Search

Optical character recognition (OCR)

by Chris Woodford. Last updated: January 9, 2017.

Do you ever struggle to read a friend's handwriting? Count yourself lucky, then, that you're not working for the US Postal Service, which has to decode and deliver something like 30 million handwritten envelopes every single day! With so much of our lives computerized, it's vitally important that machines and humans can understand one another and pass information back and forth. Mostly computers have things their way—we have to "talk" to them through relatively crude devices such as keyboards and mice so they can figure out what we want them to do. But when it comes to processing more human kinds of information, like an old-fashioned printed book or a letter scribbled with a fountain pen, computers have to work much harder. That's where optical character recognition (OCR) comes in. It's a type of software (program) that can automatically analyze printed text and turn it into a form that a computer can process more easily. OCR is at the heart of everything from handwriting analysis programs on cellphones to the gigantic mail-sorting machines that ensure all those millions of letters reach their destinations. How exactly does it work? Let's take a closer look!

Photo: Recognizing characters: To you and me, it's the word "an", but to a computer this is just a meaningless pattern of black and white. And notice how the fibers in the paper are introducing some confusion into the image. If the ink were slightly more faded, the gray and white pattern of fibers would start to interfere and make the letters even harder to recognize.

What is OCR?

Photo: When it comes to optical character recognition, our eyes and brains are far superior to any computer.

As you read these words on your computer screen, your eyes and brain are carrying out optical character recognition without you even noticing! Your eyes are recognizing the patterns of light and dark that make up the characters (letters, numbers, and things like punctuation marks) printed on the screen and your brain is using those to figure out what I'm trying to say (sometimes by reading individual characters but mostly by scanning entire words and whole groups of words at once).

Computers can do this too, but it's really hard work for them. The first problem is that a computer has no eyes, so if you want it to read something like the page of an old book, you have to present it with an image of that page generated with an optical scanner or a digital camera. The page you create this way is a graphic file (often in the form of a JPG) and, as far as a computer's concerned, there's no difference between it and a photograph of the Taj Mahal or any other graphic: it's a completely meaningless pattern of pixels (the colored dots or squares that make up any computer graphic image). In other words, the computer has a picture of the page rather than the text itself—it can't read the words on the page like we can, just like that. OCR is the process of turning a picture of text into text itself—in other words, producing something like a TXT or DOC file from a scanned JPG of a printed or handwritten page.

What's the advantage of OCR?

Once a printed page is in this machine-readable text form, you can do all kinds of things you couldn't do before. You can search through it by keyword (handy if there's a huge amount of it), edit it with a word processor, incorporate it into a Web page, compress it into a ZIP file and store it in much less space, send it by email—and all kinds of other neat things. Machine-readable text can also be decoded by screen readers, tools that use speech synthesizers (computerized voices, like the one Stephen Hawking uses) to read out the words on a screen so blind and visually impaired people can understand them. (Back in the 1970s, one of the first major uses of OCR was in a photocopier-like device called the Kurzweil Reading Machine, which could read printed books out loud to blind people.)

How does OCR work?

Let's suppose life was really simple and there was only one letter in the alphabet: A. Even then, you can probably see that OCR would be quite a tricky problem—because every single person writes the letter A in a slightly different way. Even with printed text, there's an issue, because books and other documents are printed in many different typefaces (fonts) and the letter A can be printed in many subtly different forms.

Photo: There's a fair bit of variation between these different versions of a capital letter A, printed in different computer fonts, but there's also a basic similarity: you can see that almost all of them are made from two angled lines that meet in the middle at the top, with a horizontal line between.

Broadly speaking, there are two different ways to solve this problem, either by recognizing characters in their entirety (pattern recognition) or by detecting the individual lines and strokes characters are made from (feature detection) and identifying them that way. Let's look at these in turn.

Pattern recognition

If everyone wrote the letter A exactly the same way, getting a computer to recognize it would be easy. You'd just compare your scanned image with a stored version of the letter A and, if the two matched, that would be that. Kind of like Cinderella: "If the slipper fits..."

So how do you get everyone to write the same way? Back in the 1960s, a special font called OCR-A was developed that could be used on things like bank checks and so on. Every letter was exactly the same width (so this was an example of what's called a monospace font) and the strokes were carefully designed so each letter could easily be distinguished from all the others. Check-printers were designed so they all used that font, and OCR equipment was designed to recognize it too. By standardizing on one simple font, OCR became a relatively easy problem to solve. The only trouble is, most of what the world prints isn't written in OCR-A—and no-one uses that font for their handwriting! So the next step was to teach OCR programs to recognize letters written in a number of very common fonts (ones like Times, Helvetica, Courier, and so on). That meant they could recognize quite a lot of printed text, but there was still no guarantee they could recognize any font you might send their way.

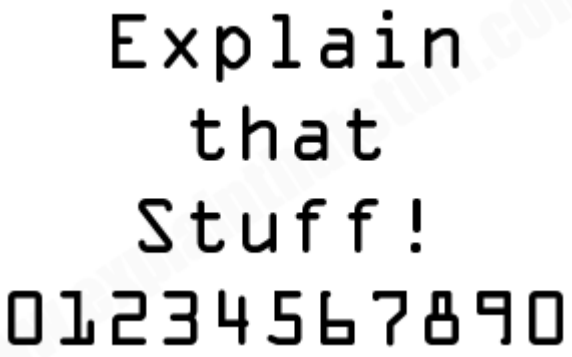

Photo: OCR-A font: Designed to be read by computers as well as people. You might not recognize the style of text, but the numbers probably do look familiar to you from checks and computer printouts. Note that similar-looking characters (like the lowercase "l" in Explain and the number "1" at the bottom) have been designed so computers can easily tell them apart.

Feature detection

Also known as feature extraction or intelligent character recognition (ICR), this is a much more sophisticated way of spotting characters. Suppose you're an OCR computer program presented with lots of different letters written in lots of different fonts; how do you pick out all the letter As if they all look slightly different? You could use a rule like this: If you see two angled lines that meet in a point at the top, in the center, and there's a horizontal line between them about halfway down, that's a letter A. Apply that rule and you'll recognize most capital letter As, no matter what font they're written in. Instead of recognizing the complete pattern of an A, you're detecting the individual component features (angled lines, crossed lines, or whatever) from which the character is made. Most modern omnifont OCR programs (ones that can recognize printed text in any font) work by feature detection rather than pattern recognition. Some use neural networks (computer programs that automatically extract patterns in a brain-like way).

Photo: Feature detection: You can be pretty confident you're looking at a capital letter A if you can identify these three component parts joined together in the correct way.

How does handwriting recognition work?

Recognizing the characters that make up neatly laser-printed computer text is relatively easy compared to decoding someone's scribbled handwriting. That's the kind of simple-but-tricky, everyday problem where human brains beat clever computers hands-down: we can all make a rough stab at guessing the message hidden in even the worst human writing. How? We use a combination of automatic pattern recognition, feature extraction, and—absolutely crucially—knowledge about the writer and the meaning of what's being written ("This letter, from my friend Harriet, is about a classical concert we went to together, so the word she's written here is more likely to be 'trombone' than 'tramline'.")

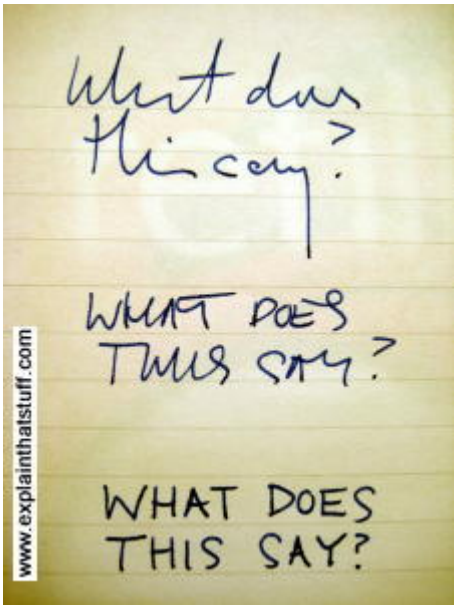

Photo: Handwriting recognition: Cursive handwriting (with letters joined up and flowing together) is very much harder for a computer to recognize than computer-printed type, because it's difficult to know where one letter ends and another begins. Many people write so hastily that they don't bother to form their letters fully, making recognition by pattern or feature extremely hard. Another problem is that handwriting is an expression of individuality, so people may go out of their way to make their writing different from the norm. When it comes to reading words like this, we rely heavily on the meaning of what's written, our knowledge of the writer, and the words that we've already read—something computers can't manage so easily.

Making it easy

Where computers do have to recognize handwriting, the problem is often simplified for them. For example, mail-sorting computers generally only have to recognize the zipcode (postcode) on an envelope, not the entire address. So they just have to identify a relatively small amount of text made only from basic letters and numbers. People are encouraged to write the codes legibly (leaving spaces between characters, using all uppercase letters) and, sometimes, envelopes are preprinted with little squares for you to write the characters in to help you keep them separate.

Forms designed to be processed by OCR sometimes have separate boxes for people to write each letter in or faint guidelines known as comb fields, which encourage people to keep letters separate and write legibly. (Generally the comb fields are printed in a special color, such as pink, called a dropout color, which can be easily separated from the text people actually write, usually in black or blue ink.)

Tablet computers and cellphones that have handwriting recognition often use feature extraction to recognize letters as you write them. If you're writing a letter A, for example, the touchscreen can sense you writing first one angled line, then the other, and then the horizontal line joining them. In other words, the computer gets a headstart in recognizing the features because you're forming them separately, one after another, and that makes feature extraction much easier than having to pick out the features from handwriting scribbled on paper.

What does OCR involve in practice?

Most people don't have to use OCR on an industrial scale: not many of us will need to scan even a hundred documents a day, never mind a million. We're more likely to want to OCR the occasional printed article and turn it into editable form, or scan an out-of-copyright book and republish it as a Web page.

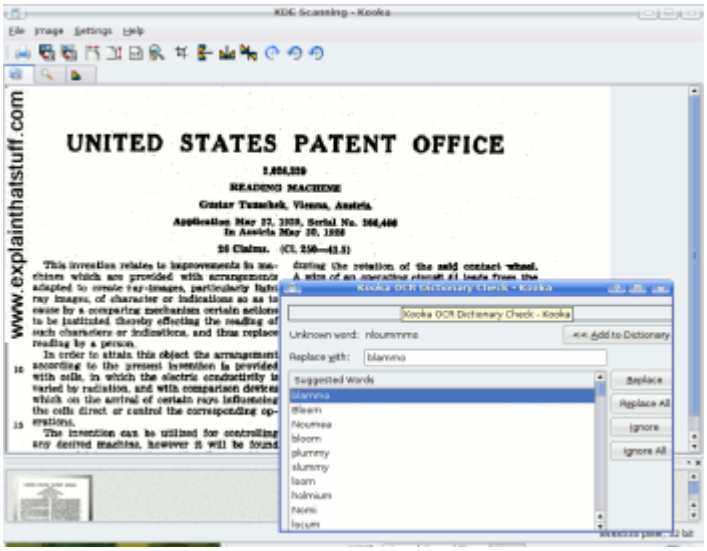

Photo: What OCR looks like in practice: you load an image of a scanned page (the words in the background are a scanned bitmap), run your OCR program to generate a text file, and then step through the "recognized" text manually correcting errors. Here, in a neat twist of history, I'm using the Linux Kooka scanning program to OCR the patent of Gustav Tauschek's original OCR system from 1928/1929 (described below).

In practice, this is what everyday OCR actually involves:

1. Printout: First, you need to get the best possible printout of your existing document. Sometimes you're just stuck with an old typewritten script, but you may be able to improve the print quality by photocopying (to increase the contrast between the print and the page). The quality of the original printout makes a huge difference to the accuracy of the OCR process. Dirty marks, folds, coffee stains, ink blots, and any other stray marks will all reduce the likelihood of correct letter and word recognition.
2. Scanning: You run the printout through your optical scanner. Sheet-feed scanners are better for OCR than flatbed scanners because you can scan pages one after another. Most modern OCR programs will scan each page, recognize the text on it, and then scan the next page automatically. If you're using a flatbed scanner, you'll have to insert the pages one at a time by hand. If you've got a reasonably good

digital camera, you may be able to create images of your pages by taking photos. You'll probably need to use a macro (close-up) focus setting to get really sharp letters that are clear enough for accurate OCR.

3. Two-color: The first stage in OCR involves generating a black-and-white (two-color/one-bit) version of the color or grayscale scanned page, similar to what you'd see coming out of a fax machine. OCR is essentially a binary process: it recognizes things that are either there or not. If the original scanned image is perfect, any black it contains will be part of a character that needs to be recognized while any white will be part of the background. Reducing the image to black and white is therefore the first stage in figuring out the text that needs processing—although it can also introduce errors. If you have a color scan of a newspaper with a large brown coffee stain over the words, it's easy to tell the text from the stain; but if you reduce the scan to a black-and-white image, the stain will turn to black and white too and may confuse the OCR process.
4. OCR: All OCR programs are slightly different, but generally they process the image of each page by recognizing the text character by character, word by word, and line by line. In the mid-1990s, OCR programs were so slow that you could literally watch them "reading" through and processing the text while you waited; computers are far faster now and OCR is pretty much instantaneous.
5. Basic error correction: Some programs give you the opportunity to review and correct each page in turn: they instantly process the entire page and then use a built-in spellchecker to highlight any apparently misspelled words that may indicate a misrecognition, so you can automatically correct the mistake. You can usually switch off this feature if you want to, if you have many pages to scan and you don't want to check them all as you're going along. Sophisticated OCR programs have extra error checking features to help you spot mistakes. For example, some use what's called near-neighbor analysis to find words that are likely to occur nearby, so text incorrectly recognized as "the barking bog" might be automatically changed to "the barking dog" (because "barking" and "dog" are two words that very often run together).
6. Layout analysis: Good OCR programs automatically detect complex page layouts, such as multiple columns of text, tables, images, and so on. Images are automatically turned into graphics, tables are (with luck) turned into tables, and columns are split up correctly (so the text from the first line of the first column isn't automatically joined to the text from the first line of the second column).
7. Proofreading: Even the best OCR programs aren't perfect, especially when they're working from very old documents or poor quality printed text. That's why the final stage in OCR should always be a good, old-fashioned human proofread!

Who invented OCR?

Most people think getting machines to read human text is a relatively recent innovation, but it's older than you might suppose. Here's a whistle-stop tour through OCR history:

- 1928/9: Gustav Tauschek of Vienna, Austria patents a basic OCR "reading machine." Paul Handel of General Electric files a patent for a similar system in the United States in April 1931. Both are based on the idea of using light-detecting photocells to recognize patterns on paper or card.
- 1949: L.E. Flory and W.S. Pike of RCA Laboratories develop a photocell-based machine that can read text to blind people at a rate of 60 words per minute. (Read all about it in the February 1949 issue of Popular Science.)
- 1950: David H. Shepard develops machines that can turn printed information into machine-readable form for the US military and later founds a pioneering OCR company called Intelligent Machines Research (IMR). Shepherd also develops a machine-readable font called Farrington B (also called OCR-7B and 7B-OCR), now widely used to print the embossed numbers on credit cards.
- 1960: Lawrence (Larry) Roberts, a computer graphics researcher working at MIT, develops early text recognition using specially simplified fonts such as OCR-A. He later becomes one of the founding fathers of the Internet.
- 1950s/1960s: Reader's Digest and RCA work together to develop some of the first commercial OCR systems.
- 1960s: Postal services around the world begin to use OCR technology for mail-sorting. They include the US Postal Service, Britain's General Post Office (GPO, now called Royal Mail), Canada Post, and the German Deutsche Post. Helped by companies such as Lockheed Martin, postal services remain at the forefront of OCR research to this day.

- 1974: Raymond Kurzweil develops the Kurzweil Reading Machine (KRM) that combines a flatbed scanner and speech synthesizer in a machine that can read printed pages aloud to blind people. Kurzweil's OCR software is acquired by Xerox and marketed under the names ScanSoft and (later) Nuance Communications.
- 1993: The Apple Newton MessagePad (PDA) is one of the first handheld computers to feature handwriting recognition on a touch-sensitive screen. During the 1990s, handwriting recognition becomes an increasingly popular feature on cellphones, PDAs (notably the pioneering Palm and PalmPilot), and other handhelds.
- 2000: Researchers at Carnegie Mellon University flip the problem of developing a good OCR system on its head—and develop a spam-busting system called CAPTCHA (see caption below).

Photo: reCAPTCHA kills two birds with one stone: We know from OCR research that computers find it hard to recognize badly printed words that humans can read relatively easily. That's why CAPTCHA puzzles like this are used to stop spammers from bombarding email systems, message boards, and other websites. This one's produced by Google as part of their reCAPTCHA system. It has an added benefit: when you type in the garbled words, you're helping Google to recognize part of the scanned text from an old book that it wants to convert to machine-readable form. In effect, you're doing a little bit of OCR on Google's behalf!

Find out more

On this website

You might like these other articles on our site covering related topics:

- Barcodes and barcode scanners
- CCDs (charge-coupled devices)
- Digital cameras
- Fingerprint scanners
- Iris scanners
- Neural networks

Books

These are academic books that may be too complex if you just want a quick overview of OCR.

- Fundamentals in Handwriting Recognition by Sebastiano Impedovo (ed). Springer Science & Business Media, 2012. A series of recent academic papers explores the cutting-edge of OCR for handwriting.
- Markov Models for Handwriting Recognition by Thomas Plötz, Gernot A. Fink. Springer Science & Business, 2012. Introduces both hidden Markov and Markov chain models.
- Character Recognition Systems: A Guide for Students and Practitioners by Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu. John Wiley & Sons, 2007. Covers the history of OCR, before going into detail about feature extraction, statistical methods, neural networks, and various case studies.
- Optical Character Recognition: An Illustrated Guide to the Frontier by Stephen V. Rice et al. Kluwer Academic, 1999. Slightly dated now, but still a useful and comprehensive guide to how OCR actually works, with a great deal of background about processing recognition errors in various ways.

Articles

- Quickly and Easily, Scanning and Storing Documents on the Go by Kit Eaton. The New York Times, December 4, 2013. Thanks to OCR scanning apps, you no longer need to photocopy and retype notes you make on the road.
- Scanner for ebook cannot tell its 'arms' from its 'anus' by Alison Flood. The Guardian, May 1, 2014. OCR errors are proving an embarrassing liability in ebooks.
- A picture of a thousand words? by Evin Levey. Google Blog, October 30, 2008. How Google converted scanned PDFs into indexable text to make its search results more useful.

Patents

If you like technical details, you'll find OCR patents worth a look. Here a few representative examples to start you off; you'll find many more on Google Patents (you can use the search operator "intitle:OCR" to find hundreds of relevant patents).

- Methods and apparatuses for controlling access to computer systems and for annotating media files by Luis Von Ahn et al, Carnegie Mellon University, published June 26, 2014. A detailed technical description of how CAPTCHAs work.
- Reading system by Raymond Kurzweil, published December 23, 1999. One of Ray Kurzweil's later OCR patents describing a computerized reading machine for the blind that scans a printed page, turns it into a text file, and then reads the text out loud.
- Letter segmenting apparatus for OCR comprising multi-level segmentor operable when binary segmenting fails by Toshio Miyazaki et al, NEC, published June 3, 1980. A detailed description of how an OCR system can identify individual letters in a written sample.
- Regional context maximum likelihood ocr error correction apparatus by Walter S. Rosenbaum et al, published September 18, 1979. A method of correcting OCR errors using a stored dictionary. It's hilariously ironic that Google's scan of this patent document contains so many OCR errors.

If you liked this article...

Follow us

-

Rate this page

Please rate or give feedback on this page and I will make a donation to WaterAid.

Share this page

Press CTRL + D to bookmark this page for later or tell your friends about it with:

-
-
-
-
-
-
-
-
-

Cite this page

Woodford, Chris. (2010/2014) OCR (optical character recognition). Retrieved from http://www.explainthatstuff.com/how-ocr-works.html. [Accessed (Insert date here)]

More to explore on our website...

- Communications
- Computers
- Electricity & electronics
- Energy
- Engineering
- Environment

- Gadgets
- Home life
- Materials
- Science
- Tools & instruments
- Transportation

- Home
- A-Z index
- Get the book
- Follow us
- Random article
- Timeline
- Teaching guide
- About us
- Privacy policy

↑ Back to top