applied system innovation

MDPI

Article

Business Process Automation: A Workflow Incorporating Optical Character Recognition and Approximate String and Pattern Matching for Solving Practical Industry Problems

Coenrad de Jager * and Marinda Nel

Department of EIT, University of Arkansas at Little Rock (UALR), Little Rock, AR 72204, USA; mxhuisamen@ualr.edu

* Correspondence: cgdejager@ualr.edu

Received: 17 September 2019; Accepted: 18 October 2019; Published: 24 October 2019

Abstract: Companies are relying more on artificial intelligence and machine learning in order to enhance and automate existing business processes. While the power of OCR (Optical Character Recognition) technologies can be harnessed for the digitization of image data, the digitalized text still needs to be validated and enhanced to ensure that data quality standards are met for the data to be usable. This research paper focuses on finding and creating an automated workflow that can follow image digitization and produce a dictionary consisting of the desired information. The workflow introduced consists of a three-step process that is implemented after the OCR output has been generated. With the introduction of each step, the accuracy of key-value matches of field names and values is increased. The first step takes the raw OCR output and identifies field names using exact string matching and field-values using regular expressions from an externally maintained file. The second step introduces index pairing that matches field-values to field names based on the location of the field name and value on the document. Finally, approximate string matching is introduced to the workflow, which increases accuracy. By implementing these steps, the F-measure for key-value pair matches is measured at 60.18% in the first step, 80.61% once index pairing is introduced, and finally 90.06% after approximate string matching is introduced. The research proved that accurate usable data can be obtained automatically from images with the implementation of a workflow after OCR.

Keywords: business process automation; Levenshtein; OCR; Google vision API; Tesseract; approximate- and exact string matching; index pairing; workflow

1. Introduction

In the data-driven age of today it is imperative for businesses to utilize their data in an efficient and timely manner in order to gain a competitive advantage. One of the major problems many organizations face however, is that most of their data assets (±80%) are in a semi- or unstructured format [1–3] resulting in an investment in costly natural language processing techniques in order to digitize the data [4]. The aim of this research is to provide a practical business automation workflow that can be implemented or integrated into an existing system in order to produce accurate digitized image data, while avoiding the possible errors introduced by manual entry of crucial identity attributes of entities [5]. Digitization is the process of turning information into a form that can be read easily by a computer [4]. Many research papers cover the use of OCR (Optical Character Recognition) technologies available on the market today as well as provide comparisons between the various technologies [6–8]. The research conducted for this paper specifically focused on automating, validating, and accurately preparing desired data for use in an existing business process or technology. More detailed discussions about the motivation for the workflow as well as each step built into the workflow will be discussed in the 'Related Work' section below.

The novel full image digitization workflow process as suggested by the authors can be seen below (Figure 1). Each step will be discussed and justified in detail and was designed to overcome the limitations or problem areas of the step before it.

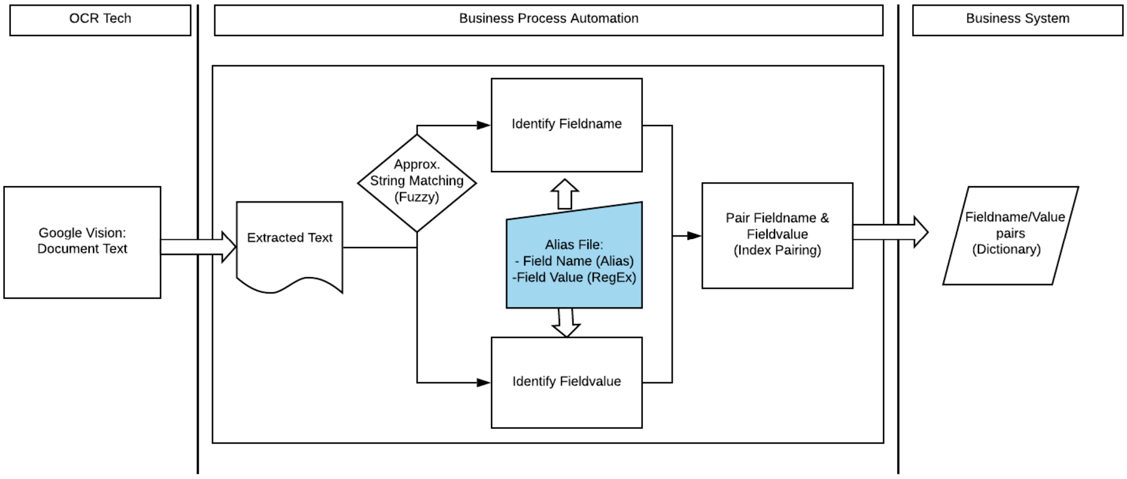

OCR Tech
Business Process Automation
Business System
Google Vision: Document Text
Extracted Text
Approx. String Matching (Fuzzy)
Identify Fieldname
Alias File:
- Field Name (Alias)
-Field Value (RegEx)
Identify Fieldvalue
Pair Fieldname & Fieldvalue (Index Pairing)
Fieldname/Value pairs (Dictionary)

Figure 1. Full image digitization workflow process.

The raw extracted text that is generated from the OCR process forms the starting point of the business process automation. Predefined domain-specific field labels are used in the process of identifying field names from the extracted text. This is similar to the approach of Ford et al. [9] that also only used a dictionary of domain-specific words during pattern matching to narrow down the possible candidates of words to match against. The predefined field labels are maintained in an external document alias file with the corresponding regular expression as a matching field-value. Regular expressions were chosen as a means to help identify candidate values for a field as the values for each key share a common structure but cannot be expressed in a finite dictionary. Key-value matching becomes more complicated where there are multiple string values in the OCR output file that conform to a specific regular expression pattern [10]. The solution to this was to use the field name position on the document and to match the first successful pattern that was located closest to the field name. There is very little literature that takes this approach, as the focus of many of the studies was simply to suggest corrections on a GUI (graphical user interface) [9] or to measure correctness of the OCR tools. Other studies aimed to determine which OCR tools performed the best and give a quantitative result [6–8]. The goal of this study was to produce a working system that generated the key-value pair dictionary for system consumption. To further improve the accuracy of the matches, ASM (approximate string matching) was added to the process in order to ensure that provision was made for small reading errors [11]. Once the program had made the key-value matches, it created a dictionary in JSON format as a final result. With the output structure being a JSON dictionary, any system that can consume such a dictionary would be able to use this process as an initial step in digitizing data.

A key objective was to maximize the amount of correct key-value matches (where the key and its corresponding value were both correctly read from the images and paired together correctly), and to minimize incorrect key-value matches (where a key would be matched with the incorrect value or where a key would be matched with no value). Another objective of the workflow was to standardize and correct any minor spelling mistakes resulting from the OCR failing to recognize a character correctly. Both these objectives contributed to the overall project goal which was to get the most accurate digital dictionary representation of the image file which could be utilized in another application or existing business system by creating a chain of processes that each overcame the limitations of the process before it.

In the next section, related work and the current state of the art solutions will be discussed and compared, and more justification will be provided for using the proposed workflow and steps to solve this problem. The section thereafter will address the implementation details and what steps were taken to attempt the digitalization process. That will be followed by a proposed evaluation technique to measure the success of the proposed workflow. The final sections include the conclusion and future work that discusses the results of the experiment along with future work for the next stages or continued work on this problem.

2. Related Work

2.1. OCR Tools

There are various excellent OCR tools and technologies available on the market today [6–8,12,13] with key distinctions between cloud- or on-premise-based as well as paid or open source. The focus of this research study was not aimed at comparing OCR tools but rather the workflow after the OCR had digitalized information. Many studies mention and address the need to run steps after OCR in order to get better results [9,14]. That is because small fonts or formatted text (like italic text) [9] as well as bad document quality [12] often contribute to characters being misread by the OCR tool. One way that this can be overcome is by document pre-processing techniques. As Wu et al. [15] state, "current OCR...do not work well for documents with text printed against shaded or textured backgrounds", and interest around different ways to pre-process these documents is growing [1,5,8].

An alternative approach to avoid steps after the OCR step is by training the OCR tools on samples of input characters to improve the reading accuracy [12,16]. This attempts to overcome cases where the characters appear different than normal characters—for example, when a new font is introduced, when the OCR characters to be read are handwritten, or when the language to be read contains different alphabetical characters. For this study, the OCR tool picked does not yet provide this functionality and the font is normal printed text, so this was not deemed necessary.

Although previous research studies combine a multi-step approach to enhance the OCR output [9,17] none of the approaches presented offered an automated workflow with minimal manual intervention which could be implemented in the industry to produce matched key-value pairs automatically. Many of the workflows stopped at the point of suggesting possible candidates for the human to analyze and select.

2.2. Combining OCR and ASM

Very little literature exists that explains a specific application and workflow in a business environment that uses OCR functionality alone to convert visual data to text data. Research has however been conducted in the implementation of natural language processing techniques to improve the results after a commercial OCR system has run [9]. Ford et al. [9] address the need to run additional pattern matching steps after OCR had been done on a document in order to get more accurate results from the text. Their findings suggest that OCR does well on normal text but performs poorly on characters that are in italics or in small text. This correlates to the finding in this study that OCR also performs poorly when the document quality is low. They have used pattern matching techniques on the OCR output and used a dictionary of predefined, well known words for the domain of their study. Previous research studies also focused on the improvement of data outputs after image digitization and were used as the base for this research study in order to extend and incorporate their findings and recommendations.

2.3. Combining OCR and Text Processing (Index Pairing)

Packer et al. [12] also found that in some parts of their documents, the original token ordering was not preserved consistently, due to the inability of the OCR engine to separate columns into distinct sections. Some studies have found that OCR tools often do not recognize or process multi-column document structures well, and are not well suited for documents organized in a tabular manner [18]. Two approaches to processing the response from the OCR tool are to do text mining on the response text, and to use the character coordinates that are included in the GCV (Google Cloud Vision) response [18]. This problem was encountered in our study too and was resolved using a Python script: when a key is encountered, a pool of suitable and valid values is identified in its vicinity, but the value located closest to the key is chosen as the key's value, even if the value occurs before the key in the digitized text.

3. Description of the Approach

3.1. OCR Tools

The OCR output files which form the starting point of this study were generated by the use of Google Cloud Vision tool, running in the document text mode. Google Cloud Vision document text detection was chosen as it provided an overall accuracy on the total data set of 89.03% compared to Tesseract (PSM3) which had and accuracy of 71.76% (Figure 2).

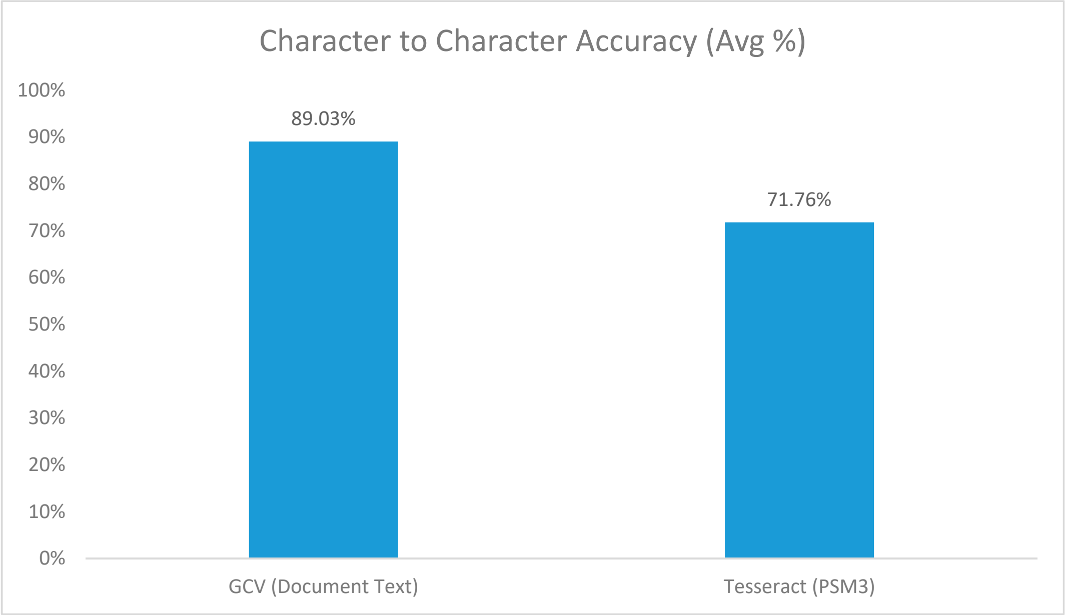


Figure 2. Average accuracy calculation—raw image digitalization across each platform (Google Cloud Vision versus Tesseract).

In order to ensure that we had the best possible starting point for the system that could easily be reproduced for the reader of this study, a quick look at widely available, constantly maintained, and low cost/free OCR tools was done. The main deliverable from investigating the OCR tools was to find the most accurate OCR tool and use its representation of the documents as the baseline for the proposed system. Only some of the best maintained and most widely used OCR tools were considered namely Google Cloud Vision (GCV) [19] and Tesseract [1,20]. It also allowed an unbiased comparison between two OCR tools where one was an "in cloud" and "paid for" service while the other is an "on-premise" and "open source" tool. The tools were each tested on the raw documents with various configurations of both tools [15,19] and the configuration with the best character-to-character accuracy against the truth set was picked for each tool. To measure the accuracy between the OCR tools, the same data set of nineteen images differing in size, dimensions, and type was used and run through each of the OCR tools. The ground truth document was created manually and formed the benchmark against which each of the OCR output files were compared. Figure 2 below illustrates the average accuracy achieved by each of the OCR tools used. The tools used were not optimized or given additional training and may thus not reflect the full capability of each tool.

3.2. Approximate String Matching

Approximate string matching (ASM) involves matching two strings while allowing a level of inexactness, as opposed to exact string matching (ESM). Two strings are said to be matching if they represent the same word, and in ESM this is determined by the two words being spelled in the exact same way. For ASM, a chosen amount of inexactness is allowed to provide for spelling mistakes or variances in spelling of the same word. One of the algorithms used in ASM is measuring the Levenshtein distance between strings. Two strings can be said to match if their dissimilarity (or Levenshtein distance) is below a predetermined threshold [10]. The Levenshtein distance between two strings is the minimum number of character-level edits required to change one word into the other. Character-level edits can include deletions, substitutions, or insertions [21]. Using this less strict version of string matching allows the system to match strings even if they have arbitrary differences which in turn decreases the need for the OCR to be completely accurate when reading from the document. ASM allows for some OCR errors, while still giving us the control to decide how strict we make the match threshold.

3.3. Regular Expressions

Regular expressions are often used to help validate correctly formatted strings or to find strings that match the regular expression from a collection of strings [22]. These regular expressions help to validate and identify strings that are often long and difficult to read to the human eye. Shahbaz et al. [22] state that handling string inputs remains a challenging task due to the complexity of real world data. They also mention that often, due to the lack of formal specifications for fields, human testers and capturers are tasked with manually determining and capturing correct strings (like the values for the tax fields in this project) which are often hard to read.

Regular expressions in this project were used as search patterns to filter out strings that had intrinsic structures defined by the regular expression. Each candidate value that fitted a regular expression for a field key was placed in a pool of possible candidates for the key. In this way, regular expressions aided in narrowing down the possible candidate values for each key, and did so without the cost and time needed to involve human capturers. Values with a rigid structure are often identified well from blobs of characters simply by having a unique structure. Some values are less easily identified by regular expressions due to their relaxed intrinsic structure, for example a company name that can contain any number of letters and numbers, as well as special characters and whitespace characters.

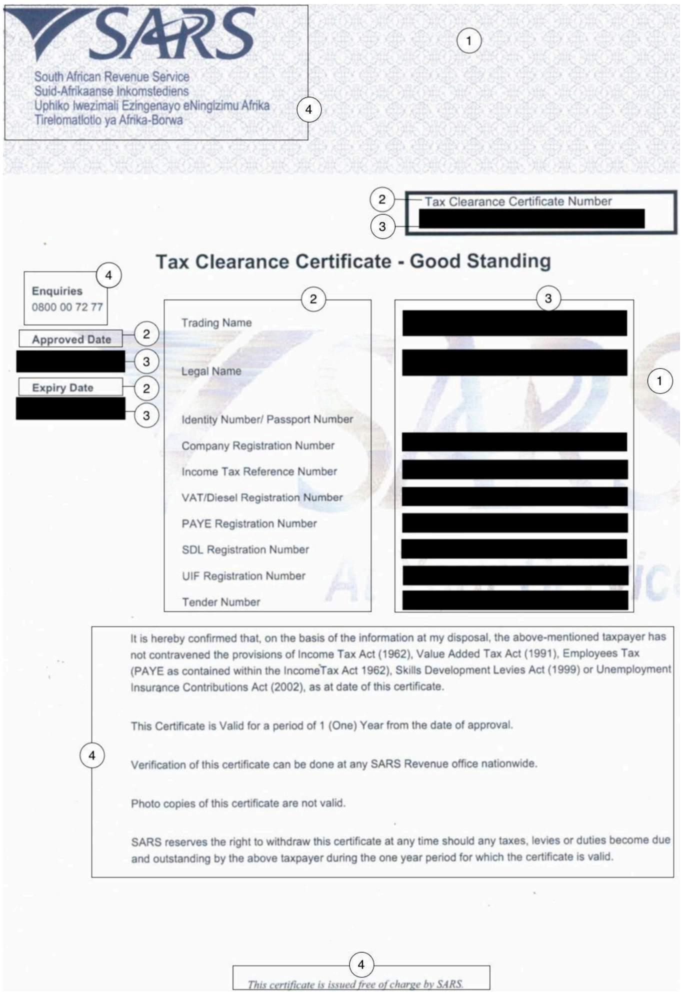
Regular expressions are used to identify specific values from longer strings, as well as classify strings that are not labeled. Many of the tax values of companies have a rigid structure as defined by the governing bodies, and regular expressions can be used to identify possible labels for tax values. This step used the strength of regular expressions to identify and validate the correct candidate from a pool of candidates using its intrinsic structure in a novel way to overcome the need for manual intervention in this process. This was unlike studies that had been done before [9] where the candidate pool was only presented to the human to select the correct value.

4. Implementation Details

The project goal was to implement an accurate business automation workflow that combined available technologies and tools to extract and digitalize data from images in order to create an accurate, universally accepted output file (key-value pairs) that could be ingested by any external system or process.

4.1. Project Input Files (Dataset)

The data input files used in developing the automation workflow were formal, publicly available tax clearance certificates (formal certificates/affidavits) that are normally obtained by registered organizations from the South African Revenue Service. The research decision on which type of image to use in the dataset was based on a typical industry requirement, with the aim of making the process extendible for other similar documents. The dataset consisted of 19 tax clearance certificate images that were scanned copies of the original documents and varied in type (JPEG vs. PNG), quality, and number of fields. An example of the image format is shown in Figure 3. The annotations on the image are explained in Table 1 in order to illustrate the effective data to be derived from the image for key-value matching pairs to be formed.


SARS
South African Revenue Service
Suid-Afrikaanse Inkomstediens
Uphiko lwezimali Ezingenayo eNingizimu Afrika
Tirelomatlotlo ya Afrika-Borwa

Tax Clearance Certificate Number
[redacted]

Tax Clearance Certificate - Good Standing

Enquiries
0800 00 72 77

Approved Date
[redacted]

Expiry Date
[redacted]

Trading Name	[redacted]
Legal Name	[redacted]
Identity Number/ Passport Number	
Company Registration Number	[redacted]
Income Tax Reference Number	[redacted]
VAT/Diesel Registration Number	[redacted]
PAYE Registration Number	[redacted]
SDL Registration Number	[redacted]
UIF Registration Number	[redacted]
Tender Number	[redacted]

It is hereby confirmed that, on the basis of the information at my disposal, the above-mentioned taxpayer has not contravened the provisions of Income Tax Act (1962), Value Added Tax Act (1991), Employees Tax (PAYE as contained within the IncomeTax Act 1962), Skills Development Levies Act (1999) or Unemployment Insurance Contributions Act (2002), as at date of this certificate.

This Certificate is Valid for a period of 1 (One) Year from the date of approval.

Verification of this certificate can be done at any SARS Revenue office nationwide.

Photo copies of this certificate are not valid.

SARS reserves the right to withdraw this certificate at any time should any taxes, levies or duties become due and outstanding by the above taxpayer during the one year period for which the certificate is valid.

This certificate is issued free of charge by SARS.

Figure 3. An example tax clearance certificate image (masked) illustrating the format of the images. Annotations on the image are explained in Table 1.

Table 1. Annotation explanations as illustrated in Figure 3.

Annotation	Description	Data Type	Relevant (Y/N)
1	Watermark/Background noise	String	N
2	Key/Field name	String	Y
3	Value/Field value	Numeric/String/Date	Y
4	Generic text	String	N

As can be seen from the annotations in Figure 3 and the descriptions in Table 1 the tax clearance certificate images contain both relevant and irrelevant fields and blocks of text. All the data from annotations 1 and 4, although available in the digitized image, were ignored for key-value identification and pairing. The key/field names found in annotation 2 are not fixed to what can be seen in Figure 3 but vary between the images in the dataset. Subsequently, the same then applies for the values/field values as displayed in annotation 3 as a field value would be available for the corresponding key/field name.

The digitized output result that needs to be derived from the images is the field names (annotation 2) and corresponding field-values (annotation 3) in a dictionary format, that excludes all of the data that is represented by annotations 1 and 4 in Figure 3.

The keys/field-names are predefined labels of tax characteristics that a company can have. A generated list of such terms had been compiled based on the dataset and was maintained in an external alias file (Figure 1). Values/field-values are the corresponding tax values relating to a specific key. These field-values mostly have a set structure that is successfully representable by regular expressions. Regular expressions were generated by studying the population of tax documents and is also maintained externally in the alias file (Figure 1) where it was subsequently mapped to the corresponding key (Table 2).

Table 2. Alias file containing a sample of field label and field-value (regex) mappings.

Key	Value
Tax Clearance Certificate Number	ˆ[0–9]{4}[/]{1}[0–9]{1}[/]{1}[0–9]{4}[/]{1}[A–Za–z0–9]{10}$
Company Registration Number	[0–9]{4}[/]{1}[0–9]{6}[/]{1}[0–9]{2}
UIF*Registration Number	ˆ[U]{1}[0–9]{9}
SDL** Registration	ˆ[L]{1}[0–9]{9}

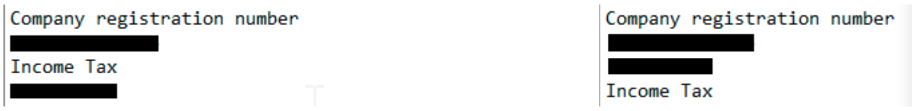
* UIF is an abbreviation for Unimployment Insurance Fund that gives short-term relief to workers who are unable to work due to illness or personal reasons like maternity or adoption leave; **SDL is an abbreviation for a Skills Development Levy that is imposed to encourage learning and development.

4.2. Project Approach

The project is a step-by-step workflow starting with the use of OCR (Step 1) to get raw text from images. Steps were iteratively added to the system (Steps 2,3) to better process the raw text to identify and match the field names and corresponding field-values. Field names and field-values were coupled correctly by the use of various methods to ensure the resulting key-value pair structure accurately reflected the information on the tax documents. The steps identified below were a process of refinement which are discussed individually in order to explain how each of them improved the accuracy of the key-value coupling.

Step 1: Raw OCR output: Initially GCV OCR was used to read the text from the tax documents and return it as a text blob. A ground truth for each tax image file was created manually by mimicking the format returned by the GCV service. Once GCV had returned a text blob the program identified field labels by iterating through each line using exact string matching (ESM) from keywords existing in the alias file. Regular expressions were used for the identification of the corresponding tax values for each key in the alias file. F-measure, precision and recall were recorded for key-value matches when comparing the results to the ground truth. Analysis of the results showed that the most common causes of inaccuracy were due to misread characters and line ordering. The resulting effect was that field labels could often not be matched to any of the values in the text blobs, or the key and values were not in the expected order and thus were not matched together.

Step 2: Introducing index pairing: To improve the pairing results and cater for lines that were read and processed out of order, index pairing was introduced in the place of transforming the raw OCR output files. Index pairing provided a more dynamic solution for image types where the OCR output file format would differ. It often occurred that the OCR output file had a value preceding its matching key in the text file. The observation was also made that a regular expression often matched a few different expressions in the file, especially the field-values with a less distinguishable intrinsic structure. Due to the layout of the documents, the true pairs of keys and values were located close to each other on the document itself, and it followed that the correct value for each key was often close to the key, even if it is not adjacent to it. The system was refined to choose a value candidate for each key out of the pool of candidates that matched the regular expression, based in part on its proximity to the corresponding key (index pairing). Figure 4 illustrates the key and value fields being read out of order at times.



```
Company registration number
[redacted]
Income Tax
[redacted]
```

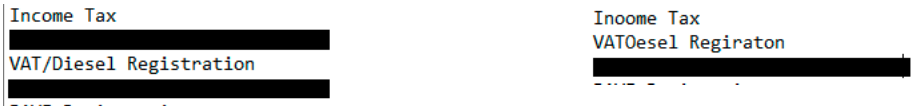
(**a**)

```
Company registration number
[redacted]
Income Tax
```

(**b**)

Figure 4. (**a**) An extract from the truth set and (**b**) an extract from corresponding OCR output. Note how the fields are read out of order in certain scenarios.

Step 3: Introduce Approximate String Matching (ASM): In order to decrease the impact of reading inaccuracies by the OCR, ASM was introduced to the workflow instead of ESM to identify field names. Figure 5 shows the typical case study for using ASM. Levenshtein was used to measure the distance between two strings and set the acceptable threshold to 90% similarity for a string to be deemed equivalent to another. After two strings were deemed to be equivalent, the corrected string was used in the resulting dictionary to ensure the keys produced by the system all matched one of the predefined tax characteristics in the alias file.



```
Income Tax
[redacted]
VAT/Diesel Registration
[redacted]
```

(**a**)

```
Inoome Tax
VATOesel Regiraton
[redacted]
```

(**b**)

Figure 5. (**a**) Example extract from the truth set and (**b**) corresponding fields in the raw OCR output. This illustrates how the field names are misread by OCR in certain scenarios.

5. Evaluation Approach

Project Metrics

The main metrics were the precision (P), recall (R), and F-measure (F) of the match results between key-value pairs in each step of the life cycle, as compared to the truth set. The calculations for each metric used are defined and shown in Table 3 and discussed below [19].

(a) True positive matches (TP): The number of times the system matched a key and its correct, corresponding value together.

(b) False positive matches (FP): The number of times a key was matched with a value other than its actual value as found in the truth set. Reasons for this can include:

    1. The field name and value are present in the tax image file and the system detects the field name through the system. However, the system fails to detect the correct value for the field due to some spelling errors in the value.
    2. The field name and value are present in the tax image file and the system detects the field name. However, the incorrect value is paired with the field name.
    3. The field name has no corresponding value in the original tax document, but the system matches this field name with some value in the tax document.

(c) False negative matches (FN): The amount of times a key or value was not identified. This could be due to a few different reasons:

1. There is a key-value pair in the original tax document, and the system identifies the key successfully. However, it fails to identify the value from the document and so concludes that the key has no value.
2. There is a key-value pair in the original tax document, but the system fails to identify the key or the value.

Table 3. Calculations for Precision, Recall and F-Measure (Talburt et al. [3]).

Metric	Calculation
Precision (P)	P = TP/(TP + FP) = TP/L
Recall (R)	R = TP/(TP + FN) = TP/E
F-Measure (F)	F = (2 × P × R)/(P + R)

The metrics are measured at three points in the system (Figure 6), to show the effect as new improvements are introduced to the workflow. The results and comparisons are discussed below.

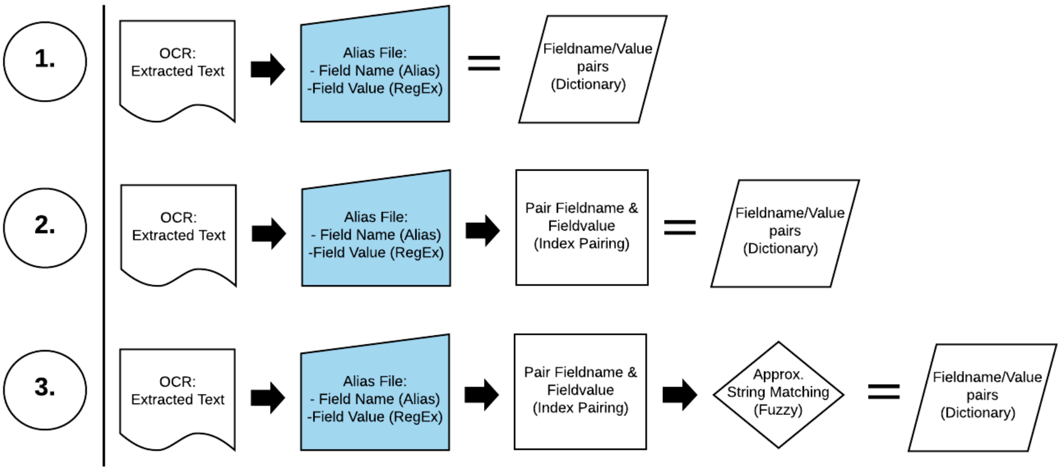

1.
OCR: Extracted Text
Alias File: - Field Name (Alias) -Field Value (RegEx)
=
Fieldname/Value pairs (Dictionary)
2.
OCR: Extracted Text
Alias File: - Field Name (Alias) -Field Value (RegEx)
Pair Fieldname & Fieldvalue (Index Pairing)
=
Fieldname/Value pairs (Dictionary)
3.
OCR: Extracted Text
Alias File: - Field Name (Alias) -Field Value (RegEx)
Pair Fieldname & Fieldvalue (Index Pairing)
Approx. String Matching (Fuzzy)
=
Fieldname/Value pairs (Dictionary)

Figure 6. Accuracy measurement points in workflow to show improvements as new functionality is introduced.

6. Results and Comparison

The detailed results for each metric at each step in the workflow were documented across the population of 19 files. Averages calculated from the detailed results are shown in the graphs below.

As can be seen in Figure 7, a moderate number of matches were made initially, but only 56.34% of those matches were correct. This is to be expected as a specific value could often satisfy a few of the regular expression patterns. The system had no intelligence about which one of the many values that match the regular expression for a key was the correct one and often simply selected the first one it could find after finding a key. After the introduction of index pairing, the precision grew considerably as not only more matches but also more correct matches were made. The increased number of matches was because the system now had the ability to read values above and below the key to find potential matches that would match the regular expression. The fact that there were more correct matches was because the irregularity that the OCR tools sometimes showed of placing a value above its key was overcome—the index pairing step addressed the instances where the OCR tool read lines out of order. A correct value could now be found for a key regardless of whether the key or value occurred first in the text. Finally, using ASM to find the keys in the files increased the precision further. Not many more matches were made, but the matches made were correct much more frequently. ASM thus assisted in overcoming the instances where OCR misread a word due to noise or document quality issues.

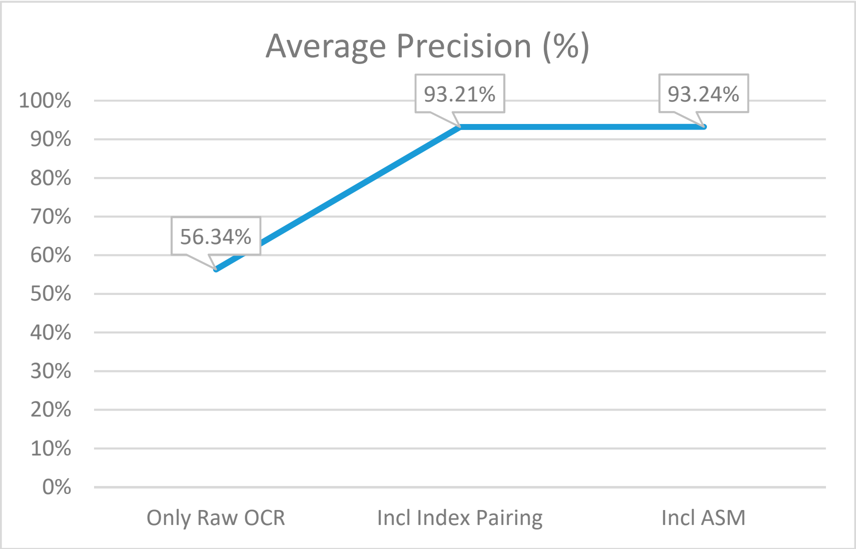


Figure 7. The average precision percentage at each of the three points of capturing the metrics.

In the raw OCR step the system identified 68.7% of the true matches. As can be seen in Figure 8, the introduction of index pairing (Step 2) increased the recall, showing that the system had a better ability to find the actual matches from the pool of possible matches. Again, this was due to the disorganized way in which the keys and values were read by OCR. Finally, the recall increased dramatically and almost all the true pairs were identified from the data text.

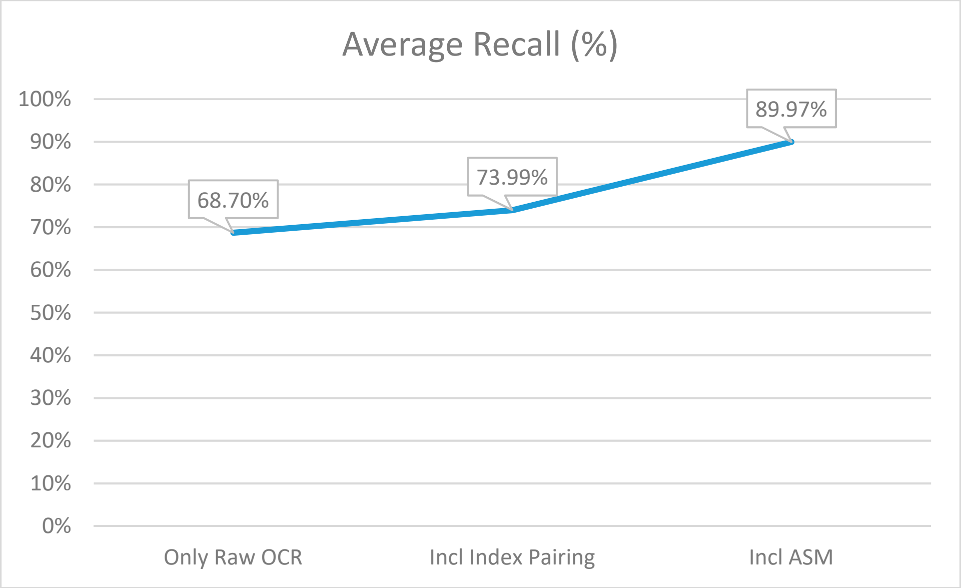


Figure 8. The average recall percentage at each of the three points of capturing the metrics.

As can be seen in Figure 9, the F-measure significantly increased after index pairing (Step 2) was introduced to the workflow and further significantly improved once ASM was added. The precision, recall and F-measure conclusions are well supported by the above graphs.

Figure 10a shows a dramatic increase in the amount of true positive matches from the start point of the system to the end point. The matches made were increasingly more correct by overcoming the flaws of the OCR tools one by one—first by reorganizing the way the document output was read, and then by allowing some level of misread characters to be processed correctly. Just like the precision graph, the average shot up from the first step to the second, and then had another less dramatic increase. This trend in combination with the dramatic drop of false negative matches (as seen in Figure 10b) contribute to the overall shape of precision and recall.

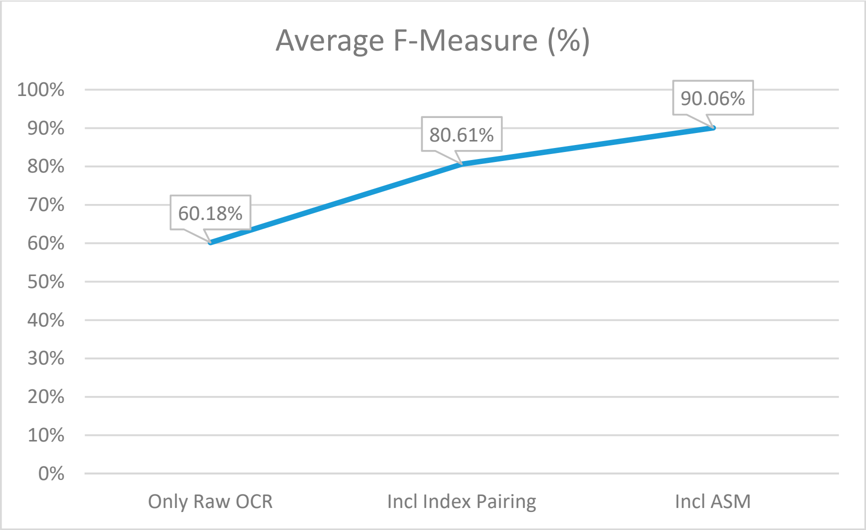
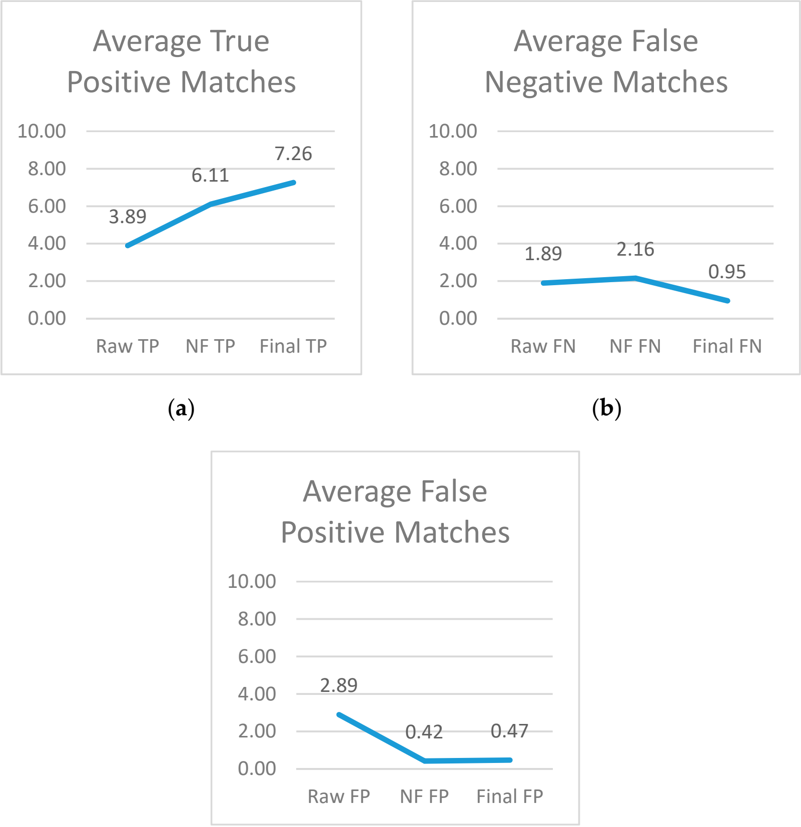


Figure 9. The average F-measure percentage at each of the three points of capturing the metrics.



(**a**) (**b**)

(**c**)

Figure 10. (**a**) The average true positive matches at each measuring point in the system, (**b**) the average of false negative matches at each measuring point in the system, and (**c**) the average of false positive matches at each measuring point in the system.

7. Conclusions

In conclusion, the research confirms that extremely accurate results can be obtained by introducing an automated workflow when digitalizing image data—either as part of an existing system or in isolation. OCR technology data output accuracy can be drastically increased, verified and validated by minimal human intervention saving time as well as costs. By introducing the workflow, the F-measure for key-value pair matches increased significantly from 60.18% to 90.06% (29.88% increase) when comparing the "raw" key-value pair matches made using only OCR outputs to the results after implementing the workflow. (Figure 9) The greatest increase in F-measure was as a result of introducing the index pairing functionality. The improvement in F-measure increased by 20.43% resulting in a total F-measure of 80.61%. By replacing the research OCR tool used with an open source OCR technology (trained to deliver the same accuracy as GCV) companies have the capability to increase the accuracy of image digitalization by more than 29.88% at no cost when implementing and automating data extraction from images.

8. Future Improvements and Focus Areas

In order to improve the accuracy of the current solution, more research time will be conducted on image pre-processing in order to ensure that the quality of images used in the dataset is as high as possible. This will ensure that the solution will provide accurate results should poor quality images need to be digitized. For purposes of the research paper the OCR technologies used were not optimized or trained and were used with standard functionality. Future research will include using a variety of OCR tools as well as experimenting with training the OCR tools in order to produce a digitized output that is as accurate as possible before introducing it to the workflow described in the article. Regular expressions used for field-value identification can be optimized further to produce increased field-value identification. Research would also be conducted in order to investigate the incorporation of field names in the regular expression patterns. The scope of the research was limited to only one type of formal document to prove the concept, even though the solution design is dynamic enough to extend it to other images as well. The solution would thus be applied on other images in order to expand on its use as well as to possibly add new concepts and methodologies to the workflow.

Author Contributions: Both authors worked closely together on the project and article and contributed equally to the design of the experiment, the testing of the results, execution of the test runs, and writing of the document. The majority of the technical coding work was done as mentioned in the acknowledgements below.

Funding: This research received no external funding.

Acknowledgments: The research team would like to acknowledge the Pilog Group for providing guidance on practical industry requirements in which a research need exists. The research team would also like to extend a special thanks to MaryEtta Morris for the major contribution she provided from a system design and coding perspective. Lastly the research team would like to thank John R. Talburt and the University of Arkansas at Little Rock for the guidance and support that was provided throughout the research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Badla, S. Improving the Efficiency of Tesseract OCR Engine. Master's Projects, San Jose State University, San Jose, CA, USA, 2014.
2. Dhiman, S.; Singh, A. Tesseract vs. Gocr A Comparative Study. *Int. J. Recent Technol. Eng.* **2013**, *2*, 80.
3. Talburt, J.; Zhou, Y. *Entity Information Life Cycle for Big Data*; Morgan Kaufmann: Amsterdam, The Netherlands, 2015.
4. Collinsdictionary.com. Digitize Definition and Meaning | Collins English Dictionary. Available online: https://www.collinsdictionary.com/dictionary/english/digitize (accessed on 15 August 2019).
5. Mithe, R.; Indalkar, S.; Divekar, N. Optical Character Recognition. *Int. J. Recent Technol. Eng.* **2013**, 72–75.
6. Suitter, J.A. Accuracy of Optical Character Recognition Software Google Tesseract. *Think. Matt.* **2015**, *46*. Available online: http://digitalcommons.usm.maine.edu/thinking_matters/46 (accessed on 10 August 2019).

7. Chakraborty, P.; Mallik, A. An Open Source Tesseract Based Tool for Extracting Text from Images with Application in Braille Translation for the Visually Impaired. *Int. J. Comput. Appl.* **2013**, *68*, 26–32. [CrossRef]
8. Prakash Sharma, O.; Ghose, M.; Shah, K.B.; Kumar Thakur, B. Recent Trends and Tools for Feature Extraction in OCR Technology. *Int. J. Soft Comput. Eng.* **2013**, *2*, 220–223.
9. Ford, G.; Hauser, S.E.; Le, D.X.; Thoma, G.R. Pattern Matching Techniques for Correcting Low-Confidence OCR Words in a Known Context. In Proceedings of the Photonics West 2001 Electron, Imaging, San Jose, CA, USA, 2–7 February 2019; 2000; Volume 4307, pp. 241–249.
10. Cinti, A.; Bianchi, F.M.; Martino, A.; Rizzi, A. A Novel Algorithm for Online Inexact String Matching and its FPGA Implementation. *Cogn. Comput.* **2019**, 1–19. [CrossRef]
11. Hosseinzadeh, S. A Fuzzy Inference System for Unsupervised Deblurring of Motion Blur in Electron Beam Calibration. *Appl. Syst. Innov.* **2018**, *1*, 48. [CrossRef]
12. Packer, T.L.; Lutes, J.F.; Stewart, A.P.; Embley, D.W.; Ringger, E.K.; Seppi, K.D.; Jensen, L.S. Extracting Person Names from Diverse and Noisy OCR Text. In Proceedings of the Fourth Workshop on Data analytics in the Cloud–DanaC'15, Melbourne, VIC, Australia, 31 May–4 June 2015.
13. OECD. *Data-Driven Innovation Big Data for Growth and Well-Being (Electronic Resource)*; OECD Publishing: Paris, France, 2015; p. 151.
14. Takahashi, H.; Itoh, N.; Amano, T.; Yamashita, A. A Spelling Correction Method and its Application to an OCR System. *Pattern Recognit.* **1990**, *23*, 363–377. [CrossRef]
15. Wu, V.; Manmatha, R.; Riseman, E. Finding Text in Images. In Proceedings of the Second ACM International Conference on Digital Libraries, Philadelphia, PA, USA, 23–26 July 1997.
16. Nayak, M.; Kumar, A. Odia Characters Recognition by Training Tesseract OCR Engine. *Int. J. Comput. Appl.* **2014**, *975*, 8887.
17. Lasko, T.; Hauser, S. Approximate String Matching Algorithms for Limited-Vocabulary OCR Output Correction. *Int. Soc. Opt. Photonics* **2000**, *4307*, 232–240. [CrossRef]
18. Hashitani. Evaluating Google Cloud Vision for OCR. Slideshare.net. Available online: https://www.slideshare.net/ShinHashitani/evaluating-google-cloud-vision-for-ocr (accessed on 12 October 2019).
19. Google Cloud. Vision AI | Derive Image Insights via ML | Cloud Vision API | Google Cloud. Available online: https://cloud.google.com/vision/#industry-leading-accuracy-for-image-understanding (accessed on 10 August 2019).
20. Patel, C.; Patel, A.; Patel, D. Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study. *Int. J. Comput. Appl.* **2012**, *55*, 50–56. [CrossRef]
21. Levenshtein, V. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Sov. Phys.- Dokl.* **1966**, *10*, 707–710.
22. Shahbaz, M.; McMINN, P.; Stevenson, M. Automatic Generation of Valid and Invalid Test Data for String Validation Routines Using Web Searches and Regular Expressions. *Sci. Comput. Program.* **2015**, *97*, 405–425. [CrossRef]