Mariner Software

An introduction to Optical Character Recognition

Jim Henson - 2015-03-25 - OCR

An introduction to Optical Character Recognition

Optical Character Recognition (OCR) is a process by which text characters can be input to a computer by providing the computer with an image. The computer uses an **OCR Engine**--a computer program with the specific function of making a guess which letter (recognizable to a computer) an image (recognizable to a human) represents.

Paperless includes an OCR Engine, which it uses to recognize text and numerical values. In order to understand how the OCR Engine in Paperless produces OCR results, it is useful also to understand how OCR Engines make these guesses.

How OCR Engines Work
An OCR engine scans an image for elements that resemble letters it is programmed to recognize.

OCR engines use sets of parameters to discern one character from another. For example:

- The letters E and F look a lot alike; the most-noticable differences between the two characters is the horizontal bar at the bottom of the letter E.
- The letters P and D look a lot alike. There are two primary differences between the letters: the letter P has a vertical line that extends beyond the loop shape at the top of the letter; the letter D does not.
- The letters e and a look a lot alike.
- The letters q and the number 9 look a lot alike in certain fonts.
- The number 2 and the letter Z look a lot alike.
- The semicolon (;) and the colon (:) symbols are nearly-identical.
- The period (.) and the comma (,) symbols are nearly-identical.

At the most-basic level, OCR locates points in a text that resemble characters it is trained to recognize; once it finds what it believes is a match, it returns a letter (recognizable to the computer) to OCR results.

This also limits the ability of OCR in some cases; the Roman letter B and the Greek capital Beta symbol (ß) also look a lot alike. B is on the standard UTF-8 character Map; ß is not. If a character is not on an OCR engine's list of characters to recognize, it will either not recognize the character or interpret it as something that is on its list.

Similarly, the OCR engine needs to make decisions that separate the letter capital letter k:

K

from the combination of a pipe character with the less-than symbol:

|<

Both look very similar visually, but to a human reader, they may represent very-different things.

OCR engines are also programmed to recognize certain fonts. Thus, if the OCR engine is programmed to recognize Bitstream Vera Sans, it will recognize text characters rendered in Bitstream Vera Sans. If the OCR engine has been instructed to recognize Bitstream Vera Sans, but it has not been configured to recognize Linux Libertine, it may not recognize text in Linux Libertine with the same (or any) degree of accuracy compared to what it is able to recognize Bitstream Vera Sans with.

The OCR Process

Here is a very-basic overview of how an OCR engine processes an image to return text contained in it:

1. An image of the document is acquired by the computer.
2. The image is submitted as input to an OCR engine.
3. The OCR engine matches portions of the image to shapes it is instructed to recognize.
4. Given logic parameters that the OCR engine has been instructed to use, the OCR engine will make its best guess as to which letter a shape represents.
5. OCR results are returned as text.

Example

In testing Paperless we added a picture of a pair of scissors to a Paperless library. Paperless returned the following

}<

The OCR engine in Paperless guessed that the handle of the scissors represented a curly bracket, and that the

This is acceptable, per the logical parameters that we provided Paperless, in order to make a best-guess as to the text equivalent of an image.

Factors that affect OCR Fidelity

For information on factors that can affect the fidelity (accuracy) of OCR output to the original, see this Knowledgebase article.