TU WIEN Informatics

Extracting tabular data from utility value appraisals

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Business Informatics

eingereicht von

Klaus Rirsch, BSc
Matrikelnummer 01425916

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Univ.Prof. Dr. Allan Hanbury

Wien, 1. November 2021

Klaus Rirsch — Klaus Rirsch

Allan Hanbury

TU WIEN Informatics

Extracting tabular data from utility value appraisals

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Klaus Rirsch, BSc
Registration Number 01425916

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Allan Hanbury

Vienna, 1st November, 2021

Klaus Rirsch — Klaus Rirsch

Allan Hanbury

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Klaus Rirsch, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. November 2021

Klaus Rirsch

Klaus Rirsch

Danksagung

Ich möchte mich bei all denjenigen bedanken, die mich bei dieser Masterarbeit unterstützt und motiviert haben.

Zuerst gebührt mein Dank Herrn Prof. Allan Hanbury, der meine Masterarbeit betreut hat. Die hilfreichen Anregungen, die konstruktive Kritik und die Geduld bei der Erstellung dieser Arbeit haben sehr zu meinem Lern- und Studienerfolg beigetragen.

Danke auch an meine Familie, die mir mein Studium erst ermöglicht hat, meine Lebensgefährtin, die viel Verständnis aufbringen musste, und meine Katze der einige Streicheleinheiten entgangen sind.

Abschließend möchte ich mich auch bei meinem Umfeld für die vielen anregenden Diskussionen und der DataScience Service GmbH für die Bereitstellung von Testdaten und Ratschlägen bedanken.

Acknowledgements

I would like to thank everyone who supported and motivated me while writing this master's thesis.

First of all, I would like to thank Prof. Allan Hanbury, who supervised my master's thesis. The helpful suggestions, the constructive criticism and the patience in the preparation of this work have contributed greatly to my learning and study success.

Thanks also to my family, who made my studies possible in the first place, my partner, who had to show a lot of understanding, and my cat, which missed out on a lot of cuddles.

Finally, I would also like to thank those around me for the many stimulating discussions and the DataScience Service GmbH for providing test data and advice.

Kurzfassung

Nutzwertgutachten beinhalten Informationen mit vielen Anwendungen im Marketing und der Analyse von Immobilien in Österreich. Relevante Daten sind vorranging in tabellarischer Form und Dokumente sind als PDF's verfügbar, die aus gescannten Bildern bestehen. In der Arbeit werden regelbasierte Methoden zur Extraktion von Zieldaten vorgestellt, und deren Ausgaben werden mit der eines kommerziellen Produkts verglichen. Für den Vergleich wird eine Probe von Nutzwertgutachten verwendet, die auch dazu dient eine Ontologie für Zieldaten zu erstellen.

Das Ziel der Arbeit war herauszufinden, ob regelbasierte Systeme, die ohne vorklassifizierte Datenbestände auskommen, bessere Resultate als eine moderne Deep-Learning-Anwendung liefern können. Precision und Recall wurden als Maßstäbe in den Bereichen der Erkennung von Tabellen, ihrer Struktur, und ihres Inhalts für drei Extraktionssysteme gemessen und verglichen. Der Entwicklungs- und Verarbeitungsprozess der regelbasierten Systeme, sowie Bereiche mit Verbesserungspotential werden anhand von Beispielen veranschaulicht. Der Einfluss von bestimmten Tabellenattributen auf die Ergebnisse wird anhand eines Modells, das verschiedene Arten von Tabellen repräsentiert, untersucht.

Die regelbasierten Prototypen konnten nur in Einzelfällen bessere Ergebnisse als das kommerzielle Produkt liefern. Im Zuge der Auswertung hat sich herausgestellt, dass Eigenschaften von Tabellen und die Komplexität ihrer Strukturen Einfluss auf die Ergebnisse von Extraktionssystemen haben können, aber auch, dass andere Faktoren, wie das Umfeld der Tabelle, Textformatierung und die Qualität der Scans Herausforderungen für alle untersuchten Software-Lösungen darstellen.

Abstract

Utility value appraisals contain data that have many applications in marketing and analyzing real-estate in Austria. Relevant information is predominantly represented in tabular format and individual documents are available as PDF's containing scanned images. Rule-based methods for extracting certain target data are proposed and their output is compared to results from a commercial product. A sample of utility value appraisals is used for ground-truthing and to derive an ontology for relevant data.

The aim was to find out whether heuristics that do not rely on the availability of labelled data-sets can outperform a modern Deep-Learning approach. Precision and Recall were used as measurements in the areas of Table-Recognition, Table-Structure-Recognition and Character-Recognition for the performance of three extraction systems to determine the answer. Examples are used to describe development and processing steps as well as to highlight areas for improvement based on the output of the different approaches. The impact of different table attributes on extraction results is examined using a model for representing different types of tables and a sample of utility value appraisals.

Even though the prototypes did manage to outperform the commercial product in some cases, it achieved better results overall. We found that the format of a table and its complexity can impact extraction results, but that other factors like scan quality, the environment of a table and text formatting also have significant impact on all software artefacts that were examined.

Contents

CHAPTER 1

Introduction

Table extraction already garnered a lot of attention from the scientific community due to its importance for digitization, open data and business intelligence. Prior efforts produced many models, methods and algorithms that can be used to extract tabular information from documents [SL15][CZ17]. The focus of research currently lies on domain independent solutions like Neural Networks and Deep Learning, although rule-based extraction systems are still widely used in the industry [CLR13] [WBM18].

Utility value appraisals are compound documents containing images, tables and text, with a typical sequential flow of information. They are a legal requirement for ownership transfer and contain information that could be used to analyze or market real estate in Austria. While their content is regulated by Austrian law (WEG §9)[1], their layout is not. Currently they can be obtained as scanned images from district courts, agents, property managers and data brokers. There are no extensive public data sets for utility value appraisals and excerpts from the land register require payment of a processing fee.

Initial tests showed that utility value appraisals still provide a challenge for available digitization tools. Many approaches for extracting data from PDF's rely on the existence of metadata, which are not present in scans. The quality of the obtainable images varies greatly and while some scans are nearly perfect, most contain flaws. Those flaws range from skewed elements, to curved text and noise that makes some of the content unintelligible.

Different appraisers use different templates, which constitutes another problem for information extraction systems (IE's). Those templates contain different types of tables that require flexible extraction methods.

Using available software would require extensive post-processing before the results can be used to enrich master-data for the evaluation, sale and analysis of real estate in Austria.

[1] https://www.ris.bka.gv.at/eli/bgbl/i/2002/70/P9/NOR40080362, 2021-11-30

Freeware is either not capable of handling batches of images, or only recognizes certain types of tables. Some proprietary tools are highly accurate and are able to deal with almost any layout, but they need to be configured extensively to obtain specific target data, are expensive and result in the dependency on vendors for projects using them.

The main issue is that domain independent tools extract all tabular data while the amount of relevant (for the purposes of linked open data) data within utility appraisals is small compared to the entire tabular content for this specific type of composite document.

1.1 Approach

1.1.1 Research Question

The research question to be answered is:
"How can domain dependent rule-based methods achieve higher Precision and Recall than domain independent machine learning approaches when extracting tabular data from utility value appraisals?"

The focus of this research is to produce a problem solving software artefact. The domain knowledge gained in each research step and development cycle will be applied continuously resulting in a design theory for extracting certain target data from utility value appraisals.

Target data is specifically tailored to facilitate use-cases ranging from the evaluation of real estate in Austria, to their sale and promotion. It was part of the tender for this thesis, that was issued by our partner company which seeks to apply the results practically.

The research problem aims at evaluating whether rule-based approaches still have merit. The continuous comparison of existing solutions and different versions of new rule-based artefacts will be used to determine the answer.

1.1.2 Research Framework

Action Design Research is the most fitting framework for our purposes. The way of approaching problems, the role of artefacts, theories, stakeholders, as well as the intended research and evaluation process closely resemble this category of Design Science Research.

The focus of Action Design Research lies on designing new artefacts for real world use cases. Research is conducted in iterative steps, which run in parallel. The iterative parts of our research are the development and evaluation which run in parallel along with the formulation of the design-theory (table model and extraction rules).

Theories are part of artefacts that address a class of problems. Evaluation and design are done concurrently as well as iteratively [Sei+11].

1.1.3 Research Process

The first step is to derive design theories from the different presentation formats that actually occur in utility value appraisals. Extraction rules resulting from these theories are then combined with current state of the art methods and implemented by prototypes. The performance of these prototypes is measured against existing domain independent solutions, until the research question can be answered.

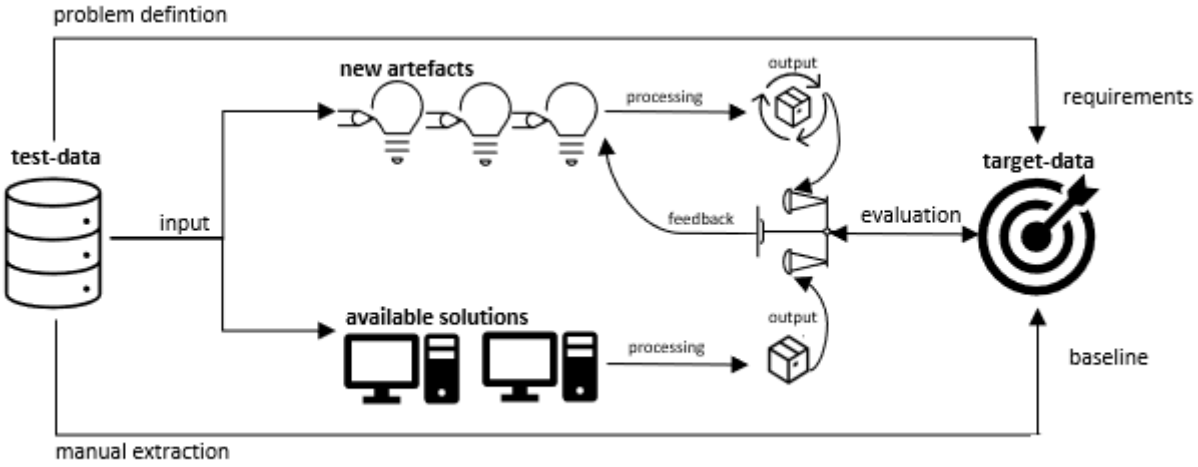

Figure 1.1: Process-Overview

Selection and annotation of test data

There is no prior public research on the topic of extracting specific data from utility value appraisals and excerpts from the land register are subject to fees, so there are no publicly available test data sets. Representative test data will be selected manually from documents that have been bought by our partner company and made available for our research. The selection process will be documented and aims at capturing the variety of different presentation formats and scan qualities while keeping the set small enough to facilitate detailed evaluation processes.

Ground-Truthing

Since establishing the ground truth regarding Table-Detection, Table-Structure-Recognition and Character-Recognition for the entire test-batch would have required extensive manual effort, a subset from all available appraisals covering different table types and scan qualities was used instead of the complete test-batch.

The extraction of target data, on the other hand, was evaluated using the entire test-set and comparing it to the results of manual extraction. In order to establish a baseline for the evaluation, target data was extracted manually.

This process produced a list with one row per appraisal for the purposes of quantitative evaluation. The rows contain the property address along with the count of apartments, housing units and business premises as well as parking lots within the property. The resulting table was then compared to the automatically extracted data from our prototypes and other available solutions for table extraction. Besides the manual extraction of target data for the entire test-set, all tabular data was extracted from the sample that has been selected for detailed evaluation purposes. There are different classifications of tabular data that impact results, therefore we define our understanding of tabular data in Section 2.1.

Artefact Development

Besides providing a baseline for comparisons, handling and categorizing raw data yielded design knowledge about the different presentation formats and commonalities, which were used to derive extraction rules. The results of the structure analysis for utility value appraisals are summarized in chapter 3. Rule sets are based on those insights and current state of the art extraction methods. A prototype capable of running different sets of extraction rules was implemented. It consists of several modules to handle different typical tasks for extraction systems:

– Preprocessing	*remove noise* *deal with scan flaws* *binarize images for computer vision*
– Table Recognition	*morphological transformations* *line recognition* *table detection*
– Table Structure Recognition	*identify headers, rows and columns*
– Data Extraction	*OCR for digitizing text* *filter target data* *organizing result data*

Preprocessing is necessary to deal with the imperfections of scanned images and to provide a machine readable representation for the input. This encompasses Erosion, Blurring and Dilation to deal with noise and to highlight important features like lines and whitespaces for instance as well as the binarization of images. This module is based mainly on best practices in computer vision and requirements for OCR solutions.

Table-Recognition captures the document layout and table regions by different means depending on the current rule set. A variety of algorithms and methods will be tested to achieve the highest possible Precision- and Recall- values.

Table-Structure-Recognition arranges the cells within a table region horizontally and vertically. Headers have to be identified, irregularities need to be addressed (positioning of an item before or after a missing value for instance) and rows as well as columns need to be recognized. Different domain independent approaches, which might be enhanced by domain dependent ones, will be used in this module.

Data extraction will be used to facilitate further automatic data processing. The data in the recognized cells needs to be converted to text first, which will be done using available OCR methods. Filtering and organizing the result data is where the domain dependent rules will have the most effect on the quality of the result. They will be tailored to make the output re-usable and to facilitate business use-cases.

Evaluation

Once the test set was completed, the performance of available solutions was assessed. The purpose was to show, whether they are sufficient to solve the problem at hand and to discover any shortcomings, that need to be addressed for improvements.

The output for different versions of our prototype is compared to the best in market solution for extraction of tabular data. The benchmark was selected during initial testing of available freeware and proprietary solutions available on the internet. Performance is evaluated based on the information retrieval indicators Precision and Recall.

While the evaluation of existing artefacts provides a baseline for expectations, the evaluation of the new artefact will show what can be achieved by domain dependent solutions. We chose Nanonets as benchmark product for our evaluation due to its focus on AI, statistical methods and domain independency as well as its wide base of renowned customers[2]. During our initial tests we could not find any products offering free trials that came close to the quality of its output and are therefore confident, that it serves as an example for the current state of the art.

The results are then used to determine the advantages and disadvantages of our domain dependent heuristic approach and to answer the research question.

The evaluation of different approaches showed the contribution of different rules to the overall results. This might serve as additional proof for accepting our hypothesis. In case we will not be able to surpass the benchmark products, we can at least document how domain dependent methods impact results in our test runs.

Research Goal

The minimal extraction goals are the objects suitable for home ownership and business premises within utility value appraisals. This goal was not extended to their properties and identifying attributes, because the minimal goal was not reached in early stages of the research due to the challenges that are highlighted in the first iteration.

[2]https://nanonets.com/, 2021-11-30

1.2 Thesis Structure

We begin by introducing the two main fields of research indicated by the title of this thesis.
The second chapter - Extracting tabular data - provides a definition for tables, which is used during implementation and evaluation. It also examines the state of the art as well as historic developments, especially with regard to the role of rule-based approaches. Previous works regarding modules of the prototype are summarized to provide reasoning and background for the architecture and distinctions made in the design process.

The third chapter - Anatomy of Utility value Appraisals - analyzes the structure of documents and types of content in the target domain. It shows the typical flow of information in utility value appraisals using examples taken from the test batch. The resulting target data model can be used as a starting point for an ontology of the target domain. It also introduces a table model that distinguishes different types of tables by features that impact recognition and extraction algorithms. Additionally an overview over the test-data is presented and the table model is applied to derive a sample, which was used for the granular evaluation of different software artefacts.

The fourth chapter - Extracting tabular data from Utility value Appraisals - first outlines the modules of the prototypes and the tool-stack used building it. Then the two iterations of development are described and the rules they are based on are explained. While the modules are the same, the approaches for detecting tables and their structure differ greatly. While the first iteration focuses on detecting rows that belong to tables, the second iteration uses a column-based approach.

The fifth chapter - Evaluation Process - begins with the detailed evaluation of the benchmark solution and both versions of the prototype on document-level using a sample of test data provided by our partner company. First the results are presented, analysed and areas for improvement are identified. Then strengths and weaknesses are highlighted by using examples and applying the table model to categorize findings. We also examine the possible application of our results for the intended use-case with the entire test data made available to us to simulate a real world scenario. Since the source code of the benchmark solution is not public, assumptions regarding it are based on its output for a variety of table types in different environments.

The last chapter - Research Results - summarizes the findings of both evaluation steps for all iterations to answer the research question. Finally contributions and opportunities for further research are highlighted and shortcomings as well as limitations of the research are addressed.

CHAPTER 2

Extracting tabular data

Before tabular data can be extracted, it is necessary to distinguish tables from other types of content in compound documents. We first define our understanding of tables, then introduce related fields of research as well current and past state of the art methods. Finally the decision to use a rule-based approach for our research is explained by elaborating the reasoning behind our choice.

2.1 Definitions for Tables

There already are many approaches to defining tables available in different formats like mathematical representations via relations [WH02], physical data models [RSS+15], graphical models [Kum15] and textual descriptions of table features [Cam89].

We decided to use a comparatively old textual definition as starting-point, since it is concise, and sufficient for our purposes by distinguishing tables from other content and highlighting their distinct features:

"A table is an object which uses linear visual cues to simultaneously describe logical connections between the discrete content entries in the table. A content entry is the basic component of information in the table [...] (and) can be any visual symbol" [Cam89].

Visual cues are gaps between table cells, that group them into rows and columns. The resulting structure represents relations between those groups of cells and hierarchies within the table. The position of a given cell carries information about its logical connections to other table entries.

Extracting visual cues does not depend on knowledge about the target domain but rather on the identification of tabular features. Interpreting logical connections on the other hand requires domain dependent knowledge and is based on the extraction of visual cues.

Whether an object is a table or not, is crucial not only for tailoring extraction algorithms, but also for the evaluation process. Visual cues are used as main criterion for table recognition, while logical connections between cells are used for extracting target data.

Examples for distinctions between tables, lists, indexes and other objects can be found in [JT+06] for instance, and are summarized in the following paragraphs, with the goal of establishing what counts as a table for the evaluation of our prototypes amd the benchmark solution.

Here are two objects that show the same data in two different ways. The left representation is a list, while the right one is a table, although they have the same content. The text is aligned horizontally and vertically in both objects, but the integers are aligned differently.

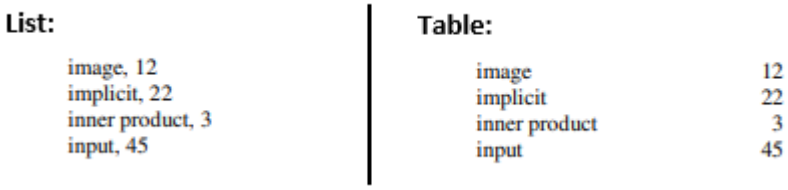

Figure 2.1: Missing visual cue [JT+06]

The interpretation of connections between the textual and numerical parts of each line requires prior knowledge for both objects, but in the first one the visual cue for distinguishing different columns is missing. The lack of common alignment for the integers in the first object makes differentiating it from normal text challenging, because text might also contain similar patterns e.g. like in addresses.

The visual cue in the right representation is the horizontal distance between items in each row. It can be used to classify the data into columns, even though the length of the distance differs for each line, other than in the left representation. The difference is the horizontal alignment of the numerical part. While it could happen that multiple blanks are used for formatting within text-blocks to emphasize a certain point for instance, a repetition of such patterns over multiple rows indicates the existence of a table.

Even though there is a logical connection between the text and integers in the left object of Figure 2.1 as well, it is rather a list than a table since it lacks the visual cues and both requirements need to be fulfilled to identify an object as a table. While Figure 2.1 concerns itself with the importance of separators between columns, Figure 2.2 emphasizes the importance of logical connections between cells of a table:

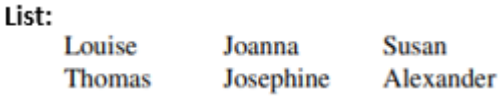

Figure 2.2: Missing logical connection [JT+06]

It is not possible to fathom the relationship between the columns and lines due to homogeneity of content and absence of headers, which establish logical connections explicitly. The object in Figure 2.2 represents a list in a tabular format, but not a table. The example shows that structural analysis is not sufficient to differentiate table-like structures from tables and constitutes an additional challenge for extraction algorithms.

It is noteworthy that headers are not necessary to establish logical connections as long as the contents within each column are different. This can be seen in Figure 2.1, which contains a table without headers, but with two distinguishable types of object that are related to each other.

According to [JT+06] there are two other objects that share common attributes with tables, but still fall into a different category: diagrams and forms.

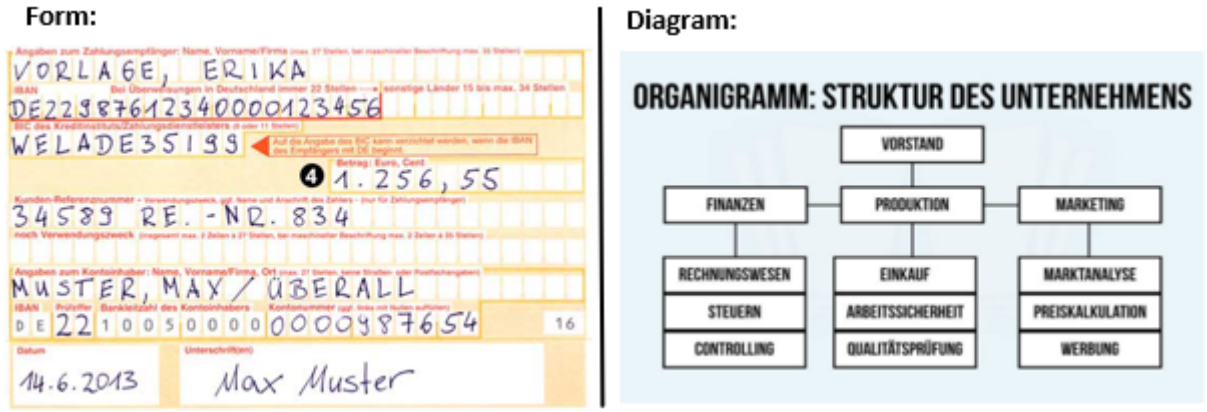

Figure 2.3: Tables compared to other Objects (3) [JT+06] [Gmb] [Mai]

Forms are templates, that are filled with printed and handwritten information that might contain tables. Diagrams show relationships between objects by using line art.

The reason why forms and diagrams are not classified as tables is the same for both object types. Understanding their content is only possible by interpreting line art, the alignment and relative positions of contained objects alone are not enough to determine logical connections.

Whenever additional information either from line art, text, or other shapes in the vicinity of contents need to be interpreted, the container is not a table, but something else.

Relations between the object in the top hierarchy level and the objects in the layer beneath for instance could not be fathomed without the lines connecting the layers in the diagram, whereas in the form relations between the contents can not be established without the separating lines and cell boundaries.

2.2 Related Fields and Methods

There already is extensive research regarding data extraction from digital images, which we aim to summarize broadly to provide a scientific basis for our prototype. Related fields are Computer Vision, Object Detection and Structure Recognition. Extraction methods range from rule-based approaches to Deep Learning and Artificial Neural Networks.

Computer vision is the opposite of computer graphics. While computer graphics generate images from multidimensional data, computer vision aims to reverse this process to extract information [Len98].

Computer vision originated from the field of artificial intelligence and meanwhile became of field of research on its own. Object detection, pattern recognition and image processing, which are required for extracting tabular data from scanned images, are subareas of computer vision. Object detection and pattern recognition distinguish structures and norms from different signals (audio and text), whereas image processing focuses on changing inputs to suit specific use-cases.

Object detection is one of the main tasks of computer vision [Zou+19]. Tables are a specific type of object in compound documents. Identifying table structures alone will not facilitate our use-cases, their content has to be extracted and analysed as well, which leads us to a subcategory of object detection: Optical Character Recognition.

The goal of OCR is to recognize characters to facilitate the extraction of text from images by implementing Machine Learning algorithms. State of the art implementations use different training data for different languages, some of which are made freely available by companies like the Open Source Vision Foundation and Google for instance.

2.2.1 Table Detection

Before tabular data can be extracted from any type of document, they need to be recognized first. The field of Table Detection aims at developing methods and tools to that end, which is determining the position and boundaries of tables in documents.

Research on Table Detection started in the early 1990's with, compared to modern statistical and deep learning approaches, simple rule based methods [Li+20]. Those methods focus on extracting lines, groups of whitespaces and text blocks to locate tables. Examples for early rule based methods are those of [CK93], [SKI94], [RFR94].

The shortcomings of rule based approaches like their domain dependence and the human effort required for inventing them caused a shift of focus to statistical methods in academia. Rule based approaches are, as [KD98] noted, top-down approaches that rely on the existence of separators, while the statistical bottom-up approach is inspired by the way humans recognize tabular data. Bottom-up methods start with single words and build neighbourhoods based on the horizontal distance between text bounding boxes.

Supervised statistical learning methods can be seen as the next historical step in the advancement of Table-Detection methods due to their use of labeled training data. The work of [WH02] for instance aims to distinguish tables from other content wrapped in HTML table tags for layout purposes. To that end they comprised a ground truth database to train table classifiers.

Nowadays different machine learning methods are applied to identify tables, pattern classification and sequence labelling for instance. Artificial Neural Networks and Deep Learning can be seen as the current scientific state of the art approach due to their superior performance on public data-sets. Research on Neural Networks dates back to the 1950's [Gra12], with the intention of resembling the way human brains work according to the understanding of the time, by creating a network of nodes with weighted connections.

Once input is provided to a node it travels along those connections to an output layer with the goal of recognizing relationships within a dataset. Deep learning on the other hand, tries to mimic the way human brains learn by employing specific types of neural networks, like Convolutional Neural Networks.

They provide multiple layers that are being used for decision making and have for instance been applied successfully by [Hao+16] to detect tables. Heuristics still play a role in such decision making processes because they are used to identify potential tables and the evaluation process of Neural Networks. Different approaches like rule-sets and machine learning are combined to extract data from tables and alleviate each other's shortcomings.

2.2.2 Table Structure Recognition (TSR)

Once the bounding region of a table has been identified, the next step is extracting the hierarchical composition of its contents and their functional elements (e.g.: rows, columns and headers) [Hu+00]. The field of study that concerns itself with this task is Table-Structure-Recognition.

The approaches to identify the row and column layout are similar to the ones that are used to detect tables, which are a composition of rows and columns. There are also rule based approaches, like [Zuy97] and [HB07] where extraction methods are based on the categorization of tables depending on the existence of separators, and machine learning approaches based on graphs that represent pages like [Xu+17] and deep learning approaches like [Sch+17] that are data driven and do not contain heuristics.

2.2.3 Role of rule-based approaches

Even though rule-based approaches are currently not in the focus of academic research, they do have advantages over statistical methods [WBM18]. As already mentioned, they have been the first attempts to extract tabular data from documents. When the problem was first addressed in Computer Science, hardware capabilities would not have been able to cope with the processing and storage demands of modern approaches.

While statistical methods are more widely applicable and generally more robust to document variations, they are usually not transparent regarding their decision making and harder to understand and adapt due to their complexity.

Compared to rule-based approaches, domain knowledge is included indirectly via parameterization instead of being expressed explicitly in extraction rules.

Rule-based systems enable transferring and evaluating expert knowledge directly, whereas in statistical methods it is represented by the choice of parameters, algorithms and training data. Statistical methods usually perform better on domain independent data sets, but require acquiring and labelling large data sets, which are not freely available for every domain and do not exist for some specific ones like utility value appraisals.

Rule-based approaches require prior knowledge about the target domain that can be used for the extraction of specific data, which is important in most application areas, since not all data that is represented in tabular format is relevant for specific use-cases. Especially when the extracted data is to be processed further automatically, removing redundant and irrelevant outputs is required to avoid manual corrections and human intervention.

Despite their age, rule-based approaches still have potential to drive innovation and digital business models. We decided to implement new rule-based approaches and explore their potential for extracting tabular data from utility value appraisals.

2.2.4 Fields and Methods - Overview

Figure 2.4 outlines the hierarchical relations between the research fields Computer Vision, Table Detection, Table-Structure-Recognition and OCR:

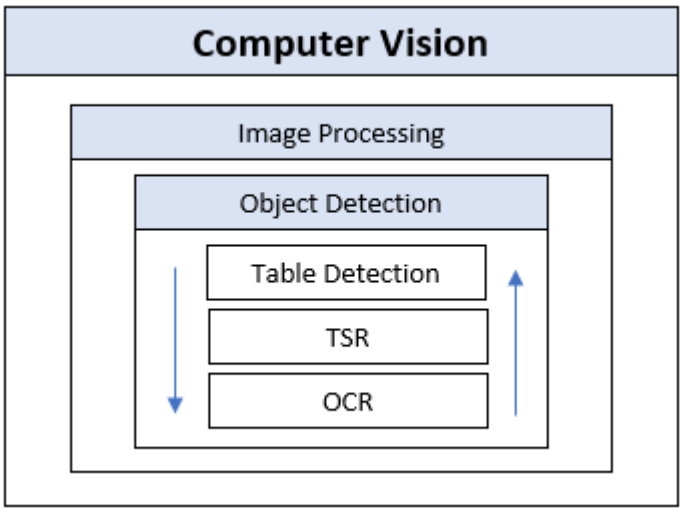

Figure 2.4: Relations between research fields

Rectangles depict containment relations, other fields that might also be contained, but are not relevant for extracting tabular data are left out and the two arrows represent bottom-up and top-down approaches approaches.

In Figure 2.5 the mentioned methodical approaches are aligned in a two dimensional graph relative to each other - the position of an approach is determined by comparison with the others:

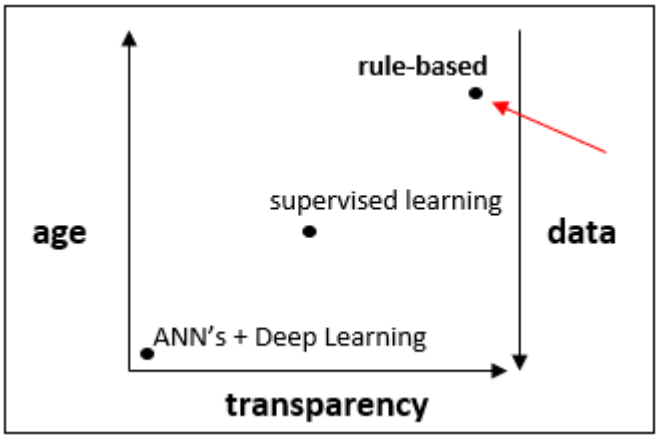

Figure 2.5: Relations between methods

The x-axis represents the degree of black-boxing regarding domain knowledge, while the y-axis represents age and the dependence on training-data.

2.3 Reasons for choosing a rule-based approach

We aim for a reusable and transparent method that is easy to grasp even for audiences without a background in computer science like real estate professionals. To that end we avoid statistical models and do not encode heuristics in parameters.

Acquiring test-data is costly since excerpts from the land register are subject to fees in Austria and labelling a sufficiently big data-set would require extensive manual work, which we aim to avoid in our use-case. Instead we use the time to invent new rules based on the insights gained from manually establishing a ground-truth that serves as a basis for the evaluation of our prototype. Rule-based methods require explicit application of domain knowledge via extraction rules, which also document design theories of extraction systems. This way we do not only formulate a design theory regarding utility value appraisals, but are also able to evaluate it in a transparent way.

Research is currently focused on modern methods like Deep Learning and its related fields. Rule-based approaches do not receive as much attention despite the fact that they are still widely being used in the industry. They suit the requirements of our use-case which is extracting a small proportion of regulated information.
Target data is well defined by Austrian law, as the next chapter will show.

CHAPTER 3

Anatomy of Utility value Appraisals

Before the different iterations of our prototype are described and evaluated, we analyse the target domain to provide a basis for the applied extraction heuristics and rule sets. Document structure and information flow are documented as well as the types of objects that constitute target data for our use-case.

The goal is to build an ontology for utility value appraisals that summarizes the insights gained from exploring and ground truthing the test-batch. Additionally we present and apply a new table model to enable a granular evaluation process for extraction rules.

3.1 Structure of Utility value Appraisals

Even though there is no regulation regarding formatting and structuring the content of utility value appraisals, the test batch shows a distinct flow of information and a list of objects that might be contained within a given appraisal.

General information about the appraised property can be found in the beginning of the document. Afterwards follow data about the units within the property, their type, size, location and other details relevant for the calculation of utility values like reasons for surcharges and deductions.

The first page usually contains information regarding the location of the evaluated property like the address and postal code along with other identifiers for the land register. Property data is most commonly presented as a list (tab-separated, no borders), table, or might also be contained in paragraphs of text, as Figure 3.1.

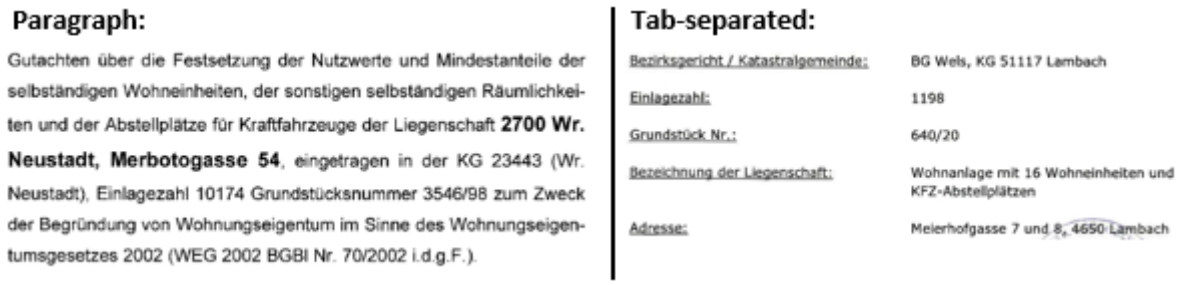

Figure 3.1: Utility value appraisals - PropertyInfo: Examples

Being able to extract this information alone would already facilitate many basic use-cases, because it enables identifying, locating and comparing different properties, although it lacks details about single units and their utility value calculation.

Before the calculation of utility values, appraisals contain a description of the different units contained within the property, that are suitable for home ownership. According to the test batch, there are seven different types of objects, that might occur in utility value appraisals (see PropertyInfo in Figure 3.7).

The type of an appraised object influences the position of information related to it. Common areas are usually listed last, after information regarding the property and its objects suitable for home ownership. They are not part of subsequent utilizable value calculations (being shared property of all owners).

Business premises are usually appraised first and are then followed by houses and apartments, while parking lots and garages come afterwards. As a last part utility value appraisals might contain addons like plans, permits, excerpts from the land register, project reports and other appraisals for construction purposes.

Not every object that is suitable for home-ownership has to be part of a given appraisal, sometimes utility values are calculated only for subsets that are of interest to clients. When analysing the test-batch, we found that many appraisers use a similar layout (see Figure 3.2) to show which objects within the property are subject of the appraisal. This layout is also published in a template document [1] provided by the Austrian government.

wohnungseigentumstaugliche Objekte	Anzahl	neu bewertet	unverändert
Wohnungen	9	4	5
Geschäftslokal	1	0	1
Abstellplätze für Kraftfahrzeuge	22	4	18

Figure 3.2: Utility value appraisals - Example Overview

[1] https://www.wien.gv.at/wohnen/schlichtungsstelle/pdf/berech02.pdf, 2022-08-17

If present this information can be found in the first pages of documents within the test batch depending on the amount of additional information, cover-sheets and the sizes of headers and footers. Besides containing target data this information also shows which objects to look for in the remainder of the document.

The main purpose of utility value appraisals is to calculate and allocate a distribution key for every object that is suitable for home ownership within a property. This key is then used to calculate the share of housing costs for owner associations. The calculation is based on the size of individual units and certain positive or negative attributes that are cause for surcharges or deductions.

Initially the areas and sizes are listed for every object that is part of the appraisal (AreaList). Then the proportional utility values are shown after taking the surcharges and deductions into account (CalcList, ValueList). Some appraisers use certain abbreviations that are explained either in a textual or tabular format (ExplainList). Most appraisals contain a summary that consolidates the relevant data in a separate table (SummaryList).

For Utility value lists there is more variety in the information flow than for object types. Explanations for surcharges and deductions might come before, or after the calculation itself. The information about the sizes of areas and rooms also either comes before the rest of the utility value data, or afterwards.

On the other hand, the utility values are usually displayed after the calculation they are based on. Additionally in most cases there also is a summary containing all relevant information about objects at the end of the appraisals before the addons are listed.

The Figures 3.3, 3.4, 3.5 and 3.6 show an example of the 3 calculation steps and a calculation-summary for an apartment in the test-batch:

							Zubehör lt. § 2/3 WEG
TOP-Nr.	Raum	WNFL [m²]	Terrasse [m²]	Balkon [m²]	Dachterr. [m²]	Summe [m²]	Garten [m²]
TOP 1		31,04	6,30			6,30	
Wohnküche	26,12						
Bad/WC	4,92						
Garten Top 1							9,90

Figure 3.3: Utility value appraisals - UtilityAreaList

Initially the sizes and types of the different rooms are being described. Accessories like gardens and balconies do not count as living space according to Austrian law and are listed separately (see Figure 3.3). They are included since they can be the cause for surcharges and deductions that are applied to the sum of space in eligible areas.

In the next calculation step two deductions are applied (see Figure 3.4) - one due to the fact that the apartment is located on the first floor, and a second one because there is no separate room for the toilet.

TOP-Nr.	NW	Z01 Fläche <50m²	Z02 Terrasse	A01 EG-Lage	A02 Bad+WC			RNW
TOP 1	1,00	10,00		-12,50	-2,50			0,95

Figure 3.4: Utility value appraisals - CalcList

The final step of the calculation consolidates the previous information and separates the available space of the flat in three categories that contribute differently to the utility value. The deductions that have been applied are factored in for the utility values for the living space, while garden and balcony are multiplied with factors according to Austrian law (Regelnutzwerte):

Nr.	Beschreibung	Nutzfl.	Nutzwert	NFL x NW	Nutzwertsummen gerundet		Anteile (verdoppelt)
1	Wohnung TOP 1	31,04	0,950	29,49	29,00	31	62
	Terrassen, Balkone	6,30	0,238	1,50	1,00		
	Garten Top 1	9,90	0,100	0,99	1,00		

Figure 3.5: Utility value appraisals - ValueList

The SummaryList (see Figure 3.6) contains utility values and distribution shares for appraised units. It frequently appears as grid table, regardless of the other table formats that might be used to display more detailed information. SummaryLists contain non property related target data in an aggregated manner, but are not present in every appraisal and might be split into separate parts for each object type.

Zusammenfassung

Geschos NR		Gegenstand	NFL/m²	Anteile	Anteile x 2	in %
EG	1	Wohnung	44,44	38	76	3,48
EG	2	Wohnung	40,34	34	68	3,11
EG	3	Wohnung	48,83	48	96	4,40
1.Stock	4	Wohnung	57,98	53	106	4,85
1.Stock	5	Wohnung	54,86	53	106	4,85
1.Stock	6	Wohnung	52,18	56	112	5,13
2.Stock	7	Wohnung	57,91	56	112	5,13
2.Stock	8	Wohnung	54,89	53	106	4,85
2.Stock	9	Wohnung	51,29	55	110	5,04
3.Stock	10	Wohnung	59,39	57	114	5,22
3.Stock	11	Wohnung	56,28	54	108	4,95
3.Stock	12	Wohnung	53,92	58	116	5,31
4.Stock	13	Wohnung	42,42	39	78	3,57
4.Stock	14	Wohnung	75,49	71	142	6,50
4.Stock	15	Wohnung	53,63	56	112	5,13
1.DG	16	Wohnung	48,11	52	104	4,76
1.DG	17	Wohnung	124,23	134	268	12,27
2.DG	18	Wohnung	107,72	125	250	11,45
			1083,91	**1092**	**2184**	**100**

Figure 3.6: Utility value appraisals - SummaryList

3.2 Objects and Types - Overview

The first pieces of relevant information can be found in the very beginning of utility values appraisals. They are location data of the property, identifiers and an overview of the objects that are contained within. After the property information, the objects contained within are described and categorized according to their type and whether they are subject of the appraisal or not. Most appraisers list different object types in a specific order. Finally the calculation is described in separate steps beginning with the types and sizes of rooms within the appraised objects. Different factors resulting from surcharges and deductions or other regulations are applied, before the resulting utility values are shown for each appraised object.

Not all of the relevant information in utility value appraisals are always presented in tabular format. The ObjectTypes can usually be inferred from the first column (since every single unit is appraised individually) of the UtilityValueInfo, which is apart from very few exceptions always tabular. In many cases ObjectTypes are shown as tables in the first few pages in the appraisal to inform readers about the subjects of the appraisal (this is often done using the template provided by the Austrian government). The first two rows of PropertyInfo on the other hand are as often part of textblocks as they are part of tables. The evaluation of our prototype shows that extraction based on natural language processing as well as table extraction are both necessary to capture all fields of the PropertyInfo reliably.

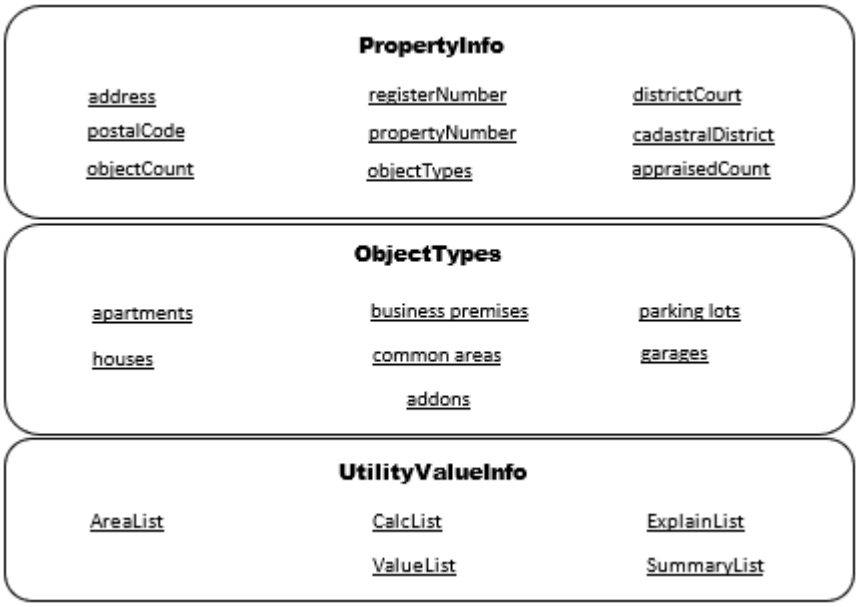

Figure 3.7: Utility value appraisals - relevant data

3.3 Target Data

Once Precision and Recall for Table Detection, Structure Recognition and OCR for both the benchmark product and prototype were measured, we proceeded to evaluate the ability of our prototype to extract target data.

There are limitations due to the fact that appraisals only contain certain information within paragraphs of text, which puts them out of reach for approaches that focus on tabular data only. To reflect these limitations, we narrowed the scope down to the number of units (housing, or business premises) and parking lots per document.

Table 3.1 shows the target output for a sample of documents within the test-batch:

Nr	Address	PostalCode	Units	ParkingLots
1	Merbotogasse 54	2700 Wr. Neustadt	6	14
2	Meierhofgasse 7+8	4650 Lambach	16	16
3	Filzmoos Nr.233	5532 Filzmoos	18	17
4	FRANZ-XAVER-GRIMM STRASSE 14	5110 OBERNDORF	13	26
5	Harterwaldsiedlung 19c/d	8523 Frauental an der Laßnitz	2	4
6	Tigringer Straße 11, Bogenweg 1	9062 St. Peter bei Moosburg	18	24
7	Löck 21	6441 Umhausen	2	3
8	Riedstraße 12	6123 Terfens	2	3
9	Blattur 27	6840 Götzis	2	1
10	Castelligasse 22	1050 Wien	15	0

Table 3.1: Target-Data - Example

The address and postal code require extracting from paragraphs of text, that are filtered while looking for tabular structures and are therefore greyed out. Besides there usually are multiple addresses present in utility value appraisals, the office of the appraiser for instance, which constitutes an additional challenge that can be addressed by Natural Language Processing rather than by Table-Extraction.

Recognizing the number and types of appraised objects is the basis for our use-case, which is the extraction of specific appraisal object data. In future research the scope for target-data can be extended by adding UtilityValueInfo to each object, which would increase the range of possible use-cases and applications.

The data displayed may or may not be part of tables within the appraisals themselves. The number of units and parking lots is either listed within a table, or can be deduced from the extraction results, although that requires a filter procedure on top of the extraction of tabular data, which was also implemented.

3.4 Test-Data and Usage

A batch of 100 utility value appraisals acquired from the Compass Group[2] (one of the leading providers of business data in Austria) serves as input for our prototypes and the benchmark solution. It resembles a real-world problem for extraction algorithms and is not modified before the extraction process.

The batch has a size of 802 MB (original resolution) and contains 2840 scanned pages of appraisal data from 89 different appraisers. Two thirds of all documents are skewed, 6 appraisals contain curved text-lines, one is entirely rotated by 270 degrees and another one contains rotated pages along with non rotated ones.

The oldest document is dated on the 10.02.1970, but the majority (90%) have been composed between 2021 and 2017. While the smallest document consists of merely 2 pages, the largest contains 118 pages with 28.4 being the average page-count.

Within the batch 2274 objects are being evaluated, thereof 971 are apartments, houses, or business premises and 1303 are parking lots or garages.

The prototype that yielded the best result during the granular evaluation phase processed the entire batch documents to assess the quality of the output target data. Due to dependencies between subsequent extraction steps, it became apparent that each step also needs to be evaluated individually. To enable a detailed evaluation process while keeping the effort for manually extracting ground-truth to a minimum, a table-model was used to compose a representative sample of the entire batch.

The resulting test-set was then used to distinguish between domain dependent and domain independent extraction heuristics, as well as to identify the causes and effects of errors that occur before the extraction of target data. Besides it enabled a detailed comparison on page-, document- and table type-level between the domain independent benchmark product and our domain dependent prototypes.

[2] `https://compass.at/de/impressum-agb`, 2022-06-20

3.5 Table Model for Utility Value Appraisals

During the review of the test batch, several attributes and their permutations were identified that distinguish tables from each other and constitute distinct challenges for extraction methods. The rectangles in Figure 3.8 represent categories of attributes, the text on the left are the attributes themselves and the decision gates below the rectangles show possible values:

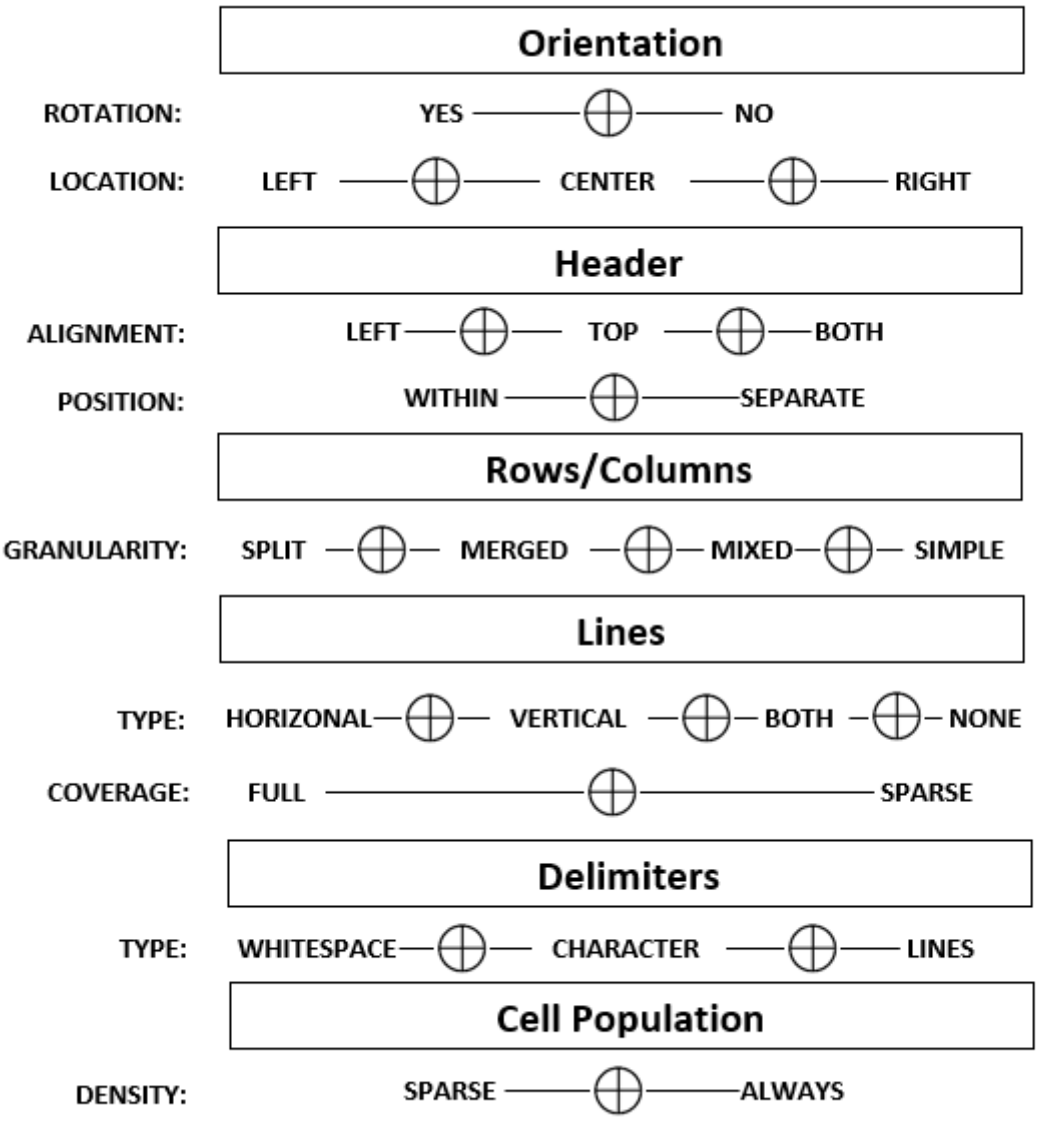

Figure 3.8: TableModel - attributes and values for table classification

Figure 3.9 is taken from the test batch to show the application of the model and the impact of its attributes:

topografische Bezeichnung		Widmung	Regel-nutzwert	Zu- und Ab-schläge Wert	Zu- und Ab-schläge Begründung	Nutzwert pro m²
H1	EG, 1. Stock, Atelier	Haus 1 WNFL	1,00		Regelnutzwert	
						1,000
	EG	Terrasse EG		25,0 %	des Wohnnutzwertes	0,250
	Atelier	Terrassen Ate-lier		25,0 %	des Wohnnutzwertes	0,250
	EG	Vorgarten, Gar-ten				0,100

Figure 3.9: TableModel - Example for table classification

Jagged lines are an indicator for skewed content resulting from issues in the scanning process. Approaches that do not depend on line-art to identify table structures are robust against this type of scan-flaw. The rotation attribute is included in case tables are transposed, or skewed to such a degree that the visual cues become slanted, which requires special treatment compared to table types without significant rotation like the example. Transposed tables need to be recognized to correctly identify headers and logical connections.

In this case the location of the table is in the center of the document, which makes it less likely that there are other objects in the horizontal or vertical range of the table, which could introduce noise during the table detection and structure recognition process.

The header is on top of the content and separated from it by line-art and vertical space. Separation of header and content needs to be taken into account when looking for visual cues, especially in case there are subheadings or other objects between the two that overlap multiple columns horizontally.

Splits occur when there is horizontal distance between the bounding boxes of content in columns, or when there is vertical distance in rows. The first column in Figure 3.9 is an example for a horizontal split and the continuation of words in the next row after a dash (e.g. Ate-lier and Gar-ten in the last two rows) is an example for a vertical split. To correctly assign split content to table-cells, the bounding boxes of all parts need to be merged, although their contours are neither connected horizontally nor vertically.

The merged header-cell in Figure 3.9 is a special case due to the clash of conflicting information from line-art and horizontal location. According to line-art, it is one hierarchy level above two sub-headings, while according to visual cues only above one. We avoid such conflicts by disregarding line-art entirely and focusing on visual cues only.

Lines are still included in our model since one of the prototypes needs to remove them, because they would otherwise obscure visual cues and introduce additional complexity in contour hierarchies.

In Figure 3.9 there are horizontal and vertical lines with sparse coverage because there is no full grid. Vertical lines are only present in the header row and horizontal lines have different lengths.

Delimiters can be characters like dots, lines and whitespace as in the example. This attribute is important, because using characters other than whitespaces for representing visual cues impacts the contours of the table which can lead to it being confused with text and subsequently filtered.

Cell population is important because sparse tables require a more complex approach during Structure-Recognition. In fully populated tables the assumption that the column-index of a cells bounding box corresponds with the row-index holds, while in sparse tables there are gaps which necessitate shifts of the column-index to achieve correct results. Tables that span multiple pages without repeating the headers are especially affected by this since their parts might be identified as individual tables if the extraction only considers one page at a time.

3.5.1 Utilization of the Table Model

The model groups tables into categories, so that tables with different attributes can be treated accordingly in the extraction process and to facilitate sampling our test-data. Table attributes determine extraction heuristics and originate from observations during early development stages. Rotation for example is necessary to correctly interpret tables in landscape format, but would lead to wrong results for tables in portrait format.

Aggregating tables using sets of attributes also enables covering instances in the evaluation that are not part of the ground truth, which is needed due to the limited availability of test data and time for manual extraction and annotation. The inclusion of more than one representative of a category in the test-set could show that the prototypes and/or the benchmark solution perform consistently within categories, but also the impact of additional factors like the environment of a table, which are not part of the model.

3.6 Sample for granular Evaluation

Since target data extraction is the final step, it depends on the success of Table-Recognition, Table-Structure-Recognition and OCR. Any error in previous steps impacts subsequent ones, which necessitates distinctions between steps during the evaluation to identify causes correctly. Other than for target data, which has a comparatively small scope, establishing a ground truth for all steps with the entire test-batch would require extensive manual effort and additional resources.

The annotation of big test-sets is a requirement for Deep-Learning, but not for Rule-based approaches. To reduce the need for manual extraction and comparison, while still covering a wide range of possible inputs, the table model was applied, until three representative documents containing a variety of table types, layouts and scan-flaws were found.

The column "Quality" in Table 3.2 refers to the presence of scan-flaws. "Medium" means there are scan-flaws, that do not hinder human interpretation of the content. Only the oldest document in the batch fell into a lower category, which is why "bad" is not included:

Test-Set				
DocNr	Skew	Pages	Quality	Types
7	YES	30	medium	8
39	NO	9	good	5
80	YES	49	medium	13

Table 3.2: Test-Set - documents and their properties

To measure the impact of table attributes, we classified all tables in the test-set. The sample contains 26 different types of tables, some of which occur multiple times either in the same or across different documents:

TYPE	ORIENTATION		HEADER		ROWS	COLUMNS	LINES		DELIM.	CELL_POP
Nr.	Rotat.	Loc.	Align.	Position	Granul.	Granul.	Type	Cover.	Type	Density
1	NO	CENTER	TOP	WITHIN	SIMPLE	SIMPLE	BOTH	FULL	LINES	FULL
2	NO	CENTER	TOP	SEPARATE	MIXED	SPLIT	H	SPARSE	LINES	SPARSE
3	NO	CENTER	NONE	NONE	SIMPLE	SPLIT	H	SPARSE	LINES	SPARSE
4	NO	CENTER	TOP	SEPARATE	MERGED	SIMPLE	H	FULL	LINES	SPARSE
5	NO	CENTER	NONE	NONE	MERGED	SIMPLE	H	FULL	LINES	SPARSE
6	NO	CENTER	NONE	NONE	MERGED	SIMPLE	NONE	NONE	BLANK	FULL
7	NO	LEFT	TOP	WITHIN	SIMPLE	SIMPLE	NONE	NONE	BLANK	SPARSE
8	NO	LEFT	LEFT	WITHIN	SIMPLE	SIMPLE	BOTH	FULL	LINES	FULL
9	NO	LEFT	LEFT	WITHIN	SIMPLE	SIMPLE	NONE	NONE	BLANK	FULL
10	NO	CENTER	TOP	WITHIN	MERGED	SIMPLE	BOTH	FULL	BLANK	SPARSE
11	NO	CENTER	NONE	NONE	SIMPLE	SIMPLE	BOTH	FULL	BLANK	SPARSE
12	NO	CENTER	TOP	WITHIN	MERGED	SIMPLE	BOTH	FULL	LINES	SPARSE
13	NO	CENTER	TOP	WITHIN	MERGED	SIMPLE	BOTH	SPARSE	BLANK	SPARSE
14	NO	LEFT	LEFT	WITHIN	MERGED	SIMPLE	NONE	NONE	BLANK	FULL
9	NO	LEFT	LEFT	WITHIN	SIMPLE	SIMPLE	NONE	NONE	BLANK	FULL
15	NO	LEFT	LEFT	WITHIN	MERGED	SIMPLE	NONE	NONE	BLANK	SPARSE
16	NO	CENTER	LEFT	SEPARATE	SIMPLE	SIMPLE	NONE	NONE	BLANK	FULL
17	NO	LEFT	LEFT	WITHIN	MERGED	SIMPLE	NONE	NONE	L+B	SPARSE
18	NO	LEFT	LEFT	LEFT	SIMPLE	SIMPLE	NONE	NONE	L+B	SPARSE
19	YES	CENTER	TOP	WITHIN	SIMPLE	MERGED	H	SPARSE	L+B	SPARSE
20	NO	CENTER	TOP	WITHIN	SIMPLE	MERGED	H	SPARSE	BLANK	SPARSE
21	YES	CENTER	TOP	WITHIN	SIMPLE	SIMPLE	H	FULL	BLANK	SPARSE
22	NO	LEFT	LEFT	LEFT	SIMPLE	SIMPLE	NONE	NONE	BLANK	FULL
15	NO	LEFT	LEFT	WITHIN	MERGED	SIMPLE	NONE	NONE	BLANK	SPARSE
23	NO	CENTER	LEFT	LEFT	MERGED	SIMPLE	NONE	NONE	BLANK	SPARSE
24	NO	LEFT	LEFT	LEFT	SIMPLE	SIMPLE	NONE	NONE	BLANK	SPARSE
25	YES	LEFT	LEFT	LEFT	SIMPLE	SIMPLE	H	SPARSE	BLANK	SPARSE
26	NO	LEFT	TOP	WITHIN	SIMPLE	SIMPLE	NONE	NONE	BLANK	FULL

Table 3.3: Table Structure Recognition - Table Types

CHAPTER 4

Extracting tabular data from Utility value Appraisals

Before the extraction process is outlined, the tools used during the implementation phases are summarized. Then both versions of the prototype and their modules are presented. Once the processing steps for each version of the prototype have been described and compared, the rule-sets are presented. Rules have been implemented for the areas of Preprocessing, Table-Recognition, Table-Structure-Recognition, and OCR. They are part of two software artefacts that represent different rule-based approaches for extracting tabular data from PDF-files. The changes between both versions originate from the first iteration of the evaluation process and are based on its findings.

4.1 Tool-Stack

Python was chosen as programming language because it facilitates rapid prototyping and provides open source software to implement best practice computer vision methods.

OpenCV is utilized for standard preprocessing tasks like morphological transformations [SP12] and finding contours. It is open source, widely used and well documented, which should facilitate understanding and adapting the proposed extraction systems.

Each version of the prototype uses different OCR-Tools.
The first iteration uses Pytesseract since it open-source and can be integrated easily. Keras-OCR was chosen due to its ability to detect text in images in the final version, because it simplifies preprocessing steps and helps focusing on Table-Detection and Structure-Recognition.

4.2 Extraction Process for the row-based approach

Figure 4.1 shows the processing steps and intermediate results of the initial prototype:

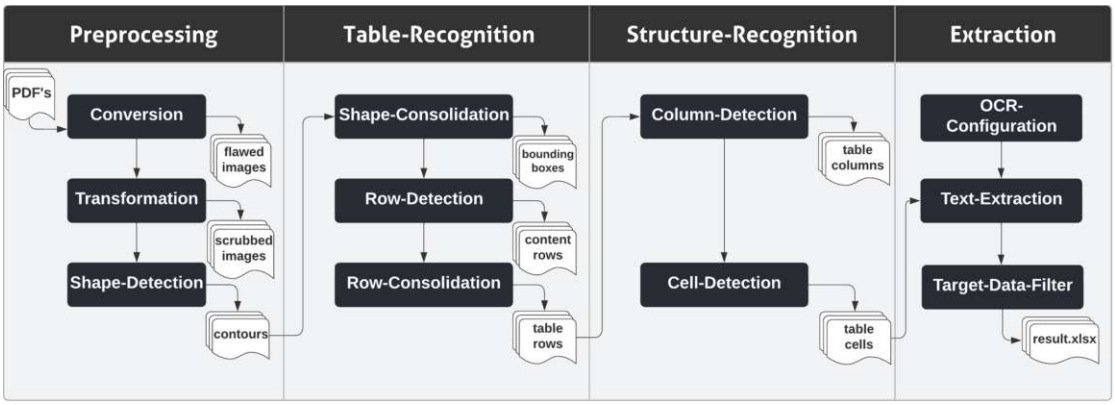

Figure 4.1: Flow-chart of the row-based approach

It begins with converting the input data into separate images. This step is necessary to meet preconditions for computer vision, which only works for image formats. Then transformations are applied to remove noise, rotations and to increase the contrast between objects and background.

Contours are detected after applying morphological operations to simplify the resulting object-hierarchy. Tables are detected by consolidating separate contours once the content has been organized into lines. During this consolidation lines without significant space between contours are removed because they are not likely to be part of a table.

Structure-Recognition rules were applied to group the content within table-areas into separate columns that form a tabular structure in conjunction with the lines that were already established in previous steps. By combining the limits from rows and columns bounding boxes of table-cells are calculated.

These cells are passed to an OCR-tool to extract text. The configuration of the tool does not change during processing. Once the extraction is complete, the table-cells are loaded to an excel-file according to their position in the calculated table-grid, which then serves as input for filtering target data in the last step and the granular evaluation process.

The focus of the initial version was to create a simple extraction procedure as a starting-point for more sophisticated rule-sets. Besides it served to test our table-model as well as assumptions regarding the effects of different table-attributes.

Despite several simplifications that lead to inaccuracies, the first version already showed potential and highlighted advantages of rule-based approaches.

On the other hand, the results of its evaluation showed several key-issues as well, which triggered changes in the extraction process for the second iteration of the prototype.

4.3 Extraction process the column-based approach

Figure 4.2 shows the final extraction process:

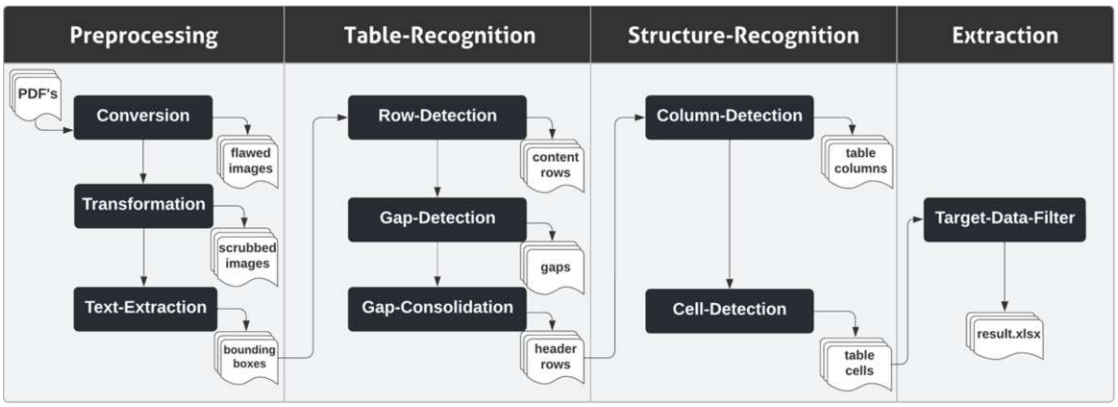

Figure 4.2: Flow-chart of the column-based approach

The conversion process was taken over from the first iteration, because evaluation showed no indication that changes at this stage would improve the quality of the output significantly. Transformations were enhanced by using a different method for handling skewed content. Besides that, many steps like removing lines and morphological operations to deal with noise were omitted completely.

Evaluation showed, that there is too much variety regarding scan-flaws, ranging from lighting issues to additional objects from perforations and ribbons, for simple rule-sets to work reliably. For this reason a Deep Learning tool for recognizing the bounding boxes of text was used.

Originally it was the intention to develop pure rule-based approaches, which did not prove to be practical since OCR is based on machine learning methods in general and constitutes a crucial step in extracting tabular data. Extracting the bounding boxes of text early on proved to be advantageous since it is robust against additional objects and lighting issues, that impacted the first iteration, which was based on contours.

The method for ordering text-boxes vertically to establish rows, was taken over from the first iteration, even though there are issues with objects that overlap multiple others vertically, which does occur in some table headers but was found not to have significant negative impact.

Table-Recognition and Structure-Recognition were inspired by [Fan+11] and are based on recognizing the distance between text-boxes and interpreting them as visual cues.

For that purpose first the space between boxes in each row was recognized and then merged with rows below until either the end of the page is reached or the space is horizontally overlapped entirely by another text-box. Tables are detected by examining all merges of a page and resolving vertical conflicts. Columns are identified by using the remaining merges as boundaries to distribute the content of each row those merges overlap.

Finally the same filter for target data is applied and the results are exported to excel.

4.4 Preprocessing

Preprocessing is the first step for both extraction rule-sets. Before tables can be detected, input needs to be converted to a machine-readable representation. Then scan-flaws, like skews, rotations and noise need to be dealt with to increase the accuracy of computer vision methods. There is a variety of issues within the test-batch, ranging from additional objects along the image-borders to ink bleed and lighting shadows.

4.4.1 PDF-Conversion

The input data for our use-case are PDF's. Since we have to assume that no metadata is available due to the fact that those PDF's consist of scanned images we can do little without converting our data to a format more suitable for computer vision. So the first step is converting the PDF's back to images. There are different suitable data-formats like JPEG, PNG, and TIFF that PDF's can be converted into. JPEG is a format with losses and thus not suitable for our purposes. According to [SL10] there is little difference in OCR performance between TIFF and PNG. PNG was chosen for both iterations since it is open source compared to TIFF which is under the control of Adobe.

The second decision to be made when converting PDF's to image is the target resolution (dots per inch), which impacts processing time, memory consumption, OCR accuracy and shape recognition. Ideally the chosen value should depend on the font-size, because high values for big fonts would lead to noise, while low values for small fonts would not offer enough information to distinguish objects that are similar to each other. We used a fixed value for both iterations to reduce complexity and to be able to focus on table-specific extraction-rules. For the OCR-Tool that was used in the first iteration a resolution of 300dpi is recommended[1], which is also used by the benchmark product[2].

Due to the initial low output-quality for OCR we finally used a resolution of 600dpi because we found that further increasing the value did not provide more accurate results.

[1] https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html, 2022-06-08

[2] https://nanonets.com/help/ocr/what-is-the-minimum-resolution-and-file-size-of-an-image-pdf-for-ocr-recognition-to-work-well, 2022-06-08

4.4.2 Image-Transformation

Both versions of the prototype use Greyscaling and Deskewing, which are standard approaches, that are recommended in scientific literature [BGR07] [Poo14] [CMG96].

Greyscaling provides intensity information that serves as a basis for thresholding algorithms to decide whether a certain area of a picture has content (i.e. is closer to black) or not (i.e. is closer to white).

It also removes color information and generates a two dimensional array containing intensity information regarding shades of grey for each pixel. Color is neither useful for Structure Recognition nor for OCR and inflates the size of pictures since grey values are a subset of RGB.

Figure 4.3 illustrates this process:

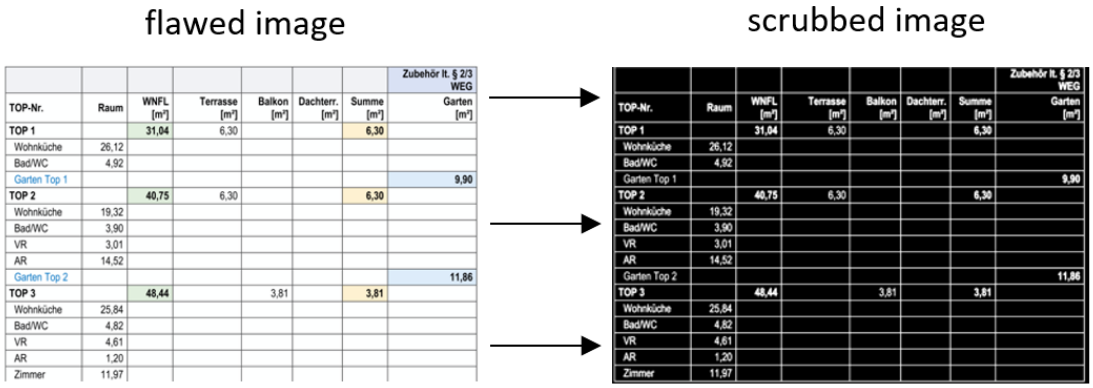

Figure 4.3: Color-Filter

Additionally, it provides a filter for relevant information and alleviates lighting issues by using a threshold, which is determined dynamically by the Otsu-Algorithm [Ots79].

While the first version uses inverse thresholding (white objects on black background), the second version uses white blackground and black objects, which is the standard for OCR-tools.

Figure 4.4 shows how shadows are removed by thresholding:

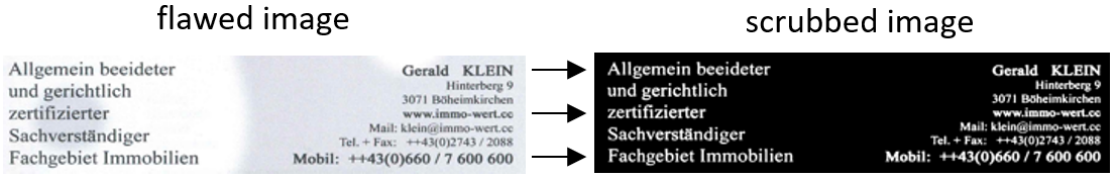

Figure 4.4: Shadow-Filter

The next step is fixing rotations that might have occured during scanning (misalignment of camera, surface not flat, ect.). This step is crucial to establish lines of content since the skew specifies the deviation of text lines from the horizontal or vertical axis [RSS13].

Figure 4.5 provides an example for the correction of rotations:

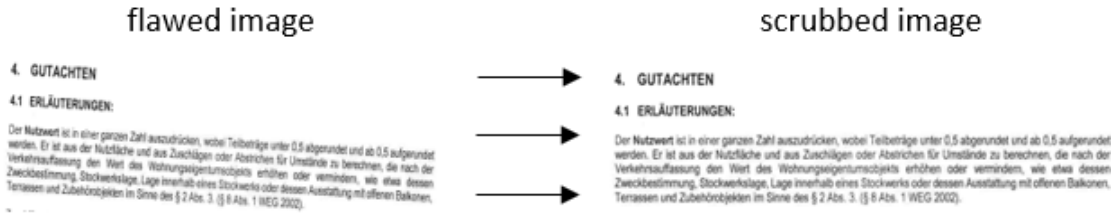

Figure 4.5: deskew - misalignment

For this purpose we used the ImageMagick Library for Python in the first iteration, which is able to deal with common slant-artefacts. Evaluation showed that this approach is not sufficient to deal with pages in landscape format. The final version uses the Hough-Transformation [Hou62] for detecting lines and rotates images according to the angles observed.

Figure 4.6 shows the additional deskew-capabilities of the final version:

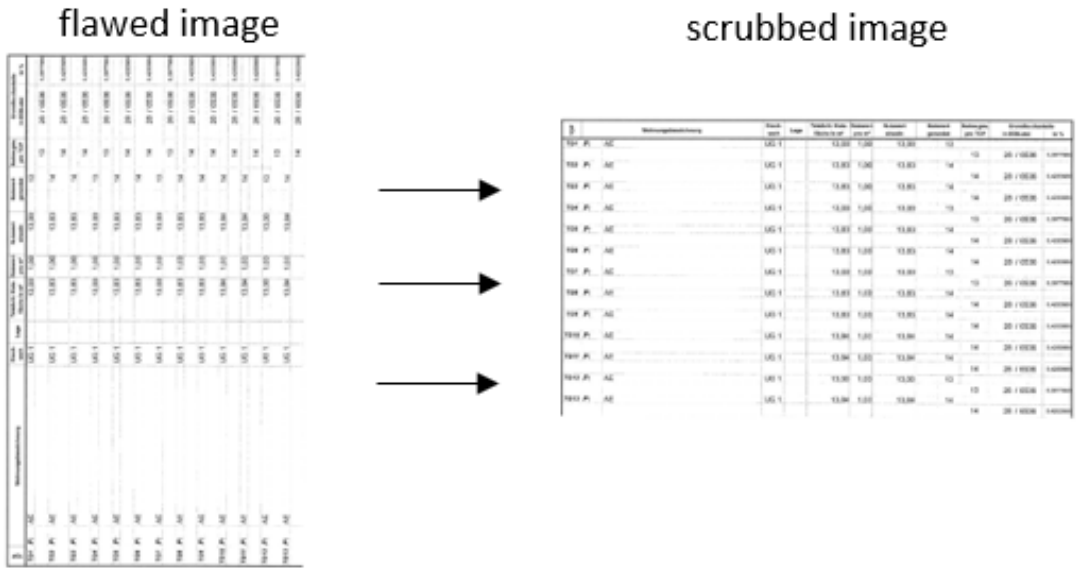

Figure 4.6: deskew - landscape

Neither version of the prototype is able to correct curved text, which might occur when pages are bent to fit on a flat surface. The test-batch of appraisals only contains a handful of such instances since these kinds of issues usually appear when books are scanned.

During the implementation it became apparent, that grid-lines add complexity to the contour-hierarchy. Comparing the external contours from tables without gridlines and tables with gridlines shows, that for the first type the output on the top hierarchy level are the boundaries of the table, whereas for the second type the output are the bounding boxes of the cells the table contains. In order to avoid this complexity, the first iteration attempts to remove all lines to ensure that the external contours always yield the cell bounding boxes regardless of table type.

Figure 4.7 shows the effect of the line removal process:

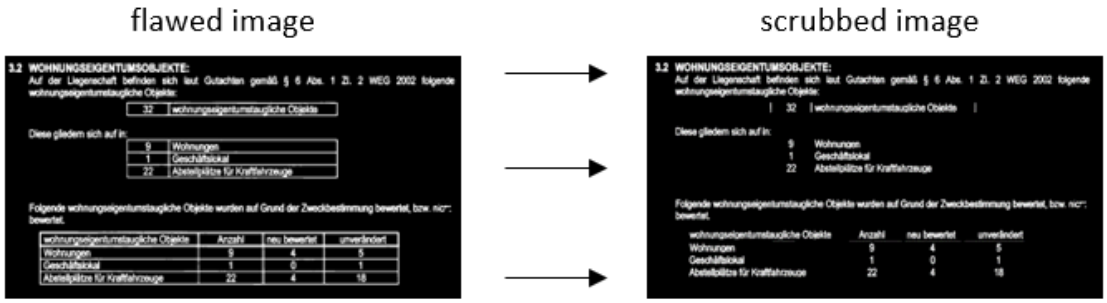

Figure 4.7: line - removal

The approach uses hardcoded limits to detect lines, which is why lines might be overlooked, if they are not significantly longer or wider than characters. The limits were chosen to preserve text. Leftover lines do not add complexity in the contour hierarchy, but they increase the size of bounding boxes. This step is omitted in the final version since it does not depend on contours to detect objects.

4.4.3 Shape-Detection/Text-Extraction

This step greatly differs between both iterations. Initially contours were used as starting-point for table-detection, which proved to be prone to noise from the scanning process. Contours of perforations, cords and other objects lead to inaccuracies in subsequent steps.

Figure 4.8 illustrates the problem of the contour-based approach:

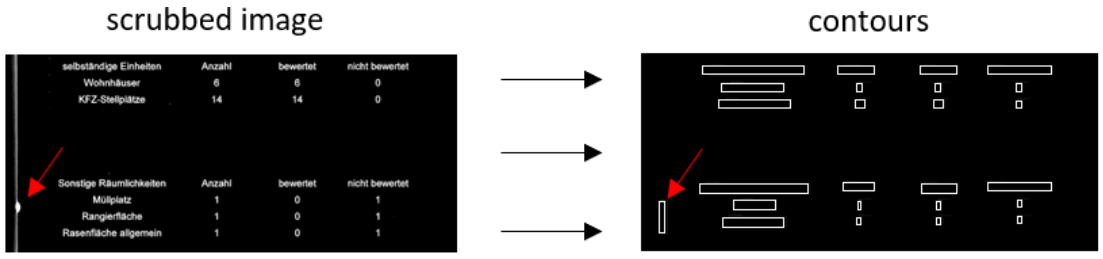

Figure 4.8: noise - issue

Otsu-Thresholding was not sufficient to remove the shadow of the perforation which is subsequently identified as a table-cell. Its shape is similar to characters, so OCR-solutions might wrongly recognize "O" or "0" as content for the false positive.

Another problem for contour-based approaches is grouping characters together correctly to form table-cell contours. In the first version this is done by dilating the content horizontally using hardcoded parameters.

This turned out not to be flexible enough because visual cues for columns that are smaller than the limit are being lost. Headers that were too close to each other have been merged, which leads to errors in Table-Structure-Recognition.

The negative impact of those unintentional merges can be seen in Figure 4.9:

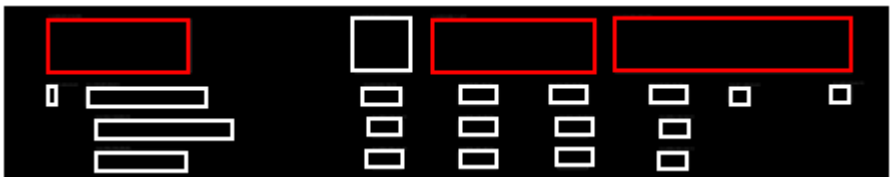

Figure 4.9: merge - issue

In order to alleviate this problem and avoid the complexity of finding parameters dynamically, the final iteration uses Keras-OCR, an open-source Deep-Learning tool, to identify bounding boxes of text.

Figure 4.10 shows that this approach solves several problems in a single step:

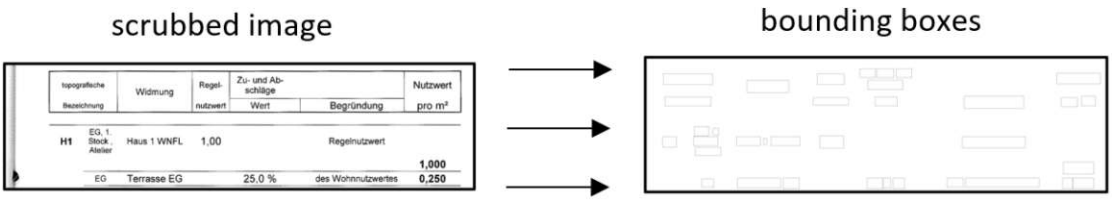

Figure 4.10: noise & merge issue - solution

Noise does not lead to false positives, which negatively impacted Table-Structure-Recognition by introducing additional columns, and characters are already grouped into words without merging content across different table columns.

Recognizing and removing lines is not necessary anymore since they are not part of the output. This simplifies the extraction process and leaves more room for sophisticated rules to identify tables without relying on hardcoded parameters.

4.5 Table-Recognition

This step differs significantly between the two versions of the prototype. The first iteration tries to distinguish rows belonging to text paragraphs from rows belonging to tables. The final version approaches the problem differently by looking for table columns. While the initial version relies on hardcoded parameters for deciding which row belongs to which table, the final version does not. The distinction between text and table content is made by examining rows and their visual cues on an individual basis in the first iteration, whereas the final version looks for visual cues that span multiple rows.

4.5.1 Row-Detection

Once the bounding boxes for text are obtained, the next step is to sort the content vertically to obtain rows. This works the same way in both iterations by first sorting using the left upper corner (y1) and then iterating over the bounding boxes in that order. A new row is created for the first box, or if the left upper corner of the current candidate is below the right lower corner of the previous box. Otherwise the current candidate is added to the current row.

Figure 4.11 depicts this processing step by highlighting the resulting rows:

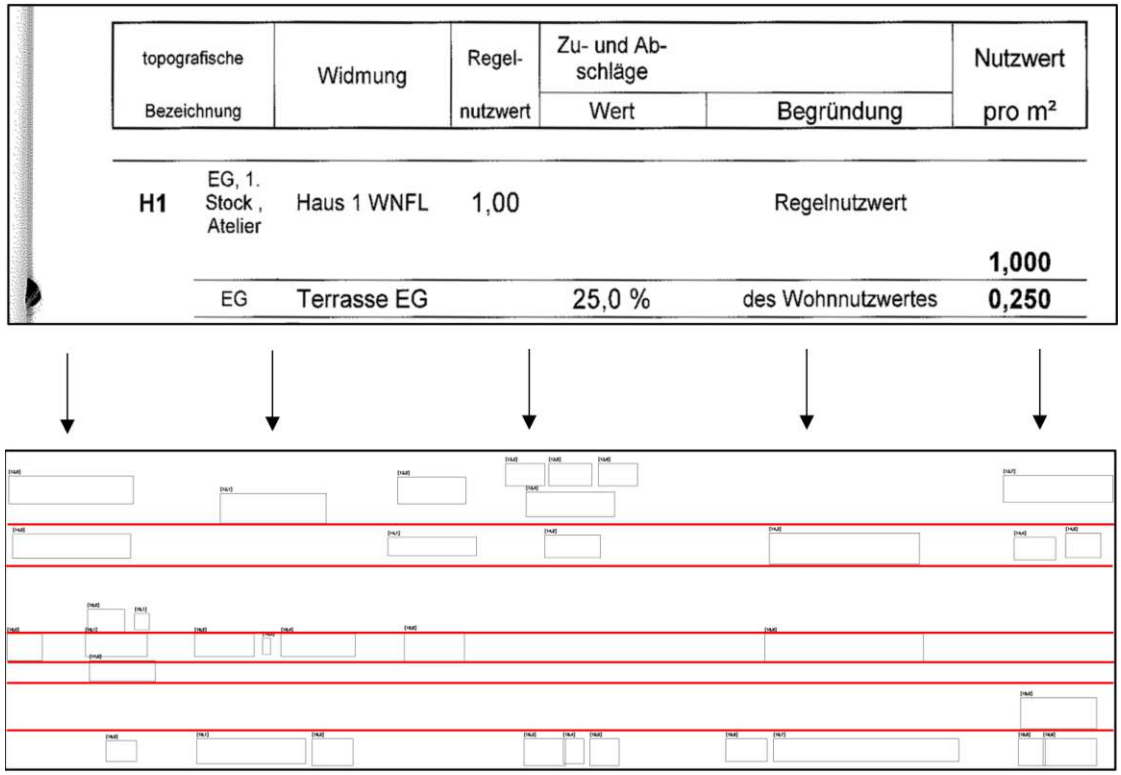

Figure 4.11: row detection

The reason for this rule is that row-contents are not arranged along perfectly straight lines. Objects that belong to the same row might have different y1- and y2- values.

Rows contain all bounding boxes that overlap each other horizontally. Using the highest, lowest or the center point would result in more rows that overlap each other vertically, which constitutes a problem for Table-Structure-Recognition. A drawback is that different bounding boxes within the same row might overlap each other vertically, which needs to be taken into account when looking for visual cues across different rows.

4.5.2 Row-Merge/Gap-Merge

The next step for both versions is detecting tabular structures using the rows obtained previously. The initial version took a comparatively simple approach by merging contours horizontally with different hardcoded values to get a granular and a consolidated view.

The aim was to distinguish text from tabular content by comparing the granular view resulting from a smaller value and the more aggregated view resulting from a higher value. Initially the approach was based on rows and assumed that a row belongs to a table if there is still a visual cue after all interword spaces have been merged.

The red boxes in Figure 4.12 highlight the difference between the contour-, granular- and the consolidated view for the example picture:

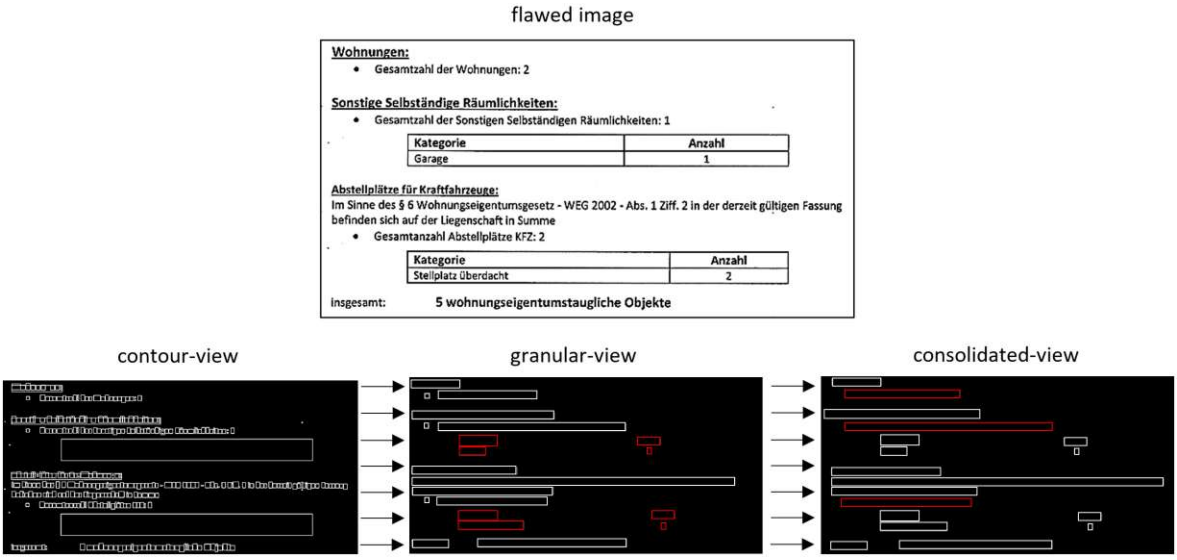

Figure 4.12: table detection - row-based

The contour view contains external contours, which lack detail due to gridlines, that need to be removed. Afterwards the contours are merged horizontally to remove interword spaces and to obtain the granular view. To avoid false positives e.g.: due to enumerations, like in the example above, the contours are merged once again horizontally.

Finally a simple rule is applied to the resulting contours:
In order to qualify as a table row, there needs to be more than one object per line. This is based on the assumptions, that tables have more than one column and that the visual cue is significantly bigger than interword spaces.

Figure 4.13 depicts the filter process for the initial version.

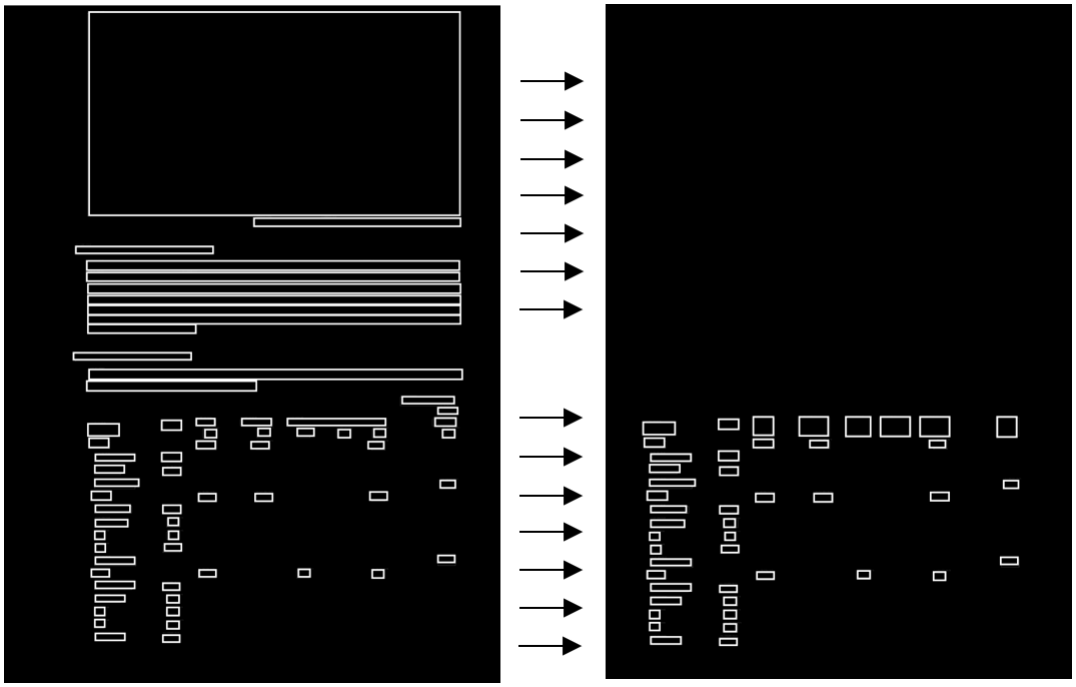

Figure 4.13: table detection - row-based filter

The biggest bounding box at the top represents an image that was part of the original document. Lines with only a single contour are removed, so that only the tables remain. Additionally headers are consolidated by detecting horizontal overlaps of bounding boxes within the same row and merging them accordingly.

Whether identified table rows belong together as a table is decided by a hardcoded parameter for the distance between them. In case the distance is greater than the parameter, lines are treated as separate tables.

Evaluation showed that the success of this table detection approach depends on the font and size of characters as well as the line format and the space between table columns. Scan quality is an additional factor since jagged lines might not be removed, which can cause tabular content to be lost entirely, because the external contour would be interpreted as a block of text and filtered accordingly.

The final iteration focuses on the gaps between bounding boxes to identify columns. Instead of determining whether a given row belongs to a table, the aim is to find header rows by merging gaps vertically and resolving conflicts between merges that overlap each other vertically.

For this purpose gaps between bounding boxes need to be identified first, which are represented by black boxes between the white boxes that stand for text in Figure 4.14:

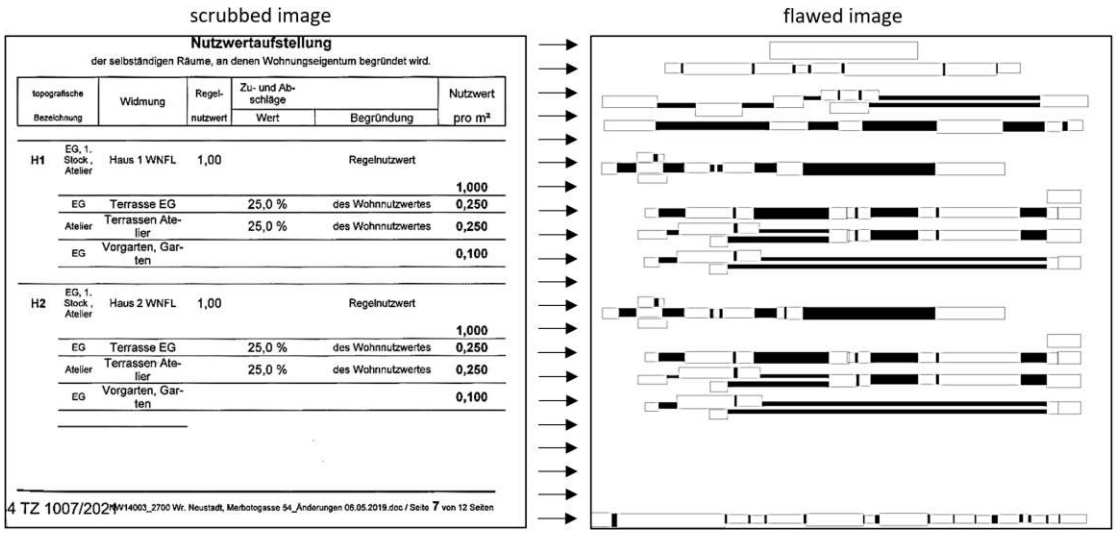

Figure 4.14: gap detection - final version

The height of a gap is limited by the size of the vertical overlap of consecutive text boxes belonging to the same row. Rows that only contain a single text box do not have gaps. It might happen, as in the third row of Figure 4.13, that gaps of a single row overlap each other horizontally. This results in a vertical collision later on, which is resolved by favoring the merge with greater length.

In the next step each gap is merged with all horizontally overlapping merges below while adjusting to the bounding boxes in those rows. In case there are no gaps in a subsequent row, the merge adjusts its width to avoid overlapping the bounding box of that row. The merge process stops either when the last row is reached, or when it is not possible to increase the height of a merge without overlapping a bounding box anymore.

A single gap might split into multiple parts in case it overlaps multiple objects in the next row (e.g. the last gap in the third row in Figure 4.14). In that case the resulting splits are picked up for the merge process in the next row.

The width of a merge gap is determined by the size of the horizontal overlap between the last two gaps being merged. The height is determined by the lowest y1-value of all bounding boxes within the row of the last merge. Merges from different rows overlap each other frequently, especially in table areas.

Figure 4.15 shows the output for the gaps from figure 4.14:

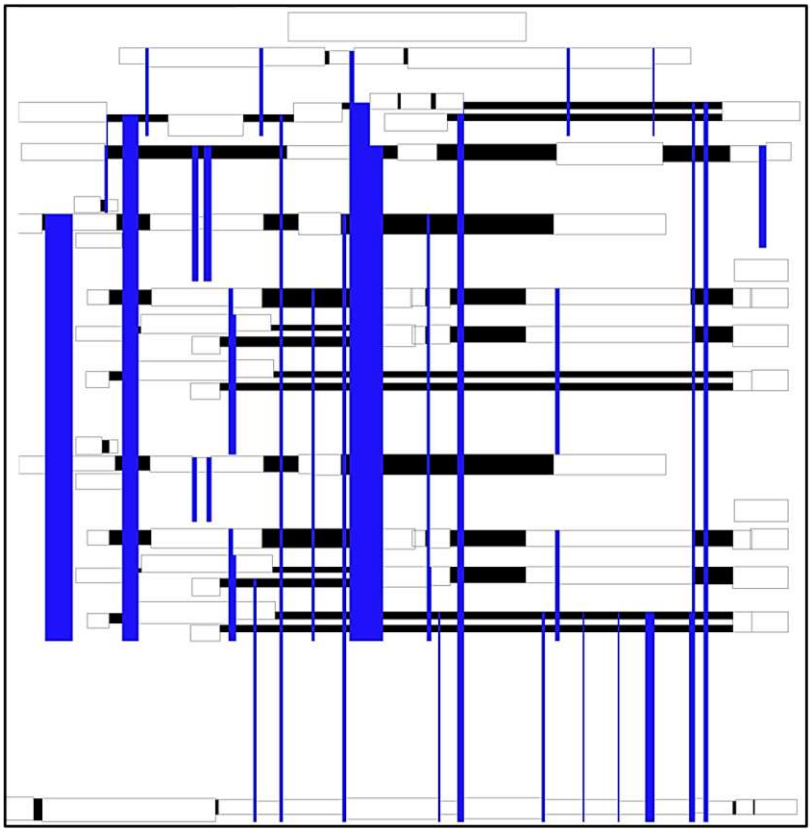

Figure 4.15: gap merge - trimmed, unfiltered

The length of the different column candidates might differ within each row, either because some overlap gaps in lower rows than others, or when they completely overlap bounding boxes horizontally in higher rows. Once the initial merge-pocess is done, merges of every row are trimmed. This is done by examining each original gap and its column candidates.

The vertical position of the last time each candidate merged with another gap are compared and the one with greater distance to the original gap is stored. The length of the merge with the minimum value across all original gaps is used as a cap for the length of all merge candidates of the same row. The goal is to ensure that resulting tables are well formed in the sense that each column has the same amount of rows.

Figure 4.16 highlights gaps, column candidates and parameters for trimming:

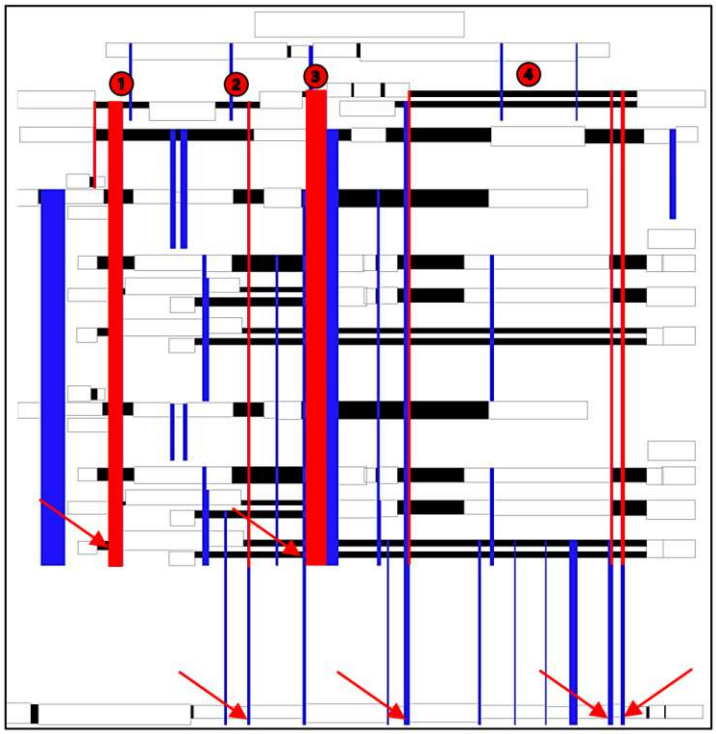

Figure 4.16: header-row: gaps, candidates and parameters

The integers enumerate the original gaps, the red blocks depict the column-candidates, and the arrows show the parameters. The results of the trimming process are not obvious due to merge overlaps of different rows, but they become visible once the header row is identified.

Even though it is not obvious in Figure 4.15, the target columns are a subset of all merges. Merges of the penultimate row cannot be distinguished visually from the ones of the third row, which is indeed the header row for the table.

Trimming is not only important to ensure of result tables, but also for determining the length of column candidates correctly. The first rows before and after table areas typically have significant vertical distance to the header and last table row. In case some original gaps of a table row find gaps suitable for merging in these content rows, the resulting column candidates either do not have equal length, or might become longer than real columns from header rows above.

At this stage multiple rows have candidates that overlap each other vertically, because every row below the header has its own merges, which partially overlap merges from the header. Besides there are multiple column candidates for original gaps of the header row.

4.5.3 Structure-Recognition

The initial version yet again takes a comparatively simple approach by iterating over the first row of bounding-boxes for each detected table. The position of the (x1,x2) pairs of the headers are used as column-boundaries. In case the start-point of a cell is smaller or equals the end-point of a header, and the end-point of a cell is greater or equals the start-point of a header, the cell is assigned to the column of the header.

Figure 4.17 depicts this process:

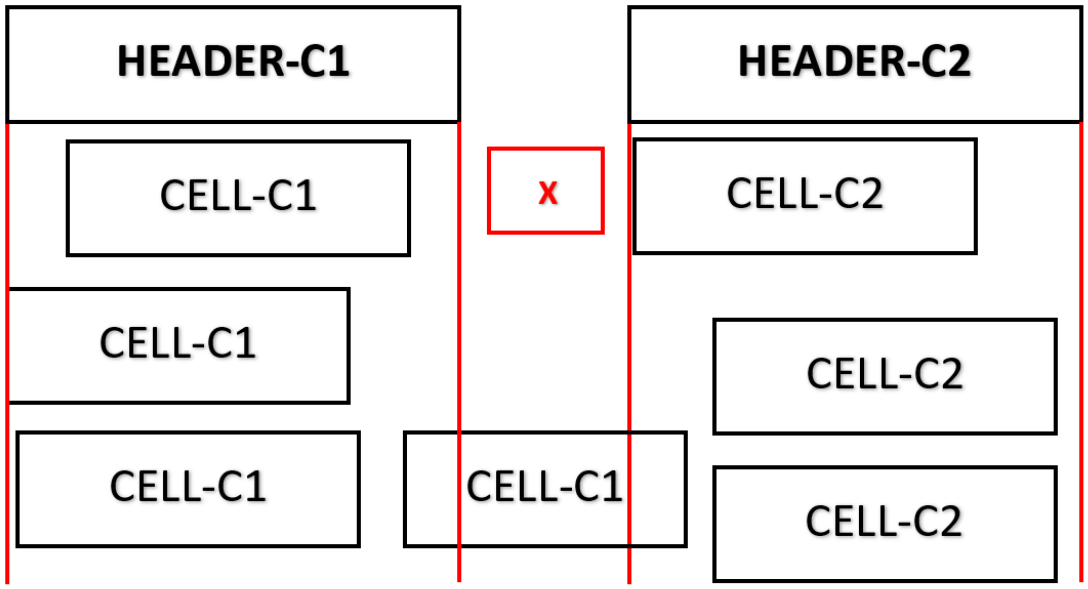

Figure 4.17: Structure-Recognition: initial version

Red lines show the column-boundaries. The red cell between the end-point of the first column (C1) and the starting-point of the second column (C2) is not picked up, because it does not fulfill the conditions for either column. The cell that overlaps both the end-point of C1 and the start-point of C2 is picked up by C1, because it comes first during the iterative process.

Sparse tables, where rows do not have content in every column require special treatment, because empty cells do not have bounding-boxes, but still need to be represented. Shifts in the table-structure would occur, if the empty cell does not belong to the last column.

To avoid this problem, the initial version inserts empty cells until the row-index of a column-cell fits to the row-index of the header. In case multiple bounding-boxes are assigned to the same column they are merged under the assumption that they belong to the same cell, as would be the case for the first two cells in the last row in Figure 4.17.

The final version detects headers by resolving conflicts between merges of rows and columns by deciding which merges of header-rows indeed represent columns. Conflicts arise when merge-candidates from different rows overlap each other vertically. This needs to be resolved since any given row can only belong to a single table.

First conflicting merge-rows below and above the current candidate-row are detected. Then scores for the candidate row and its competitors are calculated by summing up the length of every merge-candidate per row. In case the candidate-row does not have the highest score it is discarded. This is based on the assumption that real columns are longer and more numerous than other candidates which are either below and belong the the table, or above and result from incidental gap-merges.

Additionally all candidates that do not have vertical overlaps are discarded entirely, because tables should have more than one row, and merges of rows below the header should overlap header-merges. Trimming merges beforehand is necessary because otherwise any row might have a greater score than a header depending on positions of interword spaces.

Figure 4.17 shows the output from every step:

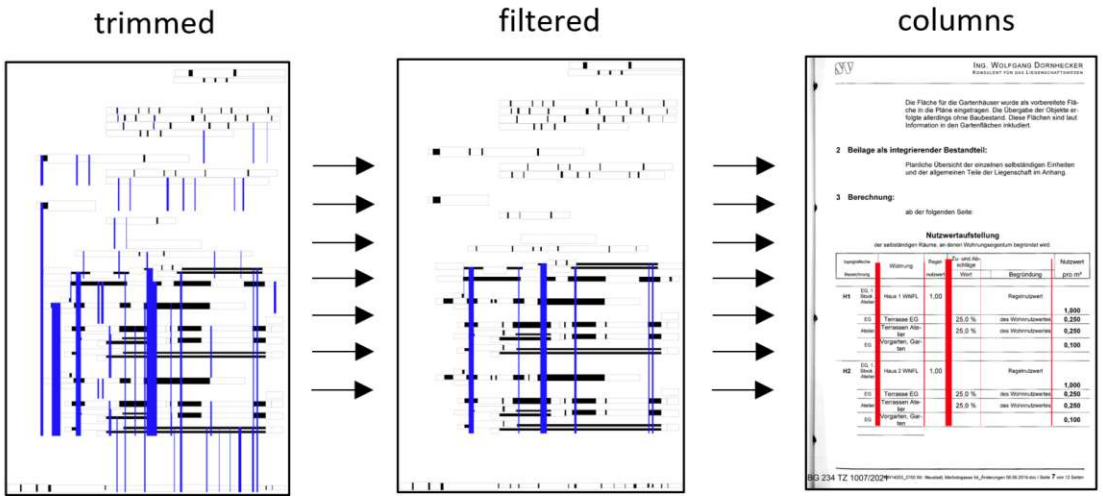

Figure 4.18: Structure-Recognition: final version

Figure 4.15 shows different cases regarding column-candidates: the first original gap and the last original gap contain multiple candidates, while all the others contain only one. When determining columns it has to be taken into account, that a single original gap might contain multiple columns as is the case for the last original gap.

In order to determine which merges represent columns, the content between column-candidates of the same original gap and their length is examined. To qualify as a column there must be at least one bounding-box and more than one gap between candidates.

For Nr.1 in Figure 4.16 the shorter candidate is discarded, because it is smaller than the other and because there is no gap between them (they only partially enclose the fist gap of the second row). Nr.4 contains 3 candidates, but only 2 columns. Between the last two candidates there is no content even though they have the same length, which is why the leftmost candidate is removed.

Once the columns are identified, the final version iterates over all rows that are enclosed by them. For the first column the content to the left is considered, for the last the content to the right and for the rest the content between the predecessor and the current column. In order to be able to deal with tables that only have two rows and not to loose the penultimate row in other cases, the last row is inserted once again at the end of the list of columns for this process.

4.5.4 Data-Extraction

Other than for the final iteration, text still needs to be extracted after detecting tables and their structure for the initial version. This is done by passing the contours for every cell to Pytesseract using a whitelist for characters, numbers and punctuation marks and German training data. The resulting textual representations for cells are stored in excel-tables according to the structures obtained for both versions.

The final step after table extraction is filtering for target-data, which is done in the same way for both versions. Initial analysis has shown, that object-id's are contained in the first two columns of UtilityValueInfo instances. Besides they usually are alpha-numeric and contain a numerical part referring to an enumeration and a textual part that represents the type of the object (parking lot, flat, or house for instance).

Cells of the first column are checked by a regular expression and all matches are gathered in a list. The elements within these lists are then split into a numerical and a textual part to facilitate counting objects per type. Should the first column not yield any result, the process is repeated for the second column. The result is a dictionary where the keys are strings representing object-types and values are numbers that show the count of objects that are part of the appraisal.

The following example shows the output for a result from the test-batch:

H	STPL	1-234-1007-2021_dss
['1', '2', '3', '4', '6', '5']	['10', '11', '12', '13', '14', '1', '2', '3', '4', '5', '6', '7', '9']	1

Table 4.1: Target Data: example

There are two Object-Types represented here, the first being "H" which stands for "house" in German and "STPL" which stands for "parking space". Mapping the abbreviations that are frequently used would improve results further, but there are also some that are too general to fathom the exact Object-Type, or lack context to derive it from.

CHAPTER 5

Evaluation Process

In order to answer the research question, results from a benchmark solution and both versions of the prototype needed to be assessed and compared. The areas of assessment are the capabilities to detect tables, their structure and their content correctly.

The ground-truth was extracted manually from a sample to provide the basis for a detailed evaluation using Precision and Recall as measurements. Additionally the ability to extract target data was evaluated using the entire test-batch to simulate real-world application and use-cases.

First the results for Precision and Recall are presented in the different areas of evaluation for each document within the sample. Conclusions are drawn from different outputs and categorized according to the table model. Examples are used to highlight reasons for differences and to detect strengths as well as weaknesses of the extraction systems to provide a basis for the research answer.

Finally the ability of the most promising rule based approach to detect target data is examined using the entire test batch. Instead of focusing on specific examples from certain documents as in the previous evaluation steps, the overall results are presented along with some background information regarding the usability of the prototype for its intended purpose.

5.1 Results for Table-Recognition

The extraction results for all three software artefacts are summarized in Table 5.1. "DocNR" in the first ground-truth-column corresponds to the ids of the sample from Table 3.2 and the correct number of tables per document can be found next to the number of pages per document. The column "Found" refers to the number of relevant instances retrieved, the column "Missed" refers to the number of false negatives, and the column "Added" refers to the number of false positives. The Precision values are calculated by dividing the number of tables per document by the sum of relevant tables and false positives that were found. The Recall values are the number of relevant tables found divided by the number of relevant tables per document.

ground-truth			row-based approach					column-based approach					AI-benchmark				
DocNR	Pages	Tables	Found	Missed	Added	P_1	R_1	Found	Missed	Added	P_2	R_2	Found	Missed	Added	P_3	R_3
7	30	11	11	0	83	**0.117**	**1**	10	1	138	**0.058**	**0.909**	10	1	21	**0.355**	**0.909**
39	9	8	8	0	2	**0.8**	**1**	6	2	9	**0.533**	**0.75**	8	0	1	**0.889**	**1**
80	49	29	25	4	69	**0.309**	**0.862**	25	4	37	**0.468**	**0.862**	27	2	0	**1**	**0.931**

Table 5.1: Table-Recognition: Results

Results show that the scope of test data was sufficient for spotting weaknesses in our prototypes as well as the benchmark product, which outperformed the rule-based approaches consistently with one exception. Even though the Recall-value for the row-based approach is slightly better for the first document, the Precision-value shows that it is much less accurate than the benchmark. Even though the sample is not big enough to establish a trend, the results hint at Recall being favored over Precision in design-choices, because Precision-values are more impacted by false positives than by false negatives.

The number of false positives is highest for the first document. This is due to an excerpt from the land register that has been attached to the appraisal, which contains many structures that are similar to tables. Both the row - and column - based approach are prone to false positives that originate from justified text, which makes content appear to have the same format as tables. The benchmark product is also affected, but to a much smaller degree.

The length of a document compared to its number of tables is a contributing factor. Amount and form of non tabular content determine the instances where extraction systems might produce false positives. The amount impacts the frequency of errors and the form impacts the types of errors that might occur. The form of content plays a bigger role in the sample since the numbers for false positives do not correspond to the ratios of pages and tables.

The third document contains tables that are embedded in plans and other graphics, which proved to be an issue for every extraction system and impacted both the number of false positives and false negatives.

The coloured blocks in Figure 5.1 shows examples for false positives that have been extracted by the different solutions. The row-based approach only detected a single row (blue), the column-based approach extracted the entire block in two columns (red), and the benchmark product detected the enumeration characters and the content of their rows as a single table while skipping the rows between them:

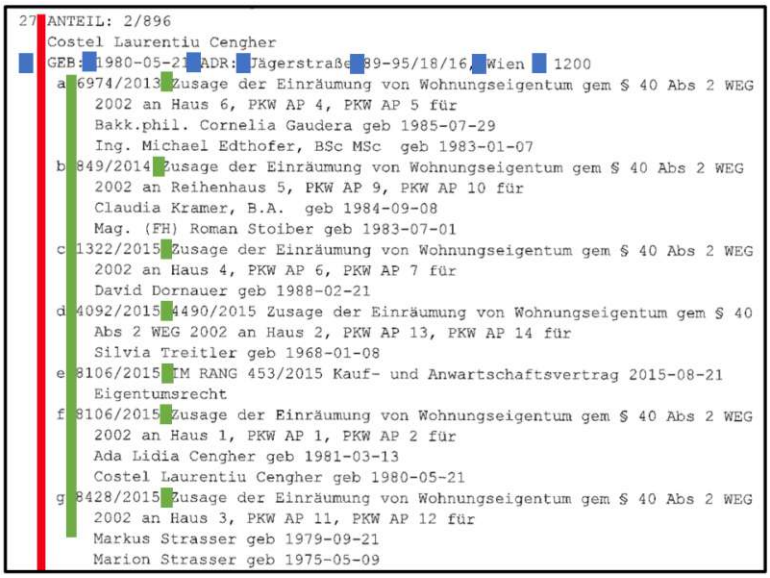

Figure 5.1: Table-Recognition - false positives

The red block shows that the column-based approach favors the tabular structure that spans the most rows, whereas the row-based approach went for the biggest horizontal visual cue. The benchmark filtered rows that did not fit the structure of the rows it extracted. It is remarkable that the benchmark result contains three columns, because it would have also been possible to extract more rows with a lower column count.

While the column-based approach struggled to identify multiple tables on a single page and favored a different structure over a table, it looks like the benchmark interpreted candidates as lists due to the leading character in every row and cannot cope with visual cues the are represented by characters rather than whitespace. The table that every solution missed is special in the sense that it is also surrounded by other more structured tables and hints at the environment playing a big role for table-extraction as well as table-features.

Figure 5.2 shows examples for false negatives:

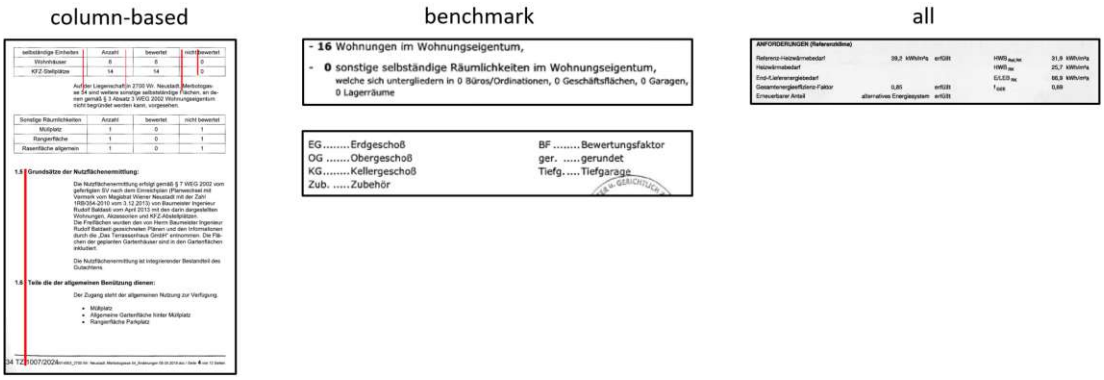

Figure 5.2: Table-Recognition: false negatives

Reasons for false positives and false negatives are consistent for the sample. The row-based approach missed tables that had visual cues that were smaller than the parameter for merging contours, which leads to table-rows being identified as text. Choosing the parameter dynamically is highly complex since there is no reliable relation between the size of visual cues for tables and the size of text separators in paragraphs of text within the same and especially across several documents. This is also due to the fact that utility value appraisals frequently contain parts from different sources with different formatting. Even on a single given page it is not guaranteed that the smallest visual cue of a table is bigger than the biggest text separator. Finding solutions dynamically could either be done with statistical approaches, or would require a much more sophisticated rule-set.

The column-based approach struggled to identify header rows correctly when resolving vertical conflicts between candidates. The enumeration in Figure 5.2 was chosen over the second table because the resulting table is larger than the real table would have been. Merging gaps across rows is very susceptible to structures that are similar to tables and also overlapping text-separators that either originate from chance or justified text. Here, more sophisticated rules would most probably lead to better results, tables that do not contain text in a header-cell could be disregarded before resolving vertical overlaps for instance. We chose not to implement such rules before filtering for target- data since they are very domain-specific and would have skewed results in domain independent evaluation areas in our favor. Another issue is highlighted by the second table the benchmark missed, which is that separators are not necessarily represented by gaps, but could also be expressed by characters like dots, or hyphens. The column-based approach did extract the table correctly but not its structure since it only focuses on whitespace and thus created 2 columns instead of 4. Regarding the ground-truth the example could also serve as a basis for discussions since it is not clear if it contains a single, or two tables.

Justified text lead to false positives for all extraction systems, although the benchmark product was least affected.

5.2 Results for Table-Structure-Recognition

This area of evaluation focuses on the correct placement of table cells for relevant tables that were extracted by the different solutions. Using the count of cells and their position for each table from the sample as ground-truth, we counted the number of cells that were found and the number of cells that were positioned correctly and calculated Precision and Recall based on those numbers. Errors from previous steps are propagated (indicator: "-"). The column "Type" corresponds to the column "Nr." in Table 3.3, which provides information about table attributes according to our table model in Figure 3.8. This dimension was added to examine whether certain attributes impact Precision and Recall, which would open up new possibilities for attribute-oriented rule-sets.

ground-truth				row-based approach				column-based approach				AI-benchmark			
DocNR	Type	Table	Fields	Found	CorrectPos	P_1	R_1	Found	CorrectPos	P_2	R_2	Found	CorrectPos	P_3	R_3
7	1	1	12	12	12	**1**	**1**	15	9	**0.6**	**0.75**	12	12	**1**	**1**
7	1	2	16	4	4	**1**	**0.25**	0	0	-	-	16	16	**1**	**1**
7	2	3	45	53	7	**0.132**	**0.156**	50	45	**0.9**	**1**	48	45	**0.938**	**1**
7	3	4	84	84	14	**0.167**	**0.167**	76	26	**0.342**	**0.31**	76	76	**1**	**0.905**
7	3	5	79	73	66	**0.904**	**0.835**	67	45	**0.672**	**0.57**	72	61	**0.847**	**0.772**
7	4	6	8	6	6	**1**	**0.75**	6	6	**1**	**0.75**	0	0	-	-
7	5	7	61	59	57	**0.966**	**0.934**	52	39	**0.75**	**0.639**	54	50	**0.926**	**0.82**
7	5	8	150	157	121	**0.771**	**0.807**	138	82	**0.594**	**0.547**	111	103	**0.928**	**0.687**
7	6	9	52	60	42	**0.7**	**0.808**	49	45	**0.918**	**0.865**	42	42	**1**	**0.808**
7	7	10	14	11	4	**0.364**	**0.286**	20	6	**0.3**	**0.429**	12	12	**1**	**0.857**
7	8	11	7	15	2	**0.133**	**0.286**	6	2	**0.333**	**0.286**	7	7	**1**	**1**
39	9	1	10	10	1	**0.1**	**0.1**	15	5	**0.333**	**0.5**	10	10	**1**	**1**
39	10	2	47	46	1	**0.022**	**0.021**	46	39	**0.848**	**0.83**	47	47	**1**	**1**
39	11	3	106	106	92	**0.868**	**0.868**	106	99	**0.934**	**0.934**	106	106	**1**	**1**
39	11	4	58	57	54	**0.947**	**0.931**	67	53	**0.791**	**0.914**	58	58	**1**	**1**
39	12	5	76	76	63	**0.829**	**0.829**	74	72	**0.973**	**0.947**	76	76	**1**	**1**
39	13	6	213	210	22	**0.105**	**0.103**	186	186	**1**	**0.873**	213	213	**1**	**1**
39	9	7	10	10	1	**0.1**	**0.1**	0	0	-	-	10	10	**1**	**1**
39	13	8	8	9	8	**0.889**	**1**	0	0	-	-	8	8	**1**	**1**
80	14	1	10	12	10	**0.833**	**1**	6	0	**0**	**0**	10	10	**1**	**1**
80	9	2	6	4	4	**1**	**0.667**	0	0	-	-	6	6	**1**	**1**
80	15	3-5	64	44	37	**0.841**	**0.578**	68	64	**0.941**	**1**	51	30	**0.588**	**0.469**
80	16	6	23	20	13	**0.65**	**0.565**	23	23	**1**	**1**	16	9	**0.563**	**0.391**
80	17	7-8	121	0	0	-	-	99	40	**0.404**	**0.331**	86	86	**1**	**0.711**
80	18	9	14	15	6	**0.4**	**0.429**	24	14	**0.583**	**1**	0	0	-	-
80	19	10-17	754	711	576	**0.81**	**0.764**	595	536	**0.901**	**0.711**	734	710	**0.967**	**0.942**
80	20	18	127	155	61	**0.394**	**0.48**	109	85	**0.78**	**0.669**	116	100	**0.862**	**0.787**
80	21	19	200	231	189	**0.818**	**0.945**	216	200	**0.926**	**1**	200	200	**1**	**1**
80	22	20	8	8	8	**1**	**1**	10	6	**0.6**	**0.75**	8	8	**1**	**1**
80	16	21	9	0	0	-	-	0	0	-	-	0	0	-	-
80	23	21	11	4	2	**0.5**	**0.182**	14	8	**0.571**	**0.727**	7	7	**1**	**0.636**
80	24	22-25	18	15	15	**1**	**0.833**	22	18	**0.818**	**1**	18	18	**1**	**1**
80	25	26-28	26	0	0	-	-	0	0	-	-	26	26	**1**	**1**
80	26	29	10	0	0	-	-	0	0	-	-	5	5	**1**	**0.5**

Table 5.2: Structure-Recognition: Results per Type

There are several table types that were included multiple times across the different documents (Types: 1, 3, 5, 9, 11, 13, 16), for those the benchmark performs consistently compared to the rule based approaches. Perfect results were achieved for the second document only and the benchmark outperformed the other approaches in most cases.

Despite the small sample size it is apparent that table attributes do not necessarily impact the quality of extraction results. This indicates, that there are other contributing factors like the surroundings of a table and the quality of the scan.

The row-based approach frequently merged headers of separate columns which depends rather on the distance between columns than the type of table, or scan quality of the document. The reason for its inconsistent performance within Type1 for instance, is due to noise from the scanning process that got misinterpreted as a table cell (see Figure 4.8).

Such noise related issues do not occur for the column-based approach, which is more prone to missing columns when a single merge split into multiple ones and to missing rows by selecting the wrong header-row within a table. The current rules are not sufficient to reliably distinguish between column-candidates and real columns. Besides the results are not consistent even for very similar tables, as Figure 5.3 shows:

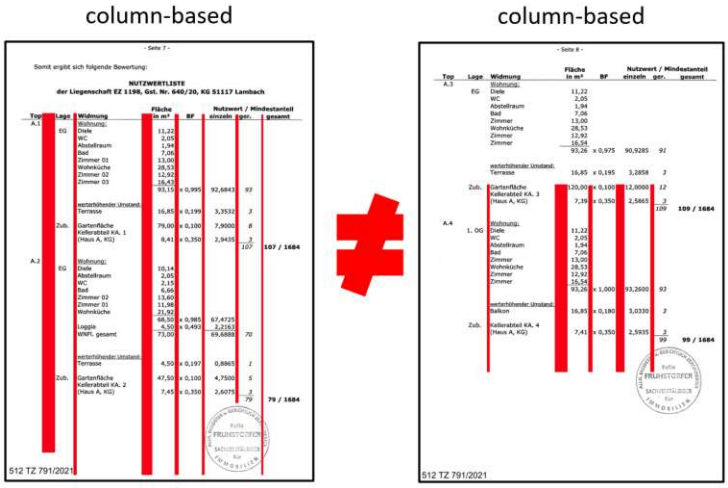

Figure 5.3: column-based approach: inconsistencies

The results of the benchmark indicate that it is more prone to missing fields than to adding additional ones. This hints at the benchmark establishing a structure and disregarding fields that do not fit that structure while the rule based approaches extract all fields within the recognized table area.

Even minor errors like adding an additional column or merging two columns have great impact because they cause shifts in the table structure that affect all subsequent columns. In general issues with row-recognition have smaller impact for long tables than issues with column-recognition.

There are differences in the quality of the extraction results across the three documents. This can in part be contributed to some documents containing simpler table types than others like DocNr.7 and DocNr.39 for instance. Besides the types present in DocNr.39 are more similar to each other than in the other documents.

DocNr.80 is special since it contains an additional ObjectType: add-ons. Those are plans and maps of the surrounding area which also contain tables in some cases. This complexity is reflected even in the results of the benchmark: Perfect scores were achieved for Types(14,9,21,22,24,25), while scores for Types(15,16,17,23,26) were negatively impacted by issues with missing fields and positioning due to small visual cues, sparse content, merged rows and missing lines.

Types(15,16) stand out due to their bad performance for positioning the recognized cells, also compared to other documents. This is again not due to their table attributes, but due to a special case: a table-like structure missing logical connections close to the actual table has been misinterpreted as header. The assumption is based on the output, where the benchmark cuts the text-column to fit below the table like structure:

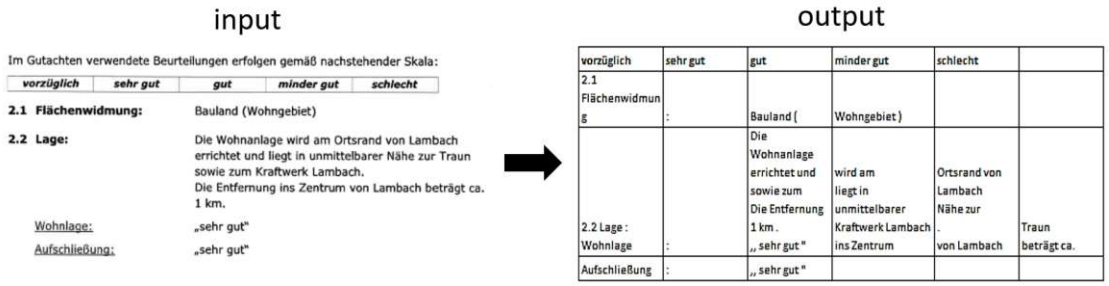

Figure 5.4: benchmark - impact of table like structures

While it is quite obvious that the benchmark outperformed the rule based approaches, it is not so easy to determine which of the latter yielded better results than the other overall. For this purpose we aggregated the results in a similar fashion like for Table-Recognition:

ground-truth			row-based approach				column-based approach				AI-benchmark			
DocNR	Tables	Fields	Found	CorrectPos	P_1	R_1	Found	CorrectPos	P_2	R_2	Found	CorrectPos	P_3	R_3
7	11	528	534	335	**0.627**	**0.634**	479	305	**0.637**	**0.578**	450	424	**0.942**	**0.803**
39	8	528	524	242	**0.462**	**0.458**	494	454	**0.557**	**0.86**	528	528	**1**	**1**
80	29	1401	1219	921	**0.756**	**0.657**	1186	994	**0.643**	**0.709**	1283	1215	**0.947**	**0.867**

Table 5.3: Structure-Recognition: Results per Document

Table types were left out since it became clear, that they do not impact results as much as initially assumed. The scores of both rule-based approaches are quite similar for the first and the last document, only for the second document both Precision and Recall are higher for the column-based approach. The row-based approach picked up more fields, but added some additional ones for the first documents and positioned less fields correctly than the column-based approach in general.

For deciding which approach is best suited for extracting target data, the capability to extract text correctly still needs to be taken into account since identifying tables and their structure alone is not sufficient to facilitate use-cases based on extracted target data.

5.3 Results for Character-Recognition

This step focuses on the quality of extracted text per recognized table cell. We only counted a cell as having been extracted correctly when the results matched the ground-truth 1:1, meaning that every single character has to be extracted accurately. Errors from Table-Recognition and Table-Structure-Recognition are propagated since fields that were not recognized have not been processed and thus cannot contribute to the correct extraction results. The same goes for unintentional splits and merges of cells which cannot lead to 1:1 matches with the ground-truth.

Due to issues with German special characters and certain punctuation marks, results for both prototypes are lower than they would have been under less strict rules. Scores were calculated by dividing the number of fields fields that were extracted correctly by the number of extracted fields (Precision) and the number of fields per document (Recall).

ground-truth			row-based approach				column-based approach				AI-benchmark			
DocNR	Tables	Fields	Found	Correct	P_1	R_1	Found	Correct	P_2	R_2	Found	Correct	P_3	R_3
7	11	528	534	327	**0.612**	**0.619**	479	199	**0.415**	**0.377**	450	425	0.944	**0.805**
39	8	528	524	453	**0.865**	**0.858**	494	253	**0.512**	**0.479**	528	509	0.964	**0.964**
80	29	1401	1219	939	**0.77**	**0.67**	1186	590	**0.497**	**0.421**	1283	1218	0.949	**0.869**

Table 5.4: Character-Recognition: Results per Document

OCR-quality does not depend on table-types but rather on image quality, the extraction approach and tools being used. Judging by the consistent performance of the benchmark, scan imperfections and noise do not have a great impact on our test-set since the documents used provide some variety in this regard, which is not reflected in the results.

There were no dedicated extraction rules for characters in both versions of the prototype. Open source tools were used with very little fine tuning and configuration. Differences in performance compared to the benchmark are clearly visible and are based on its ability to handle superscript, punctuation marks and German special characters.

The column-based approach did not use German training data, which lead to it yielding the worst results and shows the importance of language specific extraction even when similar character sets are being used by different languages like for english and German.

The row-based approach would have benefited from additional steps for character recognition. Bounding boxes being either to wide or narrow impacted results negatively especially when the text was not centered in its bounding-box or when noise was passed to the OCR-Tool within the bounding-box. It is clearly better suited for extracting target data than the column-based approach and was therefor used for the last evaluation step.

Results already indicate problems for target data extraction because it is negatively impacted by errors from all previous steps as well. The target data extraction procedure requires high output quality regarding OCR due to its sensitivity to even minor mistakes and noise during the aggregations taking place on key level as well as the extraction via regex-patterns.

5.4 Results for Target data

The results of the first iteration show potential on one hand but also give rise to the question whether the set goal is achievable with the proposed methodology on the other. They heavily depend on the output of previous steps, especially on the OCR step.

All documents of the test-batch have been processed by the row-based approach since it outperformed the column-based approach and processing larger amounts of data with the benchmark product would have required a license, which was not obtainable for free.

Whenever the output for a given document only contained relevant data and matched the number of objects and their types, we counted it as "PerfectMatch", if additional keys were picked up by the procedure but the relevant data was still correctly included, we counted it as "PerfectNoise" and if there was no output we counted those document separately as well:

Documents	PerfectMatch	PerfectNoise	NoOutput
100	6	28	4

Table 5.5: Target data extraction: Results

This shows that only a small number of documents were processed in such a way that outputs are usable without requiring human supervision and that from some documents target data could not be extracted at all. About one third of all documents yielded results that would be usable for the intended purposes but not without additional cleaning steps.

The following table shows the output for a document where target data was not extracted successfully:

OUND	A.	Balkon	LANK	B.	Heizwarmebedarf	1-512-791-2021_dss
[' 51117', ' 06', ' 2']	['1', '2', '3', '4', '5', '6', '8', '9', '10', '11', '12', '13', '14', '15', '16']	[' 16', ' 4']	[' 1', ' 06', ' 891']	['1', '2', '3', '4', '5', '6', '8']	[' 39', ' 34', ' 27']	1

Table 5.6: Target data extraction: Example with issues

The last column contains the name of the document as header and its ID below, while the others contain the different keys that were picked up by the regular expression. Only the second and fifth column contain target data, which is not discernible without mapping the abbreviations to certain object types. In this case this cannot be done without knowing more about the document itself, besides there are gaps in the enumeration for the object-types (Nr.7 is missing for both "A." and "B."). This example did not contribute to either "PerfectMatch", "PerfectNoise", or "NoOutput" in Table 5.5.

Without having access to a ground-truth it could not be said if there should be gaps in the enumerations or not since not all objects within a property have to be subject of a given appraisal. This indicates that metadata needs to be extracted in addition to tabular content to obtain meaningful and reliable results. Mappings for abbreviations like "A." and "B." to object types like "flat" and "parking lot" also require contextual data, because the meaning of the same abbreviation can differ across different utility value appraisals.

CHAPTER 6

Research Results

The different evaluation steps yielded enough data to provide a basis for answering the research question. Findings are summarized in the following to highlight research results and point towards opportunities for further research.

Limitations for both the research process and the scope of the research results are highlighted to show areas for improvement as well as to determine possible applications.

Finally scientific contributions are presented which can be reused or refined to achieve better results when creating rule-based approaches or evaluating them.

6.1 Answers to the Research Question

The research question we set out to answer is:
"How can domain dependent rule-based methods achieve higher Precision and Recall than domain independent machine learning approaches when extracting tabular data from utility value appraisals?".

We proposed two different rule based approaches with different complexity, a row-based approach and a column-based approach. Both of them managed to outperform the commercial benchmark product in a few cases, but did not achieve higher Precision and Recall values overall in any category that was part of the joint evaluation.

So with the results obtained we cannot prove that our approaches constitute an answer to the research question, but only that there is still untapped potential for rule based approaches.

The approaches we invented are very simple compared to ones based on statistical and machine learning methods. Their simplicity while being advantageous for explaining the ideas they are based on, proved to be a hindrance for dealing with complex issues such as determining the size and beginning of a table and its columns correctly.

This leaves a lot of room for follow-up research to refine the proposed approaches and add to the presented rule-sets in order to achieve better results. The research question contains the assumption that rule-based methods can indeed outperform domain independent machine learning approaches, which we did manage to show in the course of the evaluation process. Taking into account that we compared a commercial software product with two prototypes resulting from an academic study should be seen as a further indication for the potential of rule-based approaches.

While we were not able to outperform the benchmark consistently by far, we managed to spot weaknesses in all extraction systems that we evaluated. Those weaknesses hint at different factors that impact the quality of the output for extraction systems. Initially we presumed that handling certain table attributes via dedicated rules could lead to a way towards accepting the hypothesis the research question contains.

The evaluation results showed that other factors have much greater impact, like the environment of a table, the layout of non tabular content and the quality of the scanned images. While certain domain dependent rules could provide shortcuts for better results they are not necessary for achieving reliable results for utility value appraisals as the evaluation of the benchmark showed.

We also found that the different approaches focusing on rows or columns suffer from different issues, which suggests that their rule-sets could be combined to alleviate weaknesses. Results indicate that instead of trying to handle certain table attributes specifically, recognizing different table like structures and distinguishing them from tables could further improve scores for Precision and Recall by avoiding the common mistakes that were highlighted by the evaluation examples.

6.2 Limitations of the Research Process

A focal point of the research is the understanding of the term "table". We used a textual description that designates two identifying factors (visual cue and logical connection), but there are many other possible ways of understanding what a table is and what not. The distinctions between tabular and non tabular content pervade the entire research process and especially its evaluation.

While our approaches explored different ways of interpreting visual cues, logical connections were not examined in detail. The fact that they are dependent on the problem domain would even have been beneficial for our results, but might have also obscured domain independent issues of the proposed extraction systems. Still the disregard for logical connections is a shortcoming of this research that came to pass due to time limitations, which also highlight a weakness of rule-based approaches: the time necessary for coming up with and implementing heuristics.

There is a multitude of labelled test-data-sets available for compound documents. Due to the limited availability of utility value appraisals, it would have served our purposes to use public data-sets as a starting point before shifting the focus to our target domain.

This would have opened further possibilities for analysing the structure of utility value appraisals by comparing them with other types of compound documents. Besides it would have alleviated the most prominent issue with our test-data which was its comparatively small scope and the manual effort required for ground-truthing and evaluation. Additionally, using available data-sets would have freed up time for more sophisticated rule-sets and additional iterations for further refinements.

Our table-model does contain almost every possible table feature in utility value appraisals but is yet too simple to take environmental factors and scan quality into account. Adding those dimensions could provide a basis for establishing and evaluating all factors that determine the performance of different extraction systems and could be used to tailor them for specific use-cases.

The evaluation process would have benefited from additional iterations that could have been made possible by automatizing it entirely. For that purpose existing research and tools could have been utilized, which would have enabled a more detailed analysis of the impact of certain extraction rules.

Target-Data extraction could have been enhanced with more domain specific rules, but the evaluation results made clear that previous domain independent steps need to provide higher output quality before reliable results can be obtained for large numbers of documents. Besides limiting the scope to tabular data only turned out be a hindrance for applying our ontology due to the lack of contextual data. Additional funds would have enabled us to include the benchmark product in this evaluation step as well, which would have made a comparison to the results for our prototype possible.

6.3 Research Contributions

We began by examining and documenting distinctive features of utility value appraisal and built an ontology which can be reused for other extraction systems. It could facilitate not only other rule-based approaches but also machine learning ones by providing a guideline for labelling data sets and categorizing different types of target data. Besides we provided an overview of relevant data contained in utility value appraisals, their location and sequence.

By studying the test-batch we created a table-model that enables labelling tables according to a list of features. Being able to categorize tables can aid in building test-sets and discussing tables in general. Even though our rule-based approaches were not found to be decisively impacted by certain features, others are and applying our model might help spotting weaknesses in other extraction systems.

The two rule-based approaches we proposed are documented as well as their weaknesses, which could be used as a starting point for follow-up research to refine them. Our goal of presenting general ideas was met in the sense of providing a basis for more sophisticated rule-sets.

Besides the documentation of our efforts and results can aid in make-or-buy-decisions for companies that seek to enter the market of extracting tabular data from scanned images.

We showed our results by evaluating the areas of Table-Detection, Table-Structure-Recognition and Character-Recognition for our prototypes and a benchmark solution. Finally we assessed the usefulness of our approach for real world applications. This combination of methods could serve as a template for the evaluation of tabular data extraction in further research.

Bibliography

[BGR07] Wojciech Bieniecki, Szymon Grabowski, and Wojciech Rozenberg. "Image preprocessing for improving ocr accuracy". In: *2007 international conference on perspective technologies and methods in MEMS design*. IEEE. 2007, pp. 75–80.

[Cam89] James P. Cameron. "A cognitive model for table editing". In: *Technical report OSU-CISRC6/89-TR 26*. Computer and Information Science Research Centre, Ohio State University, USA. 1989.

[CK93] S. Chandran and R. Kasturi. "Structural recognition of tabulated data". In: *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*. 1993, pp. 516–519. DOI: 10.1109/ICDAR.1993.395683.

[CLR13] Laura Chiticariu, Yunyao Li, and Frederick Reiss. "Rule-based information extraction is dead! long live rule-based information extraction systems!" In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, pp. 827–832.

[CMG96] M Cumplido, P Montolio, and Antoni Gasull. "Morphological preprocessing and binarization for OCR systems". In: *Mathematical Morphology and its Applications to Image and Signal Processing*. Springer, 1996, pp. 393–400.

[CZ17] Andreiwid Sheffer Corrêa and Pär-Ola Zander. "Unleashing tabular content to open data: A survey on pdf table extraction methods and tools". In: *Proceedings of the 18th Annual International Conference on Digital Government Research*. 2017, pp. 54–63.

[Fan+11] Jing Fang et al. "A table detection method for multipage pdf documents via visual seperators and tabular structures". In: *2011 International Conference on Document Analysis and Recognition*. IEEE. 2011, pp. 779–783.

[Gmb] Sparkassen-Finanzportal GmbH. *Das Überweisungs-Formular Anleitung*. URL: https://www.sparkasse.de/service/barrierefrei/leichte-sprache/ueberweisung-dauerauftrag-lastschrift.html (visited on 03/30/2022).

[Gra12] Alex Graves. "Supervised sequence labelling". In: *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, pp. 5–13.

[Hao+16] Leipeng Hao et al. “A table detection method for pdf documents based on convolutional neural networks”. In: *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. IEEE. 2016, pp. 287–292.

[HB07] Tamir Hassan and Robert Baumgartner. “Table Recognition and Understanding from PDF Files”. In: *Proceeding of the ICDAR 2007*. Vortrag: ICDAR 2007- 9thInternational Conference on Document Analysis and Recognition, Curitiba, Brasilien; 2007-09-23 – 2007-09-26. Volume 1, 2: IEEE Computer Society Press, 2007, pp. 1143–1147. ISBN: 0-7695-2822-8.

[Hou62] Paul VC Hough. *Method and means for recognizing complex patterns*. US Patent 3,069,654. Dec. 1962.

[Hu+00] Jianying Hu et al. “Table structure recognition and its evaluation”. In: *Document Recognition and Retrieval VIII*. Vol. 4307. SPIE. 2000, pp. 44–55.

[JT+06] Alıpio M Jorge, Luis Torgo, et al. “Design of an end-to-end method to extract information from tables”. In: *International Journal of Document Analysis and Recognition (IJDAR)* 8.2 (2006), pp. 144–171.

[KD98] Thomas Kieninger and Andreas Dengel. “The t-recs table recognition and analysis system”. In: *International Workshop on Document Analysis Systems*. Springer. 1998, pp. 255–270.

[Kum15] Rohit Kumar Kaliyar. “Graph databases: A survey”. In: *International Conference on Computing, Communication & Automation*. IEEE. 2015, pp. 785–790.

[Len98] Jed Lengyel. “The convergence of graphics and vision”. In: *Computer* 31.7 (1998), pp. 46–53.

[Li+20] Minghao Li et al. “TableBank: Table Benchmark for Image-based Table Detection and Recognition”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 1918–1925. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.236.

[Mai] Jochen Mai. *Organigramm: Vorlage, Vorteile, Beispiele, Tipps Anleitung*. URL: https://karrierebibel.de/organigramm/ (visited on 03/30/2022).

[Ots79] Nobuyuki Otsu. “A threshold selection method from gray-level histograms”. In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.

[Poo14] Ponnusamy Poovizhi. “A study on preprocessing techniques for the character recognition”. In: *International Journal of Open Information Technologies* 2.12 (2014), pp. 21–24.

[RFR94] M. Armon Rahgozar, Zhigang Fan, and Emil V. Rainero. “Tabular document recognition”. In: *Document Recognition*. Ed. by Luc M. Vincent and Theo Pavlidis. Vol. 2181. International Society for Optics and Photonics. SPIE, 1994, pp. 87–96. DOI: 10.1117/12.171096. URL: https://doi.org/10.1117/12.171096.

[RSS+15] André Ribeiro, Afonso Silva, Alberto Rodrigues da Silva, et al. “Data modeling and data analytics: a survey from a big data perspective”. In: *Journal of Software Engineering and Applications* 8.12 (2015), p. 617.

[RSS13] Sepideh Barekat Rezaei, Hossein Sarrafzadeh, and Jamshid Shanbehzadeh. “Skew detection of scanned document images”. In: (2013).

[Sch+17] Sebastian Schreiber et al. “Deepdesrt: Deep learning for detection and structure recognition of tables in document images”. In: *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*. Vol. 1. IEEE. 2017, pp. 1162–1167.

[Sei+11] Maung K Sein et al. “Action design research”. In: *MIS quarterly* (2011), pp. 37–56.

[SKI94] H. Saiga, Y. Kitamura, and S. Ida. “High-speed recognition of tabulated data”. In: *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5)*. Vol. 2. 1994, 577–579 vol.2. DOI: 10.1109/ICPR.1994.577043.

[SL10] Gabriel de França Pereira e Silva and Rafael Dueire Lins. “Assessing the OCR degradation in the generation of jpeg, png, and tiff files from adobe pdf”. In: *International Telecommunications Symposium, proceedings*. 2010, pp. 1–4.

[SL15] Lijun Sun and Yao Liu. “Review of Research on Table Extraction in Scientific and Technical Literature”. In: *International Journal of Knowledge and Language Processing* 6.3 (2015), pp. 48–62.

[SP12] K Sreedhar and B Panlal. “Enhancement of images using morphological transformation”. In: *arXiv preprint arXiv:1203.2514* (2012).

[WBM18] Bernhard Waltl, Georg Bonczek, and Florian Matthes. “Rule-based information extraction: Advantages, limitations, and perspectives”. In: *Jusletter IT (02 2018)* (2018).

[WH02] Yalin Wang and Jianying Hu. “A machine learning based approach for table detection on the web”. In: *Proceedings of the 11th international conference on World Wide Web*. 2002, pp. 242–250.

[Xu+17] Qiongkai Xu et al. “Collective vertex classification using recursive neural network”. In: *arXiv preprint arXiv:1701.06751* (2017).

[Zou+19] Zhengxia Zou et al. “Object detection in 20 years: A survey”. In: *arXiv preprint arXiv:1905.05055* (2019).

[Zuy97] Konstantin Zuyev. “Table image segmentation”. In: *Proceedings of the Fourth International Conference on Document Analysis and Recognition*. Vol. 2. IEEE. 1997, pp. 705–708.