UNIVERSITY OF TWENTE.

Multi-modal Document Classification in Architecture, Engineering, and Construction Asset Management Applications

by

Floor Rademaker

A thesis submitted to the
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
in partial fulfilment of the requirements for the degree of

MSc in Business Information Technology

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

University of Twente

Enschede, Overijssel, The Netherlands

June 2025

ABSTRACT

The digitalization of asset management within the architecture, engineering and construction (AEC) sector is in need of effective methods for the automatic classification of documents. This study focuses on the development and the evaluation of multimodal document classification models, utilizing visual, textual, and layout-related information. By using the CRISP-ML(Q) methodology as well as Neural Architecture Search, we examine various state-of-the-art machine learning models, and combine them through an iterative development process. The performances of these models are evaluated on two different AEC-document datasets. The results demonstrate that each of the modalities is useful in classifying the documents, as well as the integration of the different information types. This study contributes by applying AI techniques, specifically document classification in the AEC sector, setting the initial step to automating information extraction and processing for Intelligent Asset Management, and lastly, by combining and comparing multimodal state-of-the-art classification models on real life datasets.

Author's Declaration

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Twente to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Twente to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Floor Rademaker

CONTENTS

LIST OF FIGURES

List of Tables

1

INTRODUCTION

The Architecture, Engineering and Construction (AEC) sector is increasingly embracing digital technology and data, marking a significant shift brought about by the fourth industrial revolution. Key developments in this field include the implementation of algorithms for learning from big data, the enhancement of productivity through the automation of simple tasks using artificial intelligence (AI), and the application of AI to tackle more complex problems with agents, bots, and models [6]. With these developments, we can observe the growing importance of information management. Within the AEC industry, this revolution is known as *Construction 4.0* [7]. Specific technologies of construction 4.0 include Building Information Modeling (BIM) for modeling and simulation, AI solutions, robotics and automation, and the use of sensors. For asset management applications, this shift enables the digital management of coordinated assets, known as Intelligent Asset Management (IAM) [8].

IAM offers opportunities, such as the use of digital twins to monitor, analyze, and optimize the performance of physical assets or processes, or use of sensors for real-time monitoring and analysis of physical assets [8–10]. The emergence of IAM systems and techniques has introduced a new maintenance strategy named predictive maintenance. Unlike corrective maintenance, which is performed after defects are detected, or preventive maintenance, which is scheduled, predictive maintenance uses data analytics and monitoring to predict when maintenance should be performed just in time to prevent issues [11]. Taking this approach further, prescriptive maintenance involves an AI system recommending actions and decisions based on predictive maintenance data [12]. In addition, mobile maintenance is gaining importance [13]. It enables maintenance teams to use mobile devices to receive tasks, log activities, track parts and inventory, and access asset data while on the job, thus increasing efficiency and improving data quality. This increasing reliance on digital systems imposes the need to extract, structure, and utilize both existing and historical asset data, much of which has to be extracted from unstructured documents.

The adoption of IAM presents challenges that require changes in both technological infrastructure and asset management processes. The IAM strategy should align with the company's digital strategy [14], with a specific focus on data-driven decision making. Adapting to and utilizing the advantages that industry 4.0 brings for asset management allows companies to be more responsive to the market. Many companies are eager to implement IAM systems to improve their efficiency. However, with the current state of operations in most companies in the AEC sector, many steps must be taken to digitalize asset management strategies [15].

In the Netherlands, the adoption of digital technologies in asset management is progressing, but there are still various challenges. According to a report by PwC and Mainnovation in 2023[1], 39% of companies in Northwest Europe have implemented mobile maintenance, while predictive maintenance lags behind with only 17% of companies. Other digital trends in asset management, such as digital twins, augmented reality, and 3D printing, have only been implemented by 8-10% of the companies.

Movares[2] is an advisory and engineering firm specialized in infrastructure, mobility, digital transformations, climate adaptation, energy transition, and circular building. Within their asset management team, the goal is to help companies transform their asset management strategies and systems to become more data-driven and digital.

The degree to which companies have transformed to more data-driven asset management differs by company and asset [8]. The newer and more valuable assets are more likely to have centralized and accessible data within an asset management system. Older assets often do not have centralized data, with relevant information scattered across various documents and systems. To digitalize these assets, the first step is to locate and extract this information. In the AEC sector, documents have not changed much over the years, although the information technology around them has made some fundamental changes [16]. Different Intelligent Document Processing (IDP) techniques can be applied to process documents and extract information such as Optical Character Recognition (OCR), Robotic Process Automation (RPA), image processing, classification, and Natural Language Processing (NLP) techniques [17–19].

Recent developments in IDP have changed the way companies manage their documents. Utilizing AI and Machine Learning (ML), information can be automatically extracted from documents [18, 19]. OCR is used to extract data from documents, which can then be further processed, as described in prior work [20–36]. In addition, ML techniques are utilized to classify, cluster, and extract information from documents. Finally, RPA is implemented to automate repetitive tasks and workflows [37].

In asset management, particularly within the AEC industry, various IDP techniques are applied in different contexts. These include extracting information from engineering drawings, models, and floor plans through text extraction, object detection, and segmentation [38–47]. Named entity recognition and NLP techniques are utilized to automate cost estimation in AEC projects

[1] PwC report on digital trends in maintenance and asset management
[2] Movares Website

[48], and to enhance the automatic understanding of geotechnical texts [49]. Automatic classification of asset documents is a crucial first step in digitizing asset information. Without such classification, relevant data cannot be extracted from the specific document structures of data, hindering effective use in IAM systems. This way, classification sets the first steps for data-driven decision-making in the AEC sector.

Document classification has been widely studied using various methods that focus on different information modalities, such as textual, visual, and layout data. Over the years, there has been a notable evolution in these methods. Initially, manually extracted single-modality features were used in relatively simple classifiers. However, the influence and application of deep learning models have increased in recent years [2, 50–53]. In addition, more hybrid models have been developed and evaluated, utilizing a combination of modalities to achieve superior classification results [28, 31, 35, 54–57].

Despite extensive research on document classification, most studies have focused on standard datasets. There is limited research on the classification of AEC-related documents, highlighting the need for research to optimize the classification of these specific documents. As mostly text-based classification has been performed in the AEC industry, the remaining modalities are used for classification, as well as combined into hybrid classification model. The goal of this research is to find a well-performing classification architecture for AEC asset documents, by evaluating state-of-the-art document classification architectures. To do so, we find the answers to the following research questions.

1.1. RESEARCH QUESTIONS

This study aims to develop an effective document classification approach to help automatically and effectively digitize AEC asset data. The main research question is:

> **RQ1:** How can the digitization of AEC information be improved by constructing a document classification architecture?

To answer this question, the following subquestions are answered:

- **RQ1.1:** How do state-of-the-art classification models perform at classification of asset related documents?
- **RQ1.2:** How do combinations of document modalities impact the performance of a classification in terms of accuracy?
- **RQ1.3:** How can the classification model be best deployed to make the classification model generally usable to classify AEC documents in the future?

To guarantee a clear and repeatable development process, this research uses two methodological frameworks. The Cross-Industry Standard Process for Machine Learning (Quality) (CRISP-ML(Q)) methodology [4] is used for the iterative development of the classification models, where we focus on data understanding, quality and evaluation of the models. We automate the selection of the optimal model architecture by using Neural Architecture Search (NAS) [58]. Together, these methodologies ensure a systematic and repeatable process.

To address the limitations outlined above, this thesis proposes a classification approach for asset management that uses the best-performing (multi-)modal features. The research applies and evaluates state-of-the-art document classification and machine learning techniques in the context of the AEC industry, an industry where the application of AI-based IDP is relatively underexplored. Specifically, this study explores multi-modal document classification by comparing various model architectures that combine visual, textual, and layout-based features. Using two real-world asset document datasets, this study analyzes the impact of modality fusion and preprocessing techniques on classification performances.

This thesis contributes to both academic research and professional practice. Scientifically, this study is, to the best of our knowledge, the first study to extensively evaluate such multi-modal and single-mode classification models in the AEC domain. This way, it addresses a gap in existing literature by demonstrating how these state-of-the-art models perform in a domain where document types, technical language, and layout significantly differ from general datasets. In practice, it offers actionable insights into how organizations in the AEC industry can use machine learning and information extraction techniques to digitalize legacy asset data, supporting more efficient and data-driven asset management strategies.

The rest of this thesis is organized as follows: First, we evaluate the latest research in the field, which we detail in the Systematic Literature Review (SLR) in Chapter 2. We describe the methodological frameworks utilized in this research in 3. Chapter 4 covers the experimental setup, followed by the presentation of the results in Chapter 5. We link back the found results to our initial research questions as well as discuss the found results in 6.

2

Literature Review

Within the AEC industry, as in many other industries, information is of great importance. This industry is highly data-intensive as construction projects produce substantial amounts of documentation. However, this documentation is not always utilized. Specifically, a vast section of this data is encapsulated in documents, which are stored throughout different systems in companies and projects. These documents are often structured in different ways, making it difficult to retrieve information from them. The large volume and complexity of these documents require an automatic method of retrieving the information.

The first step in this automation is document classification, which is a branch out of many other intelligent document processing practices [35, 59]. Document classification is the automatic categorization of documents. In the AEC industry, such categories could be reports, maps, architectural drawings, contracts, insurances, schemes, etc. The textual content of some of these documents is limited, as visual elements may have a more crucial role. Therefore, rather than document classification in general, this research focuses on document image classification, where documents are automatically categorized based on their visual features. Furthermore, multimodal document classification, where different document modalities are used to classify a document, is deemed relevant to construction document classification as well. Different modalities could be visual features, textual features, or even layout features, combined into a hybrid classification model.

From the objective of the review of first evaluating the developments of document image classification in the AEC industry, and secondly evaluating the document image classification techniques developed and utilized in the last five years, the following research questions emerged.

2.1. RESEARCH QUESTIONS

1. What document classification techniques have been applied specifically in the fields of the architecture, engineering and construction industries?
2. What document classification methods have utilized been in general in the last five years?

The found document classification techniques utilized in the fields of architecture, engineering, and construction contribute to an exposition and comparison of the techniques used within the sector. In this way, we are able to identify and reuse effective techniques while also learning from past mistakes.

The second objective aims to conduct an extensive exploration of all document classification techniques, not just within the AEC industry, but across various fields, however, more specifically in computer science practices. This exploration allows to set out various techniques, and model architectures that contribute to better classifying systems.

This literature review provides a comprehensive overview of current developments and applications of document classification in the AEC sector. In addition, it presents an up-to-date overview of developments in document classification models generally. By identifying gaps and opportunities in existing literature, it guides future studies and innovations in document classification. Additionally, this work contributes to the broader fields of machine learning and AI as well as their application in the AEC sector.

This study is carried out as an SLR based on the procedure proposed by Kitchenham [60]. First, the scope of the research is laid out. Then we elaborate upon the review methodology, specifying the way the material is collected and selected. In the next chapter, we go further into the literature by describing the sample of materials found in detail and performing a meta-analysis. This way, we evaluate the quality and relevancy of the materials. Quality is measured by evaluating the source of each article, based on factors such as the reputation of the journal and the credibility of the authors.We assess relevance by reading the articles and determining how closely they align with the research topic, based on their focus, approach, and findings. After that, we go further into the materials to extract and evaluate the information in a structured way.

2.1.1. SCOPE OF THIS REVIEW

The focus of this review is both document classification in the architecture, engineering and construction industry, as well as the techniques used for document image classification across all fields. The AEC industry mentioned involves all companies in the building and engineering sectors for both infrastructure and buildings. All types of document classification techniques discussed in this sector have been published in the last five years, as they are deemed to be the most relevant and recent. The documents used in this sector are generally different from other sectors; however, similar for various projects and applications in terms of document types [61]. Furthermore, construction documents are often unstructured and come in various formats, including both highly textual and visual elements. This diversity complicates classification, as a single modality will not always yield the most accurate results.

In order to accommodate the classification of these diverse documents, we explore document image classification and multimodal document classification techniques specifically. We focus on the last five years to evaluate the utilized methods, which we expect to have been developed through a process of continuous improvement, therefore increasing performance over time. In addition to the objective of evaluating only the most recent techniques, a comprehensive review of the progress that has been made in the field of document image classification from 2001 to 2021 (Liu et al. [1]) was found in the initial state of material collection. As this paper brings a structured and clear overview of the developments before 2021, we build further upon this review, and mostly consider the materials published after and thus not covered by Liu et al.

2.2. Review Methodology

2.2.1. Material Collection

This section describes the details of the material collection. We specify the keywords used for the search of the materials and the additional criteria for selection. We base the keywords directly on the research questions with the aim of finding only materials relevant to these questions. We retrieve the exact query of keywords through an iterative trial-and-error process where keywords are added and removed to find the most appropriate search query.

We use Google scholar [1] and Scopus [2] as the search databases for the materials. As the search results for the searched queries are of a quite large volume for both search queries, the materials to include have to be further specified upon. Therefore, some inclusion criteria are established. These criteria ensure that the source materials are of quality, reliable, and relevant. The inclusion criteria are specified as follows:

1. The article is available for free.
2. The article is published in a quality English journal or book.
3. The article is published in the year 2019 (first RQ) / 2020 (second RQ) or later.

The material collection and selection process is illustrated in Figure 2.1. As mentioned in section 2.1, we measure the quality of an article by evaluating the source of each article. The evaluation is done based on factors such as the reputation of the journal and the credibility of the authors. We assess the relevancy of an article by reading and determining how closely they align with the research topic based on their focus, approach, and findings.

First research question

The main goal of the first research question was to find all techniques used and applications of document classification in the AEC industry. Only including "architecture", "construction",

[1] Google Scholar
[2] Scopus Website

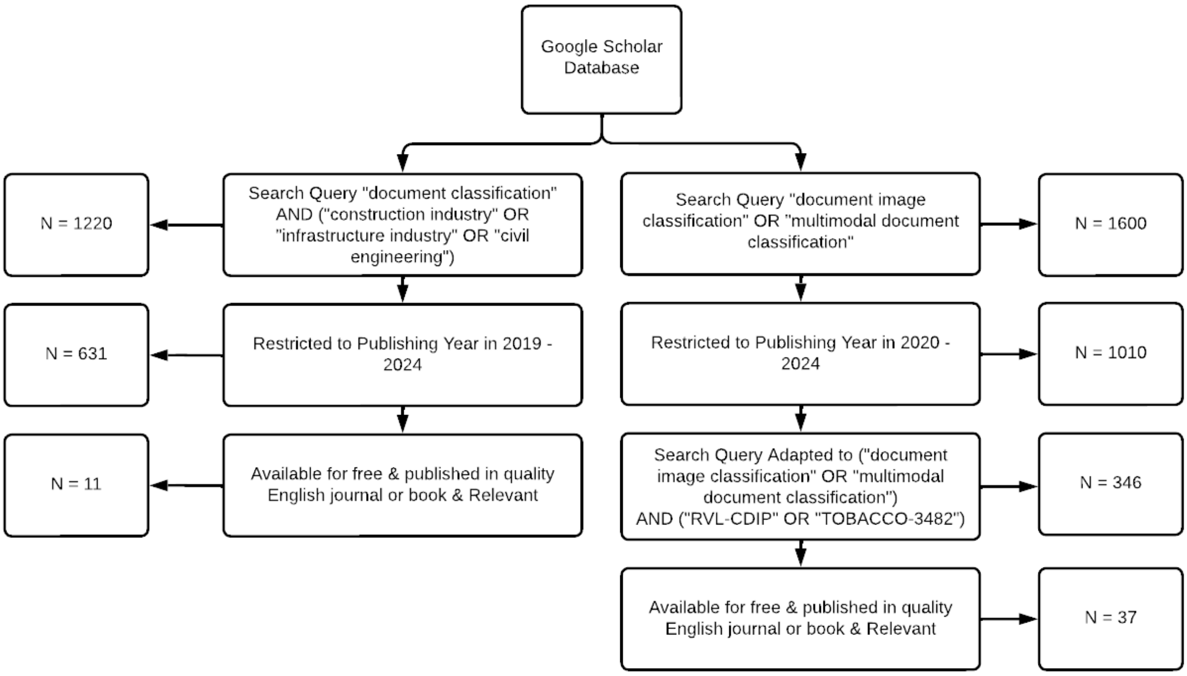

Google Scholar Database
Search Query "document classification" AND ("construction industry" OR "infrastructure industry" OR "civil engineering")
N = 1220
Restricted to Publishing Year in 2019 - 2024
N = 631
Available for free & published in quality English journal or book & Relevant
N = 11
Search Query "document image classification" OR "multimodal document classification"
N = 1600
Restricted to Publishing Year in 2020 - 2024
N = 1010
Search Query Adapted to ("document image classification" OR "multimodal document classification") AND ("RVL-CDIP" OR "TOBACCO-3482")
N = 346
Available for free & published in quality English journal or book & Relevant
N = 37

Figure 2.1: Material Collection & Selection Process

"engineering", or "AEC" with "document classification" in the search query resulted in 18800 results. Architecture, construction, and engineering could all be relevant to other IT applications as well, e.g. in the sense of constructing a system or building an IT architecture. Therefore, we needed to add "industry" to the search terms. As we also consider civil engineering applications relevant, we added the term to the search query as well. For the first research question, the following query resulted from the keyword specification process:

> "document classification" AND ("architecture industry" OR "construction industry" OR "engineering industry" OR "AEC industry" OR "civil engineering")

Searching for this query and applying the third inclusion criterion yield 678 results. By applying the first and second criteria and after determining which materials are relevant, only 11 research papers remain.

SECOND RESEARCH QUESTION

For the second research question the objective was to find all developments in the field of (automatic) document classification. The goal is to find what the state-of-the-art models are and what techniques have been used. First, the query was only specified on "document classification", which resulted in many results in text document classification. Document classification based on text is not irrelevant, however, since construction documents often include drawings and maps, the visual aspect is also crucial. Therefore, instead of just focusing on text document classification techniques, we also consider methods that address visual elements.

We decided to focus on both "document image classification", where only the visual aspect is used for classification, as well as "multimodal document classification", where multiple modalities, e.g. visual and text, or layout and text, are used for classification, mostly in order to achieve better classification performance. This resulted in the following search query:

> "document image classification" OR "multimodal document classification"

Searching this query and applying the third inclusion criterion results in 1010 search results. After initial selection by reading titles and abstracts, we quickly discovered that the two datasets "RVL-CDIP" and "Tobacco-3482" (see Section 2.3.3) are both used for performance comparison of the classification models, especially in recent models [3, 26, 27, 29]. As the models provide a fair comparison, we decided to merely select the papers using either of these datasets to measure performance, so the new search query became:

> ("document image classification" OR "multimodal document classification") AND
> ("RVL-CDIP" OR "TOBACCO-3482")

. From this search and application of the third selection criterion, the number of results amounts to 339. Two types of less relevant articles are identified: duplicates and articles that discuss the specified datasets in the related work section without actually performing classification using the datasets. We exclude these articles from this literature review. By snowballing through the initially found materials, additional materials are selected. The materials are combined and inclusion criteria one and three are applied. By filtering the materials to be only in English and in a quality journal, 38 materials are selected.

2.3. RESULTS

2.3.1. RESEARCH SAMPLE AND META-ANALYSIS

This section aims to further investigate the selected materials. This way, the materials are compared from a meta-level perspective as well as examined on quality. The aim of is to structure the found literature based on meta information rather than just focusing on the contents. This high-level overview of the materials is structured by answering the following questions.

1. When were the articles published?

2. What keywords have authors used to categorize the materials?

3. In what journals have the materials been published?

As we only select articles published in the last five years, as part of the inclusion criteria, the years from 2020 to 2024 are included. The materials for the different research questions are analyzed separately in order to give a more detailed meta-analysis.

For the first question, the years the materials were published in are analyzed. The number of articles published each year is plotted against their respective publication years. In Figure 2.2 can be seen that the number of published articles for document classification in AEC applications has been fairly equal over the last five years. On average, two articles are published in this area, which corresponds exactly to the amounts published in three of the five years.

For articles writing on (multimodal) document image classification, an evident increase in the number of publications can be observed each year. Interestingly, the number of publications in 2021. The decrease in publications in 2024 can be attributed to the time of material collection, which was around October 2024, after which possibly new materials were published.

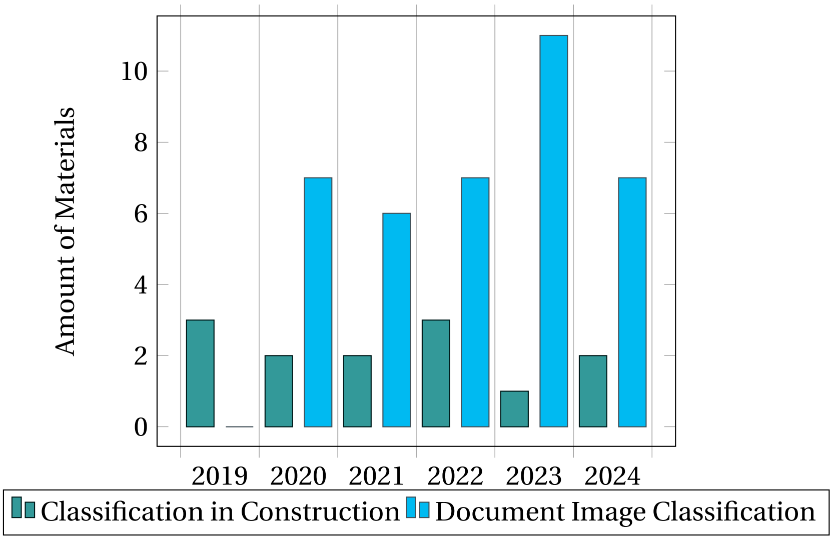


Figure 2.2: Number of Publications per Year

Further, by analyzing the keywords used by the authors to categorize their works, it can be determined what main subjects are discussed, and possibly what techniques are used in the selected research material corpus. Again, in order to separately analyze the keywords for the two research questions, keyword density word clouds (see Appendix A.1 & A.2) have been created to visualize the keywords used in the respective areas (see Figure A.1 & Figure A.2). The size of words is dependent on their "density", i.e. the number of times they were used to describe an article. Colors and fonts are used independently.

In the first word cloud all keywords are included. The words with largest density are *Text Mining, Optical Character Recognition, Deep Learning, Machine Learning* and *Document (Image) Classification.*

Secondly, for (multimodal) document image classification the keywords most dense in use are *Deep Learning, Document Image Classification, Document Classification, Multimodal Classification* and *Convolutional Neural Network.* As this word cloud visualizes the keywords used by a significantly larger number of articles compared to the first word cloud, a threshold of at least two occurrences was applied.

Lastly, to evaluate the quality of the articles, the journals in which they have been published are evaluated. For document classification in AEC applications, all journals are different, but the materials come primarily from journals in the area of information technology and AEC engineering.

For the articles selected for (multimodal) document image classification, all journals can be regarded as in the area of information technology, more specifically in the areas of machine learning, neural information processing, computer vision, and pattern recognition. From six different journals, more than one article is selected. Abbreviated, these journals are the ACMMM [3], CVPR [4], ICDAR [5], ICPR [6] and IJDAR [7]. Furthermore, 10 of the selected articles have been published directly on ArXiv [8]. As the materials on ArXiv are not peer-reviewed, we should take a critical stand towards the materials and information in them. To ensure that the Arxiv retrieved articles are as most reliable as possible, we analyze the authors. The article is removed if the author is not in academics, has a research-based function in a company, or is not researching data science or machine learning-related topics regularly. Having applied this filtering step does not mean that we can neglect the critical view to these articles.

Two tables in which the publication counts per conference/journal are listed are included in the appendix (see tables B.1 & B.2).

2.3.2. Document Classification in the AEC Industry

In the last five years, several articles have been published on various applications of document classification in AEC applications. This section discusses the details of the applications as well as the techniques used.

Noteworthy is that most applications of document classification in the AEC industry are primarily based on text classification. Bodenbender et al. [62] classify real estate documents by primarily using text classification techniques. A range of machine learning algorithms are applied to the text extracted from building documentation to automate the classification process. Secondly, Sajadfar et al. [61] use OCR techniques to detect text in construction documents. They apply a long-short-term memory model, as well as keyword-based methods to classify the documents. Kim et al. [63] propose a model for classifying construction disaster documents based on text data. The model is a Convolutional Neural Network (CNN) and the text is extracted using the Term Frequency-Inverse Document Frequency (TF-IDF) method. Ren et al. [64], as well as a self-constructed corpus in the field of construction, use a Bi-LSTM model that uses attention mechanisms for document classification of construction using text classification. In 2020, Guha et al. [65] constructed a document classification model for property-related documents in real estate. The classification model, like Kim et al., uses the TF-IDF vectorization for text classification, where the text is extracted using OCR techniques. Sun et al. [66] have created a framework whose goal it is to help managers have a quicker and easier understanding of key information in construction documents. Again, the TF-IDF algorithm is used to find the most important information, however, not specifically for document classification. Con-

[3] ACM Multimedia
[4] Conference on Computer Vision and Pattern Recognition
[5] International Conference on Document Analysis and Recognition
[6] International Conference on Pattern Recognition
[7] International Journal on Document Analysis and Recognition
[8] ArXiv

struction risks are evaluated by Kang et al. [67] through the use of text mining for unstructured data in construction project documents and a support vector machine was used for classification in order to improve risk management. Lastly, Wang et al. [68] classify defect texts and simultaneously try to improve the interpretability in construction management decisionmaking by applying SHAP-based interpretability methods. Jacques de Sousa et al. [69] presented a systematic review of the literature on the developments of artificial neural networks and neural language processing in the field of text document classification for the budgeting phase of construction projects. The work mainly deems it necessary to develop datasets in the field of construction, to be able to further develop automation and classification algorithms in the field.

As mentioned before, all of these described materials primarily use text mining techniques to classify construction documents. The TF-IDF technique is used more often, and the different applications of document classification serve diverse goals. Alongside these materials, a small number of materials is left, covering other aspects of document classification in the AEC industry.

As this SLR mostly focuses on document image classification, these earlier applications are taken into regard but not directly applicable to the case of this research. However, two of the articles found do not only make use of textual features but also include image features. Borst et al. [70] use natural language processing techniques and an EfficientNet to utilize visual information in the form of images for document classification. These features are combined into a hybrid classification approach, that is, a knowledge graph structure is proposed to store the information found. The classification is based on the graph, which represents the textual and visual information through its nodes and edges. TechDoc [52] is a multimodal deep learning architecture that uses textual and visual features. In addition, associations between documents are taken into the classification as well. The text features are A recurrent neural network learns the text features while for the image features a CNN is used, and a graph neural network learns the associations among documents.

In conclusion, the predominant body of document classification developments in the AEC industry makes use of text mining/classification techniques. Small interest is also peaked in the addition of visual information, where both applications combine textual and visual features into a graph representation, which is used as input for the classification. As the aim of further research is to design a well-performing document image classification model, it is necessary to evaluate developments outside of the AEC industry as well.

2.3.3. (MULTIMODAL) DOCUMENT IMAGE CLASSIFICATION

Document image classification techniques have been researched for over a decade. First, features were mostly handcrafted, and based on mostly document structures and/or visual aspects. These models performed well in certain scenarios using research-dependent datasets, however, the real improvements came with the introduction of deep learning in the area of document image classification.

In 2014, Kumar et al. [71] first published a work in which a CNN was used for document image classification. Before this, features used were mostly related to the structure of a document, using either template matching or graph matching, or visual features were extracted "manually". The use of a CNN was motivated by the hierarchical structure of documents, as CNNs effectively capture and process features from multiple levels as well as different patterns. The work used the Tobacco-3482 dataset [9], as well as the NIST tax-form dataset [10] to test the classification performance of the network, and achieved an overall accuracy of 65.35%. This was a large improvement to the classifications performed before, as the highest accuracy achieved classifying the Tobacco-3482 dataset without using a neural network was 43.8% [72].

From the rather successful first application of CNNs in 2014, many efforts have been taken to further improve neural networks for document image classification. Within the next six years, many networks were constructed, creating more accurate classifications over the years. The Tobacco-3482 dataset became one of the most used datasets in the research on document image classification. In 2020, a classification accuracy of 99.71% was achieved classifying the Tobacco-3482 dataset by Bakkali et al. [26], the highest performance accuracy ever reached classifying this dataset.

In 2015, a subset of the large IIT-CDIP[11] dataset was subtracted and used for document image classification. This extracted dataset is called RVL-CDIP[12] and is ten times larger than the Tobacco-3482 dataset, substantiating a more robust foundation for training a classification model [2]. Using a CNN, an initial accuracy of 89,8% was achieved. Afzal et al.[51] used different deep CNNs to train a model on the RVL-CDIP dataset. The highest accuracy of 90.97% was achieved using a VGG-16 network. Further, the dataset was used for pretraining as well, improving the classification performance on other smaller datasets, such as the Tobacco-3482 dataset, tremendously.

Even faster than for the Tobacco-3482 dataset, network developments lead to many improvements in the classification accuracy for the RVL-CDIP dataset. In 2020, Bakkali et al.[26] combined visual and textual features in a two-stream neural network. This joint learning approach outperformed all state-of-the-art networks, as a classification accuracy of 97,05% was reached classifying the the RVL-CDIP dataset.

In 2021 a comprehensive literature review by Liu et al. [1] was published. In this review, all document image classification methods constructed until the moment of writing were compared in terms of classification accuracy. The specifics of features and networks were described, laying out the research landscape of document classification. Figure 2.3 shows the classification accuracies of the classification networks discussed by Liu et al., labeled by the set the networks

[9] Tobacco-3482 Dataset
[10] NIST Dataset
[11] IIT-CDIP Dataset
[12] RVL-CDIP Dataset

are trained and/or tested on.

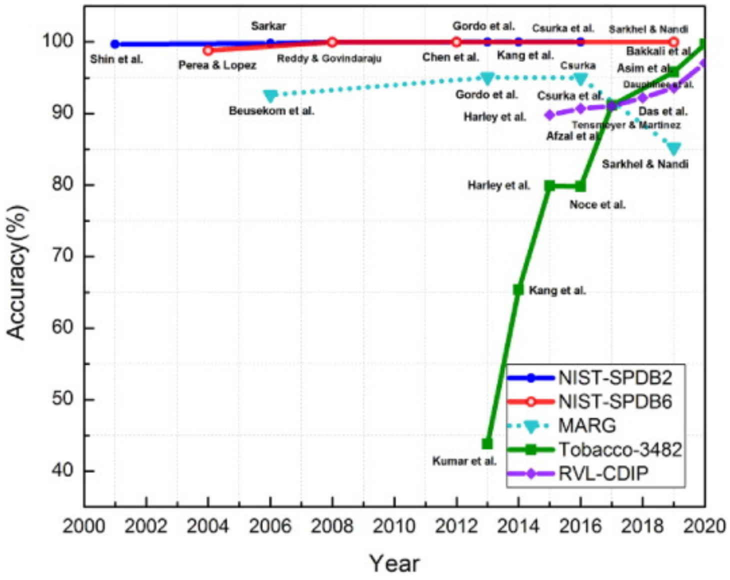


Figure 2.3: Model Accuracies for Most Popular Datasets (2000-2020) [1]

DOCUMENT CLASSIFICATION DATASETS

The review by Liu et al. [1] describes two datasets as main benchmarking datasets, the sets being the ones discussed in earlier section (See section 2.3.3): RVL-CDIP and Tobacco-3482. Research uses both of these datasets predominantly in research for testing the performance of document image classification models and therefore also used as the benchmarking datasets in this literature review.

RVL-CDIP The RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing) [2] dataset consists of a set of training images, validation images, and test images, respectively consisting of 320,000 images, 40,000 images and 40,000 images. The dataset consists of 16 classes, where every class consists of 25,000 images per class. Examples of classes in the dataset are letter, form, email, resume, and memo. Examples for each class can be seen in Figure 2.4. The dataset is a labeled subset of the IIT-CDIP collection, and subtracted specifically for training new CNNs for document analysis.

Tobacco-3482 The smaller dataset Tobacco-3482 consists of 3482 grayscale images, distributed over 10 classes. These classes are ADVE, Email, Form, Letter, Memo, News, Notes, Report, Resume, and Scientific. Examples for each class can be seen in Figure 2.5.

DOCUMENT CLASSIFICATION METHOD CATEGORIZATION

As section 2.3.3 highlights, many developments have been done in the field of document image classification in the last 10 years. Especially using CNNs, rather accurate models have been constructed.

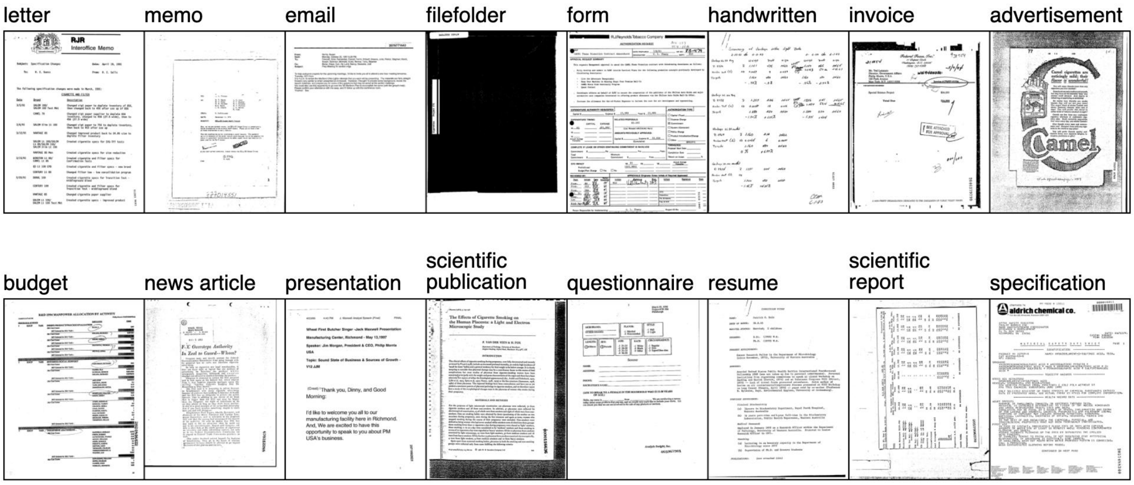


Figure 2.4: Sample of RVL-CDIP dataset. Figure by [2]

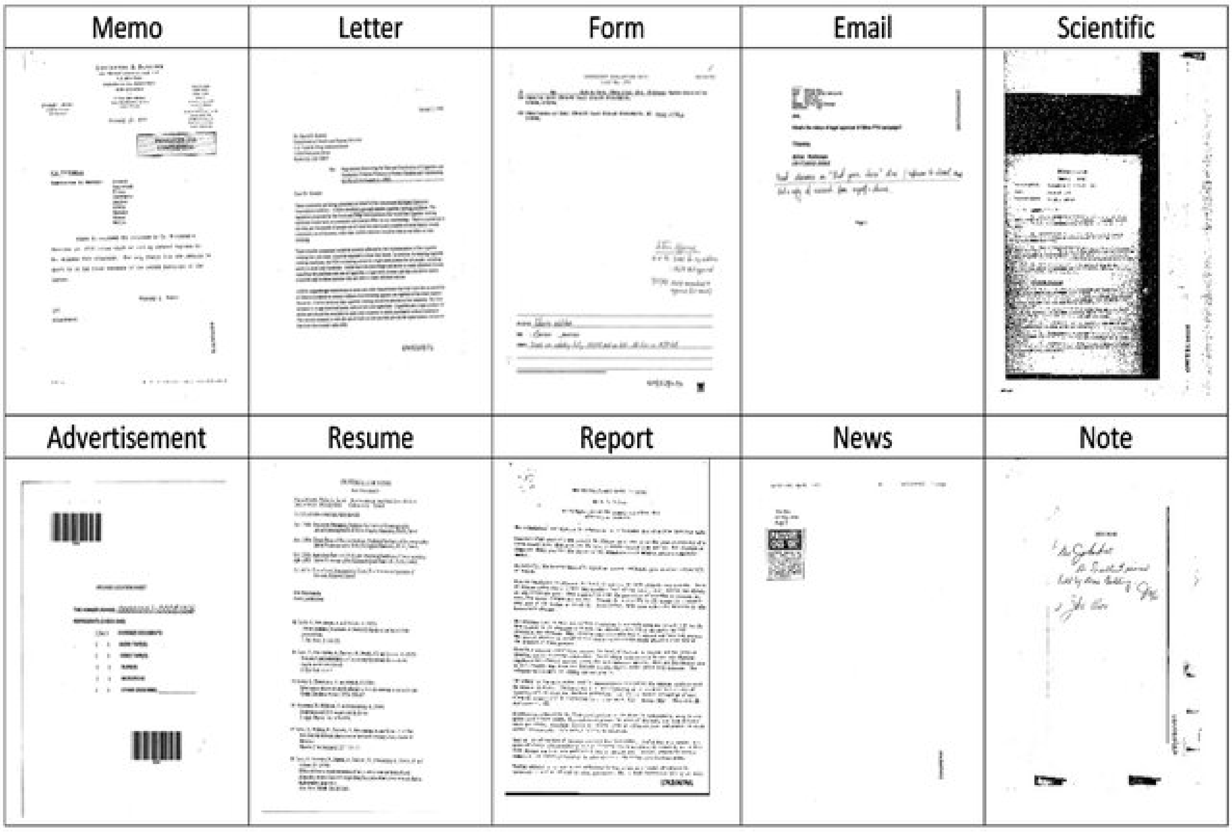


Figure 2.5: Sample of Tobacco-3482 dataset. Figure by [3]

Liu et al. [1] provide a categorization method of document classification methods, where the evaluated classification methods were categorized into four categories with subcategories. The four overarching categories with subcategories being the following:

1. Structural-based Methods
    - Template matching-based methods

- Graph matching-based methods

2. Visual-based Methods

    - Handcrafted feature-based methods
    - Deep feature-based methods

3. Hybrid Methods

    - Textual & visual-based methods
    - Textual & structural-based methods

4. Textual-based Methods

The first three method categories are regarded as document image classification methods, disregarding the textual-based methods as those only focus on text retrieved from the documents.

Liu et al. [1] review document image classification methods developed from 2000 to 2020, categorizing them into the previously discussed (sub)categories. Over the years, the types of classification methods have evolved: in the first seven years structural-based methods were primarily used. For the next ten year, handcrafted visual features dominated. In 2014, deep visual features began to be utilized, after which the development of hybrid models started rapidly. Until 2021, these hybrid models mainly utilized the textual and visual modalities of documents. In the subsequent years, improvements in classification models continued within the subcategory, and simultaneously three new subcategories emerged, combining the three modalities in various other ways. The second predominantly used combination of the modalities is all three modalities simultaneously, a subcategory not covered by Liu et al.

In terms of performance accuracy, the classification models achieving the highest performance accuracy by classifying either of the Tobacco-3482 and RVL-CDIP datasets were both developed by Bakkali et al. [26][27], which combine textual and visual features. As shown in Figure 2.6, the initial accuracy for models tested on Tobacco-3482 was 43.8%. However, this performance quickly improved over the years as deep feature-based methods and hybrid methods, which integrate both visual and textual features, were introduced. Interestingly, the performance accuracies of models using handcrafted visual features and those using deep visual features did not differ significantly. It was only with the introduction of hybrid models that classification performance improved significantly.

In order to understand why certain models are able to elevate the classification accuracy to a higher level, it is important to understand what the different models are built up like, and of what nature the features on which the classification is based are.

In this literature research, we use the document image classification methods categorization designed by Liu et al. to categorize the classification methods constructed in the past five years, in which 2020 overlaps partly with the included methods in the previous work. Additionally, some of the articles published in 2019 are taken into account as well, as for Tobacco-3482 the

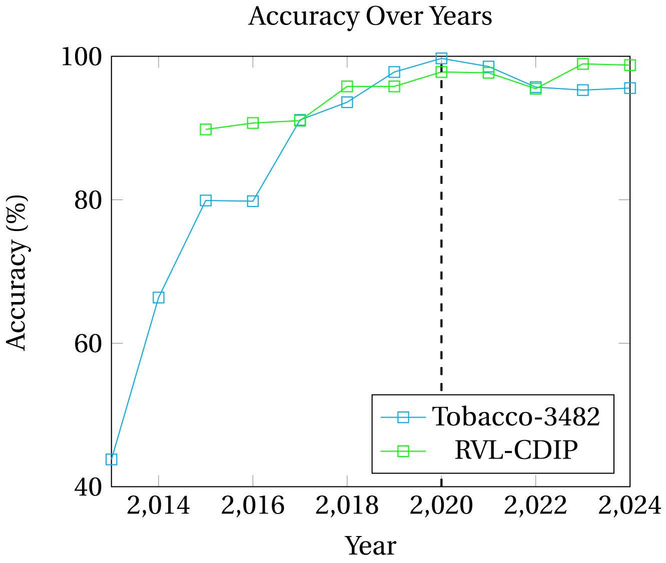


Figure 2.6: Highest Classification Accuracy per Year for Datasets Tobacco-3482 and RVL-CDIP, combining materials discussed in this review, and materials discussed by Liu et al. [1]

most accurate accuracy has been achieved in 2019 already, which can be seen in Figure 2.6. The accuracy for the RVL-CDIP dataset has still improved after 2020 and the state-of-the-art was constructed in 2023, with an accuracy of 98,94%, using a compressed version of the documents [3]. However, as can be seen in Figure 2.2, in the years after 2020, many more articles writing about new classification methods using the RVL-CDIP and Tobacco-3482 as test sets have been published, even though not necessarily improving on the classification accuracy directly.

We examine the categorization of the classification models as described in the included materials. The types of networks and networks structures used are discussed in more detail.

For the hybrid, i.e. multimodal, document classification methods, the fusion strategy is evaluated. This fusion structure can either be an early, hybrid or late fusion strategy. When applying the first strategy, the features are merged and evaluated at once through the neural network. The other possibility is a late fusion strategy, where the modalities are classified through different network structures, often specifically applicable to the specific modality, after which the individual classifications are combined into one classification by e.g. averaging, or adding the classifications. Lastly, fusion methods can be of a hybrid nature, which is often the case for fusion methods based on an attention mechanism.

All fusion methods found within the discussed materials are listed (see appendix table C.2). For each fusion method is indicated whether the fusion is either performed early on or at the end of the classification architecture. For some of the fusion methods, the fusion is done at multiple steps within the network, therefore indicated as hybrid fusion. Furthermore, the complexity of the fusion method is indicated, often depending on whether the method is based on attention mechanisms and/or on adaptivity. The idea in attention mechanisms is to simulate attention in humans, where people tend to focus on specific points or things rather than a full artefact

[64]. A fusion strategy based on an attention mechanism generally weighs the importance of different inputs or different positions in the same input sequence (self-attention) in order to make a more accurate prediction or decision. In the case of document image classification, the application of attention mechanisms means generally that feature maps, showing attention, are applied [32]. Adaptivity of a fusion method is based on whether the fusion function adapts based on the feature importance, or relatively [25].

Further, in addition to the categorization, network structure evaluation, and fusion strategy, for each model the pretraining strategy is evaluated. Both visual, as textual, as structural methods can be improved through having some pre-knowledge, which is delivered through a pretraining method. The used datasets for pretraining as well as the pretraining tasks, and the improvement in performance that the pretraining brings are discussed.

Not only have optimizations been made in the characteristics described in the previous paragraph, there has been more attention towards the speed and size of the classification models, especially for training. Secondly, there is a growing interest and desire towards explainability in artificial intelligence and machine learning in general. In order to better understand why an artificial agent or machine learning model predicts or classifies a certain thing in a certain way, different techniques have been constructed. This review evaluates the methods applied to the field of document image classification. Lastly, the problem of imbalanced data, which often tremendously decreases the quality of classifications, has been taken into regards by certain evaluated classification models.

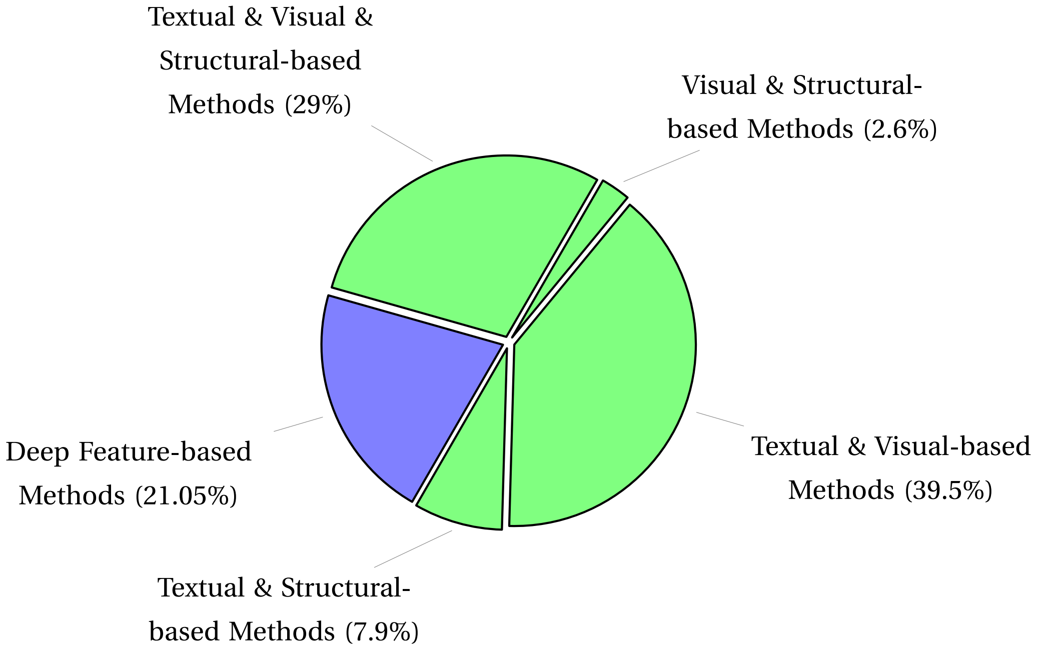


Figure 2.7: Pie Chart of Subcategorization Division

Using the earlier described categorization method (see Section 2.3.3, the categorization is made as can be seen in Figure 2.7. Only visual feature-based and hybrid feature-based models are

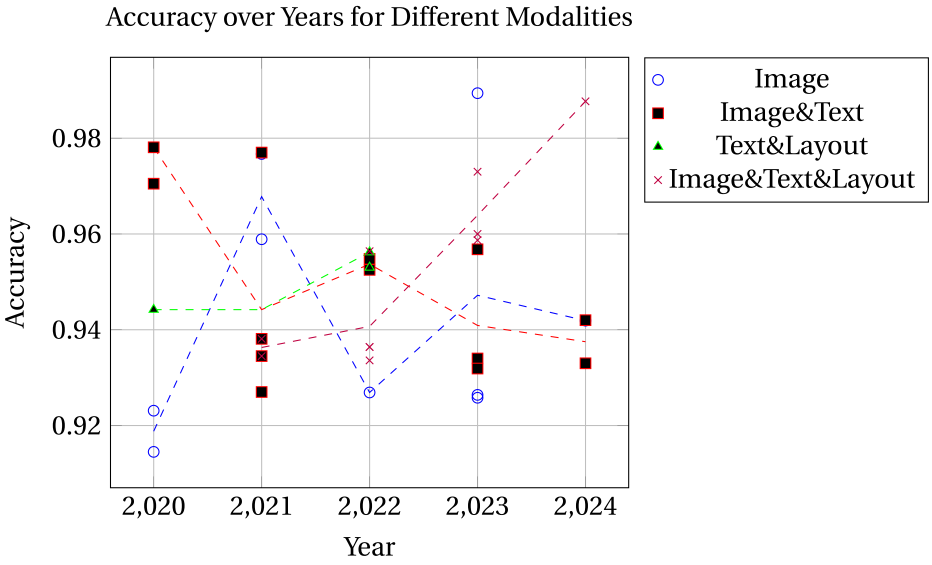


Figure 2.8: Scatter Plot of Accuracy Over Years for Different Modalities

included in this review, where the green part of the pie depicts the four kinds of hybrid methods and the blue part the two appearing kinds of visual feature-based models. The vast majority of the included materials are hybrid methods, as those have been constructed most in the past 5 years, show the best performance, and the most relevant structures.

Most constructed models in the included materials try to combine textual and visual based features, and many try to embed structural features as well. Then, it can also be seen that a visual feature-based method presumably uses deep feature-based method, rather than handcrafted features. Table 2.1 shows the specific categorization of materials into the defined subcategories as well as the developments in the categories over the years. Figure 2.8 shows the classification accuracies of all the models in the selected materials in the years their article was published, labeled by their respective category. The lines represent the category averages per year, showing the development in classification accuracy over the years. Figure 2.8 shows that over all the years the classification accuracies achieved vary widely between 90 - 100%. In 2023 and 2024, classification models combining all modalities seem to be performing the best, however, the classification performances of these classifications do not stand out significantly.

Table 2.1: Categorization of materials, which the described classification methods are based upon. Categorization categories based on Liu et al.[1]

Year	Visual Based (Deep Feature)	Textual & Visual	Textual & Structural	Visual & Structural	Textual & Visual & Structural
2020		[26],[29],[28]	[20]		[25],[55]
2021		[73],[32],[30]			[74],[75], [76]
2022		[31],[77]	[23]	[78]	[54],[22]
2023	[3],[79],[80],[81]	[34],[82], [83],[36],[33]	[24]		[84],[21]
2024	[85],[86],[87],[88]	[35]	[89]		[56]

In this section, all classification models are evaluated in more detail. In the case a model has a name, this name is used. Otherwise, the model is referred to by using the authors' name.

Visual-based Methods

The visual-based methods discussed are just 8, all utilizing a deep feature-based approach. [3, 79–81, 85–88] . The publishing years of these articles vary from 2022 to 2024, and the highest classification accuracy was achieved by the DWT-COMP [3] model, namely a classification accuracy of 98,94%. These deep visual feature-based models can be quite unclear in explaining why a certain classification was done. In recent years, there has been an increase in research on explainable document image classification, which is also the aim of a substantial part of the included materials, namely 5 of the selected articles [80, 81, 85–87].

In table 2.2 more than just the articles focusing on a visual-based method are included. Only five of the included articles in the table write solely about a visual-based method [3, 79, 81, 86, 88], and other included methods write about a method using multiple modalities, however comparing it to a visual-based baseline model, of which the performance has been included in the table. Some of the XAI models have not been included in the table as their objective was not primarily a high classification accuracy [80, 85], therefore not fairly comparing to the other models. The classification accuracies obtained using those baseline models have been included in the table. The articles using these baselines are discussed in their respective modality section.

Table 2.2: Classification Accuracies achieved classifying RVL-CDIP and Tobacco-3482 for Visual-based methods

Model/Author	**RVL-CDIP**	**Tobacco-3482 pretrained on RVL-CDIP**	**Tobacco-3482**
Bakkali et al. [26]			**96.25%**[1]
Bakkali et al. [27]	91.45%		
Ferrando et al. [29]	92,31%	94,04%	85,99%
EDNets [30]	95,89%		95,25%
Zingaro et al. [28]	97,67%		
DiT [79]	92,69%		
VLCDoC [82]	92,64%	89,73%	
GlobalDoc [34]	92,58%		
DocXclassifier [81]	94.17%	95,57%	90,14%[1]
DWT-CompCNN [3]	**98,94%**		92,04%
Sajol et al. [88]			92.25%[1]
DocXplain [87]	93.89%		94.71%
DocXclassifier [86]	94.19%	95.71%	90.29%

[1] Pretrained on ImageNet

We discuss three papers that use a solely visually-based document classification model, focusing primarily on classification accuracy. All three methods have a substantially different approach, using a transformer-based approach [79], a CNN approach [88], and an approach substantially different from all of the papers discussed in this review; a method based on discrete wavelet transform, which is further explained in the next section. This section describes the details of the three models one by one.

DiT (Document Image Transformer) [79] is a self-supervised pretrained model designed for general document AI tasks. The model preprocesses images of text documents by resizing them to 224 x 224 and then splitting the images into sequences of non-overlapping 16 x 16 patch embeddings. A transformer-based model is used as the backbone of DiT, where these image patches are processed through a stack of transformer-blocks. The use of multi-head attention allows the model to focus on different parts of the image simultaneously. The model is pretrained on the extensive IIT-CDIP dataset, which contains 42 million document images. Fine-tuning is performed on four different benchmarks, one being image classification. For this task, an average pooling layer is used, and classification is carried out using a simple linear classifier.

Another deep feature-based approach is adopted for the classification model designed by Sajol et al. [88]. They use ConvNeXt V2 [90], a high-performing deep CNN model and adapt it for document image classification. The research shows that pretraining on imagenet[13] can yield significant benefits for the performance. Several state-of-the-art models are compared in terms of performance, where ConvNext V2 proofs to be the best performing. The ConvNeXt architecture is a response to the introduction of Vision Transformers (ViT), which quickly started to outperform the earlier often used ConvNet architectures. The ConvNextv2, consequently, is the next iteration of the architecture. The ConvNeXtv2 architecture consists of a fully masked autoencoder framework and a new Global Response Normalization (GRN) layer, which is added to the earlier designed ConvNeXt architecture. This is done to enhance inter-channel feature competition. For the task of document image classification, this ConvNeXtv2 architecture performs rather competitively.

DWT-CompCNN [3] utilizes a substantially different method to classify the images, namely by extracting wavelet coefficients from JPEG 2000 compressed document images. The model utilizes *Discrete Wavelet Transform* (DWT) to break down the image into different frequency components. Discrete wavelet transform is a mathematical technique that is used to transform a signal, in this case a image, into different frequencies. These frequencies can capture information about spatial and frequency elements. An important benefit of DWT is that it can speed up the classification, while keeping the quality of images high. The frequencies captured are both high-frequency details as well as low-frequencies. This model comes close to a visual-based model using handcrafted features, as the initial extraction of wavelet coefficients can be seen as a way of handcrafted feature extraction.

XAI in document image classification In order to overcome the lack of transparency and interpretability in the deep learning document classification models, research has been done

[13] Imagenet Website

where techniques are proposed for making deep learning document classification models more transparent and interpretable. This field of research is fairly newly developing, and all of the published materials have been published in the last two years (2023 and 2024). Moreover, Saifullah et al. appear to be leading this field, having authored 4 out of the 5 published articles [81, 85–87]. This section goes over the five proposed XAI techniques for document image classification from the 5 last years.

One of the first XAI techniques designed for document image classification is designed by Saifullah et al; DocXClassifier [81] uses a ConvNeXt architecture [91] to extract visual features from the input document images. The most relevant parts of the images are highlighted through a feature selection process by using feature importance maps. This way, the model is made more interpretable instantly, as these importance maps show what the model assumes is relevant for the classification. An addition to this article is made a year later by the same authors [86], by further improving the interpretability of the model. A new improved model is created by integrating so-called feature pyramid networks, which are used to create feature maps at multiple scales within the architecture in order to improve feature maps quality and accuracy.

DocXplain [87] utilizes a fairly similar approach, as it similarly segments a document image into foreground and background segments, and assigns feature importance to the elements. This way, a model-agnostic attribution-based explainability method is presented, specifically for document image classification.

Fronteau et al. [80] present a technique that helps explaining models, but also improves the robustness of classification models. Their paper on adversarial robustness in document image classification models evaluates the effect of adversarial attacks on document classification models. Adversarial attacks are input in a machine learning purposely designed to perturb the models' predictions, e.g. by inputting data that resemble a certain class. Previous research focuses on various types of adversarial attacks as well as defenses. They are the first ones to research adversarial attacks in document image classification. They design defenses that minimize the effect of the attacks, and in this way improve the robustness of ResNet50 and EfficientNetB0 model architectures, which are frequently used model architectures in document image classification.

Lastly, Saifullah et al.[85] analyze the interpretability of state-of-the-art deep learning models that are used for document image classification. The popular interpretability method DeepSHAP is used for an approach that aims to present more interpretable explanations, as well as counterfactual explanations. This way, the most important document features can be analyzed. The article shows that many state-of-the-art models classify documents based on irrelevant features in the data, and learn counterintuitive document representations. These discoveries lead to the belief that analyzed models need improvements, as the models seem to learn shortcuts rather than really relevant classification information.

HYBRID METHODS

A total of 29 articles proposing hybrid/multimodal document classification models are selected from the last five years. In general machine learning literature, a hybrid classification model is a model which uses two or more machine learning algorithms to classify the input [92]. In this research, the hybridity of the described machine learning models stems not only from the number of machine learning algorithms used into the final classification architectures, but also from the combination of different modalities in the features used for classification. The identified modalities are textual, visual and structure, i.e. the language used, image features, and the layout of a document. As previously categorized, the materials can be classified into four distinct subcategories, where one of the categories (visual & structural-based) has not yet been identified by Liu et al [1]. Of all methods, 11 methods use all of the three modalities, 12 use just the textual and visual modality, 3 use a combination of textual and structural modalities, and just one uses the unique combination of visual and structural-based methods. The methods are evaluated per combination of modalities.

Textual & Visual-based Methods The 12 textual & visual-based methods are discussed in the following section. Something that almost all of the networks have in common is that the text of the images is extracted by utilizing OCR techniques by all of the models. Although most of the models employ Bidirectional Encoder Representations from Transformers (BERT) or fastText for text encoding, there is a clear variation in how visual features are processed and how the fusion of the textual and visual features is performed. First, we discuss all methods using OCR to extract the document text. We separate the materials based on the text encoder that is used; a BERT$_{\text{based}}$ encoder [26, 27, 29, 31, 32, 34–36, 82] or a fastText encoder[28, 30]. The first nine papers discussed use a BERT$_{\text{based}}$ text encoder to encode text extracted using OCR techniques, followed by two papers that use a fastText text encoder. Only one of the included papers in this section uses a non-OCR based text extraction method [33], which is discussed last. Table (2.4) shows the characteristics of the models, where the text elaborates on the model specifics.

Using a BERT$_{\text{based}}$ encoder The BERT$_{\text{based}}$ encoder tries to understand the context of words in a sentence by looking at the words around it. BERT$_{\text{based}}$ models are trained on a large text corpus and can be used for many different natural language processing tasks. For text classification, BERT is most often used by pre-processing the text into subword units, for which vector representations are created that consequently are encoded. These encodings are then fed into classification layers to predict text categories [93].

Bakkali et al. [26] present a deep cross-modal network that integrates textual and visual content extracted from document images. They compare three ways to combine the two modality branches, of which a function where features are added directly, maintaining the same dimensionality results in the most accurate classification, performs the best. The proposed model was tested using the Tobacco-3482 dataset and achieved the highest performance accuracy of all the discussed papers.

Table 2.3: Classification Accuracies achieved classifying RVL-CDIP and Tobacco-3482 for Textual & Visual-based methods

Model/Author	RVL-CDIP	Tobacco-3482 pretrained on RVL-CDIP	Tobacco-3482
Zingaro et al [28]	93,6%		90,5%%
Bakkali et al. [26]			99,71%
Bakkali et al. [27]	97,05%		
Ferrando et al. [29]		94,9%	89.47%[1]
EDNets [30]	97.81%		96.95%
EmmDocClassifier [31]	95,48%	95,7%	90,3%
EAML [32]	97,7%		98,57%
Structextv2 [33]	94,62%		
GlobalDoc [34]	94,04%		
VLCDoC [82]	93,19%		
Krithika et al. [35]			93.3%[1]
Voerman et al. [36]	Imbalanced dataset used, so not a fair comparison		

[1] Pretrained on ImageNet

Table 2.4: Visual-Textual Model Comparison for models where the text modality is handled through OCR extraction and a $\text{BERT}_{\text{based}}$ or fastText encoder

Model/Author	Text Encoder	Visual Modality	Feature Fusion
Bakkali et al. [26]	BERT	NASNet-Large	Superposing Function
Bakkali et al. [27]	BERT	NASNet-Large	Average Ensembling Method
Ferrando et al. [29]	BERT	EfficientNet	Average Weighted Ensembling
EmmDocClassifier [31]	BERT	EfficientNet-B0	Equal Concatenation
EAML [32]	BERT	Inception-ResNet-V2	Self-Attention-based fusion
GlobalDoc [34]	RoBERTa	ViT-B/16	Cross-Modal Attention-based Fusion
VLCDoc [82]	BERT	ViT-B/16	Cross-Modal Attention-based Fusion
Krithika et al. [35]	BERT	VGG16	Simple Concatenation
Voerman et al. [36]	BERT	VGG16	Attention-based Fusion
Zingaro et al. [28]	fastText	MobileNetV2	Weighted Concatenation
EDNets [30]	fastText	EfficientNet	Multi-View Deep Autoencoder

In the same year, the authors published another paper in which a model having a rather similar structure was described, however, now tested on the RVL-CDIP dataset [27]. The novelty of this paper, compared to the earlier one, is the comparison of backbone neural networks and word-embedding methods, for the framework branches of the two modalities image and text, respectively. Two late fusion methodologies have been adopted, after which the classification is performed using a SoftMax layer. The best performing configurations of the model conclude to a heavyweight NasNet-large model as the backbone neural network for the image modality and the $\text{BERT}_{\text{base}}$ model for the text modality, both tested as single-modality models. Merging the streams by using an average ensembling method - a method rather similar to the method used in their previous paper - boosts the performance of both single-modality models significantly.

Both Ferrando et al. [29] and EmmDocClassifier [31] try to speed up the training process. Ferrando et al. [29] propose configurations to accelerate training, and have designed a more efficient classification framework. Multiple GPUs are utilized to speed up the pretraining process. EmmDocClassifier [31] aims to find the actual benefit of pretraining using RVL-CDIP when using Tobacco-3482 as test set. They argue that only training using Tobacco-3482 would save time. As can be seen in table 2.3 does pretraining however really improve the performance. Further, they focus on improving the textual stream by incorporating a hierarchical attention network (HAN). The HAN divides the text in both sentences and words. Consequently, BERT is used to encode the text. The two modalities are combined to give an improved performance accuracy over earlier models.

The previous articles primarily utilize rather simple summation and multiplication methods to fuse the models and features at later stages in the process. EAML (ensemble self-attention-based mutual learning network for document image classification) [32], GlobalDoc [34] and VLCDoc [82] take a more hybrid fusion approach. EAML is designed to use a self-attention-based fusion module, and GlobalDoc and VLCDoc fuse the features through cross-modal attention-based modules.

The self-attention-based fusion module in the EAML model is used as a middle fusion block in the ensemble trainable network. The intermediate features from the middle blocks of the modality branches are taken into the attention block. Here, a combined fusion attention map is created by combining the attention maps of the final features with the attention maps of the intermediate features.

The cross-modal attention encoder models the inter-modality (between image regions and text sequences) and intra-modality (within image regions and within text sequences) relationships. Additionally, both GlobalDoc and VLCDoc use a vision transformer encoder (ViT) to process the visual modality. The model by Krithika et al. [35] uses a vision transformer encoder as well. The vision transformer encoder architecture is based on the standard Transformer model which is primarily applied to text. ViT's apply the transformer architecture to image patches and treat them as sequences of tokens [94].

GlobalDoc uses a RoBERTa encoder, which is an optimized encoder based on the BERT encoder. Further, the model uses three pretraining objectives, of which two are of a cross-modal nature, and one only pretrains within the modalities.

Lastly, Voerman et al. [36] study different solutions to the document image classification problem. The first studied solution is a multimodal neural network with an attention model and an adapted loss function. The network is composed by combining the best performing evaluated networks for the respective modalities. This solution performs better than the state-of-the-art approaches for imbalanced cases, however, in other cases it performs worse, e.g. incompleteness of the data or for weak classes. The second solution uses a cascade of systems, classifying documents after different stages based on the confidence of the classification. This model has been created based on the idea that there might be issues such as few-shot learning and incompleteness, therefore allowing the classification to learn further and improve specialization of the model.

Using a fastText encoder The fastText model is a designed by Facebook's AI research (FAIR) lab and is developed to understand and classify text. Just like BERT$_{\text{based}}$ models, fastText models are trained on large text corpora and can be used for many different natural language processing tasks. For text classification, fastText breaks texts into words and sub-words units as well, which are tokenized. Additionally, out-of-vocabulary words are handled by breaking them down into character n-grams, which are sequences of characters used to capture patterns at the character level. These tokens are converted into vector representations. The encodings are based on averaging the character n-grams rather than focusing on the context of words. Concluding, fastText generally performs better at handling out-of-vocabulary words and capturing sub-word information [95].

Zingaro et al. [28] propose a deep learning framework for multimodal side-tuning for multimodal document classification. The model uses a MobileNetV2 architecture for the base model with locked weights. Two side models are constructed: a MobileNetV2 architecture for the image modality and a fastText model for the text modality. Both the base model and side model for the image modality are pretrained on ImageNet, while the weights for the text modality are randomly initialized. The three models are fused by summing the results from the three models, on which the classification is based.

EDNets [30] propose a multi-view deep representation learning approach that combines textual and visual information, which is extracted using an EfficientNet model. The fusion of the text and visual modalities is achieved through a multi-view deep autoencoder (MDAE). The contributing part of the model is the multi-view feature learning stage. In this part of the network, the multimodal features are combined through concatenation, where the goal is to find a shared multi-view representation, on which the final classification is based.

Using a non-ocr text extraction technique Another network using attention mechanisms for the fusion of features is Structextv2 [33]. This is not directly a classification model, but a document image pretraining framework, aiming to overcome the shortcomings of OCR based text extraction. The pretraining is performed using two different pretraining tasks. One focuses on masked image modeling (MIM), where some pixels, patches, or latent representations are masked, which the model has to predict. The other one focuses on masked language modeling, which is similar to the MIM task but instead masks parts of the text. This way, both visual and textual features are combined simultaneously. For the further process of document image classification, a CNN for visual feature extraction and a Transformer for semantic feature extraction are utilized. Feature maps are created from the features, which are fed into a final linear layer.

This layer, combined with a SoftMax activation function, predicts the label of the document image. This pretraining approach in combination with the classification architecture achieve a competitive performance accuracy with state-of-the-art models.

Textual & Structural-based Methods The following classification methods do not use the visual modality to classify documents and do therefore do not belong to the field of document image classification. As they do focus on combining multiple modalities for classification, they are still deemed relevant for this literature research. The layout modality can be captured in different ways, but in most cases either the position of text parts is captured by OCR extraction [20], or different sections within a document are localized [24], such as a title or a paragraph. Most models combining the textual & structural modalities do not only focus on document classification, but are created to perform multiple document processing tasks [20, 23, 24, 77]. The features and the fusion strategy are compared in table 2.6, where we can see that all text is encoded using some kind of BERT$_{\text{based}}$ text encoder (RoBERTa and BART are encoder based on BERT), and attention mechanisms are predominantly used to fuse the features. We go further into the model specifics.

Table 2.5: Classification Accuracies achieved classifying RVL-CDIP and Tobacco-3482 for Textual & Structural-based methods

Model/Author	**RVL-CDIP**
LayoutLM [20]	94,42%
LiLT [23]	95,62%
Kim et al. [77]	95,3%
GVdoc [24]	87,50%
UDOP [21]	96%

Table 2.6: Textual-Structural Model Comparison

Model/Author	**Textual Modality**	**Layout Modality**	**Feature Fusion**
LayoutLM [20]	OCR & BERT	Bounding Box Positions	Self-Attention Mechanism
LiLT [23]	OCR & RoBERTa	Bounding Box Positions	Bi-directional Attention Mechanism
Kim et al. [77]	Swin Transformer & BART	Swin Transformer	Multi-Head Self-Attention
GVDoc [24]	OCR & BERT	Bounding Box Positions	Graph Representation
UDOP [21]	OCR	Word Bounding Box Positions	Unified Vision, Text and Layout Encoder

LayoutLM [20] pretrains a model for document image classification based on both text and layout information, namely position embeddings of text bits in the document and image embeddings of parts of the documents. The model is pretrained on the IIT-CDIP dataset. Lastly, it is fine-tuned on different tasks, of which one being document classification.

The benefit and novelty of LiLT [23] over LayoutLM is that it can be used for understanding structured documents without requiring English text in it. All earlier document classification models that focus on text have been pretrained on merely English language. LiLT however, is language-independent. A bidirectional attention complementation mechanism is utilized to accomplish the cross-modality interaction of the used modalities. The model learns to understand the layout structure of documents during pretraining from monolingual documents and then utilizes this knowledge to fine-tune on multilingual documents.

In contrast with models discussed earlier, Kim et al. [77] try to classify document images based on text without using an OCR-based technique. The proposed model is a Transformer-based model called Donut (Document Understanding Transformer). The input image is split into non-overlapping patches, to which a Swin Transformer is applied. A Swin Transformer is a type of vision transformer that breaks down images into smaller patches and combines these patches into feature maps. Because it focuses on smaller parts of images, it can process images faster. This Swin Transformer consists of a multi-head self-attention module and a two-layer multi-layer perceptron. The combined patches are fed into a decoder, where the encoded embedded input is decoded using the so-called BART model. The model learns to read texts from the document by pretraining it on large corpora. Lastly, it is learned how to understand the text by fine-tuning the model.

Another approach for processing the combination of textual and layout information is combining both in a graph representation. This approach is taken by GVdoc [24] , where in this graph representation edges represent spatial relationships between the different regions of the document. The combined features are fed into a graph neural network, which is pretrained for three different tasks (Masked Language Modeling, Masked Position Modeling and Cell Position Prediction). Furthermore, the model is tested for robustness to out-of-distribution data as well as identifying out-of-domain data. Two different metrics are used in the graphs as well, where the combination of both metrics grants the best performances.

Another proposal for a unified framework using a transformer-based architecture to combine textual and layout features is UDOP (Universal Document Processing model) [21]. This model utilizes textual and spatial correlation and proposes a vision-text-layout transformer that is used in the newly introduced UDOP model. The features are fused in the input stage using a transformer encoder and decoded in the VTL decoder, which consists of a text-layout decoder and a vision decoder. This vision decoder focuses mostly on spatial-textual information in the image, which is why this model is assigned to the textual-structural models. UDOP performs competitively however does not outperform the SOTA models.

Table 2.7: Classification Accuracies achieved classifying RVL-CDIP and Tobacco-3482 for Visual & Structural-based methods

Model/Author	RVL-CDIP	Tobacco-3482
Kaddas and Gatos [78]	92,95%	80,64%

Visual & Structural-based Methods The combination of the visual and structural modalities, is a unique one in document classification. Kaddas & Gatos [78] propose a method that combines visual-based with structural-based features. These layout features are based on text blocks, paragraphs, lines, words, and symbol segmentation results. These features are extracted in the pre-processing stages. The visual features are handled through a ResNet50 backbone model. Further, the layout information capturing the different levels of segmentation are each handled through an individual segment level branch. The deeper, i.e. more detailed the segment level is, the more layers are used to grasp the segment. The different segment levels are combined with the visual document information through an average pooling layer, and classified using a SoftMax function. No explicit pretraining is used for improving the performance of the model.

Textual & Visual & Structural-based Methods Lastly, fourteen classification architectures using the three identified modalities altogether have been proposed. Again, most of these models use OCR techniques to extract text and layout features. Generally, layout/structural features are being captured through coordinates of specific sections within the documents, e.g. text sections, titles, images. The way these coordinates are captured and handled is different for each of the discussed papers. A significant part of these models use a LayoutLM-based [20] or LayoutLM-influenced architecture. Furthermore, graph representations are increasingly being utilized in these proposed architectures.

Table 2.8: Classification Accuracies achieved classifying RVL-CDIP and Tobacco-3482 for Visual & Textual & Structural-based methods

Model/Author	**RVL-CDIP**	**Tobacco-3482**
DocFormer [73]	96,17%	
SelfDoc [25]	93,81%	
MGDoc [96]	93.64%	
Bi-VLDoC [97]	97,12%	
UDoc [74]	93,64%	
Pramanik et al. [54]	93,36%	
LayoutLMv2 [55]	95,64%	
Mahajan et al. [83]	97,30% (using LayoutLMv2)	
LayoutLMv3 [22]	95,93%	
Ali et al. [84]	95,87%	
Hamed et al. [56]		83,24%
Xiong et al. [76]	93,45%	
Mandivarapu et al. [75]		77,5%
Shilpa and Soma [89]	**98,77%**	

In table 2.9, the ways the different modalities are processed are and the feature fusion method are set apart. Almost all of the models use optical character recognition to extract the text from the documents, and consequently use some encoder to tokenize, embed and encode the text. In order to capture the visual modality, a backbone CNN is used. Generally, the structural feature is captured through bounding boxes at either word, text segment, object or page level.

Table 2.9: Visual-Textual-Structural Model Comparison

Model/Author	**Textual Modality**	**Visual Modality**	**Layout Modality**	**Feature Fusion**
Docformer [73]	OCR & Wordpiece Tokenizer	ResNet50	Word Bounding Box Coordinates	Multi-Modal Self-Attention
SelfDoc [25]	OCR & sentence-BERT	Regions-of-Interest Detection with Faster R-CNN	Object Bounding Box Coordinates	Modality-Adaptive Attention
MGDoc [96]	OCR & BERT	ResNet50	Bounding Box Positions at Word-, Region-, and Page-level	Cross-Modal Attention
BiVLDoc [97]	OCR & RoBERTa-Large	Mask R-CNN trained on PubLayNet	Bounding Boxes & Anchor Box Representations	Transformer Layer (Bi-directional Text-Image Self-Attention)
UDoc [74]	OCR & Hierarchical Transformer Encoder	ConvNet	Text & Image Features extracted based on document regions (Regions-of-Interest)	Gated Cross-Attention
Pramanik et al. [54]	OCR & Longformer	ResNet50 & FPN	Bounding Box Positions	LongFormer Encoder
LayoutLMv2 [55]	OCR & Wordpiece Tokenizer	ResNeXt-FPN	Text Section Bounding Box Positions	Spatial-Aware Self-Attention
LayoutLMv3 [22]	OCR & RoBERTa	Vision Transformer	Segment Level Layout Position	Multimodal Transformer
Ali et al. [84]	OCR & RoBERTa	Vision Transformer	Segment Box Positions	Multi-Head Cross Attention
Xiong et al. [76]	OCR & BERT	ResNet50 / VGG19	Text Block Coordinates	Graph Representation
Mandivarapu et al. [75]	OCR & Word2Vec	Pretrained VGG16	Region Bounding Boxes Pretrained on PubLayNet	Graph Representation

DocFormer [73] is of an encoder-only transformer architecture, where the features are combined in a multimodal self-attention layer. The input consists of the visual and textual features individually, but also combined with spatial features (visual-spatial and language-spatial). The model is pretrained using different types of pretraining tasks. The SoftMax function is applied first separately for the modalities, and then combined into one classification.

SelfDoc [25] tries to overcome overly fine granularity by focusing on large segments rather than individual words. A document object detector using Faster R-CNN is used to annotate bounding boxes for semantically meaningful components, and to find significant components, where in this case the model detects text blocks, titles, lists, tables and figures. The output features are fused in the fine-tuning phase through modality-adaptive attention. Sample-dependent attention weights are applied to the two modalities. In this way, the importance and influence on the classification of the different features can be adapted per document. By multiplying these weights with the score of the respective modality and combining these through a linear additive function, a sigmoid activation function makes the final classification.

MGDoc [96] also tries to improve performance by focusing on a different level of granularity. In contrast to SelfDoc [25], it does not overlook fine granularity, but it tries to capture information from the different levels of granularity. They argue that the relation between content at these different granularity levels, e.g. words, regions, and pages, is very important for document understanding tasks. As previous models mostly use only one granularity level for document understanding tasks, MGDoc proposes to combine the different levels of granularity into a unified text-visual encoder.

Where MGDoc focuses on different levels of granularities, Bi-VLDoC [97] focuses on bidirectional relations between the modalities through a so-called bidirectional hybrid-transformer. This technique allows the model to pay attention to visual and textual parts of a document in both forward- and backward directions. Specifically, the model uses bi-directional vision-language supervision. This combination should contribute to a cross-model feature extraction encoder. The architecture of the model is constructed by first inputting the three modalities separately into the bidirectional vision-language hybrid-attention module. The representation generated through this module is then utilized for pretraining tasks and downstream document intelligence tasks. Image pretraining on the Bi-VLDoc is done utilizing the extensive IIT-CDIP dataset. For the pretraining of the textual and layout features are done, respectively, using RoBERTa-Large and PubLayNet[14].

The work proposing UDoc (Unified pretraining Framework) [74] puts more attention towards the pretraining of the classification model. The model tries to learn from cross-modal contextualized embeddings. Finally, the classification is made by computing the element-wise product between visual and textual representations, which is averaged over all sentences/regions.

The main focus of Pramanik et al. [54] is to classify long multimodal documents. They com-

[14] PublayNet Dataset

bine information on the modality level with the information available per page level for PDFs with multiple pages. The inputs are encoded using the Longformer network architecture[98], a specific encoder for long documents. The longformer encoder generates sequence representations for the documents. The features used are based on page numbers, sequences of tokens, bounding boxes, and page images.

LayoutLM-based architectures LayoutLMv2 [55] utilizes the earlier constructed LayoutLM [20] model (see section 2.3.3), and builds upon it by also incorporating the visual aspect of images rather than just textual and layout information. The different features are combined in the pretraining stage through a multimodal transformer model. A spatial-aware self-attention mechanism is integrated into the transformer architecture. In the pretraining stage, different pretraining tasks are performed. Compared to the first LayoutLM model, as well as other SOTA models at that time, LayoutLMv2 has an improved performance.

Mahajan et al. [83] try to improve the classification performance of LayoutLMv2 [55] by applying a novel combination of image preprocessing techniques, since they recognize that OCR does not always perfectly extract all of the included text. First, a grayscale conversion is applied using cvtColor, secondly smoothing and blurring is performed in order to cut down on the amount of detail and noise in an image, thirdly the images are segmented using adaptive thresholding, where all pixels with intensities higher than the threshold are set to the same foreground value, and a null value is assigned to the remaining pixels. Lastly, a bitwise operation is done to divide the image's foreground and background in images. These operations cause the texts in an image to be more clear and, therefore, easier for OCR techniques to extract. Performing these preprocessing techiques improves the LayoutLMv2 results from an accuracy of 93.07 to 97.3%.

The LayoutLMv3 [22] architecture is also an improved version to the earlier developed LayoutLMv2 [55]. One of the adaptations in LayoutLMv3 is the adoption of segment-level layout position, where layoutLMv2 captured the focus on a finer granularity, namely the word-level layout positions. Similarly to its predecessor model, LayoutLMv3 uses base and large model sizes. Furthermore, LayoutLMv3 uses a simplified architecture compared to LayoutLMv2 by using patch embeddings, in a vision transformer-like model, rather than a CNN backbone model as used by LayoutLMv2. The performance in terms of classification accuracy of layoutLMv3 improves compared to LayoutLMv2.

Both Ali et al. [84] and Hamed et al. [56] take inspiration from the LayoutLMv3 model [22]. The architecture proposed by Ali et al. [84] propose a transformer-based model that combines features from the three modalities for different document analysis tasks, one being document classification. The model consists of a text-embedding layer, patch embeddings (visual) layer, position (layout) encoding, multi-head attention based encoder, and decoder layers, through which all of the information is encoded and decoded. The classification is performed through a linear transformation layer and then classified through a SoftMax activation function.

Hamed et al. [56] take a different a different approach by focusing on finding the balance between efficiency and performance. An early exit strategy is proposed for which the goal is to achieve a pareto-optimal balance between the performance and efficiency. The LayoutLMv3 model is utilized, and intermediate classifiers are placed at different parts of the network to explore earlier and more efficient classification.

Graph Representations Xiong et al. [76], Mandivarapu et al. [75] and Shilpa and Soma [89] like GVDoc (see section 2.3.3) present a graph convolutional network to learn the three modality features. Mandivarapu et al. [75] specifically focus on proposing a more efficient document image classification model, while combining the three modalities. Utilizing all information leads to a large amount of data needing to be processed and therefore to a decrease in the speed of the model. Graph representations should address this problem by using a more efficient representation of the data. The nodes represent the features of image and text, and the edges the structural information such as location of the objects. Both the model by Xiong et al. [76] and the model by Mandivarapu et al. [75] do not necessarily improve the SOTA models in terms of performance as measured by the accuracy metric; however, the models do improve significantly in terms of training speed.

Shilpa and Soma [89] have designed their graph-attention-driven model (GAD-DTL) with a dual-tune learning system. This dual-tune learning system uses two learning techniques to enhance the model's performance, the first one capturing the relationships between different elements within the document. The second learns the features on itself, without taking the relationships into account. Semantic region embeddings are found within document images, where textual and spatial information is combined, along with captured visual information. The features are combined through an adaptive fusion layer, in which different weights are assigned to the features based on their importance for a specific document. This graph-attention-driven model achieves the highest performance accuracy in this category of textual, visual & structural-based methods.

SUMMARY

After learning about all these specific applications of document image classification, we can evaluate which methods have been used and at what frequency. This section goes into the different parts of the methods, specifically discussing the architectures used for the different modalities, the methods for fusing the features, and how the classification is done. Then, the different pretraining strategies are discussed, followed by fine-tuning practices. Lastly, different techniques are applied to improve training efficiency, and therefore are discussed.

Architectures for handling visual features In document image classification methods, visual features are typically extracted using deep learning techniques. For this purpose, both CNNs and transformer-based models are employed. The four most commonly used architectures are ResNet, VGG16, Vision Transformer (ViT-B/16), and EfficientNet. In addition, models based on MobileNet and NasNet have been used, although less frequently.

Architectures for handling textual features Most of the models discussed use an optical character recognition (OCR) tool to retrieve the text from the documents, the most used tool being Tesseract OCR [99]. The extracted text needs to be understood in order to compare it; therefore, encoder tools such as $\text{BERT}_{\text{based}}$ encoders and fastText [100] are used to encode the text in documents. Numerous different versions of $\text{BERT}_{\text{based}}$ encoders are used; the standard BERT [93], RoBERTa [101], BART [102], and lastly a sentence-based BERT encoder. Furthermore, a more efficient and lighter language encoder is fastText [100]. Overall, the function of these language encoders is comparable. All transform the retrieved textual data into vector representations that capture semantic meaning, enabling comparison and classification of documents.

Architectures for extracting layout features The methods used to extract layout features can be divided into the following four categories, each capturing layout details in a different way.

- Capturing 2D position of objects
- Graph representation
- Subdividing the document in patches
- Using pretrained layout extracting methods (R-CNN & PublayNet)

Feature fusion methods Although the modalities are generally extracted individually, a fusion of the features is required to benefit from all the modalities at once. The characteristics by which the feature fusion methods can be characterized are the following. The characterization of all feature fusion methods applied in the discussed models can be seen in Table C.2.

- Moment of fusion (early/hybrid/late)
- Complexity of fusion
- Involvement of attention mechanism
- Staticity/Adaptivity of fusion method

Pretraining Most high-performing models utilize some form of pretraining. Language encoder models for the textual modality are typically pretrained. However, models for the visual modality are not always pretrained. Many visual models are based on ImageNet weights, which is a pretrained model in a sense. Although ImageNet is a model trained to recognize a collection of objects that may not perfectly align with the document classes, it still provides a valuable starting point in many cases. More pretraining is conducted for some models using the RVL-CDIP dataset, which is more closely related to document classification. Various pretraining tasks are utilized, focusing on different granularities and ways to understand text and images, as well as finding the relation between modalities.

Improving the efficiency For the development of models, not only performance has been the goal. The objective of making deep learning models more efficient has been a widely emerging development in the area, for document image classification practices as well. In the materials discussed, several methods of making models more efficient have been designed. First, EfficientNet is used by Borst [70], EDNets [30], and EmmDocClassifier [31]. Ferrando et al. [29] aim to decrease the computational time as one of the main goals, and therefore make use of EfficientNet and utilize parallel systems, i.e. multiple GPUs to speed up the pretraining process. By utilizing an early exit strategy to achieve a balance between performance and efficiency, Hamed et al. [56] aim to make a more efficient model. Lastly, graph representations are also used to make the classification model more efficient [24, 75, 76, 89].

2.4. GAPS IN THE LITERATURE & FUTURE RESEARCH

2.4.1. APPLICATION OF DOCUMENT IMAGE CLASSIFICATION TO THE AEC INDUSTRY

Through the evaluation of articles published on the subject of document classification in the AEC industry, we found that the applications to date focus mainly on text mining / classification practices. The previous researchers mostly used text to understand the differences between different documents, applying the results to various cases in the industry [61, 62, 65–68]. As for some of the cases the use of text classification is a fairly logical and useful decision, not for all applications in the AEC industry this is assumed to be most useful. Especially in the case where a document set consists of textual as well as rather visual documents such as maps or drawings, or specific documents that vary in layout, only evaluating the documents based on the textual information does not always make the most accurate classification. Furthermore, many document classification models based on textual features are based on OCR techniques [26, 29, 30, 34, 35, 61, 65], which is not always the most reliable source of information, as often not all words can be extracted completely correctly [77].

The combination of these two shortcomings, as well as the developments of research on document image classification and multimodal document classification in the general research fields, requests more research on document image classification models and multimodal document classification models for AEC projects. The first step in research would be to evaluate whether better performing classification models could be constructed and what these architectures would look like.

2.4.2. EXPLAINABILITY AND INTERPRETABILITY

Section *XAI in document image classification* (2.3.3) describes practices aimed at enhancing the transparency, explainability, and interpretability of document image classification models. With the rapid developments in explainable AI, numerous techniques have been designed and experimented with. The application of existing XAI techniques, as well as the design of new methods specifically tailored to document image classification, could significantly benefit the development of document image classification models. Several XAI techniques designed specifically for document image classification have already been developed [80, 81, 85–87]. However, further research, particularly into XAI techniques for multimodal document classification would be beneficial. In this way, more trust would be instantiated between users and stakeholders in these models, as decision-making becomes more understandable.

2.4.3. EFFICIENCY AND SCALABILITY

The application of deep learning models requires a lot of computational power, which increases with the complexity of the model and the volume of training data [29]. This results in extended training times and an increased amount of computational power necessary. With the addition of different modalities, the data processed by the model expands, causing the model to need even more time and power to train. Research has explored various methods to improve the efficiency of document classification models, as described in section *Improving the efficiency* (2.3.3). Despite these practices, training (multimodal) document image classification models remains time-consuming and computationally intensive [29, 31]. Therefore, further research is essential to improve efficiency in this area.

2.4.4. HANDLING IMBALANCED DATA

The datasets used in this research are rather balanced, as they have been curated and specifically designed for classification purposes. In real life projects, datasets are often not as balanced, causing challenges in model performance and accuracy. Imbalanced datasets might lead to biased predictions, as the model will likely predict the majority classes for more documents in than minority classes, as it has learned more about the majority classes [24, 36]. This problem asks for solutions using additional techniques such as resampling, data augmentation or specifically designed algorithms to address this problem. More research on imbalanced datasets for document image classification could help identify most evident problems and techniques to overcome the results of imbalanced datasets.

2.4.5. PRETRAINING AND TRANSFER LEARNING ON DOMAIN-RELATED DATASETS

In general, we have seen an improved performance of classification models after being pretrained on another dataset [26, 29, 31, 35, 81, 82, 86, 88]. The effectiveness of pretraining depends significantly on the relatedness of the documents of the pretraining with the documents being classified in the end. Both ImageNet [26, 29, 35, 81, 88] and the RVL-CDIP [31, 82, 86] dataset have been used during the pretraining process. As ImageNet consists of general images of objects, and the RVL-CDIP dataset of rather general categories of documents, these datasets might not always be relevant to AEC documents. Future research should design a dataset more relevant to AEC and construction documents, and possibly a pretrained model could be constructed that could be widely used in the industry.

3

METHODOLOGY

After building a foundation on the subject of multi-modal document image classification by means of an SLR, the retrieved knowledge is used to find a solution to the overall research goal. An ML design research is carried out to find the best contribution to the lack of document classification solutions in AEC asset management.

Two main methodologies were followed in this research; the first being CRISP-ML(Q) [4], a methodology specifically designed for ML research projects. CRISP-ML(Q) builds upon the widely used Cross-Industry Standard Process for Data Mining (CRISP-DM) [103] methodology, a methodology that describes the general data science lifecycle. CRISP-ML(Q) adapts the CRISP-DM methodology to address the complexity and iterative nature of ML development. The CRISP-ML(Q) methodology was introduced in 2021 to better guide ML development processes, and give more support for quality assurance, better meet business expectations, and post-deployment monitoring [4], something where CRISP-DM is lacking. After its introduction, CRISP-ML(Q) has been widely adopted in both academic and industrial contexts [104, 105]. Secondly, the Automated Machine Learning (AutoML) methodology is utilized [104, 106]. This methodology aims to make ML models more accessible and understandable by non-experts, as it automates time-consuming and complex ML tasks. In this section, we describe the way the constructed classification models are validated and evaluated.

3.1. CROSS-INDUSTRY STANDARD PROCESSING FOR MACHINE LEARNING (QUALITY)

The CRISP-ML(Q) methodology consists of six broad phases, which consequently consist of substeps, as demonstrated in Figure 3.1. The broad stages conclude to *business & data understanding, data preparation, modeling, evaluation, deployment* and lastly *monitoring & maintenance*. This study focuses primarily on the first four phases. While deployment and monitoring are discussed conceptually, they are not fully implemented. Given that these steps are a significant part of the CRISP-ML(Q) methodology's contribution compared to CRISP-DM, one might argue that using CRISP-ML(Q) over CRISP-DM is excessive. However, the detailed steps outlined for each of the methodological phases provide a more comprehensive contextual framework for our research and future application of the resulting models, as it addresses ML specific concerns and configurations. For the same reason, we elaborate on each of these steps, even though not all steps are followed.

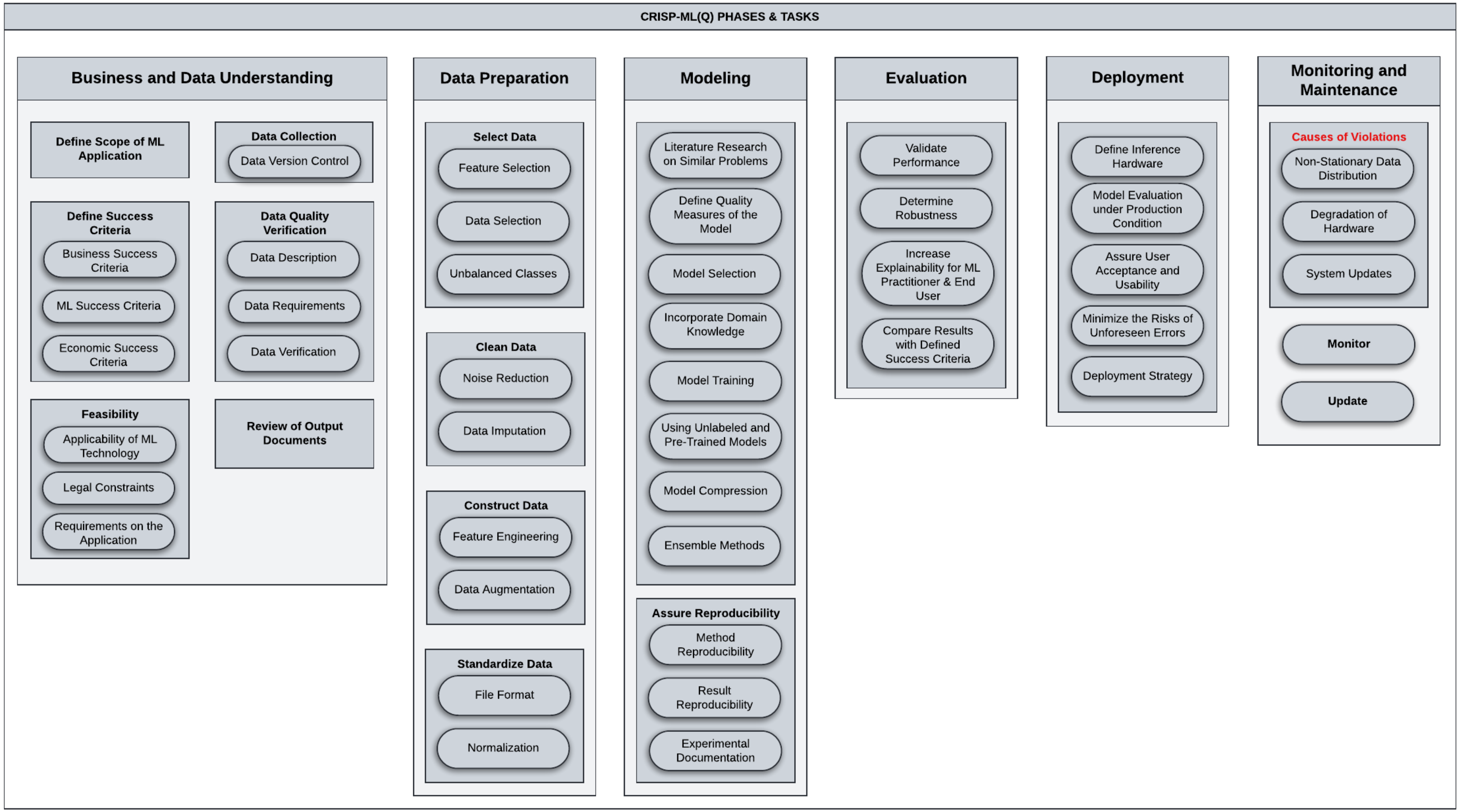

CRISP-ML(Q) PHASES & TASKS
Business and Data Understanding
Define Scope of ML Application
Data Collection
Data Version Control
Define Success Criteria
Business Success Criteria
ML Success Criteria
Economic Success Criteria
Data Quality Verification
Data Description
Data Requirements
Data Verification
Feasibility
Applicability of ML Technology
Legal Constraints
Requirements on the Application
Review of Output Documents
Data Preparation
Select Data
Feature Selection
Data Selection
Unbalanced Classes
Clean Data
Noise Reduction
Data Imputation
Construct Data
Feature Engineering
Data Augmentation
Standardize Data
File Format
Normalization
Modeling
Literature Research on Similar Problems
Define Quality Measures of the Model
Model Selection
Incorporate Domain Knowledge
Model Training
Using Unlabeled and Pre-Trained Models
Model Compression
Ensemble Methods
Assure Reproducibility
Method Reproducibility
Result Reproducibility
Experimental Documentation
Evaluation
Validate Performance
Determine Robustness
Increase Explainability for ML Practitioner & End User
Compare Results with Defined Success Criteria
Deployment
Define Inference Hardware
Model Evaluation under Production Condition
Assure User Acceptance and Usability
Minimize the Risks of Unforeseen Errors
Deployment Strategy
Monitoring and Maintenance
Causes of Violations
Non-Stationary Data Distribution
Degradation of Hardware
System Updates
Monitor
Update

Figure 3.1: Phases & Tasks as Described in the CRISP-ML(Q) Methodology [4]

In addition to updating the CRISP-DM methodology, CRISP-ML(Q) [4] incorporates quality assurance measures. Before each step within the methodology, the requirements and constraints are defined. Furthermore, the methodology specifies and instantiates the included tasks, identifying all possible risks that could affect the success of the ML application. During each phase, to ensure a quality outcome, one should design a mitigation strategy for the identified risks.

3.1.1. BUSINESS & DATA UNDERSTANDING

In the first phase, our primary objective is to discover and define the business problem and the resulting objectives. We then translate these business objectives into ML objectives. During this step, we design the requirements for the ML solution, establish its success criteria and evaluate its feasibility. Lastly, we collect and evaluate the required data in terms of its quality. In the remainder of this section we describe the steps in of the *Business & Data Understanding* phase in more detail.

One can capture the success criteria at three levels, namely business success criteria, ML success criteria, and lastly economic success criteria. These criteria help set goals and define when to deliver the ML product. The feasibility of the ML project should be evaluated in terms of different factors; applicability of ML technology, legal constraints, and requirements of the application. Further, the required data is collected and analyzed. An important consideration during the data collection is that the data collection should be an iterative process, where data can change or be updated over time. Version control of the data is essential to ensure reproducibility and data quality. Ensuring data quality is essential not only during later iterations of data collection but throughout the entire process. The CRISP-ML(Q) methodology guarantees quality by first describing and exploring the data, and then defining the requirements that specify the expected conditions of the data, where a domain expert should be involved. This way possible biases can be mitigated beforehand. Data that does not meet these requirements should either be excluded or amended to comply.

The final step in the methodology to ensure data quality is data verification, which ensures that all data meets the defined requirements. Additionally, this step aims to mitigate the risk of insufficient representation of extreme cases by using data exploration techniques to evaluate the data distribution.

3.1.2. DATA PREPARATION

Once all data is collected and its quality is ensured, the data should be prepared for the modeling phase. Just as data collection should not be a static phase, data preparation is a rather iterative phase as well. Anywhere in the process, data can be further adapted to better fit the model and ML objectives if necessary. We discuss four main steps in the data preparation phase, covering the selection, cleaning, construction, and standardizing of data.

First, data selection is subdivided into three tasks, the first being feature selection. It is best practice to select only the features that are necessary as the more features that are selected, the more data samples are necessary. By selecting as few features as possible, we try to prevent the curse of dimensionality. Features should be selected not only based on the model and the dataset, but also with input from domain experts to mitigate the risk of potential biases.

In addition to feature selection, data can be selected and filtered based on objective quality criteria. If a data sample does not satisfy the designed criteria, it should be excluded from the data set.

Then lastly, the data should be evaluated in terms of it's balancedness. Different sampling strategies could be used to improve balance in the case of unbalanced classes. Oversampling can be used to increase the importance of the minority classes, however, it increases the risk of overfitting on the minority class. Contrastingly, the majority class can be undersampled by removing data points from it. This should be done rather carefully as the characteristics of the data need to be kept and biases should not be introduced through this process. By comparing the results of both strategies, the risk of bringing biases into the model is reduced.

The next step within the data preparation phase is cleaning the data, where both the data is filtered in order to reduce noise in the data and secondly, data is imputed in order to work with a complete dataset, which could be done using various techniques and strategies. Again, the model performance should be compared between the different imputation strategies in order to decide upon the best-fitting one.

Although data construction and standardization can be important steps in data preparation, and specified as explicit steps in the CRISP-ML(Q) methodology as well, they were not performed in this study.

Constructing data is part of the data preparation process as well. New features could be derived from existing ones in order to engineer features that are more relevant to the model objectives. Further, new data can be constructed through data augmentation, where transformations are done on the data, such as applying rotations, adding noise or by augmenting the data on meta-level.

Lastly, data standardization could be applied. Some ML tools require specific data files or input types, to which data has to be converted before being able to perform ML practices. Additionally, applying normalization helps overcoming biases and achieves convergence at a faster rate.

3.1.3. MODELING

The decision on which modeling technique to use is determined by the combination of business objectives and ML objectives, the data, and the project's boundary conditions. The requirements and constraints designed should help filter which models to build and compare. Factors that help decide what model to utilize could be, among others, performance, robustness, scalability, explainability, and model complexity. In addition to requirements and constraints for the model, literature research on similar problems could be beneficial for model selection as well, as the literature could provide information about previous applications, and possible benefits and downfalls of the specific model in the specific setting.

The ML models can be evaluated on various measures. In addition to the performance of the model, many other measures could be relevant. Examples of relevant measures, similarly to factors guide model decision, could be robustness, explainability, scalability, resource demands, and model complexity. The way these measures are weighted and assessed depends on the specific application. No single model performs perfectly for all problem classes; a specialized model for a specific task will always perform better. Adapting a model towards a specific task, however, always requires domain knowledge as it brings the risk of incorporating false assumptions and biases.

Training the model depends on the learning problem, where various settings can be optimized. The objective of the application defines how to evaluate the model performance, the optimizer defines how to adapt the parameters of the model to improve, and regularization can be utilized to reduce the risk of overfitting. Cross-validation can be used to optimize the hyperparameters and to test the generalizability of the model. As labeling data is very time-consuming, performing unsupervised or semi-supervised pretraining could speed up the process. Transfer learning could be applied to reuse the weights established through another ML practice.

Multiple models can be trained individually, after which the results can be combined. In this way, the various models can account for each other's errors, increasing the confidence in the prediction as well. In general, it is complex to exactly reproduce ML models due to non-convex and stochastic training procedures, as well as randomized data splits. Therefore, reproducibility has been split down to two levels; method reproducibility and result reproducibility. In addition, the modifications and details of the model should be documented to improve overall reproducibility.

3.1.4. Evaluation, Deployment and Monitoring & Maintenance

A model is evaluated based on the goals, requirements, and criteria designed. In general, a model is evaluated by validating its performance. Additionally, the robustness of the model is important to ensure that the model performs similarly for other data samples and test sets, as well. By increasing the explainability of the model, finding errors is supposedly easier and might introduce strategies to further improve the model. Additionally, increased explainability might help increase trust and user acceptance.

After evaluation and finalization of the ML model, the model can be deployed. The way this is done depends on the designated field of application. Five main concerns should be evaluated in order to initiate the best deployment; what hardware would be most suitable and model evaluation under production, assuring user acceptance and usability, minimizing the risks of unforeseen errors, and lastly what specific strategy of deployment to employ.

Once the ML model is operational, improvements might be necessary and performance violations could occur. The main violations include non-stationary data distribution, where shifts

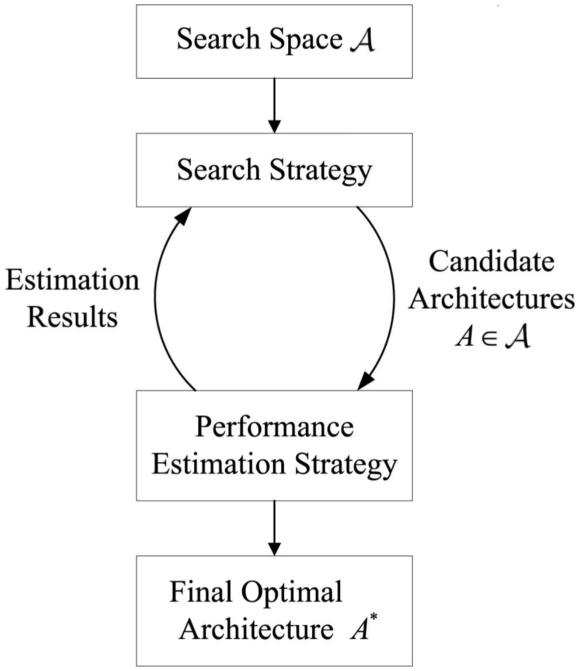

Search Space $\mathcal{A}$
Search Strategy
Estimation Results
Candidate Architectures $A \in \mathcal{A}$
Performance Estimation Strategy
Final Optimal Architecture A^*

Figure 3.2: Neural Architecture Search (adapted from [5])

in features or labels may occur. Changes in data structure or content could violate the model's performance over time. Additionally, the hardware on which the model is deployed may deteriorate over time. Therefore, hardware performance should be monitored and updated if necessary. Lastly, system updates might change circumstances on which the data is dependent, as well as the model itself. After the deployment of the model, monitoring its performance to recognize possible violations is crucial. If such events occur, the model must be updated to comply with the new conditions.

3.2. AUTOMATED MACHINE LEARNING

AutoML has been a highly researched topic with significant developments from the introduction of AutoML [107] in 2014 to now. AutoML involves the process of automating the process of applying ML to real-world problems [106]. Tasks such as data preprocessing, feature selection, model selection, hyper-parameter tuning, and model evaluation are the focus of this discipline. These tasks are often rather time-consuming and complex during the development of ML models [108, 109]. AutoML aims to speed up and simplify the development of ML applications, making them more accessible, even to non-experts.

3.2.1. NEURAL ARCHITECTURE SEARCH

NAS is the process of automating architecture engineering, one of the steps often used to automate ML. It generally designs architectures that perform better than human-designed architectures [58]. NAS has three main focuses; *search space*, *search strategy* and *performance estimation strategy*, as illustrated in Figure 3.2.

The search space includes identifying which types of architectures are suitable as the final design, where incorporating prior knowledge about architectures can be beneficial. This search space can be very large or even unbounded. The search strategy describes how to explore the search space in order to find the most fitting and best performing architecture. Lastly, the performance estimation strategy decides which method is used to retrieve a performance measure. The most desired measure in this sense would be the performance of the model on unseen data. To retrieve this measure, we train each model and validate it using unseen data. Through this process, the final optimal architecture (see Figure 3.2) is found in terms of the defined performance measure.

3.3. MACHINE LEARNING MODELS

This research uses different applications of ML models, which are described in further detail in this section. The foremost type are classification models. In addition, OCR is used to extract text from document images. This section further describes the ML models and techniques used.

3.3.1. OPTICAL CHARACTER RECOGNITION

OCR techniques allow us to recognize characters of handwritten and printed text from an image without using the human ability to read [110]. OCR does not always perfectly extract text, as its performance and accuracy are highly dependent on the quality of the input document or image. Distortion and/or noise in images often degrade the performance and accuracy of OCR techniques.

3.3.2. CLASSIFICATION MODELS

This research distinguishes between different data modalities used for classification. ML models utilizing computer vision, natural language processing, and document layout analysis are evaluated and combined. All models share the common objective of classification. Classification is a supervised ML task, where a model is trained to categorize data into predetermined categories based on previously seen related data [111]. This section discusses the selected classification models for each data modality.

VISION BASED CLASSIFICATION MODELS

This research evaluates five different CNN architectures for image classification. We select the selected architectures based on the literature review (see Chapter 2). These architectures vary in size, structure, and performance, which we discuss in chronological order in this section.

VGG16 VGG16 is a CNN architecture developed by the Visual Geometry Group at the University of Oxford [53]. The architecture was introduced in 2014 and is known for its simplicity and depth, consisting of 16 layers; specifically 13 convolutional layers, and 3 fully connected layers. The model has achieved good performance results in document classification [35, 36, 75, 76] as well as image classification [112–119].

ResNet50 ResNet50 is a variant of the Residual Network (ResNet) architecture, but consists of 50 layers [50]. The model is a CNN architecture developed by Microsoft research in 2015. With ResNet50 the concept of *residual connections* was introduced, where the model learns the residual functions, which are essentially the difference between the output of a layer and its input, and which maps the input to the desired output. This way, the problem of vanishing gradients, which hinders the training of deep networks by diminishing the gradients exponentially when they propagate backwards, has been tried to overcome in ResNet through the use of shortcut connections between layers [50]. ResNet50 has been trained on large datasets, achieving state-of-the-art results in document classification [54, 73, 76, 96], as well as image classification in general [112–119].

Inception-ResNet-V2 Inception-ResNet-v2 is a hybrid CNN architecture that utilizes the strength of the inception models and combines it with residual connections as designed for the ResNet architectures [120]. The architecture was introduced by Google in 2016 and significantly improves training speed and stability. The models is designed to be both deep and wide, in order to be able to capture complex patterns in data. Inception-ResNet-V2 has demonstrated state-of-the-art performance on various document image classification applications [27, 32]. The architecture is mostly effective for tasks that require high computational efficiency and accuracy [121–123].

MobileNetV2 MobileNetV2 is a CNN architecture designed for mobile and resource-constrained environments [124]. The architecture was introduced in 2018 by Google and builds upon the earlier introduced MobileNet (V1) by incorporating an *inverted residual structure* and *linear bottlenecks*. Compared to this earlier version, MobileNetV2 significantly reduces the number of operations and memory required, while maintaining a high performance. Various researches show a high performance in image classification [114, 125], as well as document image classication specifically [28, 57].

EfficientNet-based Models EfficientNets are CNNs developed through balancing the depth, width, and resolution of the network, instead of by developing a CNN in a fixed resource budget, after which it is scaled for better performance [126]. EfficientNet was found through NAS in 2019 and generally achieved better performances than state-of-the-art networks at the time. EfficientNets are generally smaller in size than other state-of-the-art models. The model architecture ranges from version B0 to B7, where B0 is the base model. This first base model is scaled to the next version each time by applying the *compound scaling method*, which increasingly improves the models' ability to handle more complex data.

Text Based Classification Models

This research evaluates various models for text classification, including four BERT-based models and different classification models using a TF-IDF-based input. These models were selected based on the literature review (see Chapter 2). The capabilities of each model are elaborated on further in this section.

BERT-based Models BERT is a transformer-based model that has achieved state-of-the-art performance in many natural language processing tasks [93]. BERT uses a deep neural network architecture and is pretrained on a large text dataset. It performs well in understanding the context of words by considering their surroundings, making it very effective for (sequence) classification tasks.

Robustly Optimimized BERT approach (RoBERTa)[101] builds upon the BERT architecture by improving the pretraining process. The model was trained for longer duration, with larger batch sizes and more data. These improvements enable RoBERTa to perform exceptionally well in various NLP tasks such as sequence classification.

To address the limitations of BERT and RoBERTa in handling non-English text data [127, 128], two Dutch BERT-based tokenizers/models were developed: BERTje [127] and RobBERT [128]. These models are based on the BERT and RoBERTa architectures, respectively. Both models are trained on large Dutch datasets and achieve state-of-the-art performance in various NLP tasks in Dutch. Although comparisons between the two models are limited, De Bruyne et al. [129] report a better performance for RobBERT in classifying emotions, and Rietberg et al. [130] report a better classification performance for BERTje in classifying diagnosis goals in medical reports.

Term Frequency-Inverse Document Frequency based Models TF-IDF is a statistical measure used to assess the importance of a word in a document relative to a collection of documents [131]. The measure is rather simple, but powerful for text classification and information retrieval. The measure is calculated for each word by multiplying its term frequency by its inverse document frequency. This approach highlights words that are important in a document but not common across the corpus. Using this feature, classification can be performed with various models, such as logistic regression, support vector machines, and random forests [132].

LAYOUT BASED CLASSIFICATION MODELS

For the layout modality models, we utilize one general type of models; transformer-based models, where spatial information from documents is extracted using visual and layout information.

Transformer-based Models LayoutLM is a transformer-based model designed for document understanding tasks, utilizing both textual and layout information [20]. It aims to improve natural language processing tasks such as form understanding and receipt recognition. It does so by utilizing the spatial dimensionality of text elements to understand the document structure. Specifically, LayoutLM uses the bounding-box coordinates of words or word groups to achieve this. Building on this foundation, LayoutLMv2 [55] incorporates additional visual features and better captures the relationships between text and layout through an improved pretraining objective. This improvement allows to model to better understand and process documents, especially documents that are visually rich. LayoutLMv2 integrates text, layout and visual information into one transformer-backbone and improves pretraining, making the model more efficient. Further improving the capabilities of the model, LayoutLMv3 [22] includes visual features next to the textual and layout information. Again, the pretraining tasks are improved, making it the most advanced LayoutLM model.

In addition to these models, the Language-Independent Layout Transformer (LiLT) [23] addresses the limitation of language dependency in layout models. LiLT can be pretrained on documents in a single language and then fine-tuned on other languages, making it very usable for multilingual document understanding.

Lastly, Universal Document Processing (UDOP) [21] is a model that unifies text, image, and layout modalities for various tasks, including document classification. UDOP utilizes a Vision-Text-Layout Transformer to integrate these different types of information to optimally classify documents.

3.4. Model Validation & Evaluation

3.4.1. Validation Methods

To prevent overfitting, instead of using a simple train-test split for validation, each model is validated using cross-validation [133]. Cross-validation is a widely used technique in ML and statistics for model validation and selection. It is commonly used in ML research practices.

Using k-fold cross-validation, the data is split into multiple k folds. Each fold is used as a test set once, while the remaining folds are used for training [134]. This mitigates the risk of selecting a non-representative test set, which would heavily influence the test performance. We average the performance metrics retrieved through these different training processes to provide a more robust validation result, less dependent on any single train-test split. In this way, we can compare model architectures, pre-processing steps, and hyperparameter changes without relying too heavily on the selected data. Finally, a separate test set, which was not used during the cross-validation process, is held out for the final evaluation of the models to evaluate the specific class accuracies achieved.

3.4.2. Predictive Performance Metrics

The performance measures used primarily to evaluate the performance of the classification models are classification accuracy, recall, precision, and F1 score. These metrics are widely used in the literature to evaluate classification models [135].

For each of the classes, we can define four types of predictions in terms of correctness; *True Positive*, *True Negative*, *False Positive*, *False Negative*. True positive entails the predictions that correctly predict that the outcome is part of the specific class, say *Class A*. True negative consequently entails the predictions that correctly predict that the outcome is not part of *Class A*.

Then, false positive entails the outcomes that are incorrectly predictions as part of *Class A*, and the outcomes in false negative are actually in *Class A*, but incorrectly predicted to not be part of it [134–136]. Equations 3.2, 3.3 and 3.4 respectively, define the metrics of recall, precision, and the F1-score.

$$\text{Accuracy} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives} + \text{True Negatives} + \text{False Negatives}} \tag{3.1}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.2}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.3}$$

$$\text{F1 Score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3.4}$$

In other words, accuracy measures how often an ML model correctly predicts outcomes (Equation 3.1). Recall measures whether a model can find all instances within a specific class (3.3). Precision measures how often the model is correct in predicting the target class (3.2, and lastly, the f1 score finds a balance between the recall and precision measures (3.4).

As the model classifies 7 different classes, it is a multi-class classification model. Instead of the metrics designed for binary ML cases, the metrics are retrieved per class and can be averaged to measure the complete model performance [136]. These averages can be retrieved on a micro-, macro-, or weighted level. The micro average is most suitable for a balanced dataset as it computes a global average where each instance is treated as equally important. In the case of a balanced dataset, all classes weigh equally. The macro average does compute the metric for each class independently and then takes the unweighted mean of the metrics. This means that each of the classes contributes equally to the final metric, regardless of how many instances a certain class contains. Lastly, weighted average is similar to macro average in the sense that it computes the metric per class, but weighs each class's contribution by the number of instances per class [135]. In this study, we use weighted average to compute the metrics for the tests on *dataset 1*, and the macro average on *dataset 2*. In addition to the averaged metrics, we consider the class-based metrics to be important as well, as we strive for a balanced performance between the document classes.

4

Experimental Set-Up

As described in the previous chapter, this research adheres to the CRISP-ML(Q) methodology (see Section 3.1). Initially, we establish a comprehensive understanding of the business case and the provided data to ensure the objectives of the optimal classification model are well-defined. Figure 4.1 illustrates the more specific research design. It shows that that during the second, third, and fourth phases, we incorporate a particularly iterative approach, integrating a neural search-like strategy (see Section 3.2.1).

As this research focuses on integrating various features and model architectures into one optimal document classification model, the design and construction of the machine learning models are performed repeatedly and iteratively, focusing on different modalities and model settings. We refer to this sequence of phases as the Iterative Model Development Process (see Figure 4.1). For each specific research objective, we go through this iterative process, where, based on the model's performance related to the set objectives, further iterations might be performed for that particular modality or setting. Figure 4.2 shows the specific steps taken within the iterative model development process for each objective. Then, after finding multiple architecture options, the NAS cycle (Section 3.2.1) is followed to find the best performing architecture for the specific modality.

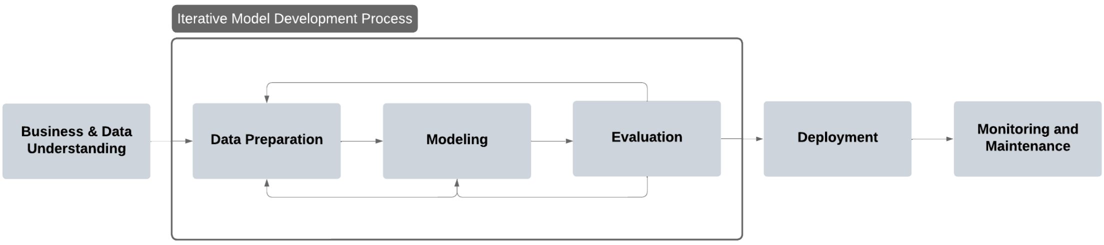

Iterative Model Development Process
Business & Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Monitoring and Maintenance

Figure 4.1: Adoption of the CRISP-ML(Q) Methodology

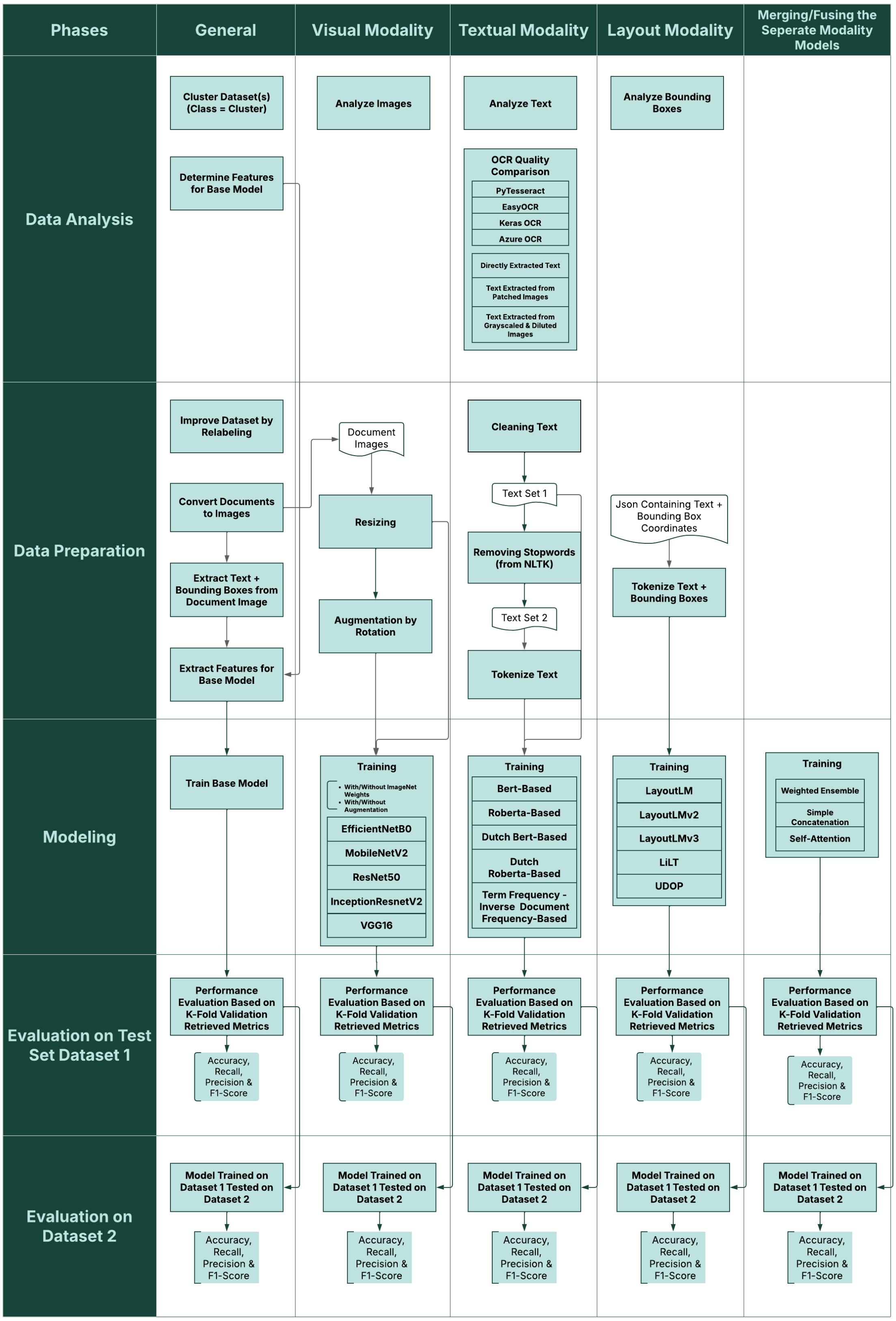

Phases
General
Visual Modality
Textual Modality
Layout Modality
Merging/Fusing the Seperate Modality Models
Data Analysis
Cluster Dataset(s) (Class = Cluster)
Determine Features for Base Model
Analyze Images
Analyze Text
OCR Quality Comparison
PyTesseract
EasyOCR
Keras OCR
Azure OCR
Directly Extracted Text
Text Extracted from Patched Images
Text Extracted from Grayscaled & Diluted Images
Analyze Bounding Boxes
Data Preparation
Improve Dataset by Relabeling
Convert Documents to Images
Extract Text + Bounding Boxes from Document Image
Extract Features for Base Model
Document Images
Resizing
Augmentation by Rotation
Cleaning Text
Text Set 1
Removing Stopwords (from NLTK)
Text Set 2
Tokenize Text
Json Containing Text + Bounding Box Coordinates
Tokenize Text + Bounding Boxes
Modeling
Train Base Model
Training
With/Without ImageNet Weights
With/Without Augmentation
EfficientNetB0
MobileNetV2
ResNet50
InceptionResnetV2
VGG16
Training
Bert-Based
Roberta-Based
Dutch Bert-Based
Dutch Roberta-Based
Term Frequency - Inverse Document Frequency-Based
Training
LayoutLM
LayoutLMv2
LayoutLMv3
LiLT
UDOP
Training
Weighted Ensemble
Simple Concatenation
Self-Attention
Evaluation on Test Set Dataset 1
Performance Evaluation Based on K-Fold Validation Retrieved Metrics
Accuracy, Recall, Precision & F1-Score
Evaluation on Dataset 2
Model Trained on Dataset 1 Tested on Dataset 2
Accuracy, Recall, Precision & F1-Score

Figure 4.2: Detailed Experimental Set-up Document Classification

Once we have tested and optimized all modalities and settings on dataset 1, we create the final model by combining the different feature architectures, as illustrated by the *Merging/Fusing the Separate Modality Models* objective in Figure 4.2. We test each of the models on dataset 2 through direct inference to measure and validate the generalizability of the model. We combine the results from the evaluations on both dataset 1 and 2 and decide which is in that sense the best performing model, using the metrics described in Chapter 3. This research aims to find a model architecture that can be further trained or applied to new datasets. We describe the way this model should be deployed, for which measures are designed to monitor and maintain the model, addressing the final phase of the CRISP-ML(Q) methodology. However, the primary focus of this research is on the iterative model development process, where various modalities and settings are explored.

4.1. Document Classification

Figure 4.1 illustrates the abstract steps in each of the phases of the CRISP-ML(Q) methodology (see 3.1), with a focus specifically on the iterative model development process, which includes *data preparation*, *modeling* and *evaluation*. Figure 4.3 illustrates the iterative model development process specifically for the development of the document classification model. We show the more detailed steps in Figure 4.2, as referred to as *phase specific subtasks*. The complete model construction setup is divided into five parts: the first being the general part from which each of modality-based models benefit; the relabeling of the data, extracting the visual, textual and layout features. For each of the visual, textual, and layout-based phases, we perform a modality-focused data analysis, prepare the modality data accordingly, and lastly, train and evaluate the selected models. We take the best performing single-modality models and fuse them in the last phase; *Merging/fusing the Separate Modality Models*. We do not include the last part of this research explicitly in the experimental set-up as it follows less of a step-wise plan, describes how the model should be deployed, and what further steps might be necessary.

The following section provides more details of the set-up designed for the complete document classification experiment. It begins with a description of the dataset used and is followed by a detailed discussion of the iterative model development process. Finally, we discuss how the final model should be deployed, monitored, and maintained. We evaluate the performance of the final model on a different dataset by fine-tuning and testing it on a second dataset, different from the dataset used for initial training.

4.1.1. Business & Data Understanding

Business Understanding

The model we aim to create is a document classification model for asset management documents, specifically within the AEC sector. This model will help automate the organization of documents and serve as an initial step in applying IDP techniques in asset management, paving the way to IAM. Previous research on IDP in the AEC sector has primarily focused on text-based processing methods. However, literature on document classification indicates that

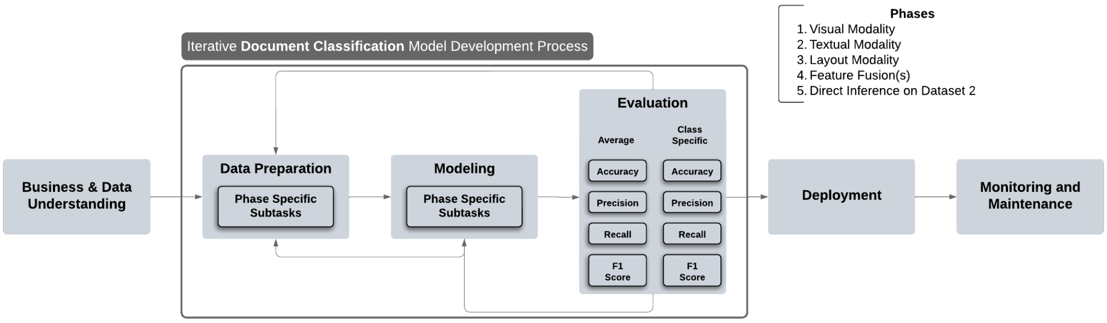


Figure 4.3: Iterative Document Classification Model Development Process

image-based classification yields excellent results, as do models utilizing multi-modal features. Many documents in asset management are drawing- or image-based, lacking substantial text that a model could utilize. Therefore, we aim to evaluate multiple modality-based classification models and test multi-modal classification models to determine which architecture best suits the problem of asset management document classification.The introduction (see Chapter 1) and the literature review (see Chapter 2) further detail the business understanding.

DATA COLLECTION

The data used consists of various types of documents, mainly in PDF format. However, text files and images are also included as these can be converted to PDF and/or PNG format. The data is sourced from the company's storage space and originates from two different asset management cases. Samples of the different document classes included in the datasets have been demonstrated by Figure 4.4[1]. Since the datasets originate from real cases, we do not delve further into the specific sources and contents of the data.

Table 4.1: Dataset 1 Contents

Class	**Num. of document pages**	
	Received Dataset	*Prepared Dataset*
Cross-Section Drawing	50	89
Detail	51	62
Photo	50	57
Installation Diagram	49	94
Floor plan	50	54
Report	57	174
Table	46	152

[1] With Floor Plan as retrieved from CubiCasa5k and Photo as retrieved from Wikipedia

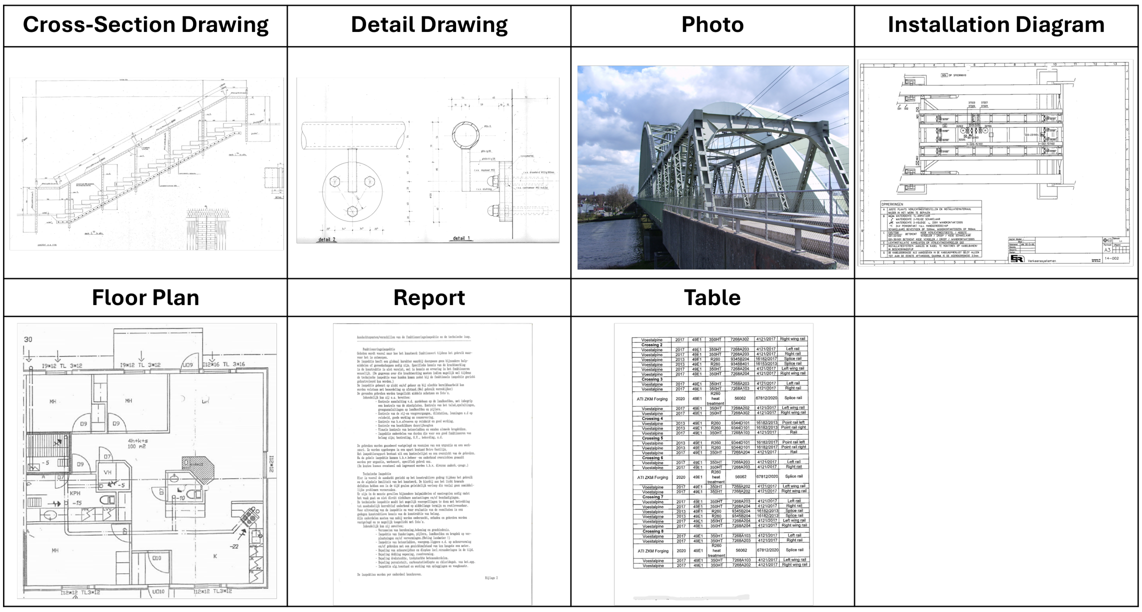


Figure 4.4: AEC Asset Management Document Classes

Dataset 1 The first dataset consists of documents in seven different classes. These seven categories were established by the company. We depict the number of document pages per class in Table 4.1. The documents that are initially labeled by the company are denoted as *Received Dataset*. However, since more unlabeled documents are available for this case, we extend and improve this dataset through a curation and labeling process, of which the result is denoted as *Prepared Dataset* in Table 4.1.

The curation and labeling process begins with a thorough analysis of the documents and establishing an understanding of the document classes. Rough definitions and specific characteristics of the classes are defined through conversations with domain experts. In this research, we focus on document images that in most cases correspond to the entire document. However, multi-page documents might contain multiple document categories, which are split into the correct categories per document page to expand the training dataset. The dataset is then filtered to ensure that only documents aligning with the specified category definitions are included. After relabeling the initial dataset, we add samples by labeling the available unlabeled documents, following the same constructed category definitions. As a result, the dataset grows from 353 to 682 document pages.

As *Dataset 1* is imbalanced, we start the model training by selecting an even number of document pages from each category, a technique known as random undersampling [137]. Specifically, we select the minimum class size, which is 54 in this case, for each class. In this way, we ensure that both the training and test sets are balanced. This approach is chosen because of its simplicity and the ability to balance the classes all at once.

Table 4.2: Dataset 2 Contents

Class	**Num. of document pages**
Cross-Section Drawing	154
Detail	20
Photo	234
Installation Diagram	21
Floor plan	31
Report	424
Table	46

Dataset 2 To evaluate the ability to correctly classify an unseen dataset, we use dataset 2. The data in this dataset originates from a different company and asset management case. This dataset is labeled in subcategories that do not directly correspond to the categories used *dataset 1*. Therefore, we relabel the dataset using the defined class definitions. *Dataset 2* is used to test the trained models through direct inference on *Dataset 2*. This dataset is also heavily imbalanced, as shown in Table 4.2.

4.1.2. ITERATIVE MODEL DEVELOPMENT PROCESS

This section further describes the specific phases and steps within the iterative model development process (see the beginning of this chapter, Chapter 4). As a general part of the modeling process applies to each of the modalities, we first describe these general regards. Then, for each of the modalities, we describe the data preparation, modeling, and evaluation phases separately.

MODELING

We find the optimal architecture using a neural architecture search-like approach, for which the search space, the search strategy, and the performance estimation strategy are defined before starting the search process. This approach is, in terms of the defined steps, rather similar to the quality assurance cycle defined by CRISP-ML [4]. We combine both methodologies to automate the model definition while ensuring quality outcomes. The *requirements & constraints* of the model are defined first, which the search space must meet. Next, we determine how to explore the search space, a process carried out in the *initiate step & task*. Finally, the *choice of the quality assurance method* is made in the performance estimation strategy phase. A crucial part of the quality assurance cycle includes the identification and mitigation of risks, which is not considered in the NAS cycle. These steps are primarily aimed at overcoming possible biases, overfitting, and a possible lack of reproducibility. We address these considerations before starting the NAS cycle in order to minimize these risks as much as possible.

For each search cycle, performance estimation is based on accuracy, recall, precision, and F1-score, both per class and overall, as achieved through k-fold testing - metrics that are generally used in machine learning model evaluation [135]. The model that achieves the highest overall metrics is selected.

Defining the Model Requirements The general requirements for selecting the final model depend on its classification performance, as well as the training possibilities, and are summarized below. Ideally, classification accuracy, recall, precision, and F1 score above 90% are achieved for both *dataset 1* and *dataset 2*, to ensure a reliable classification model [135]. Most of the classification models included in the SLR achieve metrics around this 90% threshold, demonstrating that it has been achieved by state-of-the-art classification models (see Chapter 2). Further, the final model should comply with the following two requirements.

1. The model should be best performing in terms of accuracy, recall, precision and F1 Score in comparison to the other models for the corresponding datasets.
2. The model should be trainable on either an i7 CPU or 1 GPU - NVIDIA Tesla V100.

Defining the Risks We identify a set of general risks that apply to each of the substeps within the iterative model development process, along with their mitigation strategies. First, to address class imbalance and avoid prediction bias towards majority classes, we apply random undersampling. Each class contains at least 54 and up to 190 document pages, therefore making it essential to balance the classes. After random undersampling, the data is split up in a train, test, and validation set. The training set consists of 43 document pages, the validation set of 6 and the test set of 11 document pages per category.In order to prevent over- and underfitting we use an early stopping mechanism that stops training once the validation loss stops improving. Each visual modality model is trained for 150 epochs, utilizing an early stopping callback that begins at 50 epochs and has a patience of 15 epochs. This approach prevents the model from unnecessary training time, but also mitigates the risk of overfitting by stopping training once validation performance does not improve [138], and is a generally used technique in related research [24, 31, 35]. To ensure that the model generalizes well to new data, we evaluate it on a related external dataset. This concern is taken into account by various state-of-the-art document classification models as well [34, 87, 139]. To make sure the training is scalable and efficient, all experiments are executed on an NVIDIA Tesla V100 GPU.

4.1.3. Establishing the Base Model

Since all the included models are CNN-based, as identified and selected through our literature review, our aim is to compare their results to a self-constructed traditional machine learning model that uses manually extracted document features. We conduct an exploratory data analysis to identify distinctive features that differentiate between classes, based on previous document classification research using handcrafted features [140, 141]. We then transform these features into a dataset that we classify using a logistic regression model. We aim to incorporate features from each of the three modalities into the base model.

4.1.4. ESTABLISHING THE VISUAL MODALITY COMPONENT

DATA PREPARATION

For the image modality, we follow two main preparation steps. First, we resize the document images to the dimensions required by the model to be trained [50, 53, 120, 124, 126]. Secondly, we prepare an augmented dataset [27, 142]. Since some images in the dataset are tilted or upside-down, we explore whether adding rotated versions of these images enhances model performance. Each image in the dataset is rotated 90°, 180°, and 270°. We add these rotated images to the training and testing set. As we aim to maintain a balanced dataset, we select 54 images from each of the classes using random undersampling. Similarly for the augmented dataset, we select the exact same images, but augmented, resulting in a dataset of 216 images per class.

MODELING

Based on the defined requirements and constraints, the search space is filled. The specific CNNs models are based on well-performing models from previous research (see Chapter 2). We list the specific contents of the search space in Table 4.3.

Table 4.3: Visual Modality Search Space

Model Settings	Options/Details
Neural Network Architectures	EfficientNetB0, MobileNetV2, ResNet50, VGG16, Inception-ResNet-V2
Pretraining	ImageNet Weights
Image Augmentation	Rotation

The search strategy is defined by combining the options in the search space in a specific way, as described in this section. The first step involves evaluating the impact of applying pretrained weights to the models. Since ImageNet weights are commonly implemented as a parameter in CNN models [22, 26, 31, 86, 88], this provides a straightforward method to assess their effect. We compare the performance of each of the models with and without these weights. Subsequently, we apply the setting that yields the best classification results to later models. Finally, we incorporate the augmented images into the training and testing sets for the third experiment. This experiments tests the impact of data augmentation on the performance of all neural networks, investigating whether augmentation and dataset size affect the performance of the different architectures.

1. Evaluate Impact ImageNet Weights on Model Performance
2. Evaluate Neural Network Architecture Performances
3. Evaluate Impact Image Augmentation

The risks associated with this search space and strategy align with the risks identified earlier. Notably, the risk of scalability applies to this modality as the size of the images quickly makes the model slower, and more complex architectures worsen this issue. This issue causes Out-Of-Memory issues. In this study, the models are trained using an NVIDIA Tesla V100 GPU that runs on Azure Machine Learning services.

4.1.5. ESTABLISHING THE TEXTUAL MODALITY COMPONENT

DATA PREPARATION

First, we extract text from the document images using OCR. As various OCR tools are available, we conducted a comparative analysis to evaluate their performance. This analysis, which includes PyTesseract, EasyOCR, Keras OCR, and Azure OCR, is detailed in the appendix (see Appendix D). Azure OCR emerged as the best performing OCR, and thus we use it for text extraction.

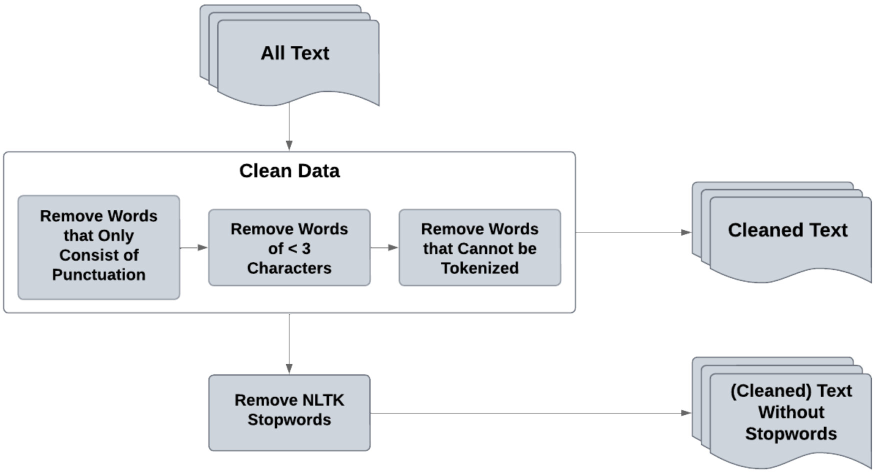

All Text
Clean Data
Remove Words that Only Consist of Punctuation
Remove Words of < 3 Characters
Remove Words that Cannot be Tokenized
Cleaned Text
Remove NLTK Stopwords
(Cleaned) Text Without Stopwords

Figure 4.5: Text Cleaning Process

After extraction, we clean the data. Although most multi-modal document classification models do not explicitly describe their text cleaning process, text classification in AEC does deem the cleaning process to be important [48, 61, 66]. The two cleaning phases are illustrated in Figure 4.5, and defined as follows. First, the initial cleaning of the text is done by removing all words that consist only of punctuation marks. All words that have only one or two characters are removed. Finally, Dutch tokenizer BERTje [127] goes over the remaining words and tries to tokenize the words. As it is a Dutch tokenizer, it will only recognize Dutch words or parts of Dutch words. Words that the tokenizer cannot process are removed. Secondly, we remove the company name from the text data, so that the model does not learn the company name, nor relates it to any of the classes. After these steps, we have derived *Textset 1*. Lastly, the Dutch stopwords dataset, as made available by the Natural Language Toolkit [2] is used to remove stopwords from the texts. With these steps, we derive *Textset 2*. The datasets are both used for evaluation on the various text classification architectures. This cleaning process is in line with previous research performing tokenizer-based text classification [143].

[2] NLTK Website

MODELING

The search space for the textual modality focuses primarily on preprocessing and classification approaches. Table 4.4 denotes the search space.

Table 4.4: Textual Modality Search Space

Model Settings	**Options/Details**
Preprocessing Technique	*Text set 1* (Cleaned Text), *Text Dataset 2* (Without Stopwords)
Tokenizer	BERT$_{\text{based}}$ Encoding, RoBERTa$_{\text{based}}$ Encoding, Dutch BERT$_{\text{based}}$ Encoding, Dutch RoBERTa$_{\text{based}}$ Encoding, TF-IDF
Classifier	Tokenizer Related Classifier, Simple Deep Learning Model (for TF-IDF)

We use four different tokenizers for text tokenization, and their corresponding sequence classification models are used to classify the texts. The BERT- and RoBERTa-based tokenizer/classifiers where selected as they have shown good classification in various previous researches [20, 22, 24–27, 29, 31, 32, 34–36, 55, 76, 77, 82, 84, 96, 97]. Research using Dutch models has shown improved classification performance compared to English-based models, which is why we apply the Dutch versions of the models as well [127–129]. Throughout the remainder of this thesis, we refer to BERTje as the Dutch BERT model, and RobBERT as the Dutch RoBERTa model. A more detailed description of the tokenizers is provided in Section 3.3.2. For each model, we evaluate the performance on both *Textset 1* and *Textset 2*.

In addition to the pretrained tokenizers, we evaluate Term-Frequency - Inverse-Document-Frequency (see Section 3.3.2), as widely used in AEC text document classification [63, 65, 66], as well as various other text classification applications[68, 131, 132, 144]. We use a simple CNN model to classify using the TF-IDF data, as resulted as the best performing model for both and . TThis simple CNN model was compared to various traditional classification models. More details on this comparison have been included in the appendix (see Appendix F).

4.1.6. ESTABLISHING THE LAYOUT MODALITY COMPONENT

DATA PREPARATION

To retrieve the input required for the layout-based models, the text and corresponding bounding boxes must be extracted from the documents. For this, we use the OCR that performs best in the OCR analysis (see Appendix D). The data is saved in a json file from which the bounding boxes and text are extracted. These are tokenized and encoded using the model-corresponding tokenizer. These tokenizers are of a textual-layout or even a textual-vision-layout transformer type and extract the information needed for the layout-based encoding tasks. Before tokenization and encoding, the bounding boxes are normalized and scaled to match the input requirements.

MODELING

The existing approaches for classifying documents using layout-related features are always combined with textual- or visual-based features [20–23, 25, 55]. The search space consists generally of transformer-based models that performed well in literature and that have a model available in the Huggingface hub[3] (see Table 4.5). From the literature review (see Chapter 2), we find that LayoutLM-based models [20, 22, 55] generally perform well accuracy-wise. Furthermore, the UDOP [21] model and the LiLT [23] model achieve state-of-the-art results. The search space consists of these modality-combining models, and the search strategy involves sequentially training and evaluating each of these models.

In the literature review, we identified graph-based models as a different strategy to classify documents using layout features. However, graph-based models have not yet achieved state-of-the-art classification performance and are therefore left out of this study.

Table 4.5: Layout Modality Search Space

Model Settings	Options/Details
Layout-Modality Based Model	LayoutLM, LayoutLMv2, LayoutLMv3, UDOP, LiLT

4.1.7. ESTABLISHING THE FEATURE FUSION APPROACH

Once we have established the best performing classification architecture for each of the modalities (Sections 4.1.4 , 4.1.5 and 4.1.6), the aim is to find the optimal fusion method. During this phase, we evaluate the performances of the models for the different modalities to select the models to be fused. Since the models tested for the layout modality are already multi-modal, we only fuse the best performing textual and visual models. The requirements and constraints outlined in Section 4.1.2 align with those of this phase.

The search space consists primarily of late-fusion methods, namely weighted ensemble, simple concatenation, and self-attention-based fusion. These fusions are selected because they have shown good performance in previous research (see Chapter 2) [20, 26, 26, 27, 29, 32, 35, 73, 97, 145]. The weighted ensemble, simple concatenation, and self-attention-based fusion models can all be categorized as late fusions. However, The LayoutLM models utilize an early fusion method as the modalities are fused into one single encoding at the beginning of the classification process. The UDOP and LiLT models employ a hybrid fusion strategy, as the data is fused multiple times throughout the classification process.

[3] Huggingface Website

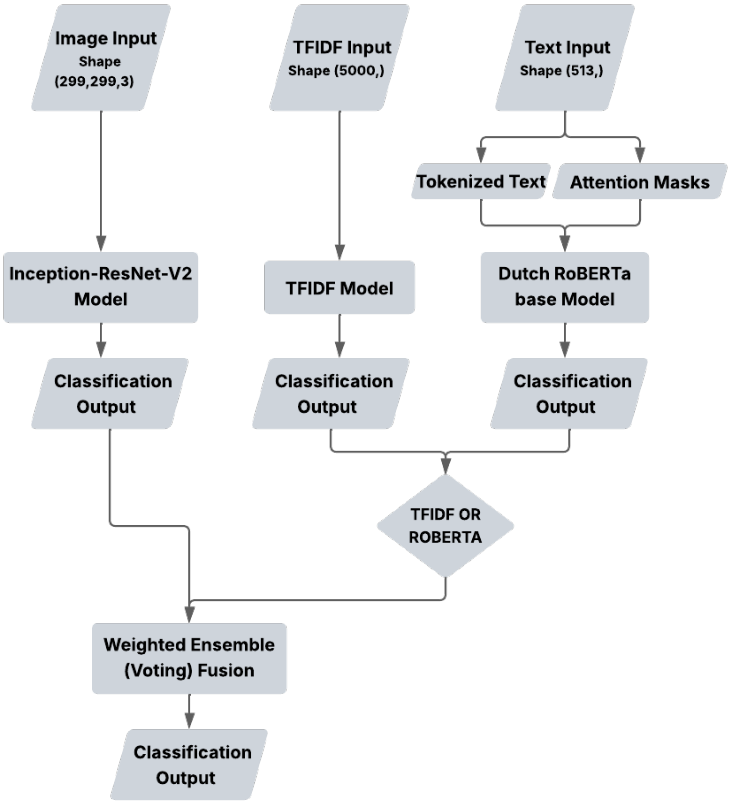


Figure 4.6: Weighted Ensemble Architecture

WEIGHTED ENSEMBLE FUSION

In this fusion, as illustrated in Figure 4.6, each included single-modality model generates a prediction, typically outputted as logit values, with as many nodes as there are classes. The class with the highest logit value is usually selected as the classified label, as these logit values represent the confidence of the model in predicting a certain label [146]. In the weighted ensemble fusion we establish, these prediction logits are concatenated using dynamic weights. The initial weights are determined based on the previous performance of the included models, with higher weights assigned to models that perform better in earlier tests. The weights are trained, as well as the individual models throughout the training process. The resulting logits are then classified through a final classification layer where the prediction logits from the individual models are multiplied by the retrieved weights. This weighted ensemble approach, used by several previous works in multi-modal document classification [26, 27, 29], attempts to improve the overall classification performance by leveraging the strengths of individual models.

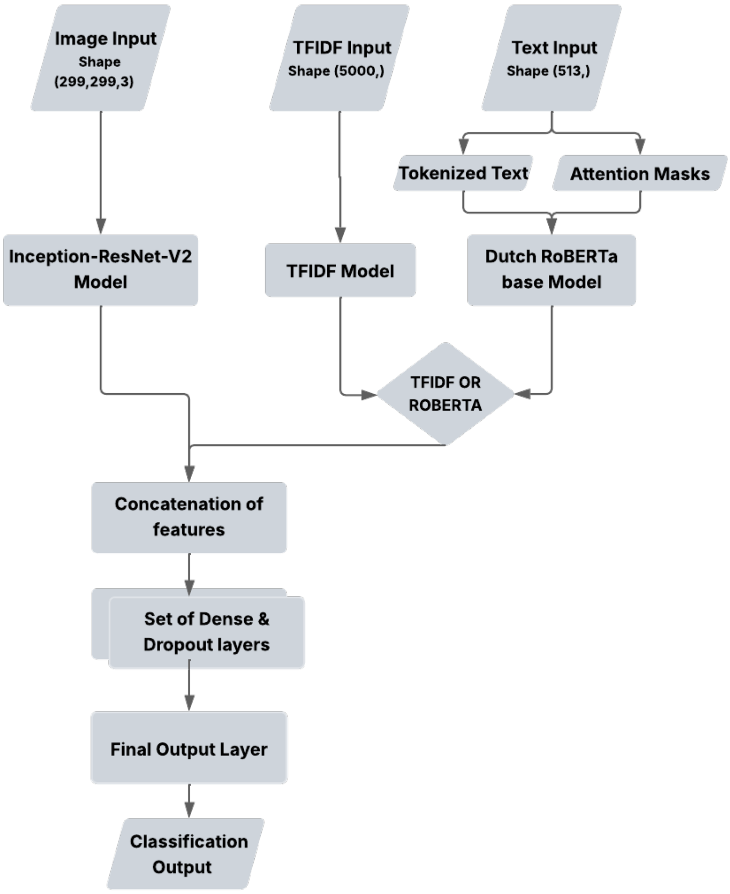


Figure 4.7: Simple Concatenation Architecture

SIMPLE CONCATENATION FUSION

As demonstrated in Figure 4.7, the individual models first extract their respective features in this concatenation model. Each modality model has its own layers, including input, hidden, and output layers, similar to several other models using a simple concatenation fusion [26, 35, 145]. The outputs from these individual modality models are concatenated in a concatenation layer, leading to the final classification via a softmax activation function.

The key difference between the weighted ensemble fusion and the simple concatenation fusion is that in the weighted ensemble model, predictions are made by the included individual models, whereas in the simple concatenation model, the output features of the two models are concatenated before making a prediction. Additionally, the weighted ensemble fusion applies weights based on model performance, while the simple concatenation model fuses the feature outputs equally.

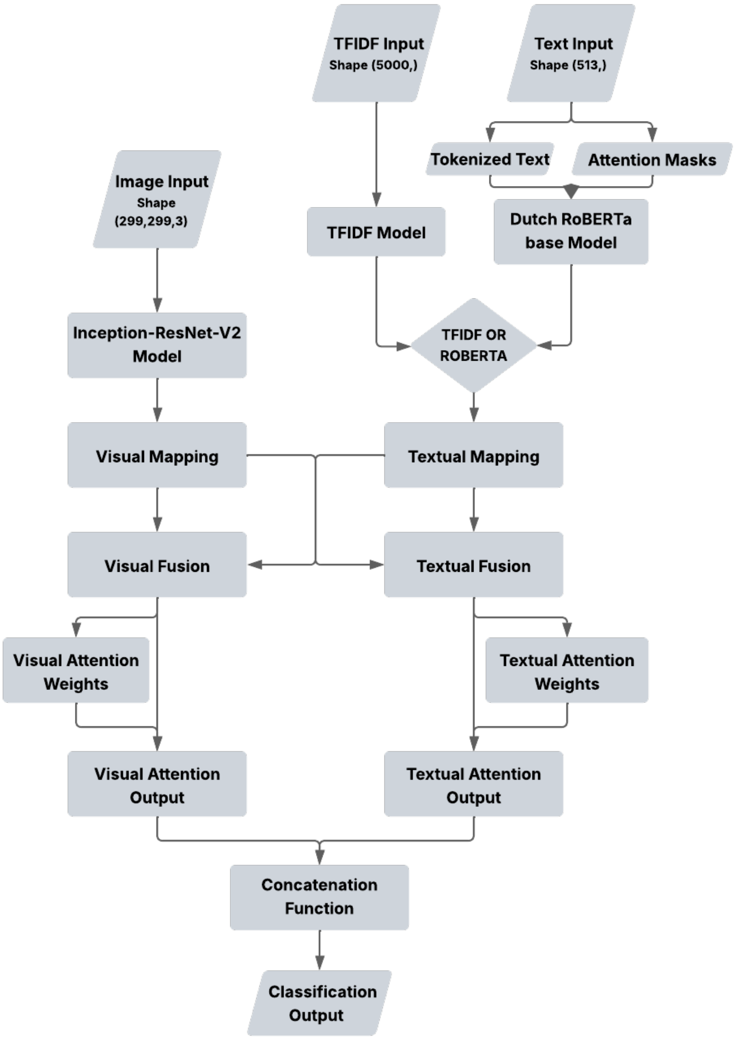


Figure 4.8: Self-Attention Based Fusion

SELF-ATTENTION BASED FUSION

The attention mechanism in machine learning, generally used in computer vision and NLP applications, is based on the human visual attention system. Human visual attention is defined as the ability to dynamically restrict processing to a subset of the visual field [147]. The attention mechanism in neural networks attaches a layer of weights to the input data that identifies the most important features in the data, overcoming the focus on irrelevant features that diminish the generalization ability of the model. Through neural networks, the attention mechanism learns further which parts of the input data to focus on. Self-attention is a variant of the attention mechanism and computes a representation of a sequence by relating different positions to the sequence itself, reducing dependence on external information [148]. In neural networks, attention is computed by weighted layers to define which parts of the data sample to focus on, often denoted as feature maps or attention maps. Computing these maps as well as integrating them with the other layers of the model is computationally much more expensive than weighted ensemble fusion and simple concatenation fusion. Our implementation of the self-attention based fusion is illustrated in Figure 4.8. This fusion is based on the self-attention based fusions as applied in earlier research [20, 32, 73, 97].

4.1.8. EVALUATION & ESTABLISHING DEPLOYMENT GUIDELINES

Within this last phase, we evaluate which model performs best in terms of both test performance and generalizability on a new dataset. For this best performing model, we evaluate which steps need to be taken in order to deploy it, considering the final classification performance results as well as the situation in which the model is to be used. We provide guidelines for ensuring and maintaining classification quality, following the deployment and monitoring & maintenance steps as described in Chapter 3.

5

Results

To better understand the data we aim to classify, we first conduct an exploratory data analysis. This analysis is performed separately for each modality, focusing on their specific characteristics. Additionally, we analyze inter-class similarity and intra-class compactness. One of the main findings from these data analyses is that for all modalities the *Photo* class stands out the most for different characteristics (E.1,E.3,E.4,E.5,E.6,E.8,E.9). Specifically, *Photo* documents exhibit lower brightness (E.3), more variation in the most dominant colour (E.4), and a higher entropy (E.5), indicating a more complex structure. Additionally, *Photo* documents contain fewer words (E.6) but use more unique words (E.7). For images of the *Photo* class, OCR detects significantly fewer bounding boxes (E.8), and the average area of these bounding boxes is approximately seven times larger than that of other classes (E.9). Furthermore, we observe that documents in the *Report* class are generally of portrait format, while documents in the *Installation Diagram* class are typically in landscape format (E.2). The other classes vary more in format. We include the detailed data analysis in the appendix (see Appendix E).

5.1. Multimodal Document Classification Model

This section outlines the classification performances of the various models as described in the experimental setup (see Chapter 3 & 4), categorized by modality.

As we balance the data using random undersampling, the stochasticity of training a deep learning model increases [149]. Alongside other factors such as random weights and the optimization algorithm, we need to account for it in the model evaluation. To do so, we train each model using 5-fold cross-validation, a widely used validation approach used in previous studies on document classification as well [29, 31, 134], where the average classification performance metrics are recorded to compare performance between models.

In order to better understand the predictive behavior of the classification models in terms of the model classes, we evaluate the performance per class specifically, using one standard test set. For this purpose, we use the first (approx.) 20% of files per class - where "first" refers to the first files when sorting them alphabetically - as a standard test set. This more static testing approach, which allows comparing different models in the exact same training, validation, and testing set, is similar to the approach taken in many of the studies evaluated in the literature review (see Chapter 2) [26, 27, 32, 34, 82, 86, 88]. We then retrain the model using the optimized settings and parameters. Since these test results are not averaged, direct conclusions cannot be drawn. However, these confusion matrices allow us to compare classification performance across different classes within the standard test set. In the appendix, we include the visualized classifications of the best-performing classification models on this standard test set (see Appendix G.1).

5.1.1. BASE MODEL

The initial model we construct is the baseline for comparison with the other models we develop. This model is built by manually extracting features that we further analyze in the exploratory data analysis (see Appendix E). These features are of various modalities, incorporating visual, textual, and layout characteristics. The specific features are listed in Table 5.1.

For the visual features, we extract several attributes: the number of different colors in the image, the aspect ratio (width-to-height ratio), brightness, contrast between the brightest and darkest colors, and the number of edges (pixels where the color changes drastically) in the image. Additionally, we identify the most dominant color and calculate the image entropy, which measures the variability of pixel intensities and, in this way, captures the complexity of images.

For the textual features, we extract the total number of words per document and the number of unique words. Furthermore, we use a library of words relevant to document categories, identified through exploratory data analysis (see Section E). We examine whether each of the words from the library is present in the document texts.

The layout modality is incorporated into this base model by determining the number of bounding boxes per document and calculating the average area of these bounding boxes per document. Additionally, we extract the number of tables that can be extracted using the Img2Table[1] package, which is a simple Python package trained to recognize tables in images.

[1] Img2Table Github

Table 5.1: Base Model Features

Textual Features	**Visual Features**	**Layout Features**
Number of Different Words	Number of Different Colours	Number of Bounding Boxes
Total Number of Words	Aspect Ratio	Average Area of Bounding Boxes (in pixels)
Contains "detail", "scale", "view", "section", "floor plan", "drawing", "wall view", "scheme", "connection diagram", "list", "table", "report", "ymvk"	Brightness	Number of Recognized Tables
	Image Contrast	
	Edge Density	
	Dominant Colour	
	Image Entropy	

The architecture selected for the base model is a multinomial logistic regression classification model utilizing the "lbfgs" solver. The model achieves a classification accuracy and recall of 84.74%, while achieving a precision and F1-score of 84.65% and 84.23%, respectively. As the model was trained, validated and tested on balanced datasets, the resulting values for accuracy and recall are equal. We include these metrics in the figures where the performance of the other models is compared (see Figures 5.2, 5.3, 5.4, 5.5, 5.6). This classification performance is comparable to the performances of other handcrafted feature-based models described in previous research [1, 140, 141]. Although various handcrafted features achieve substantive performance, they are eventually surpassed by deep learning-based models. Furthermore, we evaluate the performance of the retrained model on the standard test set, where we see that classes *Cross-Section Drawing*, *Installation Diagram*, and *Report* are predominantly predicted correctly. Class *Table*, however, is frequently misclassified as *Report*. Figure G.1 (included in the appendix) visualizes the prediction on the test set for one fold.

5.1.2. IMAGE MODALITY

As specified in Section 4.1.4, the model architectures being examined for the visual modality are the CNN architectures MobileNetV2, ResNet50, VGG16, Inception-ResNetV2 and Efficient-NetB0. As each of the CNN architectures desires a specific input shape, we reshape the document to the required shapes [50, 53, 120, 124, 126].

Figures 5.1 and 5.2 present the performance metrics of the CNN models. First, we evaluate the results of not applying the *ImageNet* weights in the classification model compared to using *ImageNet* weights, of which the classification performance metrics are shown in Figures 5.1 and 5.2. In this way, we evaluate whether pretraining the models on non-directly related datasets supports the performance of the classification models. We generally observe an increase in performance when applying *ImageNet* weights. The improvement in model performance varies per model; MobileNetV2 benefits the most from applying *ImageNet* weights, with about a 55% increase, followed by VGG16 with approximately 44%, and ResNet50 with around 27%. Inception-ResNet-V2 appears to be less dependent on *ImageNet* weights, showing only about a 7% increase in performance. Interestingly, the performance of EfficientNetB0 worsens with the application of *ImageNet* weights. Comparing these results to the base model, we observe that only Inception-ResNet-V2 outperforms the base Model, even though slightly, achieving an average accuracy, and recall of 85.41%, an average precision of 87.65%, and an average F1-score of 86.6%. ResNet50 closely follows in classification performance with an accuracy, recall, and F1-score of 82.34%, and precision of 84.06%. The model is followed by VGG16 and MobileNetV2 which achieve performance metrics around 70%. For each of the visual models, except EfficientNetB0, the precision is highest among the model performance metrics. The precision captures the average number of correctly predicted positive instances out of all instances that the model predicted as positive, showing that the selected models are particularly well in correctly identifying relevant documents for each of the classes. Since these models are all convolutional neural networks, it is not feasible to determine the exact reasons why one outperforms the other. However, we can analyze their architectures and the specific types of problem and datasets where they excel in classification.

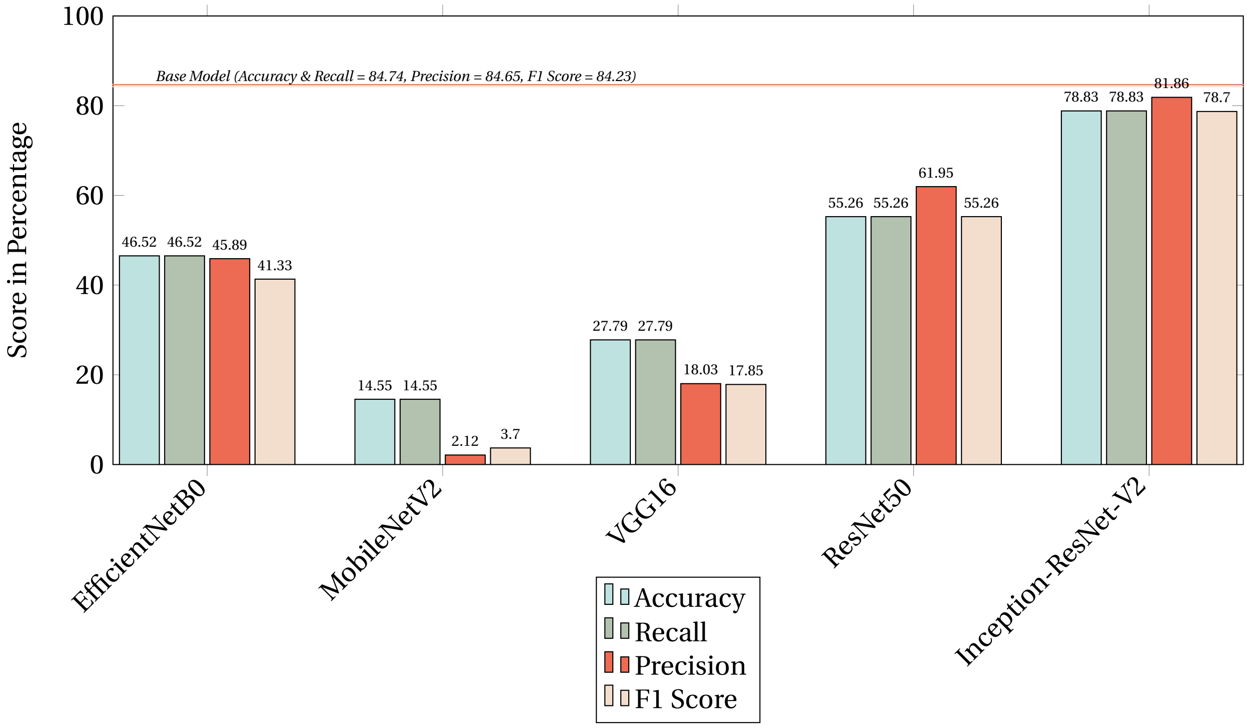


Figure 5.1: Dataset 1 Classification Performances (Image Modality - Without *ImageNet* Weights)

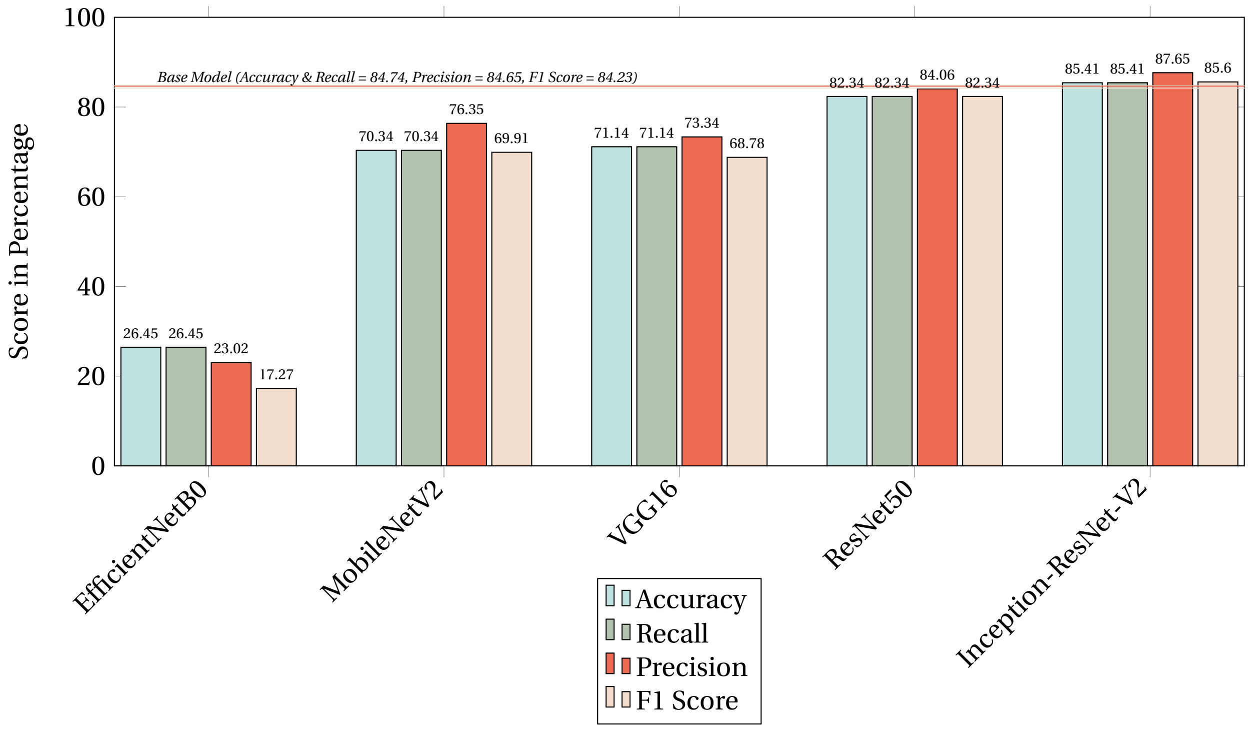


Figure 5.2: Dataset 1 Classification Performances (Image Modality - With *ImageNet* Weights)

MobileNetV2 and VGG16 are rather lightweight network with less parameters and a smaller capacity to learn complex data from scratch [53, 124]. Because of this, pretraining the model on other datasets, *ImageNet* in this case, provides stronger starting weights for the models. Even though *ImageNet* images are not directly related to document images, the weights help to recognize simple information in images, helping models to converge faster.

In contrast, deeper networks such as Inception-ResNet-V2 and ResNet50, with their deeper and more complex architectures involving residual connections, are able to learn more complex features from scratch [50, 120]. Even though they still do benefit from transfer learning, they rely less on the *ImageNet* weights than the simpler lighterweight models.

EfficientNet is designed as a lightweight model family, with different variants (B0-7) that scale up in size and complexity. In this study, we use the B0 version, which has less ability to handle a smaller amount of parameters and learn complex patterns in data compared to more advanced versions. EfficientNetB0 takes longer to learn data in more complex problems, especially when trained from scratch. Pretraining the models on *ImageNet* weights adds parameters, which EfficientNetB0 can only handle to a limited extent [126]. This result is reflected by the study constructing EmmDocClassifier as well, where an EfficientNetB0 does not improve through pretraining either [31]. In contrast, Ferrando et al. [29] train and test a hybrid model using an EfficientNet as well as a BERT model. Their hybrid model using an EfficientNet does improve through pretraining, however no specific version of EfficientNetB0 is detailed, and it is possible that a more improved version does benefit from pretraining.

These found results are generally reflected by previous research as well. Sajol et al. [88] compare different models using ImageNet pretraining compared to their ConvNext2 model. We do not test this model in our research, however in complexity - based on number of parameters - the Inception-ResNet-V2 is most similar to this model. In the study, the ConvNextV2 outperforms the other less complex models, similar to our Inception-ResNet-V2. Afzal et al. [51] evaluate both document pretraining as well as ImageNet pretraining for four different models of which we only test 2 - namely ResNet50 and VGG16. Comparably to what we find, both models improve through the application of ImageNet weights.

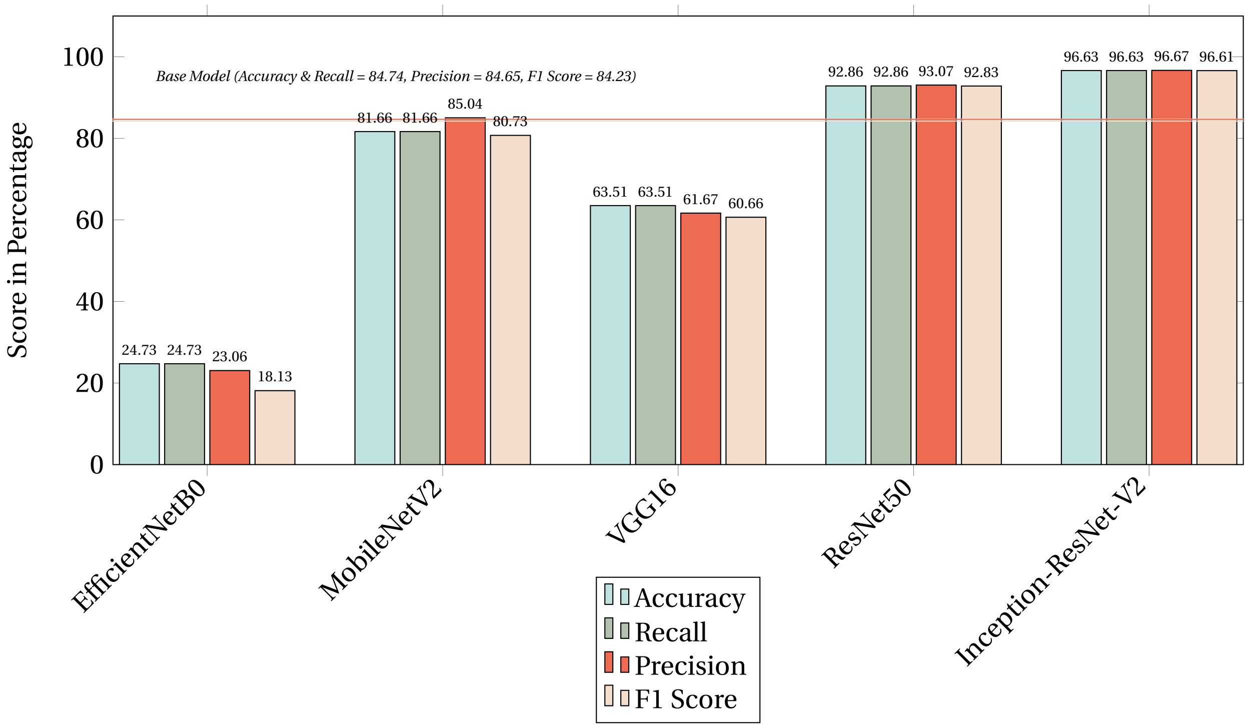


Figure 5.3: Dataset 1 Classification Performances (Image Modality - Augmenting)

Data Augmentation

Data augmentation is often performed to increase the size of the training dataset and improve model generalization [150]. As the dataset with which we work is rather small, augmentation is a logical step. As some of the document images in the dataset are already rotated, we apply rotation-based augmentation by rotating each image rotated by 90°, 180° and 270°, and including these new images to the training, validation, and test sets. Figure 5.3 shows that adding augmented images improves the performance of the MobileNetV2, ResNet50 and Inception-ResNet-V2 models. However, augmentation does not lead to performance improvement for the EfficientNetB0 and VGG16 models. This could be attributed to model complexity and capacity: more complex models like Inception-ResNet-V2, ResNet50 and MobileNetV2 are more flexible to learn from augmented data, helping these models to generalize better and reduce overfitting. Lighterweight EfficientNetB0 and VGG16 struggle to effectively capture complex classes, resulting in augmented images being perceived primarily as noise. For all models that benefit, an improvement of approximately 10% is observed in performance metrics. The augmented models are run using the same number of epochs and the same early stopping mechanism. However, on average, they converge 10-20 epochs later in the training process, suggesting that they need more iterations to fully make use of the augmented data. Various of the previous studies make use of data augmentation in different forms in their preprocessing process, however do not evaluate the effect of augmentation [26, 51]. The study introducing DocXClassifier, on the other hand, evaluates more aggressive augmentation techniques and finds that they offer slight improvements in performance [51].

TESTING ON THE TEST SET

As Inception-ResNet-V2 with augmentation achieves the best classification performance, we train the model again and test it on the testset. The prediction on the standard test set shows that the Inception-ResNet-V2 model predicts the *Photo* class most accurately for this test set, while it performs the least accurate for the *Cross-Section Drawing* class, as it seems to confuse it with the *Detail*, and *Floor Plan* classes. Furthermore, it seems to find the *Table* class closely related to the *Report* class, since it only incorrectly predicts the samples to be of that class.

5.1.3. TEXTUAL MODALITY

The classification models that we evaluate in this section are trained using two differently preprocessed datasets, diversified in the manner that they are cleaned up, following the text cleaning process as illustrated in Figure 4.5. From this cleaning process *Textset 1* and *Textset 2* emerge, as illustrated in Figure 4.5. The evaluated models are, as included in the textual modality search space (see Table 4.4), a TF-IDF-based model, the BERT$_{\text{based}}$ model, RoBERTa$_{\text{based}}$ model and their Dutch variations.

TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

Term Frequency - Inverse Document Frequency is merely a feature and not directly connected to a specific classification model. Traditionally, TF-IDF is used with classical classifiers such as Support Vector Machines, Decision Trees, Gradient Boosting, Logistic Regression, Random Forest, KNeighbors and Naive Bayes [132]. Recently, however, simple CNNs are used in combination with the TF-IDF feature as well [144, 151]. In a comparative analysis where we evaluate a number of traditional machine learning models as well as a simple CNN model (see Appendix F), we discover that only a simple CNN surpass the base model in performance. No significant difference in performance between the *Textset 1* and *Textset 2* can be observed as both achieve accuracy, recall and F1 score of 88.71%, and a precision of approximately 90%.

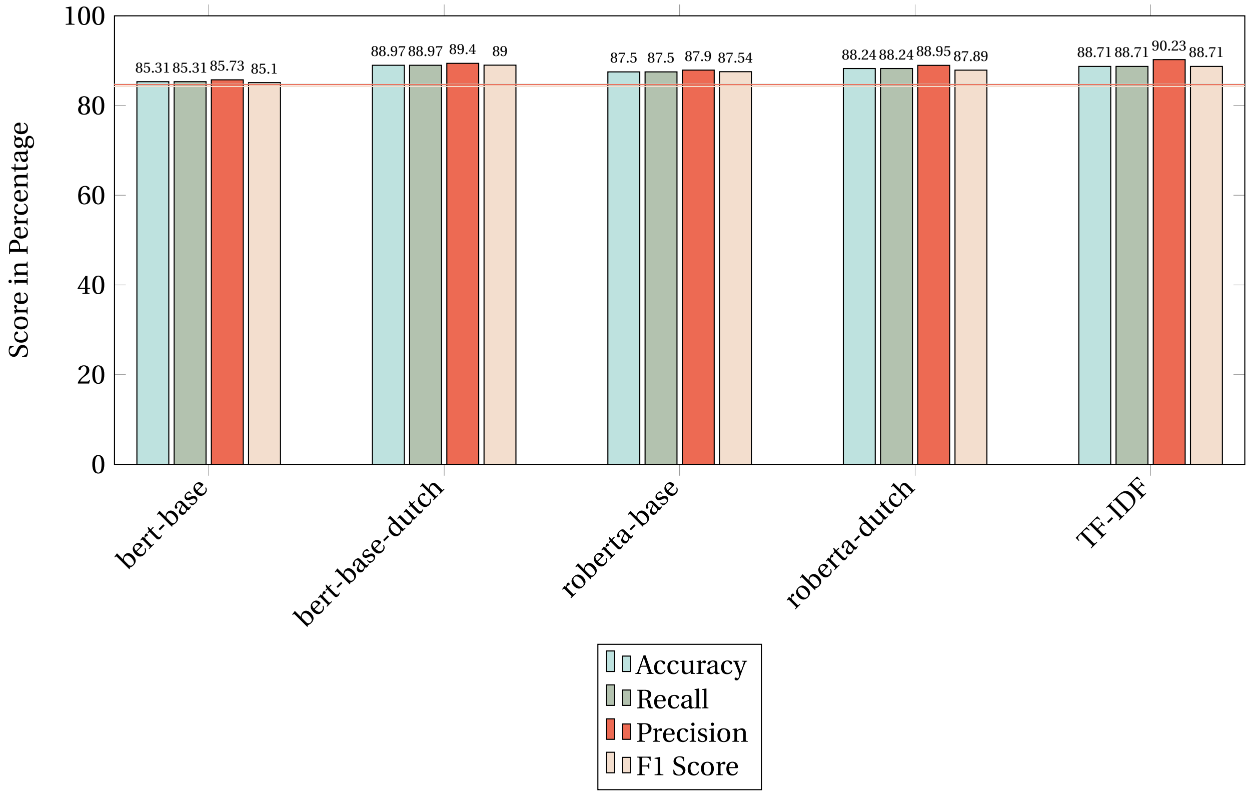


Figure 5.4: Dataset 1 Classification Performances (Text Modality - Clean)

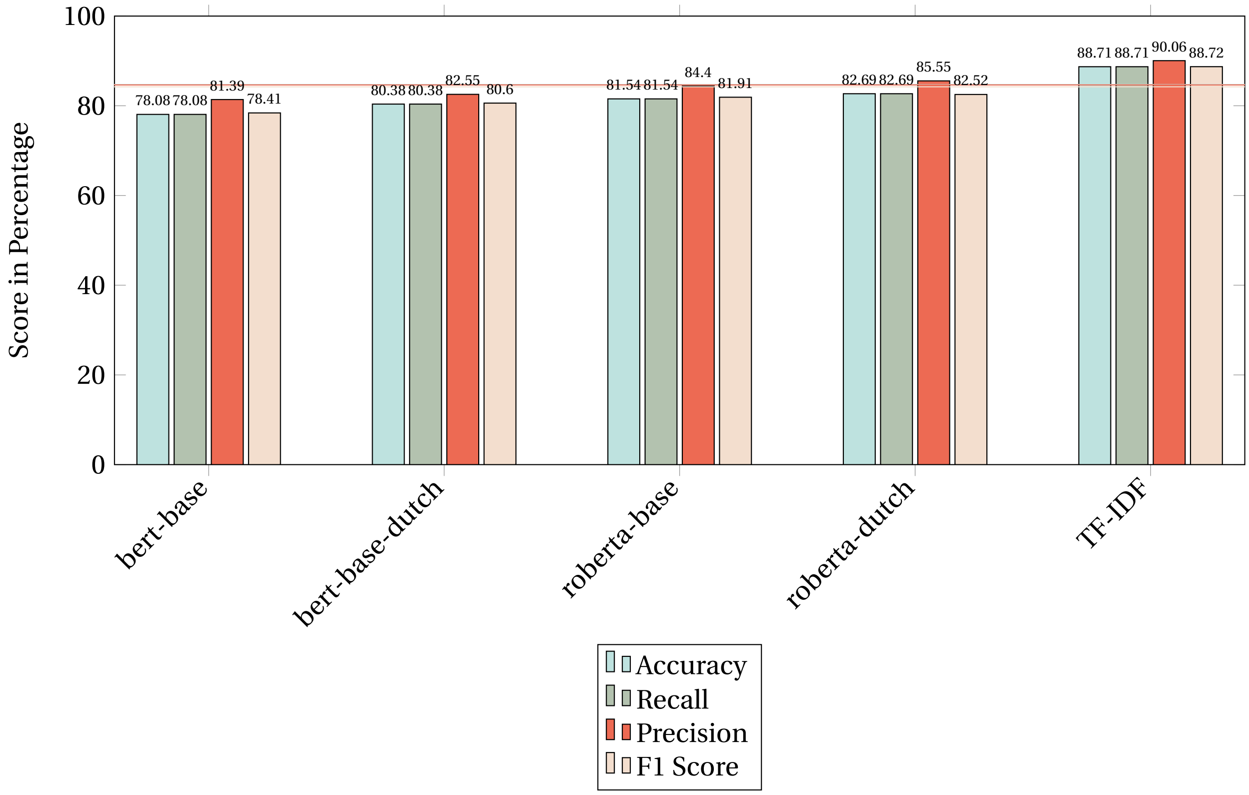


Figure 5.5: Dataset 1 Classification Performances (Text Modality - Without Stopwords)

TOKENIZER BASED MODELS

Furthermore, we assess the classification performance of tokenizer-based models on both *Textset 1* and *Textset 2* (see Figures 5.4 and 5.5). The tokenizer models perform comparably to the TF-IDF model on *Dataset 1*, but show a decrease in performance on *Textset 2*. Among these models, the two Dutch tokenizer models achieve the best performance, closely followed by RoBERTa$_{\text{base}}$, with BERT$_{\text{base}}$ model following behind. The RoBERTa architecture builds upon BERT model by, among other improvements, optimizing training procedures and using a larger dataset [101]. These optimizations contribute to an improved performance of RoBERTa compared to BERT, not only within the scope of this case in broader applications as well [143, 152]. Given that the texts we are classifying are in Dutch, it is unsuprising that models pretrained on Dutch texts (bert-base-dutch and roberta-dutch) slightly outperform their English counterparts, a finding also supported by previous research [127, 128].

TESTING ON THE TEST SET

The TF-IDF model and Dutch RoBERTa$_{\text{base}}$ model are retrained to be tested on the standard test set, which provides further insight into the prediction behavior. Both models perform optimally for the *Photo* class and perform relatively well for the *Detail* class. The TF-IDF model also performs well for the *Floor Plan* class, but shows confusion among the other classes, particularly predicting *Cross-Section Drawing* as *Detail* or *Floor Plan*. Misclassifications between the *Detail* and *Floor Plan* classes indicate a similarity between these samples. However, the *Report*, *Installation Diagram* and *Table* classes are misclassified into various classes without showing any clear patterns. These predictions are visualized by Figures G.3 and G.4 in the appendix.

The RoBERTa model shows even less consistency in its misclassifications (Figure G.4). Only the *Detail* class shows a prediction pattern similar to the TF-IDF model, with samples being misclassified as *Floor Plan*.

5.1.4. LAYOUT MODALITY

For the layout modality, we evaluate the LayoutLM, LayoutLMv2, LayoutLMv3, LiLT, and UDOP models, which are further detailed in the experimental set-up (see Section 4.1.6). Figure 5.6 demonstrates the classification performances of the layout-based models. The figure evidently exhibits the best performance for the LayoutLMv2 model. Interestingly, the LayoutLMv2 model performs better than its descendant LayoutLMv3. The original LayoutLM performs the worst of all models, illustrating the added value of incorporating visual features. The decreased performance for the LayoutLMv3 model suggests that LayoutLMv2 is better suited for our data, possibly caused by differences in the model, such as the backbone architecture, the pretraining objectives, or the tokenization methods used.

The LiLT and UDOP models perform well compared to the LayoutLM model, but do not surpass LayoutLMv2. It should be noted that LiLT only uses textual and layout information, as well as the relation between them, but does not incorporate visual information [23]. Despite this, LiLT achieves results that are relatively close performance results to the evaluated models that incorporate all three modalities.

These results contrast with the classification performance in literature, where LayoutLMv3 and the UDOP model generally perform best [22, 25] compared to the other models used in this research. Possible reasons include that the commonly used RVL-CDIP dataset better fits these models, or that our chosen hyperparameters and training settings are not optimal for training those specific architectures.

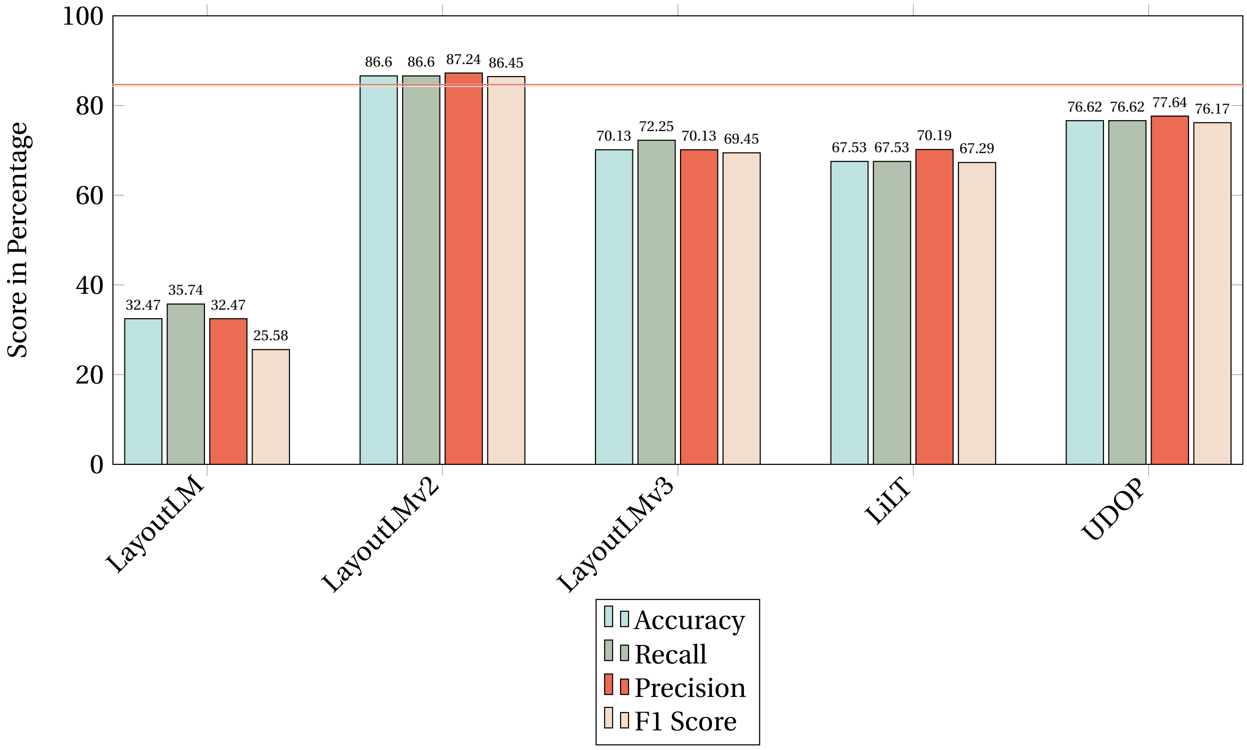


Figure 5.6: Dataset 1 Classification Performances (Layout Modality)

Testing on the Test Set

Next, as with the other models, we retrain the best-performing classification model using the optimal settings identified through cross-validation and evaluate it on our standard test set. LayoutLMv2 in this case (Figure G.5) correctly classifies all samples in the *Photo* class. Classes *Detail* and *Report* are correctly classified in most cases. Similarly to the TF-IDF model (see the confusion matrix in Figure G.3), misclassifications for the *Cross-Section Drawing*, *Detail*, and *Floor Plan* classes occur only within this trio. Misclassified samples of the *Installation Diagram* class are classified as *Cross-Section Drawing*, *Floor Plan*, and *Report*. The *Report* and *Table* generally misclassify into each other as well.

5.1.5. Best Performing Models

After evaluating and comparing the modality-specific models, we compare their classification performance. Figure 5.7 shows the classification performance metrics as retrieved through the averaged cross-validation processes for the base Model, Inception-Resnet-V2, RoBERTa Model,

Model	VSD	Detail	Photo	Installation Diagram	Floor Plan	Report	Table
Base Model	82%	64%	100%	73%	73%	91%	27%
Inception-ResNet-V2	55%	82%	100%	73%	82%	82%	64%
TF-IDF Model	27%	91%	100%	55%	91%	55%	27%
Dutch RoBERTa	55%	82%	100%	64%	27%	36%	18%
LayoutLMv2	50%	91%	100%	55%	73%	91%	55%

Table 5.2: Class Accuracies Testing on Dataset 1

Dutch BERT, Dutch RoBERTa Model, TF-IDF Model and finally the LayoutLMv2 model. The best-performing model is the Inception-ResNet-V2 model, closely followed by ResNet50, to which the text-based Dutch BERT, TF-IDF, and Dutch RoBERTa models follow. Interestingly, the model that combines multiple modalities does not perform as well as the single modality models. Since the best models for each modality surpass the base model in performance, we can conclude that each modality is more suitable for classifying the documents than the handcrafted features. More specifically however, Inception-ResNet-V2 and ResNet50 significantly outperform the textual- and hybrid layout-based models, indicating that visual features are most leading in classifying these documents.

These results do not fully align with recent research, where hybrid models that combine multiple modalities generally outperform models based only on visual or textual features [22, 23, 27, 32, 55, 73, 77, 153]. The strong performances of Inception-ResNet-V2 and Dutch BERT models are somewhat consistent with literature, as the state-of-the-art EAML integrates an Inception-ResNet-V2, and a BERT model via a self-attention fusion [32]. The EAML paper demonstrates that, even though the feature fusion performs best, the image and text models do not differ significantly in performance from the fused model.

Comparing the models in terms of their predictions on the standard test set (see Table 5.2), we observe that class *Photo* is generally classified well by each of the five models. For this detailed misclassification analysis, we selected the best-performing model from each category: one convolutional neural network (CNN) model (Inception-ResNet-V2), one tokenizer-based model (Dutch RoBERTa), the TF-IDF model, and the LayoutLMv2 model. The ResNet50, BERT Dutch, and RoBERTa models were not included in this specific analysis. Dutch RoBERTa is selected for this analysis instead of Dutch BERT as it shows better performance in terms of generalizability (see next section). There's not a second class directly following in terms of correct classifications. We do observe that classes *Cross-Section Drawing*, *Detail*, and *Floor Plan* generally misclassify as each other, showing a similarity in each of the modalities. Comparably, but less obvious, especially for the textual modality, class *Table* generally misclassifies as *Report* exhibiting a possible relatedness as well. For further details, the specific confusion matrices have been included in the appendix (see Appendix G.1).

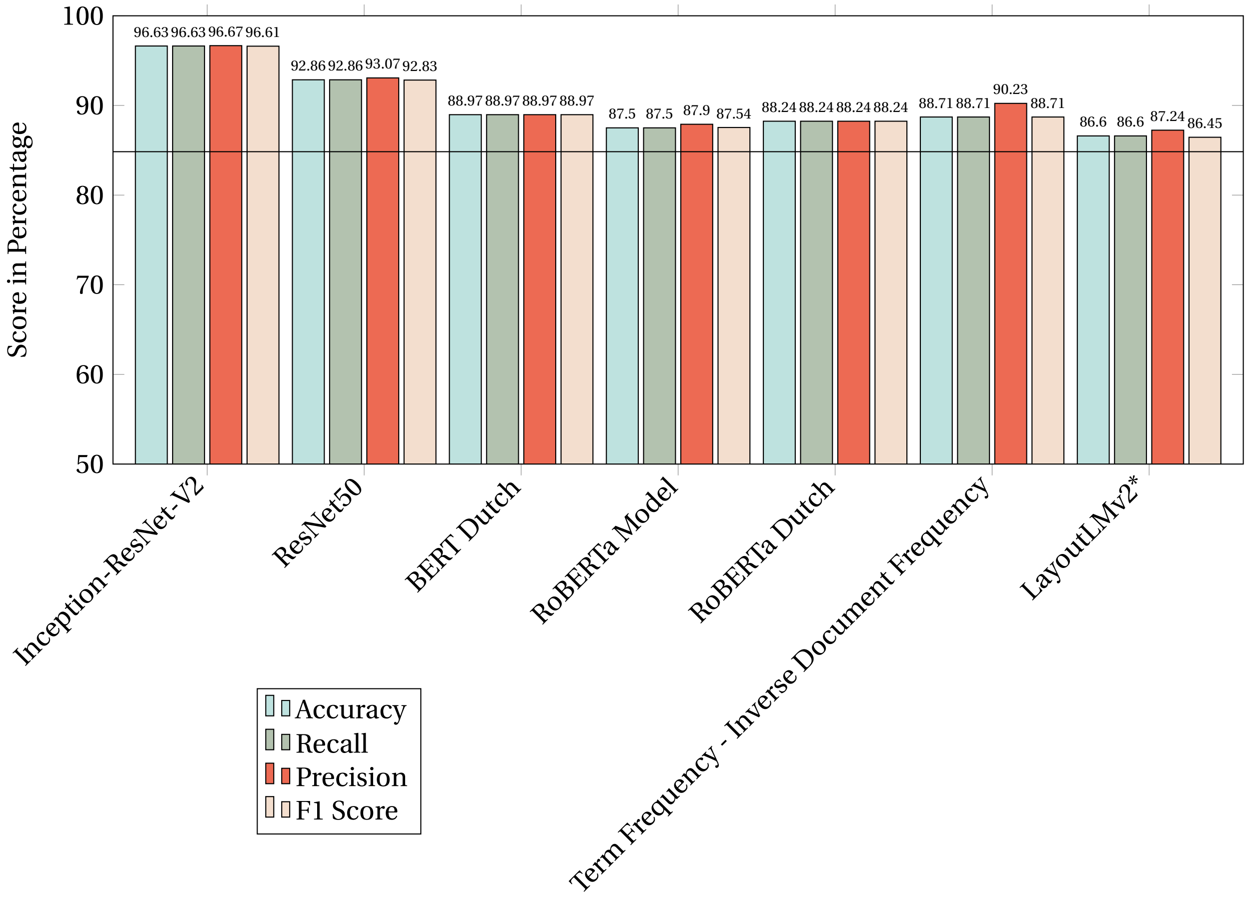


Figure 5.7: Best Performing Models. Models marked with * indicate those that use multi-modal features
Black line represents the Base Model (Accuracy Recall = 84.74, Precision = 84.65, F1 Score = 84.23)

DIRECT INFERENCE ON DATASET 2

To further analyze the performance of these classification models, we perform direct inference on our second dataset, measuring the generalizability of the models on new datasets. We first train the models on the whole dataset, selecting a balanced number of samples from each class by using random undersampling to make sure the model does not become biased to more prevalent classes, and using 10% of the dataset as the validation set, using the same training settings as constructed for the earlier training processes. The test set (as described in Section 4.1.1) is used entirely and the test results are illustrated in Figure 5.8.

Testing the best-performing models on dataset 2 generally results in a significant decrease in classification performance. Interestingly, the precision of the classification stands out above the other metrics. This is likely due to the imbalance in the test set, as the *Report* class is prevalent and the models are generally able to identify it (see Tables 5.2 and 5.3). For this text, we are using macro-averaged metrics rather than weighted-averaged metrics, meaning that if the metrics were weighted-averaged instead of macro-averaged, the metrics would have mostly been reflecting the largest classes [136]. The *Photo* class is relatively well identified in dataset 2 by each of the models, although there is still room for improvement. For the other classes, the

Model	VSD	Detail	Photo	Installation Diagram	Floor Plan	Report	Table
Inception-ResNet-V2	47%	15%	67%	71%	42%	74%	35%
TF-IDF Model	49%	5%	56%	24%	3%	70%	43%
Dutch RoBERTa	1%	0%	65%	95%	52%	75%	35%
LayoutLMv2	49%	5%	56%	24%	3%	70%	43%

Table 5.3: Class Accuracies Testing on Dataset 2

class-accuracies vary, but the *Detail* class is hardly recognized by all models.

Specifying the classification down to the specific classes models predict correctly, we observe that Inception-ResNet-V2 and LayoutLMv2 perform relatively well for classes *Photo*, *Installation Diagram*, and *Report* (as illustrated by Figures G.2 & G.5 in the appendix). The TF-IDF model performs best in predicting the *Photo* and *Report* classes, while the RoBERTa model performs best in classifying the *Photo* and *Table* classes.

The image-based models demonstrate the highest classification performance in this new dataset, indicating a higher ability to generalize. However, since each of the classification performances of the models is 20 - 30% lower than for *Dataset 1*, this indicates that the models have picked up specific characteristics of *Dataset 1*, which might not be apparent in *Dataset 2*. This shows that training the models on additional training data is required to make the model more generalizable to new cases.

The confusion matrices for this section have been included in the appendix (see Appendix G.1). From these tests, we can conclude that the models as trained on *Dataset 1* are not directly generalizable to *Dataset 2*, as they classify between ∼ 40% and ∼ 60% of the samples correctly. However, the most generalizable models of the five are the Inception-ResNet-V2, ResNet50, the RoBERTa Dutch model, the TF-IDF model, and the LayoutLMv2 model, with classification accuracy, recall, precision, and F1-scores varying between 51% and 70%.

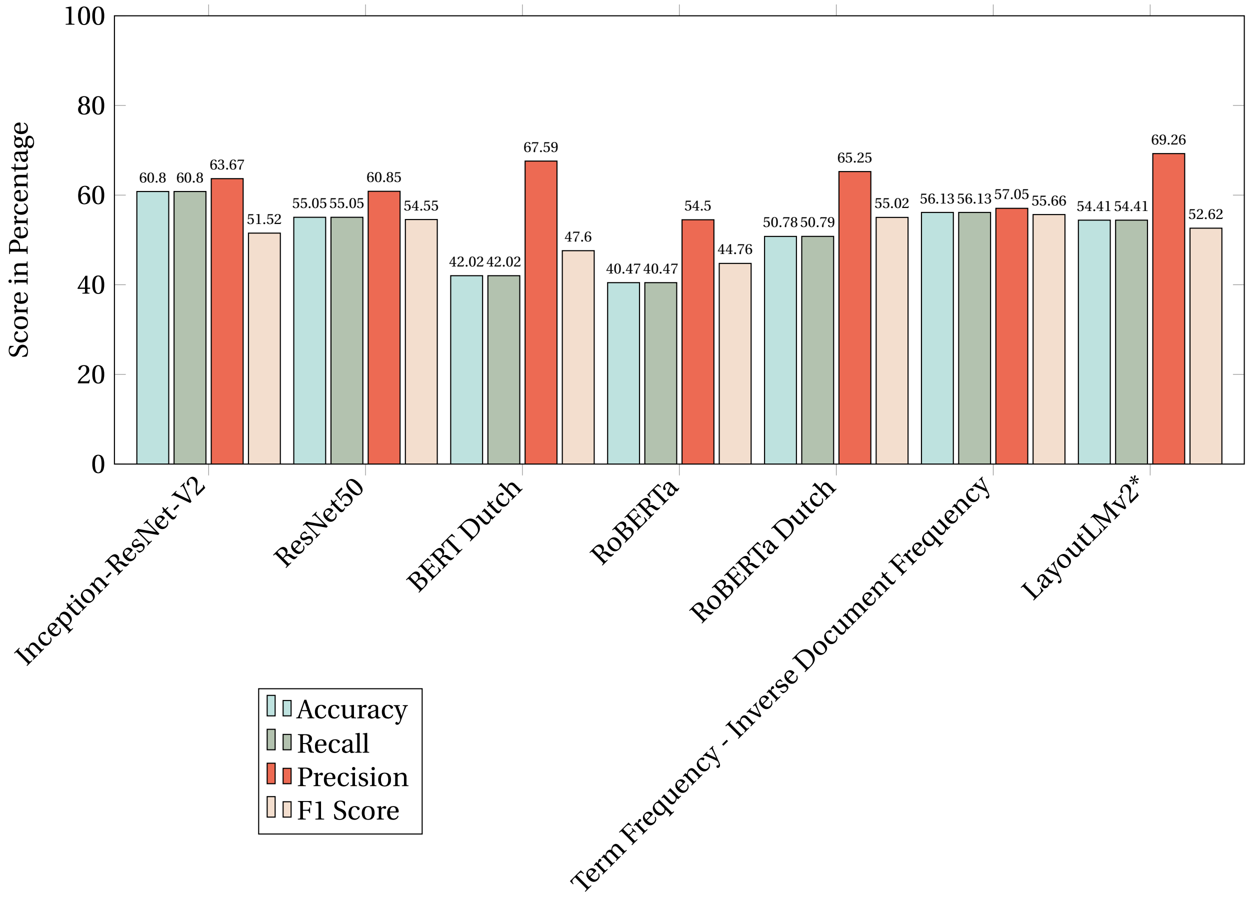


Figure 5.8: Direct Inference on Dataset 2

5.1.6. FEATURE FUSION

In the previous sections, we examined the performance of individual models. From the best-performing models, the Inception-Resnet-V2, ResNet50, TF-IDF model, RoBERTa, and the Dutch versions of the BERT and RoBERTa models utilize a single modality, while the base model and LayoutLMv2 already incorporate multi-modal features. The Inception-ResNet-V2, TF-IDF model and Dutch RoBERTa models show the best performance on *Dataset 1*, as well as best generalizability when testing on *Dataset 2*. Therefore, we evaluate the effect of fusing these models into multi-modal fusion models.

To do this evaluation, we use three different fusion methods: weighted ensembling, simple concatenation, and self-attention-based fusion. These methods are derived from previous research [26, 28, 29, 31, 32, 35]. More details about multi-modal model architectures can be found in the literature review (see Chapter 2) or in the categorization of feature fusions in literature table in the appendix (see Appendix C.2). The specific fusion methods as well as the full architectures that we use for this evaluation are described in Section 4.1.7.

For each of the fusion methods, we make the combination between the image mode and the text mode, as these are the modalities represented by the three selected single-modality models. The two combinations of multi-modal features that we fuse in the different fusion models are (1) Inception-ResNet-V2 combined fused with TF-IDF and (2) Inception-ResNet-V2 fused with the Dutch RoBERTa model. We use the non-augmented Inception-ResNet for the fusion models, as training the model takes less computational time and resources to purely evaluate the fusion results.

As described, we follow the model requirements that we have outlined in Section 4.1.2, specifying that the model should be trainable on an i7 CPU or a single NVIDIA Tesla V100 GPU. During model training, we discover that the self-attention model using the RoBERTa model for the text modality is too computationally demanding for our available resources. The self-attention fusion model should provide us with insights into the performance of a more hybrid fusion method compared to the late fusion methods. Consequently, the results obtained from the less computationally intensive TF-IDF fusion model are still deemed relevant and, therefore, included in this section.

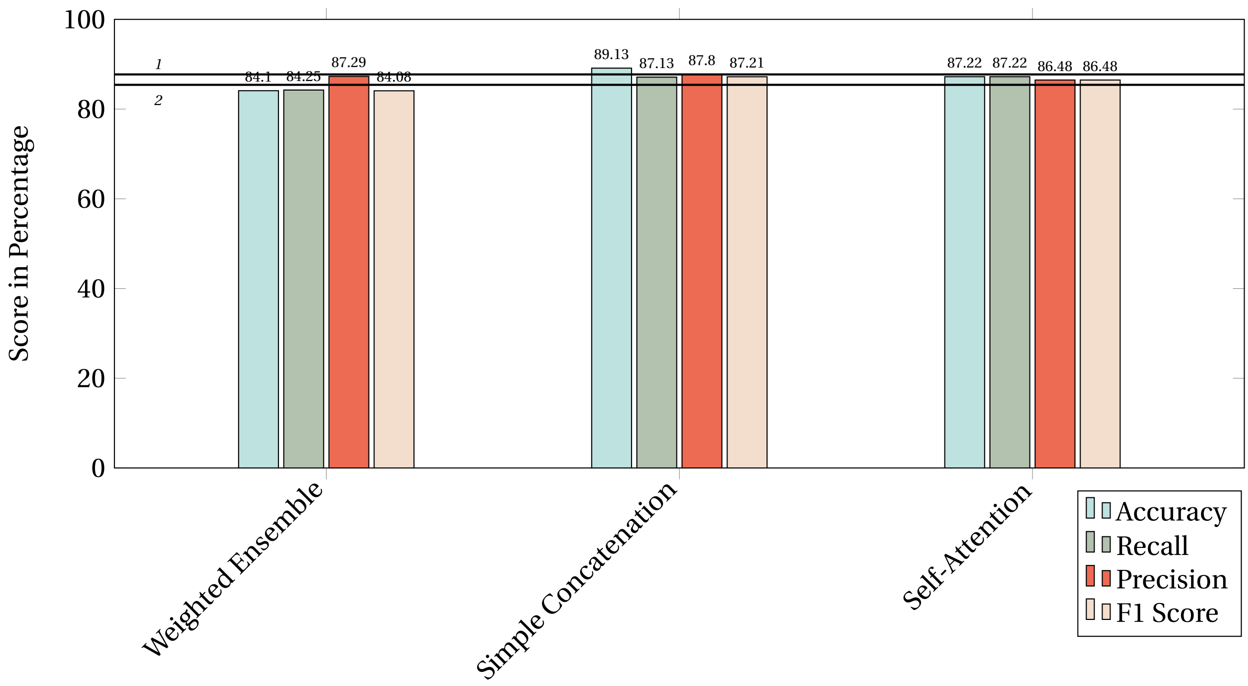


Figure 5.9: Fusion TF-IDF
1: TF-IDF Model (Accuracy & Recall = 87.73)
2: Inception-ResNet-V2 (Without Augmentation) (Accuracy & Recall = 85.41)

Figure 5.9 shows the classification performance of the fusion models that combine the TF-IDF textual model and Inception-ResNet-v2 for the visual features. The black lines represent the classification performance of the Inception-ResNet-V2 model and the TF-IDF model that were trained in the same settings (see Figures 5.2 and 5.8). However, only the simple concatenation model slightly outperforms the TF-IDF model. The Inception-ResNet-V2 model is surpassed by the self-attention fusion model as well, showing added value in combining the textual and visual characteristics of the data.

Figure 5.10 demonstrates the classification results for the weighted ensemble and simple concatenation fusion models that fuse the Dutch RoBERTa model and Inception-ResNet-V2. We observe that Inception-ResNet-V2 benefits significantly from feature fusion in the weighted ensemble model, which only slightly outperforms the Dutch RoBERTa model. The simple concatenation model performs comparably with the Inception-ResNet-V2 model without any feature fusion.

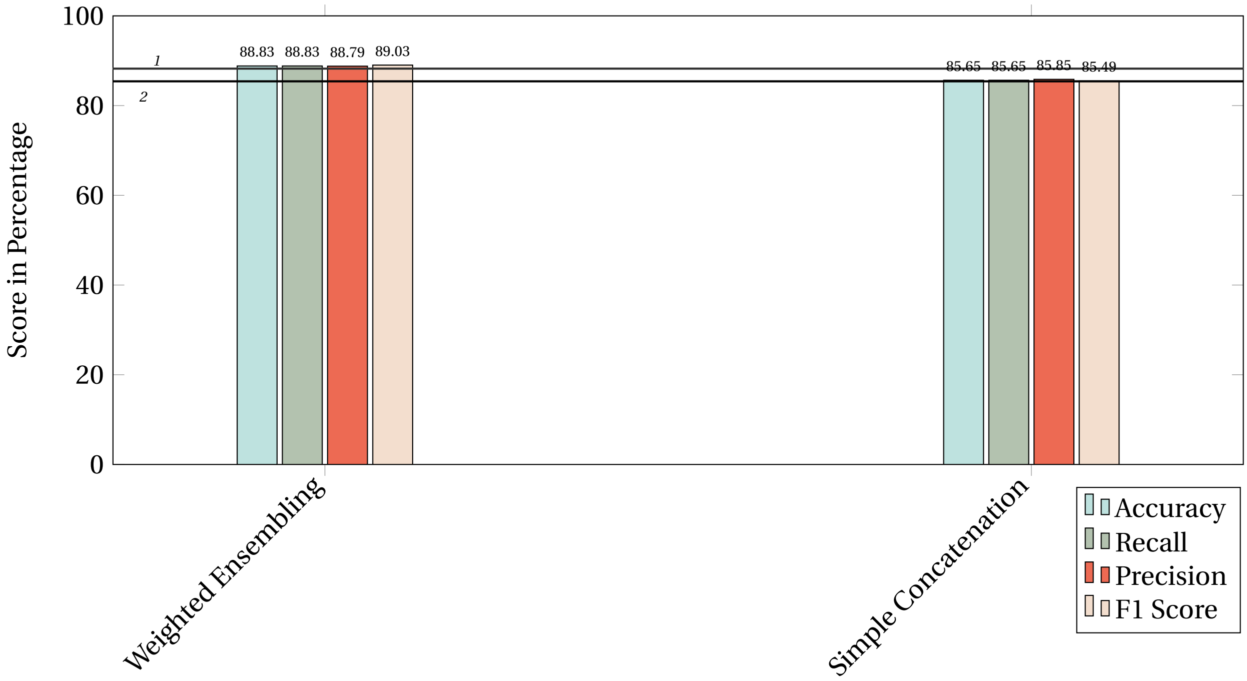


Figure 5.10: Fusion Dutch RoBERTa
1: Dutch RoBERTa (Accuracy & Recall = 88.24)
2: Inception-ResNet-V2 (Without Augmentation) (Accuracy & Recall = 85.41)

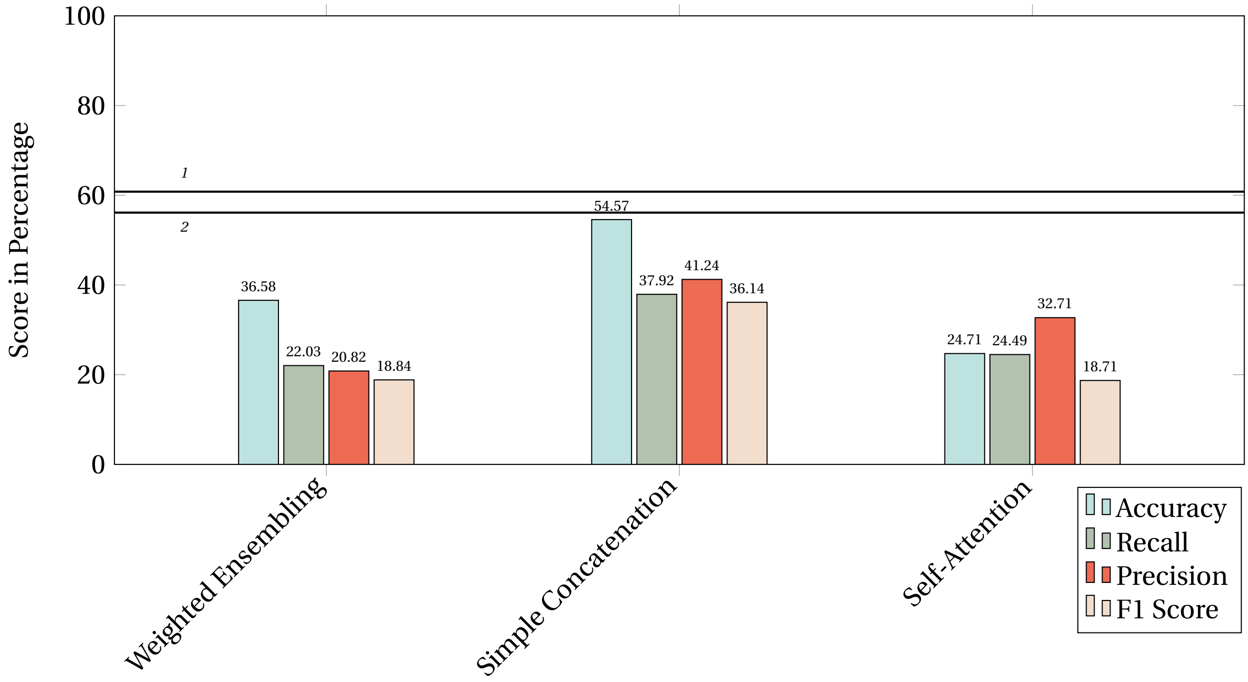


Figure 5.11: Fusion TF-IDF on Dataset 2
1: Inception-ResNet-V2 (Without Augmentation) (Accuracy & Recall = 60.8)
2: TF-IDF Model (Accuracy & Recall = 56.13)

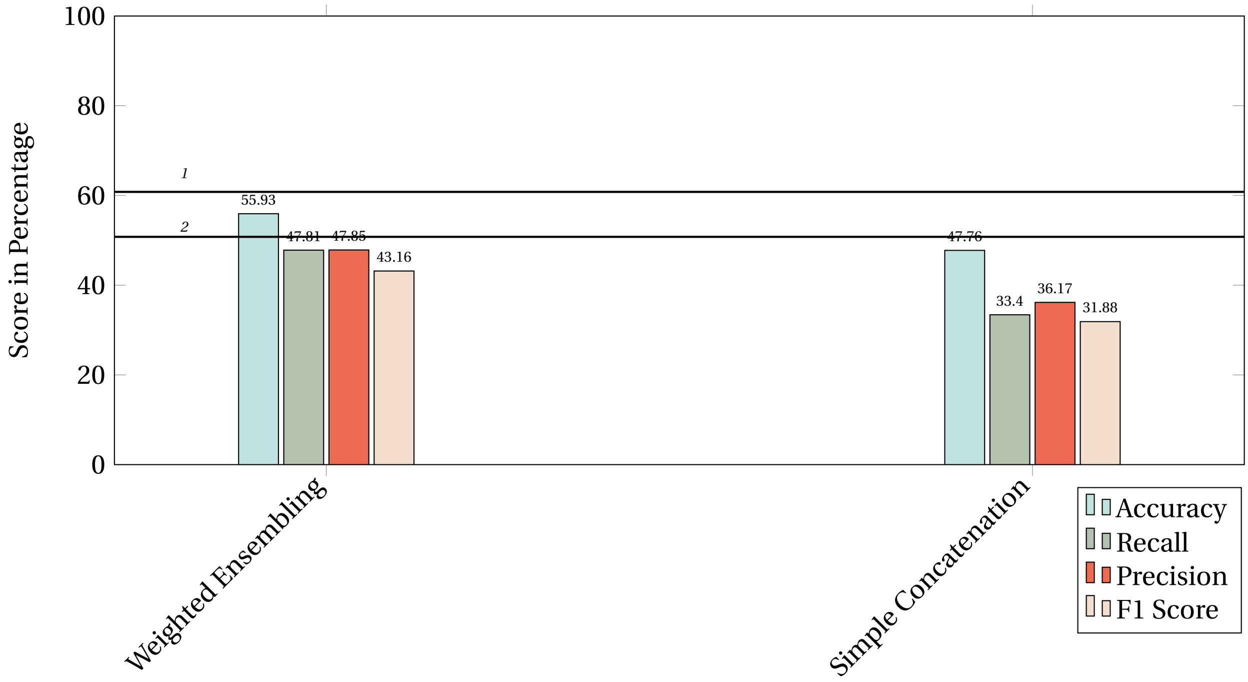


Figure 5.12: Fusion Dutch RoBERTa on Dataset 2
1: Inception-ResNet-V2 (Without Augmentation) (Accuracy & Recall = 60.8)
2: Dutch RoBERTa Model (Accuracy & Recall = 50.78)

Figures 5.11 and 5.12 illustrate the performance of the fusion models directly inferencing on

Dataset 2. None of the TF-IDF-based fusion models surpass the fused models in terms of classification performance on *Dataset 2*. The weighted ensemble fusion model does outperform the original Dutch RoBERTa model on *Dataset 2*, despite only slightly outperforming it on *Dataset 1*. Overall, we observe that the Dutch RoBERTa-based fusion models perform significantly better on *Dataset 2* than the TF-IDF-based fusion models, demonstrating better generalization capabilities.

In conclusion, we observe slight improvements in performance on *Dataset 1* using fusion models. The weighted ensemble fusion using the Dutch RoBERTa model is the only model that outperforms its single-mode models, although not significantly. The evaluation of the fusion models on *Dataset 2* demonstrates that the fusion models are evidently less generalizable than their single-mode models. In terms of generalizability, the weighted ensemble model using Dutch RoBERTa performs best among all fusion models as well. Therefore, we conclude that the weighted ensemble Dutch RoBERTa-based fusion models appear to have the greatest potential to benefit from fusion, both in terms of performance on the same dataset and in their ability to classify unseen datasets.

The fact that the fusion models do not significantly outperform the single-modality models aligns with previous research on the state-of-the-art document image classification models such as EAML [32], and other multi-modal classification networks [27, 57]. These works demonstrate that multi-modal document classification does have the potential to outperform single-mode models, even though their multi-modal models do not significantly outperform the used single-mode models. Furthermore, our findings do not align with the literature, suggesting that hybrid fusions, such as self-attention, achieve the best classification results [32]. when the right models are fused. In contrast, we do not find significantly better results for the self-attention fusion model.

5.2. FINAL MODEL DEPLOYMENT

Finally, we observe that the single-mode Inception-ResNet-V2 model achieves the best classification results when trained on the augmented dataset. We achieve a classification accuracy and recall of 96.63%, a precision of 96.67% and an F1 score of 96.61% through k-fold testing on *Dataset 1*. In terms of generalizability, the model performs best, as well as achieving a classification and recall of 60.8%, a precision of 63.67% and an F1 score of 51.52%. Although the Inception-ResNet-V2 model demonstrates the best classification performance on *Dataset 2*, further improvements are necessary before it can be used to classify more unseen datasets.

These found results are similar to those of state-of-the-art models trained and tested on RVL-CDIP and Tobacco-3482 [26–29, 88]. As these datasets are much larger, and contain different kinds of document, we cannot draw a direct connection to those results. The related research only tested performance on a holdout part of the same dataset the model was trained on, therefore not exactly testing the generalizability of the model on a new dataset through direct inference, however previous research did examine the result of pretraining on a very related dataset

[29, 31].

To enhance the model's performance, the pretrained Inception-ResNet-V2 should be fine-tuned on *Dataset 2*, or find other related datasets to further train the model on. Additional steps that could improve model performance and generalizability without a third set of documents include applying further data augmentation techniques, such as scaling the document image, flipping it horizontally or vertically, applying shearing techniques, or converting images to grayscale. Further pretraining the model on publicly available datasets such as RVL-CDIP, or available floor plan datasets could also enhance performance. To prevent overfitting on the training data, regularization techniques can be employed to penalize the model for learning to classify based on noise rather than relevant features. Lastly, deploying explainable AI techniques can help better understand what the model focuses on in images to make classifications [81, 86, 87]. As we did see slight improvement through fusion of modalities, applying data augmentation to the fusion models could further improve overall performance as well.

In this specific case, the classification made is to be deployed for two main tasks; organizing and adding metadata about the documents. For the final deployment of the model, a tool is constructed that carries out the full pipeline, from input folder with files to be classified, to an organized folder into the classified categories. Additionally, the classes are saved and formulated in the required form as metadata.

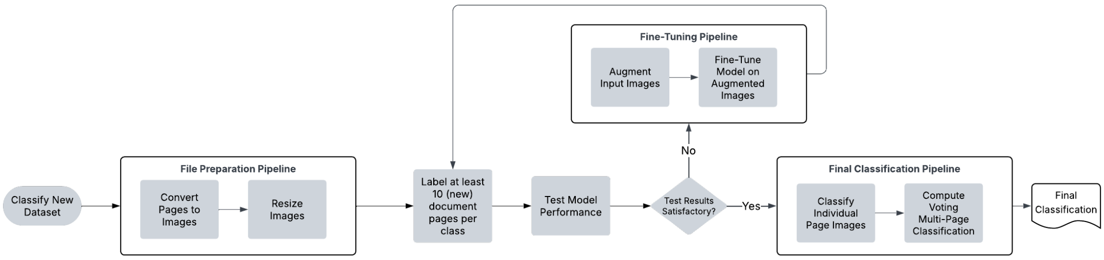

Fine-Tuning Pipeline
Augment Input Images
Fine-Tune Model on Augmented Images
No
Classify New Dataset
File Preparation Pipeline
Convert Pages to Images
Resize Images
Label at least 10 (new) document pages per class
Test Model Performance
Test Results Satisfactory?
Yes
Final Classification Pipeline
Classify Individual Page Images
Compute Voting Multi-Page Classification
Final Classification

Figure 5.13: Deployment Process Ensuring Quality Classification

Without applying any of the improvement tasks before deploying the model, we suggest the process as demonstrated in Figure 5.13 to ensure quality outcomes for direct inference. Each of the pipelines are actions that are automatically carried out. When an input folder is given, its documents are converted to images, and these images are resized to (299,299,3) as this is the optimal size for the Inception-ResNet-V2 model [120]. To be able to test the model performance on a new dataset, we need to have a set of labeled documents. A minimum of 10 document pages per class is required; however, a larger number of test files will give a greater confidence in the test results. The model should be tested on this labeled set, and if the test results are satisfactory, the model is to be used for further inference. If not, however, the labeled data is used as fine-tuning data; it is augmented, and the augmented set of images is used to further train the model. New documents should be labeled to evaluate the model performance and this process should be repeated until performance is satisfactory. The final classification is made by classifying the individual page images. For a multi-page document, the full document classification is made by applying a voting mechanism, where the class most occurring in the individual page classification is chosen as the final class.

6
CONCLUSION & DISCUSSION

6.1. REVISITING THE RESEARCH GAP

In the literature review, we evaluated document classification techniques applied specifically in the fields of the AEC industry, as well as the more general document classification methods utilized in the last five years. We found that the application of document classification to cases in the AEC industry is limited. Research on classification in the AEC sector is mainly focused on text, and mostly applied to very different problems. Classification based on visual features, layout features, and their combinations have not yet been applied in research, although documents in the AEC sector are generally not very text-dense. More studies on the use of document image classification and multimodal document classification models would be beneficial to the AEC industry to further support Construction 4.0 practices. The further development and exploration of intelligent document processing techniques, such as document classification for documents within the AEC industry, helps unlock information from documents and promotes data-driven practices.

In research towards more general document classification, we observed significant developments over the years. Initially, document classification was mostly based on manually-extracted and single-mode-based features. Over time, features were combined through hybrid modality models, and performance improved through the use of deep learning models. Two main datasets are generally used in document classification research, and the classification performance on these datasets has been optimized over the years. The application of pretraining and transfer learning to document classification models has shown to be highly beneficial for classification performance. Recently, the focus of research towards document image classification has shifted to other goals or topics surrounding the classification models, such as explainable AI in document classification and improving the efficiency of training document classification models.

Even with these improvements, there is a significant gap in the application of multi-modal document classification models and state-of-the-art document classification techniques within the AEC sector. This gap shows the need for research focused on identifying the best-performing classification model architecture for asset management documents in the AEC sector.

This study aims to address this gap by developing a generally applicable document classification model that is able to generalize to new datasets. To make sure this model is generalizable, we do not only test its ability to predict its own instances, but also measure its performance on an external dataset. This way, we evaluate the model's effectiveness in real-world scenarios and its potential for broader application.

6.2. RESEARCH OUTCOMES

6.2.1. RESEARCH QUESTION 1.1: HOW DO STATE-OF-THE-ART CLASSIFICATION MODELS PERFORM AT CLASSIFICATION OF ASSET RELATED DOCUMENTS?

BASE MODEL

The base model was constructed based on relevant features identified through exploratory data analysis. This multinomial logistic regression model uses the multi-modal handcrafted features from three modalities: visual, textual, and layout. Visual features include number of colors, aspect ratio, brightness, contrast, edge density, dominant color, and image entropy. Textual features include the total and unique word counts, as well as the presence of specific keywords. The layout features consist of the number and average size of bounding boxes, extracted using OCR. This model serves as a baseline, comparing a handcrafted multi-modal document classification model to other deep learning-based models. This base model achieves an accuracy and recall of 84.74%, with precision and F1-score around 84.6%. These results are comparable to other document classification models using handcrafted features in the literature.

IMAGE MODALITY MODELS

Within this section, we evaluate the performance of five CNN architectures on document image classification; MobileNetV2, ResNet50, VGG16, Inception-ResNet-V2 and EfficientNetB0. Three main research objectives were evaluated; the effect of transfer learning, the impact of data augmentation, and finally the classification performance of the models themselves. To evaluate the effect of transfer learning, the difference between applying and not applying the standard ImageNet weights was compared. Applying these weights generally improved accuracy, recall, precision, and F1 score for the CNN models, with MobileNetV2 benefiting the most (around 55% increase) and Inception-ResNet-V2 the least (around 7% increase). Notably, the performance of EfficientNetB0 did not improve, but decrease with the application of ImageNet weights. These differences in performances can be explained by examining their architecture and complexity. Lighter models with fewer parameters benefit more from transfer learning, where deeper networks can learn more complex features from scratch. We perform data augmentation by adding rotated images, which quadruplicates the data size, and improves the performance of the more complex models (Inception-ResNet-V2, ResNet50 and MobileNetV2). The lighter models (EfficientNetB0, VGG16) did not benefit from the augmentation, possibly because they observe the augmented data as noise. Lastly, comparing the performance of the five individual models, Inception-ResNet-V2 achieves the best classification performance, achieving an accuracy and recall of 85.41%, a precision of 87.65%, and an F1-score of 85.6%, which improves approximately 10% when applying augmentation.

TEXTUAL MODALITY MODELS

After extracting the text using OCR, we cleaned the text to evaluate the result of two different cleaning steps; first, filtering on Dutch words, and second, filtering out stopwords, resulting in *Textset 1* and *Textset 2*, respectively. The models tested include traditional TF-IDF features and tokenizer-based transformers. The TF-IDF features were classified using a simple CNN model, slightly outperforming the base model. However, no significant difference in performance was observed between the two text sets. The tokenizer-based models perform similarly to TF-IDF on *Textset 1*, but generally decreased in performance when using *Textset 2*. For the tokenizer-based models, we specifically evaluated BERT, RoBERTa, and the Dutch versions of these two models. The text was tokenized and classified using the respective tokenizers and classification models. The Dutch versions of BERT and RoBERTa (BERTje and RobBERT, respectively) achieved the best performance, closely followed by RoBERTa and BERT. This aligns with expectations, as Dutch pretrained models are expected to better capture the Dutch text in the document dataset, which is in line with prior research as well.

LAYOUT MODALITY MODELS

This section evaluated five layout-based classification models: LayoutLM, LayoutLMVv2, LayoutLMv3, LiLT and UDOP. LayoutLMv2 achieves the best performance across all metrics, outperforming its successor, LayoutLMv3. The original LayoutLM performs the worst, highlighting the added value of incorporating visual features in addition to textual-spatial information. The better performance of LayoutLMv2 may be because of architectural differences, pretraining objectives, or tokenization techniques.

The LiLT and UDOP models perform better than the original LayoutLM model but do not outperform LayoutLMv2. LiLT, like LayoutLM, only uses textual and layout information, yet still achieves a competitive classification performance compared to the models that combine all three modalities.

In recent literature, LayoutLMv3 and UDOP generally achieve the best classification performances, which does not align with the results of this study. This difference may be because of differences in datasets, as the RVL-CDIP dataset might be better suited to these models, or the possibility of not tuning the hyperparameters in the suiting way for these models.

GENERALIZABILITY OF BEST PERFORMING MODELS

Over all tested models, Inception-ResNet-V2 achieves the best classification performance, closely followed by ResNet50. Textual models such as Dutch BERT, Dutch RoBERTa, and the TF-IDF model perform well but achieve approximately 10% lower classification metrics compared to the image-based models. The multi-modal LayoutLMv2 model outperforms the base model, but does not compete in performance with the single-modality models. Since visual models perform significantly better than textual or hybrid models, image-based information seems to be most important for the classification of this dataset.

When evaluating these best-performing models on the second dataset through direct inference to evaluate their generalizability, each model shows a significant decrease in performance (decrease of 20% - 30%). The image-based models demonstrate the best performance, closely followed by the TF-IDF model. The large decrease in performance for all models suggests that they primarily learned dataset-specific features rather than more generally applicable features. This indicates that models should be trained on more data to improve performance. Further data collection and fine-tuning of the single-modality models is therefore needed to develop an asset management document classification model capable of correctly classifying unseen data in new datasets.

6.2.2. RESEARCH QUESTION 1.2: HOW DO COMBINATIONS OF DOCUMENT MODALITIES IMPACT THE PERFORMANCE OF A CLASSIFICATION IN TERMS OF ACCURACY?

This study combines the best performing single-modality models found with the first research question. Three types of fusions are used: weighted ensembling and simple concatenation, and self-attention-based fusion. The best performing visual-based Inception-ResNet-V2 is fused with the best performing text-based TF-IDF and Dutch RoBERTa models.

In the weighted ensemble fusion model, each individual model is trained during the training process and delivers a final classification, which is then merged using trainable weights. This fusion method achieves the best performance and generalizability, specifically combining Inception-ResNet-V2 with Dutch RoBERTa. Generally, the fusion models only showed very little to no improvement over the single-modality models, for both *Dataset 1* and *Dataset 2*. These findings align with some existing work, where multi-modal models do not always significantly outperform single-modality models. However, in contrast to the literature, the weighted ensemble outperforms the more complex self-attention-based fusion in this study.

6.3. LIMITATIONS & FUTURE RESEARCH

Several limitations were encountered in this research that provide an important context for interpreting the results and additionally help indicate interesting directions for future work. We categorize the limitations into two main groups; data-related limitations and model-related limitations.

6.3.1. LIMITATIONS

First, the literature review for this research was conducted between October 2024 and January 2025, and the remainder of the research was carried out between January 2025 and June 2025. Given the fast-evolving field of data-driven solutions and artificial intelligence, new relevant research may have emerged after January, as well as after the finalization of this thesis.

We first describe the data-related limitations, of which some have a direct effect on the generalizability of the model. One of the main limitations of this study is the use of small datasets. Each class contained only 40-50 document pages per class, which restricted the model's ability to generalize. Both of these datasets were private and may not be representative of the wider AEC industry or asset management documentation as a whole. As such, each time the model is applied to a new dataset, its performance and generalizability should be re-evaluated. In a related sense, this study uses Dutch datasets. Although tokenizer-based Dutch models outperformed others in this context, these results cannot be directly generalized to other languages or domains, as model performance may vary significantly for other languages or domains. Given these constraints, we cannot make industry-wide claims based on the results we obtained. In addition, class imbalance posed challenges. As we did foresee biases in classification training on an imbalanced dataset, we applied a simple method to equalize the number of instances used per class. Throughout the study, we discovered that the method used, namely random undersampling, has other implications as well, such as the loss of information by excluding data samples and that possibly other balancing techniques would have yielded better performance.

The model-related limitations related to the model include computational constraints. Although an improved CPU and GPU were provided, more complex deep learning models (e.g., EfficientNetB2-B7, NasNet, or advanced fusion architectures) could not be tested. Secondly, the models we tested were selected based on whether they could be trained on our data, and past performance on standard datasets (e.g., RVL-CDIP, Tobacco-3482), which are structurally different from asset management documents. This means that conclusions about the effectiveness of specific models are not necessarily transferable between cases.

6.3.2. FUTURE RESEARCH

While overcoming the limitations identified in this study is a natural next step, we also find four key directions for future research that emerge from our findings. These directions aim to improve the robustness of the model, improve applicability of the model to the real-world, and to apply intelligent document processing (IDP) techniques within the AEC sector.

Future research should focus on creating and publicly publishing AEC-related document datasets to promote reproducibility and progress in the field. A larger variety of datasets from different companies in different languages and document types to train the classification models on would improve model generalizability and help establish more robust and transferable solutions. More importantly, such datasets would open the door to a broader range of research into intelligent document processing (IDP) techniques within the AEC sector. Such research could improve efficiency by automating and accelerating currently manual, document-heavy tasks.

Moreover the models evaluated in this research; future work could investigate more advanced fusion architectures, making different combinations in CNN-, transformer-, and non-deep learning-based models. With the development of new models, new advantages could arise for document classification as well.

Additionally, feature analysis for both deep learning models and the hand-crafted features could help identify the most informative document attributes, improving both accuracy and explainability. In addition, investigating various other document interpretation tasks such as symbol recognition, named entity recognition, and object detection, would further broaden the application of intelligent document processing (IDP) in the AEC sector.

6.4. RECOMMENDATIONS FOR MOVARES

As in the current state of model capabilities, the model is not generalizable enough to correctly classify new datasets, we recommend further training the final model on new datasets from other companies, increasing the generalizability of the model. Each time the model is used for a new case, a range of 10 - 50 documents should be labeled, and the performance of the model should be tested. If the performance metrics retrieved through this test are not satisfactory, the labeled data should be used to further fine-tune the model to the new data. As the model has been pretrained on the initial dataset(s), it will learn the new data much faster.

Incorporating publicly available datasets into the training set can enhance generalizability. Currently, only a few available floor plan datasets are directly applicable to this classification. Pretraining the dataset on document datasets with different classes than those used in this research (e.g., RVL-CDIP and Tobacco-3482) is likely to improve the classification results, as demonstrated by the improved classification results when applying ImageNet weights. However, since these publicly available datasets are generally in English, the primary benefit is expected to be in visual-based classification. There may be benefits in text-based models if the texts are translated before inclusion.

As demonstrated by the EDA and final classification, the intra-class compactness and inter-class separability are not optimal for each of the classes. The *Photo* and *Report* are generally separable from the other classes, however, the drawing-based classes (*View-Segment Diagram*, *Detail*, *Floor Plan*, and *Installation Diagram*) are more intertwined. The *Table* class is very closely related to the *Report* class. As the drawing-based documents often contain words related to the specific category (e.g., a *Detail* drawing would contain the words "Detail" or a specific scale that can easily be searched for), it would be interesting to evaluate the results of a model classifying documents into just the three classes Photo, Report, or Drawing. In this sense, documents cannot be classified into the specific drawing-base classes, but documents can contain e.g. a *View-Segment Diagram*, *Detail*, *Floor Plan*, or *Installation Diagram*. Similarly, an effective way of identifying *Tables* in reports would allow to extract tables in the same format.

Additionally, since two utilized datasets were initially labeled into two different, yet relevant sets of classes, it is important to design a standard for asset management document classes that optimally reflects the possible classes of documents. Furthermore, as this study classifies single document pages, a strategy should be designed for how to determine the final class for these longer documents.

Further specifying the complete IDP pipeline could help better determine the value of the classification made by the models in this study. This includes detailing the pipeline down to all required information and identifying which IDP tasks could be used to retrieve the information. By doing so, it will be clearer to what end the classification model will serve and how it can be utilized optimally.

6.5. CONTRIBUTIONS TO RESEARCH & PRACTICE

The contributions of this study can be summarized into three main areas:

- The application of intelligent document processing (IDP) in the architecture, engineering, and construction (AEC) industry, and an initial step towards automated document classification for asset management within the AEC sector
- A comprehensive evaluation of multi-modal classification architectures on a real-world dataset, and
- An initial step towards automated document classification for asset management within the AEC sector

First, although the AEC sector is increasingly taking on construction 4.0, derived from Industry 4.0, and applying more and more data-driven approaches, the use of document classification applications is still limited in this sector. In our literature review, we identified a gap in the application of document classification techniques, specifically in the use of image-based features in AEC document classification. By evaluating document classifications on an AEC document dataset, which includes floor plans, drawings, and installation schemes, this work broadens the use of document classification within the AEC sector.

Secondly, apart from contributing to the AEC sector specifically, this study contributes to the broader academic fields of document classification and multi-modal classification. This study provides a detailed comparison of different modality-specific models, using visual, textual, and layout-based approaches, on a real-world dataset. We demonstrate what impact the decision of modality has on the classification, and broaden the standard features used in document classification. Furthermore, we evaluate the behavior of state-of-the-art pretrained models outside of their original domains, and how transfer learning may or may not generalize across domains outside of standard benchmarking datasets. Finally, in evaluating Dutch-language NLP models, we confirm the added value of language-specific pretraining as Dutch-language NLP models outperform their English counterparts in this research.

Lastly, as limited research has been conducted on the use of intelligent document processing in asset management within the AEC sector, this research represents an initial step in that direction. We explore the possibilities and challenges of processing asset-related document types using state-of-the-art models and discover new possible directions for beneficial future research in this area.

REFERENCES

[1] L. Liu, Z. Wang, T. Qiu, Q. Chen, Y. Lu, C. Y. Suen. Document image classification: Progress over two decades. Neurocomputing 453 (2021) 223–240. URL: https://www.sciencedirect.com/science/article/pii/S0925231221006925. doi:https://doi.org/10.1016/j.neucom.2021.04.114.

[2] A. W. Harley, A. Ufkes, K. G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval, in: Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), ICDAR '15, IEEE Computer Society, USA, 2015, p. 991–995. URL: https://doi.org/10.1109/ICDAR.2015.7333910. doi:10.1109/ICDAR.2015.7333910.

[3] T. Bisen, D. M. Javed, S. Kirtania, P. Nagabhushan. Dwt-compcnn: Deep image classification network for high throughput jpeg 2000 compressed documents. Pattern Analysis and Applications 26 (2023) 1–15. doi:10.1007/s10044-023-01190-8.

[4] S. Studer, B. Bui, C. Drescher, A. Hanuschkin, L. Winkler, S. Peters, K.-R. Müller. Towards crisp-ml(q): A machine learning process model with quality assurance methodology. Machine Learning and Knowledge Extraction 3 (2021) 392–413. doi:10.3390/make3020020.

[5] X. Li, J. Zheng, M. Li, W. Ma, Y. Hu. One-shot neural architecture search for fault diagnosis using vibration signals. Expert Systems with Applications 190 (2022) 116027. URL: https://www.sciencedirect.com/science/article/pii/S0957417421013737. doi:https://doi.org/10.1016/j.eswa.2021.116027.

[6] A. Darko, A. P. Chan, M. A. Adabre, D. J. Edwards, M. R. Hosseini, E. E. Ameyaw. Artificial intelligence in the aec industry: Scientometric analysis and visualization of research activities. Automation in Construction 112 (2020) 103081. URL: https://www.sciencedirect.com/science/article/pii/S092658051930651X. doi:https://doi.org/10.1016/j.autcon.2020.103081.

[7] A. Sawhney, M. Riley, J. Irizarry, Construction 4.0: An Innovation Platform for the Built Environment, Routledge, 2020. doi:10.1201/9780429398100.

[8] D. Maleti, M. Grabowska, M. Maleti. Drivers and barriers of digital transformation in asset management. Management and Production Engineering Review (2023). URL: https://api.semanticscholar.org/CorpusID:258871377.

[9] J. Vieira, N. M. d. Almeida, J. Poças Martins, H. Patrício, J. G. Morgado. Analysing the value of digital twinning opportunities in infrastructure asset management. Infrastructures 9 (2024). URL: https://www.mdpi.com/2412-3811/9/9/158. doi:10.3390/infrastructures9090158.

[10] A. Crespo Marquez, J. F. Gomez Fernandez, P. Martínez-Galán Fernández, A. J. Guillén Lopez. Maintenance management through intelligent asset management platforms (iamp). emerging factors, key impact areas and data models. Energies 13 (2020) 3762. doi:10.3390/en13153762.

[11] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, G. P. Li. Predictive maintenance in the industry 4.0: A systematic literature review. Computers Industrial Engineering 150 (2020) 106889. URL: https://www.sciencedirect.com/science/article/pii/S0360835220305787. doi:https://doi.org/10.1016/j.cie.2020.106889.

[12] H. Marques, A. Giacotto. Prescriptive maintenance: Building alternative plans for smart operations, in: Proceedings of the 10th Aerospace Technology Congress, pp. 231–236. doi:10.3384/ecp19162027.

[13] C. Emmanouilidis, J. P. Liyanage, E. Jantunen. Mobile solutions for engineering asset and maintenance management. Journal of Quality in Maintenance Engineering 15 (2009) 92–105. doi:10.1108/13552510910943903.

[14] M. Trindade, N. Almeida. The impact of digitalisation in asset-intensive organisations. Network Industries Quarterly 20 (2018). URL: https://www.network-industries.org/2018/12/14/the-path-towards-digitalisation-in-road-infrastructure/.

[15] I. Mancuso, A. M. Petruzzelli, U. Panniello, Industry 4.0 for AEC Sector: Impacts on Productivity and Sustainability, Springer International Publishing, Cham, 2024, pp. 33–50. URL: https://doi.org/10.1007/978-3-031-36922-3_3. doi:10.1007/978-3-031-36922-3_3.

[16] C. Wang, J. Plume. A review on document and information management in the construction industry: From paper-based documents to bim-based approach, in: Proceedings of 2012 International Conference on Construction and Real Estate Management, volume 1, Kansas City, USA, pp. 369–373.

[17] A. Martínez-Rojas, J. M. López-Carnicer, J. González-Enríquez, A. Jiménez-Ramírez, J. M. Sánchez-Oliva, Intelligent Document Processing in End-to-End RPA Contexts: A Systematic Literature Review, Springer Nature Singapore, Singapore, 2023, pp. 95–131. URL: https://doi.org/10.1007/978-981-19-8296-5_5. doi:10.1007/978-981-19-8296-5_5.

[18] S. V. Mahadevkar, S. Patil, K. Kotecha, L. W. Soong, T. Choudhury. Exploring ai-driven approaches for unstructured document analysis and future horizons. Journal of Big Data 11 (2024). URL: http://dx.doi.org/10.1186/s40537-024-00948-z. doi:10.1186/s40537-024-00948-z.

[19] N. Kamaleson, D. Chu, F. E. B. Otero. Automatic information extraction from electronic documents using machine learning, in: M. Bramer, R. Ellis (Eds.), Artificial Intelligence XXXVIII, Springer International Publishing, Cham, 2021, pp. 183–194.

[20] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou. Layoutlm: Pre-training of text and layout for document image understanding, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining, KDD ’20, ACM, 2020, p. 1192–1200. URL: http://dx.doi.org/10.1145/3394486.3403172. doi:10.1145/3394486.3403172.

[21] Z. Tang, Z. Yang, G. Wang, Y. Fang, Y. Liu, C. Zhu, M. Zeng, C.-Y. Zhang, M. Bansal. Unifying vision, text, and layout for universal document processing. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 19254–19264. URL: https://api.semanticscholar.org/CorpusID:254275326. doi:10.1109/CVPR52729.2023.01845.

[22] Y. Huang, T. Lv, L. Cui, Y. Lu, F. Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking, in: Proceedings of the 30th ACM International Conference on Multimedia, MM ’22, Association for Computing Machinery, New York, NY, USA, 2022. URL: 10.1145/3503161.3548112. doi:10.1145/3503161.3548112.

[23] J. Wang, L. Jin, K. Ding. LiLT: A simple yet effective language-independent layout transformer for structured document understanding, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 7747–7757. URL: https://aclanthology.org/2022.acl-long.534. doi:10.18653/v1/2022.acl-long.534.

[24] M. Tharani, M. Zaki, C. Finegan-Dollak, A. Verma. Gvdoc - graph-based visual document classification, in: Findings of the Association for Computational Linguistics, pp. 5342–5357. doi:10.18653/v1/2023.findings-acl.329.

[25] P. Li, J. Gu, J. Kuen, V. Morariu, H. Zhao, R. Jain, V. Manjunatha, H. Liu. Selfdoc: Self-supervised document representation learning, in: Conference on Computer Vision and Pattern Recognition, pp. 5648–5656. doi:10.1109/CVPR46437.2021.00560.

[26] S. Bakkali, Z. Ming, M. Coustaty, M. Rusiñol. Cross-modal deep networks for document image classification, in: 2020 IEEE International Conference on Image Processing (ICIP), pp. 2556–2560. doi:10.1109/ICIP40778.2020.9191268.

[27] S. Bakkali, Z. Ming, M. Coustaty, M. Rusinol. Visual and textual deep feature fusion for document image classification, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2394–2403. doi:10.1109/CVPRW50498.2020.00289.

[28] S. P. Zingaro, G. Lisanti, M. Gabbrielli. Multimodal side- tuning for document classification, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, p. 5206–5213. URL: http://dx.doi.org/10.1109/ICPR48806.2021.9413208. doi:10.1109/icpr48806.2021.9413208.

[29] J. Ferrando, J. L. Domínguez, J. Torres, R. García, D. García, D. Garrido, J. Cortada, M. Valero, Improving Accuracy and Speeding Up Document Image Classification Through Parallel Systems, Springer International Publishing, 2020, p. 387–400. URL: http://dx.doi.org/10.1007/978-3-030-50417-5_29. doi:10.1007/978-3-030-50417-5_29.

[30] A. Sellami, S. Tabbone. Ednets: Deep feature learning for document image classification based on multi-view encoder-decoder neural networks, in: Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV, Springer-Verlag, Berlin, Heidelberg, 2021, p. 318–332. URL: 10.1007/978-3-030-86337-1_22. doi:10.1007/978-3-030-86337-1_22.

[31] S. Kanchi, A. Pagani, H. Mokayed, M. Liwicki, D. Stricker, M. Z. Afzal. Emmdocclassifier: Efficient multimodal document image classifier for scarce data. Applied Sciences 12 (2022). URL: https://www.mdpi.com/2076-3417/12/3/1457.

[32] S. Bakkali, Z. Ming, M. Coustaty, M. Rusiñol. Eaml: ensemble self-attention-based mutual learning network for document image classification. Int. J. Doc. Anal. Recognit. 24 (2021) 251–268. URL: https://doi.org/10.1007/s10032-021-00378-0. doi:10.1007/s10032-021-00378-0.

[33] Y. Yu, Y. Li, C. Zhang, X. Zhang, Z. Guo, X. Qin, K. Yao, J. Han, E. Ding, J. Wang. Structextv2: Masked visual-textual prediction for document image pre-training. International Conference on Learning Representations 2023 (2023). URL: https://arxiv.org/abs/2303.00289. doi:10.48550/arXiv.2303.00289. arXiv:2303.00289.

[34] S. Bakkali, S. Biswas, Z. Ming, M. Coustaty, M. Rusinol, O. R. Terrades, J. Llad'os. Globaldoc: A cross-modal vision-language framework for real-world document image retrieval and classification, in: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). URL: https://api.semanticscholar.org/CorpusID:261696768.

[35] S. Krithika, A. R. Priyadharshini, G. Bharathi Mohan, M. Gayathri. A multimodal neural network architecture for document image classification, in: 2024 15th International

Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–6. doi:10.1109/ICCCNT61001.2024.10725474.

[36] J. Voerman, I. Souleiman Mahamoud, M. Coustaty, A. Joseph, V. Poulain d'Andecy, J.-M. Ogier. Automatic classification of company's document stream: Comparison of two solutions. Pattern Recognition Letters 172 (2023) 181–187. URL: https://www.sciencedirect.com/science/article/pii/S0167865523001915. doi:https://doi.org/10.1016/j.patrec.2023.06.012.

[37] X. Ling, M. Gao, D. Wang. Intelligent document processing based on rpa and machine learning, in: 2020 Chinese Automation Congress (CAC), pp. 1349–1353. doi:10.1109/CAC51589.2020.9326579.

[38] M. Ondrejcek, J. Kastner, R. Kooper, P. Bajcsy. Information extraction from scanned engineering drawings. National Center for Supercomputing Applications, University of Illinoisat Urbana-Champaign, Image Spatial DataAnalysis Group (2009) 1.

[39] C. Haar, H. Kim, L. Koberg. Ai-based engineering and production drawing information extraction, in: K.-Y. Kim, L. Monplaisir, J. Rickli (Eds.), Flexible Automation and Intelligent Manufacturing: The Human-Data-Technology Nexus, Springer International Publishing, Cham, 2023, pp. 374–382.

[40] T. Al-Wesabi, A. Bach, P. Schönfelder, I. Staka, M. König. Extracting information from old and scanned engineering drawings of existing buildings for the creation of digital building models, in: S. Skatulla, H. Beushausen (Eds.), Advances in Information Technology in Civil and Building Engineering, Springer International Publishing, Cham, 2024, pp. 171–186.

[41] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, J. Kannala. Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis, in: M. Felsberg, P.-E. Forssén, I.-M. Sintorn, J. Unger (Eds.), Image Analysis, Springer International Publishing, Cham, 2019, pp. 28–40.

[42] D. Cho, J. Kim, E. Shin, J. Choi, J.-k. Lee. Recognizing architectural objects in floor-plan drawings using deep-learning style-transfer algorithms, in: CAADRIA 2020: RE:Anthropocene, pp. 717–725. doi:10.52842/conf.caadria.2020.2.717.

[43] H. Bhanbhro, Y. K. Hooi, Z. Hassan, N. Sohu. Modern deep learning approaches for symbol detection in complex engineering drawings, in: 2022 International Conference on Digital Transformation and Intelligence (ICDI), pp. 121–126. doi:10.1109/ICDI57181.2022.10007281.

[44] X. Xiao, Z. Li, S. Zhao, L. Yang, F. Zhao, C. Ge. Improved pid symbol detection algorithm based on yolov5 network, in: 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 120–126. doi:10.1109/SMC53992.2023.10394450.

[45] H. Bhanbhro, Y. K. Hooi, W. Kusakunniran, Z. H. Amur. Symbol detection in a multi-class dataset based on single line diagrams using deep learning models. Int. J. Adv. Comput. Sci. Appl. 14 (2023). doi:10.14569/IJACSA.2023.0140806.

[46] C. Liu, J. Wu, P. Kohli, Y. Furukawa. Raster-to-vector: Revisiting floorplan transformation, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2214–2222. doi:10.1109/ICCV.2017.241.

[47] H. Kim, S. Kim, K. Yu. Automatic extraction of indoor spatial information from floor plan image: A patch-based deep learning methodology application on large-scale complex buildings. ISPRS International Journal of Geo-Information 10 (2021) 828. doi:10.3390/ijgi10120828.

[48] C. Gatto, A. Farina, C. Mirarchi, A. Pavan. Development of a framework for processing unstructured text dataset through nlp in cost estimation aec sector, in: European Conference on Computing in Construction. doi:10.35490/EC3.2023.232.

[49] G. Chen, Y. Hu, Z. Wang, Z. Song, J. Hu, T. Yang, Q. Wang. Nested named entity recognition in geotechnical engineering based on pre-training and information enhancement, in: D.-S. Huang, Z. Si, Q. Zhang (Eds.), Advanced Intelligent Computing Technology and Applications, Springer Nature Singapore, Singapore, 2024, pp. 291–303.

[50] K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[51] M. Z. Afzal, A. Kolsch, S. Ahmed, M. Liwicki. Cutting the error by half: Investigation of very deep cnn and advanced training strategies for document image classification, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2017. URL: http://dx.doi.org/10.1109/ICDAR.2017.149. doi:10.1109/icdar.2017.149.

[52] S. Jiang, J. Hu, C. L. Magee, J. Luo. Deep learning for technical document classification. IEEE Transactions on Engineering Management 71 (2024) 1163–1179. URL: http://dx.doi.org/10.1109/TEM.2022.3152216. doi:10.1109/tem.2022.3152216.

[53] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014). URL: https://api.semanticscholar.org/CorpusID:14124313.

[54] S. Pramanik, S. Mujumdar, H. Patel, Towards a multi-modal, multi-task learning based pre-training framework for document representation learning, 2022. URL: https://arxiv.org/abs/2009.14457. arXiv:2009.14457.

[55] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, M. Zhang, L. Zhou. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding, in: Proceedings of the 59th Annual Meeting of the Association for Computational

Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2579–2591. doi:10.18653/v1/2021.acl-long.201.

[56] O. Hamed, S. Bakkali, M. Blaschko, S. Moens, J. Van Landeghem. Multimodal adaptive inference for document image classification with anytime early exiting, in: E. H. Barney Smith, M. Liwicki, L. Peng (Eds.), Document Analysis and Recognition - ICDAR 2024, Springer Nature Switzerland, Cham, 2024, pp. 270–286. doi:10.1007/978-3-031-70546-5_16.

[57] N. Audebert, C. Herold, K. Slimani, C. Vidal. Multimodal deep networks for text and image-based document classification, in: P. Cellier, K. Driessens (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2020, pp. 427–443.

[58] T. Elsken, J. H. Metzen, F. Hutter. Neural architecture search: A survey. Journal of Machine Learning Research 20 (2019) 1–21. URL: http://jmlr.org/papers/v20/18-598.html. doi:10.5555/3322706.3361996.

[59] F. Esposito, S. Ferilli, T. Basile, N. Di Mauro. Intelligent document processing, in: Eighth International Conference on Document Analysis and Recognition (ICDAR'05), pp. 1100–1104 Vol. 2. doi:10.1109/ICDAR.2005.144.

[60] B. Kitchenham. Procedures for performing systematic reviews. Keele, UK, Keele Univ. 33 (2004).

[61] N. Sajadfar, S. Abdollahnejad, U. Hermann, Y. Mohamed. Text detection and classification of construction documents, in: Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC), IAARC, pp. 446–452. doi:10.22260/ISARC2019/0060.

[62] M. Bodenbender, B.-M. Kurzrock, P. M. Müller. Broad application of artificial intelligence for document classification, information extraction and predictive analytics in real estate. Journal of general management 44 (2019) 170–179.

[63] H. Kim, Y. Jang, H. Kang, J. Son, J.-S. Yi. A suggestion of the direction of construction disaster document management through text data classification model based on deep learning. Korean Journal of Construction Engineering and Management 22 (2021) 73–85. doi:10.6106/KJCEM.2021.22.5.073.

[64] Z. Ren, F. Wan, H. Yu, T. Wu. Research on document classification in the field of construction, in: Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019), Atlantis Press, pp. 501–506. doi:10.2991/iccia-19.2019.78.

[65] A. Guha, D. Samanta. Real-time application of document classification based on machine learning, in: Intelligent Computing Paradigm and Cutting-edge Technologies, Springer, pp. 366–379. doi:10.1007/978-3-030-38501-9_37.

[66] J. Sun, K. Lei, L. Cao, B. Zhong, Y. Wei, J. Li, Z. Yang. Text visualization for construction document information management. Automation in Construction 110 (2020) 103048. doi:10.1016/j.autcon.2019.103048.

[67] D. Kang, M. Cho, G. CHa, S. Park. Development of svm-based construction project document classification model to derive construction risk. KSCE JOURNAL OF CIVIL AND ENVIRONMENTAL ENGINEERING RESEARCH (2023). doi:10.12652/Ksce.2023.43.6.0841.

[68] Y. Wang, Z. Zhang, Z. Wang, C. Wang, C. Wu. Interpretable machine learning-based text classification method for construction quality defect reports. Journal of Building Engineering 89 (2024) 109330. URL: https://www.sciencedirect.com/science/article/pii/S2352710224008982. doi:https://doi.org/10.1016/j.jobe.2024.109330.

[69] L. Jacques de Sousa, J. Poças Martins, L. Sanhudo, J. Santos Baptista. Automation of text document classification in the budgeting phase of the construction process: A systematic literature review. Construction Innovation 24 (2024) 292–318. doi:10.1108/CI-12-2022-0315.

[70] W. Borst, D. Wiegreffe, G. Neumann. A demonstration system towards nlp and knowledge driven data platforms for civil engineering, in: GI-Jahrestagung, Gesellschaft für Informatik e.V., pp. 271–282. doi:10.18420/gi2022-03.

[71] L. Kang, J. Kumar, P. Ye, Y. Li, D. Doermann. Convolutional neural networks for document image classification, in: 2014 22nd International Conference on Pattern Recognition, pp. 3168–3172. doi:10.1109/ICPR.2014.546.

[72] J. Kumar, P. Ye, D. Doermann. Structural similarity for document image classification and retrieval. Pattern Recognition Letters 43 (2014) 119–126. URL: https://www.sciencedirect.com/science/article/pii/S0167865513004224. doi:https://doi.org/10.1016/j.patrec.2013.10.030, iCPR2012 Awarded Papers.

[73] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie, R. Manmatha. Docformer: End-to-end transformer for document understanding, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 973–983. doi:10.1109/ICCV48922.2021.00103.

[74] J. Gu, J. Kuen, V. I. Morariu, H. Zhao, N. Barmpalios, R. Jain, A. Nenkova, T. Sun. Unified pretraining framework for document understanding, in: Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21, Curran Associates Inc., Red Hook, NY, USA, 2021.

[75] J. K. Mandivarapu, E. Bunch, Q. You, G. Fung, Efficient document image classification using region-based graph neural network, 2021. URL: https://arxiv.org/abs/2106.13802. arXiv:2106.13802.

[76] Y. Xiong, Z. Dai, Y. Liu, X. Ding. Document image classification method based on graph convolutional network, in: T. Mantoro, M. Lee, M. A. Ayu, K. W. Wong, A. N. Hidayanto (Eds.), Neural Information Processing, Springer International Publishing, Cham, 2021, pp. 317–329.

[77] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, S. Park. Ocr-free document understanding transformer, in: Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII, Springer-Verlag, Berlin, Heidelberg, 2022, p. 498–517. URL: https://doi.org/10.1007/978-3-031-19815-1_29. doi:10.1007/978-3-031-19815-1_29.

[78] P. Kaddas, B. Gatos. Using multi-level segmentation features for document image classification, in: S. Uchida, E. Barney, V. Eglin (Eds.), Document Analysis Systems, Springer International Publishing, Cham, 2022, pp. 702–712.

[79] J. Li, Y. Xu, T. Lv, L. Cui, C. Zhang, F. Wei. Dit: Self-supervised pre-training for document image transformer, in: Proceedings of the 30th ACM International Conference on Multimedia, MM ’22, Association for Computing Machinery, New York, NY, USA, 2022, p. 3530–3539. URL: 10.1145/3503161.3547911. doi:10.1145/3503161.3547911.

[80] T. Fronteau, A. Paran, A. Shabou. Evaluating adversarial robustness on document image classification, in: G. A. Fink, R. Jain, K. Kise, R. Zanibbi (Eds.), Document Analysis and Recognition - ICDAR 2023, Springer Nature Switzerland, Cham, 2023, pp. 290–304.

[81] S. Saifullah, S. Agne, A. Dengel, S. Ahmed. Docxclassifier: High performance explainable deep network for document image classification. Authorea Preprints (2023).

[82] S. Bakkali, Z. Ming, M. Coustaty, M. Rusiñol, O. R. Terrades. Vlcdoc: Vision-language contrastive pre-training model for cross-modal document classification. Pattern Recognition 139 (2023) 109419. URL: https://www.sciencedirect.com/science/article/pii/S0031320323001206. doi:https://doi.org/10.1016/j.patcog.2023.109419.

[83] A. D. Mahajan, S. Karuppasamy, S. Lakshminarayanan. Improving classification of scanned document images using a novel combination of pre-processing techniques, in: 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp. 1109–1114. doi:10.1109/ICAAIC56838.2023.10140911.

[84] T. Ali, P. P. Roy, Enhancing document information analysis with multi-task pre-training: A robust approach for information extraction in visually-rich documents, 2023. URL: https://arxiv.org/abs/2310.16527. doi:10.48550/arXiv.2310.16527. arXiv:2310.16527.

[85] Saifullah, S. Agne, A. Dengel, S. Ahmed. The reality of high performing deep learning models: A case study on document image classification. IEEE Access 12 (2024) 103537–103564. doi:10.1109/ACCESS.2024.3425910.

[86] S. Saifullah, S. Agne, A. Dengel, S. Ahmed. Docxclassifier: towards a robust and interpretable deep neural network for document image classification. International Journal on Document Analysis and Recognition (IJDAR) 27 (2024) 447–473. URL: https://link.springer.com/article/10.1007/s10032-024-00483-w. doi:10.1007/s10032-024-00483-w.

[87] S. Saifullah, S. Agne, A. Dengel, S. Ahmed. Docxplain: A novel model-agnostic explainability method for document image classification, in: E. H. Barney Smith, M. Liwicki, L. Peng (Eds.), Document Analysis and Recognition - ICDAR 2024, Springer Nature Switzerland, Cham, 2024, pp. 103–123. doi:10.1007/978-3-031-70546-5_7.

[88] M. S. I. Sajol, A. Hasan, M. Islam, M. Rahman. A convnext v2 approach to document image analysis: Enhancing high-accuracy classification, in: Proceedings of the 2024 IEEE 3rd Conference on Information Technology and Data Science (CITDS), Debrecen, Hungary, pp. 26–28.

[89] S. S. Shilpa. Graph attention-driven document image classification through dualtune learning. Indonesian Journal of Electrical Engineering and Computer Science 33 (2024) 278–289.

[90] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, S. Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16133–16142. doi:10.1109/CVPR52729.2023.01548.

[91] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie. A ConvNet for the 2020s , in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2022, pp. 11966–11976. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01167. doi:10.1109/CVPR52688.2022.01167.

[92] A. Tripathy, A. Anand, S. K. Rath. Document-level sentiment classification using hybrid machine learning approach. Knowledge and Information Systems 53 (2017) 805–831. doi:10.1007/s10115-017-1055-z.

[93] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp.

4171–4186. URL: https://aclanthology.org/N19-1423/. doi:10.18653/v1/N19-1423.

[94] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, T. Unterthiner, X. Zhai. An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=YicbFdNTTy.

[95] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5 (2016). doi:10.1162/tacl_a_00051.

[96] Z. Wang, J. Gu, C. Tensmeyer, N. Barmpalios, A. Nenkova, T. Sun, J. Shang, V. Morariu. MGDoc: Pre-training with multi-granular hierarchy for document image understanding, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3984–3993. URL: https://aclanthology.org/2022.emnlp-main.265. doi:10.18653/v1/2022.emnlp-main.265.

[97] C. Luo, G. Tang, Q. Zheng, C. Yao, L. Jin, C. Li, Y. Xue, L. Si. Bi-vldoc: Bidirectional vision-language modeling for visually-rich document understanding. International Journal on Document Analysis and Recognition (IJDAR) (2025) 1–12.

[98] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, 2020. URL: https://arxiv.org/abs/2004.05150. doi:10.48550/arXiv.2004.05150. arXiv:2004.05150.

[99] A. Kay. Tesseract: an open-source optical character recognition engine. Linux J. 2007 (2007) 2.

[100] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov. Bag of tricks for efficient text classification, in: M. Lapata, P. Blunsom, A. Koller (Eds.), Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 427–431. URL: https://aclanthology.org/E17-2068/.

[101] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[102] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL:

https://aclanthology.org/2020.acl-main.703/. doi:10.18653/v1/2020.acl-main.703.

[103] R. Wirth, J. Hipp. Crisp-dm: Towards a standard process model for data mining, in: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, volume 1, Manchester, pp. 29–39.

[104] F. Karl, J. Thomas, J. Elstner, R. Gross, B. Bischl, Automated Machine Learning, Springer Nature Switzerland, Cham, 2024, pp. 3–25. URL: https://doi.org/10.1007/978-3-031-64832-8_1. doi:10.1007/978-3-031-64832-8_1.

[105] T. Czvetkó, A. Kummer, T. Ruppert, J. Abonyi. Data-driven business process management-based development of industry 4.0 solutions. CIRP Journal of Manufacturing Science and Technology 36 (2022) 117–132. URL: https://www.sciencedirect.com/science/article/pii/S1755581721001929. doi:https://doi.org/10.1016/j.cirpj.2021.12.002.

[106] F. Hutter, L. Kotthoff, J. Vanschoren, Automated Machine Learning: Methods, Systems, Challenges, 1st ed., Springer Publishing Company, Incorporated, 2019.

[107] R. Barbudo, S. Ventura, J. R. Romero. Eight years of automl: categorisation, review and trends. Knowledge and Information Systems 65 (2023) 5097–5149. URL: http://dx.doi.org/10.1007/s10115-023-01935-1. doi:10.1007/s10115-023-01935-1.

[108] Y. Li, Z. Wang, Y. Xie, B. Ding, K. Zeng, C. Zhang. Automl: From methodology to application, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM ’21, Association for Computing Machinery, New York, NY, USA, 2021, p. 4853–4856. URL: https://doi.org/10.1145/3459637.3483279. doi:10.1145/3459637.3483279.

[109] S. K. Karmaker (“Santu”), M. M. Hassan, M. J. Smith, L. Xu, C. Zhai, K. Veeramachaneni. Automl to date and beyond: Challenges and opportunities. ACM Comput. Surv. 54 (2021). URL: https://doi.org/10.1145/3470918. doi:10.1145/3470918.

[110] R. Mithe, S. Indalkar, N. Divekar. Optical character recognition. International journal of recent technology and engineering (IJRTE) 2 (2013) 72–75.

[111] V. Nasteski. An overview of the supervised machine learning methods. Horizons. b 4 (2017) 56.

[112] I. Dimitrovski, I. Kitanovski, D. Kocev, N. Simidjievski. Current trends in deep learning for earth observation: An open-source benchmark arena for image classification. ISPRS Journal of Photogrammetry and Remote Sensing 197 (2023) 18–35. URL: https://www.sciencedirect.com/science/article/pii/S0924271623000205. doi:https://doi.org/10.1016/j.isprsjprs.2023.01.014.

[113] T. Zebin, S. Rezvy. Covid-19 detection and disease progression visualization: Deep learning on chest x-rays for classification and coarse localization. Applied Intelligence 51 (2020) 1010–1021. URL: http://dx.doi.org/10.1007/s10489-020-01867-1. doi:10.1007/s10489-020-01867-1.

[114] L. Yao, B. Liu, Y. Xin. Visualization-based comprehensive feature representation with improved efficientnet for malicious file and variant recognition. Journal of Information Security and Applications 86 (2024) 103865. URL: https://www.sciencedirect.com/science/article/pii/S2214212624001674. doi:https://doi.org/10.1016/j.jisa.2024.103865.

[115] S. Mascarenhas, M. Agarwal. A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification, in: 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), volume 1, pp. 96–99. doi:10.1109/CENTCON52345.2021.9687944.

[116] A. Victor Ikechukwu, S. Murali, R. Deepu, R. Shivamurthy. Resnet-50 vs vgg-19 vs training from scratch: A comparative analysis of the segmentation and classification of pneumonia from chest x-ray images. Global Transitions Proceedings 2 (2021) 375–381. URL: https://www.sciencedirect.com/science/article/pii/S2666285X21000558. doi:https://doi.org/10.1016/j.gltp.2021.08.027, international Conference on Computing System and its Applications (ICCSA- 2021).

[117] G. Nijaguna, J. A. Babu, B. Parameshachari, R. P. de Prado, J. Frnda. Quantum fruit fly algorithm and resnet50-vgg16 for medical diagnosis. Applied Soft Computing 136 (2023) 110055. URL: https://www.sciencedirect.com/science/article/pii/S156849462300073X. doi:https://doi.org/10.1016/j.asoc.2023.110055.

[118] O. Polat, C. Güngen. Classification of brain tumors from mr images using deep transfer learning. The Journal of Supercomputing 77 (2021) 7236–7252. URL: http://dx.doi.org/10.1007/s11227-020-03572-9. doi:10.1007/s11227-020-03572-9.

[119] M. Talo, O. Yildirim, U. B. Baloglu, G. Aydin, U. R. Acharya. Convolutional neural networks for multi-class brain disease detection using mri images. Computerized Medical Imaging and Graphics 78 (2019) 101673. URL: https://www.sciencedirect.com/science/article/pii/S0895611119300886. doi:https://doi.org/10.1016/j.compmedimag.2019.101673.

[120] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, AAAI Press, 2017, p. 4278–4284.

[121] F.-C. Chen, A. Subedi, M. R. Jahanshahi, D. R. Johnson, E. J. Delp. Deep learning–based building attribute estimation from google street view images for flood risk assessment

using feature fusion and task relation encoding. Journal of Computing in Civil Engineering 36 (2022) 04022031. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CP.1943-5487.0001025. doi:10.1061/(ASCE)CP.1943-5487.0001025.

[122] W. Wang, W. Hu, W. Wang, X. Xu, M. Wang, Y. Shi, S. Qiu, E. Tutumluer. Automated crack severity level detection and classification for ballastless track slab using deep convolutional neural network. Automation in Construction 124 (2021) 103484. URL: https://www.sciencedirect.com/science/article/pii/S0926580520310645. doi:https://doi.org/10.1016/j.autcon.2020.103484.

[123] R. Ehtisham, W. Qayyum, C. V. Camp, V. Plevris, J. Mir, Q. uz Zaman Khan, A. Ahmad. Computing the characteristics of defects in wooden structures using image processing and cnn. Automation in Construction 158 (2024) 105211. URL: https://www.sciencedirect.com/science/article/pii/S0926580523004715. doi:https://doi.org/10.1016/j.autcon.2023.105211.

[124] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520.

[125] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. A. Emadi, M. B. I. Reaz, M. T. Islam. Can ai help in screening viral and covid-19 pneumonia? IEEE Access 8 (2020) 132665–132676. doi:10.1109/ACCESS.2020.3010287.

[126] M. Tan, Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

[127] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, M. Nissim, Bertje: A dutch bert model, 2019. URL: https://arxiv.org/abs/1912.09582. arXiv:1912.09582.

[128] P. Delobelle, T. Winters, B. Berendt. RobBERT: a Dutch RoBERTa-based Language Model, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 3255–3265. URL: https://aclanthology.org/2020.findings-emnlp.292/. doi:10.18653/v1/2020.findings-emnlp.292.

[129] L. De Bruyne, O. De Clercq, V. Hoste. Emotional RobBERT and insensitive BERTje: Combining transformers and affect lexica for Dutch emotion detection, in: O. De Clercq, A. Balahur, J. Sedoc, V. Barriere, S. Tafreshi, S. Buechel, V. Hoste (Eds.), Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social

Media Analysis, Association for Computational Linguistics, Online, 2021, pp. 257–263. URL: https://aclanthology.org/2021.wassa-1.27/.

[130] M. T. Rietberg, V. B. Nguyen, J. Geerdink, O. Vijlbrief, C. Seifert. Accurate and reliable classification of unstructured reports on their diagnostic goal using bert models. Diagnostics 13 (2023). URL: https://www.mdpi.com/2075-4418/13/7/1251. doi:10.3390/diagnostics13071251.

[131] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown. Text classification algorithms: A survey. Information 10 (2019). URL: https://www.mdpi.com/2078-2489/10/4/150. doi:10.3390/info10040150.

[132] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, L. He. A survey on text classification: From traditional to deep learning. ACM Transactions on Intelligent Systems and Technology 13 (2022). URL: https://doi.org/10.1145/3495162. doi:10.1145/3495162.

[133] R. Nisbet, J. Elder, G. Miner, in: R. Nisbet, J. Elder, G. Miner (Eds.), Handbook of Statistical Analysis and Data Mining Applications, Academic Press, Boston, 2009, pp. 285–312. URL: https://www.sciencedirect.com/science/article/pii/B9780123747655000139. doi:https://doi.org/10.1016/B978-0-12-374765-5.00013-9.

[134] P. Refaeilzadeh, L. Tang, H. Liu, Cross-Validation, Springer US, 2009, p. 532–538. URL: http://dx.doi.org/10.1007/978-0-387-39940-9_565. doi:10.1007/978-0-387-39940-9_565.

[135] M. Sokolova, G. Lapalme. A systematic analysis of performance measures for classification tasks. Information Processing Management 45 (2009) 427–437. URL: https://www.sciencedirect.com/science/article/pii/S0306457309000259. doi:https://doi.org/10.1016/j.ipm.2009.03.002.

[136] P. Branco, L. Torgo, R. Ribeiro. Relevance-based evaluation metrics for multi-class imbalanced domains, in: Lecture Notes in Computer Science, pp. 698–710. doi:10.1007/978-3-319-57454-7_54.

[137] H. He, E. A. Garcia. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering 21 (2009) 1263–1284.

[138] L. Prechelt, Early Stopping - But When?, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 55–69. URL: https://doi.org/10.1007/3-540-49430-8_3. doi:10.1007/3-540-49430-8_3.

[139] S. Larson, N. Singh, S. Maheshwari, S. Stewart, U. Krishnaswamy. Exploring out-of-distribution generalization in text classifiers trained on tobacco-3482 and rvl-cdip, in: E. H. Barney Smith, U. Pal (Eds.), Document Analysis and Recognition – ICDAR 2021 Workshops, Springer International Publishing, Cham, 2021, pp. 416–423.

[140] C. Shin, D. Doermann, A. Rosenfeld. Classification of document pages using structure-based features. International Journal on Document Analysis and Recognition 3 (2001) 232–247.

[141] S. S. Bukhari, A. Dengel. Visual appearance based document classification methods: Performance evaluation and benchmarking, in: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 981–985. doi:10.1109/ICDAR.2015.7333908.

[142] C. Tensmeyer, T. Martinez. Analysis of convolutional neural networks for document image classification, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pp. 388–393. doi:10.1109/ICDAR.2017.71.

[143] S. Rehman, A. Irtaza, M. Nawaz, H. Kibriya. Text document classification using deep learning techniques, in: 2022 International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE), pp. 1–6. doi:10.1109/ETECTE55893.2022.10007316.

[144] H. Zhou. Research of text classification based on tf-idf and cnn-lstm, in: journal of physics: conference series, volume 2171, IOP Publishing, p. 012021.

[145] Y. Jiang, L. Zheng. Deep learning for video game genre classification. Multimedia Tools and Applications 82 (2023) 21085–21099.

[146] H. Wu, D. Klabjan. Logit-based uncertainty measure in classification, in: 2021 IEEE International Conference on Big Data (Big Data), IEEE, pp. 948–956.

[147] S. Ahmad. Visit: A neural model of covert visual attention, in: J. Moody, S. Hanson, R. Lippmann (Eds.), Advances in Neural Information Processing Systems, volume 4, Morgan-Kaufmann, 1991. URL: https://proceedings.neurips.cc/paper_files/paper/1991/file/7f24d240521d99071c93af3917215ef7-Paper.pdf.

[148] D. Soydaner. Attention mechanism in neural networks: where it comes and where it goes. Neural Computing and Applications 34 (2022) 13371–13385.

[149] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera, Learning from imbalanced data sets, volume 10, Springer, 2018.

[150] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, volume 1, MIT press Cambridge, 2016.

[151] L. Li, L. Xiao, N. Wang, G. Yang, J. Zhang. Text classification method based on convolution neural network, in: 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 1985–1989. doi:10.1109/CompComm.2017.8322884.

[152] R. Qasim, W. H. Bangyal, M. A. Alqarni, A. Ali Almazroi. A fine-tuned bert-based transfer learning approach for text classification. Journal of healthcare engineering 2022 (2022) 3498123. doi:10.1155/2022/3498123.

[153] R. Powalski, Ł. Borchmann, D. Jurkiewicz, T. Dwojak, M. Pietruszka, G. Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer, in: J. Lladós, D. Lopresti, S. Uchida (Eds.), Document Analysis and Recognition – ICDAR 2021, Springer International Publishing, Cham, 2021, pp. 732–747.

[154] R. Karpinski, D. Lohani, A. Belaid. Metrics for Complete Evaluation of OCR Performance, in: IPCV'18 - The 22nd Int'l Conf on Image Processing, Computer Vision, & Pattern Recognition, Las Vegas, United States. URL: https://inria.hal.science/hal-01981731.

[155] D. L. Davies, D. W. Bouldin. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1 (1979) 224 – 227. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0017953820&doi=10.1109%2fTPAMI.1979.4766909&partnerID=40&md5=9221608c222ec58263374584437c8f52. doi:10.1109/TPAMI.1979.4766909.

[156] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20 (1987) 53–65. URL: https://www.sciencedirect.com/science/article/pii/0377042787901257. doi:https://doi.org/10.1016/0377-0427(87)90125-7.

[157] Y. Luo, Y. Wong, M. Kankanhalli, Q. Zhao. $\mathscr{G}$ -softmax: Improving intraclass compactness and interclass separability of features. IEEE Transactions on Neural Networks and Learning Systems 31 (2020) 685–699. doi:10.1109/TNNLS.2019.2909737.

[158] Y. Liu, L. Li, J. Tan, Y. Rao, X. Tan, Y. Li. Cross-domain fisher discrimination criterion: A domain adaptive method based on the nature of classifier. Applied Intelligence 54 (2024) 5389–5405. URL: http://dx.doi.org/10.1007/s10489-024-05376-3. doi:10.1007/s10489-024-05376-3.

[159] L. van der Maaten, G. Hinton. Visualizing data using t-sne. Journal of Machine Learning Research 9 (2008) 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html. doi:10.1145/1390156.1390257.

A

KEYWORD CLOUDS

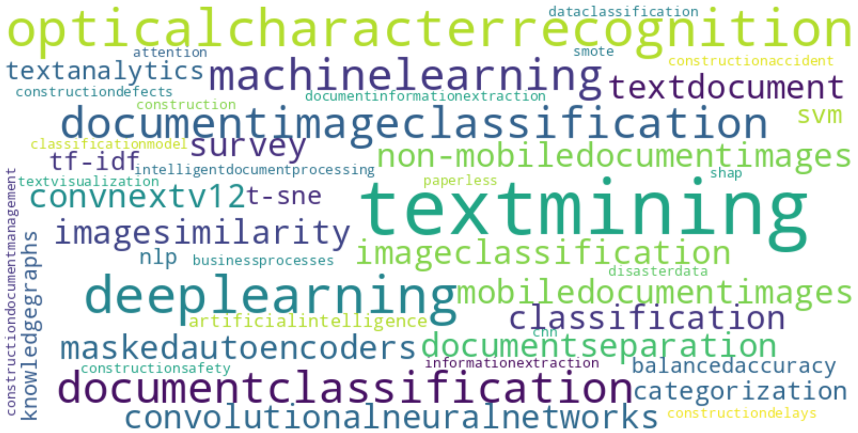


Figure A.1: Keyword Density Wordcloud for document classification in AEC applications keywords

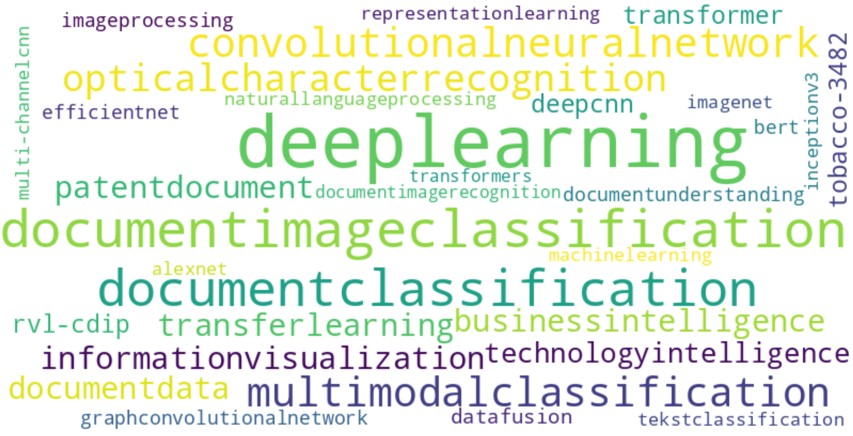


Figure A.2: Keyword Density Wordcloud for (multimodal) document image classification.

B

Publication Counts per Conference/Journal

Table B.1: Publication Counts per Conference/Journal (Document Classification in AEC applications)

Conference/Journal	**Count**
ArXiv	1
Automation in Construction	1
Construction Innovation	1
IEEE Transactions on Engineering Management	1
INFORMATIK 2022. Gesellschaft für Informatik	1
Intelligent Computing Paradigm and Cutting-edge Technologies	1
International Conference on Computer Engineering, Information Science & Application Technology (ICCIA)	1
International Symposium on Automation and Robotics in Construction (ISARC)	1
Journal of Building Engineering	1
Journal of General Management	1
KSCE Journal of Civil and Environmental Engineering Research	1
Korean Journal of Construction Engineering and Management	1

Table B.2: Publication Counts per Conference/Journal (Multimodal Document Classification/ Document Image Classification), only including conference/journals having published more than one of the selected materials.

Conference/Journal	**Count**
ArXiv	10
International Conference on Document Analysis and Recognition (ICDAR)	9
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)	5
International Journal on Document Analysis and Recognition (IJDAR)	4
International Conference on Pattern Recognition (ICPR)	3
Proceedings of the ACM International Conference on Multimedia	2

C

Categorization of Feature Fusion Methods

Table C.1: Categorization of Feature Fusion Methods (part 1)

Fusion Methods	Fusion Level	Complexity (Simple/ Intermediate/ Advanced	Based on Attention Mechanisms	Based on Adaptivity
Multi-View Deep Autoencoder (MDAE)	Early Fusion	Intermediate	No Attention	Adaptive Fusion
multimodal Transformer Model	Early Fusion	Advanced	No Attention	Adaptive Fusion
Cross-Modal Attention Encoder	Hybrid Fusion	Advanced	Attention	Adaptive Fusion
Cross-Modal Interaction Attention Module (InterMCA and IntraMSA)	Hybrid Fusion	Advanced	Attention	Adaptive Fusion
Self-Attention	Hybrid Fusion	Advanced	Attention	Adaptive Fusion
Multi-Head Attention Based Encoder	Hybrid Fusion	Advanced	Attention	Adaptive Fusion
Adaptive Fusion Layer	Hybrid Fusion	Advanced	Attention	Adaptive Fusion
Feature Maps	Hybrid/Late Fusion Fusion	Intermediate	No Attention	Static Fusion

Table C.2: Categorization of Feature Fusion Methods (part 2)

Fusion Methods	**Fusion Level**	**Complexity (Simple/ Intermediate/ Advanced**	**Based on Attention Mechanisms**	**Based on Adaptivity**
Average Ensembling (Superposing) Method	Late Fusion	Simple	No Attention	Static Fusion
Equal Concatenation	Late Fusion	Simple	No Attention	Static Fusion
Sum of Weighted Probabilities	Late Fusion	Simple	No Attention	Static Fusion
Linear Transformation Layer	Late Fusion	Simple	No Attention	Static Fusion
Average Pooling Layer & Softmax	Late Fusion	Simple	No Attention	Static Fusion
Element-Wise Product and Averaging	Late Fusion	Intermediate	No Attention	Adaptive Fusion
Separate Softmax for Text and Visual, then Combined	Late Fusion	Intermediate	No Attention	Static Fusion
Sample-Dependent Attention Weights	Late Fusion	Intermediate	Attention	Adaptive Fusion

D

Optical Character Recognition Tool Comparison

OCR is the widely used method to extract text from images, which we elaborate on in Section 3.3.1. Over the years, numerous tools have been developed that are available for public and commercial use. In this study, we analyze the performance of three different publicly available OCR tools; PyTesseract OCR [1], EasyOCR[2], and Keras OCR[3]. Furthermore, we compare the results with the performance of a commercial OCR tool, Azure OCR[4].

For each of the classes, the text is extracted manually from two class images; in other words, we read the texts ourselves. We refer to these texts as *reference texts*. These texts are used as reference texts to compare the OCR results, which we refer to as *hypothesis texts*. For each of the words in the reference texts, we test whether it is included in the corresponding hypothesis text. The best performing OCR is the OCR that, on average, includes most of the manually extracted words.

[1] Tesseract OCR Github

[2] EasyOCR on Jaided AI

[3] Keras OCR Website

[4] Azure AI Vision

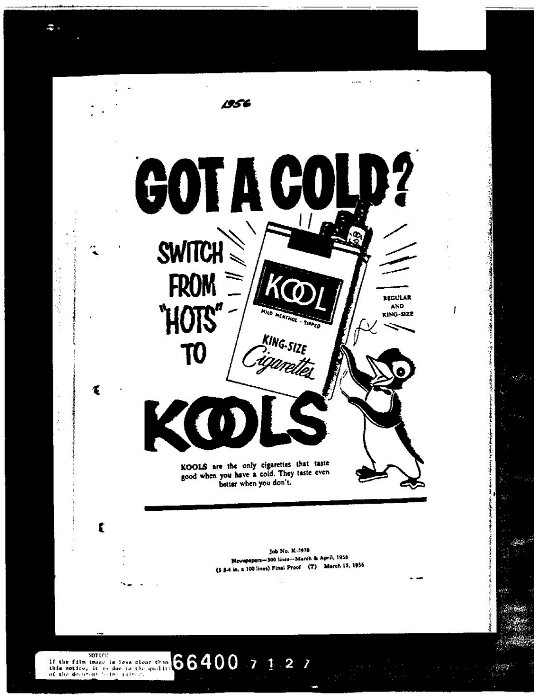

1956
GOT A COLD?
SWITCH FROM "HOTS" TO
KOOL
MILD MENTHOL · TIPPED
KING-SIZE Cigarettes
REGULAR AND KING-SIZE
KOOLS
KOOLS are the only cigarettes that taste good when you have a cold. They taste even better when you don't.
Job No. K-2978
Newspapers—300 lines—March & April, 1956
(5 3-4 in. x 100 lines) Final Proof (T) March 15, 1956
NOTICE
If the film image is less clear than this notice, it is due to the quality of the document being filmed.
66400 7 1 2 7

Figure D.1: Image from the Tobacco-3482 dataset

We use Figure D.1, which is an image from the Tobacco-3482 dataset, to illustrate this process. Manually we extract the following text from the image.

> *1956 GOT A COLD? SWITCH FROM "HOTS" TO KOOLS KOOLS are the only cigarettes that taste good when you have a cold. They taste even better when you don't. KOOL MILD MENTHOL TIPPEO REGULAR AND KING-SIZE job No K-2978 Newspapers - 300 lines - March & April, 1950 100 lines) Final Proof (T) March 15, 1956 66400 7 1 2 7*

Figure D.2: Reference Text

In this example, we test the performance of PyTesseract OCR. The following Figure shows the text as extracted by PyTesseract OCR. We convert both texts to lowercase, and then for each word in the reference text we is extract whether it is included by the hypothesis text as well.

> *KOOLS are the only cigarettes that taste good when you have & cold. They taste even 'better when you don't. Job No, K-2978 'Mewapapars—300 iner—Mateh & April, 1956 (8 9-4 in, 2 108 ines) Pinal Proof (7) March 18, 1956*

Figure D.3: Hypothesis Text

The reference text contains 61 words, while the hypothesis text, extracted by PyTesseract contains only 48 words. Among these, 28 words have an exact match between the two methods. We evaluate the performance of the OCR using the Strict Word Error Rate (SWER) as defined by Karpinski et al. [154]. This metric divides the number of incorrectly extracted words by the original length of the text (the reference text). A word that is not extracted at all is also seen as incorrectly extracted. For this image, the SWER is 54.10%. The aim is to have a SWER as close to 0 as possible.

> *1956 kools kools are the only cigarettes that taste good when you have cold. they taste even when you don't. job K-2978 march april, proof march 1956 2*

Figure D.4: Corresponding Words between Reference Text and Hypothesis Text

The limitation of this method is that it only checks whether a word is included somewhere in the extracted text OCR without ensuring that it is in the right position. It does however, provide a useful measure for comparing OCR performance in terms of the number of recognized words. The strict word error rate is calculated for 14 images in the dataset, with images selected per category. We denote the resulting error rates, subtracted from 100 percent, in Table D.1.

Table D.1: Optical Character Recognition Analysis Results in 100 - Strict Word Error Rate

OCR package	**Basic**	**Patches**	**Grayscale + Dilution**
PyTesseract	29.03%	11.73%	12.82%
EasyOCR	16.53%	17.15%	15.21%
Keras OCR	5.79%	10.03%	3.18%
Azure OCR	**32.95%**	x	x

Furthermore, not only is the text directly extracted from the images. We also test the impact of applying grayscale and dilution to images, as well as extracting text from image-patches instead of solely from the full images. We compare the hypothesis texts with the reference texts in the same way. The further used OCR tool and setting is the best found combination of the two.

E

Exploratory Data Analysis

E.0.1. Image Data Analysis

The metrics used for image data analysis are based on widely recognized image analysis standards. In this section, we discuss the results of the analysis and highlight the most evident differences between the classes. First, we evaluate the number of colors per class, as illustrated in Figure E.1. On average, the *Cross-Section Drawing* and *Detail Drawing* classes contain the least different colors. Aspect ratio, which represents the relation between the width and height of documents, indicates whether a document is in portrait (aspect ratio of approximately 0.7) or landscape (aspect ratio of approximately 1.4), or possibly does not conform to a a standard document form. We find that *Installation Diagram* is most consistent with landscape orientation, while *Report* is most consistent with portrait orientation. The other classes seem to vary between the two orientations and may included different document formats as well (see Figure E.2). The brightness of the images represents the average pixel intensity of the class images. A pixel intensity of 255 means that the image is completely white, while 0 signifies a completely black pixel. The average brightness of most classes varies between 238 and 248, while the average brightness of class *Photo* is significantly lower, indicating higher degree of darkness. For each image, we compute the most dominant color, i.e. the color that most pixels have. We see that for most images the dominant color is white, or a teint of white having a closely related RGB code. Figure E.4 shows the number of documents that have white as the dominant color as a percentage of the total number of documents per class. We observe that for most classes white is the dominant color for most documents while for *Photo* this is not the case. Lastly, we measure the image entropy, which quantifies the complexity of an image by evaluating the range and distribution of pixel values. A high entropy indicates a less predictable and more information-containing image, while a low entropy indicates the contrast. We observe that images in class *Photo* on average have higher entropy, whereas the rest of the classes have more similar entropy values E.5.

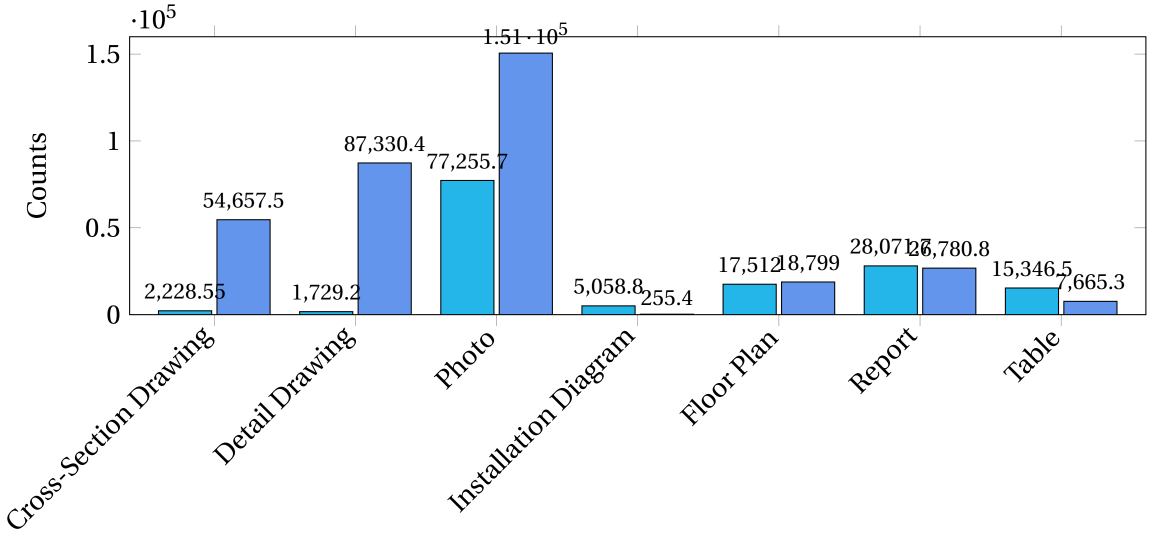


Figure E.1: Average Number of Colours per Class

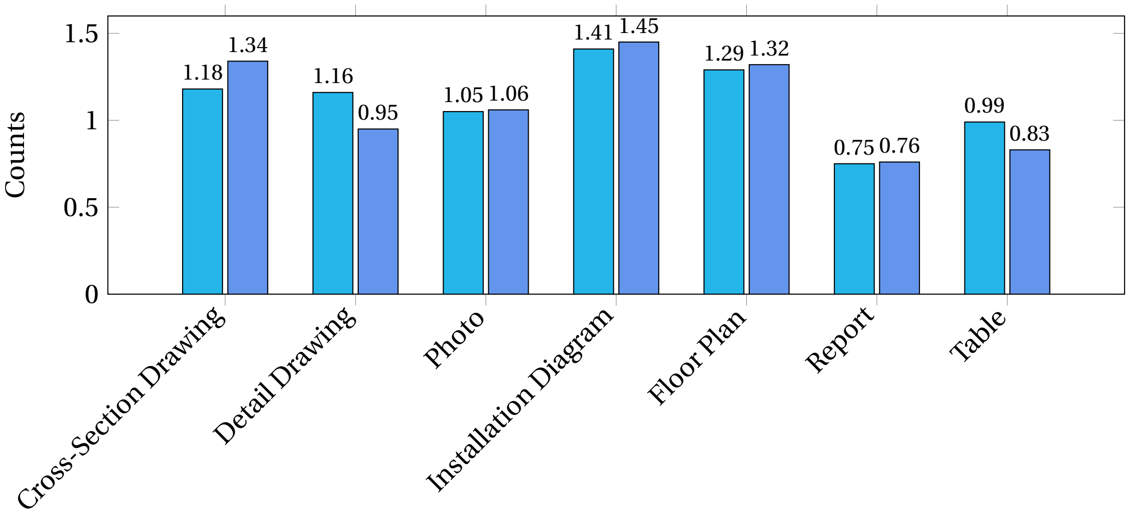


Figure E.2: Average Aspect Ratio per Class

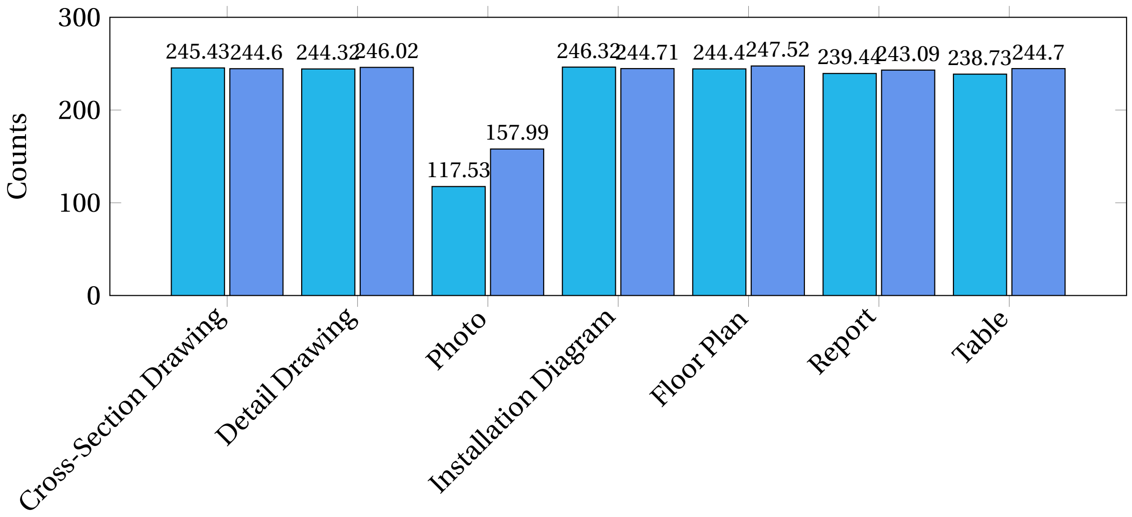


Figure E.3: Average Brightness per Class

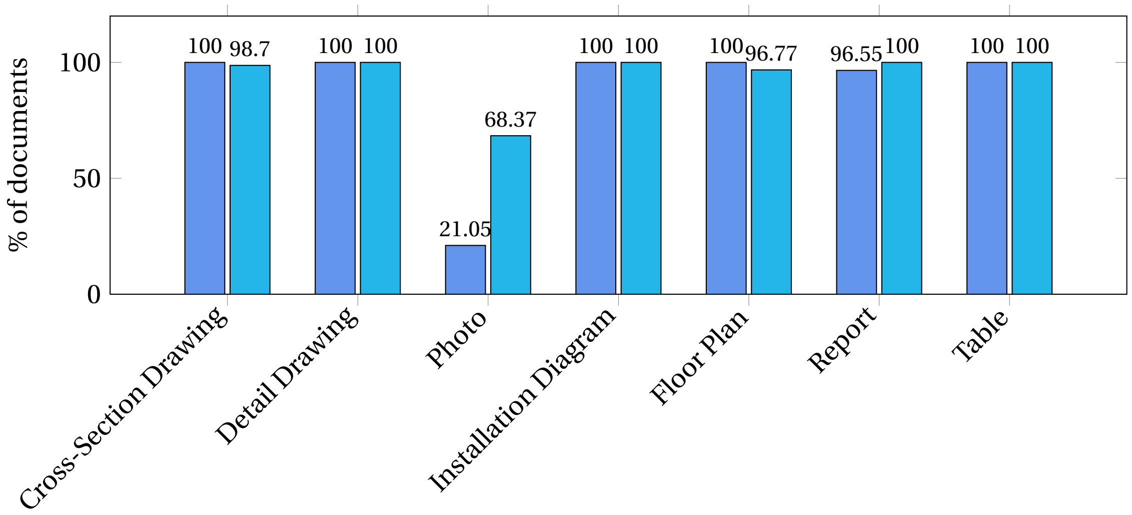


Figure E.4: Documents with Dominant Color White

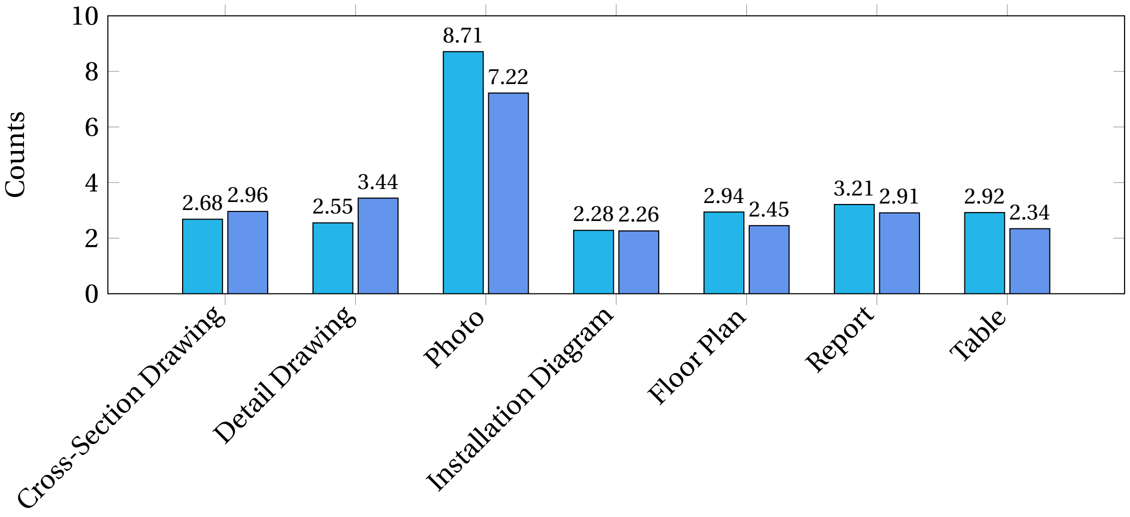


Figure E.5: Average Entropy per class

E.0.2. Text Data Analysis

For the text evaluation, we analyzed the number of words per document and the number of unique words per document across two datasets. We found that the number of words is not quite directly related to the classes, as there is significant variation between the two datasets (see Figure E.6. Generally, documents in dataset 2 contain fewer words on average, except for the classes *Cross-Section Drawing* and *Photo*. Both have the fewest words on average for the class *Photo*. Additionally, dataset 2 generally has a higher percentage of unique words per text than dataset 1 has, except for the class *Photo* (see Figure E.7).

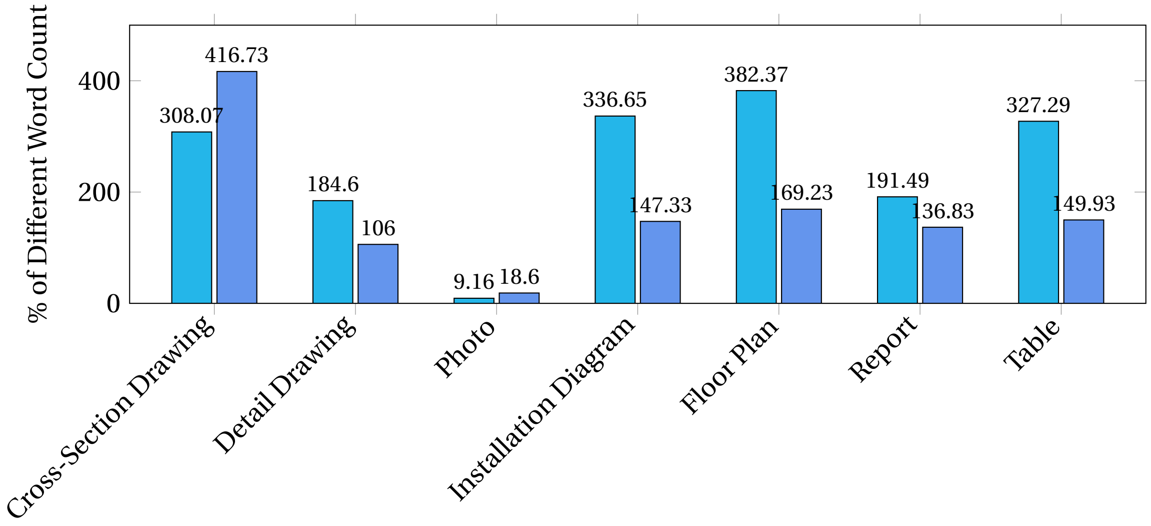


Figure E.6: Average Number of Different Words Per Class

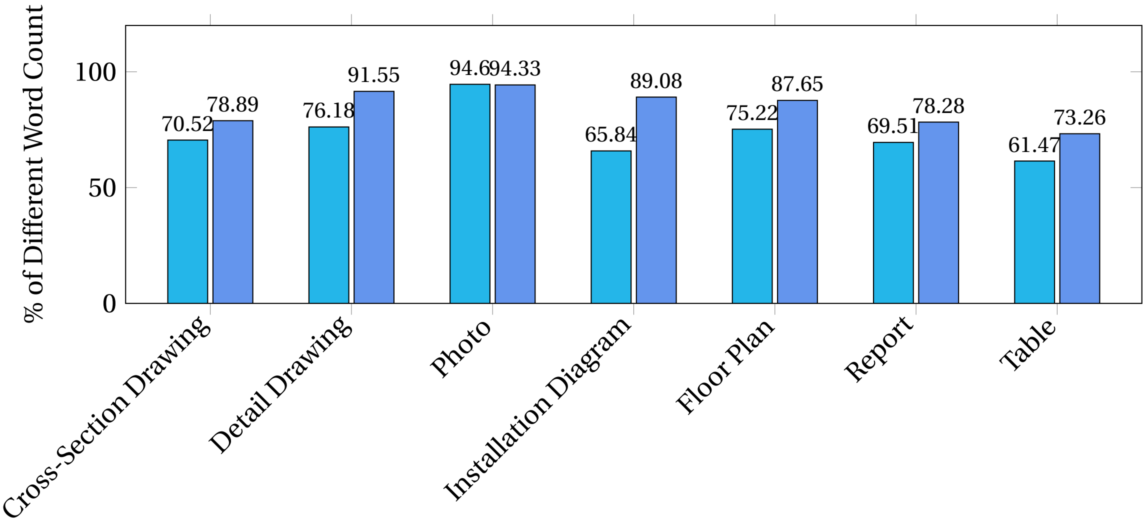


Figure E.7: Average Unique Words Per Class as Percentage of the Number of Different Words

E.0.3. Layout Data Analysis

For the layout data analysis we evaluate the bounding boxes that we extract by using by using OCR. We observe that classes *Installation Diagram*, *Floor Plan*, and *Table* on average have the most bounding boxes (see Figure E.8). In contrast, documents in class *Photo* have the least number of bounding boxes, but at the same time with the largest area (see Figure E.9).

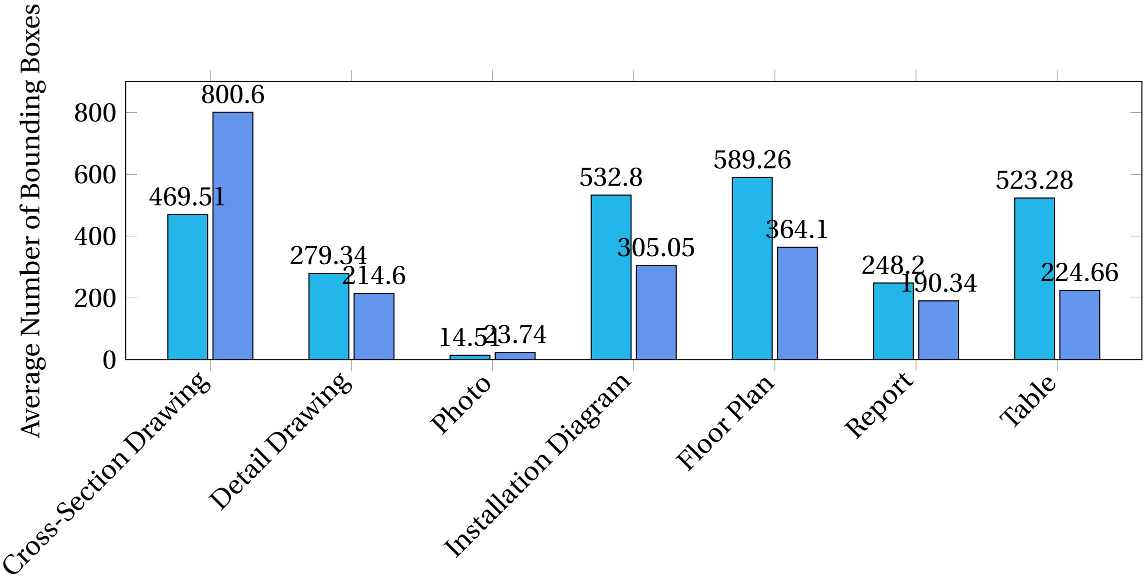


Figure E.8: Average Number of Bounding Boxes per Subfolder

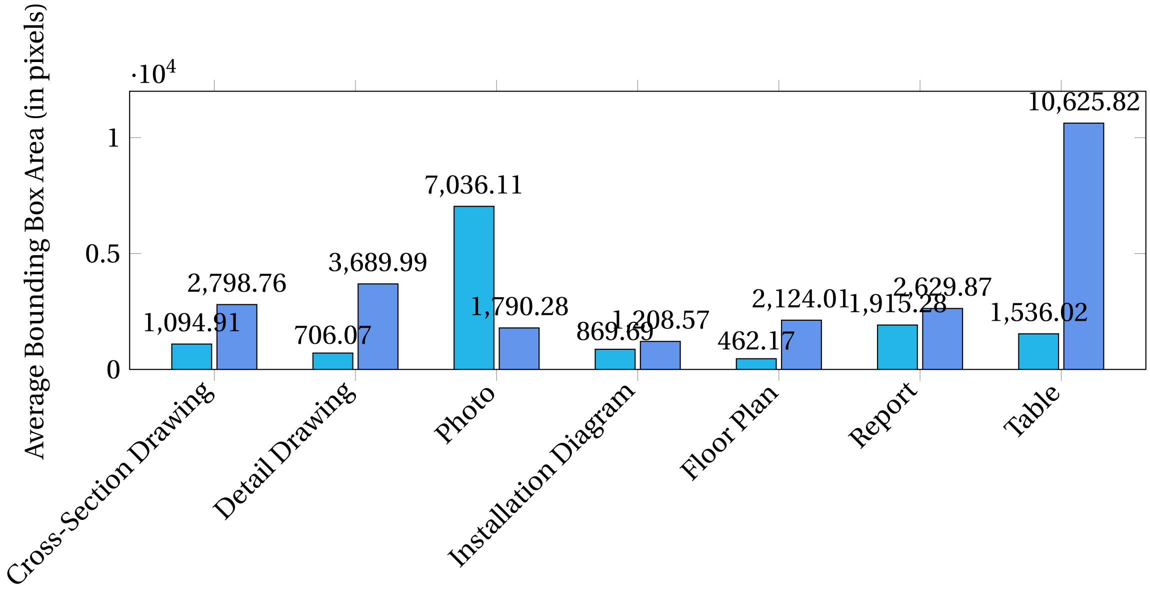


Figure E.9: Average Bounding Box Area per Category

E.0.4. Inter-class Similarity and Intra-class Compactness Analysis

The Davies–Bouldin Index (DBI) [155] and Silhouette Score [156] are the metrics used to assess the quality of the clusters. The DBI measures the average similarity ratio of each cluster with the cluster that is most similar to it. A lower DBI value indicates better clustering as it indicates well-separated and compact clusters. The Silhouette Score, on the other hand, measures how similar an object is to its own cluster compared to other clusters. A higher Silhouette Score indicates better-defined clusters.

In our analysis, we compute both the DBI and the Silhouette Score to assess the inter-class separability and intra-class compactness numerically, where the clusters represent the 7 classes. Using the features extracted from the classification models, we ensure that the clustering is based on the same features used for classification. This approach is applied to both image and text data, providing a comprehensive evaluation of the clustering performance (see Table E.1).

Table E.1: Clustering Measures

Dataset	Davies-Bouldin-Index	Silhouette Score
Dataset 1 Images	2.7042713282921897	0.09954584389925003
Dataset 1 Text	0.9090793139348567	0.3266998529434204
Textset 1	0.9865165012612194	0.3549281656742096
Textset 2	0.8861770139306265	0.3549281656742096
Dataset 2 Images	2.631281052960695	0.09031537920236588
Dataset 2 Cleaned Text	0.8322079619122981	0.38891518115997314

The resulting DBI and silhouette scores indicate that the inter-class separability and intra-class compactness are the best for the *Textset 2*, indicating a higher likelihood that the dataset is well classifiable [157, 158].

To visualize inter-class separability and intra-class compactness, we cluster the datasets using t-SNE [159] (see Figures E.10,E.11,E.12,E.13). t-SNE or t-distributed Stochastic Neighbor Embedding, is a dimensionality reduction technique that maps high-dimensional data to a lower-dimensional space, through which it maintains the relative distances between nearby data points. This makes it easier to identify clusters and patterns. This technique is used as it is generally effective for high-dimensional data such as images. We cannot take any direct measures from these clusters, but they do give a visual of the separability of the classes.

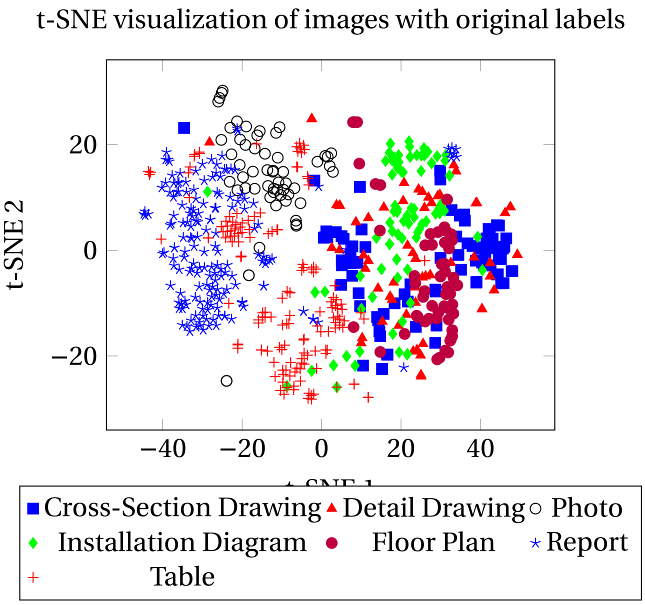


Figure E.10: t-SNE visualization of images with original labels (dataset 1)

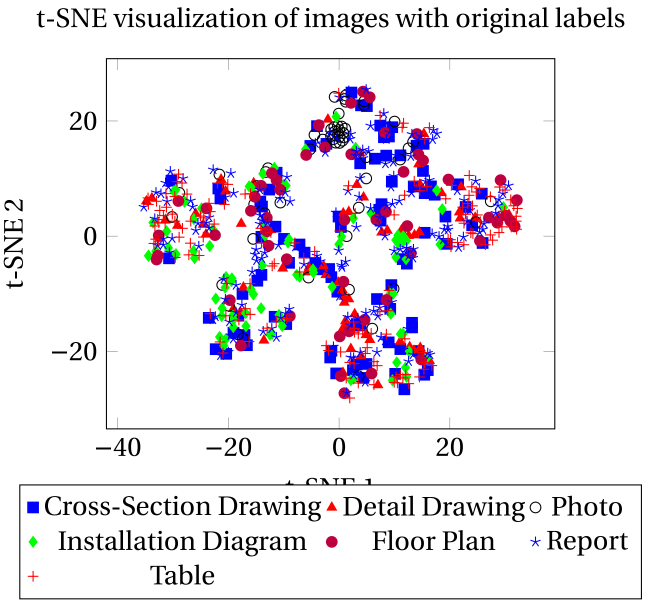


Figure E.11: t-SNE visualization of texts with original labels (dataset 1)

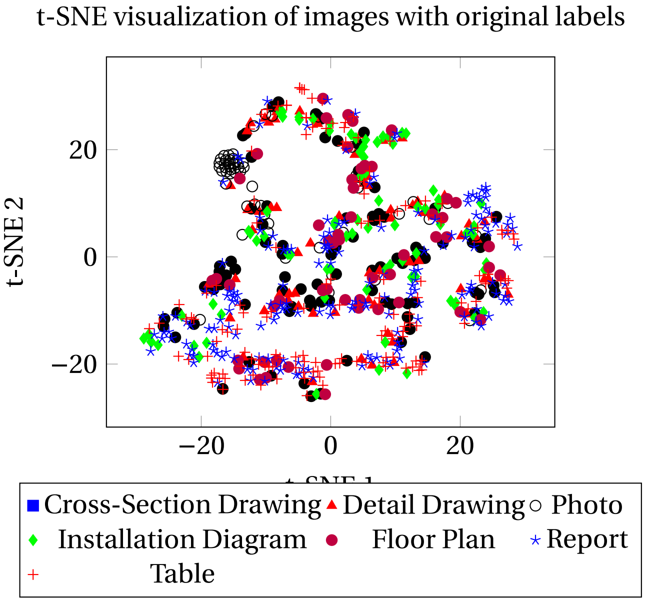


Figure E.12: t-SNE visualization of *Textset 1* with original labels (dataset 1)

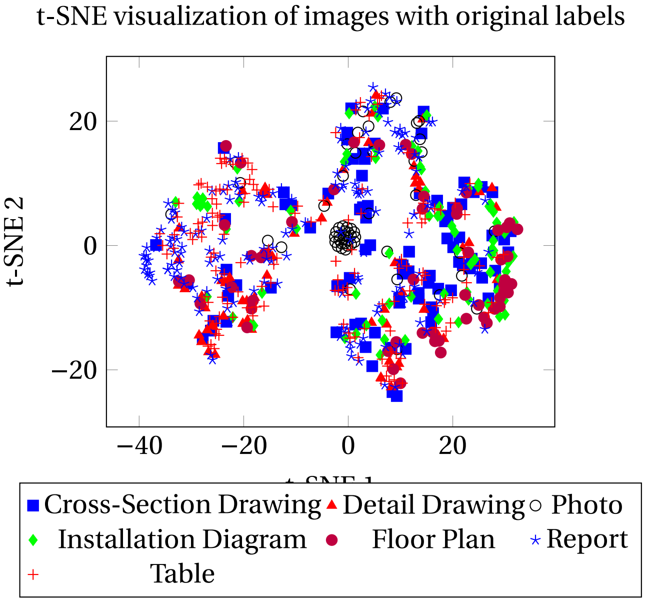


Figure E.13: t-SNE visualization of *Textset 2* with original labels (dataset 1)

F

Term Frequency - Inverse Document Frequency Classification Model Analysis

Figures F.1 and F.2 illustrate the classification performance of various classifiers using the Term Frequency - Inverse Document Frequency (TF-IDF) feature. The deep learning model consistently outperforms other models across both datasets, showing only minimal variation in performance between them. For other classifiers, the text set classification 2 generally achieves more accurate classifications, except the decision tree classifier, which performs worse on this dataset.

This shows a variation in how the models work and what they focus on. As found in Chapter 5, the tokenizer (BERT) models each achieve an improved performance on *Textset 1*, instead of on *Textset 2*, which we observe for the traditional machine learning models. This difference could indicate that the stop words that are included in *Textset 1* could be perceived as noise or distracting to the model, affecting classification performance to which these models are more vulnerable than the pretrained tokenizer models. As the simple deep learning model clearly outperforms the other models for the TF-IDF feature, this model is used further in this study, which is further detailed in Chapter 5.

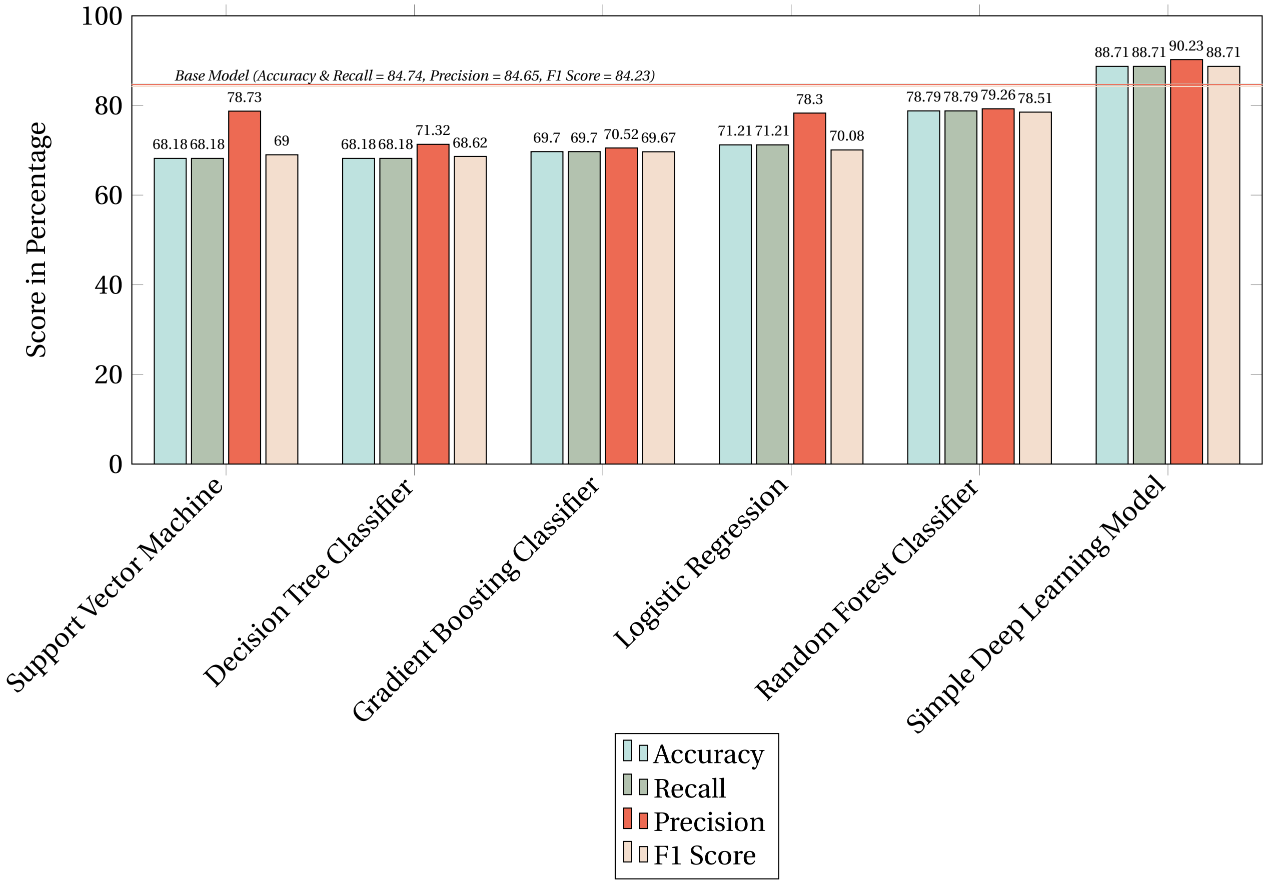


Figure F.1: Dataset 1 Classification Performances (Text Modality - *Textset 1*, TF-IDF)

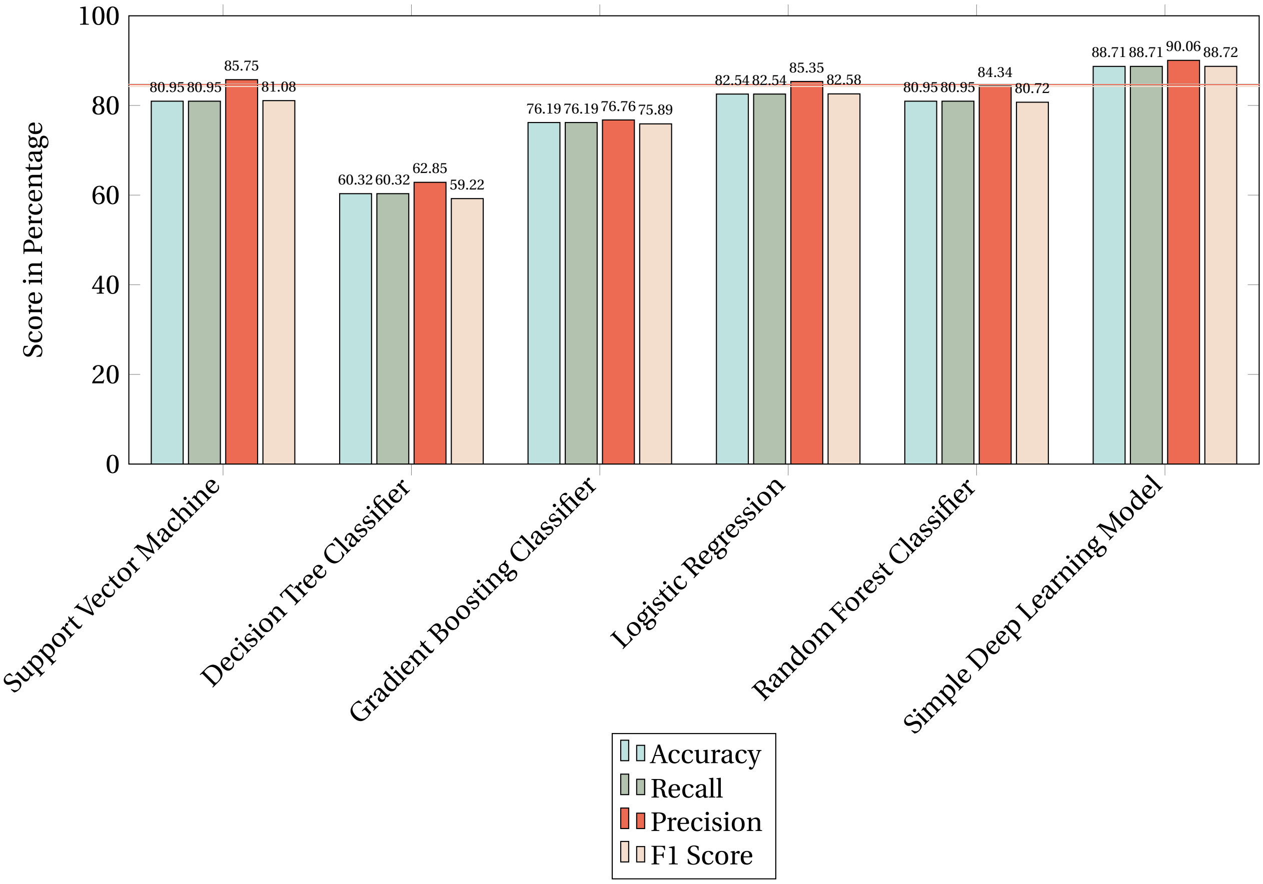
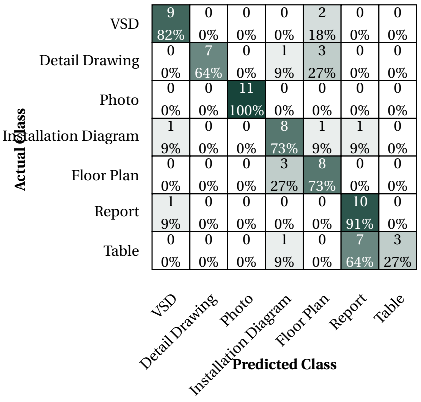


Figure F.2: Dataset 1 Classification Performances (Text Modality - *Textset 2*, TF-IDF)

G

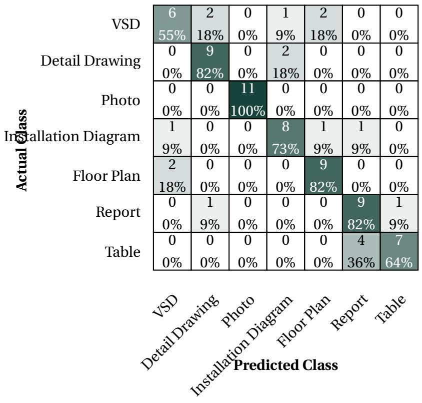
Confusion Matrices of Best Performing Models

G.1. Testing on Dataset 1



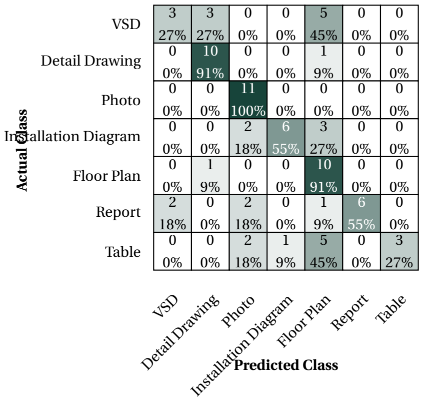
	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	9 82%	0 0%	0 0%	0 0%	2 18%	0 0%	0 0%
Detail Drawing	0 0%	7 64%	0 0%	1 9%	3 27%	0 0%	0 0%
Photo	0 0%	0 0%	11 100%	0 0%	0 0%	0 0%	0 0%
Installation Diagram	1 9%	0 0%	0 0%	8 73%	1 9%	1 9%	0 0%
Floor Plan	0 0%	0 0%	0 0%	3 27%	8 73%	0 0%	0 0%
Report	1 9%	0 0%	0 0%	0 0%	0 0%	10 91%	0 0%
Table	0 0%	0 0%	0 0%	1 9%	0 0%	7 64%	3 27%

Figure G.1: Base Model Confusion Matrix



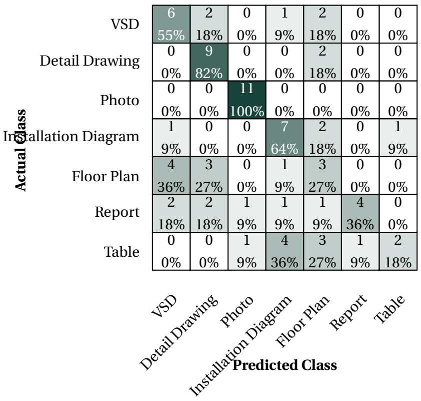
	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	6 55%	2 18%	0 0%	1 9%	2 18%	0 0%	0 0%
Detail Drawing	0 0%	9 82%	0 0%	2 18%	0 0%	0 0%	0 0%
Photo	0 0%	0 0%	11 100%	0 0%	0 0%	0 0%	0 0%
Installation Diagram	1 9%	0 0%	0 0%	8 73%	1 9%	1 9%	0 0%
Floor Plan	2 18%	0 0%	0 0%	0 0%	9 82%	0 0%	0 0%
Report	0 0%	1 9%	0 0%	0 0%	0 0%	9 82%	1 9%
Table	0 0%	0 0%	0 0%	0 0%	0 0%	4 36%	7 64%

Figure G.2: Inception-ResNet-V2 Confusion Matrix



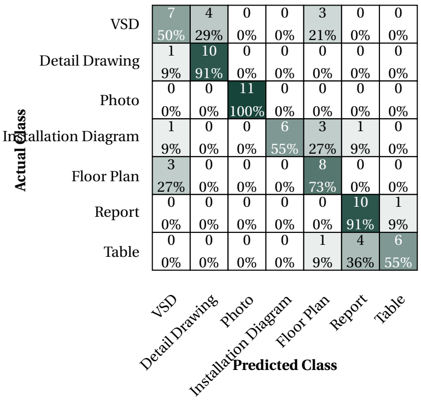
	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	3 27%	3 27%	0 0%	0 0%	5 45%	0 0%	0 0%
Detail Drawing	0 0%	10 91%	0 0%	0 0%	1 9%	0 0%	0 0%
Photo	0 0%	0 0%	11 100%	0 0%	0 0%	0 0%	0 0%
Installation Diagram	0 0%	0 0%	2 18%	6 55%	3 27%	0 0%	0 0%
Floor Plan	0 0%	1 9%	0 0%	0 0%	10 91%	0 0%	0 0%
Report	2 18%	0 0%	2 18%	0 0%	1 9%	6 55%	0 0%
Table	0 0%	0 0%	2 18%	1 9%	5 45%	0 0%	3 27%

Figure G.3: TF-IDF Confusion Matrix



	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	6 55%	2 18%	0 0%	1 9%	2 18%	0 0%	0 0%
Detail Drawing	0 0%	9 82%	0 0%	0 0%	2 18%	0 0%	0 0%
Photo	0 0%	0 0%	11 100%	0 0%	0 0%	0 0%	0 0%
Installation Diagram	1 9%	0 0%	0 0%	7 64%	2 18%	0 0%	1 9%
Floor Plan	4 36%	3 27%	0 0%	1 9%	3 27%	0 0%	0 0%
Report	2 18%	2 18%	1 9%	1 9%	1 9%	4 36%	0 0%
Table	0 0%	0 0%	1 9%	4 36%	3 27%	1 9%	2 18%

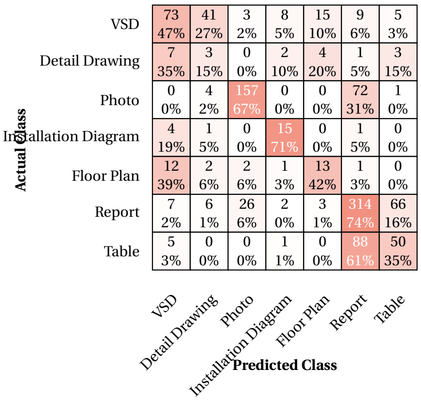
Figure G.4: Roberta Base Confusion Matrix



	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	7 50%	4 29%	0 0%	0 0%	3 21%	0 0%	0 0%
Detail Drawing	1 9%	10 91%	0 0%	0 0%	0 0%	0 0%	0 0%
Photo	0 0%	0 0%	11 100%	0 0%	0 0%	0 0%	0 0%
Installation Diagram	1 9%	0 0%	0 0%	6 55%	3 27%	1 9%	0 0%
Floor Plan	3 27%	0 0%	0 0%	0 0%	8 73%	0 0%	0 0%
Report	0 0%	0 0%	0 0%	0 0%	0 0%	10 91%	1 9%
Table	0 0%	0 0%	0 0%	0 0%	1 9%	4 36%	6 55%

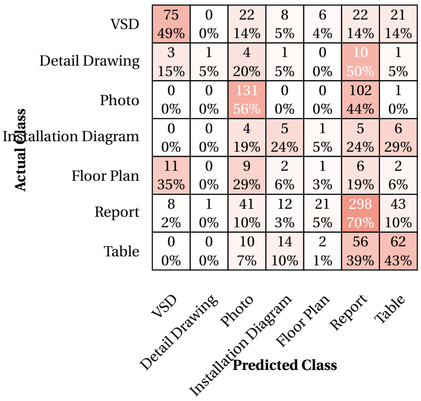
Figure G.5: LayoutLMv2 Confusion Matrix

G.2. Direct Inference on Dataset 2



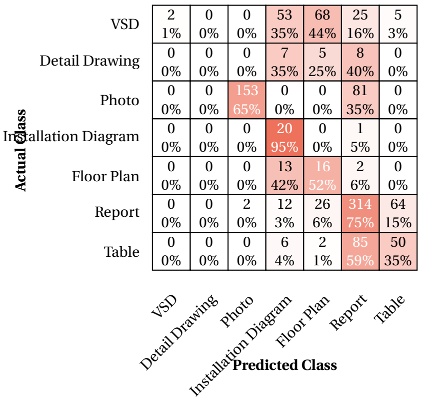
	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	73 47%	41 27%	3 2%	8 5%	15 10%	9 6%	5 3%
Detail Drawing	7 35%	3 15%	0 0%	2 10%	4 20%	1 5%	3 15%
Photo	0 0%	4 2%	157 67%	0 0%	0 0%	72 31%	1 0%
Installation Diagram	4 19%	1 5%	0 0%	15 71%	0 0%	1 5%	0 0%
Floor Plan	12 39%	2 6%	2 6%	1 3%	13 42%	1 3%	0 0%
Report	7 2%	6 1%	26 6%	2 0%	3 1%	314 74%	66 16%
Table	5 3%	0 0%	0 0%	1 1%	0 0%	88 61%	50 35%

Figure G.6: Inception-ResNet-V2 Inferencing Dataset 2 Directly Confusion Matrix



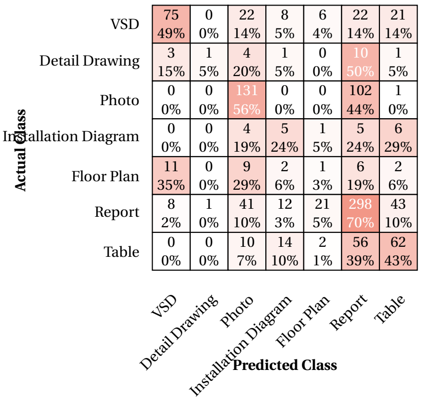
	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	75 49%	0 0%	22 14%	8 5%	6 4%	22 14%	21 14%
Detail Drawing	3 15%	1 5%	4 20%	1 5%	0 0%	10 50%	1 5%
Photo	0 0%	0 0%	131 56%	0 0%	0 0%	102 44%	1 0%
Installation Diagram	0 0%	0 0%	4 19%	5 24%	1 5%	5 24%	6 29%
Floor Plan	11 35%	0 0%	9 29%	2 6%	1 3%	6 19%	2 6%
Report	8 2%	1 0%	41 10%	12 3%	21 5%	298 70%	43 10%
Table	0 0%	0 0%	10 7%	14 10%	2 1%	56 39%	62 43%

Figure G.7: TF-IDF Inferencing Dataset 2 Directly Confusion Matrix



	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	2 1%	0 0%	0 0%	53 35%	68 44%	25 16%	5 3%
Detail Drawing	0 0%	0 0%	0 0%	7 35%	5 25%	8 40%	0 0%
Photo	0 0%	0 0%	153 65%	0 0%	0 0%	81 35%	0 0%
Installation Diagram	0 0%	0 0%	0 0%	20 95%	0 0%	1 5%	0 0%
Floor Plan	0 0%	0 0%	0 0%	13 42%	16 52%	2 6%	0 0%
Report	0 0%	0 0%	2 0%	12 3%	26 6%	314 75%	64 15%
Table	0 0%	0 0%	0 0%	6 4%	2 1%	85 59%	50 35%

Figure G.8: Roberta Base Inferencing Dataset 2 Directly Confusion Matrix



	VSD	Detail Drawing	Photo	Installation Diagram	Floor Plan	Report	Table
VSD	75 49%	0 0%	22 14%	8 5%	6 4%	22 14%	21 14%
Detail Drawing	3 15%	1 5%	4 20%	1 5%	0 0%	10 50%	1 5%
Photo	0 0%	0 0%	131 56%	0 0%	0 0%	102 44%	1 0%
Installation Diagram	0 0%	0 0%	4 19%	5 24%	1 5%	5 24%	6 29%
Floor Plan	11 35%	0 0%	9 29%	2 6%	1 3%	6 19%	2 6%
Report	8 2%	1 0%	41 10%	12 3%	21 5%	298 70%	43 10%
Table	0 0%	0 0%	10 7%	14 10%	2 1%	56 39%	62 43%

Figure G.9: LayoutLMv2 Inferencing Dataset 2 Directly Confusion Matrix