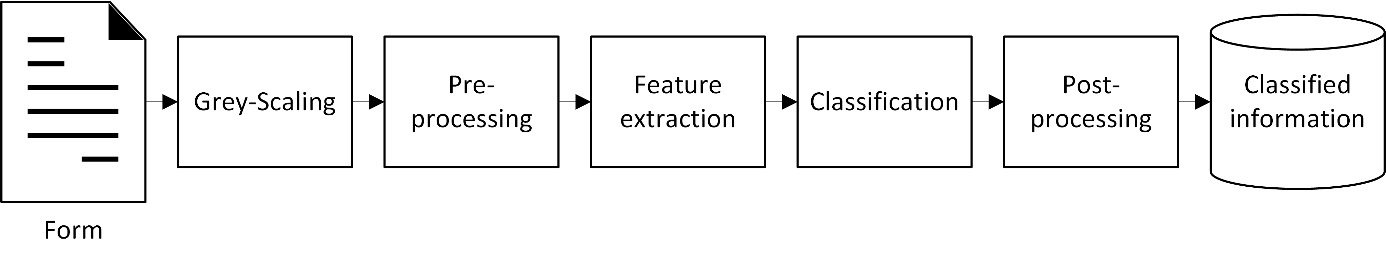
# OCR – State of the art, possibilities and drawbacks

During the processing of a form, the image of it is not only scanned. To retrieve additional information and work it we will use Optical Character Recognition (OCR).

The process of retrieving data, for instance in form of characters or numbers, requires several steps. In the beginning, the image to be processed is converted to a gray-scale image. Then preprocessing takes place. In this process several algorithms such as noise-reduction and the canny-edge-algorithm are applied on the image to reduce irritating and / or unnecessary information in the image and to enhance contrast.

After that, features are extracted. Those features are single characters whose vectors are defined afterwards. With the information about the feature vector it is possible to classify each character (and to detect which character of the alphabet is most likely to be represented by the feature).

When all characters have been classified, post-processing takes place. Here possible failures can be corrected for example by comparing words with a predefined dictionary.  
These steps are also shown in figure 3.1:



The whole process of OCR is part of research since decades and several papers, dissertations and books have been published on the matter. Developing our own OCR algorithm would not only exceed the size of this master thesis, but also most likely retrieve less successful results than already developed and improved algorithms. Instead, this chapter will introduce several available OCR algorithms and compare open source solutions to find the best fit for our need.

## Currently available OCR algorithms

OCR has been of interest for companies over many years. Hence we expect several possible solutions we could choose. As the amount of solutions can easily be very high we want to reduce the presented solutions to a maximum amount of 10.

Each description of a solution will contain information about the license used, the supported operating systems, programming languages used as well as if there is a software development kit. Also general information about the company will be given, the currently released version of the solution and the release date and, if existent, the number of languages supported.

See also: https://en.wikipedia.org/wiki/Comparison\_of\_optical\_character\_recognition\_software

### ABBYY Fine Reader

ABBYY Fine Reader is a proprietary solution from the identically named company ABBYY founded in 1989. It is usable for all three operating systems, whereby linux-distributions are only supported as a command-line-interface. ABBYY supports a SDK for all three operating systems. Although it is written in C/C++ there exists a wrapper for java development for linux and windows(TODO: Link für Abbyy broschüre version 11). The SDK is currently in Version 11, it supports 185 languages, the latest update was on 03.10.2016.

### Anyline SDK

Anyline is an Austrian company founded in 2013 who aim at OCR solutions for mobile systems. They offer their SDK as a free license for non-commercial use. However, as they are focused on mobile systems, they do not explicitly support Windows-Desktop, Linux or MacOS.

It can be developed with Java and Objective-C as well as Swift, C# and Javascript. The current version of the SDK is 3.8.1 and has been released on 13.01.2017.

Currently supported are two languages: English and German.

### Asprise OCR

Asprise has been founded in 1998 in Singapure. The company offers OCR SDKs in various programming languages (Java, C#, VB.NET, Python, C/C++ and Delphi Pascal). The SDKs are under loyalty-free license and therefore proprietary. More than 20 languages are supported, English and German are included.

The SDK supports Windows, MacOS and Linux, whereby support of multiple operating systems at once increases the price.

### GOCR

GOCR is an OCR application started by Joerg Schulenburg in 2000. The program is developed under the GNU Public License.

The latest version is 0.5 and has been released in March 2013. It is working under Windows, Linux and MacOS. The code is written in C, but is not known if there is a SDK which enables usage of the application inside another application. The number of supported languages is also unkown.

### LEADTOOLS

Leadtools is an American company founded in 1990 and offers various products in the range of document and image processing.

The Leadtools OCR Engine can be used as an SDK and integrated in another application. Development with the SDK is possible with C# and VB as well as C/C++ and Java (and some others). The engine supports more than 40 languages, containing German and can be used on Windows, Linux and MacOS.

The current version of the SDK is 19 and has been released in December 2014.

### MathOCR

MathOCR is a document recognition system written in Java with focus on formulas. The MathOCR project started in March 2014 and is based on the GNU General Public Liense.

The current version of the application is 0.0.3, which was released in May 2015 and is therefore still in a pre-alpha status. It can be used on Windows, Linux and MacOS. The amount of supported languages is not stated on the project page.

### OCROpus

The OCRopus Open Source OCR System is an open source system developed and maintained by the German Research Laboratory for Artificial Intelligence under guidance by Thomas M. Breuel. It is licensed under the Apache 2.0 License.

The command-line application is written in Python and C++ and only supports Linux as Operating System. It currently uses the Tesseract as a text line recognizer but will be replace it in the future.

The current stable version is 1.0 and has been released in November 2014. The amount of supported languages is unkown, whereby it is able to work with latin-based languages.

### OCRad

OCRad is a free OCR application under the GNU Public License and part of the GNU project. Antonio Diaz Diaz developed the application since 2003.

The current version is 0.25 and has been released in April 2015. It comes as a stand-alone console application but can also be used in the background by other applications.

### OmniPage Capture SDK

The Nuance Communication Inc. offers an OCR tool called OmniPage Capture SDK, which enables document processing on Windows, Linux and MacOS. Depending on the underlying operating system it supports C/C++, Objective-C or C# and VB.NET.

The current version is 20 and has been released in 2016. It supports over 120 languages (German included).

### Tesseract

The Tesseract OCR Engine historically was an early project developed by Hewlett Packard between 1984 and 1994. In 2005, it was put on an open source license. It is currently maintained by Google under the Apache 2.0 license.

Tesseract is originally written in C/C++ and can be used on Windows, MacOS and Linux. There also exists a wrapper which allows development in Java, the open source project Tess4J.

The current stable version of the Tesseract is 3.04.01 and has been released in February 2016. It supports over 100 languages (including German).

## Comparison between open source algorithms

While several companies exist that offer good OCR libraries and SDKs, we have to stick to free software, since we are not able to afford a proprietary license. In a later state of the application, it could be possible to switch to a proprietary solution in order to increase our OCR efficiency. Until then, we will decide for the best fitting open source algorithm library and improve our efficiency by preprocessing the forms ourselves. This is explained in Chapter 4, Module 1.

Upon the 10 presented solutions, only 5 are free for use. These are: GOCR, Tesseract, OCROpus, MathOCR and OCRad.

The following table shows the named solutions and shows their differences regarding their version, the latest release date, the supported programming languages and operating systems as well as the license they are put on:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Application | **GOCR** | **Tesseract** | **OCROpus** | **MathOCR** | **OCRad** |
| Version | 0.5 | 3.04.01 | 1.0 | 0.0.3 | 0.25 |
| Release Date | 03.2013 | 02.2016 | 11.2014 | 05.2015 | 04.2015 |
| Supported programming languages | C | C/C++, Java (with Tess4J) | C++,  Python | Java | C++ |
| Supported Operating Systems | Windows, Linux, MacOS | Windows, Linux, MacOS | Linux | Windows, Linux, MacOS | Windows, Linux, MacOS |
| License | GNU Public License | Apache 2.0 | Apache 2.0 | GNU Public License | GNU Public License |

MathOCR is a relatively young application and therefore in a pre-alpha state. OCRad and GOCR are one step closer to the first major release. OCROpus has reached that state on November 2014. Tesseract is already on Version 3.04 and has recently released an alpha version of 4.0.

While Tesseract, OCROpus and OCRad are supporting C++, GOCR is only working with C whereas MathOCR is only Java. Tesseract is supporting C and Java as well (while using Tess4J as a wrapper). OCROpus instead is working with Python, too.

All solutions support Windows, Linux and MacOS except OCROpus, which is only working on Linux.

While GOCR, MathOCR and OCRad are licensed under the GNU Public License, Tesseract and OCROpus are licensed under the Apache 2.0 License. The difference between those two is mainly the following: Applications that are developed under usage of another program under the GNU Public License have to be licensed under the GNU Public License as well. The Apache 2.0 License allows usage of other application and enables free choice of licensing, but requires the mentioning of the underlying use of an Apache 2.0 licensed application.

## Decision finding and explanation

In the beginning of this thesis, it was defined that the application should work on a Linux based operating system and be written in Java. While all presented solutions support Linux as an Operating System, not all of them work with Java as a programming language. In addition to that, OCROpus only supports Linux, which is fine for the current focus of the application, but could be of an issue later on if it should be ported to another operating system.

MathOCR is in a pre-alpha state which makes it difficult to use due to several missing functionalities and persistent bugs in the code. As it is mostly focused on mathematical equations and formulas, we will not consider MathOCR any longer, even though it supports Java as a programming language.

As explained in section 3.2, the GNU Public License requires our application to be licensed under the GNU Public License as well if we use another code which is licensed under this license. Therefore, the Apache 2.0 license is considered better, as it allows us to decide about the license for ourselves.

In the interest of the application, we want to use solutions that are up-to-date and are still under development. Therefore, the latest release date gives us insights about the activity on a project. Since a lot of open source projects suffer from missing developers, we expect longer development cycles. But the last version of GOCR has been released around 4 years ago. Hence we consider GOCR as not up-to-date anymore.

The following table shows the solutions again, but with underlying colors regarding their ability to fit to our problem. Green is used as a best fit, whereas red significates a major problem. Yellow shows that this attribute is not as good as others but no kick-out criterion.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Application | **GOCR** | **Tesseract** | **OCROpus** | **MathOCR** | **OCRad** |
| Version | 0.5 | 3.04.01 | 1.0 | 0.0.3 | 0.25 |
| Release Date | 03.2013 | 02.2016 | 11.2014 | 05.2015 | 04.2015 |
| Supported programming languages | C | C/C++, Java (with Tess4J) | C++,  Python | Java | C++ |
| Supported Operating Systems | Windows, Linux, MacOS | Windows, Linux, MacOS | Linux | Windows, Linux, MacOS | Windows, Linux, MacOS |
| License | GNU Public License | Apache 2.0 | Apache 2.0 | GNU Public License | GNU Public License |

As shown in the table, MathOCR will not fit our needs as it is a pre-alpha version. GOCR is outdated and also supports only C as a programming language. OCROpus and OCRad only support C++ (and in the case of OCROpus Python). In addition to that, OCROpus could be of a problem when porting the application to other operating systems whereas OCRad is licensed under the GNU Public License.

Hence the Tesseract seems to be the best fit for our application. It is consistently updated and improved and by the history of it, the application itself has grown mature. The possibility to work with Tess4J enables the usage of it with Java. Multiple operating systems are supported and the Apache 2.0 license enables us to decide for ourselves under which license we will put the application.