SMARTBEAR

TestComplete™

Speed Up Testing & Improve Software Quality with Optical Character Recognition

SMARTBEAR

Introduction

What is Optical Character Recognition?

For the past year, you've probably about the large-scale merger between HP and Micro Focus. Specifically, HP has decided to focus on its infrastructure rather than its application software. Because of this, they spun off their software division and Micro Focus ended up buying it. HP recently sold its software assets to Micro Focus in an $8.8 billion transaction, making Micro Focus one of the world's largest pure-play software companies. This has the potential to cause a high degree of uncertainty among HP customers. You may be asking yourself, which products were a part of this sale? Are the tools I use every day a part of this transaction? Or do they still belong to HPE?

The classic example of OCR is a large machine that processes paper. A human may review the results, line up text-boxes on forms with fields in a data-base, and press submit. In other cases, the text is printed clearly on a standard form and the import can run automatically.

OCR is a key aspect of the digital transformation for enterprises throughout many different industries, from the medical industry to the finance and software development industries. In many situations, automated workflows are replacing paper-based processes, but the ancient file cabinets taking up space off-site need to be replaced with something that is searchable and easy-to-retrieve.

For example, one monumental and ongoing undertaking that will need a decade or more to complete, is the conversion of older medical records into electronic copies. Over the past decade, tens of thousands of hospitals and private practices have installed and started using electronic medical record systems to track patient treatment. Despite this upgrade, many patient records include legacy paper documents, which will take years to turn into a digital format.

In the software development industry, OCR is revolutionizing how quality assurance teams can test the front-end of their software. When an application's underlying code is updated by developers, on-screen elements might change and objects will move, making graphical user interface (GUI) tests unreliable. Some objects aren't even recognizable by the tools in the market today. When UI elements can't be found, tests will fail, making it challenging for teams to scale their testing efforts.

> "OCR is a key aspect of the digital transformation for enterprises"

Automated testing tools that incorporate OCR are now able identify those previously unrecognizable objects. For testers, this means they can quickly build stronger and more scalable UI tests because they don't have to waste time writing object rules and maintaining object repositories.

Keep reading this white paper to learn more about how OCR is impacting automated UI testing, what the introduction of Artificial Intelligence (AI) means for OCR, and what SmartBear tool you can use to take advantage of OCR and improve the quality of your software.

Accelerating OCR Value with Artificial Intelligence

Classic OCR recognizes the pre-defined characters of ASCII, Unicode, and other fonts on a screen. The software being used to digitize a printed document will take a "best guess" to figure out whether an image matches a character in order to translate that image into text that the computer can process. Artificial Intelligence can take this to an entirely new level. AI OCR, such as the Google Cloud Vision API, has the opportunity to *learn.*

This is not the sort of learning that is portrayed in the movies, where the software becomes self-aware. Instead, the simplest example is when a user creates training data that allows the software to recognize similarities and make predictions. Once a few pictures are tagged, say photos with a name, applications like Facebook or Apple Photo can try to guess who is tagged. This is a revolutionary functionality that works amazingly well.

Now, think about training your software to recognize a shopping cart. Instead of coding a link to an image at a specific place on the screen, or relying on accessibility attributes for your UI elements, you tell the software testing tool to click the shopping cart icon. This is only possible through Artificial Intelligence, specifically supervised Machine Learning (ML).

A human may still need to do that training and execute the workflow around what to test. What AI systems with optical recognition can do is find objects or UI elements that are otherwise hard for a computer that is running traditional code to identify.

OCR combined with AI gives the operator the ability to train character recognition for a new font incredibly easily. The process is remarkably similar to the Facebook or Apple process. Select an area, allow the OCR to guess the text, then correct it to the right answer and try again. Unless the text is the scratchiest of inconsistent medical handwriting, the software will become incredibly accurate over a very short period of time. OCR has existed for decades, but this *ability to learn* means the operator can step away after minutes, instead of spending hours inspecting every action and making minor corrections.

OCR In Test Automation

By now, you can start to see the incredible potential of OCR and AI in software testing. Traditional test automation tools need to know exactly where to find user interface elements. Named IDs are the most common way to do this, but they are not always available. Some legacy systems don't use Named IDs or use IDs that are not predictable. In some cases, the software may need to interact directly as a user and would not be capable of finding the ID at all. Some platforms create the user interface in such a way that there is no ID to connect to - only an (x, y) coordinate. Bitmaps and charts typically present challenges for automated testing tools. Other systems can recognize elements using class names, captions, or other unique values for the specific context - and those are just the traditional elements!

Text can appear on the screen in all sorts of places that may not be readable. It may be generated by a computer program so that even the operating system does not recognize it as text, but instead as a series of lines and dots painted in the display. Alternatively, the software might be showing a PDF or other document that was scanned. In this scenario the characters would not be recognizable to the operating system. Finally, to test the new custom software, a company may need to interface with an external system such as SAP, which was not designed to be read from or automatically driven. For testing, we treated these as black-box applications.

All of these variations pose serious challenges to traditional test automation. By grabbing an area of the screen and reading it into a string, the tester can compare actual to expected results, indicate where to click, or do any required processing.

Use Cases of OCR In Test Automation

Test automation benefits from OCR in ways automation engineers may not anticipate, providing more stable and reliable automation scripts. Applications include:

* ***Reading PDFs and Scanned PDFs.*** Adobe's Portable Document Format may be the most popular way to create and share write-protected documents. It can be a challenge for test-automation tools to read or process PDFs in order to compare them to expected results. If the document is scanned, comparison is usually not possible. Testers typically do a few inspections by hand to make sure the file name matches and then give up. OCR with AI can reach into scanned PDFs - even those that are imperfect or scanned at an angle - and pull out the text to compare it to text pulled from a database, to other documents, to other screenshots, or to data stored in the test software itself.

* ***Terminal Server-Like Applications.*** Microsoft Terminal Server works by pushing a bitmap from the server onto the computer screen. Like the mainframe green screens of old, a terminal server app is a dumb client that simply sends mouse clicks, screen coordinates, and type commands back to a server. Terminal Server applications do not have any elements other than one big window. They do not have unique IDs for controls such as buttons. They do, however, have buttons, and anyone who can find the text on the button can find the center of the button and click it. The same applies to menus and other user interface elements.

* ***Unpredictable Identifiers.*** Sometimes an element ID is smart-coded and unpredictable. Imagine a logistics company that started using a testing tool on a legacy system. After Users create an order, they went back to the "active orders" screen, and a radio button for the new order appears at the top of the screen for users to select the order. The ID of the radio button was the ID of the order number. This was helpful to the server, as the user could click the View Order button and the ID would

be sent to the server to retrieve the order. To the tester, this represented a challenge. The request to click the radio button with the number on it that we don't yet know sounds difficult. With OCR and AI, this is as easy as reading the page into a string, splitting that on eight-digit numbers (i.e. the size of an order id), and clicking on the radio button next to the first set of numbers.

* ***Missing Identifiers.*** Some old platforms do not have identifiers at all. In Java Swing applications, for example, a drop-down list might simply be created in code at display time. AI for OCR can find the selected text, move the dropdown, and select a specific choice.
* ***Text that won't change but might move.*** Most web, mobile, and Windows desktop applications read text by finding a parent container object or object ID. Traditional test tools will struggle if the parent changes or the ID is generated randomly. AI for OCR can read the object and find whether the text is still a match.

The idea here is not to replace test automation with "automated AI.". Humans will still need to design and implement test tooling for automation. What is different, is the ability to take data that looks unstructured to the automation program and easily make it structured. The text on the screen that used to be impossible to reach has become a straightforward function call. As a bonus, the hard-to-reach objects that once required experimenting for connecting - or would result in brittle or flaky automation scripts - are now just a function call away.

The Google Cloud Vision API enables direct programmatic REST HTTP requests to the server through JSON payloads or through client libraries. Simply grab a screen capture that contains text as an image, send it to the API with a list of desired annotations, and the pre-trained models will identify what it recognizes. The API combines usability with analytical power to quickly classify your text data in information. The problem is connecting that API to your existing test automation tool. This is also a huge opportunity, especially for companies that build automation tools. As OCR software like Google Cloud Vision is exposed as an API, it gives test automation tools the capability of incorporating it. To a tester, the tool simply has another way of finding objects and getting values off of the screen. Done right, this is almost invisible. The first generation of those tools is available now, and the second is on the horizon.

> "The big advantage of incorporating AI with OCR is maintainability."

TestComplete Enhances UI Test Automation with OCR and AI

TestComplete is an automated testing tool that enables teams of all skillsets to build, run, and manage UI tests for desktop, mobile, and web applications. The tool natively includes the ability to use AI for Optical Character Recognition right in the user interface. While Text Recognition allows you to find existing plain text on the screen and interact with it like an object, AI-powered OCR allows you to find any type of text and interact with it. Using AI with OCR is as easy as recording a test step and clicking on a specific button or comparing data to an expected value.

The big advantage of incorporating AI with OCR is maintainability. Current OCR functionality is only as good as the data or object libraries it is based on, making it less accurate if an image, type font, or handwriting style not previously read by the system is required. By applying AI to OCR, libraries are able to stay up-to-date without any intervention by the user, allowing the tool to be more reliable when testing new parts of an application.

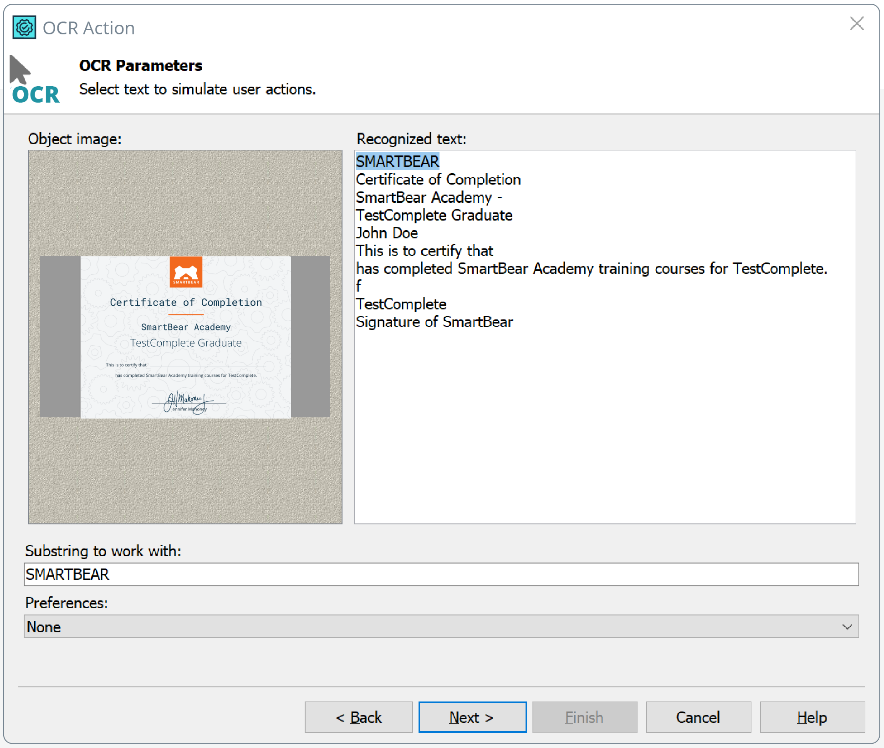
TestComplete supports many recognition capabilities and test actions through automatic detection of object properties and on-screen application components *(see Figure 1)*. With OCR in the Intelligent Quality Add-On, TestComplete can extract and validate text found embedded in images, charts, mainframes, and PDFs *(see Figure 2)*. Even captured screenshots can be converted into data for evaluation.

Intelligent Quality add-on, removing the dependency on user interface placement and coordinate-based recognition for more reliable identification. Elements identified using OCR simplify test script maintenance by providing another reliable identification method for hard to test objects. OCR element data can be analyzed as full text or in separate blocks based on the needs of your script. Additional options such as custom search, regular expressions, and text fragment selection allow fine-tuning the use of OCR data. You can manually add OCR-based commands to test scripts to find the right balance with property-based recognition or alter a recorded script. The Google Cloud Vision API that powers OCR

Controls not supported by your current automation framework can be recognized using their text by the

with AI constantly improves their recognition algorithms, leading to more effective outcomes when scripting automation using OCR.

Compared to the other technologies available in the market, the OCR functionality in TestComplete is more accurate, making it possible for UI automation engineers to detect and test against application components that were not previously recognizable. Migrate your existing flaky or unreliable UI tests to TestComplete to experience the full benefit of automated UI testing and OCR with AI using the tool's Intelligent Quality Add-On.



OCR Action

OCR Parameters

Select text to simulate user actions.

Object image:

Recognized text:

SMARTBEAR
Certificate of Completion
SmartBear Academy -
TestComplete Graduate
John Doe
This is to certify that
has completed SmartBear Academy training courses for TestComplete.
f
TestComplete
Signature of SmartBear

Substring to work with: SMARTBEAR

Preferences: None

Figure 1

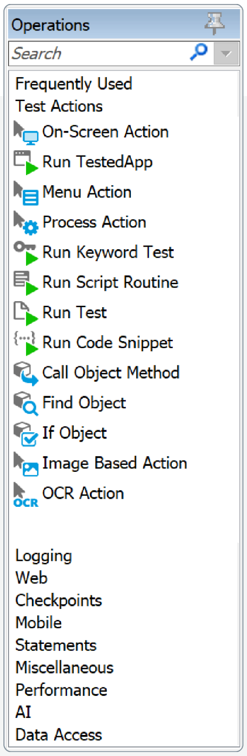


Figure 2