Optical Character Recognition

The optical character recognition (OCR) is the process of recognizing text (machine-readable) from the photos or scanned copies of the handwritten documents and pictures with text like sign boards etc. It is so much more useful than a mere scanned document, because it allows computers to do operations like searching, copying text as we do it in every word file or notepad file.

Figure 1: Text recognition user OCR

A few of the many applications of OCR:

1. Automation of human work - OCR saves human effort of retyping the paper based documents when storing the data into a computer. This allows quick storing of data with high quality (since it is not prone to human error). The stored data can be used in several business processes.
2. Machine-translations - instead of manually typing the text of a handwritten document into the translator app, we can just click a picture of the text to get translated.

Internal mechanism

The OCR consists of the following steps

1. Bitmapping
2. Text area and background area recognition
3. Pre-processing
4. Character recognition
5. Post-processing

Bitmapping:
It is the process of converting the input image (RGB or Grayscale) into black and white image.

Figure 2: Bitmap conversion

Text area and background area recognition:
The regions of the converted bitmap are classified into text region and background region. The region of image with dark colour (black) is considered as text and the region of image with light colour (white) is considered as background.

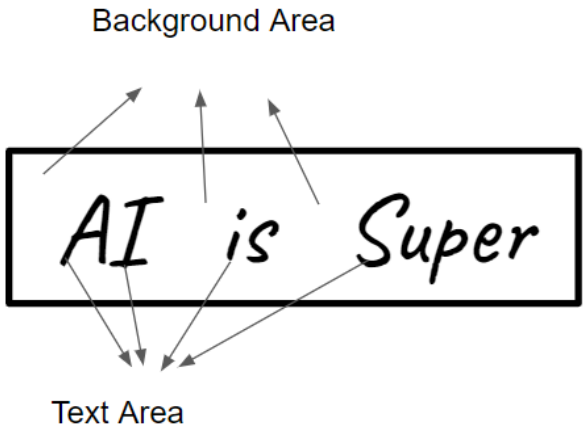

Figure 3: Text area and background area recognition

Pre-processing:
The image is further pre-processed using image processing techniques for alignment corrections (rotations of images) and noise removal (remove image spots and smoothening of edges).

Character recognition:
The pre-processed image is then analysed and processed for character recognition by one of the following techniques - Pattern Recognition/Matching or Feature Detection/Extraction

Terminology - glyph : handwritten or scanned character (symbol, alphabet or number)

Pattern recognition or Pattern Matching:

Here the glyph from the image is compared with already stored glyphs (a large number of different glyphs stored in different styles) to identify what is the glyph.

For example, a glyph of alphabet “h” in the scanned image, is compared with different stored characters (cursive style, times new roman style, etc), it is found that it is letter “h”, since its pattern is closed to existing “h” alphabets in the stored database.

This method works better for tasks like converting the scanned documents to machine readable text files since the fonts and styles of the scanned documents are limited and can be stored easily.

Feature detection or Feature extraction:

Here the low level features of the glyph like loops, lines and edges are detected. Using these low level features and a rule based algorithm, the character recognition is done.

For example, a glyph is recognised as “P”, if it has one loop and one vertical left to the loop- Likewise a glyph is identified as “8” if it has two loops (one over other) and no horizontal and vertical lines.

Post-processing:
The recognised characters are converted to computerised text (e.g.ASCII values) and further used for the application.

OCR + AI:
With the help of AI, this OCR technology is enhanced to a very advanced level (high speed processing and high quality recognition (comparable with human capability)).The OCR + AI models are combinedly used in many business applications today.

Existing Models

There exist many state of the art models with this OCR technology. Few examples are as below.

1. Google Cloud Vision OCR - used in smartphones for scanning and storing documents.
2. IBM Cloud Pak® for Business Automation - it uses OCR for document processing and automation
3. TrOCT - Transformer based Optical character recognition model. These pre-trained models are available in Huggingface.

Transfer learning

If OCR based applications need to be developed for a certain business specific application. It is not necessary (and not advised) to develop them from scratch. We can make use of already available pre-trained models which are trained with huge amounts of data.

Advantages of Transfer learning: We know that in machine learning, the more the data the better the model performance. It is difficult for us to gather a huge amount of data, so by transfer learning we take advantage of a pre-trained model which consists of parameters learned from a huge dataset. Likewise, developing a model might take a lot of time and resources (CPUs, GPUs, TPUs) for training the model, this time and resources can be saved through transfer learning since we train very little for fine tuning of the model, once we have

the pre-trained model. It is like transferring the knowledge and intelligence of a super model to our model.

How to perform transfer learning:
When the application is similar: e.g. we need text recognition of the scanned document. If a model is available with the same functionality, we can take that model (weights and architecture) and re-use it. If required we do fine tuning of model (with the data we have - data close to our application need)

When the application is different: e.g. we need to recognise the font style used in the scanned document, but the pre-trained model is available only for text recognition from the scanned document. Then we freeze the weights of the feature extraction part of the model, and modify the last layers of the model as per our need and perform fine-tuning (training for last layers).

Serving and Evaluation

Serving (RESTful API):
The model can be served as a REST API and can be integrated as a web application or a mobile application.
Input of API - input image, output of API - text content or translated text content.

Evaluation (AB testing and check for bias):
AB testing
The application can be evaluated using AB testing. For example, in a company the employees currently use a version 1.0 version of text recognition software for documenting the data in compute. Now we made some advancement in the software through transfer learning and developed version 2.0. If we release the 2.0 version and we observe improvement in efficiency of the work from employees. Does it mean the version 2.0 is successful? Actually not, causal inference is tricky. This improvement may be because of new CPUs or GPUs installed in the computers or because some bugs were fixed in the last server maintenance.

In order to find the actual causal inference, we randomly select a set of employees and separate them into two groups, for one group we provide actual 2.0 software (with improvements) - Treatment group and another group we provide a dummy 2.0 software (i.e. the already existing version 1.0 software) - Control group. After a period of time (say one month), we can compare their performances of two groups - number of documentation tasks and quality of the documents. Then we come to a conclusion, whether 2.0 is working fine or not.

Evaluation - bias check
Most of the machine learning models are black box models- we know it works with good accuracy but we don't know how it is working. If a model is trained with biased data, this bias might affect the model. So before releasing the product, it needs to be checked for bias, especially bias with respect to gender, race etc.

This is possible with the help of XAI (explainable AI) techniques. By analysing the feature importances using an explainable AI package like SHAP, we can identify whether there is any bias.

References

1. https://aws.amazon.com/what-is/ocr/?nc1=h_ls
2. https://www.ibm.com/cloud/blog/optical-character-recognition
3. https://medium.com/analytics-vidhya/the-power-of-transfer-learning-in-deep-learning-681f86a62f79