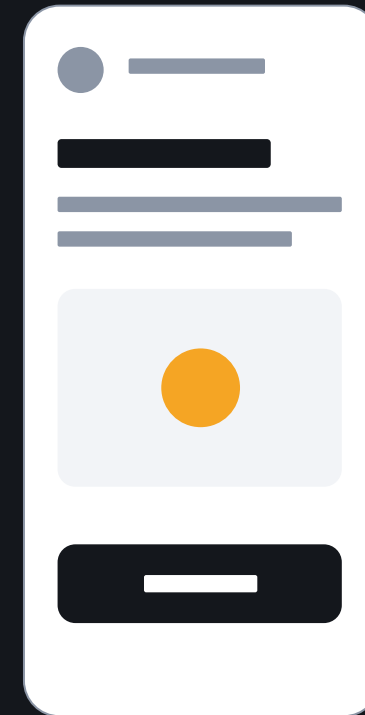


ChromaEyes

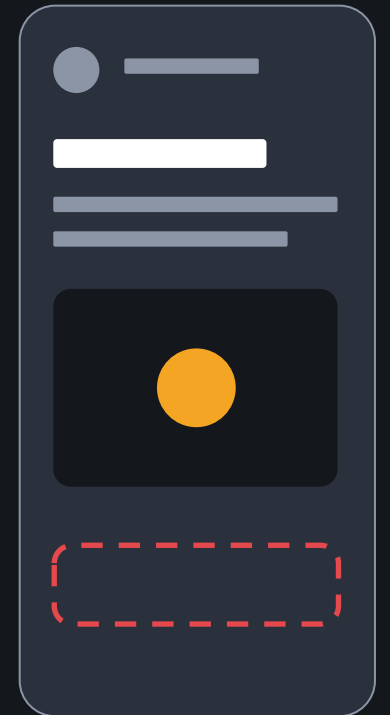
Detecting Inconsistencies of User Interface
Elements between Light and Dark Modes of Web
Applications

K C Shweta¹ · Byungchul Tak² · Tegawendé F. Bissyandé³ · Dongsun Kim¹

¹ Korea University ² Kyungpook National University ³ University of Luxembourg



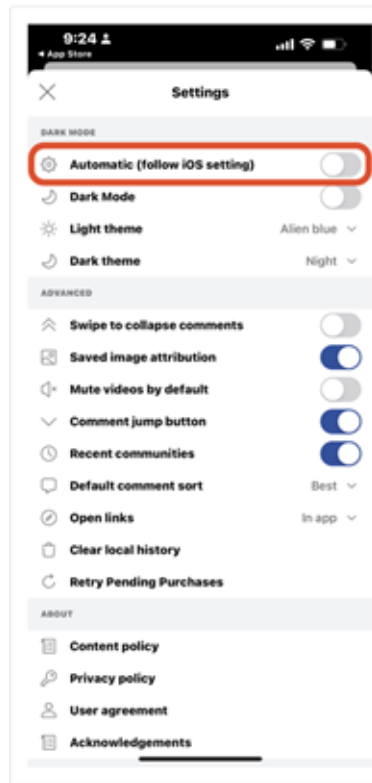
Light mode



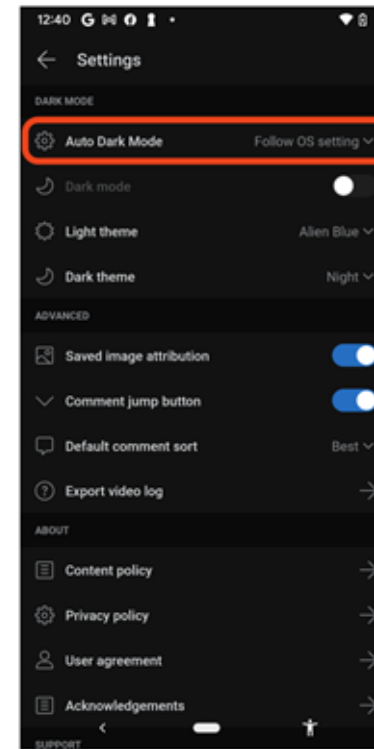
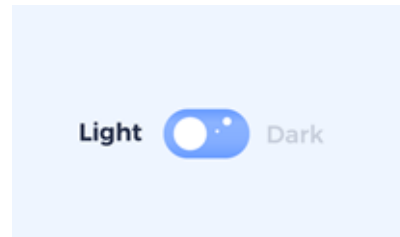
Dark mode

MOTIVATION

What is Dark Mode?



Light mode — dark text on a light background



Dark mode — light text on a dark background

Prevalence of Dark Mode, and Its Fragility

92% of software engineers prefer dark mode in their development environment

82.7% of mobile users keep dark mode as their system default

60% reduced bounce rate after one site (Terra) added a dark theme

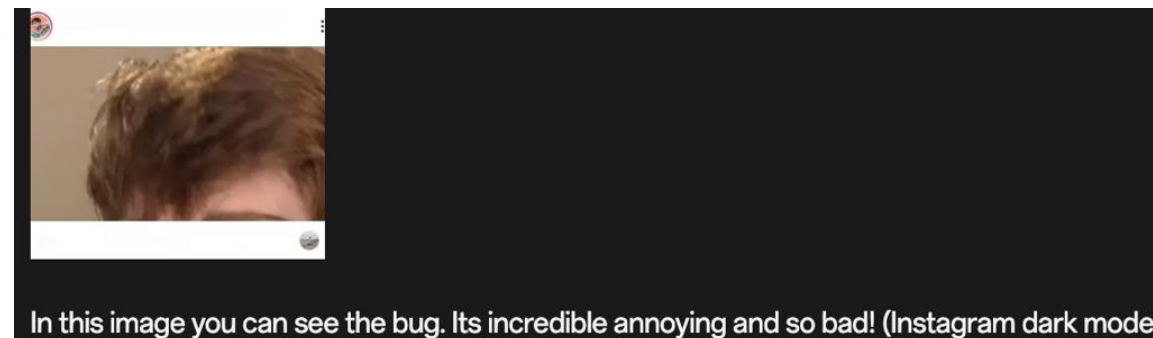
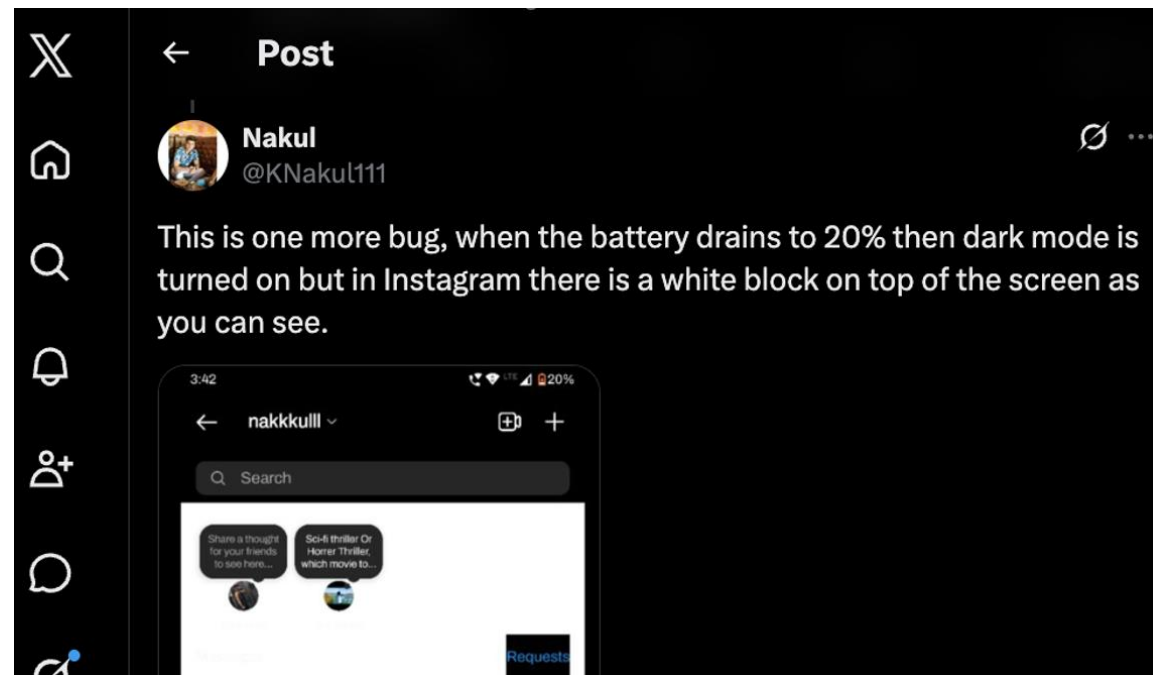
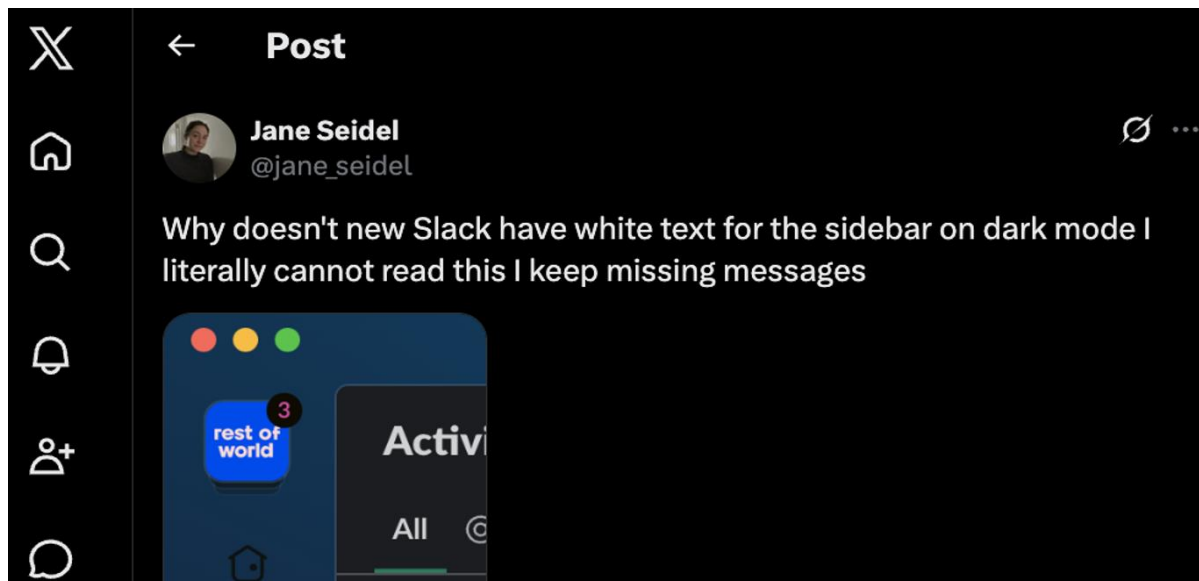
Drift Issue of Dark Mode Design

- Built by hand or by a browser extension, the dark version often drifts from the light one.
- No automatic checking is available, yet.

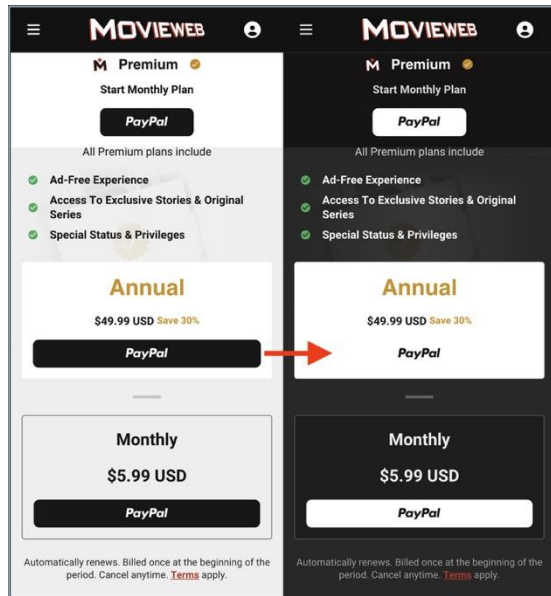
We examined **213** real-world complaints: 104 GitHub issues, 54 X/Reddit posts, 55 dark-mode extension reviews.

Sources as cited in the paper: forms.app and thesmallbusinessblog.com (2025); web.dev Terra case study (2021).

Complaints About Inconsistency Issues

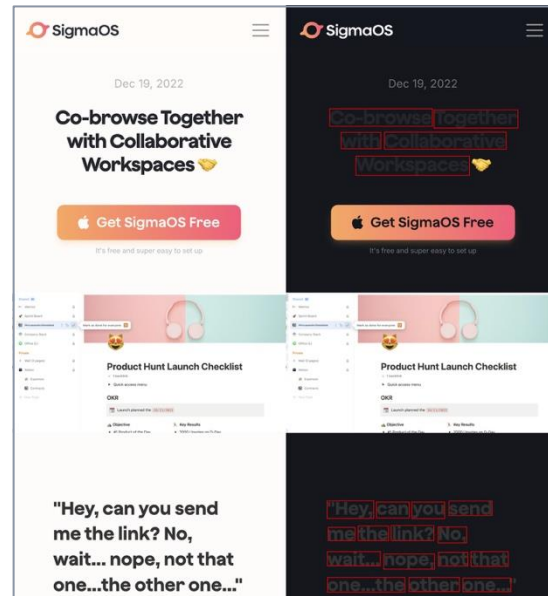


Four Ways Dark Mode Goes Wrong



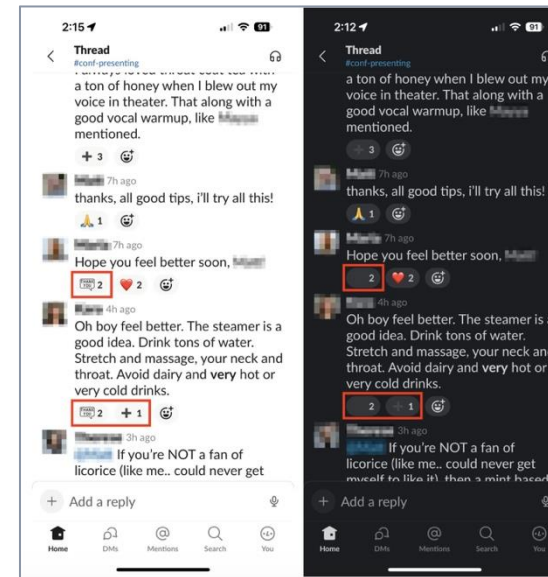
1 Layout

A button's outline vanishes



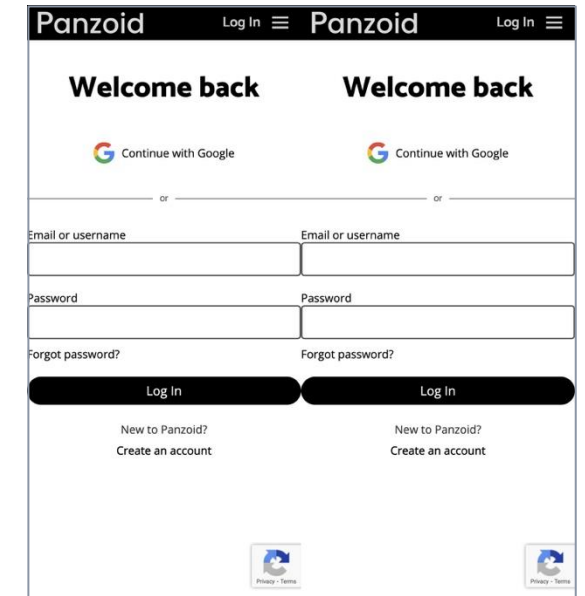
2 Text

A heading turns unreadable



3 Icons (Objects)

Reaction icons disappear



4 Incomplete conversion

The page simply stays light

Inconsistency: an element that is visible or functional in one mode becomes invisible, illegible, misplaced or functionally different in the other, beyond the expected change of color and contrast.

Visual Similarity Does Not Imply Consistency

A correct conversion



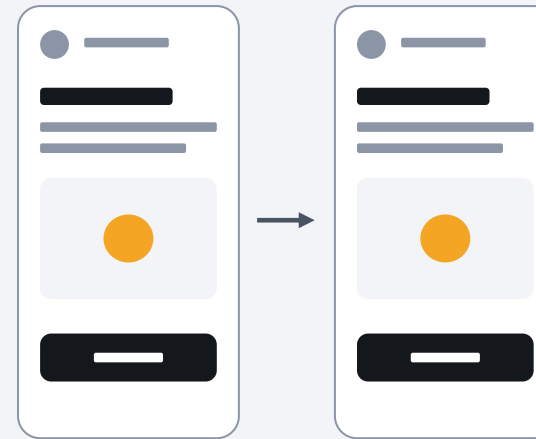
Light

Dark

**Many pixels
change during
a correct
conversion**

A pixel diff flags the whole screen.

A conversion that never happened



Light

"Dark"

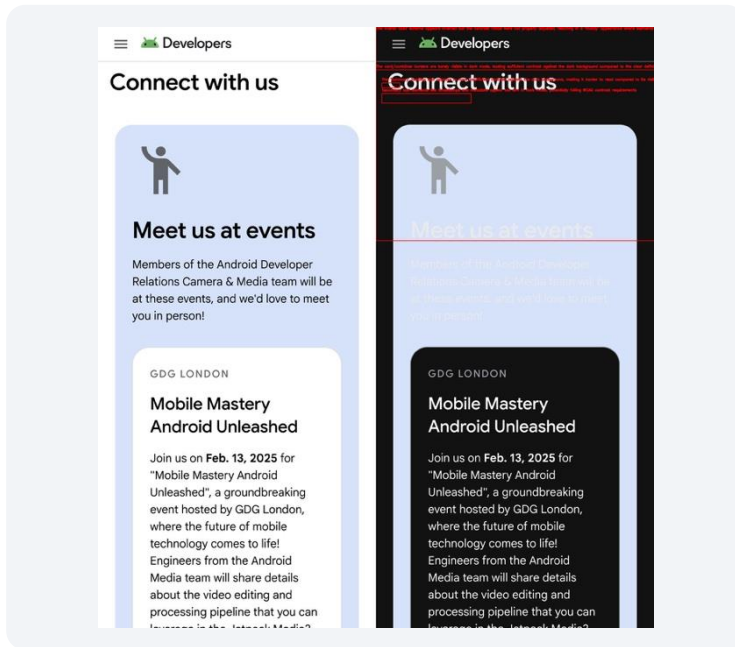
**No pixel
changed**

A pixel diff reports a perfect match. The page is broken.

**Visual similarity is not the criterion for inconsistency.
Instead, we check this: Does the conversion preserve visible structure and elements?**

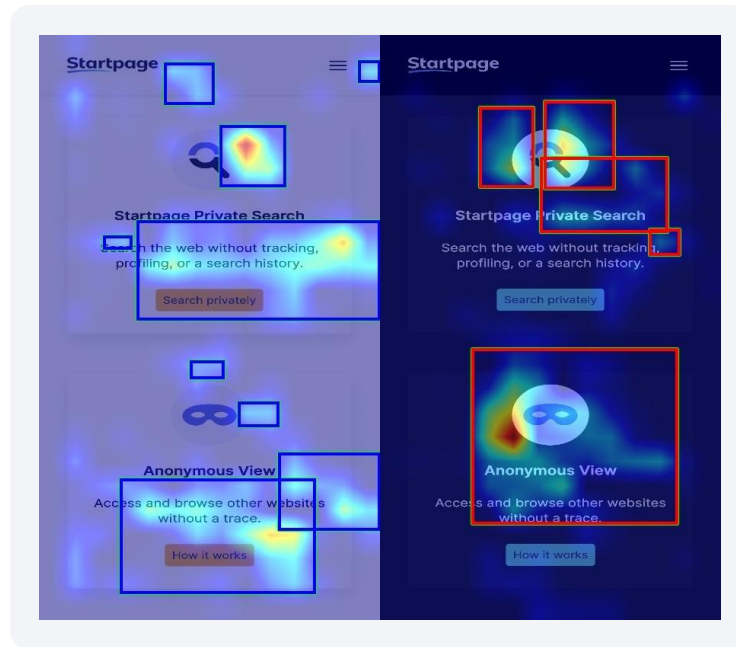
Off-the-shelf Tools Are Ineffective

Vision-language models (VLMs)



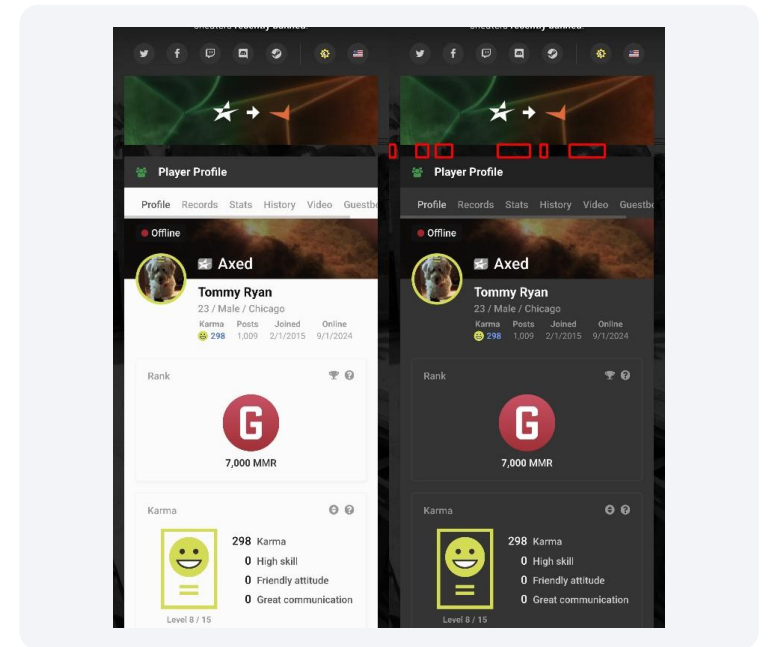
The verdict may be right, but the boxes land in the wrong place.

Accessibility checkers



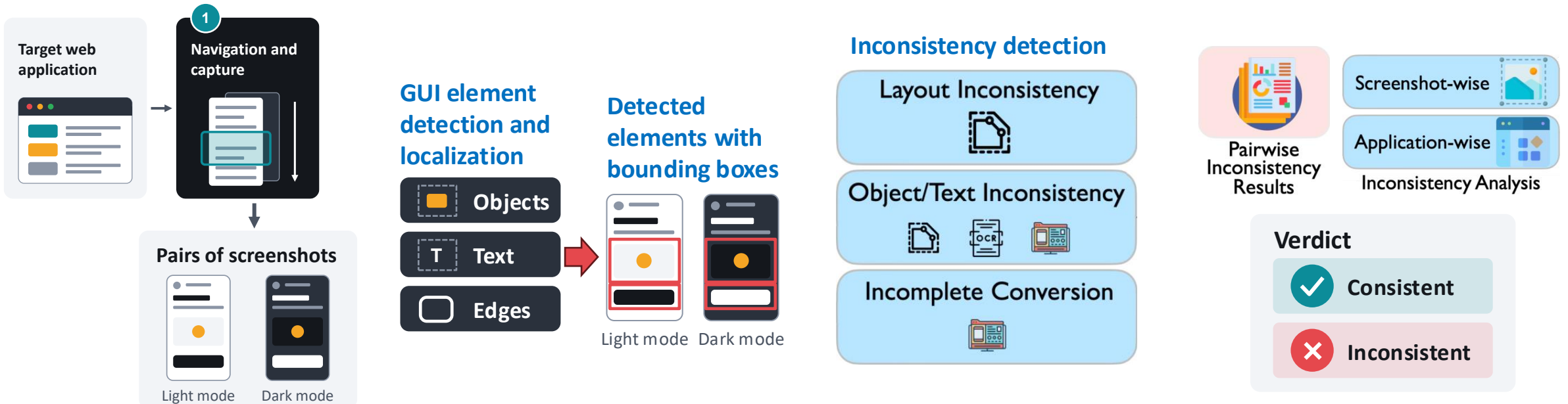
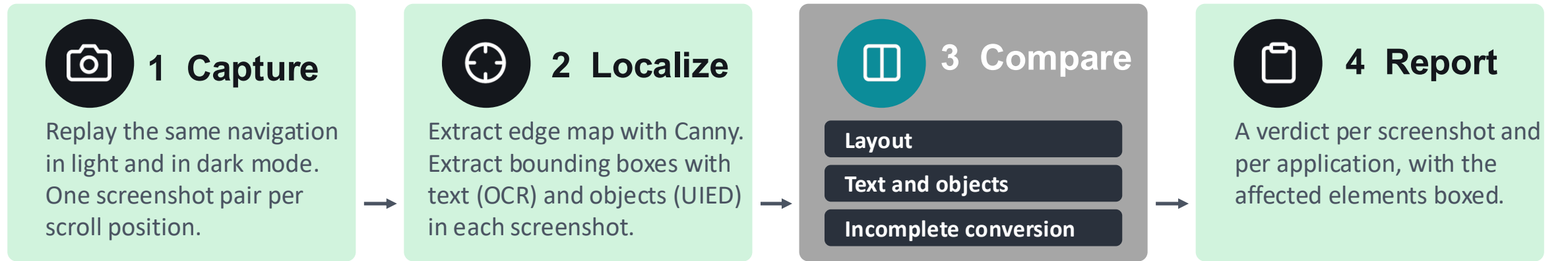
Built for one mode at a time. Here **OwlEye** raises issues on a pair that is actually consistent.

Naive element matching



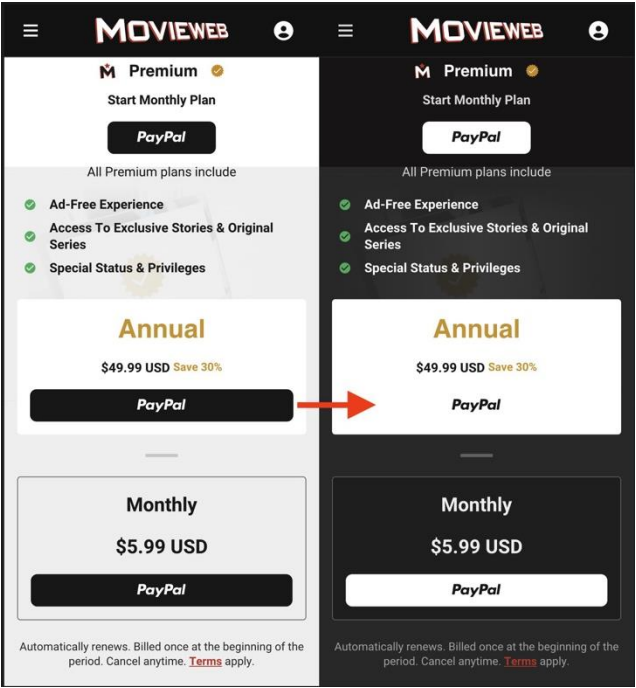
OCR and object detectors are unstable across modes
→ a consistent pair gets flagged

ChromaEyes Methodology

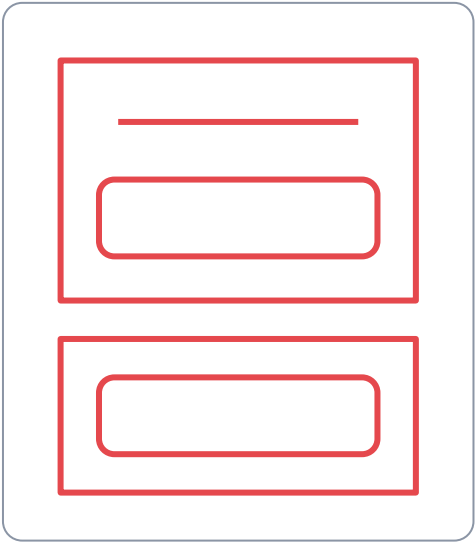


Layout Inconsistency Detection

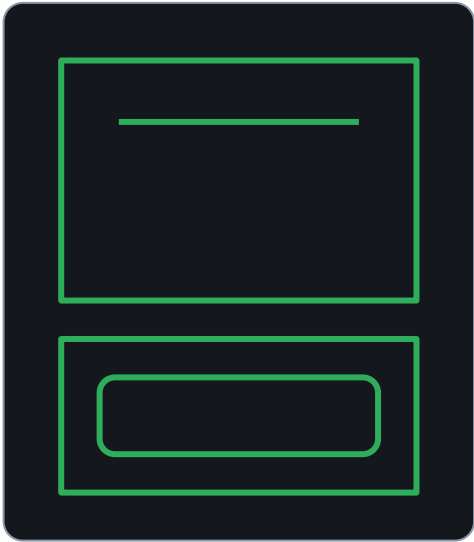
- only in light mode
- only in dark mode
- in both modes



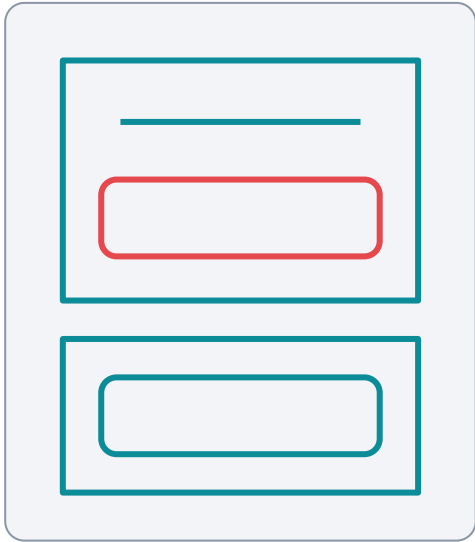
Light-mode edges



Dark-mode edges



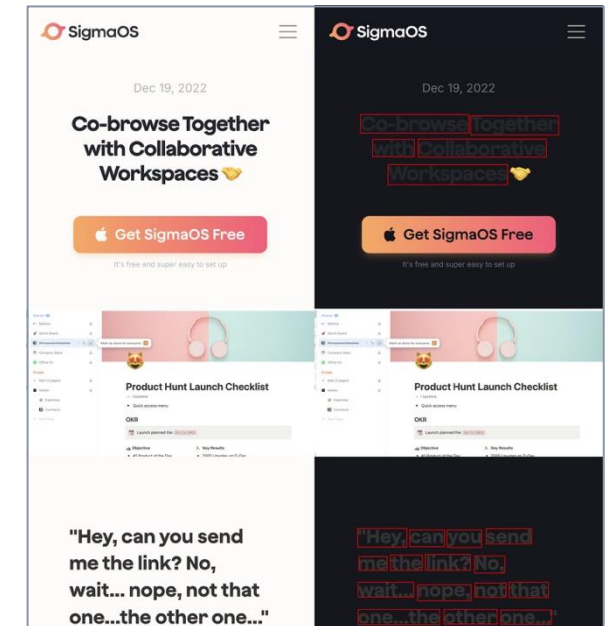
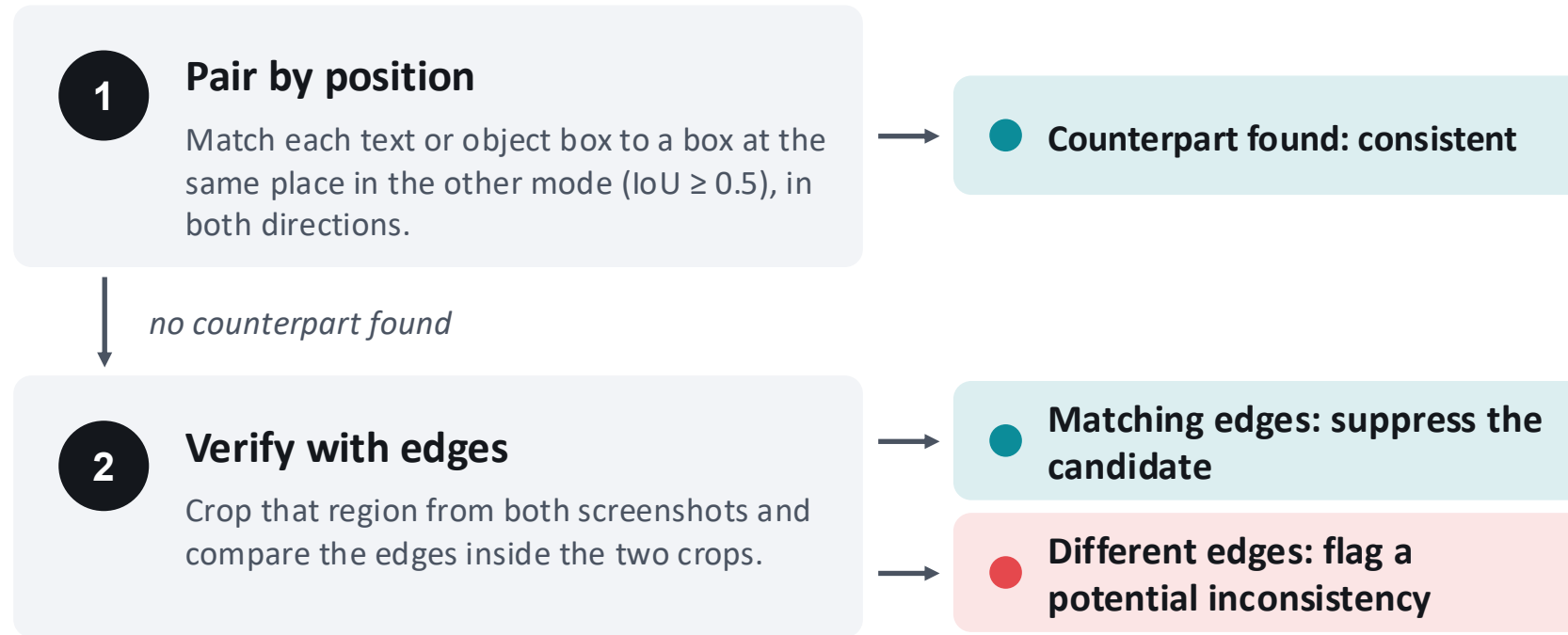
Overlay



Real case: the PayPal button loses its outline in dark mode.

Any red or green left in the overlay trigger a layout-inconsistency flag

Text and Objects Inconsistency Detection



Real case: headings found in light mode only, edges differ, so the pair is flagged.

OCR and object detectors often see an element in one mode and miss it in the other. Edge verification reduces false positives from unmatched detections.

Incomplete Conversion Detection

Real case: Panzoid

Light mode

Dark mode, switched on

Panzoid

Log In

≡

Panzoid

Log In

≡

Welcome back

Welcome back

 Continue with Google

 Continue with Google

or

or

Email or username

Email or username

Password

Password

Forgot password?

Forgot password?

Log In

Log In

 RGB 255, 255, 255

 RGB 255, 255, 255

● **Unchanged: incomplete conversion**

- 1 Mask every detected element. What remains is the background.
- 2 Take the dominant background color in each mode.
- 3 Judge the dark-mode background:

Example: light → dark	The dark-mode background is...	Verdict
 → 	Still light R, G and B all at least 250	● Incomplete conversion
 = 	The same as in light mode Identical color in both modes	● Incomplete conversion
 → 	Anything else The background color changed	● Converted

colDiff = 80/250 (Material Design guidance). Pages that are dark in light mode (RGB ≤ 80) are skipped.

Other tools look for what is missing.
This detector looks for what failed to change.

Research Questions in the Evaluation

RQ1

Detection

How effective is ChromaEyes at detecting inconsistencies in real web applications?

147 apps · 1,470 screenshot pairs

RQ2

Comparison

Does it outperform vision-language models and accessibility checkers?

4 VLMs and 2 accessibility checkers

RQ3

Dark-mode extensions

How effective is it when a browser extension generates the dark mode?

7 sites × 7 extensions · 539 pairs

196

apps: 147 with native dark mode, 49 converted by extensions

2,009

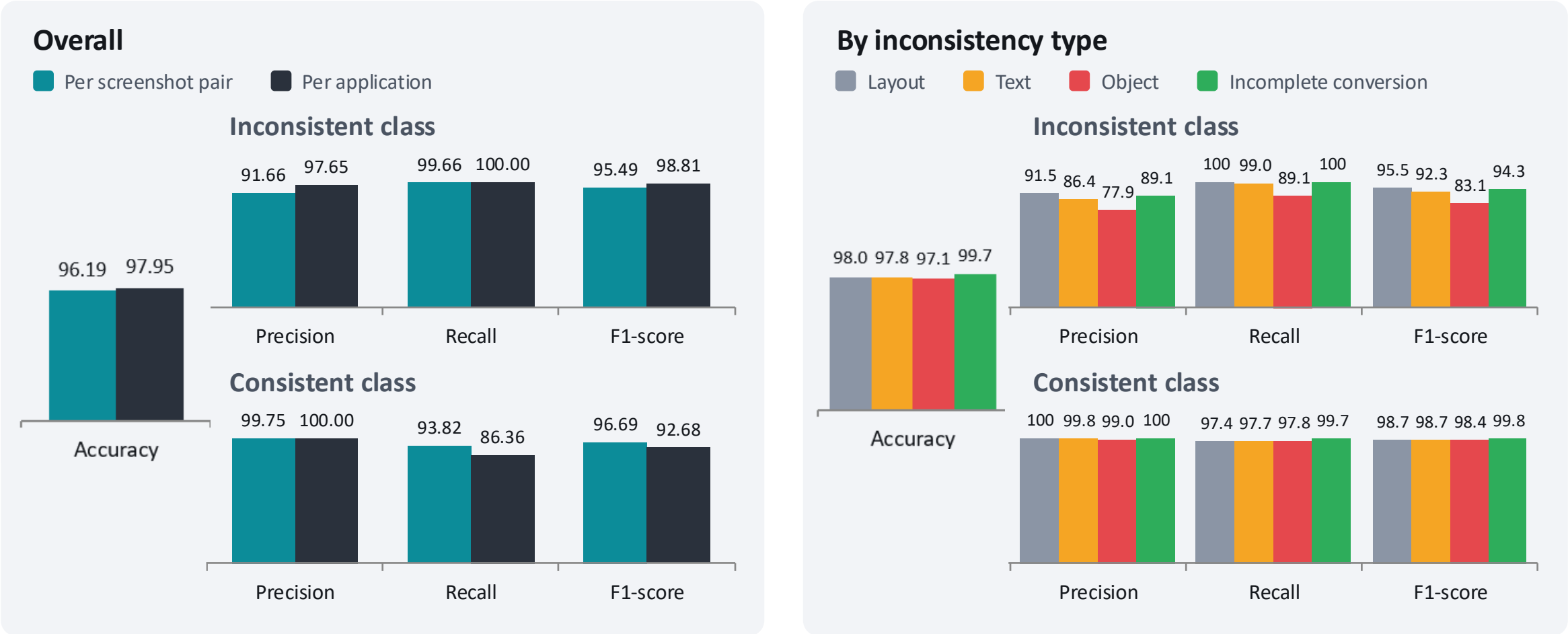
screenshot pairs, labeled consistent or inconsistent

0.81

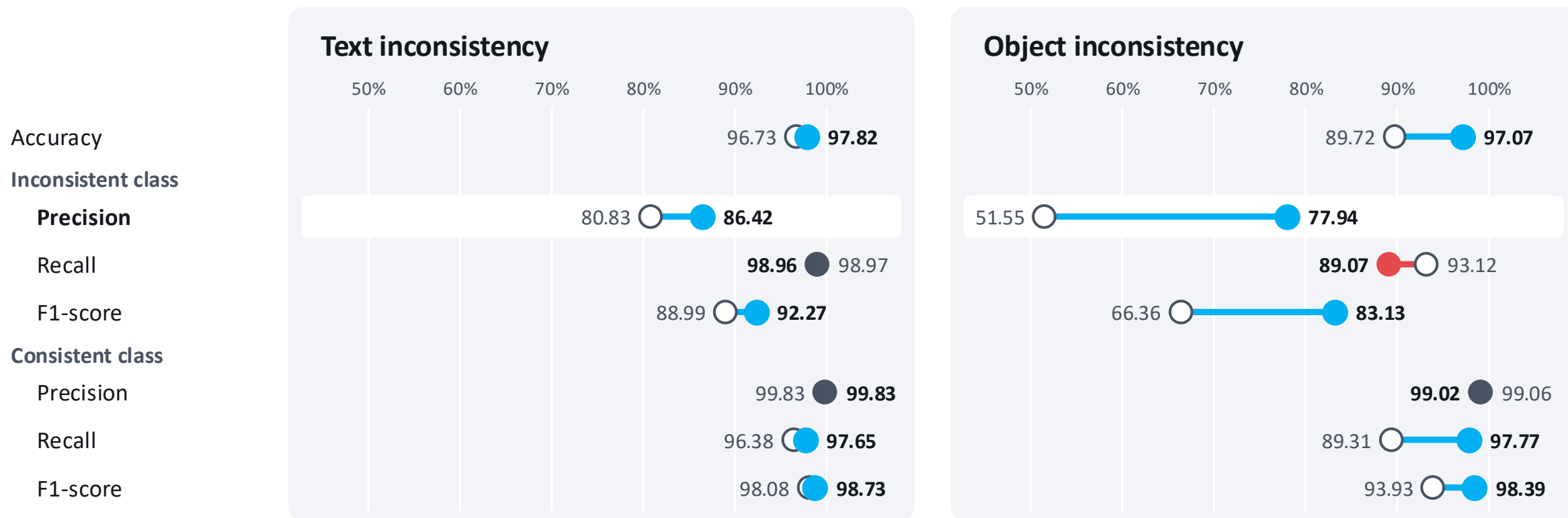
Cohen's κ between two authors labeling independently

The 147 native-dark-mode subjects were filtered from 3,500 top-ranked sites (Tranco, Alexa, DataForSEO, Moz); selection details are in the paper.

RQ1: ChromaEyes’s Efficacy of Inconsistency Detection



Improvement from Edge-based Verification Step In Object and Text Inconsistency Detection

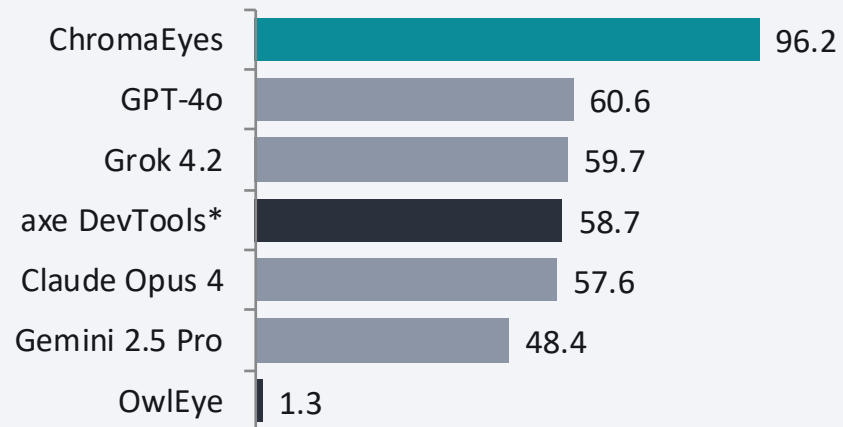


RQ2: Comparison with Baselines

→ No baseline above 61% accuracy

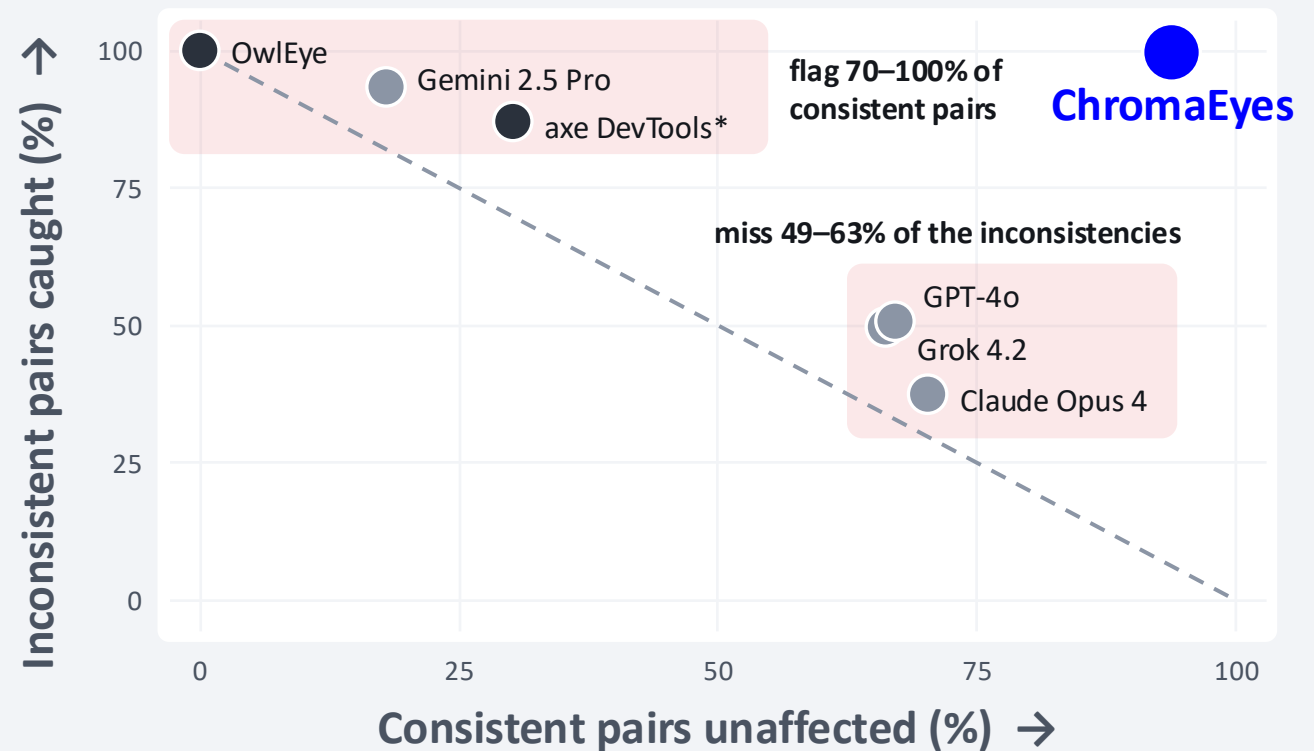
Accuracy

share of correct verdicts on 1,470 screenshot pairs



*axe DevTools is scored per page.

- Always answering “consistent” scores 59.5%
- No VLM localized an inconsistency correctly
- Statistical significance of the comparison results: McNemar, $p < 0.01$



RQ3: Inconsistencies in Extension-made Dark Modes

 7 light-only sites ×  7 dark-mode extensions =  49 converted apps →   539 screenshot pairs

Native dark mode

147 apps · 1,470 pairs, for comparison

596 inconsistent

874 consistent

40.5%

inconsistent

Extension-made dark mode

49 converted apps · 539 pairs

494 inconsistent

45

91.7%

inconsistent

ChromaEyes' verdict

on the same 539 pairs

492 correctly flagged

41

98.88%

accuracy

6 wrong verdicts: 2 missed, 4 false alarms

- All 49 combinations are affected, on average in 10 screenshot pairs per converted apps
- ChromaEyes finds an inconsistency in every one of the 49 combinations → 100% Accuracy application-wise

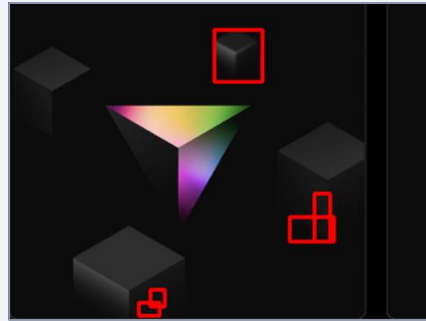
Where ChromaEyes Can Still Go Wrong

54 false alarms consistent pairs reported as inconsistent

Light mode



Dark mode · red boxes = detected objects

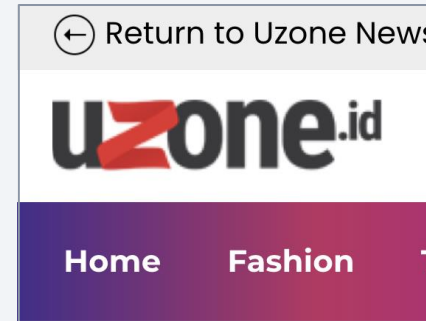


- 1 The object detector boxes these cubes in dark mode only
- 2 ChromaEyes finds no counterpart for them in light mode
- 3 It reports missing elements; a person sees the same page

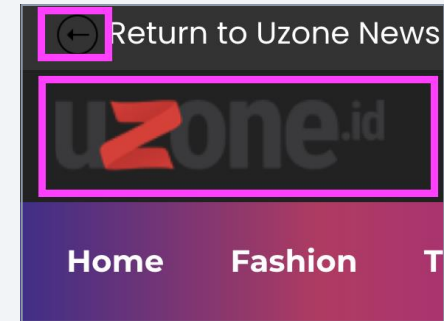
Cause: the detectors behave differently in the two modes. Content that changes between the two captures has the same effect.

2 misses inconsistent pairs reported as consistent

Light mode



Dark mode · pink boxes = hard to read



- 1 The logo and back arrow fade into the dark background
- 2 The detectors still recognize both of them, in both modes
- 3 Every element has a counterpart, so nothing is flagged

Cause: ChromaEyes checks that an element is present in both modes, not how readable it is.

The vision models tend to give reliable results only at small resolutions.

Conclusion

ChromaEyes for Light and Dark Mode Inconsistency Detection

- 1 The first tool specifically designed for detecting light and dark mode inconsistencies.
- 2 Accurate detection of inconsistencies requires jointly checking **UI element correspondence**, **layout structure**, and **background conversion**.
- 3 Automatic conversion by extension tools needs much room for improvement.



Code, data and labels

github.com/PALM-Group-Korea-University/ChromaEyes-ISSTA-2026

doi.org/10.5281/zenodo.21715790

bctak@knu.ac.kr < darksw@korea.ac.kr