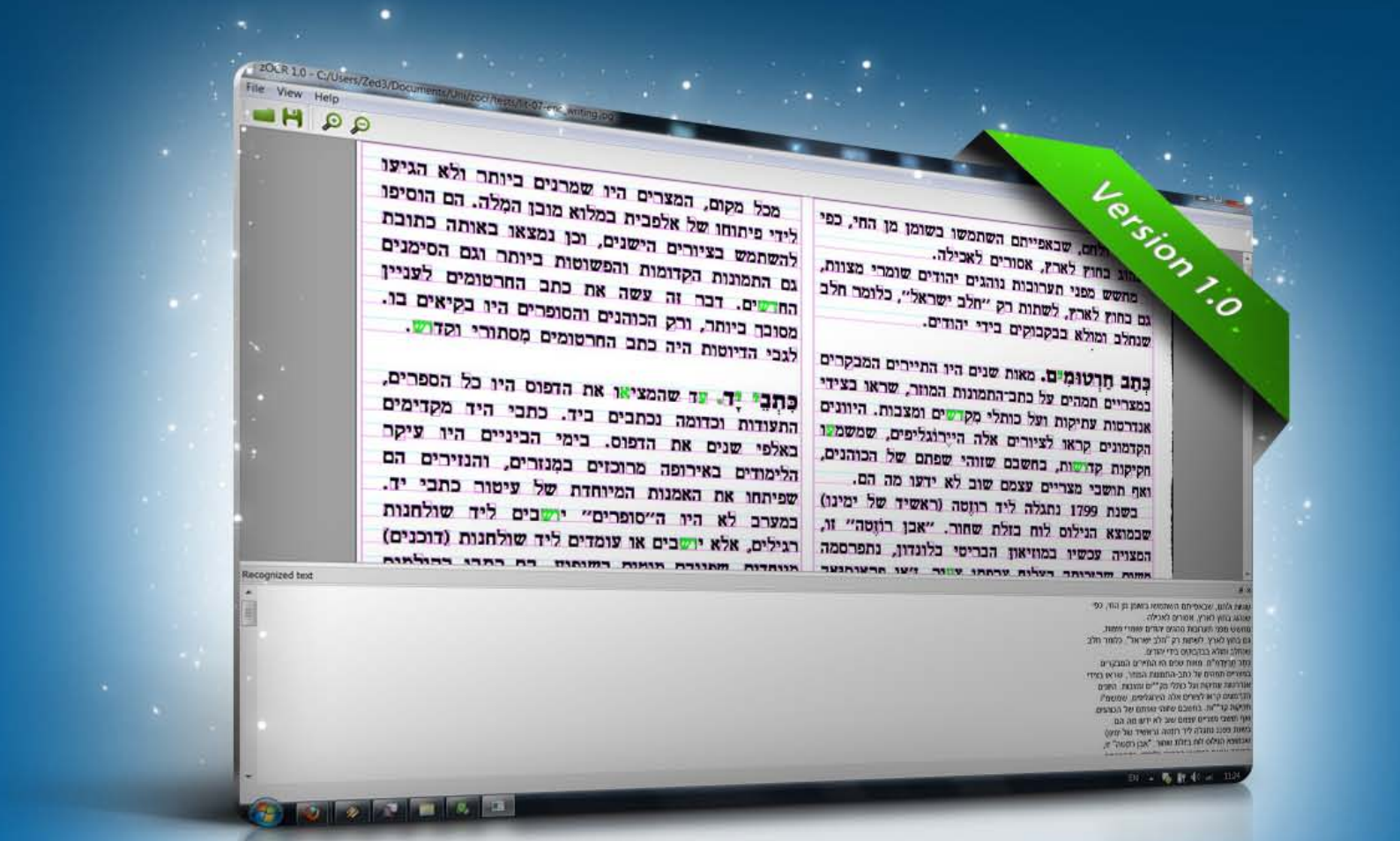
zOCR - Hebrew Optical Character Recognition



Contents

General

BSc Project by Shai Shofet and Jenya Shtopelman

Project Website : http://zocr.zed3.net

Project Description (from the projects' website) : Improving an open source OCR program to include Hebrew writing recognition.

Tested Platforms

Name	OS	Updated	Pros	Cons
Tessaract	Linux, Windows	June 2009	Accurate Relatively Updated	Supports up to 32 different fonts
Ocropus	Linux	May 2009		Command Line
hOCR	Linux	August 2008	Supports RTL	Disbanded.

Main Program Classes

PixmapViewer

Controls the supported files, Image manipulations (brightness, image size)

MainWindow

Main menu, hOCR initialization, Program Options and keyboard command.

Hopfield

a Neuro Network class.

Takes a given char and tries to recognize.

If the recognition fails, it will ask the user for the appropriate letter and will add random noise as well.

hOCR

uses the libhocr created by Yaakov Zamir

scans document images, improve the image, analyses the page layout, recognises the characters and outputs the text

several modifications were made (see section regarding problems and improvements)

hSpell

free Hebrew linguistic project written by Nadav Har'El and Dan Kenigsberg..

Problems and Solutions

hSpell Problems

We used hSpell as spell check and dictionary.

hSpell is made for linux distros and so we compiled on a Unix system, and "transferred" to Windows platform, and added it to tesseract.

RTL Support - With all the tested platforms, non had RTL Support. The initial solution was to do LTR scan, flip the result, match the dictionary and print on screen.

As we went on, there was a demand for punctuation and hand writing recognition.

That was impossible to achieve with the tested platforms. We did manage to add Hebrew support to Tesseract, and could punctuation signs (most of the time), but hand-writing support was impossible as the program supports up to 32 different font types per language.

Possible solutions were :

Disband hand-writing recognition.

Switch to a different platform.

After the meeting, it was decided to switch to a different platform in order to keep the writing recognition, and so, hOCR was selected.

hOCR Problems:

the hOCR project seems to be disbanded, the website is not updated later than July 2008.

author does not answer emails.

the main hOCR problem was memory leaks. Seems that the latest stable version suffer from major memory leaks under Windows.

the library was unable to process any image.

We did manage to solve the memory leak problem, however there are other issues we still could not solve

hSpell issue:

Mixing hOCR and hSpell under Windows crashes the program. We could not find appropriate solution to that problem except for replacing hSpell with another speller, aSpell.

hand writing issue:

It seems that although we fixed the libhocr main issue, we could not get it to work with the NN class we wrote.

The problem seems to be with libocr graphics process. When we paused the process and tried to transfer the result to the NN class, the program would crash.

We could not fix this bug in time, the solution is to create a buffer between the libhocr and the NN class.

Program Advantages

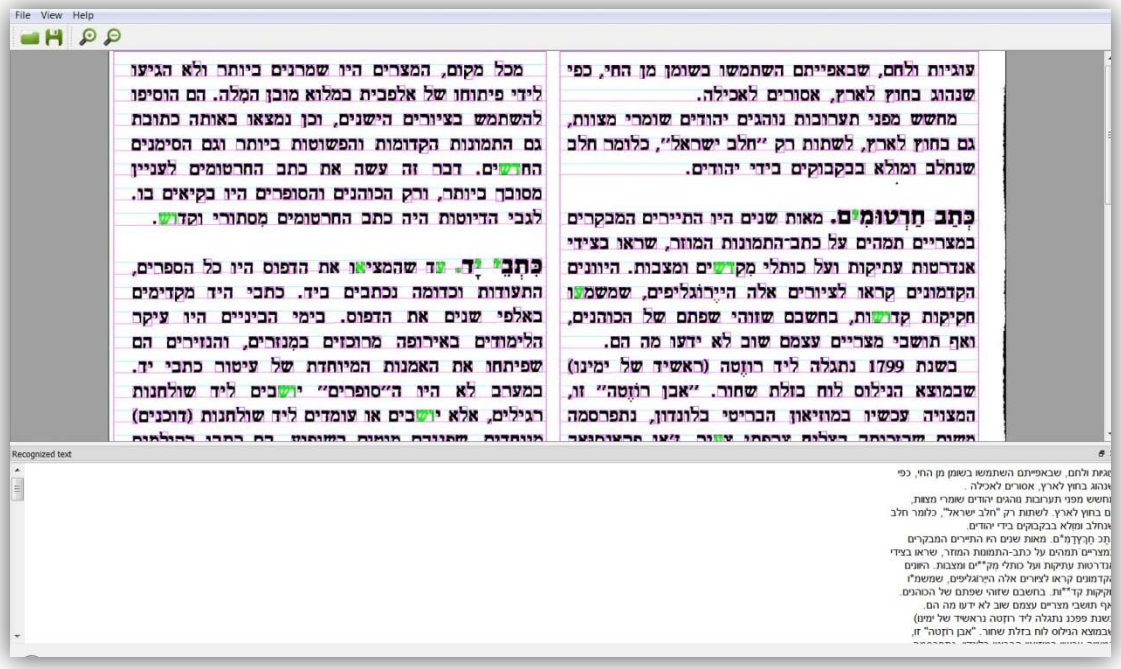
- Fast - it is a lot faster than commercial products.
- Free - it is under GPL license (including open source products like hOCR).
- OOP based - easy to tweak and build.

What's next?

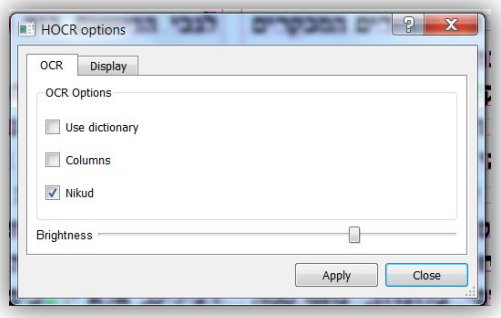
- Fix the hand-writing recognition bug.
- Create a new source from scanner.
- aSpell instead of hSpell.

User Guide

Main Program Window:



Program Settings:



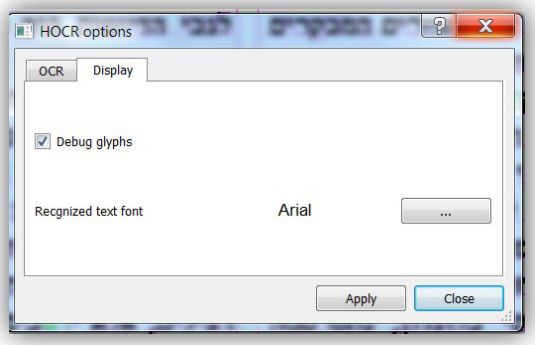
OCR Options:

Use Dictionary - Enables hSpell support for unidentified words.

Columns - Enables columns support, instead of linear output.

Nikud - Enables punctuation characters.

Brightness - Enables control over the source brightness to improve detection.



Display Options

Debug glyphs - Shows row detection and character detection.

Recognized text font - Controls the output text font type and size.

Options are applied as soon as you click the "Apply" button, will reload the selected image and will re-OCR according to the newly selected options

Program Menu

File Menu

Open Existing Image - Opens file browser, filtered by the supported file types (see section) for OCR.

Save Output Text - Opens file browser for saving the OCRed text.

zOCR options - Opens the Options window.

Quit - Exits the program.

View Menu

Zoom in - Render the image +1 size.

Zoom out - Render the image -1 size.

1:1 - Renders the image in the original resolution.

Best fit - Renders the image to fit the current program window.

About Menu

About - Opens the program about us window.

About Qt - Opens Qt about window.

Technologies Used

Qt Framework
Version: 4.7.0

Qt is a cross-platform application development framework widely used for the development of GUI programs , console tools and servers.
Qt is most notably used in Google Earth, KDE), OPIE, Skype, VLC media player and VirtualBox.
Qt uses standard C++ but makes extensive use of a special pre-processor to enrich the language.
It runs on all major platforms and has extensive internationalization support.
Distributed under the terms of the GNU Lesser General Public License (among others), Qt is free and open source software. All editions support a wide range of compilers, including the GCC C++ compiler and the Visual Studio suite.

Microsoft Visual Studio
Version: 2008

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

LibHocr
Version: 0.10.5

LibHocr is a GNU Hebrew optical character recognition library. It scans document images, improve the image, analyses the page layout, recognises the characters and outputs the text. The output texts are now editable text, ready for your blog, word processor or any other use.

Program: Subversion (SVN)

Version: 1.6.12

Subversion (SVN) is a version control system initiated in 2000 by CollabNet Inc. It is used to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-compatible successor to the widely used Concurrent Versions System (CVS).

Subversion is well-known in the open source community and is used on many open source projects such as: Apache Software Foundation, KDE, GNOME, Free Pascal, FreeBSD, GCC, Python, Ruby, and Mono. SourceForge.net and Tigris.org also provide Subversion hosting for their open source projects.

Program: TortoiseSVN
Version: 1.6.10

TortoiseSVN is a Subversion client, implemented as a Microsoft Windows shell extension. It is free software released under the GNU General Public License.

TortoiseSVN won the SourceForge.net 2007 Community Choice Award for Best Tool or Utility for Developers

Programmer's Guide

Classes Diagrams

MainWindow
+MainWindow(QWidget *parent=0) +createActions() +createMenus() +createToolbars() +on_aboutButton_clicked() +on_exitButton_clicked() +on_loadButton_clicked() +on_saveButton_clicked() +on_zoomInButton_clicked() +on_zoomOutButton_clicked() +on_zoomNormalButton_clicked() +on_changeFont_clicked() +on_bestFit_clicked() +on_options_clicked() +apply_hocr_settings() +saveStatus() +loadStatus() +closeEvent(QCloseEvent *event) +doOCR() +timerEvent(QTimerEvent *event) -viewImage(QString fileName) -saveHTML(QString fileName, QString text) -saveText(QString fileName, QString text, bool unicode)

PixmapViewer
+PixmapViewer() +PixmapViewer(QWidget *parent) +loadImage(QString f) +setPixmap(QPixmap p) +setImage(QImage i) +wheelEvent(QWheelEvent *e) +paintEvent(QPaintEvent *event) +resizeEvent(QResizeEvent * event) +zoomBy(float f) +getBestFit() +setBestFit(bool a) +getZoomFactor() +setZoomFactor(float f)

HOCRThread
+HOCRThread(hocr_pixbuf *p, hocr_text_buffer *t); +Run() -hocr_pixbuf -hocr_text_buffer

Hopfield
+print_letter(Letter &L) +get_letters(vector<Letter> &letters) +insert_letter(Weights &W, Letter &L) +energy(Weights &W, Letter &L) +check_pattern(Weights &W, Letter &L, bool disp) +noise(vector<Letter> &letters,double rate) +select_letters(int selected[], int &usedsize) -select_display(bool &disp) -select_noise(double &rate) +main()

Major Methods and Functions

MainWindow Class

Name	Description
HOCRThread(hocr_pixbuf *p, hocr_text_buffer *t)	**Define hOCR**
HOCRThread::run()	**Starts hOCR**
MainWindow(QWidget *parent):QMainWindow(parent)	**Creates program window**
createActions()	**Defines keyboard shortcuts, icons and Options**
createMenus()	**Creates program main menu**
createToolbars()	**Creates program toolbar**
saveStatus()	**Saves current settings before shutdown**
loadStatus()	**Loads last session settings**
viewImage(QString fileName)	**Loads an image**
saveHTML(QString fileName, QString text)	**Saves recognized text**

Hopfield Class

Name	Description
get_letters(vector<Letter> &letters)	**get the input letters from a file**
insert_letter(Weights &W, Letter &L)	**learn a new letter**
energy(Weights &W, Letter &L)	**calculate the energy of a letter**
check_pattern(Weights &W, Letter &L, bool disp)	**converge to a stable pattern**
noise(vector<Letter> &letters,double rate)	**add random noise to samples**
select_letters(int selected[], int &usedsize)	**letters selection by user**

PixmapViewer Class

Name	Description
paintEvent	**Loads an image**
resizeEvent(QResizeEvent * event)	**Resize the image**
PixmapViewer::zoomBy(float f)	**Image zoom according to factor**

hOCR Class

Name	Description
hocr_do_ocr (hocr_pixbuf * pix, hocr_text_buffer * text_buffer)	**Start OCRing**
resizeEvent(QResizeEvent * event)	**Resize the image**
PixmapViewer::zoomBy(float f)	**Image zoom according to factor**