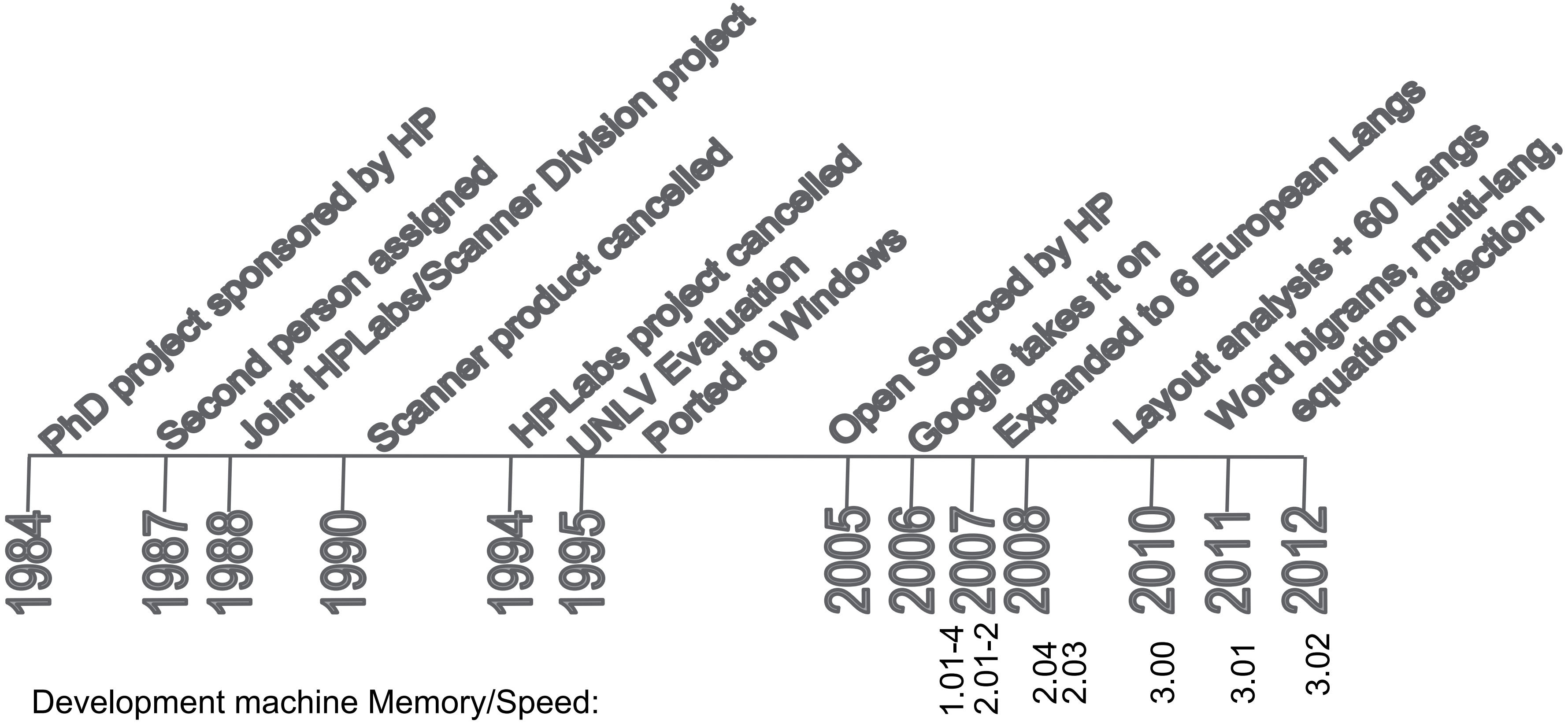
1. Motivation and History of the Tesseract OCR Engine

Lessons Learned from What Worked and What Didn't

Ray Smith, Google Inc.

What is Tesseract?

Tesseract Timeline



1984 PhD project sponsored by HP

1987 Second person assigned

1988 Joint HPLabs/Scanner Division project

1990 Scanner product cancelled

1994 HPLabs project cancelled

1995 UNLV Evaluation

Ported to Windows

2005 Open Sourced by HP

2006 Google takes it on

2007 Expanded to 6 European Langs

2008

2010 Layout analysis + 60 Langs

2011 Word bigrams, multi-lang,

2012 equation detection

1.01-4

2.01-2

2.04

2.03

3.00

3.01

3.02

Development machine Memory/Speed:

4MB	16MB	64MB	256MB	3GB	24GB
10MHz	25MHz	100MHz	166MHz	3.5GHz	12 x 3.5GHz

The Stealth Era: 1984-1994

Office equipment trade-show:
Earl's Court, London late 1984

Dest has a scanner with built-in OCR with >99.5% accuracy.

Office equipment trade-show:
Earl's Court, London late 1984

Dest has a scanner with built-in OCR with >99.5% accuracy.

Caveat: It only recognizes 8 fonts from IBM Selectric "golf-ball" typewriters.

Office equipment trade-show:
Earl's Court, London late 1984

For documents conventional OCR can't handle, Kurzweil offers a better approach: ICR.

Kurzweil Reading Machine
(circa 1978)

For almost two decades, optical character recognition systems have been widely used to provide automated text entry into computerised systems. Yet in all this time, conventional OCR systems have never overcome their in- ability to read more than a handful of type fonts and page formats. Propor- tionally spaced type (which includes vir- tually all typeset copy), laser printer fonts, and even many non-proportional typewriter fonts, have remained beyond the reach of these systems. And as a result, conventional OCR has never achieved more than a marginal impact on the total number of documents needing conversion into digital form.

Early Lessons:

Early Lessons:

- Be able to OCR your own sales literature!

Early Lessons:

- Be able to OCR your own sales literature!
- OCR should distinguish text from non-text.

Early Lessons:

- Be able to OCR your own sales literature!
- OCR should distinguish text from non-text.
- OCR should read white-on gray and black-on-gray as easily as black-on-white.

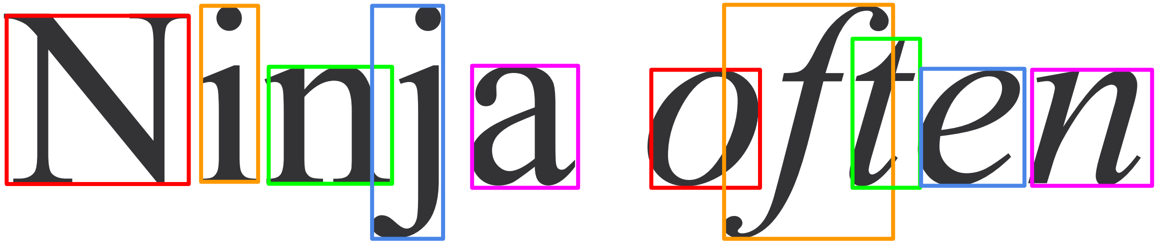
Early Lessons:

- Be able to OCR your own sales literature!
- OCR should distinguish text from non-text.
- OCR should read white-on gray and black-on-gray as easily as black-on-white.

Decision:
Extract outlines from grayscale images and features from the outlines.

- Profound impact.
- Key differentiator.

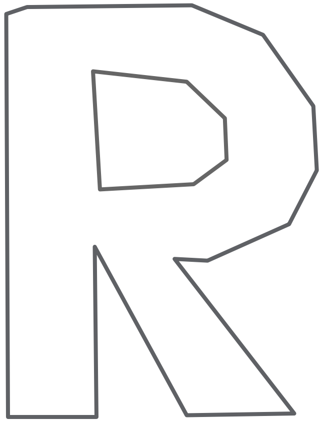
Connected component outlines vs bounding boxes:



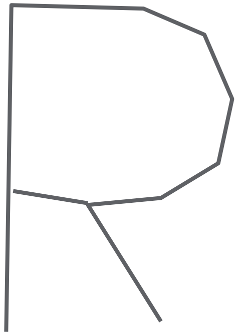
- Bounding boxes of characters often overlap without the characters actually touching.

How to extract features from Outlines?

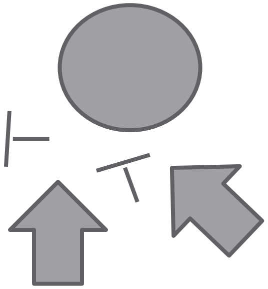
Outline



Skeleton



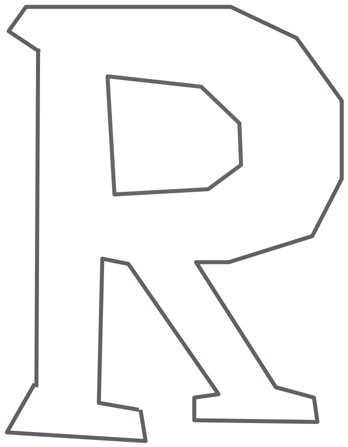
Topological Features

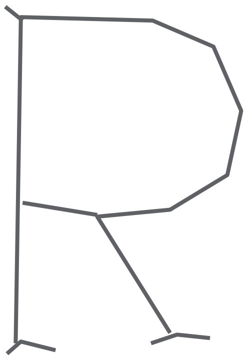


Skeletonization is Unreliable

Outline: Serifed

Skeleton: Decorated





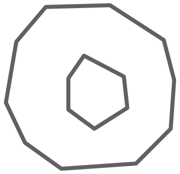
Arrrrh!

Lesson: If there are a lot of papers on a topic, there is most likely no good solution, at least not yet, so try to use something else.

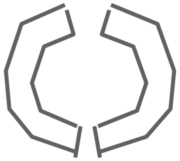
Topological features are Brittle

Damage to 'o' produces vastly different feature sets:

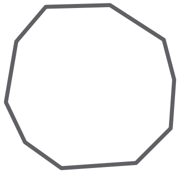
Standard 'o'



Broken 'o'



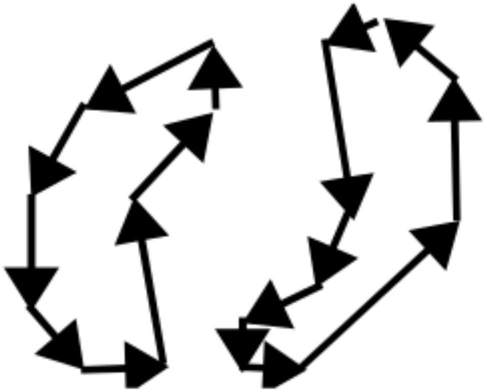
Filled 'o'



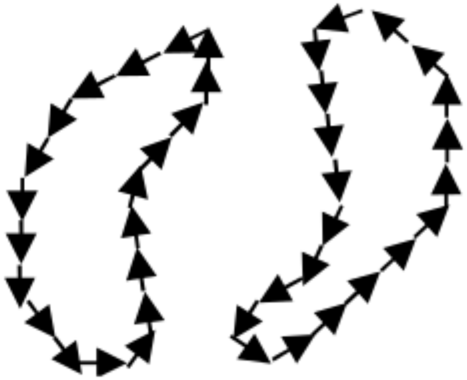
Lesson: Features must be as invariant as possible to as many as possible of the expected degradations.

Shrinking features and inappropriate statistics

Segments of the polygonal Approximation



Even smaller features



Statistical: $$argmax(k) \prod_{l,i} \frac{1}{\sigma_{ijk}} \exp\left[-\frac{1}{2}\left(\frac{x_{il}-\mu_{ijk}}{\sigma_{ijk}}\right)^2\right]$$

Geometric: $$argmin(k) \frac{1}{M+J_k}\left(\sum_{l,i}(x_{il}-\mu_{ijk})^2+\sum_{j,i}(x_{il}-\mu_{ijk})^2\right)$$

Lesson: Statistical Independence is difficult to dodge.

Now it's Too Slow!

~2000 characters per page x
~100 character classes (English) x
32 fonts x
~20 prototype features x
~100 unknown features x
3 feature dimensions

= 38bn distance calculations per page...

Now it's Too Slow!

~2000 characters per page x
~100 character classes (English) x
32 fonts x
~20 prototype features x
~100 unknown features x
3 feature dimensions

= 38bn distance calculations per page...
... **on a 25MHz machine.**

What Were Other OCR Systems doing in 1988-1990?

Calera Wordscan: Required a $2000 "real computer" add-on board to run a fixed dimension feature space on a PC (that used a 12.5MHz 80286 in those days):

Recognita: Decision tree using topological-like features. Very fast in software. Accuracy degraded very fast on poor quality.

Neural Networks: Were just becoming popular. Would there be a startup threat?

Testing

Summary:

- Small test sets are meaningless -> Get lots of compute power and data.
- Test on very different data to the training data.
- Test every change.
- What you measure improves -> Measure lots of dimensions.
- If it can break it will.
- Make your code faster while waiting for tests to run.
- Think/test out of the box...

Creative testing

Christmas 1992: Company shutdown for 10 days. What should we do with all that compute power that would otherwise be wasted?

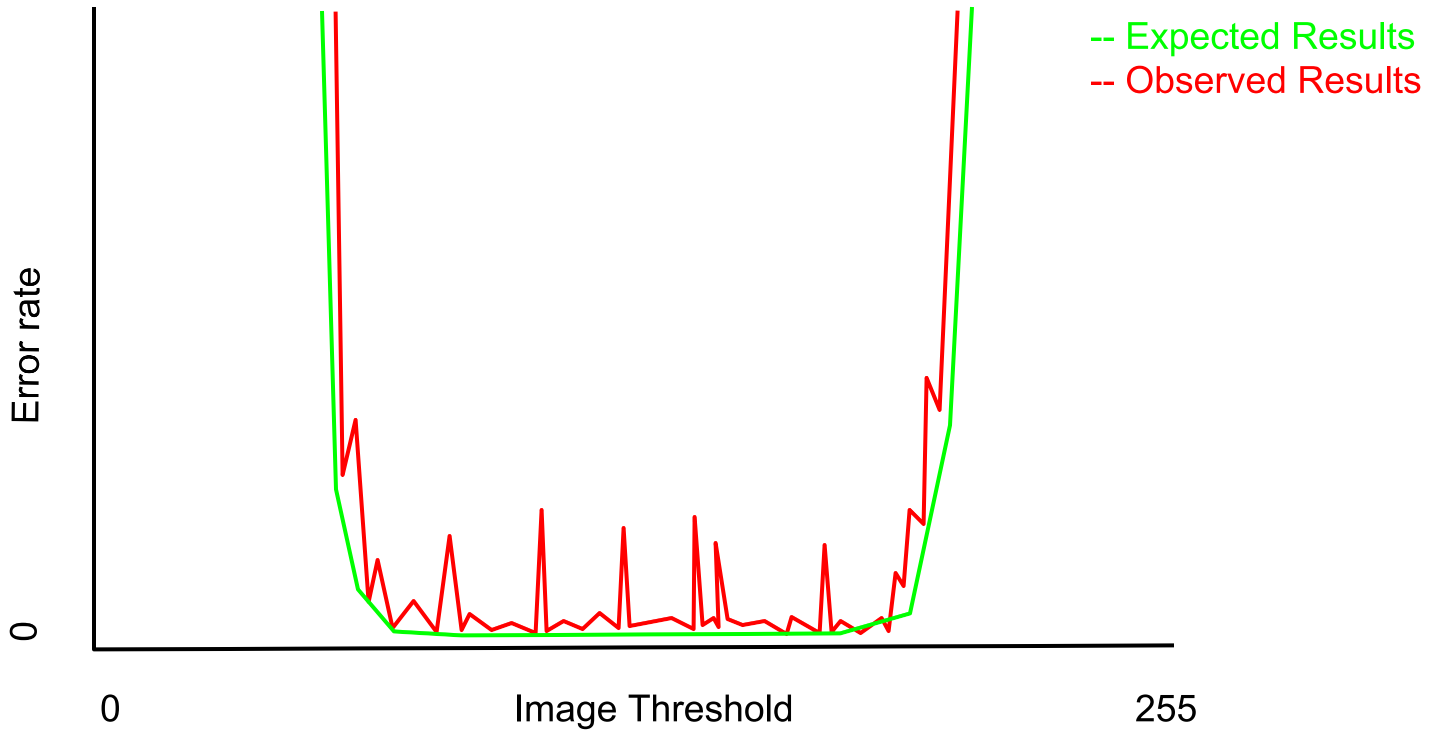
Creative testing

Christmas 1992: Company shutdown for 10 days. What should we do with all that compute power that would otherwise be wasted?

- Tune the hand-tuned parameters with a genetic algorithm.
- Process the 400 image test set thresholded at each possible threshold in the range 32-224

Thresholding Experiment Results

- Exposed several bugs
- Some fascinating accuracy results



The Dark Ages: 1995-2005: Gathering Dust

The Dark Ages

Commercial Engines:

- 1994: Caere buys Calera.
- 1995: First use of character n-grams in commercial system. (Wordscan 3.1).
- 1996: Caere buys Recognita.
- 2000: Scansoft buys Caere.
- Voting improves accuracy, but engine is much slower.

Tesseract:

- Ported to Windows.
- Moore's law:
  - 20x Memory Capacity.
  - 20x Speed.

2005: HP Open Sources Tesseract

The Open Source Era: 2006 - Present

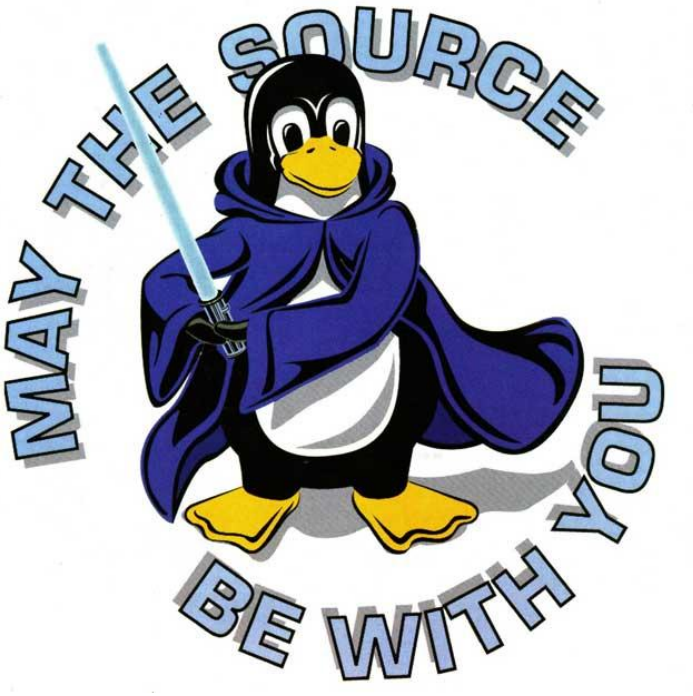
Tesseract source is available from:

http://code.google.com/p/tesseract-ocr

and for direct install on Linux via:

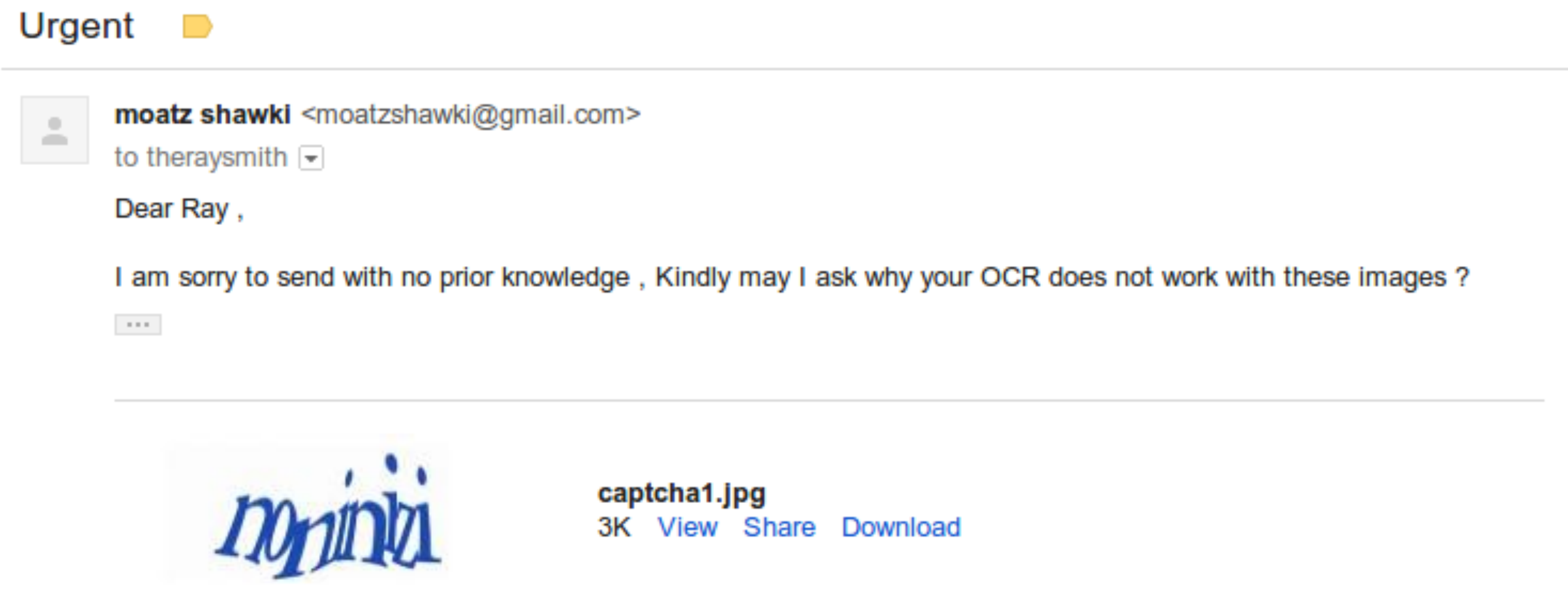
```
apt-get install tesseract-ocr
```

60+ Languages available.



Open Source OCR is Used for Various Applications

From my inbox:



Urgent

moatz shawki <moatzshawki@gmail.com>

to theraysmith

Dear Ray ,

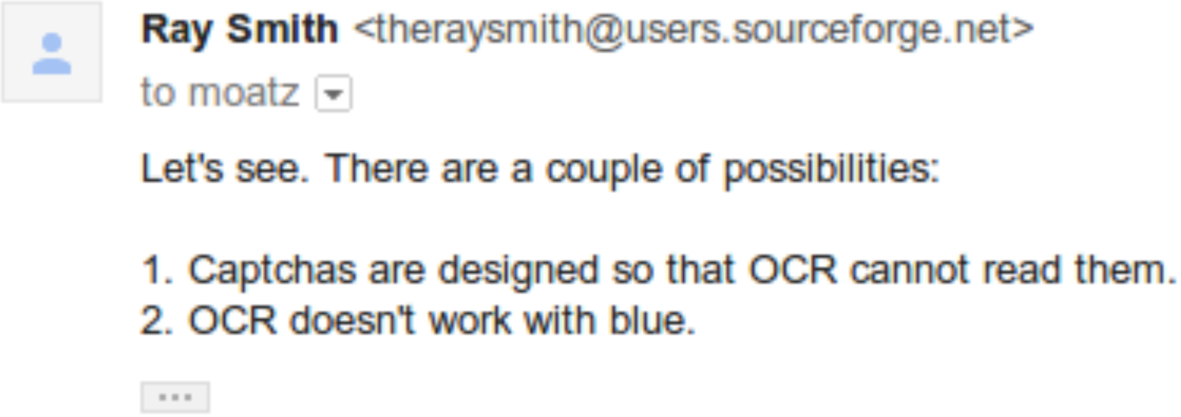
I am sorry to send with no prior knowledge , Kindly may I ask why your OCR does not work with these images ?

captcha1.jpg

3K View Share Download

Open Source OCR is Used for Various Applications

My reply:



Ray Smith <theraysmith@users.sourceforge.net>

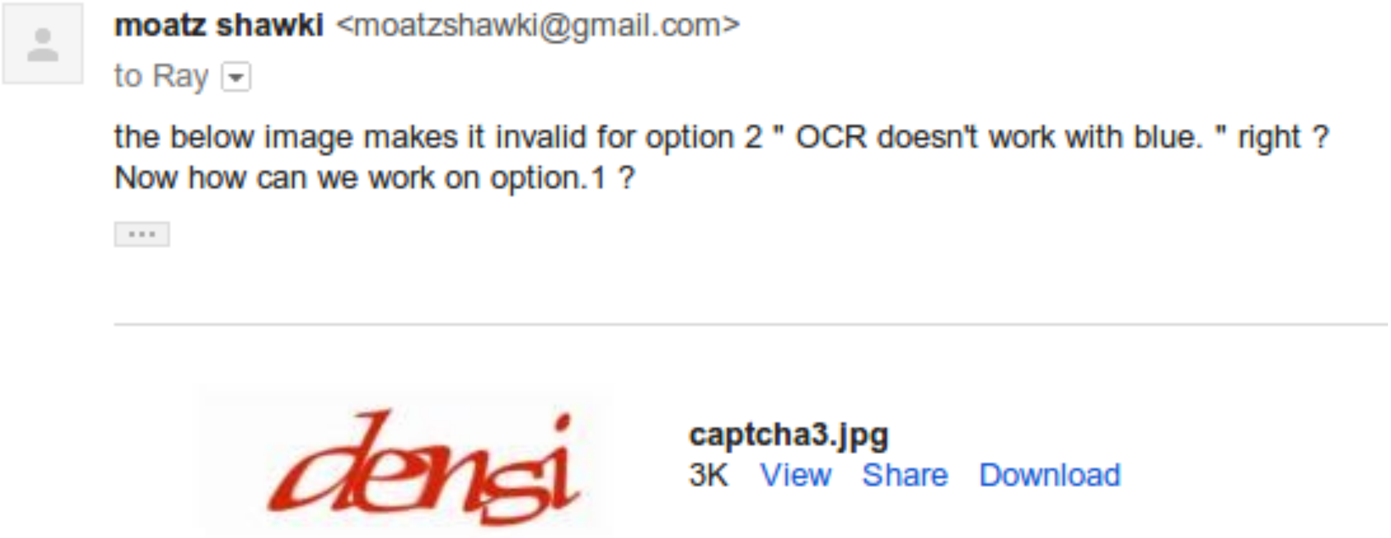
to moatz

Let's see. There are a couple of possibilities:

1. Captchas are designed so that OCR cannot read them.
2. OCR doesn't work with blue.

Open Source OCR is Used for Various Applications

He doesn't give up that easily:



moatz shawki <moatzshawki@gmail.com>

to Ray

the below image makes it invalid for option 2 " OCR doesn't work with blue. " right ?
Now how can we work on option.1 ?

captcha3.jpg
3K View Share Download

Recent Improvements

- Internationalize to 60+ Languages.
- Add Full Layout Analysis.
- Table Detection.
- Equation Detection.
- Better Language Models.
- Improved Segmentation Search.
- Word Bigrams.

Results

Language	Ground Truth Chars (million)	Ground Truth words (million)	Char Err Rate%	Word Err Rate%
English	271	44	0.47	6.4
Italian	59	10	0.54	5.41
Russian	23	3.5	0.67	5.57
Simplified Chinese	0.25	0.17	2.52	6.29
Hebrew	0.16	0.03	3.2	10.58
Japanese	10	4.1	4.26	18.72
Vietnamese	0.41	0.09	5.06	19.39
Hindi	2.1	0.41	6.43	28.62
Thai	0.19	0.01	21.31	80.53

Thanks for Listening!

Questions?