Visual testing toolbox

Martin Schneider

Taqelah Lightning Talks, 5.9.2020

Overview

Visual testing can be more than “just” regression testing against a baseline.

Some useful tools and techniques include:

1. Template matching using OpenCV
2. Layout testing using Galen
3. OCR using Tesseract

Template Matching

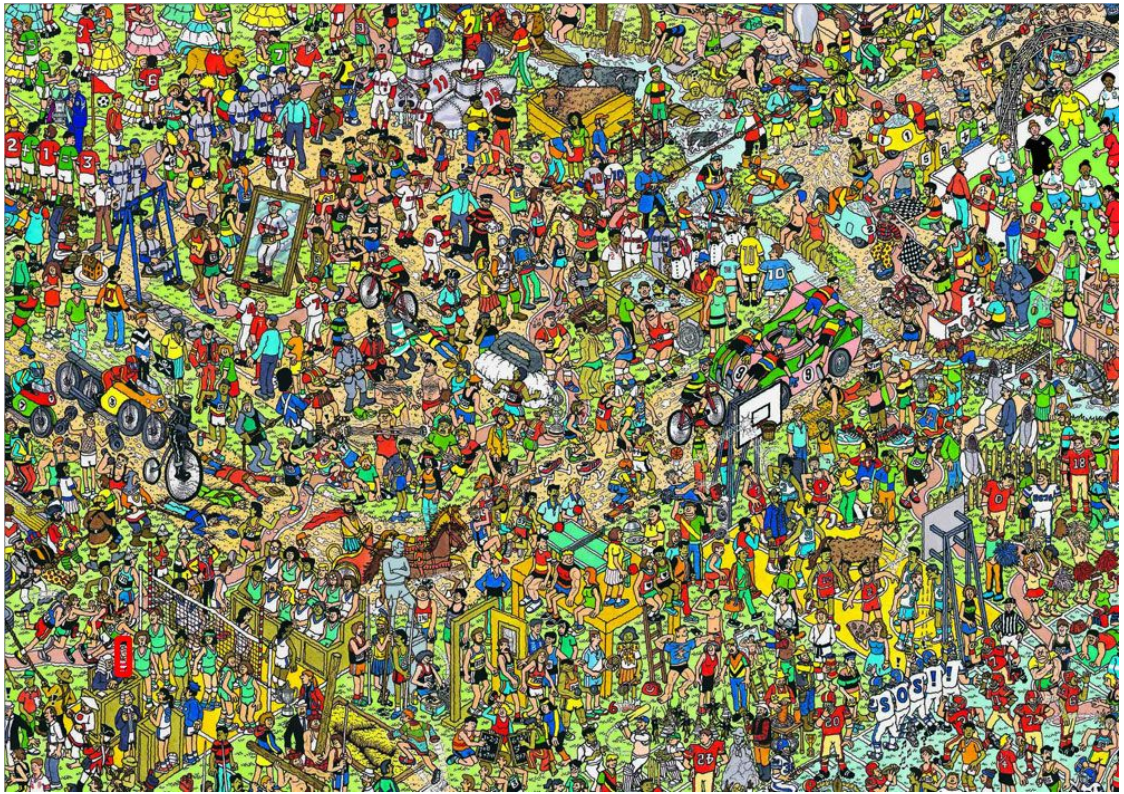


image taken from the Appium docs

Template Matching

- ***Task: Given an image (“template”), find it on the current screen.***

- Tool of choice: OpenCV
- There are two approaches
  - Server-side (with Appium): OpenCV runs on the same instance as the Appium server.
  - Client-side: OpenCV runs on the same instance as the test execution.

Template Matching contd.

1. **Appium**

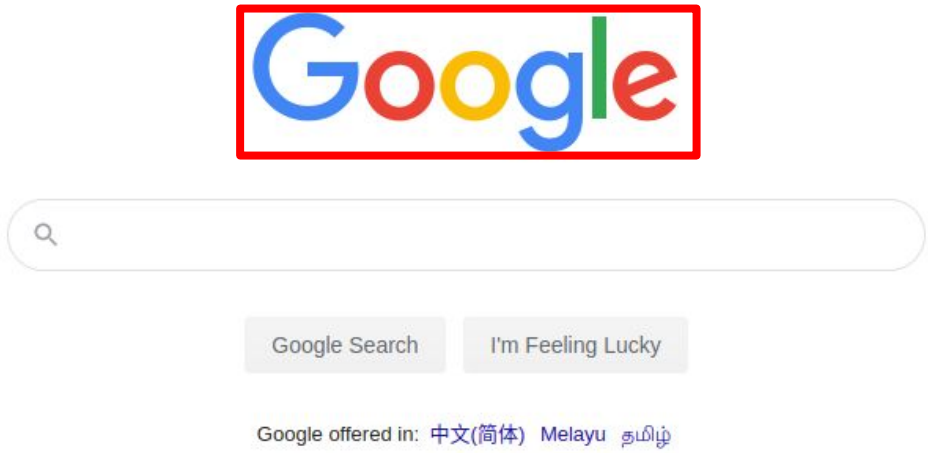
```
WebElement element = driver.findElementByImage(base64Image);
```

2. **OpenCV**

```
Mat result = new Mat(resultRows, resultCols, CvType.CV_32FC1);
Imgproc.matchTemplate(image, templ, result,
Imgproc.TM_CCOEFF_NORMED);
MinMaxLocResult match = Core.minMaxLoc(result);
if (match.maxVal >= threshold) ...
```

Template matching contd.

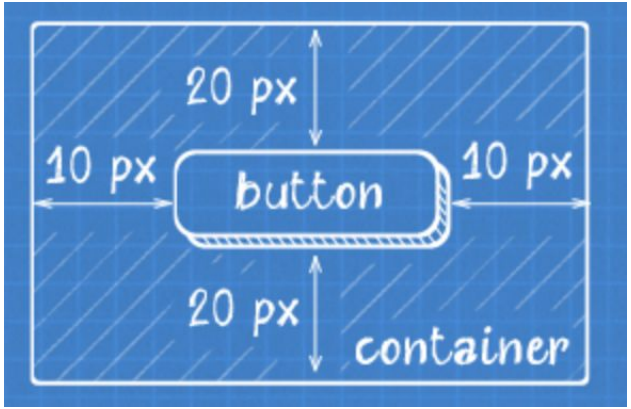
```
Feature: Google search
Scenario: Search
  Given I am on the homepage
  Then the Google logo is displayed
```



Layout testing

- ***Task: Verify the layout of a screen.***

- Rather than image-based comparison, we can verify the layout of a screen by checking the position of UI elements relative to other elements or the screen.
- Tool of choice: Galen



Layout testing

= Login =

SEARCH_FIELD:
  below LOGO
  centered horizontally inside viewport
  visible

LOGO:
  above SEARCH_FIELD
  centered horizontally inside viewport
  width < 100% of SEARCH_FIELD/width
  visible

SEARCH_BUTTON:
  near LUCKY_BUTTON 20px left
  visible

Layout testing contd.

BasePage class

```
public T verify() {
  …
  galen.checkLayout(specPath, locators);
  return (T) this;
}
```

Test code

```
private GooglePage google;

@Given("I am on the homepage")
public void homepage() {
  google.verify().enterSearchTerm(testdata(Search.class));
}
```

OCR

- ***Task: Find a text on the screen even if it is rendered as a graphic.***

- Tool of choice: Tesseract

Tesseract OCR

github.com/tesseract-ocr/tesseract

OCR contd.

Feature: Google search
Scenario: Search
  Given I am on the homepage
  Then the Google logo shows the correct text

Google

OCR contd.

OCR service

```
private String getText(File file) {
  return new Tesseract().doOCR(file).trim();
}

public String getText(WebElement element) {
  return getText(element.getScreenshotAs(OutputType.FILE));
}
```

Page object

```
public String getLogoText() {
  return ocr.getText($("LOGO"));
}
```

Demo

Resources