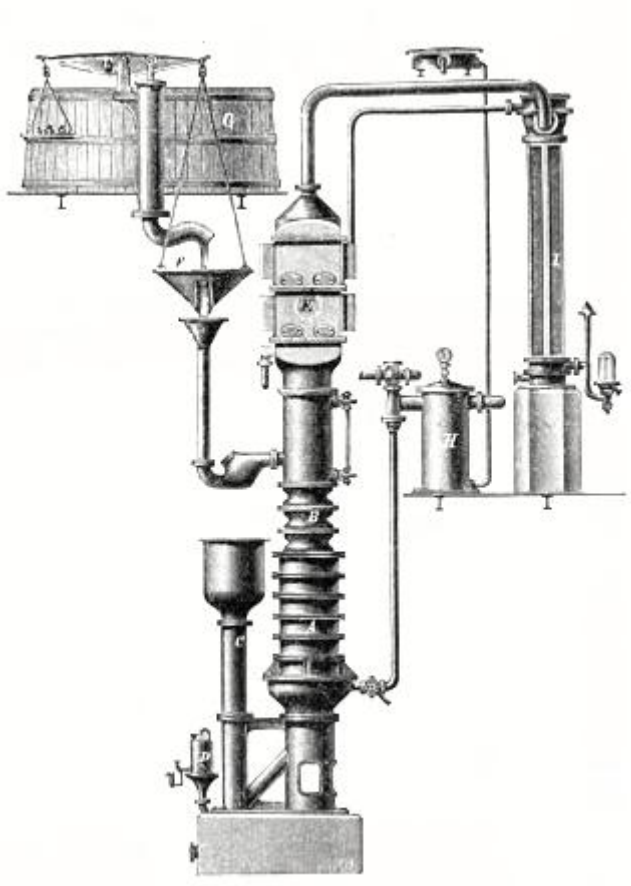


cbrTekStraktor

“Strange adventures on other planets”

Space Detective - Issue 4
Published Apr 1952 by Avon Publications.

Chapter 0

Preface

Notices

Copyright (c) 2017 - cbrTekStraktor

cbrTekStraktor is free software

Permission is granted to copy, distribute and/or modify this software under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA to obtain the GNU General Public License
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

GNU General Public License: www.gnu.org/copyleft/gpl.html

Contact details for copyright holder: cbrtekstraktor@gmail.com

Trademarks

cbrTekStraktor relies on the following freely available technology
Java SDK 1.5 (or higher)

The cbrTekStraktor java source code has been developed to be deployed on various platforms and operating systems. The java source code has been tested on the following platforms and Operating Systems.

Platform	Operating system
Intel – AMD	Windows 7 (32 and 64 bit)
Intel – AMD	Linux Ubuntu 16.04

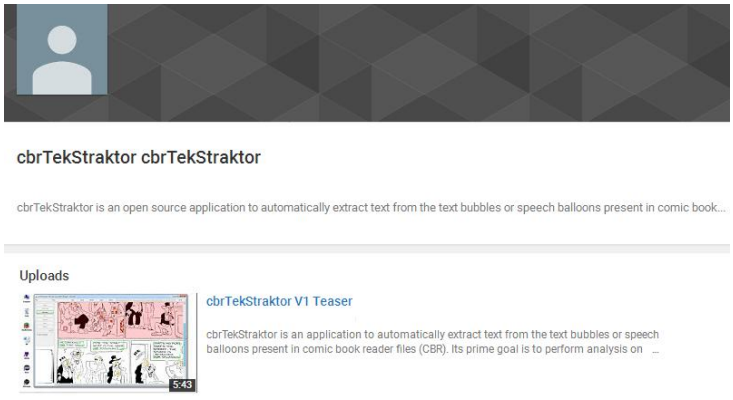
Release note

Release history of this document

Date	Document version	cbrTekStraktor version
2017-06-05	Draft	V01 – Build 2017_06_05
2017-06-12	Draft	V02 – Build 2017_06_12

YouTube channel

https://www.youtube.com/channel/UCy0NfU7-N8RcyI-rj3fSCEw



Comments welcome
Mail your comments or defects reports to : cbrtekstraktor@gmail.com

Chapter 1

Introduction

cbrTekStraktor is an application to automatically extract text from the text bubbles or speech balloons present in comic book reader files (CBR). Its prime goal is to perform analysis on the texts of comic books. cbrTekStraktor can however also be used for scanlation or similar purposes.

The application also enables to manually define text areas in CBR files. The application comprises a simple graphical editor for further processing the extracted text.

The text extraction is achieved by a combination of statistical and graphical processing operations. It is based on the following 3 major algorithms

- Binarization of color images (Niblak and other methods)
- Connected components
- K-Means clustering

Apache Tesseract is used to perform Optical Character Recognition on the extracted text.

cbrTekStraktor has some known limitations. It has been conceived to perform extraction of Western (Roman) characters and will only work on comic pages with a light background.

A subsequent version of the application will

- integrate with translation software in order to provide automated translation of comic book texts.
- Provide a mechanism to automatically re-inject translated text into the text balloons

[Wikipedia] Scanlation (also scanslation) is the scanning, translation, and editing of comics from a language into another language. Scanlation is done as an amateur work and is nearly always done without express permission from the copyright holder. The word "scanlation" is a portmanteau of the words scan and translation.

[Wikipedia] A comic book archive or comic book reader file (also called sequential image file) is a type of archive file for the purpose of sequential viewing of images, commonly for comic books. Comic book archive files mainly consist of a series of image files, typically PNG or JPEG files, stored as a single archive file. The file name extension indicates the archive type used, e.g. CBR or CBRZ

Associated documents

[01] Christophe Rigaud, Norbert Tsopze, Jean-Christophe Burie and Jean-Marc Ogier : Robust frame and text extraction from comic books, La Rochelle (France) and Yaoundé (Cameroon)

[02] Christophe Rigaud, DimsthenisKaratzas, Joost Van De Weijer, Jean-Christophe Burie and Jean-Marc Ogier : Automatic tekst location in scanned comic books, Barcelona (Spain), 2013

[03] Christophe Rigaud, Jean-Christophe Burie and Jean-Marc Ogier : An active contour model for speech balloon detection in comics, La Rochelle (France)

[04] Karl Tombre, SalvatorTabbone, LoïcPélissier, Bart Lamiroy and Philippe Dosch : Text/graphics separation revisited, Vandoeuvre-lès-Nancy (France)

[05] Muhammad MuzamilLuqman, Hoang Nam Ho, Jean-Christophe Burie and Jean-Marc Ogier : Automatic indexing of comic page images for query by example based focused content retrieval, la Rochelle (France)

[06]Zhongliang Fu, FulinBian, Songtao Zhou and QingwuHu : Algorithm for fast detection and identification of characters in gray-level images, Wuhan (Republic of China)

Public domain Comic Book Archives

The Comic Book Images which are used in this manual have been downloaded from the "Digital Comic Museum". DCM is a great site for downloading free public domain Golden Age Comics. All files here have been researched by DCM's staff and users to make sure they are copyright free and in the public domain.

http://digitalcomicmuseum.com/

Chapter 2

Installation

Distribution

The most recent version of cbrTekStraktor is published on

- GitHub (https://github.com/cbrTekStraktor/cbrTekStraktor)
- SourceForge (https://sourceforge.net/projects/cbrTekStraktor).

The material distributed via GitHub and SourceForge comprises the source code, an executable JAR file and this reference manual.

Quick Installation

Prerequisites

A recent Java Runtime Engine (JRE) or Java Software Development Kit (JSDK) is required.

cbrTekStraktor has been tested with 64-bit Oracle Java SDK 7 and 8 on Windows 7 and Linux Ubuntu 16.04

The application is based on standard Java Swing functionality and will therefore more than likely also function correctly on other operating systems (e.g. OS-X, Red Hat, Windows 10, etc.) and Java other JRE's and JSDKs.

WINDOWS

Windows users need to manually create the folder C:\temp and c:\temp\cbrTekStraktor\bin

LINUX

Linux users need to create the directory $HOME/cbrTekStraktor and $HOME/cbrTekStraktor/bin, in which $HOME is to be substituted by the actual location of the Linux user's home directory.

Installation

Just put the JAR file (cbrTekStraktor.jar) in c:\temp\cbrTekStraktor\bin (or $HOME/cbrTekStraktor/bin)

Starting the application

It should suffice to double click on the cbrTekStraktor Jar file to start and run the application.

In the event that double-clicking on the Jar file does not work, you can manually start the application as follows

WINDOWS

Open a Windows command window
CD c:\temp\cbrTekStraktor
java –jarcbrTekStraktor.jar

LINUX

Open a Linux command window
cd $HOME/cbrTekStraktor/bin
java –jar ./cbrTekStraktor.jar

Command line parameters

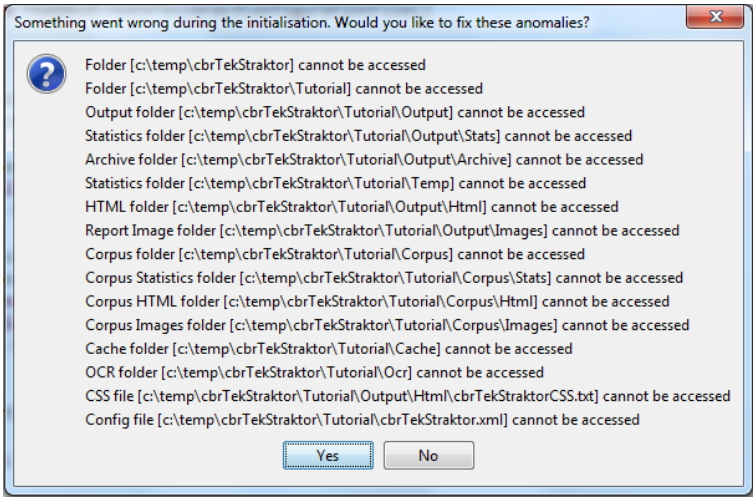
The following command line parameters are supported

-D {project folder name}. The –D options enables to specify the foldername of the project to be opened. If the project root folder is not specified as a command line parameter, it will be defaulted to "c:\temp\cbrTekStraktor" or $HOME/cbrTekStraktor

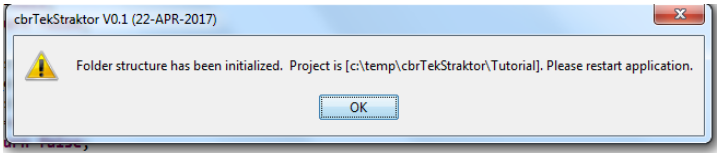
First time usage

The following dialog will be shown when the application is started for the first time or whenever one of the required file system components is found to be missing.

The dialog reportson all folders which are missing and will prompt you to confirm whether those missing folders can be created automatically. Click "Yes" if you want to have the missing folders created.



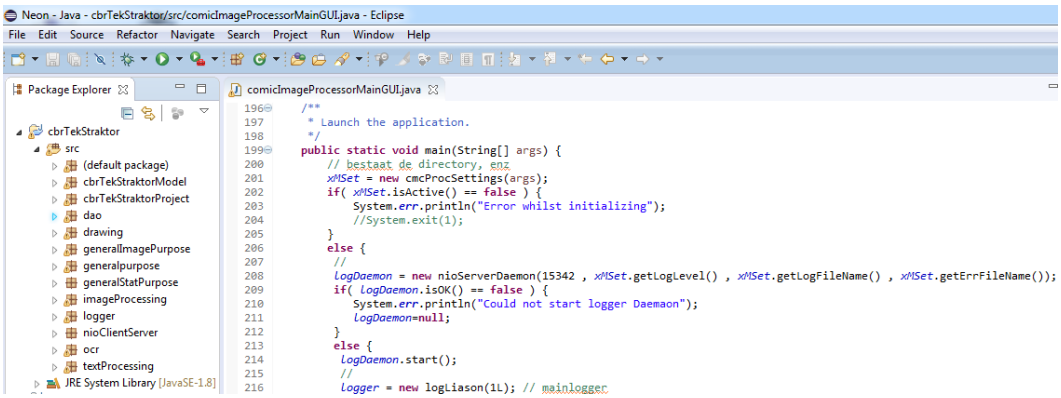
Upon having successfully completed the creation of the missing folders, the following dialog will be displayed. You should close the application at this stage and then restart.



Source code installation

Download or clone the cbrTekStraktor application package from GitHub or SourceForge using the approach you are comfortable with and install the material in the workspace folder of your preferred Java IDE.

The source code of cbrTekStraktor was created using the Eclipse Neon IDE. The following screenshot depicts the structure of the Java packages.



[Wikipedia] A JAR (Java ARchive) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file for distribution. JAR files are archive files with which include a Java-specific manifest file. They are built on the ZIP format and typically have a .jar file extension.

Apache Tesseract Installation

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, Version 2.0 Development has been sponsored by Google since 2006. Tesseract is considered one of the most accurate open-source OCR engines available.

cbrTekStraktor uses tesseract (e.g. tesseract-ocr 4.00.00alpha) to perform OCR. Given that only basic OCR functionality is needed, it can safelybe assumed that other versions of Tesseract will integrate with cbrTekStraktor too.

Read the Tesseract home page on GitHub for a quick introduction
https://github.com/tesseract-ocr/tesseract/wiki

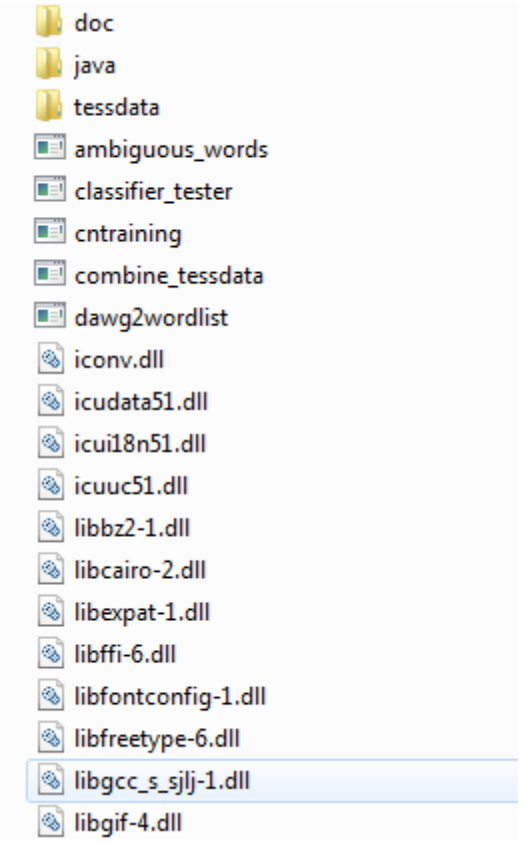
MICROSOFT WINDOWS

The binaries of the Tesseract OCR engine can be found on https://github.com/tesseract-ocr/tesseract/wiki/Downloads

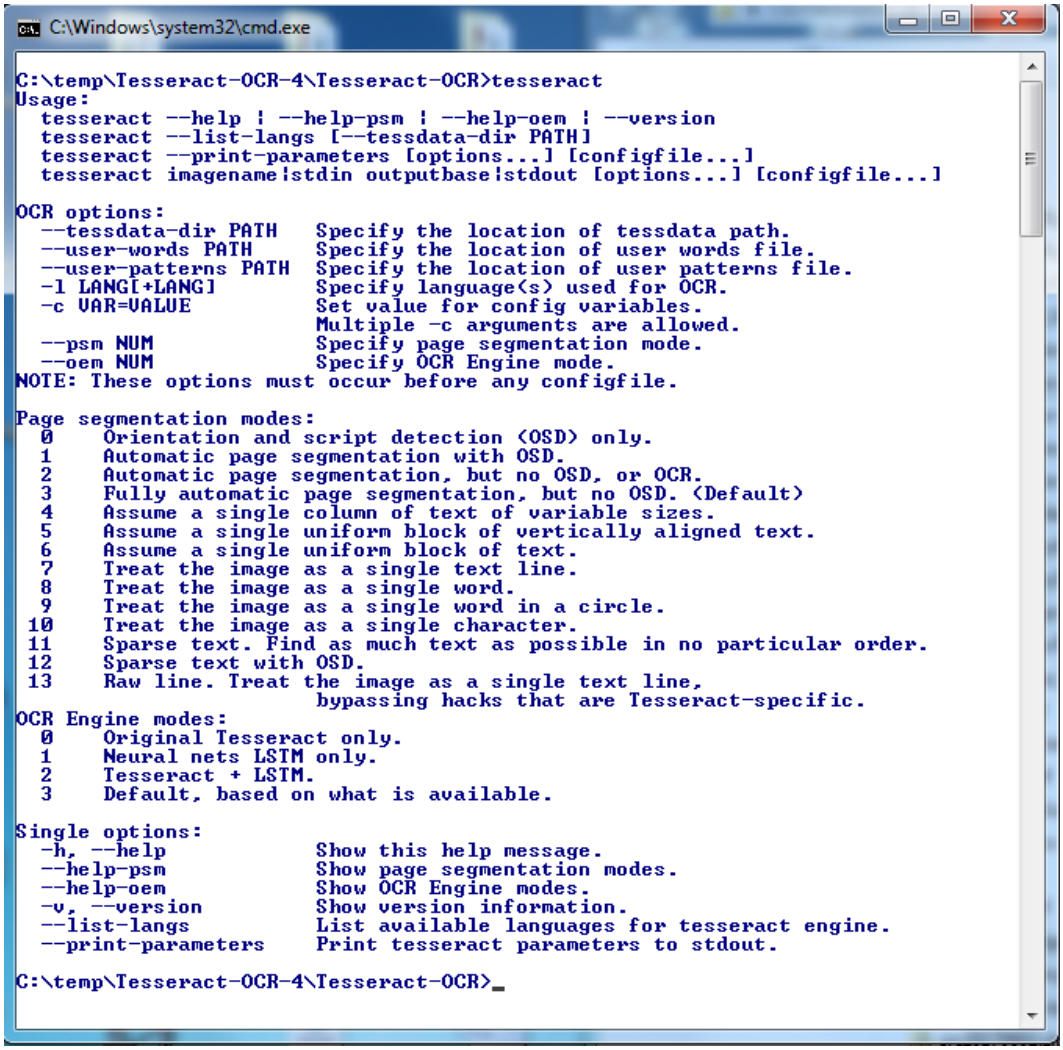
Whilst installing Tesseract make sure to make a note where the installer is putting the binaries. cbrTekStraktor accesses the Tesseract OCR client via the Windows command shell or the Linux shell. You therefore need to set the name of the folder holding the Tesseract binaries via the Project Configuration dialog.

You might also consider installing additional language packs for Apache Tesseract. cbrTekStraktor will detect which language packs have been installed and use those when appropriate. Language packs can be found via https://github.com/tesseract-ocr/tesseract/wiki

Once installed Tesseract'sinstallation directory should resemble the following.



You should run an installation test on Tesseract by issuing the following instruction from a command window: tesseract. This will result in an exhaustive usage message. Alternatively run the tesseract - - versioncommand. A double dash is required.


C:\Windows\system32\cmd.exe
C:\temp\Tesseract-OCR-4\Tesseract-OCR>tesseract
Usage:
tesseract --help | --help-psm | --help-oem | --version
tesseract --list-langs [--tessdata-dir PATH]
tesseract --print-parameters [options...] [configfile...]
tesseract imagename|stdin outputbase|stdout [options...] [configfile...]
OCR options:
--tessdata-dir PATH Specify the location of tessdata path.
--user-words PATH Specify the location of user words file.
--user-patterns PATH Specify the location of user patterns file.
-l LANG[+LANG] Specify language(s) used for OCR.
-c VAR=VALUE Set value for config variables.
Multiple -c arguments are allowed.
--psm NUM Specify page segmentation mode.
--oem NUM Specify OCR Engine mode.
NOTE: These options must occur before any configfile.
Page segmentation modes:
0 Orientation and script detection (OSD) only.
1 Automatic page segmentation with OSD.
2 Automatic page segmentation, but no OSD, or OCR.
3 Fully automatic page segmentation, but no OSD. (Default)
4 Assume a single column of text of variable sizes.
5 Assume a single uniform block of vertically aligned text.
6 Assume a single uniform block of text.
7 Treat the image as a single text line.
8 Treat the image as a single word.
9 Treat the image as a single word in a circle.
10 Treat the image as a single character.
11 Sparse text. Find as much text as possible in no particular order.
12 Sparse text with OSD.
13 Raw line. Treat the image as a single text line,
bypassing hacks that are Tesseract-specific.
OCR Engine modes:
0 Original Tesseract only.
1 Neural nets LSTM only.
2 Tesseract + LSTM.
3 Default, based on what is available.
Single options:
-h, --help Show this help message.
--help-psm Show page segmentation modes.
--help-oem Show OCR Engine modes.
-v, --version Show version information.
--list-langs List available languages for tesseract engine.
--print-parameters Print tesseract parameters to stdout.
C:\temp\Tesseract-OCR-4\Tesseract-OCR>

LINUX (UBUNTU)

The installation of Tesseract 3 on Ubuntu 16.04 is straightforward.There are plenty of resources on the World Wide Web commenting on how to install and use Tesseract on Linux. The following URL explains how to install Tesseract on Ubuntu 16.04.
https://www.howtoforge.com/tutorial/tesseract-ocr-installation-and-usage-on-ubuntu-16-04/

In a nutshell, you need to consecutively run the following commands in a Unix shell

>sudo apt install tesseract-ocr
>sudo apt-get install tesseract-ocr-[lang]

The first command will prompt for the root password and then install the latest version of Tesseract for Ubuntu.
The second command will install a language pack. For example tesseract-ocr-fra will install the French language pack. Re-run the command for any of the language packs you want to install.

Once the installation is complete you should perform a quick installation test on Linux. Open a Unix command window and run the following two commands
In order to determine where tesseract has been installed, issue the command "type –a tesseract". In most cases this will result in /usr/bin/tesseract.
Should the installation directory be different from /usr/bin you will need to correctly configure the Tesseract Installation Folder parameter on your cbrTekStraktorproject. The section "HowTo: Projects" provides detailed instructions on cbrTekStraktor projects and how to configure the Tesseract installation folder.
Next simply run the command "tesseract" in the Unix shell. Detailed status and error messages will be displayed.

&&TODO screen shot

Chapter 3

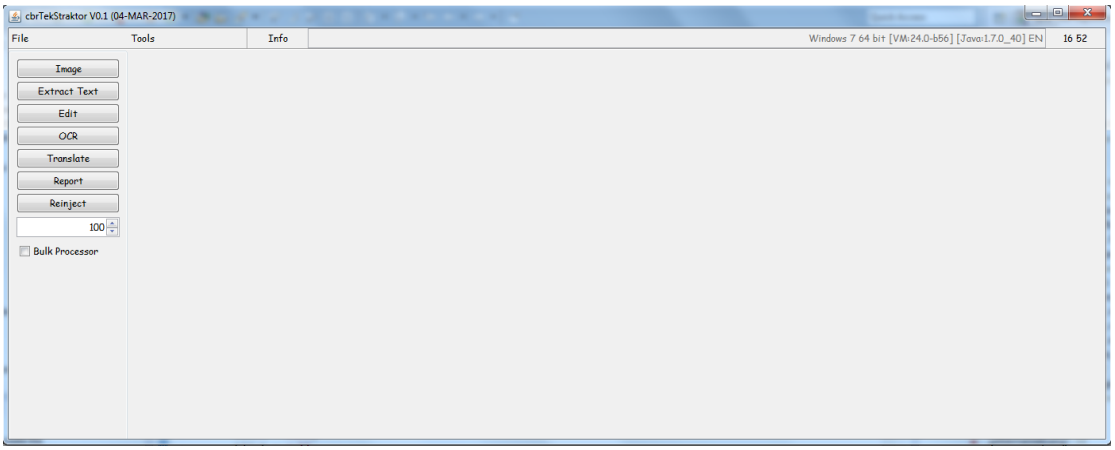
Main screen

Summary

This section provides step by step instructions on how to use the cbrTekStraktor application.

Starting the application

See previous section to learn how to start the application.



The above picture shows the main screen. The size and the location of the main screen are reused when the application is restarted.

The major functions of the application can be accessed via set of buttons located on the left of the application's main canvas.

Status information is displayed on the top right corner of the application's main window.

Image This button enables to select a scanned image of a comic book page (or any other image in JPG, GIF or PNG format) and display it on the canvas.

Extract Text This button enables to select a scanned image of a comic book page and extract the textual information on it.

Edit Pressing this button will open the graphical editor, which enables to examine the various graphical components of a single comic book page.
The editor also enables to manually select or deselect textor graphical components and to further edit or translate the extracted text.

OCR The Optical Character Recognition functionality is accessed via this button.

Translate The translation component is currently not implemented.

Report By pressing this button one opens the reporting component, which provides access to summarized graphical and statistical information of a single comic book page.

Re-inject This will enable to re-inject text into the previously identified speech balloons or other text areas. This option is currently under development.

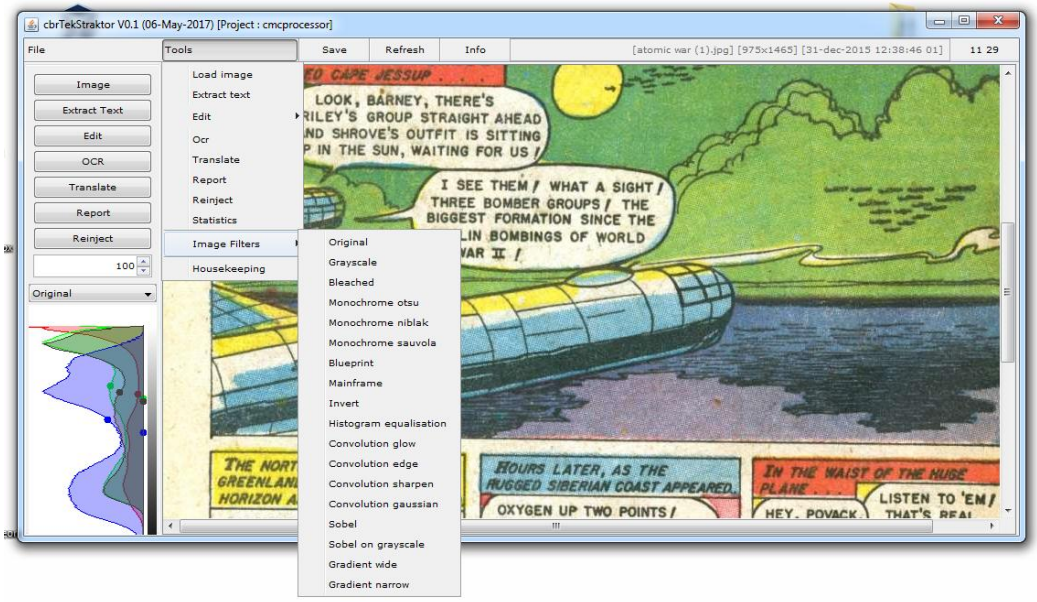
Other The "Bulk processor" checkbox will activate the bulk processing option. This enables to perform the text extraction on a set of comic book pages stored in a single folder. This option might therefore be used to extract and OCR the text of an entire comic book.

The "spinner" component enables to enlarge or shrink the image displayed on the screen. It should only be used in "Image" mode. In fact it is recommended no to use when in "Edit" mode.

Menubar

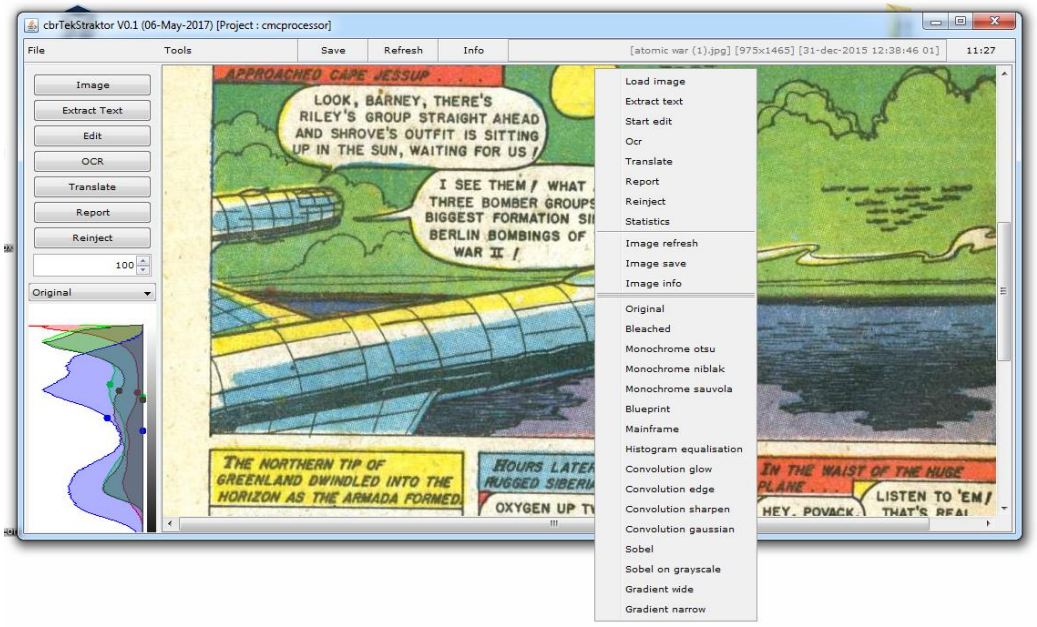
All of the above and additional functions can also be accessed via the menu bar items "File" and 'Tools".

See the picture below for a quick overview of the available menu items.



Pop-up menu

Right-clicking of the main canvas will open a pop-up menu, comprising similar menuitems as the ones on the menus discussed in the above section.



Additional major functionality

Additional functionality which can be accessed via the menu bar. There are 2 main menu items

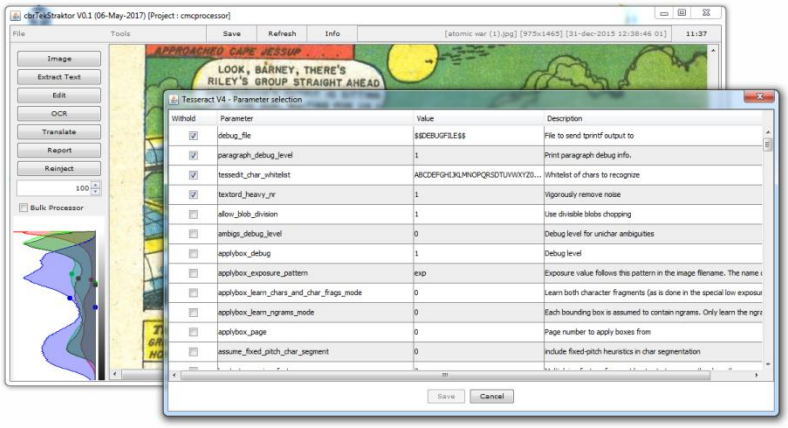
- Files
- Tools

Properties

The Files>Properties menu item provides access the "Edit option" and "Tesseract Option" Dialogs.

The "Edit option" dialog enables to customize the look and feel of the Comic Page Editor, e.g. the background drop. See section "Edit mode"

The Tesseract Option dialog screen enables to handpick one of the many Tesseract options and parameters and set it to an appropriate value.See section "OCR mode".



Statistics The Tools>Statistics function collects statistical information on the entire set scanned comic book files available in the $PROJECTDIR folder. See the "Developer Notes" section for more information on the folder structure of the application and to learn where to locate the $PROJECTDIR folder.

Housekeeping Tools>Housekeeping. This will prune (remove) temporary files from the cbrTekStraktorworkfolders.

Import Files>Import. The import function is currently not implemented.

Export Files>Export. The Export function creates a file of all extracted textual information within a project (see cbrTekStraktor Projects).

Chapter 4

How To :Image mode

Introduction

The following picture shows the cbrTekStraktor application when running in "Image" mode.

When you click on the "Image" button a file selection dialog will be presented, enabling to browse through your comic book (or other) image files.In this example a 18th century caricature "1024px-Caricature_gillray_plumpudding.jpg" image file is rendered.

Note: The 5 most recent selected images can quickly be re-accessed on the "File" menu.

Marquee

The following buttons are available on the marquee

- Save, this enables to save the image
- Refresh, this will reload the original image
- Info, this will open the "Image Info" dialog

Colour histogram

A color histogram is present in the bottom left corner. The histogram (or frequency diagram) shows the distribution of the Red, Green and Blue (RGB) color components of the pixels present in a picture (JPG,GIF, PNG) on a scale of 256. In which 0 is the most intense and 255 the lightest value.

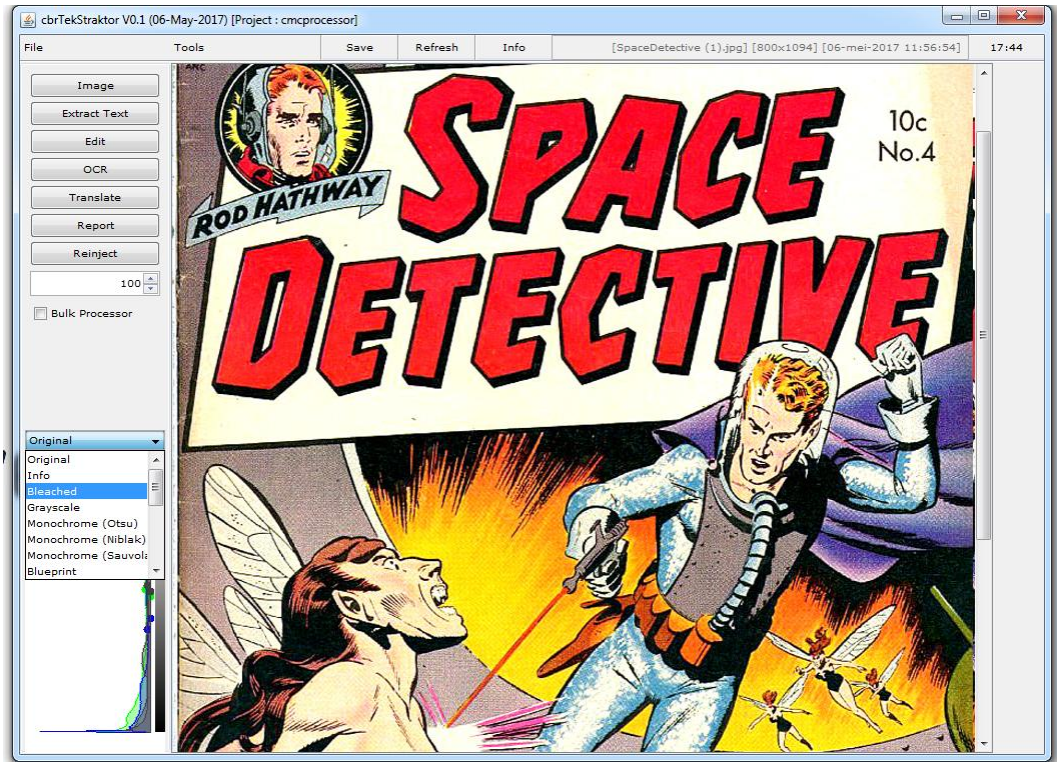
The circles separate from the vertical axis show the median of each RGB component.

The circles on the vertical axis show the means of the RGB component.

The histogram also shows the frequency distribution of the luminance or the "Alpha" channel.

Image filters

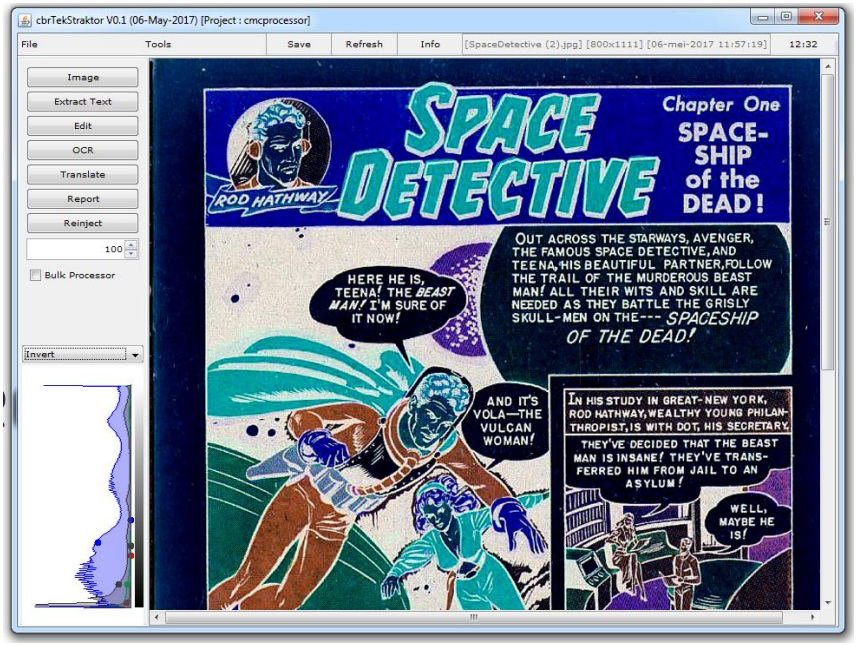
A choice of image filters is available on the drop-down list to further process the image.



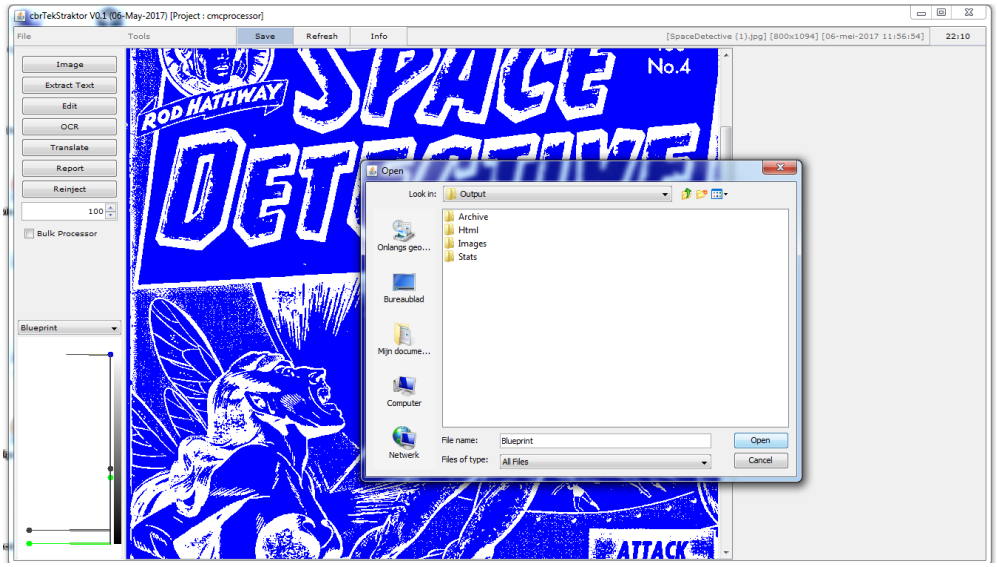
Filter	Description
Bleach	RGB to HSB conversion followed by lowering the hue component. (see the detailed info on HSL and HSV in the appendix to this reference manual).
Blueprint	Binarization (by default Niblak is used) and subsequent reduction to the Blue color component.
Convolution Blur	Blurs the image via convolution (explained in the appendix).
Convolution Edge	Applies the edge convolution filter
Convolution Gaussian	Applies a Gaussian blur filter.
Convolution Sharpen	Applies a sharpening convolution filter

Gradient narrow	Applies a narrow gradient transformation
Gradient wide	Applies a wide gradient transformation (explained in the appendix).
Grayscale	RGB to grayscale conversion using the formula describedat the end of this section.
Histogram Equalization	Histogram equalization image transformation (explained in the appendix).
Info	Opens a pop-up window displaying the Image properties
Invert	Inverts the colors on the RGB image
Mainframe	Binarization and subsequent reduction to the Green color component.
Monochrome (Niblak)	Binarization using the Niblak transformation (see appendix)
Monochrome (Otsu)	Monochromization or binarization using the Otsu transformation (explained in the appendix).
Monochrome (Sauvola)	Binarization using the Sauvola transformation (explained in the appendix).
Original	This option redisplays the original image.
Sobel	Applies a Sobel filter (explained in the appendix).
Sobel on grayscale	Applies a Sobel filter on the grayscaled image

The next picture shows the result of applying the “inverse” image filter.Apart from a bizarre aesthetical interest there is no practical usage known for inverting the color schema of an image.



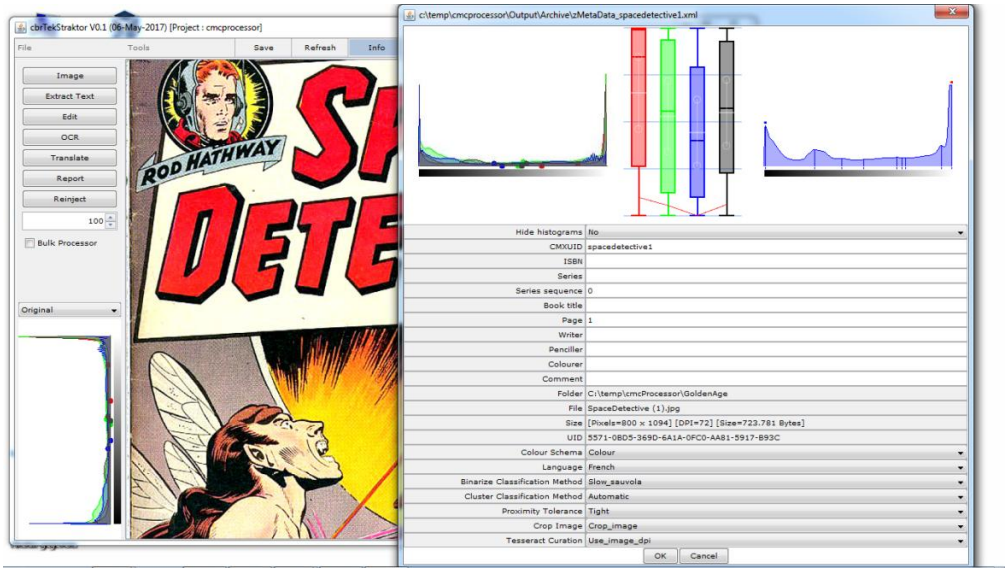
You can save the result of the image processing by pressing “Save” and providing a file name. The screenshot below shows the how the result of the “Blueprint” image filter are about to be saved to a PNG file.



Comic Page Info Screen

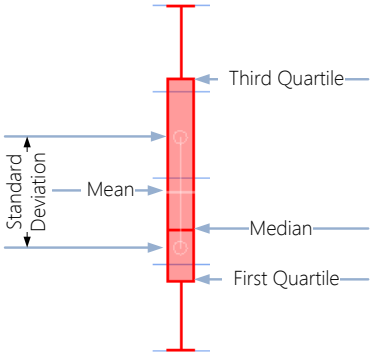
The Comic Page Info Screen provides access to a selection of characteristics of a Comic Page Image. One can either examine or specify various characteristics of a comic book and page.

The Comic Page Info Screen can be accessed via the marquee buttons, the menubar or the pop-up menu. Click on "Info" to open this dialog.

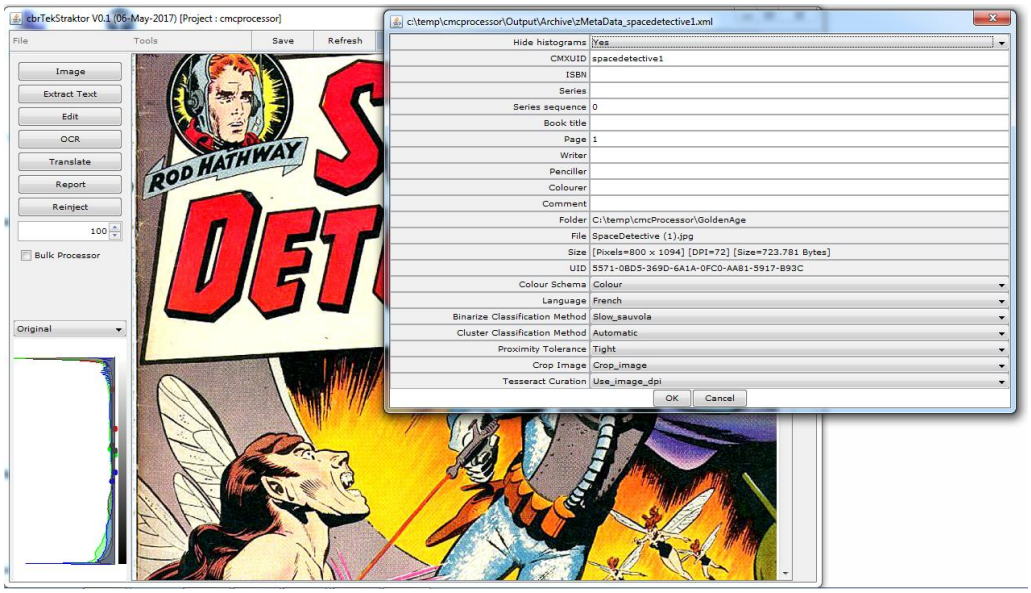


The top of the screen comprises

- The RGB Histogram of the image, is located on the left hand side.
- On the right is the histogram of the gray scale information. The peaks and valleys on this histogram are used to determine whether the picture is a monochrome image or not.
- The Box Plot of the RGB information. The Box Plot diagram shows the first and third quartiles, median and mean of the RGB and Grayscale values of the pixels.



The histograms on the top of the screen can be collapsed in order to reduce the clutter on your desktop by setting the "Hide histogram" option to "Yes".



[Wikipedia] In descriptive statistics, a box plot or boxplot is a convenient way of graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagram. Outliers may be plotted as individual points. Box plots are non-parametric: they display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution. The spacings between the different parts of the box indicate the degree of dispersion (spread) and skewness in the data, and show outliers. In addition to the points themselves, they allow one to visually estimate various L-estimators, notably the interquartile range, midhinge, range, mid-range, and trimean. Box plots can be drawn either horizontally or vertically.Box plots received their name from the box in the middle.A JAR (Java ARchive) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file for distribution. JAR files are archive files with which include a Java-specific manifest file. They are built on the ZIP format and typically have a .jar file extension.

The bottom of the Image Info Screen consists of

Label	Description
CMXUID	Comic Book Unique Identifier, which is a simplified normalization of the filename of the Comic Book Page picture. The normalization consists of removing all non-alphabeticalcharacters from the file's name.
UID	A unique identifier, comprised of a hexadecimal number of 32 characters.
ISBN	The International Standard Book Numberof the comic book. You will need to enter the ISBN number manually. On www.isbn.org most ISBNs can be found.
Series	Name of series of the comic book
Series sequence	The sequence number of a comic book in a comic book series
Book title	Comic book title
Page	The page numberof the image in the comic book
Penciller Colorer Writer	Information on the various authors who contributed to the creationof the comic book.
Comment	Can be used to provide additional comments
Folder	The folder where the comic book page image is stored
File	The name of the scan (or picture) of the comic book page
Size	The size of the picture in pixels and in Bytes, as well as Dots per Inch (DPI) information (if present in the image).
Color schema	cbrTekStraktor will determine whether the picture has a monochrome, grayscale or colorscheme. If needed you can overrule the color schema detected.
Language	The language used in the speech balloons or text areas. If the matching language pack has been installed, Tesseract will be

	instructed to use it.
Binarization technique	One can select the binarization technique to be used when processing the comic book page image. The options are {NIBLAK, SAUVOLA, OTSU, BLEACHED, ITERATIVE}.
Cluster classification method	This is the method used when determining which one of the connected components cluster comprises the textual information, a.k.a. the character cluster or text paragraph. By default the method is set to "automatic". If needed one can select "Cluster 1" through "Cluster 5" to override the automatically identified character cluster. See section "Text Extraction" for a detailed discussion of cluster classification.
Proximity tolerance	{WIDE, LENIENT, TIGHT, ULTRA_WIDE} The proximity level is used when adjacent characters are combined into words and paragraphs. The default value is "Tight".
Crop Image	By selecting this option one can remove the margins of an comic book page. Cropping the image reduces the size of an image and will therefore shorten albeit marginally the length of the text extraction process.
TesseractCuration	This option enables to apply additional image processing filters prior to performing the OCR step. In particular blur an image or increase the DPI (Dots Per Inch). Tesseract notoriouslyperforms best on 300+ DPI images.

The characteristics of a comic page are stored in the zMetadata_<CMXUID>.xml file, which is part of the Archive file. See section "Developer Notes" for a discussion of the contents of the cbrTekStraktor Archive file.

Troubleshooting

The cbrTekStraktor application functions optimal when you are using a monitor of substantial size and a display card supporting the higher resolution ranges. Images present in CBR files often have widths and heights of more than 1000 pixels , so you will also need a hefty computer to support cbrTekStraktor image processing activities.

Chapter 5

How To :Projects

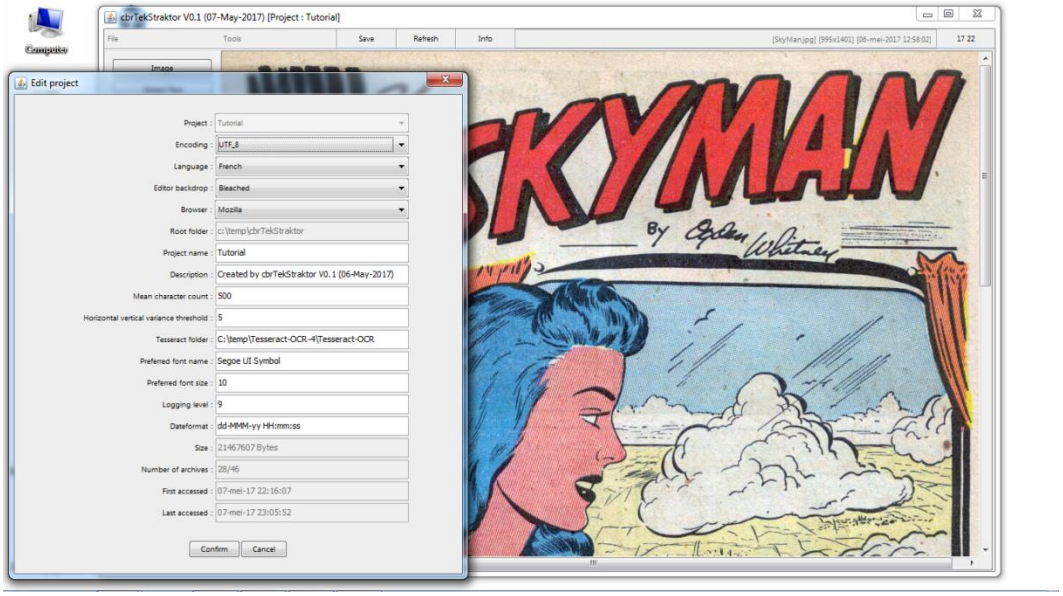
A project is an arbitrary grouping of comic book images. One could envisage to create a cbrTekStraktor project that comprises all images of a single comic book or to create a project of comic book images created by the same penciller.

A project is physically little more than the name of the folder on the Windows or Linux file system containing a predefined set of files and folders which are required by the cbrTekStraktor application.

See "Developer Notes" for detailed information on the folders and files that constitute a project.

Editing a project

The propertiesor settings of a project can be set and modified via the menu item"File>Project>Edit project"



When cbrTekStraktor is started for the first time the "Tutorial" project is automatically created with default configuration settings. It is recommended to change these settings to match your local environment and preferences.

Property	Description
Encoding	Options are { UTF-8, UTF-16, ISO-8859-01, ASCII} You can select the encoding of the files created by cbrTekStraktor. Default encoding is UTF-8.
Editor backdrop	There are loads of options { BLEACHED, BLUEPRINT, GRAYSCALE RASTERIZED, COLOUR RASTERIZED, NIBLAK , SAUVOLA, MAINFRAME, NONE, ORIGINAL } When in Edit mode the backdrop defines the backdrop, the picture which is displayed on the workarea. Just choose the backdrop which you like best.
Language	The drop-down list contains all the languages supported by Tesseract. The project language is the default language to be used for all image

	files of that project. The language can be overruled per page.
Browser	The drop-down list contains the supported HTML browser. These browser are used to render the reports. Options are { MOZILLA , EXPLORER, CHROME }
Project Name	The name of the project. The name of the project will stripped of non-alphanumeric characters and will be used as the name of the $PROJECTDIR folder's name.
Description	A short description of the project
Mean Character count	See developer notes
Horizontal vertical variance threshold	See developer notes
Tesseract folder	This is the name of the folder where the Tesseract binaries are stored.
Preferred Font name	The name of the preferred Font. The Preferred Font is used on all screens and dialogs. Be aware that the default Font is set to "Comic Sans Serif". Change ad lib.
Preferred Font Size	The preferred size of the font. Sizes between 10 and 12 work best.
Logging Level	The logging level can range between 0 and 9. Level 0 is terse logging, level 9 is very detailed logging. The logging and error information is displayed on stdout and stderr and redirected to the log and error files in the $PROJECTDIR folder.
Date format	The Java Date Format string. See docs.oracle.com on the options of the Java Date Time format
Size	This is the totalbyte size of all objects in the current project.
Number of archives	This is the number of all objects in the current project.

First accessed	Timestamp of the moment the current project was created.
Last accessed	Timestamp of the moment when the last time an object was created.

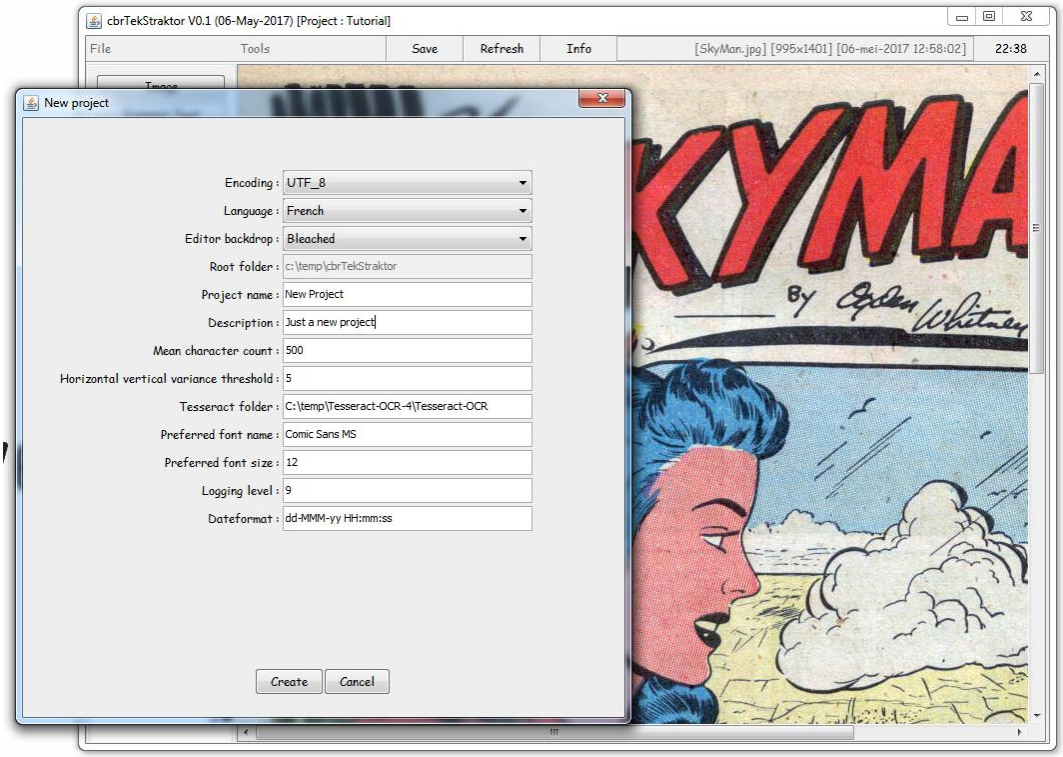
Prior to saving the Project properties a validation of the property values will be performed. This will for example prevent of specifying an in correct Java Date Format.

Creating a project

A new project can be created via "Files>Projects>New project"

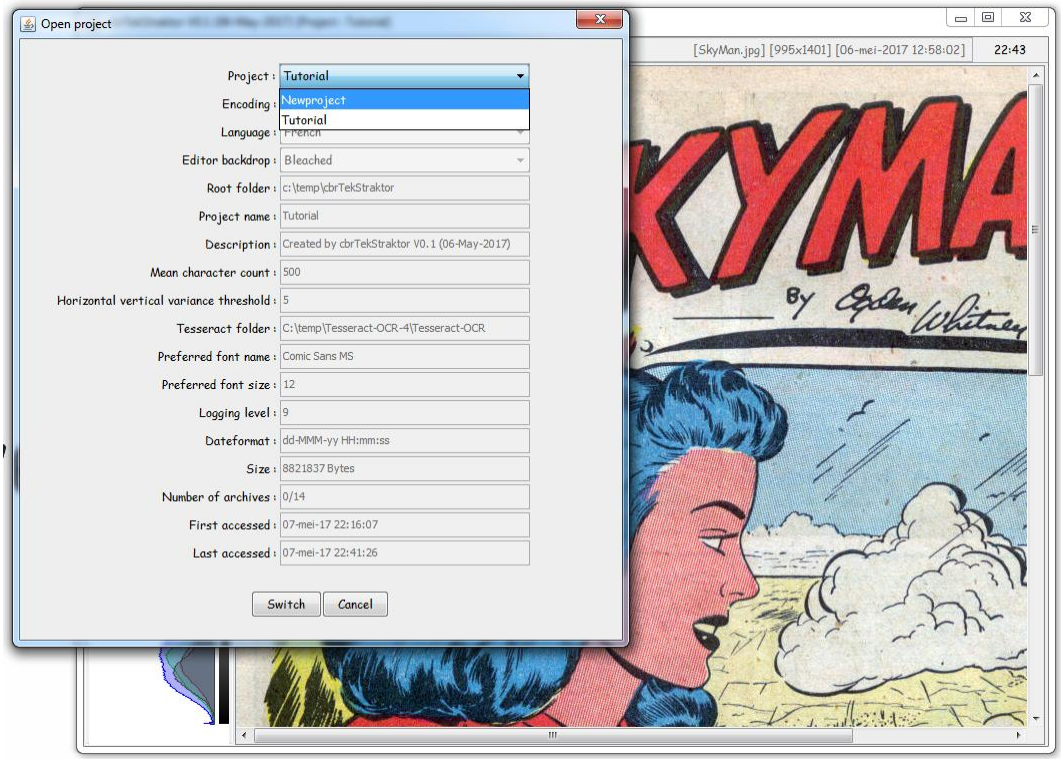
The configuration settings of the current project will be inherited by the newly created project.

When creating a new project it often suffices to merely provide the name and description of the new project.



Selecting a project

You can switch between project via the menu item "Files>Projects>OpenProject". An overview of the projects which are available will be presented on the topmost drop-down list. Select the project you want to switch to and then click on the "Switch" button.



Project properties file and current project

The current project will be reused when the application is restarted. The current project is saved in the Project Properties Files.

The Project Properties file is located one folder up from the current project. When using windows this will more than likely be in C:\temp\cbrTekStraktor. When using Linux the Project Properties files will be stored in $HOME/cbrTekStraktor.

In both cases the file is named "cbrTekStraktorProjectConfig.txt". See the "Developer Notes" section for more information on the contents of this file.

Defining the project via the command line

You can set the project to be used via the command line option –D.

For example: java –jar cbrTekStraktor.jar –D C:\temp\myComicProject

Chapter 6

HowTo :Text extraction

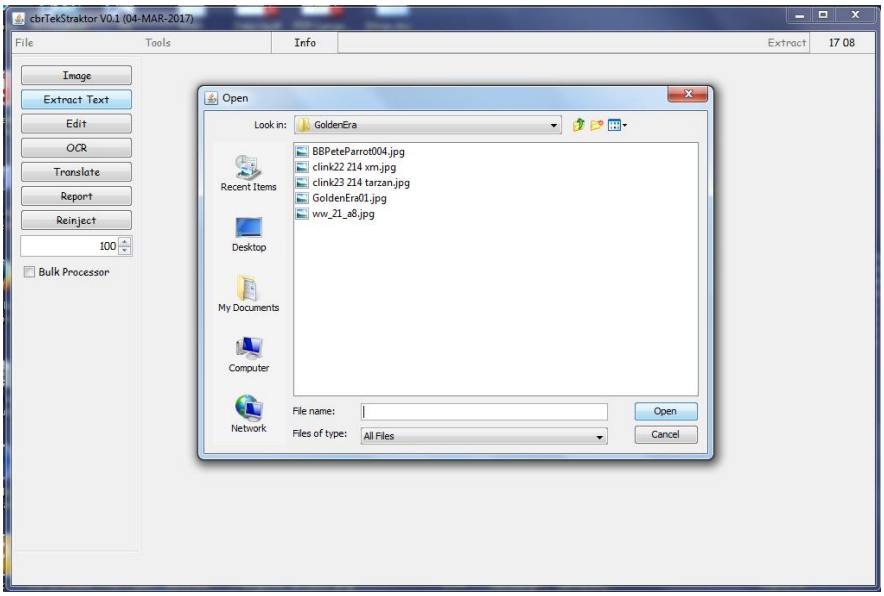
The text extraction process is started by pressing the "Extract" button on the main screen.

Alternatively when in "Image mode" you can opt to start the text extraction process on the current image by double clicking on it or by pressing the "Extract" button.

The text extraction process runs through the following four steps. The extraction process can be interrupted by pressing the "Stop" button on the marquee.

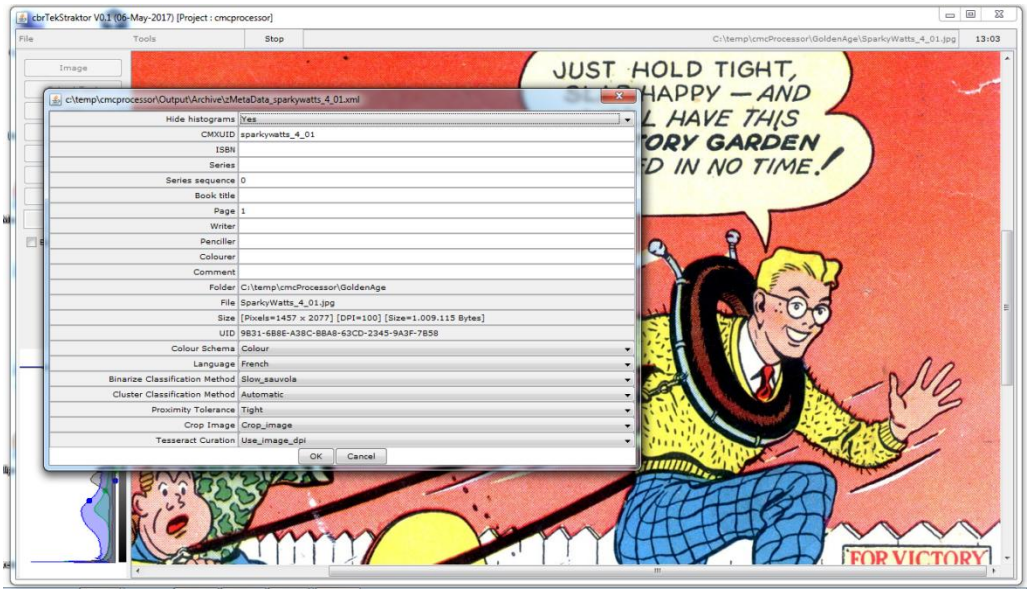
First step

You will be prompted to select an image for which the text is to be extracted. The images need not be stored in the cbrTekStraktor $PROJECTDIR.



Second step

The image which was chosen in the previous step is displayed and the Comic Book Metadata dialog opens.



In most cases itsuffices to just select the language of the comic book's text on this dialog.It is important to correctly define the language, because later on it is used by the Tesseract OCR engine.

Additional fine-tuning of the extraction process

The image which was chosen in the previous step is displayed and the Comic Book Metadata dialog opens.

Additional fine-tuning of the extraction process

Option	Comment
Color schema	In the event that the color scheme was not correctly identified, one can set its correct value either to Color, Grayscale or Monochrome. You should ensure that the correct colorscheme is set prior to starting the next phase of the text extraction. The accuracy of the color schema detection algorithm is monitored for future enhancements. The information is stored in the zMetadata XML file and is used to track the correctness of

	the color scheme detection logic.
Binarize Method	Options are {FAST_BLEACHED,ITERATIVE,SLOW_NIBLAK,SLOW_SAUVOLA,OTSU} It is advised to use the default binarization method (Sauvola or Niblak). OTSU and Bleached are valid options too. It is not recommended to use the "Iterative" option any longer.
Cluster Classification Method	Options are {AUTOMATIC, CLUSTER1,CLUSTER2,CLUSTER3,CLUSTER4,CLUSTER5} It is advised to use the "Automatic" option. A key element of the text extraction process is the module that decides which of the clusters that have been created using the K-Means algorithms, contains the characters. The idea is to create clusters of similar components and subsequently classify these clusters in • A cluster containing frames and borders • A cluster containing groups of characters, i.e. paragraphs. • Clusters containing noise In some cases Cluster Classification module picks the wrong cluster for the Paragraph cluster. This happens when the characters on the comic page are rather large. The symptoms of such a misclassification can easily be spotted in "Edit" mode: most of the characters will tagged to be "noise" and smaller pictorial elements resembling characters will be tagged to be "Letter". (*"Letter" is the tag used for components which are deemed to be characters. It is a misnomer, typical for Dutch native speakers.*) In the event the Automatic Cluster Classification designates the wrong cluster to comprise text, you can override it by arbitrarily setting it to Clusters 1 through 5. Start by setting the Text Paragraph to be Cluster2.
Proximity Tolerance	Options are {TIGHT,LENIENT,WIDE, ULTRA_WIDE} The proximity tolerance is used to group characters into words

	and paragraphs. In essence this is achieved by clustering graphical components based on the distance between the components. The default Proximity Tolerance is "Tight". Tight assumes the inter-character space to be rather narrow. Adapt to Lenient or Wide if deemed appropriate.
Crop Image	Options are {YES,NO} When Crop is set to Yes, the margins on a comic book page will be detected and removed in order to reduce the size of the image. The text extraction process runs faster when images are cropped to their actual payload.
TesseractCuration	Options are {IGNORE_DPI,USE_IMAGE_DPI,INCREASE_AND_CONVOLVE,INCREASE_AND_FOURRIER} Tesseract reads the DPI information from the Image File metadata. Tesseract works best on 200+ DPI images. USE_IMAGE_DPI will ensure to set the DPI on the image submitted to Tesseract. INCREASE_DPI will attempt to increase the DPI of an image to approximately 300 DPI. INCREASE_AND_CONVOLVE and INCREASE_AND_FOURRIER will increase the DPI and will apply some blurring to the image prior to being processed by Tesseract. INCREASE_AND_FOURRIER is not implemented in cbrTekStraktor V01.

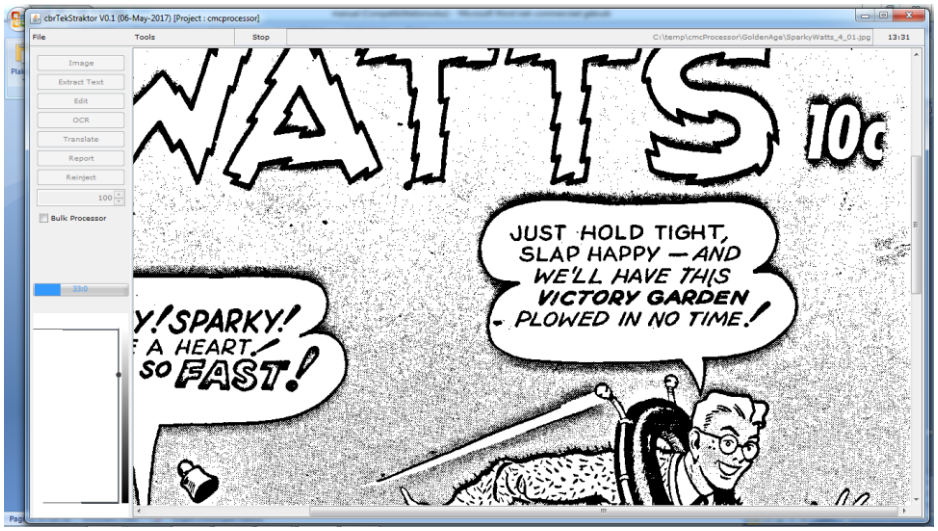
The actual text extraction process will commence when pressing OK on the Image Info dialog box.

Third step

The original image is cropped, turned into a grayscale image and then binarized. These intermediary images are displayed as the text extraction progresses. No user actions are required during this phase.

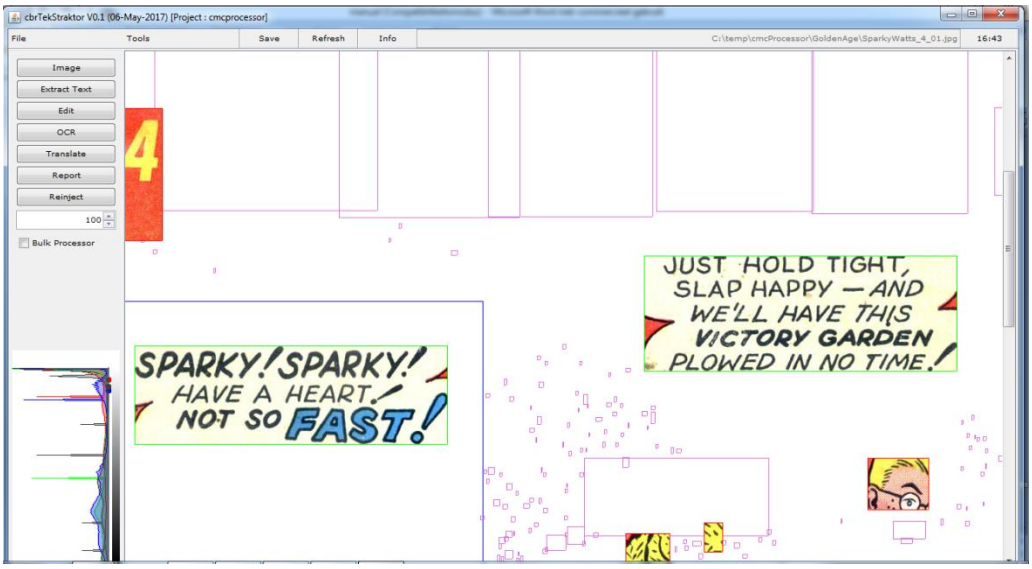
Example grayscale image.

Example binarized(monochrome) image. This is an important step in the process. Its purpose is to get crisp characters that distinctively stand out. By default the Sauvola or Niblak binarization method is used.



Concluding Step

The extraction process ends by displaying cut-outs of the text paragraphs which have been identified.



Character Paragraphs have a green border and non-character paragraphs have a red border. Frames have a lilac border.

The text extraction process stores the results in an "Archive" file in the $ROOTDIR/Output/Archive folder. An "Archive" file is a ZIP file that holds a broad variety of results, e.g. statistical data, image information, etc. See the "Developer Notes" section for detailed information.

Marquee

The following actions can be performed via the buttons on the marquee.

- The Save Button enables to save this resulting picture
- The Refresh button will redisplay the picture
- The Info button will open the Comic Book metadata dialog

The "Edit mode" can be activated by double clicking on the canvas or by pressing the "Edit" button.

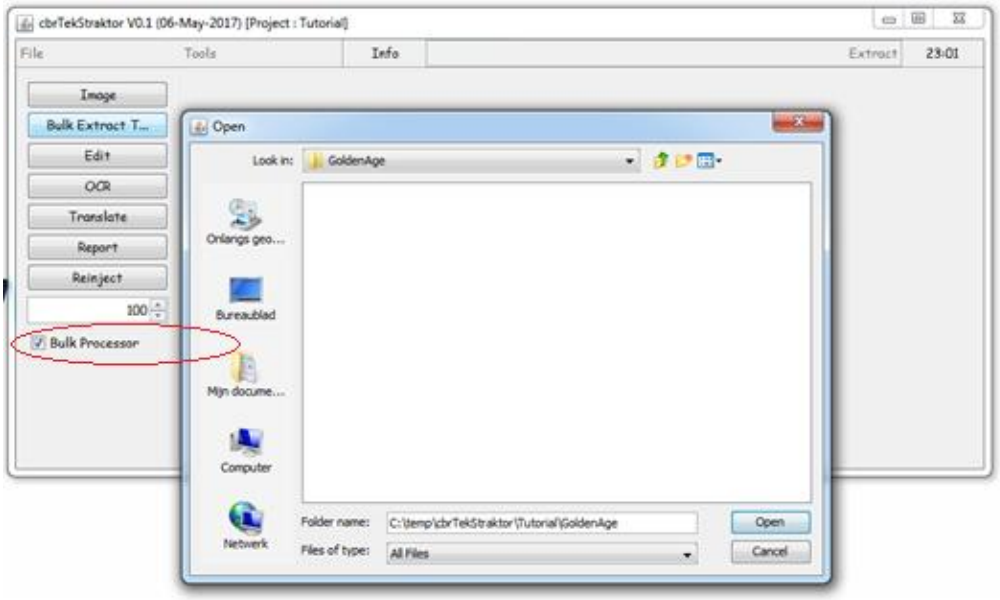
Bulk mode

The text extraction process can run on entire sets of comic book pages. The bulk extraction will process all images within a single folder.

The Bulk extraction process is similar to the single page text extraction process. It can be interrupted by clicking on the "Stop" button on the marquee.

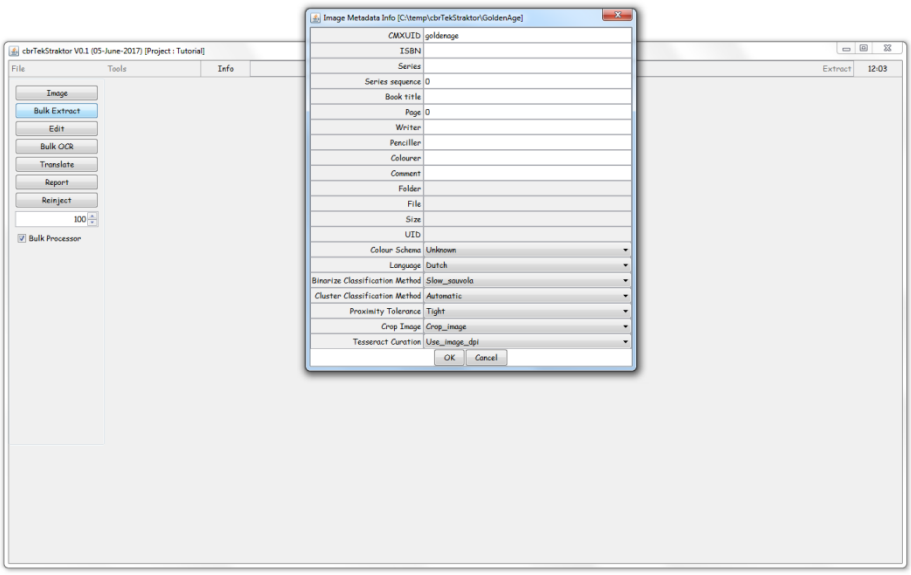
Bulk mode initial step : folder selection

You need to set the "Bulk extraction" option on the main screen before The caption on the "Extract" button will change to "Bulk". When you click on this button you will be prompted to select the folder containing the set of images to be processed.



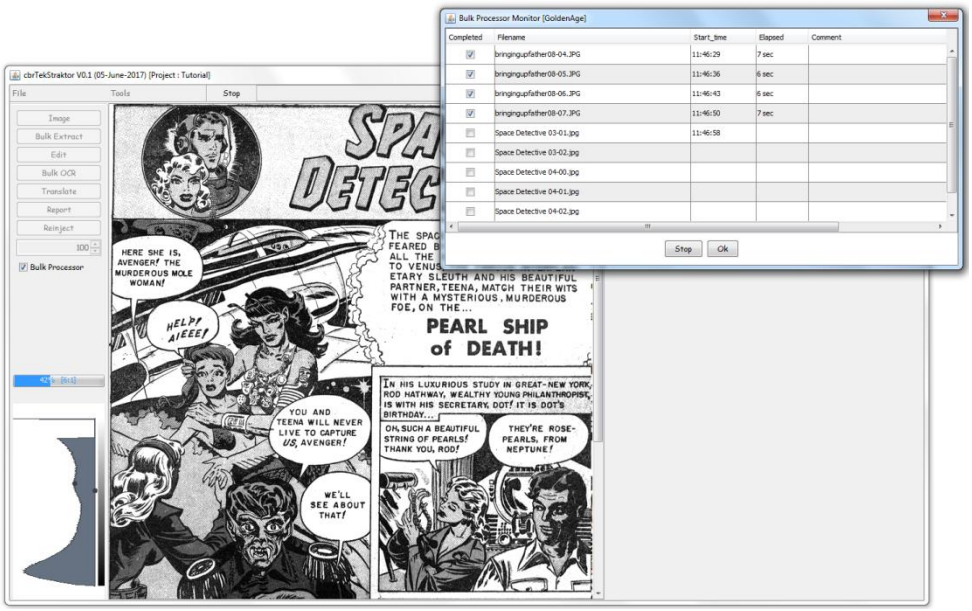
Bulk mode second step : Comic Page Info

The Comic page Info screen will then appear. The settings that you define on this screen will be applied to all images present in the bulk extraction folder.it is therefore recommended to only run the bulk extraction process on images sharing the same characteristics, e.g. pages from the same comic book, images with are all monochrome, pages etc.



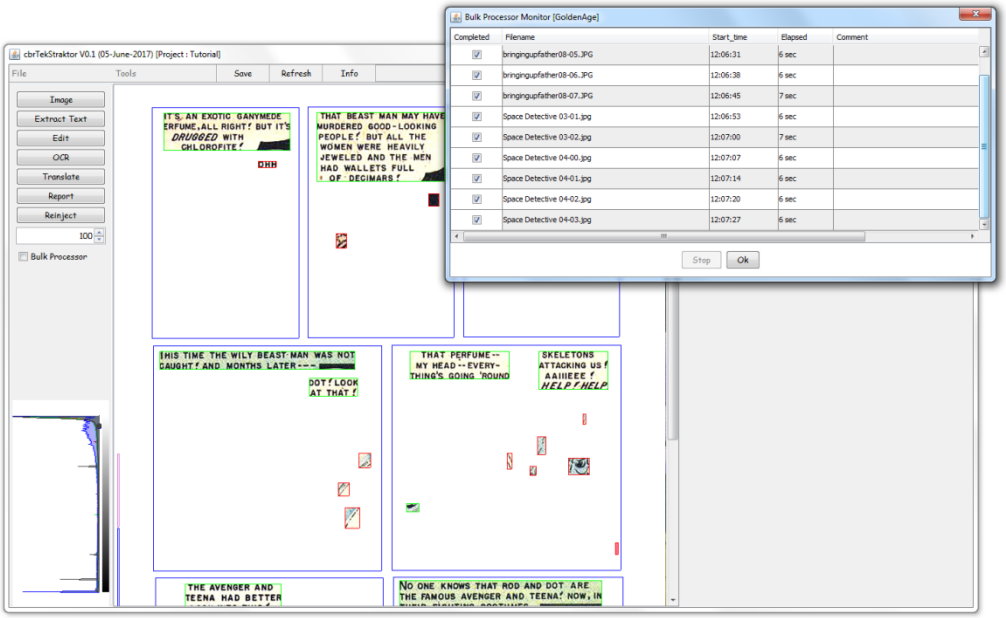
Bulk mode : Progress monitor

The text extraction process will be performed on each image file in the folder selected. The progress of the text extraction can be observed on the monitoring screen.



Bulk mode : concluding step

The extraction process will stop by displaying the cut-out pictorial elements of the last image in the folder. The monitor dialog will close automatically after a short period of time.



Troubleshooting

The cbrTekStraktor application functions optimal when you are using a monitor of substantial size and a display card supporting the higher resolution ranges.

&&TODO – comments on
Proximity
Cluster Classification
Comments on the scanned images

Chapter 7

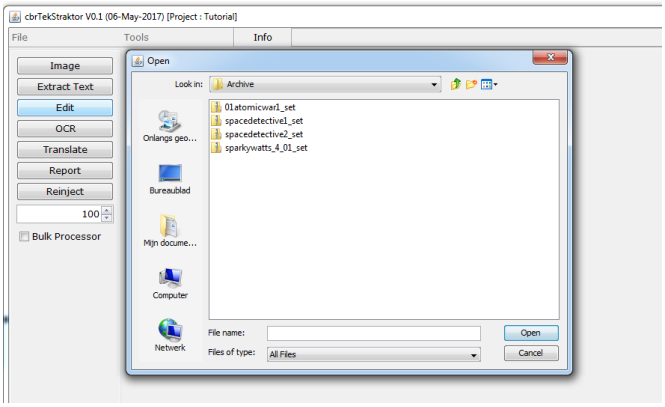
How To:Edit mode

This section describes actions which can be performed when in Edit Mode. The edit mode provide a GUI enabling

- to modify the results of the text classification process: remove paragraphs, define character paragraphs, define non-character paragraphs, etc.
- to manually enter the text within a speech balloon
- to translate the text within a speech balloon

Opening an archive for editing

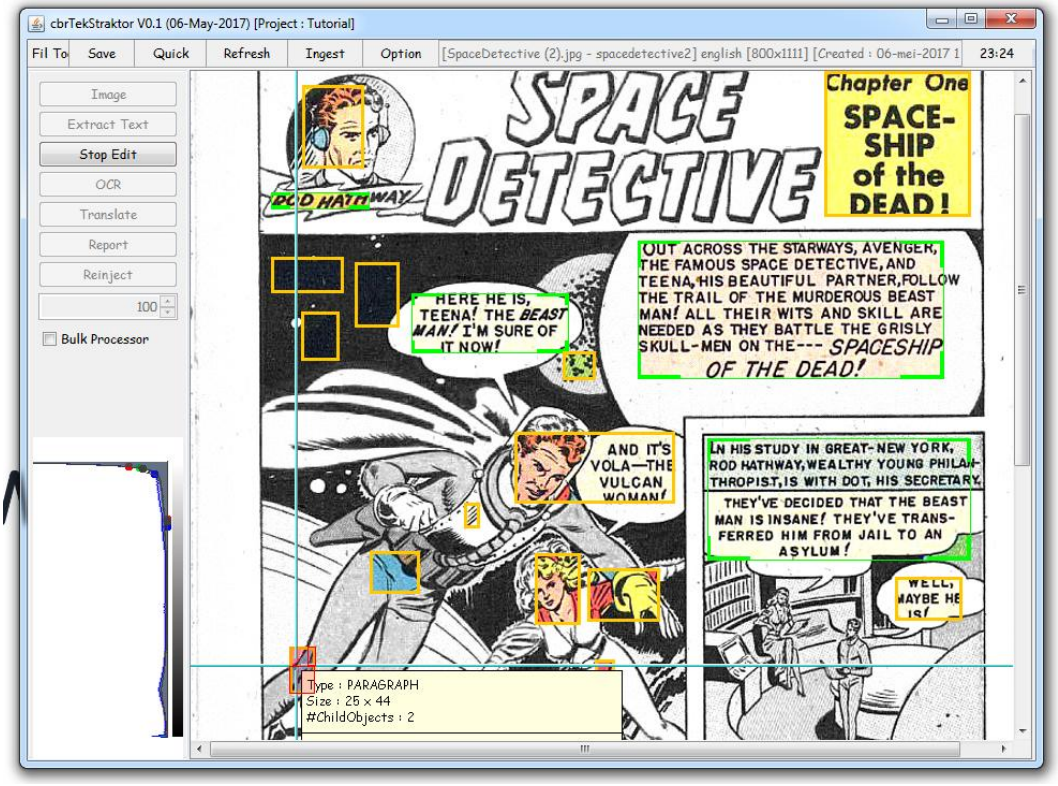
Click on "Edit" to start editing a previously processed Comic Page. A file browse dialog will open enabling you to select an Archive file. Archive Files are located in $PROJECTDIR/Output/Archive and have a name ending on "_set.zip".



Alternatively you can double click on the canvas once the text extraction process has been completed on a Comic Page image. You will be asked whether you want to start editing the current comic book image.

Editing

After choosing the "spacedetective2_set" archive on the previous dialog the below screen will open.



Marquee

The following buttons are active on the marquee

- Save
- Quick
- Refresh
- Ingest
- Option

These functions are commented upon in the remainder of this section.

The backdrop of the edit screen is the Comic Image file upon which an image processing filter has been applied. It is recommended to select a backdrop that nicely contrasts with the original comic page image. In the example above the "Black Bleached" filter has been applied.

Character paragraphs have a green border

Non-character paragraphs have an amber border

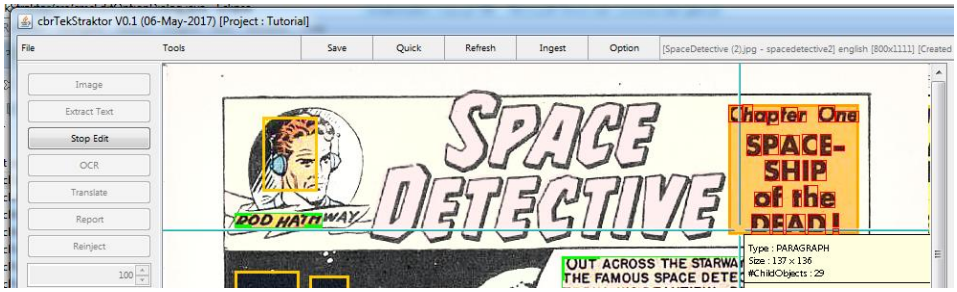
There is a crosshair mouse pointer. Crosshair pointers are tacky, but have the advantage to be able to precisely select an image element.

In the example above you will see that some character paragraphs have erroneously been classified to be non-character paragraphs, e.g. "Chapter One Spaceship of the dead" has an orange border, whereas this is a character paragraph and therefore should have a green border. In the next section you will be shown how to fix this.

Hovering

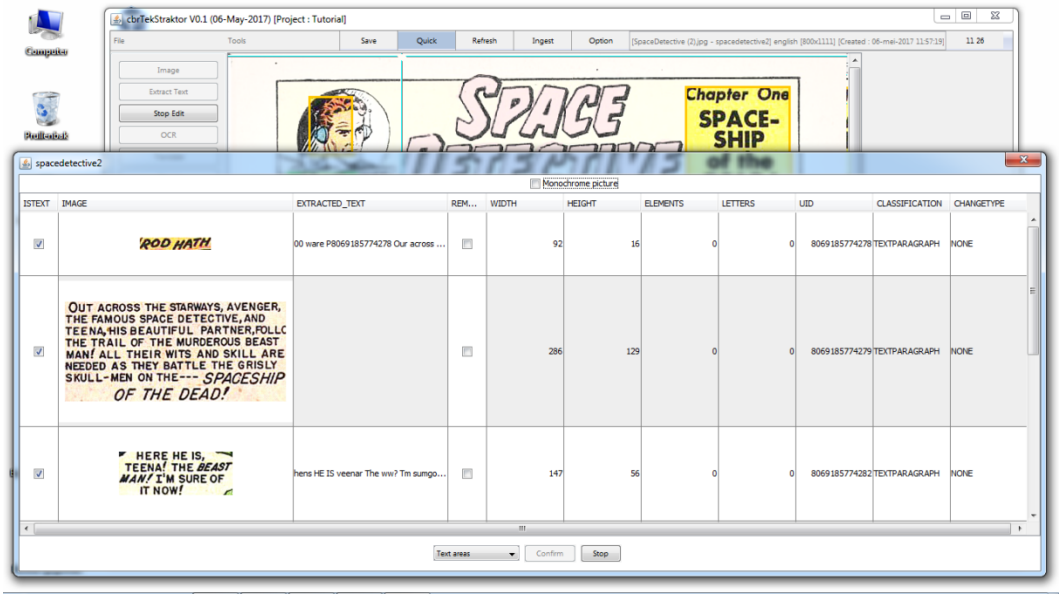
When hovering over a pictorial element an information box will be shown for a couple of seconds, providing succinct information on that element.In the example above a non-character paragraph, size 25x44 and featuring 2 Child objects; is in the crosshairs.

When you move the crosshairs over an element enclosed by a border, its background will momentarily adapt a rosy sheenand its constituting elements will be outlined in red. In the following example the characters which are part of the "Chapter One" text balloon are displayed.



Quick edit

The quick edit screen can be accessed by clicking on the "Quick" button on the marquee.



The quick edit screen puts detailed information on character paragraphs, non-character paragraphs, frames, noise and other types of component at your fingertips.

The tick box in the first column you can define whether or not a paragraph contains text.

The tick box on the "removed" column enables to remove (or delete) a paragraph.

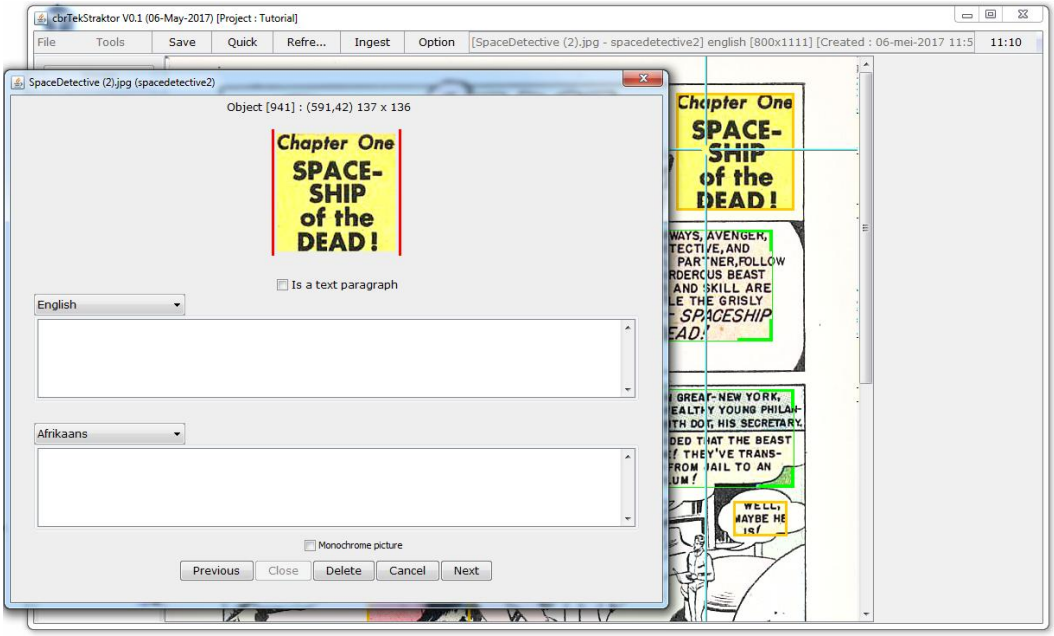
The "Extracted text" column can be used to enter or edit the textual information on the speech balloons. In the event that the image has been OCR'ed, it will contain the automatically extracted textualinformation.

On the drop down list you can select which type of pictorial information you want to see displayed, e.g. noise, frames, potential text, etc.

If you changed to characteristics of a pictorial element the "Confirm" button will become active.

Detailed edit – Ingest

The detailed edit dialog can be accessed by clicking on "Ingest" button on the marquee or by double clicking on a paragraph.



The dialog enables to navigate through the various paragraphs. Use the Previous and Next Buttons.

The image of a character paragraphs is displayed between thick green vertical bars. Non-character paragraphs have red borders.

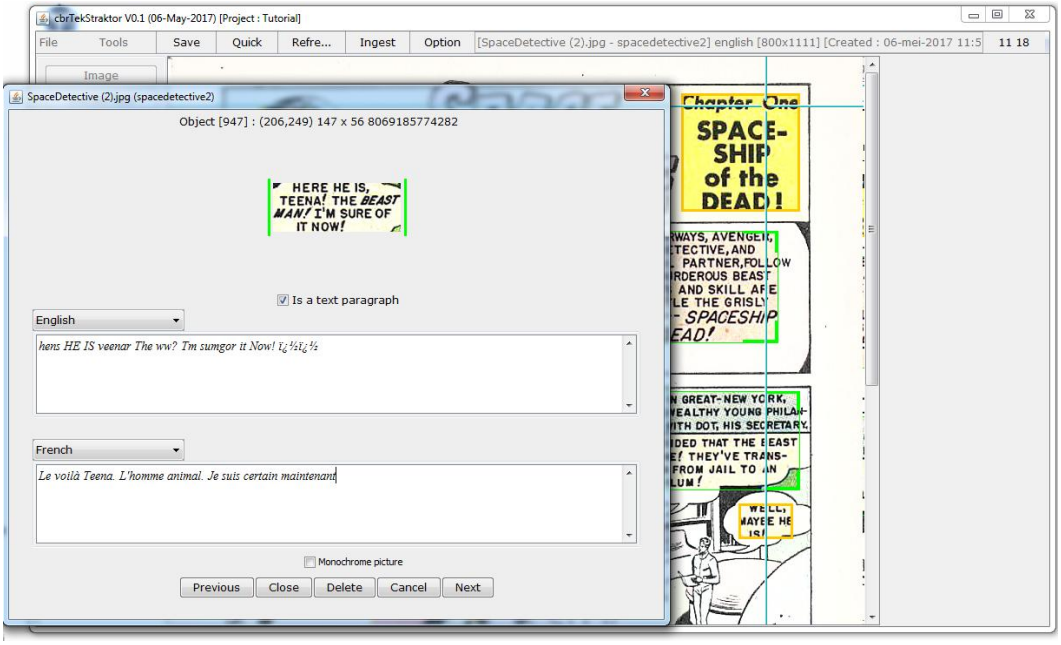
You can define whether a paragraphs contains text or not via the "Is a text paragraph" tick box.

The paragraph can be deleted by pressing the "Delete" button.

The monochrome tick box can used to display a monochrome version of the paragraph image.

Keying in text

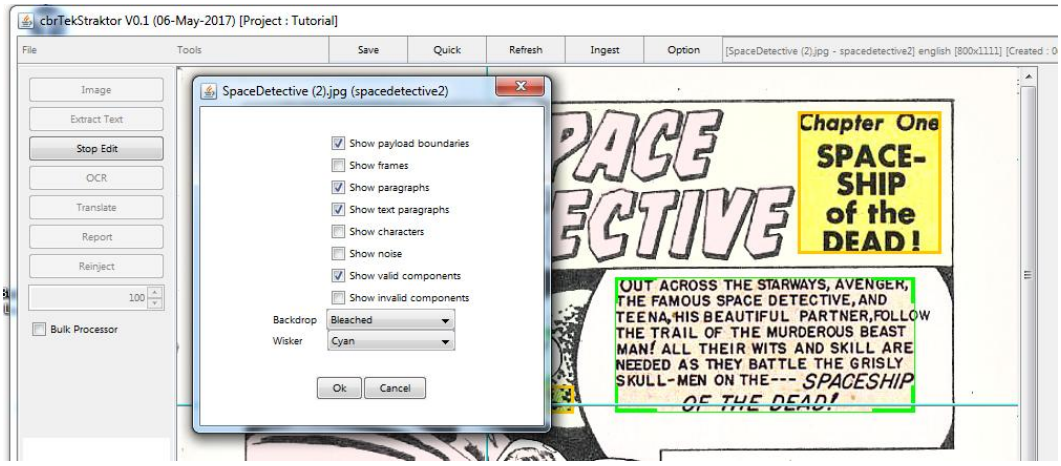
The topmost text entry box is used to edit the original text. The bottom text box can be used to store translated text.



In the event that the OCR process has been performed, the OCR'ed text will be displayed in the topmost text entry box. See the example above.

Edit options

The edit option dialog opens when you click on Edit Options



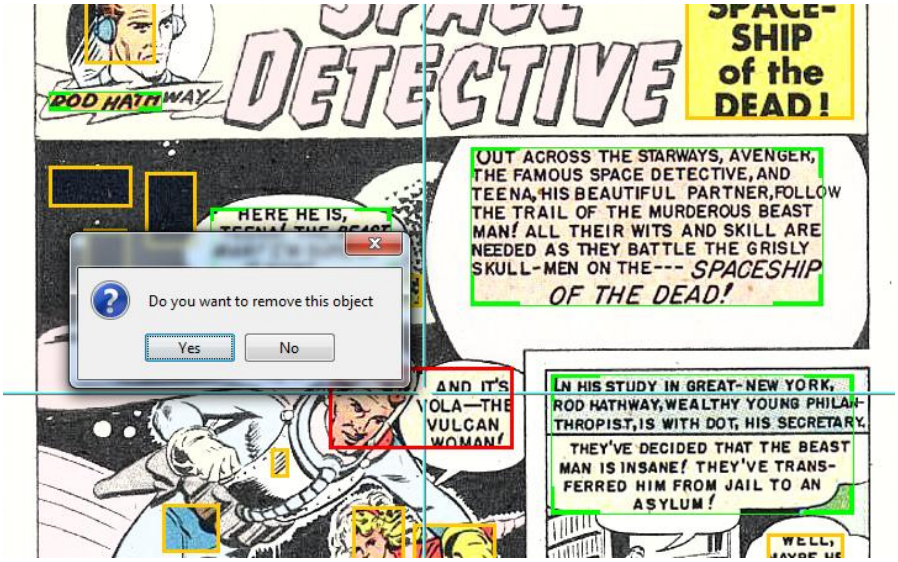
The edit option dialog is used to change the appearance of the Edit canvas

Option	Description

Show payload boundaries	This will set out the boundaries of the comic page margins in lilac.
Show frames	This will put bluish lines around the frames within a comic book page.
Show paragraphs	This will set out the non-character paragraphs in red.
Show text paragraphs	This will draw a green border around character paragraphs, e.g. speech balloons.
Show characters	This will put a pink border on the image components which have been identified to be characters.
Show noise	This will put pink borders around any image component.
Show valid components	This will show the components which are valid.
Show invalid components	Puts border on those image components which are invalid. See "developer notes".
Backdrop	This drop-down list enables to set the type of backdrop you want on see displayed in Edit mode.
Wisker	This drop-down list enables to define the color of the crosshair pointer.

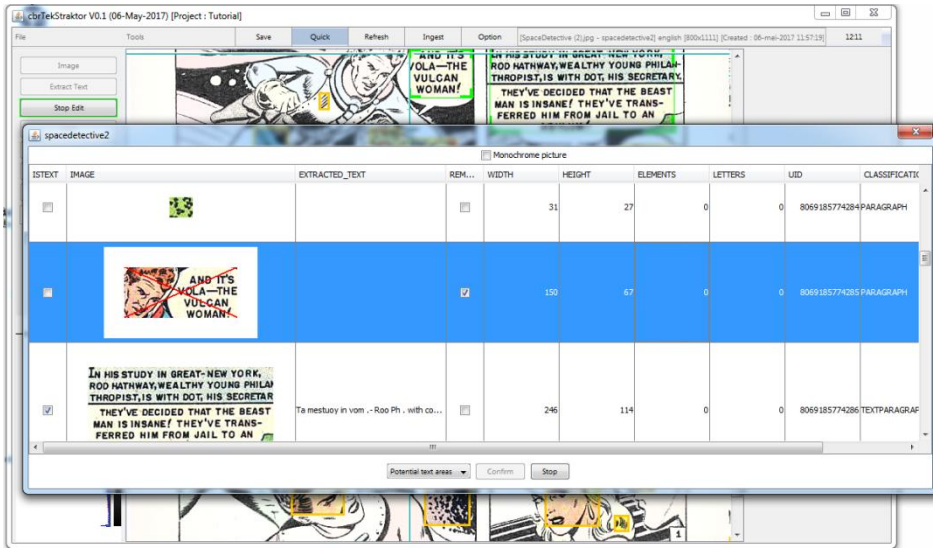
How to delete a paragraph

If you select a paragraph and left-click on it for more than 2 seconds, a thick red border will be put around the paragraph and you will be asked whether you want the paragraphs to be removed (deleted).



In the example above the object that comprises the face of the Space Detective Hero and snippet of text are combined into a single object. A possible manner to correct is to remove the object and create a new object that only contains the text.

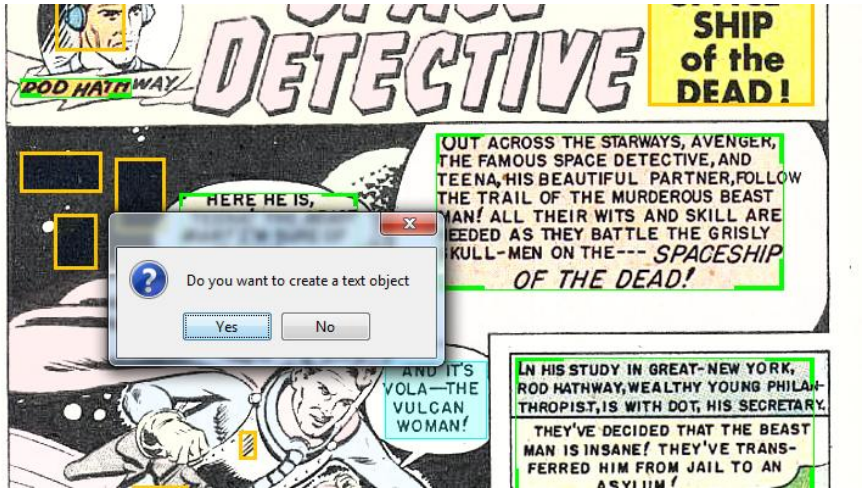
The objects which have been removed can be seen in the Quick Edit dialog by selecting "Potential Text Area" and trawling for images which have been crossed-out by a thick red line.



How to create a new text paragraph

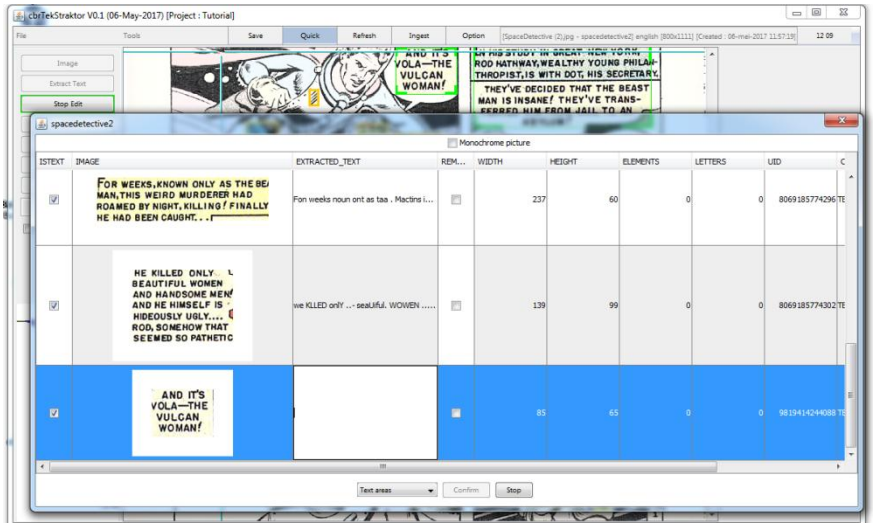
A new paragraph can be created by positioning the pointer on the top-left corner of the object to be created and dragging the cursor to the bottom-right corner of the object to be created.

Whilst dragging the pointer, a light-blue rectangle will be displayed. When the dragging operation is completed, you will be prompted to confirm the creation of a new object.



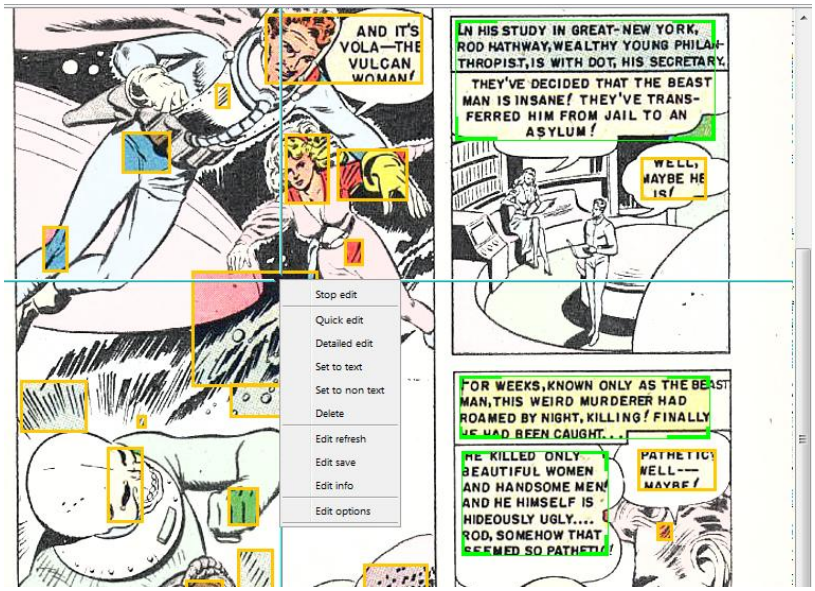
It is recommended to refresh the screen to reflect the changes made (by pressing the "Refresh" button on the marquee).

The freshly create object should now be visible in both the Quick Edit and Detailed Edit dialog screens.



How To quickly delete or change the characteristics of a paragraph

A pop-menu will open when you position the crosshairs over a paragraph and right-click on it.



The pop-menu permits to

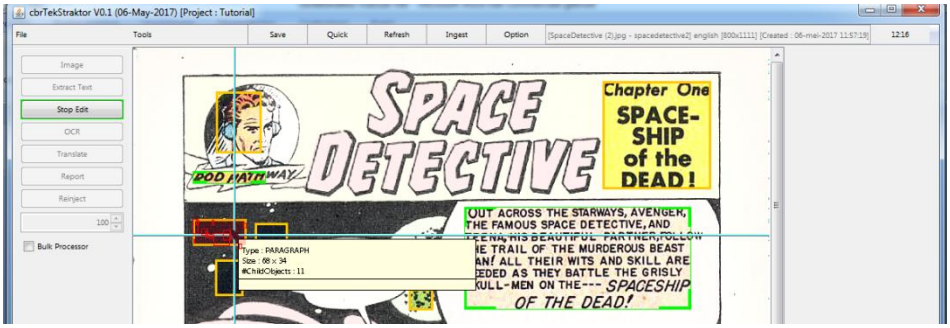
- Delete the paragraph
- Toggle between character and non-character

Pop-up menu

The functionality described in this section can also be accessed via the pop-menu which is opened by right-clicking anywhere on the canvas.

Saving changes

In the event that changes have been made to any of the components of the comic book page a greenish hue can be observed around the "Stop Edit" button. When you click on this button you will be asked to confirm the changes.



Note. Changes to the image components will be stored in the _stat.xml file and changes to the text information will be stored in the _language.xml file.
These XML files are part of the Archive file. The previous version of these XML files will be timestamped and maintained in the Archive file. This enables you to roll-back any of the changes made.

Chapter 8

HowTo:OCR

This section describes the Optical Character Recognition process

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, Version 2.0 and development has been sponsored by Google since 2006.

Prerequisite

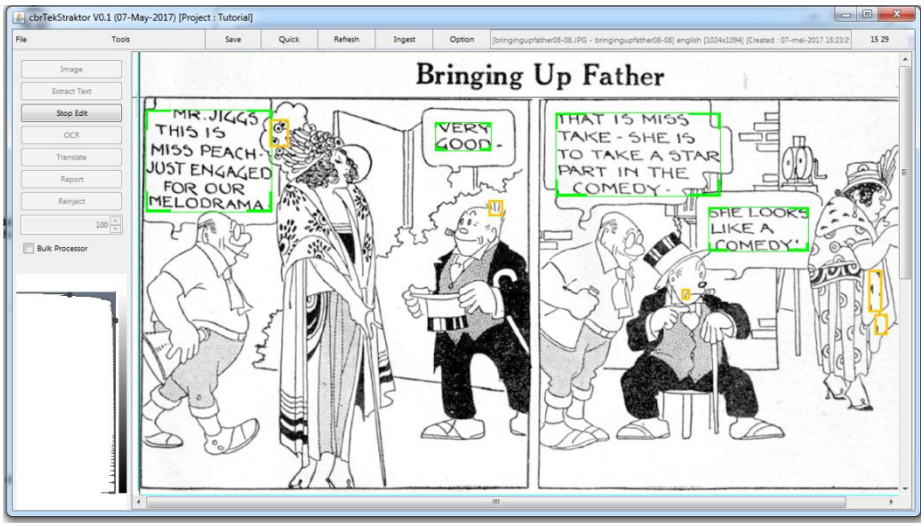
Tesseract is required to be installed prior to be able to perform OCR. You can also opt to install additional language packs.

You need to set the name of the folder in which the Tesseract binaries are stored via the Project Configuration dialog.

The Text Extraction process must have been completed on a comic page image before you can perform OCR. The OCR process uses the Archive file as its prime input. If you are not satisfied with the result of the Text Extraction process you can manually change the contents of the Archive file via the "Edit" option.

Example

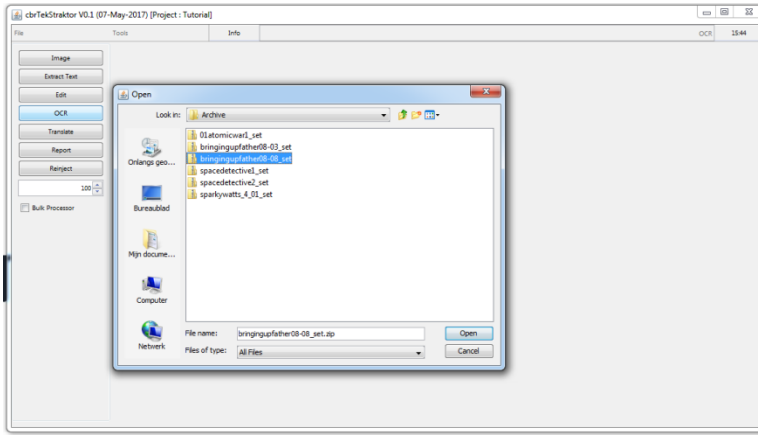
The next screenshot shows the Comic Page which is used as an example to perform OCR upon.



Starting the OCR process

First step

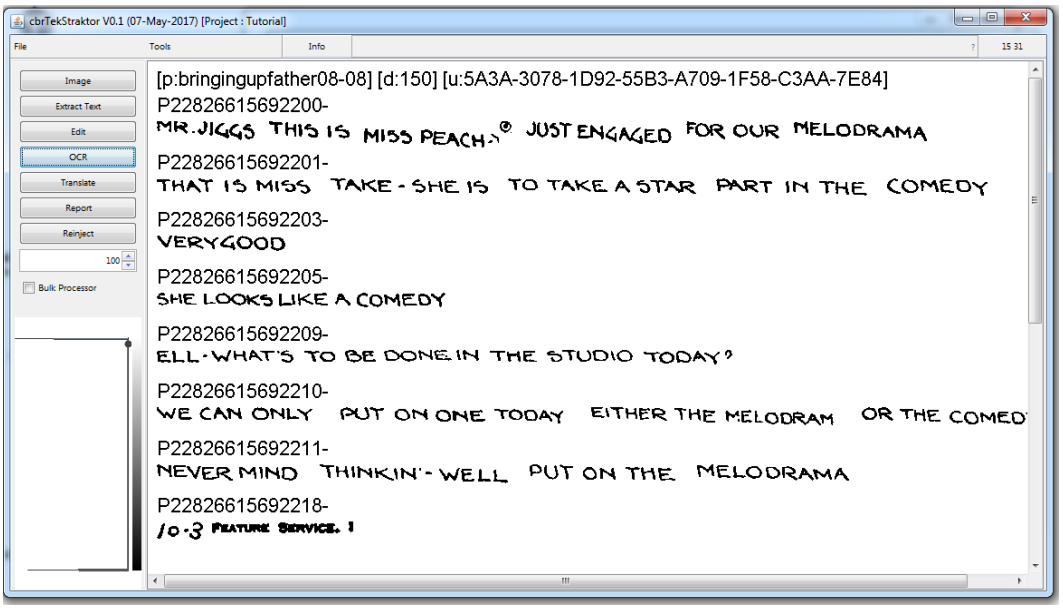
The OCR process is started by clicking on the OCR button and selecting the Archive file of the comic page that you want to OCR.



Second Step

The text in the speech balloons will be sourced from the results of the text extraction and edit processes. The text within a paragraph will be extracted from the original image and it will be flattened and put onto a single line. Each paragraph will be preceded by a Unique Identifier (UID).

The results will be displayed and saved in an Image file (OCRoutput.png).



The header line of the Tesseract OCR Image has a reference to the Comic Page Image file and the Comic Page 32 Hex Character UID.

In order to enhance the Tesseract OCT process, the resolution of the image file might be increased to 300 DPI and might be slightly blurred (using a convolution or Fourier transformation). See "TesseractCuration" option on the Comic Page Info dialog.

Third step

The Tesseract options set via the "Tesseract Option" dialog are fetched and stored in a parameter file (TesseractOptionRepository.xml which is located in the $PROJETDIR/OCR folder).

The Tesseract OCR client is called via its command line interface using the recently created OCR image file and the Tesseract parameter file.

The result of the Tesseract OCR process are stored in the OCR Result File (OCRResult.txt) which is located in the $PROJECTDIR/OCR folder.

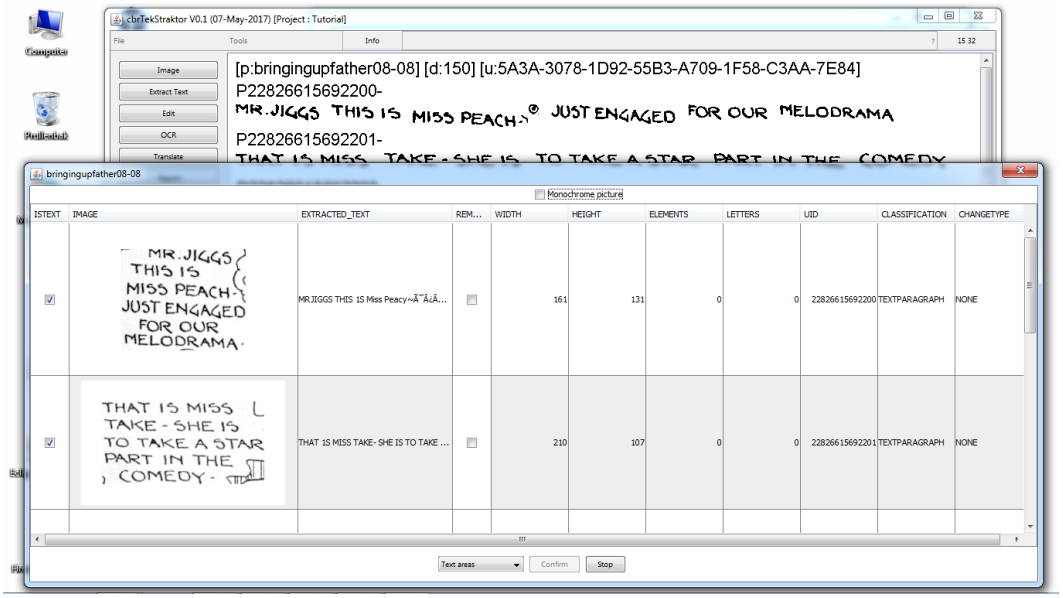
The Header on the Tesseract OCR file is used to determine the maximum accuracy of the OCR process. The accuracy is reported on the log file.

See the "Developer Notes" section for detailed information on the OCR Folder and files.

Fourth Step

The OCR Result file is read and parsed. The text of each paragraph is stored in the Language File which is part of the Archive file.

The OCR'ed text is read from the Archive file and displayed on the following screen.

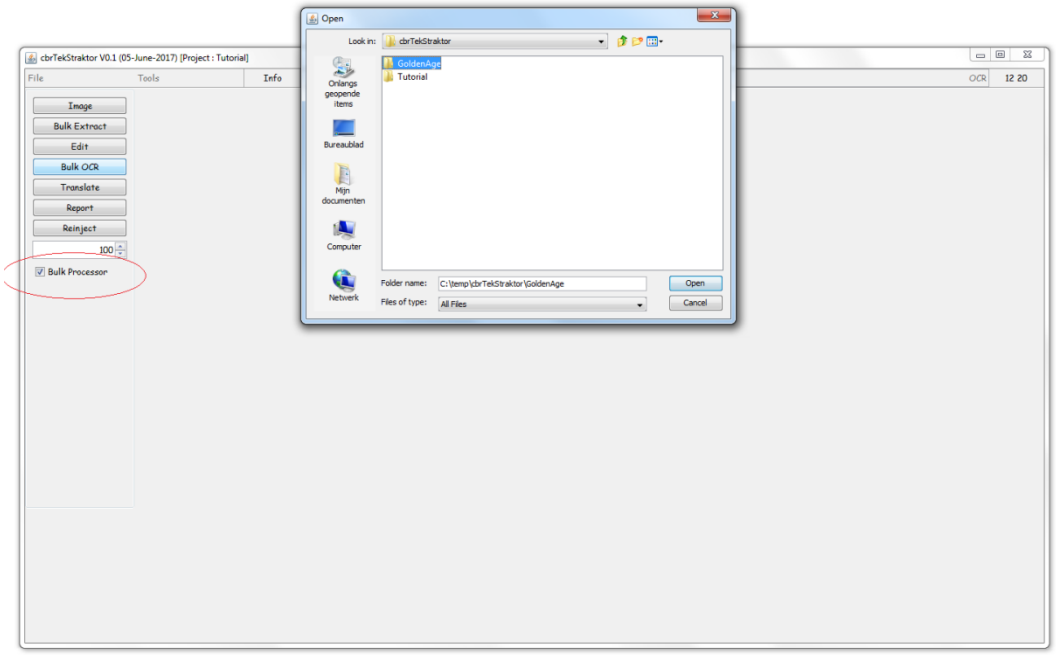


Bulk mode

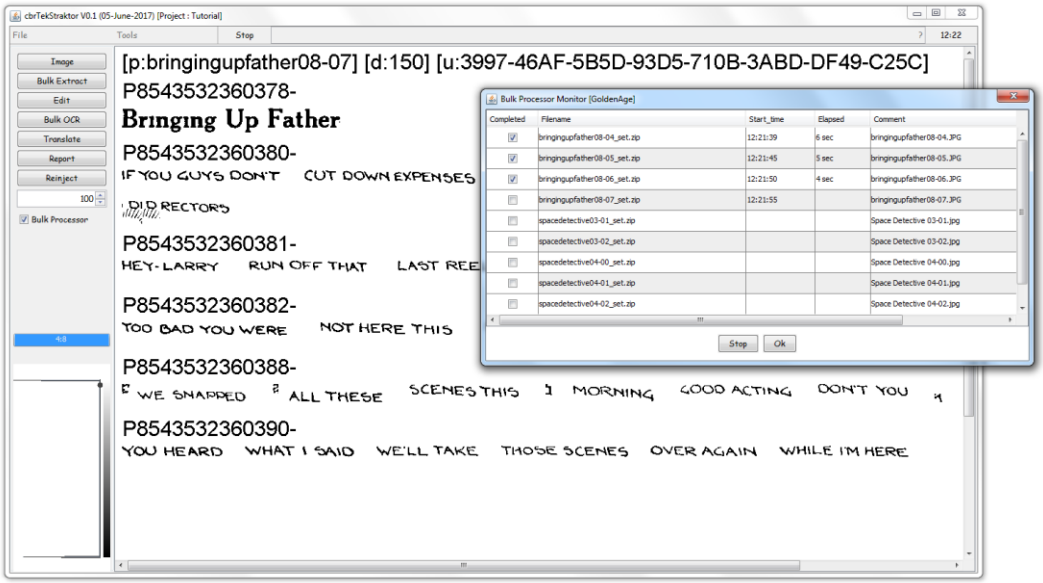
The OCR process can run on entire sets of comic book pages. The bulk extraction will process all images within a single folder, of which the text has previously been extracted.

The Bulk OCR process is similar to the single page OCR process. It can be interrupted by clicking on the "Stop" button on the marquee.

OCR Bulk mode initial step : folder selection



OCR Bulk mode : progress monitor

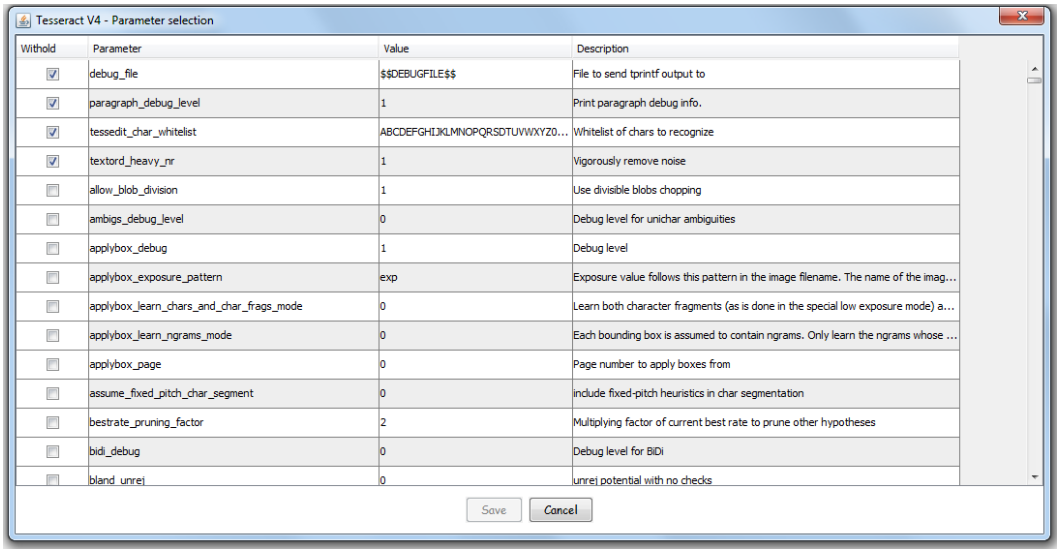


Tesseract Option File

Tesseract has loads of control parameter settings which can be used to modify its behavior. A list of all parameters with default value and short description can be retrieved by issuing the following command: tesseract --print-parameters

The cbrTekStraktor application enables to browse through the various Tesseract V4 parameters and to set or unset those.

The Tesseract option dialog is accessed via “File > Properties > Tesseract options”.



The options which have been activated to be used are displayed at the beginning of the list and have the tick box “Withold” set.

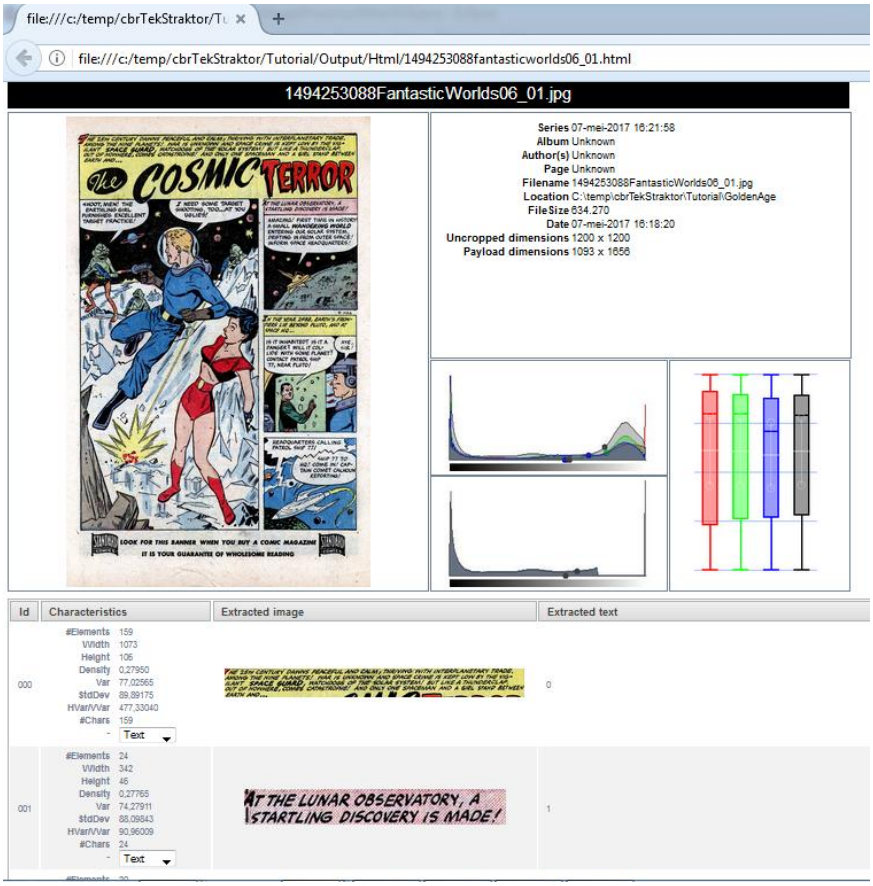
Note. $$DEBUGFILE$$ is an internal cbrTekStraktor variable for the default name of the file holdingTesseract Logging information;TesseractLog.txt which is located in the $PROJECTDIR/OCR folder.

Note. If you close the monitor dialog during a bulk run, you can re-open it via “Tools > More > Monitor”.

Chapter 9

HowTo : Report

An overview report will be created inHTML format when you click on themain screen's "Report" button. The report will be displayed in Mozilla or any other browser that you have specified to be the vehicle reporting client.



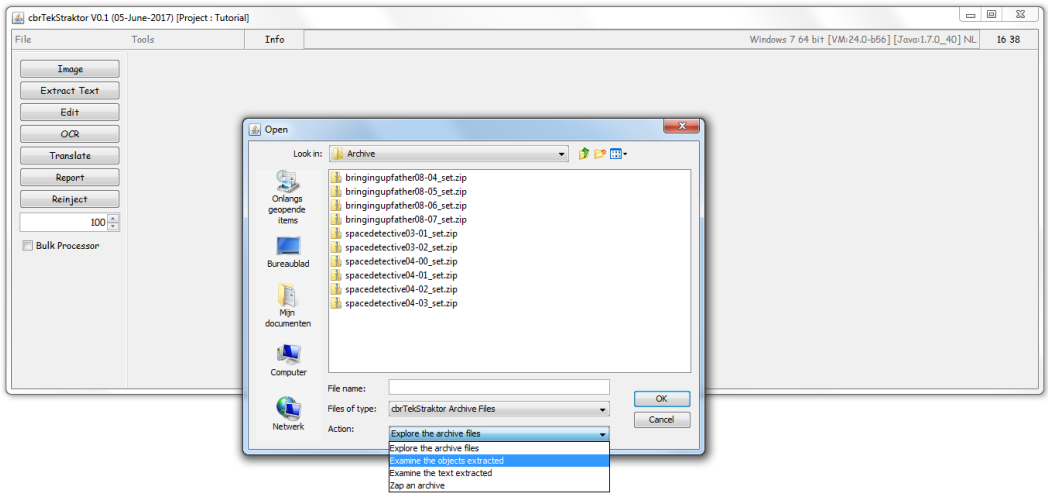
The Report functionality is rather sparse and will be enhanced in future releases.

Chapter A

HowTo : Miscellaneous items

Archive browser

The archive browser utility "Tools > More > Archive Browser" enables to examine the contents of a cbrTekStraktor archive file.



The "Action" item on the dialogs provides to

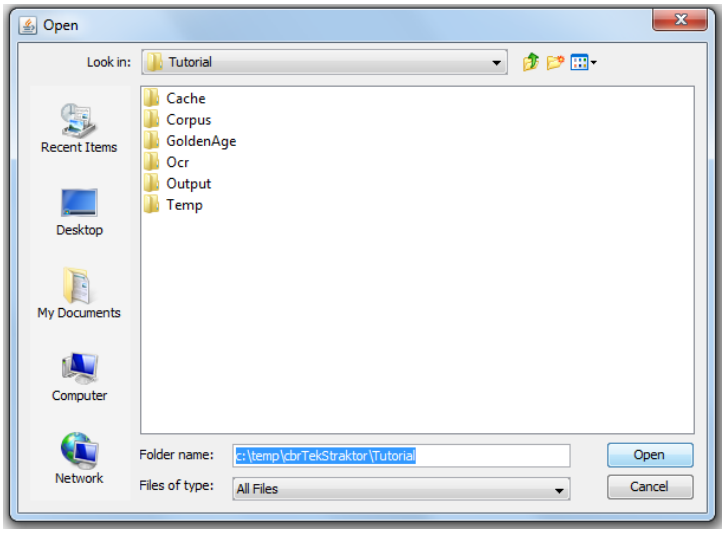
- Explore the archive file; which gives an overview all files in the archive
- Examine the objects extracted; which will open the STAT xml file in a web browser
- Examine the text extracted; which will open the Language (extracted text and translated text) in a web browser
- Zap an archive, i.e. delete an archive file

Exporting all extracted text

The option "File > Export > Export Text" enables to export all textual information to a single file. The file comprises the results of the OCR process, the modifications to the OCR'ed text and the translated texts. Exported text can be used to quickly modify of translate texts.

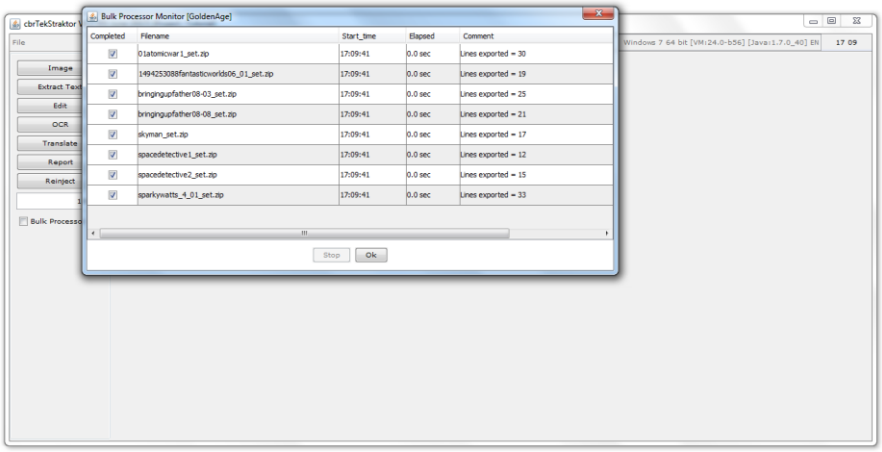
First step

Select the folder containing the comic book pages from which the texts should be exported.



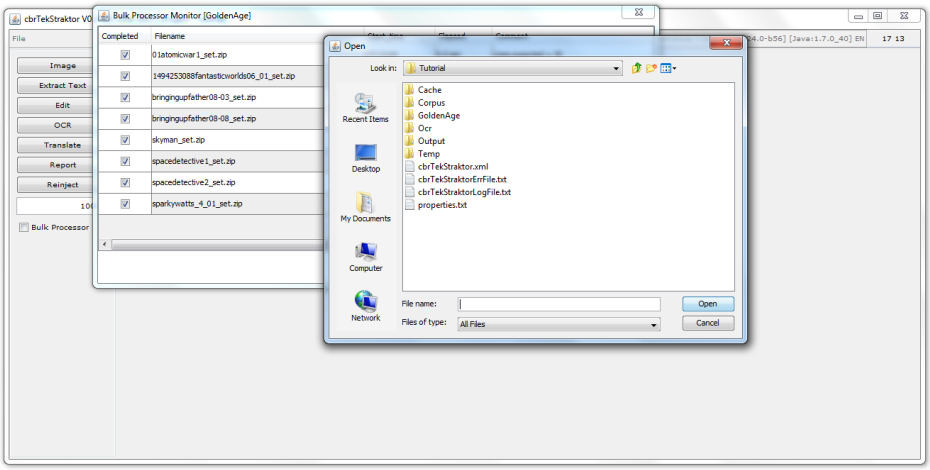
Second step

The text extraction process runs. The batch monitor screen is displayed. The monitor screen will close automatically a few seconds after the export is completed.



Third step

A dialog box opens in which you are able to define the provide the location and name of the export file.



Example export

```
<!-- Application    : cbrTekStraktor V0.2 (05-June-2017) -->
<!-- Start          : 02-JUN-2017 21:55:51 -->
<!-- Folder         : C:\temp\cbrTekStraktor\Tutorial\GoldenAge -
->
<!-- Encoding       : ISO-8859-1 -->
<!-- Format Version : 20170601 -->

_____________________________________________________________
<!-- UID Start         [795B-7AA8-39EA-7EE8-F1D4-3E10-06DB-0717]
-->
<!-- Image             [bringingupfather08-03.JPG] -->
<!-- CMXUID            [bringingupfather08-03] -->
<!--Artefact Language [ENGLISH] -->

<!-- Language [ENGLISH] -->
$30253054928566: THAT MUST pF A PICTURE THEY ARC REHEARING
$30253054928567: The count de Cay got me to buy a studio for
$100000
$30253054928568: whatstrig?
$30253054928569: OH" IM JUST CRAZY To se A MOVE Stam
$30253054928571: Lovely
$30253054928583: HE Cemtaminr DID THAT WELL
$30253054928584: SHE THE STAQ? WHAT 15 THE NaMEOf THIS PAN >
$30253054928585: THAT'S NO PLAy: THAT'S THE SHEmirP. Th??STUDIO
1?? ATTACHED FOR Gaamics
$30253054928595: cises
$30253054928596: her'. Fextune Service


<!-- Language [FRENCH] -->
```

```
$30253054928567: Le comte de Cay m'a convaincu d'achèterune
studio de cinéma pour $100000
$30253054928569: Oh je suis folle. Je veux être une star du
cinéma.
$30253054928571: Parfait.
```

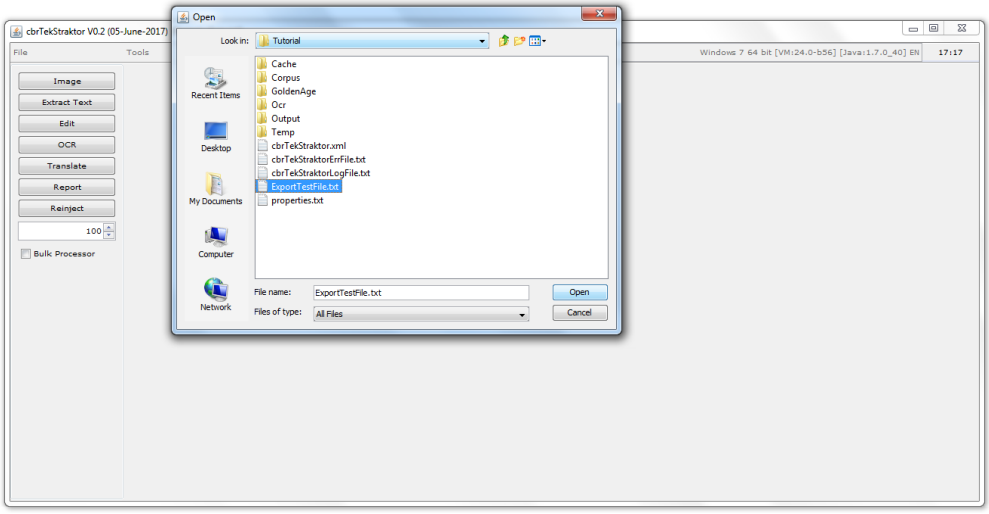
<text abbreviated>

The texts are grouped per image file, per language and per paragraph.Each paragraph has a unique UID, e.g.30253054928571. UIDs are enclosed between a dollar sign and a semi-colon. You can quickly assess the correctness of the OCR and modify or correct where deemed appropriate. Alternatively you can quickly enter translated text and prepare it for import. When entering or modifying text, make sure not to change the structure of the file, i.e. leave a space between the UID's terminating semi-colon and the text.

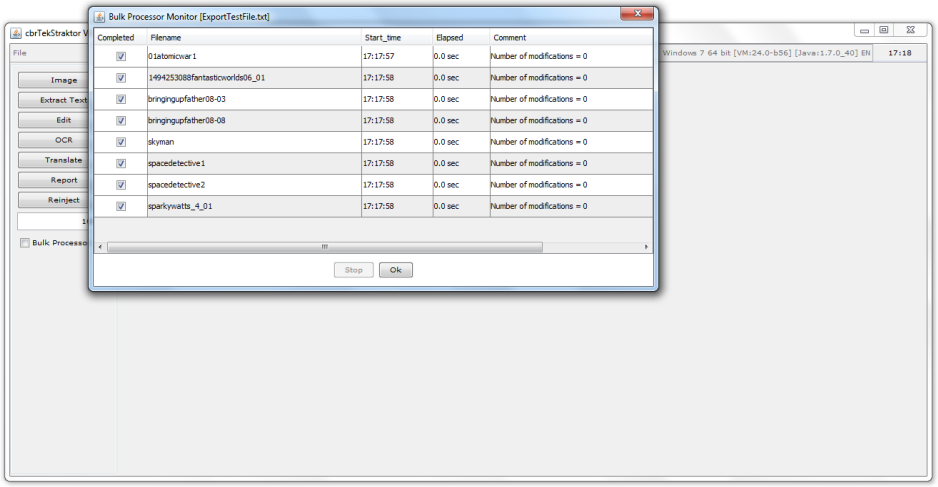
Importing text

This option to be found under "File > Import > Import text" enables to load text from a text source file into the cbrTekStraktor archive file. Its purpose is to re-import modified or translated texts into the cbrTekStraktor application.
The structure of the source file must adhere to the format of the text Export file (see previous section)

First step
Select the import file from which you want to import data.



Second step
The data from the file will be import. A monitor progress screen will be shown.



Round tripping

Round tripping is a quick way for modifying and translating text. Just perform the following steps.

- Export the text for a given folder
- Modify, translate or spell-check the exported data in a text editor.
- Import the modified data

Logging information

&&todo

Chapter B

Developer notes

This section contains information that might be useful for developers.

Folder structure of the application

cbrTekStraktor relies on a predefinedand static folder structure.

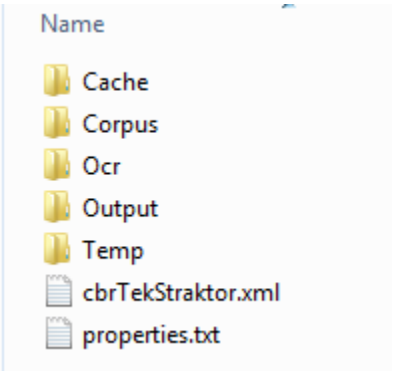
$PROJECTDIR folder

The root folder is the Project Directory or root folder ($PROJECTDIR).

By creating several root folders, the application is able create multiple and separated projects.

The root of the folder structure can be specified on the command line when starting the application (option –D). If the root folder is not specified as a command line parameter, it will be defaulted to "c:\temp\cbrTekStraktor" or $HOME/cbrTekStraktor.

The $PROJECTDIR must adhere to the following structure.



$PROJECTDIR folders

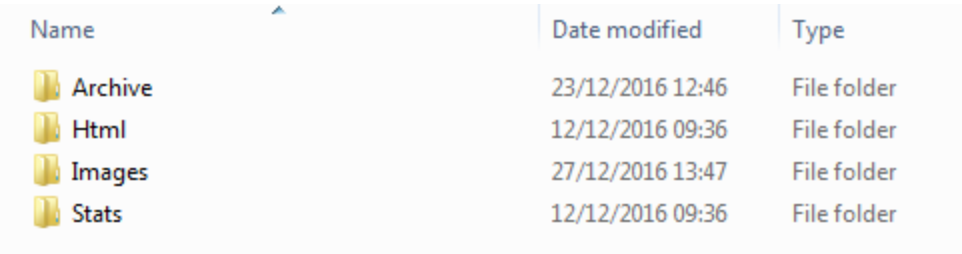
Folder	Description
Cache	This is a mandatory directory. It is used to cache files temporarily, for example when in Edit mode files are cached in this folder.
Corpus	This is a mandatory directory. It comprises statistical information gathered by the application, e.g. timing information.
Output	Required directory. Additional information to be found in the next section.
Temp	This is a required directory. It is used to store temporary files, for example when the Image screen Info dialog is opened, the boxdiagram.png of the RGB box plot diagram are stored in this directory.
OCR	Required directory. It is used to store the Tesseract configuration, debug and result files; as well as a PNG image of the text to be OCR'ed.

$PROJECTDIR files

File	Description
cbrTekStraktor.xml	This is a required configuration file
properties.txt.	This file comprises the GUI properties of the application, for example the width and height of the main canvas. It is created and maintained by the application.
cbrTekStraktorLogFile.txt	File with logging information
cbrTekStraktorErrFile.txt	File with error information

Output folder

These are the folders which are to be found in $PROJECTDIR/Output



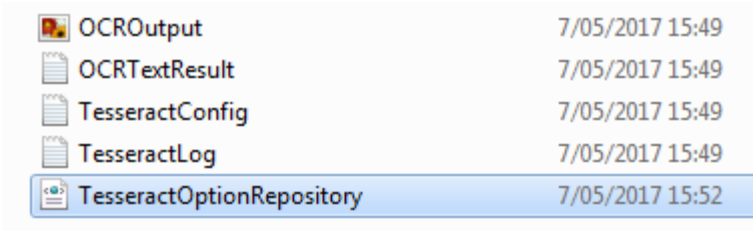
Name	Date modified	Type
Archive	23/12/2016 12:46	File folder
Html	12/12/2016 09:36	File folder
Images	27/12/2016 13:47	File folder
Stats	12/12/2016 09:36	File folder

File	Description
Archive	This is a required folder in which the cbrTekStraktor Archives are stored. An archive is a ZIP file comprising reports, statistical, pictorial and textual information all of which have been generated by the application. See the next section for an overview of the content of an archive file.
HTML	This a required folder in which the HTML reports are stored. It also comprises the CSS (Custom Style Sheet) file (cbrTekStraktorCSS.txt). If the CSS file is missing, a new one will automatically be created by the application. There is a single HTML file for each CBR file (<CMXUID>.html). When the HTML report file is no longer required it is automatically transferred from the HTML folder into the applicable Archive file. Conversely HTML reports are extracted from the Archive files when a report is requested.
Images	This is a required folder for temporarily storing images created by the application. In general temporary files will have a name preceded by 'z' e.g. zPeakDiag_wonderwoman.png is the RGB Peak histogram image.
Stats	This is a required folder for storing the statistical information on each CBR file in XML format in a file named <CMXUID>.xml. Detailed information in the statistical elements maintained in this file is available at the end of this appendix.

Note. A housekeeping routine is automatically performed on a frequent basis. This routine will remove obsolete files in the $PROJECTDIR folders. The housekeeping routine can also manually be started from the "Tools>Housekeeping" menu.

OCR Folder

These files might be present in the $PROJECTDIR/OCR folder



File	Description
TesseractOptionRepository.xml	This file will only be present if you used the "properties>Tesseract Option". The file comprises all options for Apache Tesseract 4.0.
TesseractLog.txt	This is the debug file which the Tesseract client creates during the OCR process.
TesseractConfig.txt	This is the Tesseract Parameter file which is created by cbrTekStraktor before the Tesseract client is called. It comprises the Tesseract options that have been defined in "Properties>Tesseract Options"
OCRTextResult.txt	This is the result file created by Tesseract.
OCROutput.png	This is the image that has is created from extracting the characters form the comic book image.

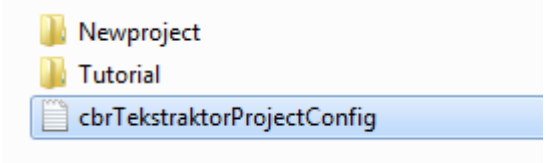
Corpus folder

Filename	Comment
AllStat.txt	This file collates various metrics and statistical information on the CBRs in the project. These statistics are created when pressing the "Statistics" button on the "Tools" menu.
TimingAccuracyStats.txt	Statistical information which is used to monitor the accuracy of the execution time predictive analysis. This module used Euclidian distances.
TimingInputStats.txt	Timing info of the various process gathered by the application whilst executing these processes.

File structures of the application

Project File

A reference to project which was last accessed will be stored in the configuration file "cbrTekStraktorProjectConfig.txt".



This file is located one folder above the current $PROJETDIR folder. In most cases this will be in C:\temp\cbrTekStraktor or when using Linux it will be found in $HOME/cbrTekStraktor.

The content of the Project Properties File is rather sparse and might look like this

```
================================================
cbrTekStraktor V0.1 (07-May-2017)
Started=07-may-2017 13:37:30
Stopped=07-may-2017 13:38:36
================================================
EntryFolder=c:\temp\cbrTekStraktor
RecentProject=Tutorial
```

cbrTekStraktor configuration file

"cbrTekStraktor.xml" is the main configuration file and is located in the $PROJECTDIR folder.

Example configuration file

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Application : cbrTekStraktor V0.1 (01-MAY-2017) -->
<!-- File Created: 01-MAY-2017 17:47:07 -->
<cbrTekStraktor>
<Project>
        <Created>20170407174617</Created>
        <Updated>20170407174707</Updated>
        <Name>Tutorial</Name>
```

```
        <Description>Created by cbrTekStraktor V0.1 (01-MAY-
        2017)</Description>
        <dateformat>dd-MMM-yyHH:mm:ss</dateformat>
        <LoggingLevel>9</LoggingLevel>
        <MeanCharacterCount>500</MeanCharacterCount>
        <HorizontalVerticalVarianceThreshold>5</HorizontalVerticalVariance
        Threshold>
        <PreferredFont>Verdana</PreferredFont>
        <PreferredFontSize>10</PreferredFontSize>
        <Encoding>UTF_8</Encoding>
        <TesseractFolder>C:\temp\Tesseract-OCR-4\Tesseract-
        OCR</TesseractFolder>
        <Browser>CHROME</Browser>
    </Project>
    </cbrTekStraktor>
```

Tag	Comment
Browser	IS used to define which Web Brower to use when reporting. Supported values are {CHROME,MOZILLA,EXPLORER}
Created	The time the cbrTekStraktor project was created
Dateformat	Enables to specify the display format of date and time information.
Description	A description of the project
Encoding	{UTF8,UTF16,LATIN1, ASCI}
HorizontalVerticalVarianceThreshold	A threshold value which is used to identify connected components which are characters. This value should not be modified.
Logginglevel	A number ranging from 0 to 9 to define the level of detail of the logging information. Setting the level to 9 will provide the most detailed logging information.
MeanCharacterCount	This is a threshold value which is used to identify the cluster with textual information.

Name	The name of the cbrTekStraktor project
PreferredFont	The name of the font to be used in the majority of the application's screens and dialogs.
PreferredFontSize	The size of the preferred font.
TesseractFolder	This is the folder where the Tesseract OCR application is installed.
Updated	The time the cbrTekStraktor project has been updated

Content of the archive file

The Archive files are to be found in the $ROOTDIR/Output/Archive folder. An archive file name always ends on "_set.zip".

An archive is a ZIP file comprising reports, statistical, pictorial and textual information generated by the application.

The archive file contains various files.

Filename	Comment
Binarized_Out.png	This is the monochrome version of the comic book image file. It is used by the OCR component.
{CMXUID}.html	This is the HTML report file.
{CMXUID}_lang.xml	XML file containing the extracted and translated textual information
{CMXUID}_Lang_Ver_{YYMMDDHHMISS}.xml	All versions of the Lang.XML file are maintained in the archive. Versions are timestamped.
{CMXUID}_stat.xml	XML file comprising the statistical and graphical information of the comic book image file.
{CMXUID}_Stat_Ver_{YYMM	Previous version of the Stat.XML file

DDHHMISS.xml	
{CMXUID}{00.NNN}.png	These are the cut-out images of the text and non-text paragraphs. These images are used by the reporting component.
zBoxBiagr_{CMXUID}.png	Box Plot diagram of the RGB histogram. This image is used by the reporting component.
zCharacts_{CMXUID}.png	This is the image comprising the cut-out paragraphs. It is created and displayed at the end of the text extraction process and stored in the archive for further usage by the reporting component.
zClusters_{CMXUID}.png	This is an image that comprises an overview of the clusters identified during the text extraction process.
zColrHist_{CMXUID}.png	An image of RGB histogram used by the reporting component.
zGrayHist_{CMXUID}.png	An image of the grayscale histogram used by the reporting component.
zMetaData_{CMXUID}.xml	This is the Comic Book Metadata XML.
zPeakDiag_{CMXUID}.png	An image of the Peak Detection histogram.

cbrTekstraktor STAT XML file

The STAT XML file comprises the majority of the results of the image and text processing activities on an Image file.
Detailed information on this file is provided in a separate section at the end of this appendix.

Language file

The language file contains the result of the OCR and translation processes.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Application : cbrTekStraktor V0.1 (Mon 01-MAY-2017) -->
<!-- Start : 01-MAY-2017 09:50:26 -->
<!-- Last update : 01-MAY-2017 11:22:43 -->
<ComicPageText>
```

```
<ProcessHistory>
        <ProcessTimeStamp>20170501095026</ProcessTimeStamp>
        <ProcessTimeStamp>20170501112243</ProcessTimeStamp>
</ProcessHistory>
<Languages>
        <OriginalLanguage>French</OriginalLanguage>
        <TranslationLanguage01>Afrikaans</TranslationLanguage01>
        <TranslationLanguage02>Albanian</TranslationLanguage02>
        .. etc ..
</Languages>
<ComicPageInfo>
        <File>
                <FileName>comic01.jpg</FileName>
                <FilePath>C:\temp\cmcProc\test</FilePath>
                <FileSize>288775</FileSize>
        </File>
        <Image>
                <CMXUID>comic01</CMXUID>
                <UID>6C58-E110-A319-547F-F401-5341-3623-2EC9</UID>
        </Image>
</ComicPageInfo>
<PageText>
<!--NumberOfParagraphs : includes Text Paragraphs and Non-text
Paragraphs -->
<NumberOfParagraphs>11</NumberOfParagraphs>
<TextBundle>
        <TextBundleIdx>100</TextBundleIdx>
        <TextBundleUID>8870019149256</TextBundleUID>
        <TextConfidence>text</TextConfidence>
        <TextBundleRemoved>false</TextBundleRemoved>
        <TextBundleChangeDate>20170501131957</TextBundleChangeDate
        >
        <TextOCR>
        <![CDATA[Viensvoir]]>
        </TextOCR>
        <TextFrom>
        <![CDATA[Viensvoir]]>
        </TextFrom>
        <TranslatedText_French>
        <![CDATA[Viensvoir]]>
        </TranslatedText_French>
</TextBundle>
</PageText>
```

</ComicPageText>

Tag	Comment
TextBundleChangeDate	The time the content of the paragraph was created or updated.
TextBundleIdx	This is the sequence number of the paragraph.
TextBundleRemoved	Possible vales are {True,False}
TextBundleUID	This is a Unique Identifier of the paragraph
TextConfidence	Possible values are {Text,Nontext}
TextFrom	This field contains the text in its original language
TextOCR	This is the result of the OCR operation on the paragraph
TranslatedText_{language}	This field contains the translated text.

zFiles

The following zFiles are present in the $PROJECTDIR\temp folder when the extraction and edit processes are active. The files are subsequently stored in the Archive file.

Image File	Description
zBoxDiagr	An image of the RGB frequency distribution's Quartile Box diagrams
zCharacters	An image file in which the character clusters are visualized
zClusters	A image file in which the clusters are vizualised
zColHist	Color Histogram image
zGrayHist	Grayscale histogram image
zPeakHist	Peak Histogram, which is used to determine whether an image file comprises a black and white, grayscale or color image. image

zMetadata File

The characteristics of a single commix page are stored in the file $PROJECTDIR\Output\Archive\zMetadata_<CMXUID>.xml.

This is an example of a zMetaData _<CMXUID>.xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Application : cbrTekStraktor V0.1 (Sun 16-APR-2017) -->
<!-- Created : 16-APR-2017 09:27:36 -->
<PageMetadata>
<FolderName>C:\temp\cmcProc\test</FolderName>
<FileName>superman_01.jpg</FileName>
<UID>6C58-E110-A319-547F-F401-5341-3623-2EC9</UID>
<CMXUID>superman01</CMXUID>
<ISBN/>
<Series/>
<SeriesSequence>0</SeriesSequence>
<BookName/>
<PageNumber>1</PageNumber>
<WriterName/>
<PencillerName/>
<ColourerName/>
<Language>FRENCH</Language>
<ColourSchema>colour</ColourSchema>
<ProximityTolerance>TIGHT</ProximityTolerance>
<BinarizeClassificationMethod>SLOW_SAUVOLA</BinarizeClassificationMeth
od>
<ClusterClassificationMethod>AUTOMATIC</ClusterClassificationMethod>
<TesseractCuration>USE_IMAGE_DPI</TesseractCuration>
<Browser>CHROME</Browser>
<CropImage>CROP_IMAGE</CropImage>
</PageMetadata>
```

Tesseract command file

The following command will be generated when to run Tesseract. This is a temporary file which is removed by the application once Tesseract has completed the OCR.

```
C:\temp\Tesseract-OCR-4\Tesseract-OCR>tesseract
c:\temp\cbrTekStraktor\Tutorial\Ocr\OCROutput.png
c:\temp\cbrTekStraktor\Tutorial\Ocr\OCRTextResult -l eng
c:\temp\cbrTekStraktor\Tutorial\Ocr\TesseractConfig.txt
```

Example TesseractOptionRepository file

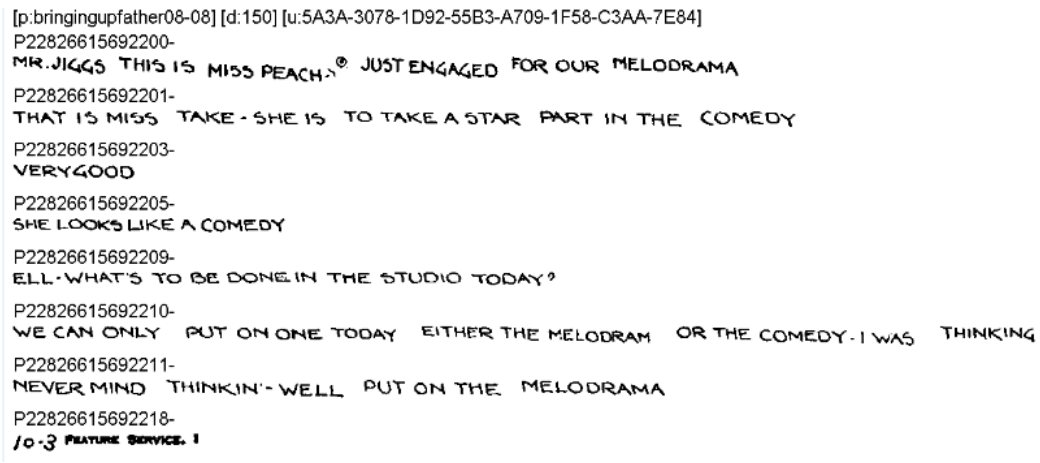
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Application : cbrTekStraktor V0.1 (07-May-2017) -->
<!-- File Created: 07-MEI-2017 15:52:32 -->
<!-- Overview of the possible Tesseract Options -->
<TesseractOptionList>
<![CDATA[
<Option><Withold>false</Withold><Parameter>allow_blob_division</Parameter><Value>1</Value><Description>Use divisible blobs chopping</Description></Option>

<Option><Withold>false</Withold><Parameter>ambigs_debug_level</Parameter><Value>0</Value><Description>Debug level for unichar ambiguities</Description></Option>

<Option><Withold>false</Withold><Parameter>applybox_debug</Parameter><Value>1</Value><Description>Debug level</Description></Option>
```

File abbreviated

Example OCROutput.PNG file



Example TesseractConfig file

```
debug_file c:\temp\cbrTekStraktor\Tutorial\Ocr\TesseractLog.txt
paragraph_debug_level 1
tessedit_char_whitelist ABCDEFGHIJKLMNOPQRSDTUVWXYZ012345789
textord_heavy_nr 1
```

Example OCRResultFile file

```
[p:bringingupfather08-08] [d:150] [u:5A3A-3078-1D92-55B3-A709-
1F58-C3AA-7E84]
P22826615692200
MRJIGGS THIS 1S Miss Peacy~® JUST ENgAgep FOR OUR MELODRaMA
P22826615692201
THAT 1S MISS TAKE- SHE IS TO TAKE A STAR® PART IN THE COMEDy
P22826615692203
VERyGOOD
P22826615692205
SHE LOOKS LIKE A COMEDY
P22826615692208
ELL WHAT'S TO BE DONE IN THE STUDIO TODAY *
P22826615692210
weCAn Only - POY on omeTooay | ETHER THE mELopRam - OR THE
COMEDY. | was - THiNKNq
P22826615692211
NEVER MIND | Thinkin' weir PUY On THE MELODRAmA
P22826615692218
Jo -g reatume.sarvice. 1
```

Example TesseractLog file

```
Tesseract Open Source OCR Engine v4.00.00alpha with Leptonica
# Final Paragraph Segmentation
```

```
#row space ..lword[widthSEL]                    rword[widthSEL]
[lmarg,lind;rind,rmarg] model text
0    8        [p:bringingupfather08-08][236SEl] [u:5A3A-3078-1D92-55B3-
A709-1F58-C3AA-7E84][490sEl] [  1,  0;  0,  0]       S:1
[p:bringingupfather08-08] [d:150] [u:5A3A-3078-1D92-55B3-A709-1F58-C3AA-
7E84]
1    12       P22826615692200[180Sel]           P22826615692200[180sel]
[  1,  1;737,  0]       C:1   P22826615692200
2    13       MRJIGGS[108Sel]                   MELODRaMA[147sel]
[  1, -1; 68,  0]       S:1   MRJIGGS THIS 1S Miss Peacy~® JUST ENgAgep
FOR OUR MELODRaMA
3    12       P22826615692201[177Sel]           P22826615692201[177sel]
[  1,  2;739,  0]       C:1   P22826615692201
4    13       THAT[62Sel]                       COMEDy[111sel]
[  1,  0;  0,  0]       S:1   THAT 1S MISS TAKE- SHE IS TO TAKE A STAR®
PART IN THE COMEDy
Active Paragraph Models:
 1: margin: 1, first_indent: 0, body_indent: 0, alignment: LEFT
# Final Paragraph Segmentation
#row space ..lword[widthSEL]         rword[widthSEL]
[lmarg,lind;rind,rmarg] model text
0    7        P22826615692203[180Sel] P22826615692203[180sel] [  1,  1; -
1,  1]       U:0   P22826615692203
1    7        VERyGOOD[142Sel]        VERyGOOD[142sel]        [  1, -1;
39,  1]       U:0   VERyGOOD
Active Paragraph Models:
# Final Paragraph Segmentation
#row space ..lword[widthSEL]         rword[widthSEL]
[lmarg,lind;rind,rmarg] model text
0    7        P22826615692205[180Sel] P22826615692205[180sel] [  1,
1;136,  1]       U:0   P22826615692205
1    8        SHE[44Sel]              COMEDY[98sel]           [  1, -1; -
1,  1]       U:0   SHE LOOKS LIKE A COMEDY
Active Paragraph Models:
# Final Paragraph Segmentation
#row space ..lword[widthSEL]         rword[widthSEL]
[lmarg,lind;rind,rmarg] model text
0    11       P22826615692208[180Sel] P22826615692208[180sel] [  1,
1;453,  1]       U:0   P22826615692208
1    11       ELL[47Sel]              *[8SEL]                 [  1, -1; -
1,  1]       U:0   ELL WHAT'S TO BE DONE IN THE STUDIO TODAY *
Active Paragraph Models:
# Final Paragraph Segmentation
#row space ..lword[widthSEL]         rword[widthSEL]
[lmarg,lind;rind,rmarg] model text
0    10       P22826615692210[180Sel] P22826615692210[180sel] [  1,
0;1017,  1]      U:0   P22826615692210
1    11       we[39sel]               THiNKNq[111sel]         [  1, -1; -
1,  1]       U:0   we CAn Only - POY on omeTooay | ETHER THE mELopRam -
OR THE COMEDY. | was - THiNKNq
Active Paragraph Models:
# Final Paragraph Segmentation
#row space ..lword[widthSEL]         rword[widthSEL]
[lmarg,lind;rind,rmarg] model text
0    7        P22826615692211[177Sel] P22826615692211[177sel] [  1,
1;585,  1]       U:0   P22826615692211
1    11       NEVER[84Sel]            MELODRAmA[168sel]       [  1, -1; -
1,  1]       U:0   NEVER MIND | Thinkin' weir PUY On THE MELODRAmA
Active Paragraph Models:
# Final Paragraph Segmentation
```

```
#row space ..lword[widthSEL]          rword[widthSEL]
[lmarg,lind;rind,rmarg] model text
0    7        P22826615692218[180Sel] P22826615692218[180sel] [  1,  1;
34,  1]       U:0   P22826615692218
1    10       Jo[25Sel]                1[5SeL]                 [  1, -1; -
1,  1]       U:0   Jo -g reatume.sarvice. 1
Active Paragraph Models:
```

AllStat File

The AllStat.txt file is to be found in the $PROJECTDIR/Corpus/Stats. The file is generated by selecting "Tools > More > Statistics". Its purpose is to provide key statistical information (metrics) on each Comic Book Page processed. These metrics can be used for further analysis or prediction.
The file is a comma delimited ASCI text file.

Column	Comment
UID	The UID of the image processed
UncroppedWidth	The Uncropped Width in pixels
UncroppedHeigth	The Uncropped Height in pixels
PayloadWidth	The Payload width
PayloadHeigth	The PayloadHeight
NbrOfElementsInLetterCluster	The number of elements found in the paragraph classified to be the text paragraph.
NbrOfParagraphs	The number of paragraphs
NbrOFLetterParagraphs	The number of paragraphs which have been identified to comprise text
NbrOfLetters	The total number of characters
Colour	{TRUE, FALSE} Set to TRUEby the Color schema detection algorithm if a full color scheme is detected.
MonochromeDetected	{TRUE,FALSE} set to TRUE if the color scheme detection logic found a monochrome schema

MonochromeDetectionStatus	{TRUE,FALSE}-{POSITIVE,NEGATIVE} True-positive : correctly identified monochrome schema False-positive : incorrectly identified monochrome schema True-negative: correctly identified color schema False-negative: incorrectly identified color schema
NbrPeak	The number of peaks found by the color schema detection logic
NbrValidPreak	The number of valid peaks
%PeakCoverage	Coverage of pixels within valid peaks

Example
UID,UncroppedWidth,UncroppedHeigth,PayloadWidth,PayloadHeigth,NbrOfElementsInLetterCluster,NbrOfParagraphs,NbrOFLetterParagraphs,NbrOfLetters,Colour,MonochromeDetected,MonochromeDetectionStatus,NbrPeak,NbrValidPreak,%PeakCoverage
e7f2-2af1-d553-b70e-8cb8-b350-ebc3-bd42,975,1465,877,1355,858,25,15,826,unknown,false,false-negative,8,0,46%,
ce6d-3395-7b87-e18e-b53f-2f55-112f-d37e,1200,1857,1093,1656,597,36,10,485,unknown,false,false-negative,9,0,21%,
795b-7aa8-39ea-7ee8-f1d4-3e10-06db-0717,1024,1105,998,1012,321,31,10,250,unknown,true,false-positive,39,0,87%,

TimingInputStat file

The TimingInputStat.txt file is to be found in the $PROJECTDIR/Corpus/Stats. The file stores the elapsed time in nanoseconds of various text extraction and image loading processes.
The timing information is gathered during the image loading and text extraction activities.
It is the prime source for estimating the duration of image loading and text extraction activities, i.e. the lead time of a process is calculated using Euclidian Distance and is used by the progress indicator on the main screen.
It is a vertical pipe delimited ASCI file.

Column	Comment
UID	UID of the image
Width	Uncropped width of the image (in pixels)
Heigth	Uncropped height of the image (in pixels)
FileSize	The filesize of the image (in bytes)
ColourScheme	{COLOR,GRAYSCALE,MONOCHROME}
FileType	{PNG,JPG,GIF,JPEG}
BinarizationType	This is the binarization mechanism used, e.g. OTSU, SAUVOLA, etc.
ConnectedComponents	The overall number of connected components
Paragraphs	The number of paragraphs
BWDensity	Black and White density of the entire picture
ImageLoadTime	The time required to load the image (in nanoseconds)
PageLoadTime	The time required to load the comic book page (the page is an object that envelops the image) in nanoseconds.
Preprocess	The duration of the pre-process step (in nanoseconds)
BinarizeTime	The duration of the binarization process (in nanoseconds)
CoCoTime	The duration of t connected components process(in nanoseconds)
LetterTime	The duration of the process that identifies and expands characters (in nanoseconds). See the section where the text extraction process is explained.
ParagraphTime	The duration for creating an processing the contents of

	paragraphs
OverheadTime	This is the duration of the end to end text extraction process minus all of the above durations.
Timestamp	The time the record was written to the file

Example
UID|Width|Heigth|FileSize|ColourScheme|FileType|ConnectedComponents|Paragraphs|BWDensity|ImageLoadTime|PageLoadTime|Preprocess|BinarizeTime|CoCoTime|LetterTime|ParagraphTime|OverheadTime
629E-BF04-2D3B-5C27-29BB-C6BC-FBFC-E7F2-2AF1-D553-B70E-8CB8-B350-EBC3-BD42|975|1465|256874|COLOR|JPG|SLOW_SAUVOLA|10183|25|0.3010666072368622|245314798|61627908|199194031|4462646481|246614265|904632564|834179973|925567697|03-JUN-2017 08:27:33
CE6D-3395-7B87-E18E-B53F-2F55-112F-D37E|1200|1857|634270|COLOR|JPG|SLOW_SAUVOLA|31328|36|0.38288450241088867|310097424|83530141|245702033|7934692773|346997986|1079128458|1232142512|1286570859|03-JUN-2017 08:27:46
795B-7AA8-39EA-7EE8-F1D4-3E10-06DB-0717|1024|1105|354385|GRAYSCALE|JPG|SLOW_SAUVOLA|3518|31|0.18374991416931152|186662399|64333023|179845335|3157337579|63852631|549405945|691859669|989257304|03-JUN-2017 08:27:52

TimingAccuracyStat

The TimingAccuracyStat.txt file is to be found in the $PROJECTDIR/Corpus/Stats. The file stores difference between the calculated and actual process times of various image loading and text processing activities.
The timing information is gathered during the image loading and text extraction activities.
Its major objective tis to report on the accuracy of the process time prediction algorithm.

Column	Comment
Timestamp	The time the record was creaed
EstimatedImage	Estimated time to load the image
ActualImage	Actual time to load the image
Ratio	Correctness or accuracy ratio (estimated – actual) /

	actual
EstimatedBeforePreprocess	Estimated Before preprocess
ActualbeforPreprocess	Actual before preprocess
Ratio	Correctness ratio
EstimatedBeforeBinarize	Estimated before Binarize step
ActualBeforeBinarize	Actual before binarize step
Ratio	Correctness ratio
EstimatedBeforeCoCo	Estimated before Connected component
ActualBeforeCoCo	Actual before connected components
Ratio	Accuracy ratio

Example
TimeStamp (EstimatedImage,ActualImage,Ratio)
(EstimatedBeforePreprocess,ActualbeforPreprocess,Ratio)
(EstimatedBeforeBinarize,ActualBeforeBinarize,Ratio)
(EstimatedBeforeCoCo,ActualBeforeCoCo,Ratio)
03-JUN-2017 13:05:45 (250ms, 283ms, -11%) (5886ms, 14896ms, -60%) (5882ms, 14896ms, -60%) (5882ms, 14896ms, -60%)
03-JUN-2017 13:12:00 (306ms, 408ms, -25%) (7879ms, 8606ms, -8%) (7879ms, 8606ms, -8%) (7879ms, 8606ms, -8%)
03-JUN-2017 13:12:13 (393ms, 301ms, 30%) (12518ms, 12436ms, 0%) (12518ms, 12436ms, 0%) (12518ms, 12436ms, 0%)
03-JUN-2017 13:12:19 (283ms, 151ms, 87%) (5882ms, 5926ms, 0%) (5882ms, 5926ms, 0%) (5882ms, 5926ms, 0%)

cbrTekstraktor STAT XML file

The Stat file comprises detailed information on the characteristics of the file and image; as well as the results of the image analysis and text extraction processes.

```
<!--  Application : cbrTekStraktor V0.1 (05-June-2017)  -->
<!--  Start : 05-JUN-2017 16:01:28  -->
<ComicPage>
  <ProcessHistory>
    <ProcessTimeStamp>20170605160128</ProcessTimeStamp>
  </ProcessHistory>
  <File>
    <FileName>Space Detective 03-02.jpg</FileName>
    <FilePath>C:\temp\cbrTekStraktor\GoldenAge</FilePath>
    <FileSize>1116676</FileSize>
    <FileLastModification>05-JUN-2017 11:07:14</FileLastModification>
  </File>
  <Image>
    <CMXUID>spacedetective03-02</CMXUID>
    <UID>D49E-D846-049A-76EB-B5E5-570E-420A-5695</UID>
    <UncroppedWidth>800</UncroppedWidth>
    <UncroppedHeigth>1167</UncroppedHeigth>
    <UncroppedSize>933600</UncroppedSize>
    <PayLoadX>26</PayLoadX>
    <PayLoadY>26</PayLoadY>
    <PayLoadWidth>743</PayLoadWidth>
    <PayLoadHeigth>1098</PayLoadHeigth>
    <PayLoadSize>815814</PayLoadSize>
    <PhysicalWidthDPI>200</PhysicalWidthDPI>
    <PhysicalHeigthDPI>200</PhysicalHeigthDPI>
    <ColourSchema>unknown</ColourSchema>
    <MonochromeDetected>false</MonochromeDetected>
    <MonochromeDetectedStatus>false-negative</MonochromeDetectedStatus>
    <Grayscale>false</Grayscale>
    <NbrOfPeaks>10</NbrOfPeaks>
    <NbrOfValidPeaks>1</NbrOfValidPeaks>
    <PeakCoverage>0.3762874892887746</PeakCoverage>
    <BlackBorder>false</BlackBorder>
    <GrainyBackGround>true</GrainyBackGround>
  </Image>
  <OriginalHistogram>...</OriginalHistogram>
  <PayloadHistogram>...</PayloadHistogram>
  <ConnectedComponentFrequencyDistribution>...</ConnectedComponentFrequencyDistribution>
  <!--...-->
  <ClusterClassification>...</ClusterClassification>
  <ConnectedComponentClusters>...</ConnectedComponentClusters>
  <!--...-->
  <!--  Number of Single Element Paragraphs [42] in [702]  -->
  <!--...-->
  <FinalConnectedComponentClusters>...</FinalConnectedComponentClusters>
  <paragraphs>...</paragraphs>
  <GraphicalEditorArea>...</GraphicalEditorArea>
  <TimingInfoNanoSec>...</TimingInfoNanoSec>
  <!--  Stop : 05-JUN-2017 16:01:35  -->
</ComicPage>
```


Major segments in the Stat file

The root node in the Stat file is <ComicPage>. The following segments are its immediate descendants.

Tag	Comment
ProcessHistory	Creation and modification timestamps
File	Details on the Image file
Image	High level details on the image
OriginalHistogram	The RGB histogram data of the original image
PayloadHistogram	The RGB histogram of the image once the margins have been removed.

ConnectedComponentFrequencyDistribution	Connected Components frequency distribution
ClusterClassification	Results of the Cluster classification process
ConnectedComponentClusters	Details on the connected components before the post-process
FinalConnectedComponentClusters	Details on the post-processed connected components, e.g., reallocation of character components.
paragraphs	Details information on the text and non-text paragraphs
GraphicalEditorArea	A dump of the connected components and objects that are part of the text and non-text paragraphs
TimingInfoNanoSec	Timing information on nanosecond granularity.

ProcessHistory
&&todo

File
&&todo

Image
&&todo

OriginalHistogram
&&todo

PayloadHistogram
&&todo

ConnectedComponentFrequencyDistribution
&&todo

ClusterClassification
&&todo

ConnectedComponentClusters
&&todo

FinalConnectedComponentClusters
&&todo

Paragraphs
&&todo

GraphicalEditorArea
&&todo

TimingInfoNanoSec
&&todo

Text extraction process

Step1 . Determining the payload area of a comic book page image (determining the width and height of the margins)

Step 2. Cropping the image to its payload area (on the Comic Book Metadata Dialog one can opt not to crop the image)

Step 3. The grayscale version of the cropped image is displayed.

Step 4.Binarization of the image. Various binarization methods can manually be selected on the previously displayed Comic Book Metadata Dialog screen: Otsu, Niblak or Sauvola. It is recommended to use either Niblak or Sauvola.

Step 5. Display of binarized image

Step 6.Gathering the connected components, i.e. gathering information on every single graphical element present on the image. See Connected Component in appendix.

Step 7.K-Means clustering of the connected components. This is a straightforward classification of the connected components. The classification criterion is the pixel height of a connected component
On a comic book page, letters or characters more or less all have the same height. The idea is to create groups of connected components which all have a similar height and therefore constitute a cluster which contains merely characters. cbrTekStraktor uses K = 5. See K-Means clustering in appendix.

Step 8. Identification of those connected components which are characters. This is performed in various steps, e.g. by analyzing the number of vertical and horizontal elements of a single connected component. Typically a character of the Latin Script has 2 or 3 horizontal elements and 1 or 2 vertical elements. There are also more white pixels than dark pixels in a character, i.e. the density of these connected components should be less than 50%. Connected component having a pixel height of less than 6 are excluded and classified as noise.

Step 9. Identification of the K-Means cluster comprising the characters. There are typically between 300 and 500 characters on a single comic book page. So the cluster holding approximately 500 connected components resembling a character is initially chosen to be the "Text Cluster" (a.k.a. in Dutch as the Letter Cluster).Additional fine-tuning steps are performed .

It is possible to override the automatic detection of the cluster comprising characters via the "Cluster Classification Method" drop-down on the Image Info dialog.

Step 10.Expansion of the characters. Previously determined characters which are part of the TextCluster are subsequently grouped based on their proximity on the image. This will result in a set of characters that are part of the same speech balloon or text area. CbrTekStraktor uses the noun "paragraphs" for these groups. The "Proximity Tolerance" can be set on the Comic Book metadata Screen to be either tight, lenient, wide or ultra-wide.

Step 11.Adjustment of characters. Any Connected component that is part of the area determined by the previously found Text Paragraphs boundaries are re-assessed whether to be a potential character or not. Paragraphs that comprise little or no characters are then set to be "non-character" or "non-text" paragraphs; the remaining paragraphs are hence onward referred to as "character" or "text" paragraphs.

Step 12. The results of the text extraction are stored in an archive file (see the definition of the stat.xml and language file). Cut-outs of the paragraphs are displayed. Text paragraphs have a green border and non-textual paragraphs have a red border. Frames have a bluish border.

Image processing concepts

This section comprises a quick overview of the image processing concepts, e.g. image processing filters, used in the application.

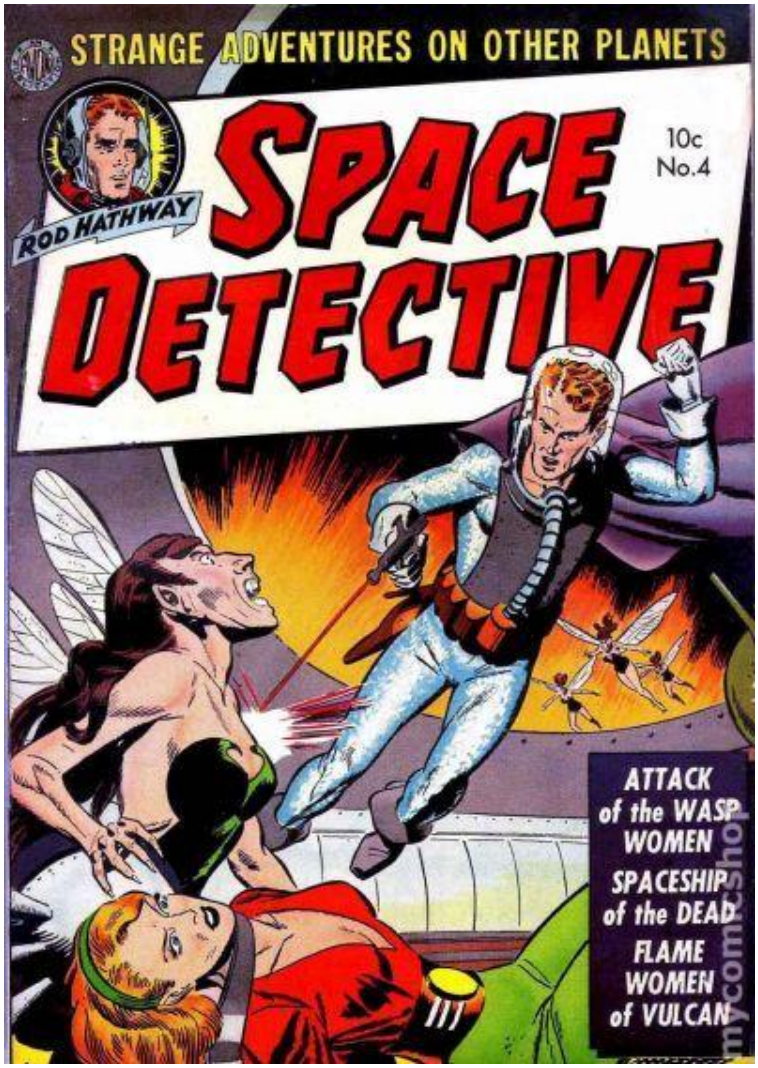
Concept	Comment
Convolution	[Wikipedia] In mathematics convolution is a mathematical operation on two functions; it produces a third function, that is typically viewed as a modified version of one of the original functions, giving the integral of the pointwise multiplication of the two functions as a function of the amount that one of the original functions is translated. It has applications that include probability, statistics, computer vision, natural language processing, image and signal processing, engineering, and differential equations. In image processing, a kernel, convolution matrix, or mask is a small matrix. It is used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image.
Gaussian blur	[Wikipedia] In image processing, a Gaussian blur is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through translucent scree. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales. Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function. Since the Fourier transform of a Gaussian is another Gaussian, applying a Gaussian blur has the effect of reducing the image's high-frequency components; a Gaussian blur is thus a low pass filter.
Grayscale	[Wikipedia] In photography and computing, a grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the

	strongest. A common strategy is to use the principles of photometry or, more broadly, colorimetry to match the luminance of the grayscale image to the luminance of the original color image. To convert a color from a colorspace based on an RGB color model to a grayscale representation of its luminance, weighted sums must be calculated in a linear RGB space, that is, after the gamma compression function has been removed first via gamma expansion. Formula: 0.2126R + 0.7152G + 0.0722B
Histogram equalization	[Wikipedia] Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique and an invertible operator. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.
HSL/HSV	[Wikipedia] HSL and HSV are the two most common cylindrical-coordinate representations of points in an RGB color model. The two representations rearrange the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the Cartesian (cube) representation. Developed in the 1970s for computer graphics applications, HSL and HSV are used today in color pickers, in image editing software, and less commonly in image analysis and computer vision. HSL stands for hue, saturation, and lightness (or luminosity), and is

	also often called HLS. HSV stands for hue, saturation, and value, and is also often called HSB (B for brightness). A third model, common in computer vision applications, is HSI (I for intensity). However, while typically consistent, these definitions are not standardized, and any of these abbreviations might be used for any of these three or several other related cylindrical models. (For technical definitions of these terms, see below.) In each cylinder, the angle around the central vertical axis corresponds to "hue", the distance from the axis corresponds to "saturation", and the distance along the axis corresponds to "lightness", "value" or "brightness". Note that while "hue" in HSL and HSV refers to the same attribute, their definitions of "saturation" differ dramatically. Because HSL and HSV are simple transformations of device-dependent RGB models, the physical colors they define depend on the colors of the red, green, and blue primaries of the device or of the particular RGB space, and on the gamma correction used to represent the amounts of those primaries. As a result, each unique RGB device has unique HSL and HSV absolute color spaces to accompany it (just as it has unique RGB absolute color space to accompany it), and the same numerical HSL or HSV values (just as numerical RGB values) may be displayed differently by different devices.
Image gradient	[Wikipedia] An image gradient is a directional change in the intensity or color in an image. In graphics software for digital image editing, the term gradient or color gradient is also used for a gradual blend of color which can be considered as an even gradation from low to high values, as used from white to black in the images to the right. Another name for this is color progression. Mathematically, the gradient of a two-variable function (here the image intensity function) at each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction.
NiblackSauvola	Niblack and Sauvola thresholds are local thresholding techniques that are useful for images where the background is not uniform,

	especially for text recognition. Instead of calculating a single global threshold for the entire image, several thresholds are calculated for every pixel by using specific formulae that take into account the mean and standard deviation of the local neighborhood (defined by a window centered around the pixel).
OTSU	[Wikipedia] In computer vision and image processing, Otsu's method, named after Nobuyuki Otsu, is used to automatically perform clustering-based image thresholding or the reduction of a graylevel image to a binary image. The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal, or equivalently (because the sum of pairwise squared distances is constant), so that their inter-class variance is maximal.
RGB	[Wikipedia] The RGB color model is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green and blue. The main purpose of the RGB color model is for the sensing, representation and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors. To form a color with RGB, three light beams (one red, one green and one blue) must be superimposed (for example by emission from a black screen or by reflection from a white screen). Each of the three beams is called a component of that color, and each of them can have an arbitrary intensity, from fully off to fully on, in the mixture. Zero intensity for each component gives the darkest color (no light, considered the black), and full intensity of each gives a white; the quality of this white depends on the nature of the primary light sources, but if they are properly balanced, the result is a neutral white matching the system's white point. When the intensities for all the components are the same, the result is a shade of gray, darker or lighter depending on the intensity. When the intensities are different, the result is a colorized hue, more or less saturated

	depending on the difference of the strongest and weakest of the intensities of the primary colors employed. When one of the components has the strongest intensity, the color is a hue near this primary color (reddish, greenish or bluish), and when two components have the same strongest intensity, then the color is a hue of a secondary color (a shade of cyan, magenta or yellow).
RGBA	[Wikipedia] RGBA stands for red green blue alpha. While it is sometimes described as a color space, it is actually simply a use of the RGB color model, with extra alpha channel information. The color is RGB, and may belong to any RGB color space, but an integral alpha value as invented by Catmull and Smith between 1971 and 1972 enables alpha compositing. The alpha channel is normally used as an opacity channel. If a pixel has a value of 0% in its alpha channel, it is fully transparent (and, thus, invisible), whereas a value of 100% in the alpha channel gives a fully opaque pixel (traditional digital images). Values between 0% and 100% make it possible for pixels to show through a background like a glass, an effect not possible with simple binary (transparent or opaque) transparency. It allows easy image compositing.
Sobel	[Wikipedia] The Sobel operator, sometimes called the Sobel–Feldman operator or Sobel filter, is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasizing edges. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel–Feldman operator is either the corresponding gradient vector or the norm of this vector. The Sobel–Feldman operator is based on convolving the image with a small, separable, and integer-valued filter in the horizontal and vertical directions and is therefore relatively inexpensive in terms of computations.



Space Detective - Issue 4
Published Apr 1952 by Avon Publications.
A groovy funny mystery book, cover art by Gene Fawcette and artwork by Gerald McCann.