

Enhance your testing strategy with **visual testing**



Learn the differences between functional and visual testing, gaps that functional testing misses, and how visual testing can help you **deliver a better user experience**.

What is functional testing?

Functional testing validates that the application **works as expected**. Tests are defined based on the requirements of the application, and measure whether a given input returns the desired output.

What is visual testing?

Visual testing validates that the application **looks as expected**. It tests that an application's UI elements — like colors, fonts, and buttons — appear correctly and that they aren't inhibiting usability.

Why functional testing isn't enough

Functional tests are important to test how your app works, but they only cover what you write for and each test needs to be asserted individually. If you don't expect a certain use case, you won't catch it.

Functional tests can easily miss bugs like:

- **Overlapping text or buttons**
- **Misspelled content**
- **Accessibility issues**
- **Invisible or off-page HTML elements**
- **Broad CSS changes**

Using only traditional assertions in a functional testing tool like Selenium WebDriver, Cypress, WebDriverIO, or Appium, you'd have to check the following for each visual element:

- **Visible (true/false)**
- **Upper-left coordinates (x, y)**
- **Height**
- **Width**
- **Background color**

Checking each of these **5 attributes** for each of the **7 elements** (a product image, title, description, button, and 3 color options) in the web page example in **Figure 1** would require **35 assertions**.

You also need to test your application across multiple combinations of:

- **Operating systems**
- **Browsers**
- **Screen sizes**
- **Fonts**

Dynamic, responsive content can render differently on each combination of these. And that's only one page. For more complex views and multiple pages, you could end up with **thousands of lines of assertion code**.

Figure 1: A bug like this button that is covered by the description could be missed by traditional functional tests.

Functional test scripts can be written to validate the size, position, and color scheme of visual elements. But if you do this, your test scripts will soon **balloon in size due to assertion bloat**.

What is automated visual testing?

Automated visual testing uses software to **automate the process of comparing visual elements across various screen combinations** to uncover visual defects. Automated visual testing integrates with your existing functional test frameworks like Selenium WebDriver, Cypress, WebdriverIO, or Appium to add an entire new level of validation.

AI-powered automated visual testing:



- Uses algorithms instead of pixels to determine when errors have occurred.
- Can easily test dynamic content such as personalized content, ads, and media.
- Automatically analyzes tests across different browsers and devices

An AI-powered automated visual testing tool can test a wide range of visual elements across a range of OS, browser, orientation, and resolution combinations. Simply **running the first baseline rendering and functional test on a single combination** is sufficient to guide an AI-powered tool to test results across the range of potential platforms.

AI-powered layout comparison ignores content differences and instead validates the existence of the content and relative placement. This helps **reduce false positives when testing dynamic content**.

Benefits of including AI-powered automated visual testing in your testing lifecycle



Test coverage

Drastically increase test coverage of use cases across your app.



Single-capture assertions

Reduce test code and the overhead of writing new tests for new features.



Catch more bugs

Catch unexpected bugs and defects before they reach production.



Easier test maintenance

Maintain test cases and baselines without needing coding skills.

To learn more about how visual testing can enhance your testing lifecycle, feel free to [reach out to our sales team](#).

Technical use cases



Visual and functional regression testing



Web and mobile testing



Cross-browser and device testing



Document validation and PDF testing

Business use cases



Brand investment, design validation, and accessibility



Legal, compliance, and digital asset management



Geographic expansion and localization testing



Customer experience and production monitoring