Tesseract OCR Engine

What it is, where it came from, where it is going.

Ray Smith, Google Inc
OSCON 2007
## Contents

- Introduction & history of OCR
- Tesseract architecture & methods
- Announcing Tesseract 2.00
- Training Tesseract
- Future enhancements
A Brief History of OCR

- What is Optical Character Recognition?

My invention relates to statistical machines of the type in which successive comparisons are made between a character and a character.
A Brief History of OCR

• OCR predates electronic computers!

US Patent 1915993, Filed Apr 27, 1931
<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1929</td>
<td>Digit recognition machine</td>
</tr>
<tr>
<td>1953</td>
<td>Alphanumeric recognition machine</td>
</tr>
<tr>
<td>1965</td>
<td>US Mail sorting</td>
</tr>
<tr>
<td>1965</td>
<td>British banking system</td>
</tr>
<tr>
<td>1976</td>
<td>Kurzweil reading machine</td>
</tr>
<tr>
<td>1985</td>
<td>Hardware-assisted PC software</td>
</tr>
<tr>
<td>1988</td>
<td>Software-only PC software</td>
</tr>
<tr>
<td>1994-2000</td>
<td>Industry consolidation</td>
</tr>
</tbody>
</table>
Tesseract Background

- Developed on HP-UX at HP between 1985 and 1994 to run in a desktop scanner.
- Never used in an HP product.
- Highly portable.
Tesseract OCR Architecture

Input: Gray or Color Image [+ Region Polygons]

Adaptive Thresholding

- Binary Image

Find Text Lines and Words

Character Outlines

Connected Component Analysis

Character Outlines Organized Into Words

Recognize Word Pass 1

Recognize Word Pass 2
Adaptive Thresholding is Essential

Some examples of how difficult it can be to make a binary image
Taken from the UNLV Magazine set.
(http://www.isri.unlv.edu/ISRI/OCRtk)
Baselines are rarely perfectly straight

- Baselines are approximated by quadratic splines to account for skew and curl.
- Meanline, ascender and descender lines are a constant displacement from baseline.
- Critical value is the x-height.

*Volume 69, pages 872–879.*
Spaces between words are tricky too.

- Italics, digits, punctuation all create special-case font-dependent spacing.
- Fully justified text in narrow columns can have vastly varying spacing on different lines.

of 9.5% annually while the Federated junk fund returned 11.9% fear of financial collapse,
Tesseract: Recognize Word

Character Chopper → Character Associator → Done?

Static Character Classifier → Dictionary → Adaptive Character Classifier → Number Parser

Adapt to Word

No → Yes
Outline Approximation

Polygonal approximation is a double-edged sword. Noise and some pertinent information are both lost.
**Tesseract: Features and Matching**

<table>
<thead>
<tr>
<th>Prototype</th>
<th>Character to classify</th>
<th>Extracted Features</th>
<th>Match of Prototype To Features</th>
<th>Match of Features To Prototype</th>
</tr>
</thead>
</table>

- Static classifier uses outline fragments as features. Broken characters are easily recognizable by a small-\(\rightarrow\)large matching process in classifier. (This is slow.)
- Adaptive classifier uses the same technique! (Apart from normalization method.)
Announcing tesseract-2.00

- Fully Unicode (UTF-8) capable
- Already trained for 6 Latin-based languages (Eng, Fra, Ita, Deu, Spa, Nld)
- Code and documented process to train at http://code.google.com/p/tesseract-ocr
- UNLV regression test framework
- Other minor fixes
Training Tesseract

Word List

Training page images

Box files

Character Features (*.tr files)

Unicharset

Tesseract

Tesseract + manual correction

Wordlist2dawg

Word-dawg, Freq-dawg

inttemp, pffmtable

normproto

unicharset

DangAmbigs

Unicharset_extractor

Addition of character properties

Manual Data Entry

mfTraining

cnTraining

Manual Data Entry

Unicharset

Training page images

User-words

Training page images

Tesseract Data Files
**Tesseract Dictionaries**

- **Word List**
  - **Infrequent Word List**
  - **Frequent Word List**

**Tesseract Data Files**

- **User-words**
- **Word-dawg, Freq-dawg**
Tesseract Shape Data

Training page images

Tesseract + manual correction

Box files

Tesseract

Character Features (*.tr files)

Prototype Shape Features

Expected Feature Counts

inttemp, pffmtable

mfTraining

cnTraining

normproto

Character Normalization Features
Tesseract Character Data

- Training page images
- Box files
  - Unicharset_extractor
  - unicharset
  - Addition of character properties
  - unicharset
  - DangAmbigs

List of Characters + ctype information

Typical OCR errors eg e<->c, rn<->m etc

Tesseract Data Files
## Accuracy Results

Comparison of current results against 1995 UNLV results

<table>
<thead>
<tr>
<th>Testset</th>
<th>Character</th>
<th>Testid</th>
<th>Errors</th>
<th>Accuracy</th>
<th>Change</th>
<th>Non-stopword</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1995</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus.3B</td>
<td>5959</td>
<td></td>
<td>1293</td>
<td>95.73%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Doe3.3B</td>
<td>36349</td>
<td></td>
<td>7042</td>
<td>94.87%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mag.3B</td>
<td>15043</td>
<td></td>
<td>3379</td>
<td>94.99%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>News.3B</td>
<td>6432</td>
<td></td>
<td>1502</td>
<td>96.94%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gcc4.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus.3B</td>
<td>6258</td>
<td></td>
<td>1312</td>
<td>95.67%</td>
<td>1.47%</td>
<td></td>
</tr>
<tr>
<td>Doe3.3B</td>
<td>28589</td>
<td></td>
<td>6692</td>
<td>95.12%</td>
<td>-4.97%</td>
<td></td>
</tr>
<tr>
<td>Mag.3B</td>
<td>14800</td>
<td></td>
<td>3123</td>
<td>95.37%</td>
<td>-7.58%</td>
<td></td>
</tr>
<tr>
<td>News.3B</td>
<td>7524</td>
<td></td>
<td>1220</td>
<td>97.51%</td>
<td>-18.77%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>Gcc4.1</td>
<td>57171</td>
<td>-10.37%</td>
<td></td>
<td>12347</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-6.58%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Comparison of current results against 1995 UNLV results.
<table>
<thead>
<tr>
<th>Commercial OCR v Tesseract</th>
</tr>
</thead>
<tbody>
<tr>
<td>100+ languages.</td>
</tr>
<tr>
<td>Accuracy is good now.</td>
</tr>
<tr>
<td>Sophisticated app with complex UI.</td>
</tr>
<tr>
<td>Works on complex magazine pages.</td>
</tr>
<tr>
<td>Windows Mostly.</td>
</tr>
<tr>
<td>Costs $130-$500</td>
</tr>
<tr>
<td>6 languages + growing.</td>
</tr>
<tr>
<td>Accuracy was good in 1995.</td>
</tr>
<tr>
<td>No UI yet.</td>
</tr>
<tr>
<td>Page layout analysis coming soon.</td>
</tr>
<tr>
<td>Runs on Linux, Mac, Windows, more...</td>
</tr>
<tr>
<td>Open source – Free!</td>
</tr>
</tbody>
</table>
Tesseract Future

- Page layout analysis.
- More languages.
- Improve accuracy.
- Add a UI.
The End

• For more information see: http://code.google.com/p/tesseract-ocr