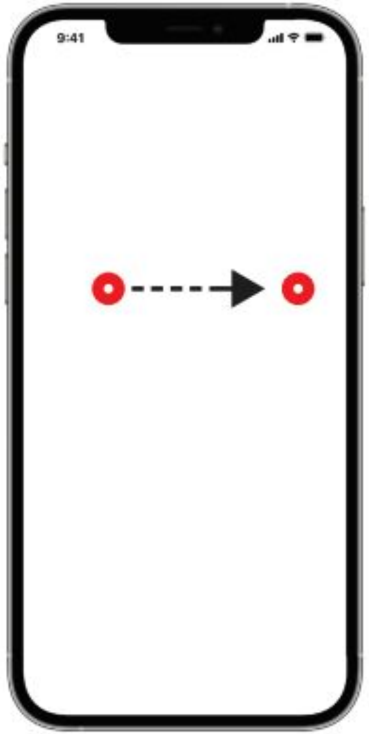
Google Pixel 4

Kinect 2

Our intuition:
Even a coarse depth sensor can provide information of rough head orientation and distance between the screen to the phone in diverse use contexts.

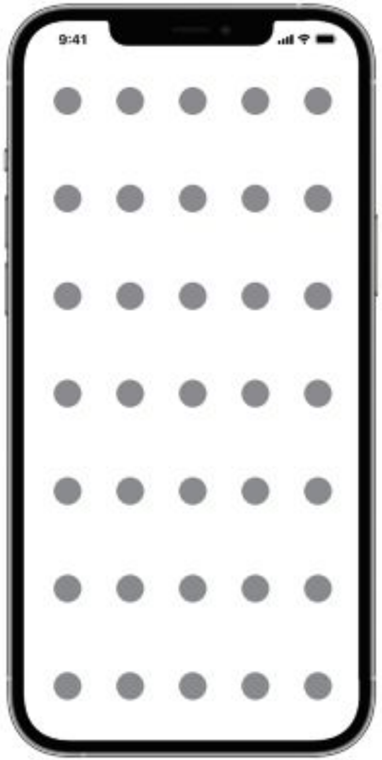
Data Collection

- A dot target moving on the screen
- Record synchronized data at 8 Hz
  - RGB image
  - Depth map
  - ARKit prediction (used for evaluation)
  - (+ motion data from the IMU sensor)



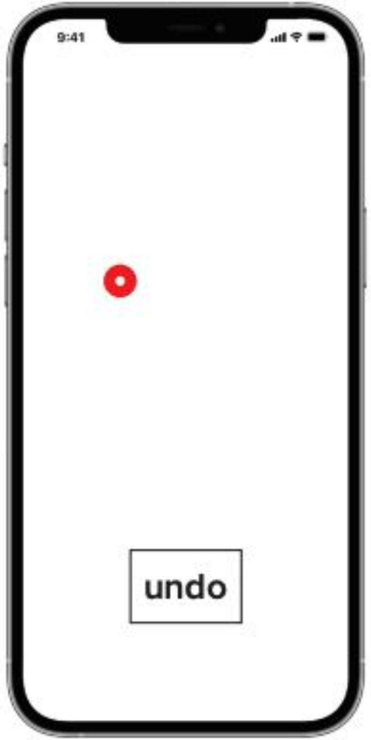
Data Collection

- 7 x 5 locations
- Move from one to another location linearly



Data Collection

- "Undo" function
- Tap screen to stop the animation and jump back to the previous dot location



Data Collection: attention check

- To ensure data reliability (e.g., preventing looking away)
- Participants need to take an action for each dot animation based on its color.

Do nothing

Tap right side

Tap left side

Data Collection: use contexts

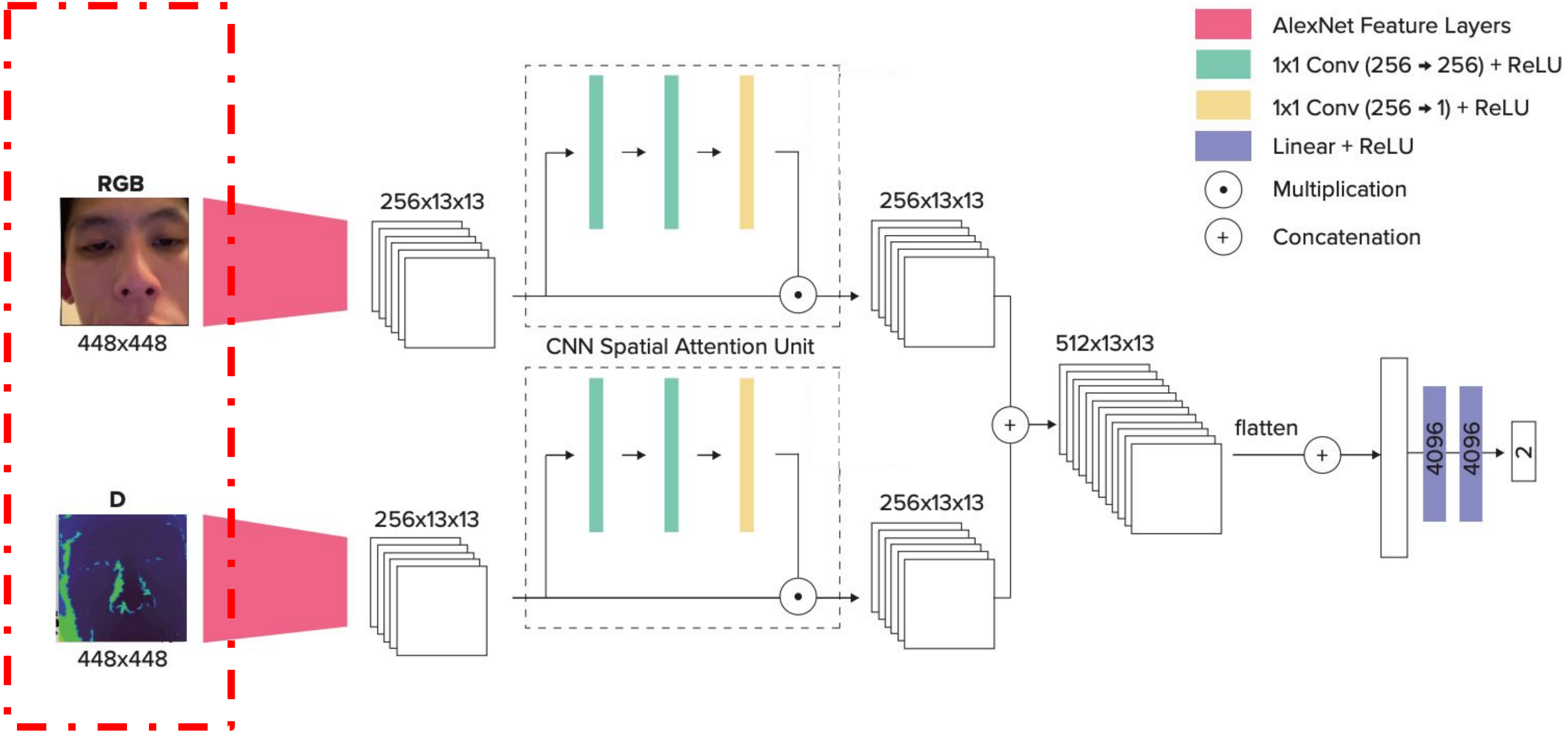
Data Collection: summary

- We did not control the environment.
- Images with blink are automatically detected and removed.

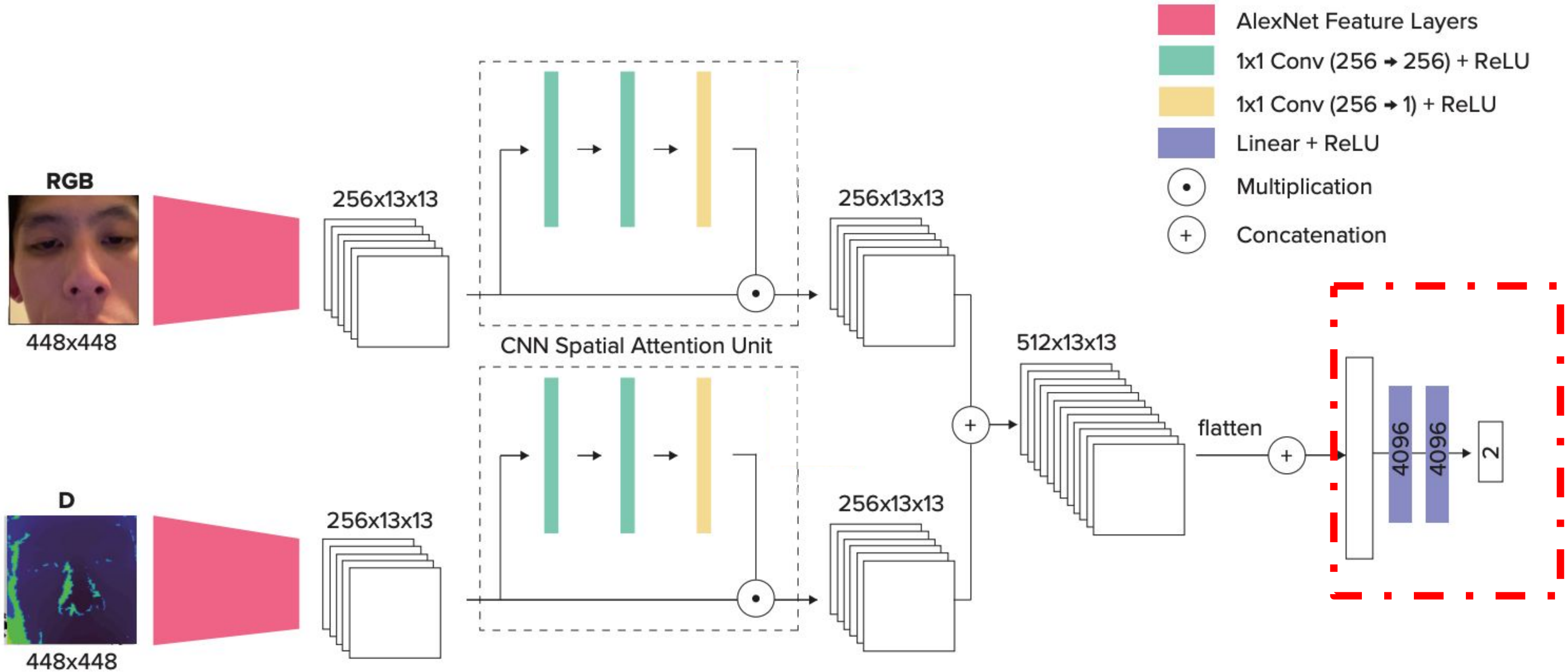
As a result,

- 50 participants
  - 15 different iPhones (>= iPhone X)
  - 14 wore glasses
- 160,120 samples across 4 use contexts

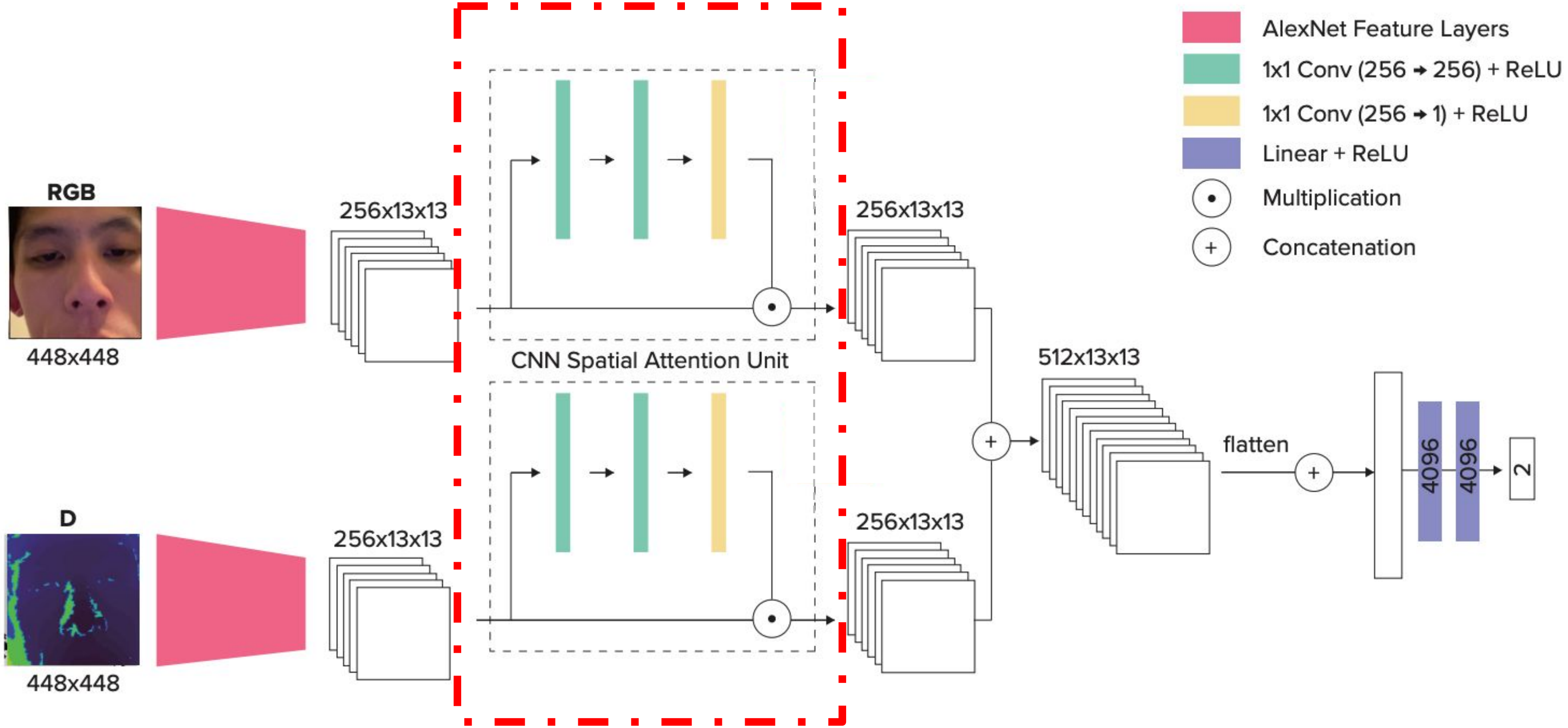
Implementation: model



Implementation: model



Implementation: model



Result: all context

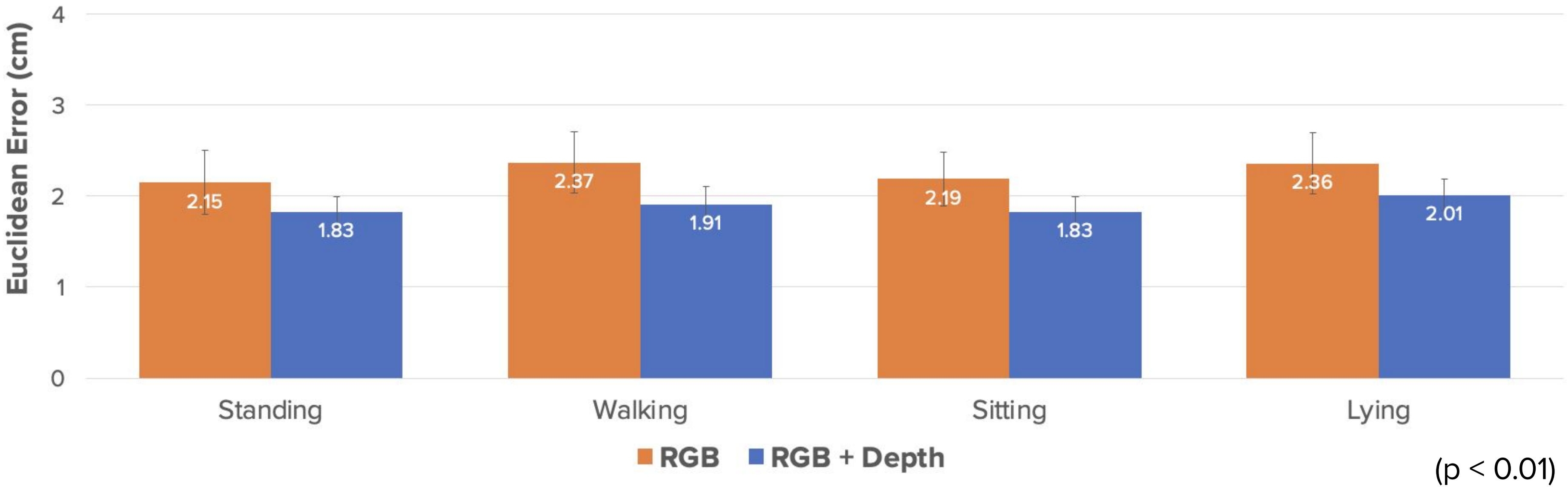
- Leave-one-participant-out (50 participants)
- Compare RGB model and RGB + D model

Result: all context

(p < 0.01)

- **Depth contributed to the error reduction by 16 %**

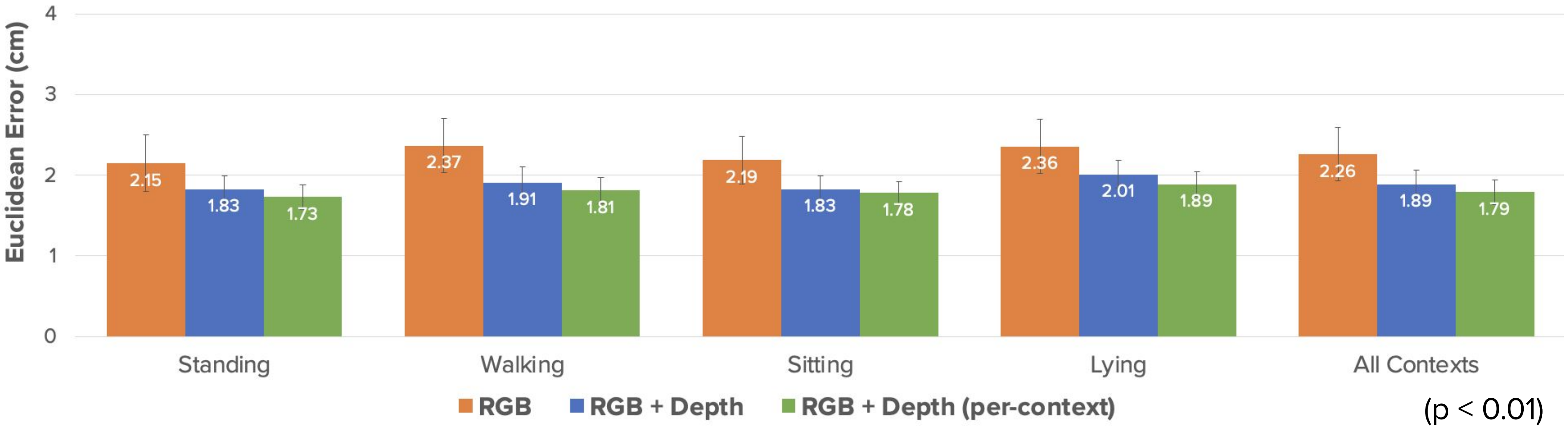
Result: each context



($p < 0.01$)

- **Use context affects the performance.**
- **In every case, the depth helps the model.**

Result: per-context-calibrated model



($p < 0.01$)

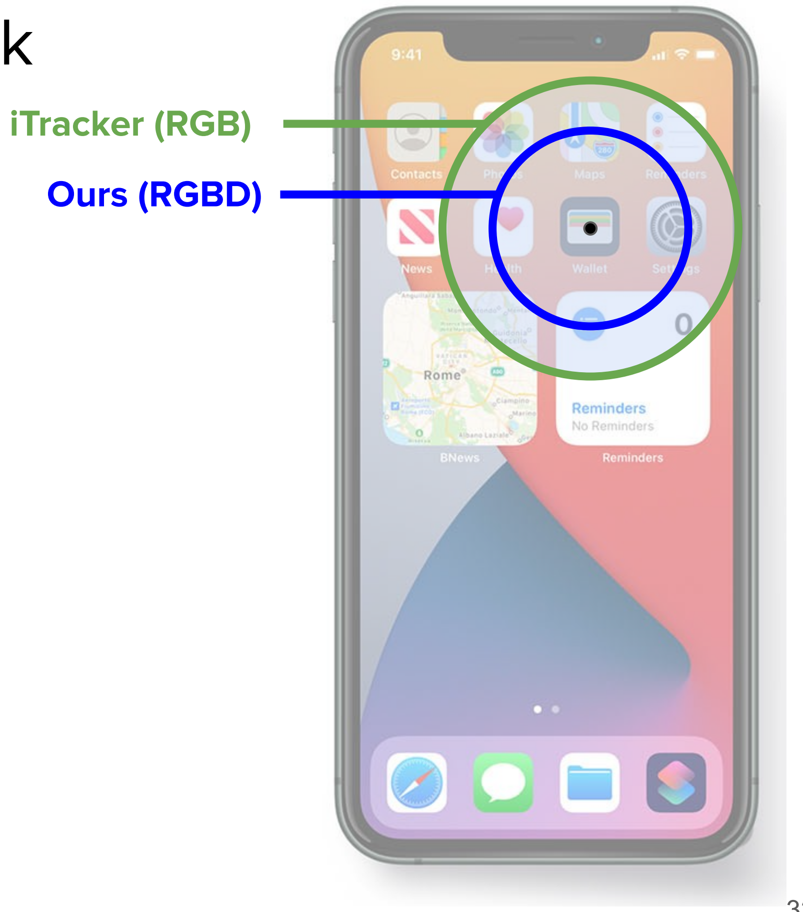
- **Context-specific models perform better than a single "general" model.**
- **Existing activity recognition techniques can boost the tracking performance.**

Result: comparison with prior work

- Our dataset
  - ARKit:
  - iTracker:
  - Our RGB model:
  - Our RGBD model:

Result: comparison with prior work

- Our dataset
  - ARKit: 6.38 cm
  - iTracker: 2.77 cm
  - Our RGB model: 2.26 cm
  - **Our RGBD model: 1.89 cm**



Result: comparison with prior work

- Our dataset
  - ARKit: 6.38 cm
  - iTracker: 2.77 cm
  - **Our RGB model: 2.26 cm**
  - Our RGBD model: 1.89 cm
- GazeCapture dataset
  - **iTracker: 2.04 cm**
  - **Our RGB model: 2.03 cm**

Result: comparison with prior work

- Our dataset
  - ARKit: 6.38 cm
  - iTracker: 2.77 cm
  - **Our RGB model: 2.26 cm**
  - Our RGBD model: 1.89 cm
- GazeCapture dataset
  - iTracker: 2.04 cm
  - **Our RGB model: 2.03 cm**

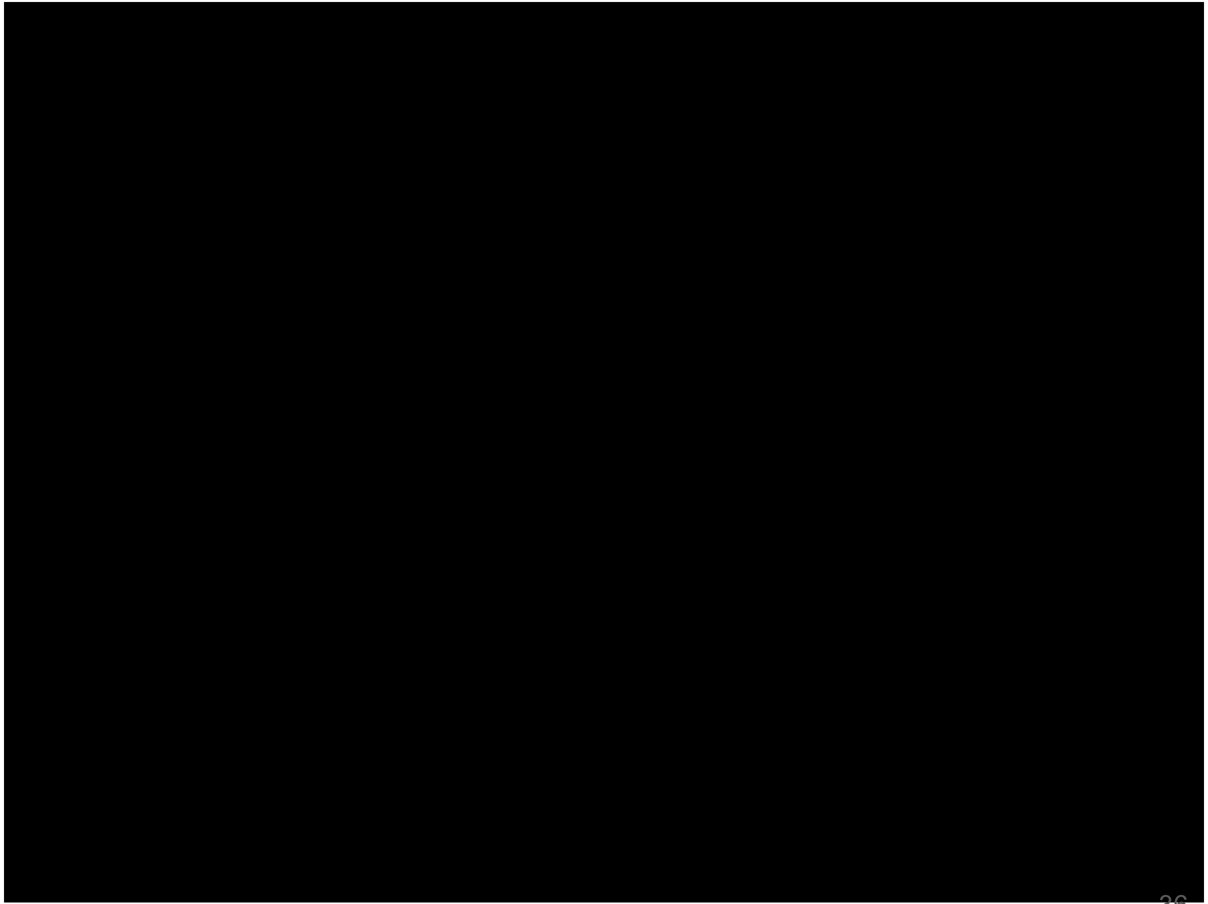
Our dataset has

- Challenging capture scenarios
- Larger screen size

Result: qualitative error analysis

On-device Model

- iPhone 12 Pro Max
- RGBD Model
    - 7 FPS (121.3 ms latency)
- RGB Model
    - 10 FPS (85.3 ms latency)



Summary

- We collected a dataset of mobile **RGB + Depth (RGBD)** gaze tracking.
  - 50 participants
  - 4 use contexts: standing, walking, sitting, lying
- Our RGBD model outperformed existing systems.
  - Adding depth channel **reduced the error by 16.3%** (RGB: 2.26 cm, RGBD: 1.89 cm)
- We developed the on-device system to enable real-time interactions.

Limitation and Future Work

- Further improvement of accuracy
  - Larger scale data collection
  - IMU sensor fusion

Limitation and Future Work

- Diverse data collection on crowd-sourcing
    - More contexts: climbing stairs, biking, etc

GazeCapture dataset

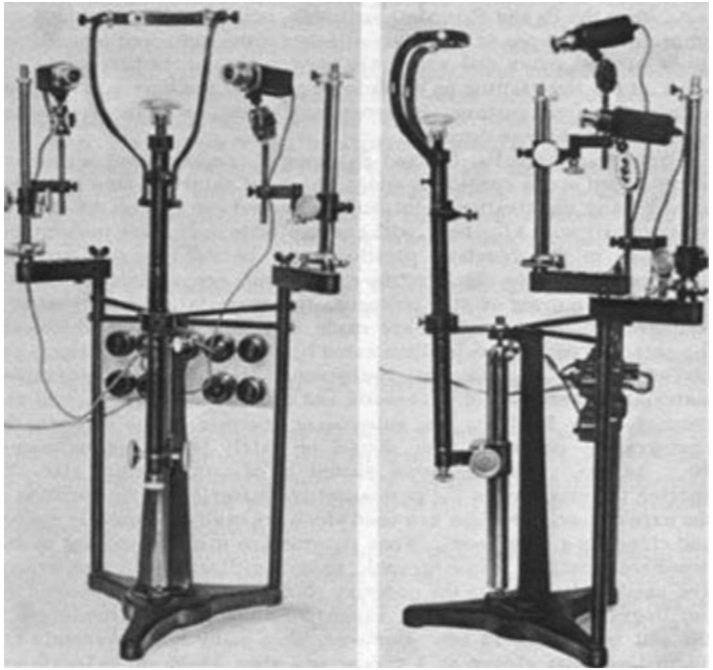
RGBDGaze: Gaze Tracking on Smartphones with RGB and Depth Data

Riku Arakawa

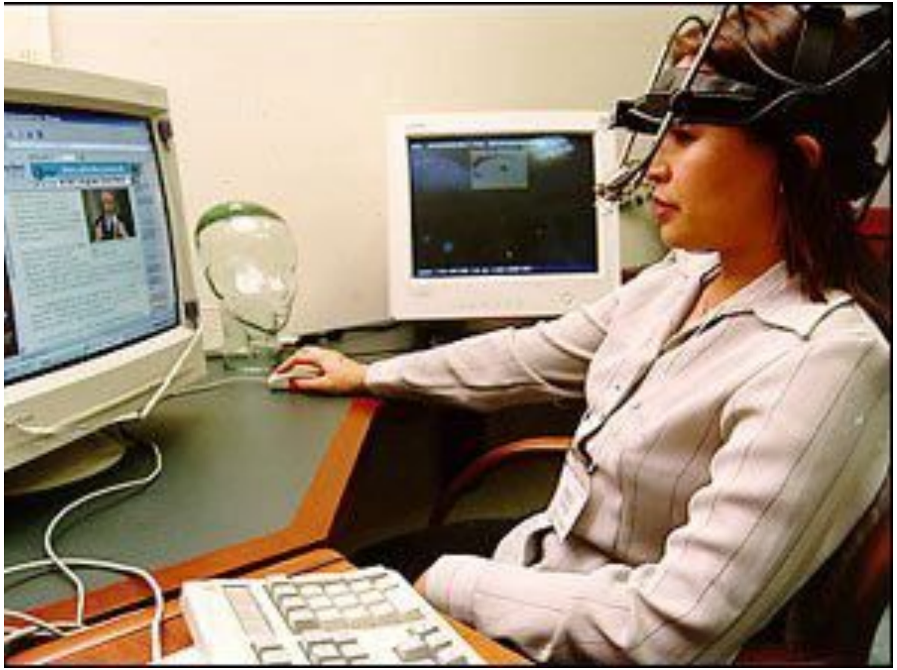
Dataset and source code: https://github.com/FIGLAB/RGBDGaze

Appendix

Background: gaze tracking technology



1960s
Chin rest



1990s
Wearable

Implementation: training protocol

- PyTorch 1.9.1
- Batch Size: 16
- SGD Optimizer
    - Initial learning rate: 5e-4
    - Momentum: 0.9
    - Weight decay: 1e-4
- Loss function: mean squared error
- 20 epochs
- Leave-one-participant-out evaluation
    - 12 hours x 50 participants

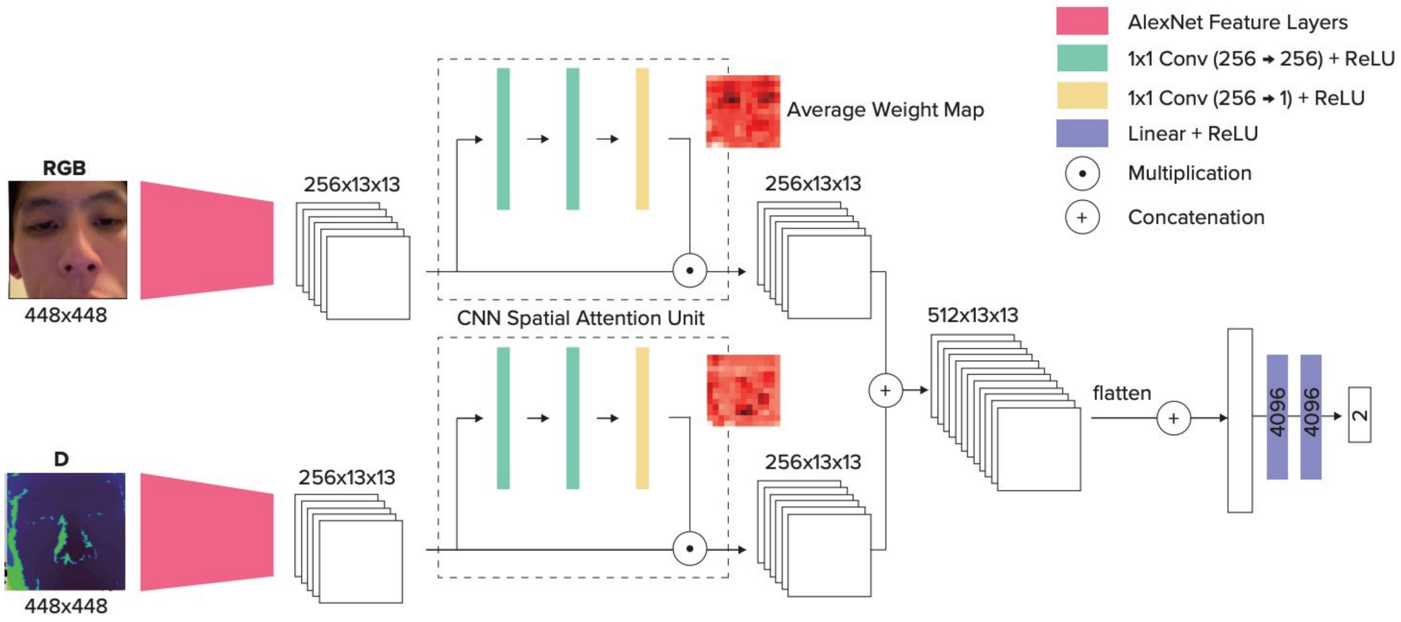
Result: comparison with prior work

- Our dataset
  - ARKit: 6.38 cm
  - iTracker: 2.77 cm
  - Our RGB model: 2.26 cm
  - **Our RGBD model: 1.89 cm**
- Other mobile gaze tracking systems (Different dataset)
  - Tablet Gaze: 3.17 cm
  - EyeTab: 2.58 cm

Related Work

System	Capture Modality		Mobile Device	Unconstrained Study	Calibration –Free	Mean Gaze Error
	RGB	Depth				
Columbia Gaze [37]	✓					-
UT MultiView [39]	✓				✓	6.5°
ETH-XGaze [51]	✓				✓	4.7°
MPII Gaze [52]	✓			✓	✓	6.3°
RT-GENE [14]	✓			✓	✓	7.7°
Gaze360 [21]	✓			✓	✓	13.5°
Wang and Ji [45]	✓	✓		✓		4.0°
Zhou et al. [55]	✓	✓		✓		1.99°
EyeDiap [34]	✓	✓		✓	✓	8.1°
ShanghaiTechGaze+ [30]	✓	✓		✓	✓	3.87 cm
EyeTab [47]	✓		✓		✓	2.58 cm
Valliappan et al. [44]	✓		✓	✓		0.46 cm
EyeMU [24]	✓		✓	✓		1.7 cm
iTracker [25]	✓		✓	✓		1.34 cm
iMon [18]	✓		✓	✓	✓	1.57 cm
TabletGaze [17]	✓		✓	✓	✓	3.17 cm
iTracker [25]	✓		✓	✓	✓	2.77 cm
Apple ARKit [7]	✓		✓	✓	✓	6.38 cm
Our System	**✓**	**✓**	**✓**	**✓**	**✓**	**1.89 cm**

Result


RGB
448x448
D
448x448
256x13x13
256x13x13
CNN Spatial Attention Unit
Average Weight Map
256x13x13
256x13x13
512x13x13
flatten
4096
4096
2
AlexNet Feature Layers
1x1 Conv (256 → 256) + ReLU
1x1 Conv (256 → 1) + ReLU
Linear + ReLU
Multiplication
Concatenation

Result