Introduction to Dynamsoft OCR SDK

1. Introduction to OCR

a. What is OCR

OCR (optical character recognition) is software used by a computer to recognize text in a graphic format and turn it into computer text, which can be read and edited normally. For example, one might take a picture of a car's license plate, and OCR software could then be used to read the text from the picture into a word document. OCR is implemented through a complex system of trained pattern recognition, which can also recognize fonts and formatting. Modern OCR is very accurate, and thus is practical for use in a wide variety of areas, and is constantly being improved through training and artificial intelligence.

b. The power of modern OCR applications

Computer OCR has been developed for over 60 years. In its most primitive form, it was able to recognize most letters of the English alphabet. Today, OCR is very powerful, and software can be found that is able to support almost all languages in usage, with very reasonable accuracy, and it's only getting better.

In many cases, the quality of recognition is dependent on the quality of the image. The ideal image is one that has a plain background with a minimal amount of spots and artifacts. However, modern OCR applicati ons are also powerful enough to detect anomalies and ignore them in processing. Wordlist data is also used to reduce mistakes, as processed words can be compared to dictionary words.

The Tesseract OCR engine is an example of a powerful modern OCR engine, which supports over 40 languages and is flexible enough to be trained to improve accuracy and add new languages. Tesseract is a mature engine that has existed since 1985, created by HP labs and currently developed by Google. Called an "engine", it is the lowest level component of an OCR system, meaning its job is to perform recognition and recognition only. To take full advantage of OCR technology and implement features such as output to complex formats, text formatting, and graphical interfaces, a more complete software package is required.

c. How can OCR be used

While the past was a world where documents were all physical, and the future is a world where documents may all be digital, the present is in a state of transition. In this transition state, physical

and digital documents coexist, and it is important to have technologies like OCR to allow for conversion back and forth.

OCR is useful for a great variety of purposes, including document recovery, data entry, and accessibility. Most applications of OCR are from scanned documents, but in some cases photos are also used. OCR is an essential time saver, as in many cases the only alternative is retyping the document. Some of the ways in which OCR can be used follow:

- Recovering editable text files from scanned documents including faxes
- Categorizing forms based on an approximation of their handwritten contents
- Creating searchable and editable eBooks from book scans
- Searching and editing text from screenshot images
- Computerized reading of books for visually impaired individuals through text-to-speech

While these are just some of the ways that OCR can be used, they show the flexibility of OCR technology in a great variety of fields. Almost all employees of all businesses rely heavily on documents every day, so business usage is al so an important focus in the development of OCR systems.

d. Business applications of OCR

Business usage of OCR generally falls within the field of data organization and input. Many businesses receive documents in a traditional printed form, such as forms that are mailed or faxed in. In other cases, some documents may only be available in written form, such as manuals or printed documents for which the original file has been long lost. Processing of these documents is much more expensive than for documents in a digital form, as they require a human to read the documents and manually categorize or record data.

Using OCR, the manual process is eliminated, only requiring the document to be scanned. After a document has been processed by OCR, its data can be used to automatically categorize it by the computer, and the information can be edited and searched by employees. OCR is used by post offices, libraries, and offices of any kind.

2. Important features in an OCR interface

In the following section, important features that may not be found in all OCR engines will be discussed, as well as reasons why they are important, and why they should be supported by OCR engines and libraries.

a. Accurate recognition with font identification

Accurate recognition is the most obvious feature that is important to OCR. In modern OCR software, recognition is incredibly accurate. Better recognition means less need for manual corrections, and the ability to use the data directly to create things like PDFs, or write to databases. Accurate recognition reduces the manual work required, which saves employees time, and companies money.

Depending on the use of the data, font identification can be very important. In the case of outputting PDFs or word documents, in order to maintain a natural look similar to the original document, the same font should be used. Detecting and using the same font gives a very professional look to OCR processed documents.

b. Flexible language support and trainability

While modern OCR supports a multitude of languages and has great accuracy, OCR is still a developing field. Development will continue almost forever, until recognition can be perfected for every text size, font, language, and handwritten style. While the Tesseract OCR engine supports over 40 languages, there are still written languages and scripts that are not yet supported.

It is incredibly important that an OCR engine be extendable and trainable, so that contributors and developers can all easily add to the pool of knowledge within the engine. Through the power of distributed contributions, languages and scripts from all across the world can be better recognized by OCR. With a proper training mechanism, an engine can be given a document's image, its text and font information, and can use that information to learn how to process such images. Given thousands of these types of images, the recognition data itself can evolve, and contributors can submit their training results to developers to increase accuracy for all users. The engine that provides the best training mechanism is the engine that will grow the fastest.

c. Support for many input file formats

OCR input may come from a few different sources, including scans, online images, and photos. These different sources traditionally use different image formats, and different compression methods. In order to support all appropriate media, OCR software should support all relevant image files, including TIFF images (common for scans) of various compression formats, such as the fax4 format used for blackand-
white. Online images often come in PNG, GIF, or JPEG formats, so it is important that all three are supported. Extensive input format support is important in OCR as it saves time and money in converting formats.

d. Extensive output and export options

As previously discussed, OCR has a great variety of uses, and thus its output may need to be formatted in a great variety of ways. For documents that attempt to mimic the original scanned document, it is important to support output that maintains original formatting and fonts, and outputs in a popular document format such as PDF.

Image-Over-Text i s a useful method of OCR output for PDF documents, where the original source image is written over formatted OCR text. This means that the reader sees no difference between the look of the original document and the OCR processed document. With the OCR text information below the image, readers can select, search, and copy text as if it were any other typed document.

For non-document usage, or usage where the user wishes to create their document manually, it is important that OCR software has the ability to export the resulting data in a useable form. This means that the text, position, font, and size information should all be accessible to the user if desired. This allows great flexibility for programmers that wish to output information in new formats or within their own programs.

e. Intuitive page control and settings

One of the issues common in OCR is page control. Since many original documents consist of multiple pages, OCR needs to process those pages appropriately, and output in a form that respects the original page layout. The TIFF format is one format that supports multiple pages for input, and i ntelli gent OCR engines will read it page-by-page, with options to read specific pages if desired. The PDF output format is ideal for such multi-paged documents, and a good engine should output the appropriate text to the appropriate pages in a PDF document.

3. Dynamsoft OCR SDK

a. Basic introduction

The Dynamsoft OCR SDK by Dynamsoft is an efficient and stable C++ library for OCR operations. It was released in 2012 and is a state of the art solution for delivering OCR to customers. Dynamsoft OCR relieves programmers from worrying about the details, and allows them to simply use an intuitive, well documented API for all their OCR needs.

Dynamsoft OCR is built on top of the highly developed Tesseract OCR engine, which provides the lowest level OCR data, and Dynamsoft's SDK brings it all together for programmers to use. Since Tesseract is a strong base that is constantly being improved on by companies like Google, the SDK will only become more and more useful as recognition becomes even more accurate. The

Dynamsoft OCR SDK was developed with the needs of programmers in mind, and provides all the features required to make the best use of OCR technology.

b. How the Dynamsoft OCR SDK implements these important features of OCR

Dynamsoft OCR fulfills each one of the five previously discussed key features of OCR engines. The SDK not only provides basic OCR results, but also detailed position and format information, including font names, font sizes, line widths, and more. The way this kind of information can be used is demonstrated within the SDK's PDF output itself, where documents can be written plaintext, or using the image-over-text style. Additionally, the library can export this information for use in other applications or formats; the library was built to be both flexible and extensible.

In terms of input, the Dynamsoft OCR SDK supports the TIFF format and all its compression methods, as well as JPEG, GIF, PNG, PNM, and BMP formats. It doesn't matter if your image comes from a scanner, the web, or your camera, Dynamsoft OCR will process i t.

The base of Dynamsoft OCR, Tesseract OCR, supports over 40 languages and is growing even further. Language support varies with popularity, but the community behind the engine comes from all over the world, and anyone can become part of i t by providing data that improves recognition. This process is called "Training", and is well documented and easily performed, which means Tesseract is always getting better through contributions. There's not much holding back an OCR engine that can learn from its own mistakes.

Finally, Dynamsoft OCR's API was designed with intuitiveness in mind, and it supports page control in a way that OCR APIs should. Everything is written in clear object oriented C++ style, but can still interface in a low-level way if need be, with reading and writing directly to and from memory.

c. The SDK's interface

The interface consists of five primary classes:

- **OCR**

The OCR class is the main OCR system, which can be instantiated for a particular language and data source. The way in which OCR is performed can be set here. All OCR operations are performed by this class – it acts as an OCR processor as well as result exporter.

- **PageSet**

A PageSet takes the place of a document, which can contain multiple pages within it. For example, a TIFF file with 3 pages would load into a PageSet with 3 pages. A PageSet could also be created from three different single-page image files, which come together and form a single document.

- **Page**

A Page is just that, a single page of a document. A Page has size, contains lines, and can be part of a PageSet.

- **Line**

A Line is contained within a Page, and consists of many Words. A Line has its own positioning and size information, which can be used to make even rows of its words, as well as underlines or surrounding boxes.

- **Word**

A Word contains text, a font name, font size, and size and positioning information. It is the smallest unit of the OCR system, and is always contained within a Line.

While each class is used internally, they are also flexible enough to be used by programmers using the API. If you wish to provide a file directly to the OCR class, it will do all the work for you, but if you prefer to make your own instance of the Page class and then provide it, that can be done too. The library is both simple and complete, and easy to use for programmers.