A Deep Learning Based Methodology for Information Extraction from Documents in Robotic Process Automation*

Massarenti Nicola[1][0000−0002−8882−4252] and Lazzarinetti Giorgio[1][0000−0003−0326−8742]

Noovle S.p.A, Milan, Italy https://www.noovle.com/en/

Abstract. In recent years, thanks to Optical Character Recognition techniques and technologies to deal with low scan quality and complex document structure, there has been a continuous evolution and automation of the digitization processes to allow Robotic Process Automation. In this paper we propose a methodology based both on deep learning algorithms (as generative adversarial network) and statistical tools (as the Hough transform) for the creation of a digitization system capable of managing critical issues, like low scan quality and complex structure of documents. The methodology is composed of 5 modules to manage the poor quality of scanned documents, identify the template and detect tables in documents, extract and organize the text into an easy-to-query schema and perform queries on it through search patterns. For each module different state-of-the-art algorithms are compared and analyzed, with the aim of identifying the best solution to be adopted in an industrial environment. The implemented methodology is measured with respect to the business needs over real data by comparing the extracted information with the target value and shows performance of 90%, in terms of Gestalt Pattern Matching measure.

Keywords: Robotic Process Automation · Optical Character Recognition · Information Extraction · Deep Learning · Image Denoising.

1 Overview

With the spread of cameras on mobile devices, more and more images of scanned documents are collected in order to be digitized for different uses. Most of the digitization processes are still done manually today, however, thanks to recent advances in machine learning, it is possible to further automate these processes [1].

* Activities were partially funded by Italian ”Ministero dello Sviluppo Economico”, Fondo per la Crescita Sostenibile, Bando “Agenda Digitale”, D.M. Oct. 15th, 2014 - Project n. F/020012/02/X27 - “Smart District 4.0”.

When dealing with information extraction from documents, Optical Character Recognition (OCR) techniques are the key technology, however these alone are not enough to extract all the visual and structural information from scanned documents. Moreover their power is limited when dealing with poor-scan quality documents or with documents with complex structure. In this research we define a methodology for extracting information from scanned or editable documents, trying to manage and limit the noise coming from the low quality of the scans and taking into account document structure. The methodology is designed using real data coming from two different companies and is tested by considering the companies' real business needs. The methodology that we propose firstly uses Generative Adversarial Network (GAN) [12] to clean the scanned documents, then identifies the document template by using Siamese Neural Network (SNN) [19] and then, by using a method based on a computer vision technique called Hough Transform (HT) [27] and on the Google Cloud Vision API [37] for OCR, identifies tables. Then an information mapping process that delegates the personalization of content extraction to the drafting of a set of queries is defined, thus making the information retrieval simpler and more immediate. The rest of this paper is organized as follows. Firstly, in chapter 2, the analysis of the state of the art for information extraction is presented; in chapter 3 the actual use case and the real-world dataset is described; in chapter 4 the defined methodology is shown in details; in chapter 5 the experimental results obtained are resumed; in chapter 6 some conclusions and future directions are mentioned.

2 State of the art

Information retrieval from documents has been an important research area for several decades. With the advent of deep learning, OCR systems have become extremely powerful and usable, thanks to open source systems such as Tesseract [2] and cloud API based solutions such as Google Cloud Vision API [37]. Today, interpreting documents with a simple text layout and good scan quality has become a trivial problem thanks to these recent developments [3], especially if the PDF is software-generated and editable, as described by H. Chao and J. Fan in [4]. In case of non editable PDF (scanned documents) the only applicable solution for text extraction is represented by OCR techniques. OCR techniques generally consist of 5 phases: pre-processing, segmentation, normalization, feature extraction and post-processing. In Table 1 the main algorithms of each phase are presented.

The preprocessing step, which aims at eliminating noise in an image without missing any significant information, traditionally was performed with statistical and computer vision techniques but recently also some deep learning approaches have been successfully proposed. One of the most used in the field of image denoising is represented by GAN [12] which have been used in different image-to-image translation contexts, showing very good performance. Also for the feature extraction phase there are various techniques. Today the main ones are based on the use of neural networkss [9]. In [10] an overview of the state of

Table 1. Main algorithms and approaches for each OCR phase

OCR Phase	Algorithms and Approaches
Pre-Processing	Binarization, skew correction, filtering, thresholding, compression, thinning [5], GAN [12]
Segmentation	Top-down methods, bottom-up methods, hybrid methods [6, 7]
Normalization	Standard approaches [8]
Feature Extraction	Neural Network based approach [9, 10]
Post-Processing	Rule-based methods [5]

the art of algorithms based on neural networks for OCR is presented, showing how these algorithms are able to achieve the best performance in the context of feature extraction. The study also highlights the impact of the features extracted from these algorithms on the classification task. As described by Suen in [11], the main classes of features are two: statistical and structural. Statistical (like momentum, zoning, crossing, fourier transform and histogram projections) are also known as global features, while structural (like convexity or concavity in the characters, number of holes in the document or number of endpoints) are known as local features. Nowadays, there are a lot of tools that automatically perform these steps with high accuracy, as Tesseract [2] or Google Cloud Vision API [37] however the steps of pre-processing and post-processing are extremely difficult to be generalized, since they merely depend on the data given in input and the expected output. Moreover, in the implementation of a Robotic Process Automation (RPA) systems, text extraction is just one of many issues and the main challenges are understanding the structure of the document and extracting visual entities like tables. In 2013 M. Göbel et al. proposed a first meticulous comparison of the performance of various table identification techniques over the ICDAR 2013 dataset [14]. From 2013 to date, in addition to the enhancement of these deterministic approaches, several new techniques based on machine learning algorithms have been introduced, like a method based on the identification of the horizontal and vertical lines classified through Support Vector Machine (SVM) [15] or methods based on Fast-RCNN (FRCNN) [34] trained on the Marmot dataset for table recognition [35]. Even though these techniques are extremely powerful, they need a lot of data to be trained and generalized. For this reason, besides machine learning techniques, many computer vision techniques based on the Hough Transform (HT) [36], which is used universally today especially thanks to the discovery of its generalized form by Dana H. Ballard [28], are proposed. Its power relies on the fact that this transform does not need training data to be used since it can be applied as a mathematical function. Overall, table extraction techniques are even more effective when combined with document structure identification techniques. Indeed, being able to identify the document template allows one to have a priori knowledge of the structure of the text and this knowledge can be exploited to facilitate the identification of objects within the document. With respect to this, there are several related studies [16–18]. In particular, some deep learning techniques like Siamese neural networks (SNN) have been recently proposed. These models consist of two groups of parallel layers of CNNs that extract features from two distinct input documents and a series of documents that represent the knowledge-base, i.e. the set of possible templates in which the document can be mapped. These algorithms are extremely powerful, because they allow to obtain very high performance even with little training data, thanks to a learning technique called one-shot learning [19].

2.1 Related Works

Extracting information from documents is an active research field and in recent years some works on the topic have been published. For instance, in [31] Vishwanath D. et al. present an end-to-end framework that maps some visual entities (such as tables, printed and handwritten text, boxes and lines) into a relational schema so that relevant relationships between entities can be established. The framework performs image denoising by means of GAN and horizontal clustering to localize page lines. In [32], instead the authors build an invoice analysis system that does not rely on templates of invoice layout, but learns a single global model of invoices that naturally generalizes to unseen invoice layouts. In [33], a framework which makes use of an attention mechanism to transfer the layout information between document images is proposed. The authors also applied conditional random fields on the transferred layout information for the refinement of field labeling.

3 Problem Setting

The goal of this research is to identify a methodology that allows the creation of an RPA system for extracting specific information from documents. The information to be extracted are driven by business needs and vary according to the use case of two pilot companies that furnished the data. The goal of these two companies is that of digitizing the information in order to activate a business process of notarization and supplier management. In order to understand the approach to be pursued, in the following the datasets provided to implement the solution are described.

3.1 Dataset Description

The datasets used to define and test the methodology are composed of a collection of documents from two different companies. In one case, the documents are editable pdf of production sheets (in the following we will refer to this data as dataset A), in the other case the documents are scanned pdf of technical product sheets, invoices and transport documents (we will refer to this data as dataset B). In the first case, the production sheet is composed of different pages each composed of a body in the form of a table that contains several production sub-sheets, each representing an order to be performed to specific suppliers.

Production sub-sheets are identifiable by a set of contiguous populated lines separated by other production sub-sheets by a white line. The goal is to extract all the lines corresponding to each production sub-sheet to automatically activate the process of supplier selection. In the second case, the dataset is composed of three kinds of scanned documents: invoices, with different templates, document of transport with different templates, signed and dated by hand, and technical product sheet in the form of a table that contains specific information about the products (EAN, ingredients, nutrition values, sterility requirements etc). Each document has its own set of information to be retrieved in order to be notarized via block chain.

4 Methodology

By considering the business needs expressed by the partner companies, a methodology that consists of five main steps has been identified. Starting from the document, a first phase of pre-processing is envisaged with GANs, with the aim of reducing the noise of scanned documents. Subsequently, a template identification module based on deep learning models (CNN or SNN) is implemented. Then a module to detect and extract tables is defined, looking for the vertical and horizontal lines within the document (using the HT) to trace the number of columns and rows that may constitute them. Follows the OCR phase (with Google Cloud Vision API) to extract the textual content and the content mapping module to encapsulate the information within a matrix schema. Finally, the last module deals with extracting the required information, looking for patterns defined in advance and depending on the type of document. The goal is to ensure that the methodology can be applied in different industrial environments with respect to the business needs expressed. The individual steps of the methodology are described in detail below.

4.1 Pre-processing: image denoising

The first step of the methodology is represented by the pre-processing phase. Specifically, at this stage the goal is to reduce the noise present in the images of the scanned documents since the quality of the documents influences all the next steps of the methodology. As described in chapter 2, one of the most recent and effective methods to perform image denoising is represented by GAN [12]. GANs are neural networks made up of two parts: a convolutional network called generator, trained to generate synthetic samples from input data, and a convolutional network called discriminator, trained to understand if an image is real or generated. Formally, consider a generative network G that captures the distribution of data and a discriminative network D that estimates the probability that an example derives from the training dataset rather than G. To learn the generator distribution p_g over data x, the generator build a non-linear mapping function of the distribution of the a-priori noise $P_z(z)$ in a space $G(z; g)$. The discriminator $D(x; d)$ produces as output a value that represents the probability that x derives from the training set rather than from p_g. G and D are trained contemporarily: G's parameter are adjusted to minimize $log(1 - D(G(z)))$ and D 's parameter to minimize $logD(x)$ as if they follow a min-max game with two players and value function

$$V(G, D) : min_G max_D E_{x-p_{data}(x)}[logD(x)] + E_{z-p_z(z)}[log(1 - D(G(z)))] \quad (1)$$

According to [24], there are different existing GAN architectures and we propose to compare conditional GAN (cGAN) [13] and cycle GAN [21] since they represent the most recent developments in the field of image-to-image translation.

Conditional GAN GANs can be extended to a conditional model if both the generator and the discriminator are conditioned by extra information y. The conditioning can be done by inserting y both in the generator and in the discriminator as an additional input layer. In the generator the a-priori input noise $p_z(z)$ and y are combined in a joint hidden representation [21]. In this case the value function of the two player min-max game is

$$V(G, D) : min_G max_D E_{x-p_{data}(x)}[logD(x|y)]+E_{z-p_z(z)}[log(1-D(G(z|y)))] \quad (2)$$

Using a model of this type it is possible, by providing the model with images of noisy documents and their respective noise-free, to train the model to produce, given in input an image of a document, the image of the same document but with a reduction of the disorder. To perform the training, it is necessary to have for each input also the target image, i.e. the dataset must be composed of pairs formed by noisy images and their respective images without noise.

Cycle GAN Another extension of GANs, called cycleGAN [20], is that of letting them to learn a mapping function between two domains X and Y, given some training examples $\{x_i\}_{i=1}^{N}$ where $x_i \in X$ and $\{y_j\}_{j=1}^{M}$ where $y_j \in Y$. The cycleGAN model includes two mapping functions $G : X \to Y$ and $F : Y \to X$. moreover two adversary discriminators D_x and D_y are introduced, where the goal of D_x is to distinguish between image $\{x\}$ and its corresponding translated $\{F(y)\}$. The same for D_y with respect to image $\{y\}$ and the corresponding translated $\{G(x)\}$. The goal is twofold: to reduce the opposing losses to align the distributions of the generated images and the distributions of the target images and to reduce the cycle consistency loss to prevent the mapping learned from G and F from being contradictory.

4.2 Document template identification: image classification

The second step of the methodology is represented by the document template identification module. This module aims at defining the template of the input document and this can be accomplished by any image classification algorithm. In our research we compare two kinds of algorithms: a more consolidated one based on CNN [23] and a more recent one based on SNN [19], which has shown high performance even with small datasets thanks to what is called "one-shot-learning".

Siamese Neural Network SNNs are models that aim at recognizing, starting from two distinct inputs, whether they belong to the same class or not. The network is made up of two main phases: in a first phase, two input images are passed through a series of convolutional layers, in order to obtain an embedding of them; between these two output vectors in a second step, a distance measure is calculated. The output is a value that indicates the distance between the two inputs used to classify the images. This learning approach is also called ”one-shot-learning” since it is not necessary to have thousands of documents to carry out training, but works on the features extracted from the CNN that make up the SNN and therefore even a single image, that constitute the comparison sample, for each class is sufficient.

4.3 Tables identification

Once the images have been denoised and the templates identified, the next module aims at locating the structural elements of the document, not only by applying OCR to extract text, but also by identifying and mapping the structural elements of the document, such as tables. This module therefore consists of two blocks: an OCR block for extracting textual content based on a pre-trained OCR tool and a block for identifying the tables for organizing the content based on computer vision techniques.

Google Cloud Vision API As an OCR tool we decided to use the Google Cloud Vision API tool, the documentation of which is reported in [37]. This API performs an analysis of the image layout to segment the areas where text is present. After the general localization phase has been carried out, the OCR module recognizes the text on the specified areas and, consequently, extracts it. Finally, the result is corrected through post-processing techniques based on language models and dictionaries. Most of these steps are carried out through the use of CNNs. The extraction performed by the Google Cloud Vision API OCR module returns the textual content and the organization and position of the content within the image. More precisely, the output produced consists of a dictionary containing a structure divided into a hierarchy of blocks and paragraphs that, at the lowest level, contains the individual words extracted and even single symbols with the coordinates of their position within the image, a confidence score and the detected language.

Hough Transform By analyzing the state-of-the-art for table extraction it emerges that one of the last trends is based on deep learning techniques. However, to train such models a lot of data must be available and often it is difficult to generalize well with public dataset. To overcome this limitation, we follow an unsupervised approach based on the HT. Before applying the HT to detect lines, however, we propose to pre-process the image to highlight the lines contained in it using a computer vision method called Edge Detection [26] that aims at drastically reducing the amount of data to be processed, while preserving the structural information on the contours of the objects. This method uses a convolution mask and its gradients together with two threshold values (upper and lower) that define whether the pixel is accepted as an edge or not. More precisely, the pixel is maintained if the gradient value is greater than the upper threshold value and discarded if it is below the lower threshold value, while if it is in the middle it is maintained if at least one neighboring pixel is above the upper threshold value. Once the information has been cleaned through the Edge Detector, it is passed to the module that deals with identifying lines using an approach based on the HT. As described in [27], the Hough Line Detector is a computer vision techniques that extracts all the lines from the image, considering as lines all the series of pixels in a row that exceed a certain number of pixels and with a maximum value of missing pixels.

4.4 Content mapping

The fourth module of the methodology takes care of mapping the extracted information into a schema that is easily searchable. In order to quickly access the content of the text, we decided to organize it within a dynamic matrix structure. Indeed, if each content is assigned to a cell whose position is known, it is easy to search for the other contents associated with it in the adjacent cells, since their positions can be used in the text search. As mentioned, thanks to the Google Cloud Vision API, not only the textual content of the document is available, but also the organization and position of the content within the image. Thus, to create the matrix structure we used an approach which assumes that in a document there could be different separated tables and each table necessarily has at least one vertical lines. The approach has six steps: 1) Scans all the vertical lines and groups them by considering their ordinates: if lines have overlapping ordinates they are grouped together. 2) For each group of vertical lines add to the group all the horizontal lines that are in the group's range of ordinates. 3) Divide each group into subgroups of directly and indirectly connected lines, where directly connected lines are lines that have a pixel in common, and indirectly connected lines are lines that are not directly connected with each other but are both directly connected with the same line or with a set of indirectly connected lines. Each subgroup thus identified represents a table. 4) For each table, detect the number of rows and columns by taking the number of intersections of respectively the vertical and the horizontal lines with the highest number of intersection points. 5) Create a matrix with the identified number of rows and columns. 6) Fill in the matrix with the text extracted by the OCR tool selecting the proper cell using the coordinates returned and the coordinates of the table identified. This approach generally creates a matrix with more cells with respect to the actual table, since it also consider tables with complex structures (as rows or columns with different number of cells). For this reason, in such cases, text is replicated in different cells of the matrix if these cells correspond to the same cell of the table.

Table 2. Fields of the anchor search pattern.

Field	Description
anchor_regex	Regular expression to search for to find the anchor.
content_position_rows	Relative lines (starting from the anchor position) within which to search for the content.
content_position_cols	Relative columns (starting from the anchor position) within which to search for the content.
content_regex	Regular expression to be applied to candidate cells within which to search for the content; each cell will satisfy the expression or not.
content_lambda	Function to apply to the cell(s) that matched the regular expression in content_regex.
content_dim	Number of cells to keep among those returned.

4.5 Content parsing

The last step of the methodology deals with retrieving the information of interest starting from the generated content matrix. The goal is therefore to define search patterns for each value to be extracted. We defined two types of patterns: anchor and text. The anchor pattern first searches for an anchor, i.e. one or more terms from whose position it is then possible to go back to the actual content. An anchor pattern is formed by the fields described in Table 2. The second type of pattern does not provide for the existence of an anchor, but deals with searching directly within the cells for a specific content that satisfies a certain condition. The fields are similar to the previous case but instead of starting searching from the anchor it starts from the first cell of the matrix. With these two types of patterns it is possible to cover all searches within the matrix and it is therefore evident that the task of retrieving is enormously simplified. Indeed, the reorganization of the content in matrices and the two search patterns defined allows for more complex and flexible searches with respect to simple rule-based systems.

5 Experimental Results

In the following the experimental results of the methodology are presented. For each step of the methodology, the results are distinguished per dataset A (composed by good quality editable pdf of the same type) and dataset B (composed of scanned pdf of three different types) with the aim of verifying that the methodology can be applied to both kinds of dataset.

5.1 Image denoising

To train the GANs algorithms a public Kaggle dataset has been used [22]. The dataset contains pairs of images with and without noise. Images have been reduced to 256x256 crops and divided into training and test with a 80-20 split. The total number of crops used is 436, equally divided in the class with noise and the class without noise. As far as cGAN is concerned, the network architecture is composed of 3 layers of convolutions with ReLu activation function, followed by max-pooling and dropout. As far as cycleGAN is concerned the convolutional network has been created using a U-Net architecture [25]. This kind of network is formed by a series of convolution layers followed by a series of transpose convolution layers to bring the image back to its original size with skip connections. Both the models have been trained using Google Cloud Vertex AI training jobs that leverage a hyperparameter tuning tool based on Google Vizier [29]. In both cases, learning rate, batch size and number of epochs have been automatically selected by the optimization algorithm. To measure the performance of these algorithms, a metric called peak signal-to-noise ratio (PSNR) is used. This measures the quality of a compressed image compared to the original one and is defined as the ratio between the maximum power of a signal and the power of noise that can invalidate the fidelity of its compressed representation. Since many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. The PSNR is defined as

$$PSNR = 20log\frac{MAX\{I\}}{\sqrt{MSE}} \tag{3}$$

Where the Mean Square Error (MSE) between two images I and K is defined as:

$$MSE = \frac{1}{MN}\sum_{j=0}^{M-1}\sum_{i=0}^{N-1}||I(i,j) - K(i,j)||^2 \tag{4}$$

The PSNR obtained for the cGAN model is 8,93db while the one obtained for cycleGAN is 23,245db. Thus, the model based on cycleGAN outperforms the one based on cGAN.

5.2 Image classification

To train the image classification module we used the dataset A and B together. Since the dataset A is composed of just one kind of document and the dataset B is composed of three kinds of documents, the outcome of the model is composed of four classes. The images have been divided into training and test with an 80-20 splits. The models compared to perform template identification are SNNs and CNNs. As far as the SNN is concerned, the two embedding layers placed in parallel consist of 2 convolutional layers with ReLu activation function followed by fully-connected layer. In order to train the algorithm, training samples have been paired randomly to have 500 paired samples of the same class and 500 of different classes. To test the algorithm, all the training images have been used as comparison samples against the test images and, to select the final class, majority voting has been performed. As far as CNN is concerned, the network is composed of 3 different levels of convolution with ReLu activation function, each followed by a max-pooling layer and dropout to avoid overfitting and with a final fully-connected layer. The model has been tested using the same test images of the SNN model. Both the models have been trained using Google Cloud Vertex AI training jobs with automatic hyperparameter optimization. In both cases, learning rate, batch size and number of epochs have been automatically selected by the optimization algorithm. As a comparison metric we decided to rely on overall accuracy (OA). On the four classes, the CNN yielded 93,71% of OA while the SNN model 94,33% of OA. Thus, the SNN model yields slightly better performance, however, its great advantage relies on the fact that, in an industrial environment, this kind of model needs less training data with respect to the CNN, letting it be easier to train and use such a model.

5.3 Table identification

To evaluate the performance of the table detection algorithm based on the HT, we decided to use the number of lines (vertical and horizontal) correctly extracted, undetected, partially identified and exceeding. Furthermore, to understand the actual goodness and usefulness of the pre-processing phase based on cycleGAN, this calculation was done both with and without the application of the cycleGAN model. The results are reported in Table 3.

Table 3. Overall results of table extraction algorithm

Metric	A-noGAN	A-GAN	B-noGAN	B-GAN	All-noGAN	All-GAN
% correct lines	88.5%	88.5%	87%	90%	88%	89%
% exceeding lines	3%	3.5%	0.5%	0%	2%	2%
% undetected lines	5.5%	5%	12%	10%	8%	7.5%
% partial lines	3%	3%	0.5%	0%	2%	1.5%

From the results we can see that the application of GANs does not actually impact the performance of lines detection in the case of dataset A, since these data are software-generated and, thus, perfectly cleaned. Instead, with respect to dataset B, their application enhance the results, showing that the algorithm is able to improve the quality of some scanned documents.

5.4 Information extraction: overall results

Finally, in order to evaluate the overall performance of the methodology, since the final objective is to retrieve specific information from documents, we decided to compare each single field extracted with the target of the extraction. To evaluate the performance of our system we decided to use a metric called Gestalt Pattern Matching (GPM), which assigns a similarity value between two strings S_1 and $S2$, based on the size of the substrings and on the number of matching K_m characters between the strings where the matching characters are defined as the longest common substring (LCS) plus recursively the number of matching characters in the non-matching regions on both sides of the LCS

$$GMP = \frac{2K_m}{|S_1| + |S_2|} \tag{5}$$

As can be seen from Table 4, in the case of dataset A, the extractor has an exact matching both with the use of the GAN and without. This is likely because the pdf is perfectly clean but also because the search task of these documents is simpler and seems to be less influenced by a precise identification of the tabular scheme (on which, however, there is good performance). As for the dataset B, on the other hand, since the search patterns are more complex and the quality of documents lower, the percentage of matching obtained (better in the case of application of GAN, confirming the effectiveness of the pre-processing layer) is 0.81. Overall, our methodology has a 90,5% of performance for the task of information extraction in terms of GPM.

Table 4. Overall results

Metric	A-noGAN	A-GAN	B-noGAN	B-GAN	All-noGAN	All-GAN
% GMP score	1	1	0,765	0,81	0,88	0,905

6 Conclusion and future works

This study presents a methodology based on machine learning for the realization of a general and customizable RPA system based on the type of documents and information to be extracted. After an extensive analysis of the state-of-the-art, we developed a modular methodology that could adapt to different documents in terms of template and content. For each module we tested different options. The identified methodology consists of 5 modules: an image denosing module based on cyceGAN; a document template identification module, based on SNN; an information extraction module based on table identification via HT and on text extraction via Google Cloud Vision API; a custom information mapping module to organize the content into a matrix structure and a query module that extracts the necessary information through search patterns. The methodology has been tested using the GPM score. Overall, the methodology perform well with a score of 0.905 (corresponding to a matching of 90%) and also proved the effectiveness of the image denoising algorithm. Finally, the implemented methodology has been deployed in different industrial environments, with different document formats just by fine tuning the template identification model and by defining the search pattern. Some enanchements can be applied for better generalization, such as using a deep learning approaches to detect tables in the document, thus reducing the error in line identification and, consequently, in information retrieval, or designing some optimization algorithms to keep the complexity of the SNN model low (due to the necessary scan of the images to perform prediction) that could slow down the extraction of information in an industrial context.

References

1. Peanho, C.A., Stagni, H., da Silva, F.S.C.: Semantic information extraction from images of complex documents. Applied Intelligence **37**, 543–557 (2012)
2. Smith, R.: An Overview of the Tesseract OCR Engine. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02 (ICDAR '07) pp. 629–633. IEEE Computer Society, USA (2017)
3. Adobe system Incorporated, PDF Reference, `https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/pdf\_reference\_archives/PDFReference.pdf`. Last accessed 11 Aug 2021
4. Chao H., Fan J.: Layout and Content Extraction for PDF Documents. In Marinai S., Dengel A.R. (eds.) Document Analysis Systems VI. DAS 2004. LNCS, vol 3163, pp 213-224. Springer, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28640-0_20
5. Hamad, K., Kaya, M.: A Detailed Analysis of Optical Character Recognition Technology. International Journal of Applied Mathematics, Electronics and Computers **4**. 244-244 (2016)
6. Kaur, S., Mann, P., Khurana, S.: Page Segmentation in OCR System-A Review. International Journal of Computer Science and Information Technologies **4**, 420-422 (2013)
7. Shinde, A.A., Chougule, D.G.: Text Pre-processing and Text Segmentation for OCR. International Journal of Computer Science Engineering and Technology. **2**(1), 810-812 (2012)
8. Trier, Ø.D. , Jain A.K., Taxt T.: Feature extraction methods for character recognition - a survey. Pattern recognition **29** (4), 641-662 (1996)
9. Shah P., Karamchandani S., Nadkar T., Gulechha N., Koli K., Lad K.: OCR-based chassis-number recognition using artificial neural networks. In: Proceedings of the 2009 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp. 31-34, IEEE Computer Society, India (2009)
10. Rehman, A., Saba, T.: Neural networks for document image preprocessing: state of the art. Artificial Intelligence Review **42**, 253–273 (2014)
11. Suen CY.: Character recognition by computer and applications, Handbook of pattern recognition and image processing, 569-586 (1986)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks. Advances in Neural Information Processing Systems **3**(11), (2014)
13. Zhang, H., Sindagi V., Patel V.M.: Image De-Raining Using a Conditional Generative Adversarial Network. IEEE Transactions on Circuits and Systems for Video Technology **30** , 3943-3956 (2017)
14. Göbel, C., Hassan, T., Oro, E., Orsi , G.: ICDAR 2013 Table Competition. In :2013 12th International Conference on Document Analysis and Recognition, pp.1449-1453. IEEE Computer Society, USA (2013)
15. Thong, H.V., Khuong, N.A., Trinh, L.B.K., Hyung-Jeong, Y., Tuan, A.T., Soo-Hyung, K.: Learning to detect tables in document images using line and text information. In: Proceedings of the 2nd International Conference on Machine Learning and Soft Computing (ICMLSC '18), pp. 151–155, Association for Computing Machinery, New York, USA (2018)
16. Breuel, T. M.: High performance document layout analysis In Proceedings of the Symposium on Document Image Understanding Technology, pp. 209-208 (2003) Understanding Technology, 2003.

17. Hamza, H., Bela¨ıd, Y., Bela¨ıd, A.: A case-based reasoning approach for invoice structure extraction. In Proceedings of the Ninth International Conference on Document Analysis and Recognition 2007, pp. 327–331. IEEE Computer society (2007)
18. Schulz, F.,Ebbecke, F., Gillmann, M., Adrian, B., Agne, S., Dengel, A.: Seizing the treasure: Transferring knowledge in invoice analysis. In Proceedings of the 10th International Conference on Document Analysis and Recognition 2009, pp. 848–852, IEEE Computer Society (2009)
19. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition, ICML Deep Learning Workshop. vol. 2 (2015)
20. Zhu, J.Y., Park, T., Isola, P., Efros, A.: Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 2242–2251 (2017)
21. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, (2014)
22. Kaggle, Denoising Dirty Documents, Dataset Competition, `https://www.kaggle.com/c/denoising-dirty-documents/data`, Last accessed 11 Aug 2021
23. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition, **77**, 354-377 (2018)
24. Sharma, M., Abhishek, V., Vig, L.: Learning to Clean: A GAN Perspective In: Carneiro, G. You, S. (eds.) Computer Vision - ACC Workshops, ACCV 2018, LNCS vol. 11367. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-21074-8
25. Ronneberger, O., Fisher, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells W., Frangi A. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, LNCS, vol 9351. Springer, Cham 2015. https://doi.org/10.1007/978-3-319-24574-4_28
26. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698
27. Heikki, K., Petri, H., Lei, X., Erkki, O.: Comparisons of Probabilistic and Non-probabilistic Hough Transforms. In: Eklundh JO. (eds) Computer Vision — ECCV '94. ECCV 1994. LNCS, vol 801. Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0028367
28. Ballard, D.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition **13**,111-122. (1981)
29. Golovin, D.,Solnik, D., Moitra, S., Kochanski, G., Karro, J., Sculley, D.: Google Vizier: A Service for Black-Box Optimization. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), pp. 1487–1495. Association for Computing Machinery, New York,USA (2017)
30. Téllez-Valero A., Montes-y-Gómez M., Villaseñor-Pineda L.: A Machine Learning Approach to Information Extraction. In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2005. LNCS, vol 3406. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30586-6_58
31. Vishwanath D. et al.: Deep Reader: Information Extraction from Document Images via Relation Extraction and Natural Language. In: Carneiro G., You S. (eds) Computer Vision – ACCV 2018 Workshops. ACCV 2018. LNCS, vol 11367. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-21074-8_15
32. Palm,R.: End-to-end information extraction from business documents, (2018)

33. Cheng, M., Qiu, M., Shi, X., Huang, J., Lin, W.: One-shot Text Field labeling using Attention and Belief Propagation for Structure Information Extraction. In: Proceedings of the 28th ACM International Conference on Multimedia. Association for Computing Machinery, New York, USA (2020)
34. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images. In: Proceeding of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 1162-1167. IEEE, Japan (2017)
35. Fang, J., Tao, X.,Tang, Z., Qiu, R., Liu, Y.: Dataset, Ground-Truth and Performance Metrics for Table Detection Evaluation. In: Proceedings of the10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 445–449. IEEE Computer Society (2012)
36. Hussein, A., Abdullah, H.: A new Approach for Detection and ExtractionTables in Scanned Document Image using Improved Hough Transform. Engineering and Technology Journal **34**, 738-753 (2016)
37. Google Cloud Vision API Documentation, `https://cloud.google.com/vision/docs`, Last accessed 11 Aug 2021