Building a fast digit recognition solution with Python

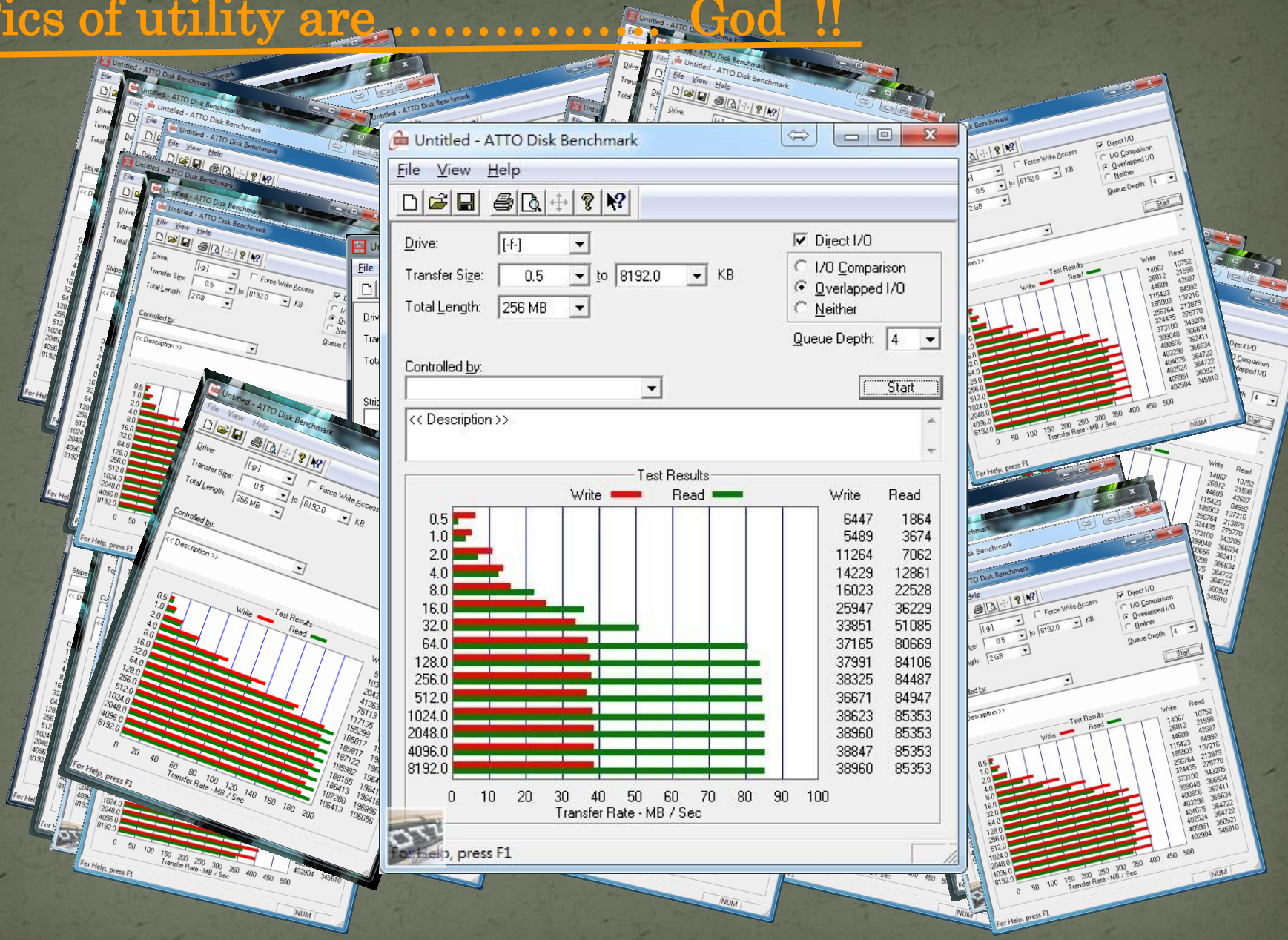
Stephen Hsu

http://about.me/cchhsu

2013.05.22

One day ……

Pics of utility are ..... God !!




OH !
NO

Image Recognition

I don't want to use openCV

There is a python module named

pytesser

PyTesser 0.0.1

Optical Character Recognition module for Python

PyTesser is an Optical Character Recognition module for Python. It takes as input an image or image file and outputs a string.

PyTesser uses the Tesseract OCR engine, converting images to an accepted format and calling the Tesseract executable as an external script. A Windows executable is provided along with the Python scripts. The scripts should work in other operating systems as well.

Author: Michael J.T. O'Kelly
Maintainer: Michael J.T. O'Kelly
Home Page: http://code.google.com/p/pytesser/
Download URL: http://code.google.com/p/pytesser/downloads/list
Keywords: Python, OCR, Optical Character Recognition, Tesseract
License: Apache License 2.0
Requires PIL

```
>>> from pytesser import *
>>> image = Image.open('fnord.tif')  # Open image object using PIL
>>> print image_to_string(image)     # Run tesseract.exe on image
fnord
>>> print image_file_to_string('fnord.tif')
fnord
```

Tesseract OCR

- ✓ Introduction
  - Open source OCR engine
  - Started at the HP labs between 1985 and 1994
  - C, C++
  - Google used it for document scan project

- ✓ Training
  - Images fonts of Benchmark tools are non-standard.
  - Training Process is fun !

Technologies

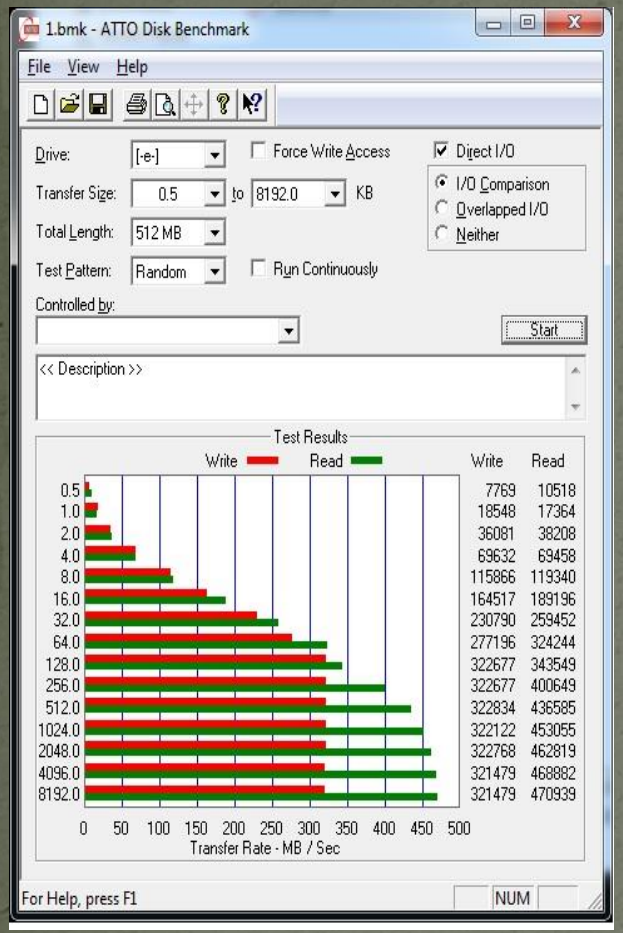
- ✓ Common
  - ✦ Windows
  - ✦ Linux
  - ✦ Python 2.7
- ✓ 3rd party APP
  - ✦ jTessBoxEditor
  - ✦ Tesseract OCR

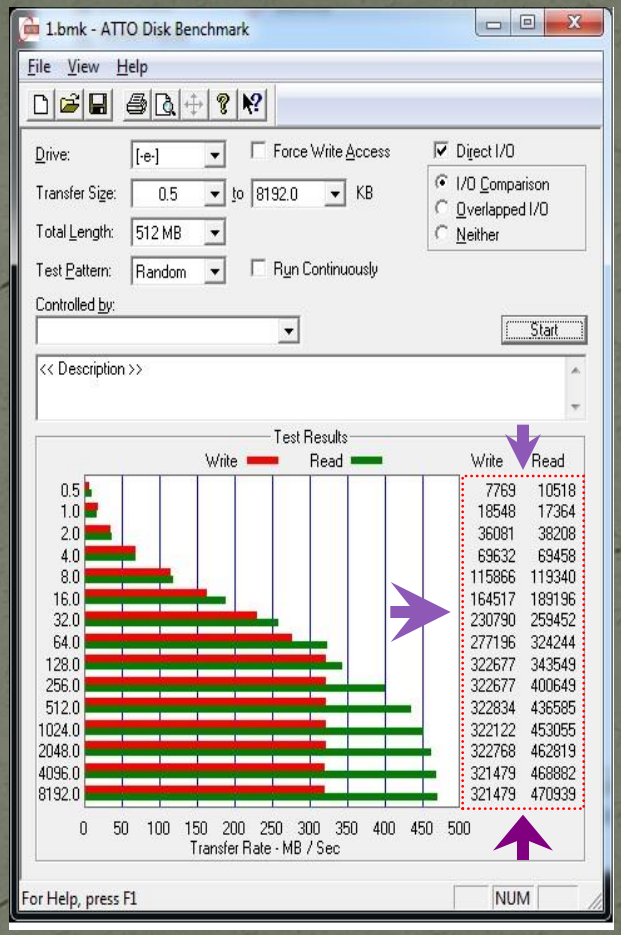
- ✓ Imaging Process
  - ✦ pytesser
  - ✦ PIL
  - ✦ datetime, time
  - ✦ re
  - ✦ os, sys
  - ✦ csv
  - ✦ glob

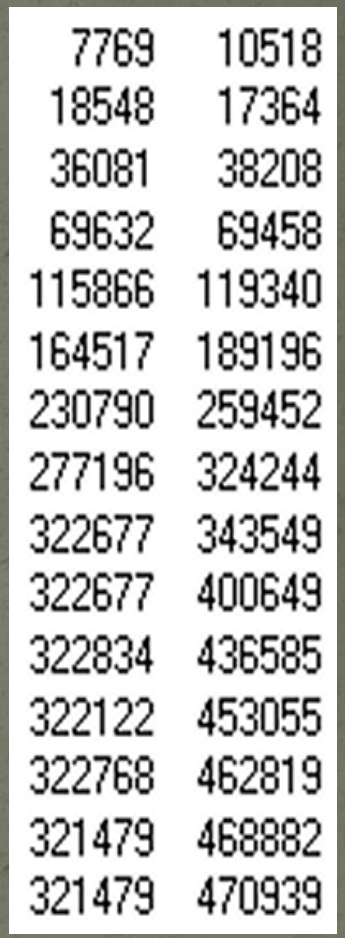
Image Pre-Processing

- ✓ Image Cropping
  - Crosshairs
- ✓ Pixel Interpolation
  - ➔ 20M Pixel ➔ 99M Pixel
  - ➔ Recognition rate from 93% to 99.99%
- ✓ Binarization Processing
- ✓ Lines Recognition

Image Pre-Processing Sample







Original Image

Image Cropping

Pixel Interpolation

then .....

```
7769
18548
36081
69632
115866
164517
230790
277196
322677
322677
322834
322122
322768
321479
321479

10518
17364
38208
69458
119340
189196
259452
324244
343549
400649
436585
453055
462819
468882
470939
```

Extracted Text

- ✓ Parser
  - In: Number
  - Out: Structured number
- ✓ Store & Convert
  - In: Structured Number
  - Out: CSV File

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	filename	0.5KB Wri	0.5KB Rea	1KB Write	1KB Read	2KB Write	2KB Read	4KB Write	4KB Read	8KB Write	8KB Read	16KB Writ	16KB Read	32KB Writ	32KB Read	64KB Writ
2	C:\ATTO\1	7769	10518	18548	17364	36081	38208	69632	69458	115866	119340	164517	189196	230790	259452	277196
3	C:\ATTO\2	7944	9124	18040	17235	30720	41265	69113	76039	103158	121362	165332	193242	231941	263969	277196
4	C:\ATTO\3	8426	10011	16007	17321	32256	38019	68942	69956	115580	119340	168139	178381	230790	258819	281999
5	C:\ATTO\4	7788	8766	15321	17152	32869	37376	66228	66560	106677	114688	170905	181554	229651	267766	275854
6	C:\ATTO\5	7808	8809	16181	17193	33623	36914	64726	68942	114401	114401	160878	181554	227962	267766	278552

Sample_20130522214241.csv

END