
# DOCUMENT LAYOUT ANALYSIS ON MULTI-STRUCTURED INFORMATION DOCUMENTS USING MASK R-CNN MODELS

## AN EXTENDED OBJECT DETECTION STUDY COMPARING MEAN AVERAGE PRECISIONS AND ELEMENT PRECISIONS OF MASK R-CNN MODELS ON MULTI-STRUCTURED INFORMATION DOCUMENTS REPRESENTED BY TWO DIFFERENT DATASETS

MICK HELLER

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2003152

COMMITTEE

dr. Gonzalo Nápoles
Mariana Dias Da Silva-van Riel

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

February 7, 2022

ACKNOWLEDGMENTS

I would like to give my warmest thanks to my supervisors dr. Gonzalo Nápoles and Mariana Dias Da Silva-van Riel for providing feedback and helping me during the supervisor meetings. Their guidance and advice helped me achieve this study. I would also like to thank Semmtech B.V., in particular, Bram Bazuin and Louise Dam. Bram for introducing me to the company Semmtech and giving me a warm welcome. Louise for her feedback, guidance, and fun meetings. Without her, this study would not be the way it is now. Finally, I want to thank my family and my girlfriend, for standing beside me through the whole graduation period. [1]

[1] Word count: 8639

# DOCUMENT LAYOUT ANALYSIS ON MULTI-STRUCTURED INFORMATION DOCUMENTS USING MASK R-CNN MODELS

## AN EXTENDED OBJECT DETECTION STUDY COMPARING MEAN AVERAGE PRECISIONS AND ELEMENT PRECISIONS OF MASK R-CNN MODELS ON MULTI-STRUCTURED INFORMATION DOCUMENTS REPRESENTED BY TWO DIFFERENT DATASETS

MICK HELLER

**Abstract**

Deep learning models – specifically Detectron2 Mask Region-Based Neural Networks (R-CNN) – have recently become dominant for Document Layout Analysis (DLA) and object detection tasks. However, applying layout analysis by using object detection on multi-structured documents is lacking. To narrow the gap, this study proposes a comparison of different pre-trained Mask R-CNN model variations to apply DLA and object detection on multi-structured documents by answering the following research question, which is the best performing Detectron2 Mask Region-Based Convolutional Neural Network variation for Document Layout Analysis on multi-structured information documents?. This will be done by comparing a synthetic dataset to a manually annotated dataset to determine whether such a synthetic dataset can account for the manually expensive annotation process.

The results demonstrate that the manually annotated dataset model variations have significantly outperformed the baseline model and achieved high performance on both average precisions and element precisions on multi-structured documents. The models from the synthetic dataset performed worse than the baseline method, indicating low detection powers. The reason for this could be the number of annotations, document creation, and the non-structured nature of the synthetic dataset. However, implications for future research can argue the contribution of introducing a synthetic dataset in DLA if the dataset is created more critically including more pre-determined rules for multi-structured document re-creation.

**Data Source, Code, and Ethics Statement**

- The author of this thesis acknowledges that they do not have any legal claim to part of this data or code.

## 1 INTRODUCTION

Documents in Portable Document Format (PDF) are a great source of information for humans, with over 2.5 trillion created documents worldwide (Yepes, Zhong, & Burdick, 2021). However, PDF as a format is not understandable for machines. Machines cannot understand the PDF layout and extract information using the PDF format (Zhong, Tang, & Jimeno Yepes, 2019). Data needs to be structured to be machine-readable (Open Knowledge, 2015). Structured data refers to data where the structural relation between elements is explicit in the way the data is stored on a computer disk. PDF documents contain unstructured data because the representation of PDF documents reflects the position of entities on the page and not their logical structure, which is difficult to extract automatically (Open Knowledge, 2015). However, extracting information is essential since PDF documents usually contain important information, key results, or summarizations (Zhong et al., 2019).

A potential way of automatically extracting information from PDF documents is through Document Layout Analysis (DLA). Using DLA, it is possible to annotate the physical layout structure of a document. DLA is the first process in the pipeline of a document understanding system that detects and labels homogeneous document regions (Binmakhashen & Mahmoud, 2019). The DLA process separates the document into zones, and subsequent classification of individual zones into one of the pre-defined categories such as text, tables, images, or lines (Tran et al., 2017). An important aspect of DLA is annotating the relevant elements in an image. Document annotation refers to the transformation of textual documents to linked knowledge structures that represent relevant information (Handschuh, Staab, & Maedche, 2001). It identifies fields and associated values in a text document and extracts relevant information using a set of criteria. It involves labeling and organizing data to deliver key insights and make it approachable for analysis (Handschuh & Staab, 2003). The annotation approach applied in this research involves the bounding box approach. This approach is elaborated on in section 3. In this study, the bounding box approach will be employed with the focus on a multi-model layout analysis, which combines the visual and semantic modalities of DLA (Zhang et al., 2021). This study will focus particularly on the DLA application document understanding. Document understanding refers to region detection, labeling of meaningful regions, and semantic interpretation using layout

analysis and arranging them according to a predefined domain knowledge (Rigaud, Guérin, Karatzas, Burie, & Ogier, 2015).

The provided PDF documents will be analyzed using object detection models. Object detection is a computer vision technique that tries to identify objects in an image and label them accordingly (Dasiopoulou, Mezaris, Kompatsiaris, Papastathis, & Strintzis, 2005). By doing this, object detection models draw bounding boxes around the objects of interest. These bounding boxes are later used to identify these objects. In this object detection study, images containing lists, figures, texts, titles, and tables are used as input data and objects to be detected. The outputs contain bounding boxes and class labels for every bounding box on the images. Deep learning-based object detection will be used in this research. These methods employ the R-CNN models to perform unsupervised object detection. Here, the specific features do not need to be defined and extracted separately (Deng & Liu, 2018).

Besides the dominant role of deep learning-based object detection, deep learning algorithms also became dominant in DLA applications (Binmakhashen & Mahmoud, 2019). R-CNN models are deep learning models, used specifically for object detection (K. He, Gkioxari, Dollár, & Girshick, 2017). These models have become more efficient and more accurate (Hafiz & Bhat, 2020). Recently, Facebook introduced an object detection model named 'Detectron2'. Detectron2 is a deep learning model that consists of a vast arsenal of models. Most notably, R-CNN models are used to provide the target area and classify and recognize the target on the proposal region (K. He et al., 2017). A more detailed elaboration and implementation of the models described above are given in section 3.

DLA plays an important role in terms of document understanding and in extracting relevant information from documents. The practical and societal implications are manifold, including document retrieval, content categorization, text recognition, and document understanding (Binmakhashen & Mahmoud, 2019). Also, the layout analysis performed will be done for business usage to contribute to machine readability of the PDF documents for document understanding. As mentioned, PDF documents as data are unstructured. However, A PDF document can have a structured, semi-structured, or non-structured layout (Belhadj, Belaïd, & Belaïd, 2021). A combination of multiple documents with these different layout structures can refer to multi-structured documents (Rigaud et al., 2015). In this research, multi-structured documents are PDF documents that differ in the type of PDF document (e.g. contract document or standardization document) or differ in document structure (e.g. semi-structured or unstructured). To perform DLA, the generation of data is an important

process. The input data is generated by annotating document images. High-quality data can be generated in multiple ways. In this study, both a manual process and an automatic generation process were used, resulting in an automatically generated dataset and a manually annotated dataset. These two datasets both represent multi-structured information documents and were used to train models for DLA to compare the results in terms of their ability to correctly classify different document structures. This research will contribute to the existing literature in the following ways. Many papers apply DLA on documents having similar layout structures. Section 2 will further elaborate on this point when revising the pertinent literature. Therefore, models on more diverse layout documents are needed to facilitate DLA (Binmakhashen & Mahmoud, 2019). Moreover, object detection algorithms are widely used in images with objects, instead of text-based documents including figures and tables. Lastly, the annotation process is an expensive manual process. By introducing a synthetic created dataset, this study will investigate whether such a dataset can account for the expensive annotation process by training on the different datasets and comparing the results on a manually annotated real-life test set to investigate the ability to classify the different elements in the document structures. Consequently, this research will focus on new insights in DLA using Mask R-CNN models by answering the following research question;

> *Which is the best performing Detectron2 Mask Region-Based Convolutional Neural Network variation for Document Layout Analysis on Multi-Structured Information Documents?*

To examine this research question, three sub-questions are derived and will be answered in addition to the main research question.

RQ1 *What is the performance difference of Mask R-CNN architectures on the synthetic and manually annotated datasets?*

> This research question introduces the baseline method together with its performance as a benchmark for this research. The baseline models consist of the Detectron2 model pre-trained on the PubLayNet dataset with the FPN backbone and three ResNet variations 'as is'. The model 'as is' refers to running inference on the test set without training on either of the two introduced datasets using only the pre-trained model. The PubLayNet dataset is a dataset containing over 360 thousand images of scientific articles to identify tables, texts, lists, titles, and figures (Zhong et al., 2019). A pre-trained model is a saved model that was previously trained on a larger dataset (Abadi et al., 2015). During this research, the pre-trained

models are used for transfer learning on their own dataset for better performance. Different architectures and pre-trained models are evaluated because deep neural networks depend on a wide range of hyperparameters concerning their optimization, regularization, and architecture (Hutter, Kotthoff, & Vanschoren, 2019). This research question will also introduce the proposed Mask R-CNN Detecron2 models and compare the precision results to the baseline models. The proposed models differ in architecture and trained dataset and are the Detectron2 Mask R-CNN pre-trained on PubLayNet and trained on either the synthetic dataset or the manually annotated dataset, using the FPN backbone and three ResNet variations.

RQ2 *What are the element precisions of the Detectron2 Mask R-CNN model variations for object detection?*

To evaluate the performances of the Detectron2 model variations on individual elements for object detection, element precisions will be calculated for the baseline models and proposed models. Element precision is preferred since it provides information about which elements the models can detect best for individual inference. The elements to be detected are text, titles, figures, tables, and lists.

RQ3 *To what extent can the synthetic dataset be used for inference on manually annotated real-life documents?*

The final question evaluates whether the synthetic dataset can be used for inference on real-life data. The manually annotated test set is derived from real-life documents. Hence, this test set will be used to evaluate the models and to determine whether inference is possible using the synthetic dataset. The results of the synthetic dataset will be compared to the results of the manually annotated dataset. When performance is high, it could mitigate the expensive manual process because of the automatic generation.

By answering the research questions, the goal of this research is to examine the performance of the Mask R-CNN model Detectron2 for predicting document structures. This will be done by performing object detection and DLA on a synthetic created and manually annotated dataset, representing formal multi-structured information documents. Furthermore, a comparison between the results of the two datasets will be made to examine whether the synthetic dataset can be used as training data to account for the expensive manual process.

This study demonstrates that Detectron2 Mask R-CNN variations trained on the manually annotated dataset have high detection precision on multi-structured documents. The performance of the models trained on the synthetic dataset was lower compared to the baseline and manually annotated dataset, indicating the difficulty of the multi-structured layout analysis. Model variations of the baseline and synthetic datasets were not able to detect all elements. Considering the manually expensive annotation method and the limiting annotated element, future research regarding multi-structured documents will be argued for.

This paper is structured in the following manner. In section 2, the related work on DLA, Object Detection, and Detectron2 will be discussed. Section 3 gives insights into the method of the research to apply DLA on multi-structured documents. The methods section focuses on the composition of the Detectron2 model and the Mask R-CNN algorithm. Section 4 focuses on the experimental setup of the research, which includes the data generation phase. Section 5 shows the results of the research whereas sections 6 and 7 focus on the discussion and conclusion consequently.

## 2 RELATED WORK

This related work section presents previous work from the field of DLA, document annotation, deep learning, and object detection. It will discuss the limitations of these studies, methods used in this research, and how the limitations can be accounted for in this research.

### 2.1 *Document Layout Analysis*

DLA aims to divide documents images into different region types. It plays an important role in document understanding and information extraction from documents (X. Wu, Ma, Li, Chen, & He, 2021). The purpose of DLA is to extract valuable information from the different document images (X. Wu et al., 2021). Many previous studies in DLA have used the uni-modal layout analysis, which only focuses on either visual features or semantic features for document understanding (Zhang et al., 2021). For example, studies of Gatos, Louloudis, and Stamatopoulos (2014), D. He, Cohen, Price, Kifer, and Giles (2017) have used the visual features to segment text paragraphs, figures, and tables. Zhang et al. (2021) proposes the multi-model layout analysis, a method combining both uni-model layout analyses (meaning; both visual features and semantic features). Such a method contributes to better recognition of the document layout, because of the combined information (Zhang et al., 2021). During this research, the proposed multi-

model method will be used. By introducing this multi-model method, the limitations of the above-mentioned studies – which only use a uni-model method – will be disclosed.

Typically, documents fall into three main categories: structured documents, semi-structured documents, and unstructured documents (Belhadj et al., 2021). Structure documents have the same layout, where elements are always positioned in the same place on the page. Semi-structured documents contain data and fields of data that vary between documents. Lastly, unstructured documents contain information embedded in texts. Many previous studies focused on specific structures, mostly structured or semi-structured, and types of documents. For example, Yepes et al. (2021) focused on structured multi-column research papers, Binmakhashen and Mahmoud (2019) researched handwritten historical manuscripts, and Rigaud et al. (2015) focused on structured comic books.

DLA can have two tasks – the physical and the logical layout analysis. The physical layout analysis is used to detect the document structure and identify boundaries of its homogeneous regions (Binmakhashen & Mahmoud, 2019). Meanwhile, the logical layout analysis labels these regions in different elements like tables, text, or titles (Binmakhashen & Mahmoud, 2019). Previous studies have focused on either one of these tasks. Pletschacher and Antonacopoulos (2010) and Antonacopoulos, Pletschacher, Bridson, and Papadopoulos (2009) focused solely on page segmentation. Hence, combining both logical analysis and physical layout analysis can improve overall DLA performance (Binmakhashen & Mahmoud, 2019).

## 2.2 *Document Annotations*

Besides different analyses, datasets need to be annotated before being usable for a model. Annotating is usually conducted using three major modes; fully automatic, semi-automatic, and manual (Trivedi & Sarvadevabhatla, 2021). Manually annotating documents is a slow and expensive process, which can be a limiting factor when willing to use these specific DLA techniques in different domains (Yepes et al., 2021). According to K. He et al. (2017), fully automated annotation works well with structured printed documents and structured handwritten documents, but it is less accurate predicting highly unstructured documents (Hafiz & Bhat, 2020). Many DLA datasets rely on manual annotations. Since the process is slow and expensive, these datasets are of limited size (Pletschacher & Antonacopoulos, 2010). In Yepes et al. (2021)'s study, the dataset is generated with millions of automated annotations used by matching the XML format with the PDF document. It is a method that can lead to great results, but it is

very time-consuming (Yepes et al., 2021). Limitations of the previous study are that in the Yepes et al. (2021) study, the same structured documents are used, which can make the model less versatile while using multi-structured documents. Synthetic data is an important approach to solving the data problem imposed by the manual expensive process (Nikolenko, 2019). This is done by producing diverse artificial data. According to Nikolenko (2019), the synthetic data approach is exemplified by standard computer visions, defined as a high-level understanding of images by computers (Freeman, Tanaka, Ohta, & Kyuma, 1996). However, it can also be relevant in other, more complex, computer vision domains.

In sum, the main limitations concerning the previous studies regarding DLA consist of the lacking research direction to multi-structured PDF documents, multi-modal layout analysis, and the possible solution of synthetic data in DLA.

## 2.3 *Deep Learning and Object Detection*

Before the introduction of deep learning algorithms, techniques based on image representation and optical character recognition were first used for understanding documents (Zhong et al., 2019). However, deep learning algorithms became dominant in the last few years. This is because of more powerful algorithms, infrastructures, and methods as well as unsupervised learning, where features do not need to be defined and extracted separately (Deng & Liu, 2018). Deep learning can improve the classification performance and robustness (Deng & Liu, 2018) for DLA. Consequently, deep learning-based DLA can address more complex layout analysis for PDF documents (Binmakhashen & Mahmoud, 2019). The R-CNN frameworks are important deep learning networks for object detection (Hafiz & Bhat, 2020). By the broad adaptation of deep learning, object detection performance grew immensely (Hafiz & Bhat, 2020).

Mask R-CNN, a region-based convolutional neural network and state-of-the-art algorithm for object detection, is one of the latest introduced R-CNN models (K. He et al., 2017). It avoids the heavy CNN computations of the Faster R-CNN and is one of the most successful techniques for object detection (K. He et al., 2017). Therefore, it will be used throughout this research as a benchmark, but also as multiple proposed models. Although Mask-RCNN is commonly used in object detection for images, research is lacking regarding PDF text documents. Many previous studies apply object detection on COCO dataset images. COCO is a large-scale object detection dataset, including 80 object categories, 330 thousand images, and 1.5 million object instances (Lin et al., 2014). For example, studies like Kirillov, Wu, He, and Girshick (2020) and Bolya, Zhou, Xiao, and Lee (2019)

focus on object detection on pictures similar to the COCO dataset. The goal is to advance the state of the art in object detection by broadening the scene understanding by gathering images of complex everyday scenes containing common objects Lin et al. (2014). However, if studies perform object detection on text-based documents, the focus will either be on structured documents or handwritten text.

Therefore, it is interesting to research DLA with an extensive amount of synthetic data which can possibly account for the manual annotation process. By using the pre-trained model of Yepes et al. (2021) and including an extra dataset, this study tries to compare the results of a smaller manually annotated dataset with the synthetic created dataset. The above-mentioned studies did not work with a synthetically created dataset as an option to account for manual annotation. It is therefore interesting to see whether such a synthetic dataset can be used as training input for DLA to mimic multi-structured PDF documents. Also, object detection for DLA is a research direction where many previous studies focus on images containing real objects (COCO dataset) instead of PDF documents. Hence, document understanding using object detection can give interesting results and will therefore be researched in this work.

## 3 METHODS

This chapter describes the different models used for object detection applied to document layout analysis. First, a description will be given of the method object detection including its steps. Second, an explanation of the contents of the relevant Detectron2 models and architectures, as well as their application in this study will be provided. Besides explaining the Detectron2 model, the focus will also be on illustrating the Mask R-CNN model and the Faster R-CNN model. The former is used in this research, the latter is its predecessor and an explanation of both is needed.

R-CNN models are machine learning models which can be used for computer vision, specifically for object detection (K. He et al., 2017). These models have rapidly improved results over short periods of time. These fast improvements have led to powerful baseline systems, such as the faster R-CNN models. The R-CNN models are useful because of their fast training and inference time, robustness, and flexibility (K. He et al., 2017).

### 3.1 *Object Detection Steps*

Object detection localizes all the objects present in an image using a predetermined model. Multiple important constructs related to object detection

need to be elaborated to provide the reader with a solid understanding of object detection in this research. These constructs are; bounding boxes, anchor boxes, the intersection over union, and the non-max suppression. The bounding box is a rectangle used to enclose the object in the image and is described by the following values: *(bx, by, bh, and bw)*. Below in figure 1, examples of the bounding boxes are presented. Here, the green bounding boxes surround titles, whereas the red bounding boxes surround text elements.

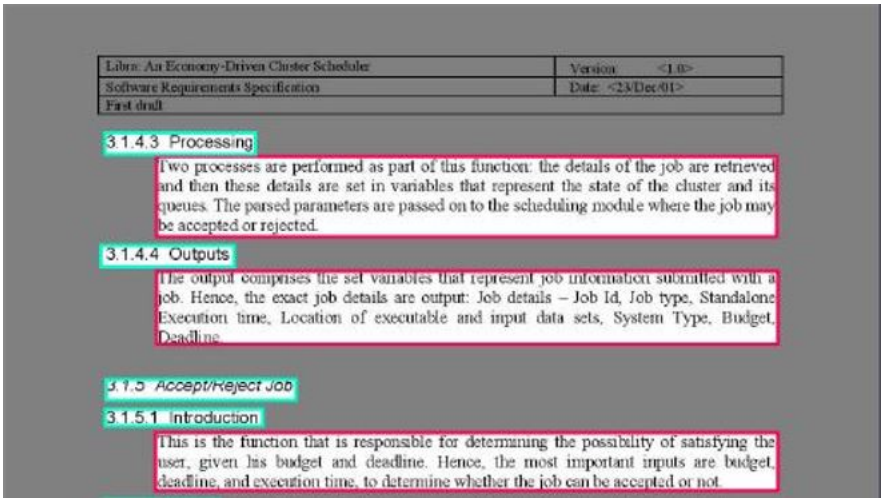


Figure 1: Bounding Box example used in this research

Anchor boxes are predefined bounding boxes. During the detection phase, these anchor boxes are tiled across the image. After identifying the bounding boxes and anchor boxes, the Intersection of Union (IoU) is calculated. The IoU is an evaluation metric for the prediction of the bounding box with respect to the ground truth. An elaboration of the IoU as a metric can be found in section 4 experimental setup. For object detection, detection can be represented by multiple boxes. The non-max suppression gives the box that has the highest IoU and discards the other boxes.

## 3.2 *Detectron2*

The model selected for object detection in this research is Detectron2. It is a Facebook AI Research (FAIR) software system that implements state-of-the-art object detection algorithms, including Faster R-CNN, Mask R-CNN, Cascade R-CNN, RetinaNet, Densepose, and TensorMask (A. M. F. L. W. G. R. Wu Y.; Kirillov, 2019). Detectron2 is a newer version of Detectron, and it is implemented in Pytorch with a more modular design (A. M. F. L. W. G. R. Wu

Y.; Kirillov, 2019). To provide a clear understanding of the used models in this research, the following models relevant to object detection will be explained; Faster R-CNN, Mask R-CNN, Panoptic Future Pyramid Network (FPN), and ResNet. The Detectron2 models will be distinguished using a pre-trained model and different backbones. The pre-trained model used in this study is trained on the PubLayNet dataset. According to K. He et al. (2017), a backbone is a standard CNN model. In this study, the ResNet and FPN backbone will be used with different layer depths for the ResNet backbone.

### 3.2.1 *Faster R-CNN*

Before the main model will be introduced, an explanation of the Faster R-CNN model is given. The Faster R-CNN model is a network introduced as a computationally efficient solution for object detection and the precursor of the Mask R-CNN model (Ren, He, Girshick, and Sun (2015)). A Faster R-CNN consists of two stages. The first stage is a deep fully CNN that proposes regions, more specifically a region proposal network (RPN) (Ren et al., 2015). The second stage extracts features from each bounding box and uses bounding-box regression and classification (K. He et al., 2017). The RPN takes an image as an input value and generates the proposal for the objects as outputs (Ren et al., 2015).

### 3.2.2 *Mask R-CNN*

A Mask R-CNN model consists of three outputs for each object; a class label, bounding box offset, and the object Mask (K. He et al., 2017). Mask R-CNN is also a CNN specified for object detection (K. He et al., 2017). It is developed on top of the Faster R-CNN model. Therefore, it has the same two stages as the Faster R-CNN model. Mask R-CNN is an extension of Faster R-CNN in the way that it adds the prediction of an object mask, which is the region of interest, simultaneously with the bounding box recognition already present in the Faster R-CNN model. Both stages described above are connected to a backbone. The RPN generates region proposals for every image. Each proposal (ROIs) goes through the object detection and Mask prediction network.

### 3.2.3 *Panoptic FPN*

A Panoptic FPN is an extension of an FPN that can generate object detection through FPN (Kirillov, Girshick, He, & Dollár, 2019). Starting from the deepest FPN level, three upsampling stages are performed to yield a feature map at scale. This strategy is repeated for every FPN scale, with progressively fewer upsampling stages. The result is a set of feature maps

at the 1/4 scale, which are then element-wise summed. A final convolution, bilinear upsampling, and softmax are used to generate the per-pixel class labels at the original image resolution. In addition to stuff classes, this branch also outputs a special other class for all pixels belonging to objects to avoid predicting stuff classes for such pixels (Kirillov et al., 2019). The backbone network extracts the feature maps from the input image.

#### 3.2.4 *ResNet*

In this study, multiple ResNets will be used as a backbone for comparison and evaluation. These are ResNet50, ResNet101, and ResNeXt101. ResNets are residual networks that learn (residual) functions with specified input layers. After doing so, these residual 'nets' fit the layers in a residual map (K. He, Zhang, Ren, & Sun, 2016). ResNets then stack these residual blocks to form their network. The number in the ResNet variation stands for the depth of the layers (ResNet50 has a depth of 50 layers). According to K. He et al. (2016), ResNets are easier to optimize and gain accuracy from their increased depth.

## 4 EXPERIMENTAL SETUP

This section describes the dataset generation and steps to perform DLA for model comparison between the datasets using the Detectron2 Mask R-CNN model variations. This section is composed of two subsections. First, the data will be explained by introducing the datasets and the data preparation steps. Second, the experimental procedure will be presented, comprising of the development of the models, the parameter tuning, implementation, evaluation metrics, and the software used to realize this research.

### 4.1 *Data*

In this subsection, the data used in this research will be explained. It will discuss the creation, explanation, and preparation of the datasets. This research uses multiple different PDF documents, which were provided by Semmtech B.V. The PDF documents consist of different types of documents and are all semi- or unstructured, thus resulting in multi-structured data. The input data consists of mostly contract documents, requirements documents, and standardization documents. Examples of these documents are 'Rules and Regulations for the Classification of Ships' and 'OpenSG EIM System Requirements Specification'. To correctly use the PDF documents as input data for the models, two COCO format datasets were created; A synthetic generated dataset and a manually annotated dataset.

### 4.1.1 *COCO JSON Format*

The COCO format is necessary as an input format for the Mask R-CNN Detectron2 models to perform object detection. COCO format is a specific JSON structure dictating how labels and metadata are saved for an image dataset (Lin et al., 2014). It defines how annotations and metadata are stored. The datasets are formatted in JSON and are a collection of multiple inputs; "info", "licenses", "images", "annotations", "categories", and "segment info". The info section contains information about the datasets. Since the datasets are self-generated, the "info" sections are structured following the available information. The "licenses" sections contain information about the image licenses needed when sharing or selling the data. Since the documents are open source, licensing information is not needed. The "images" section contains all the images and the information about each image used in this research. In this file, there are no labels, bounding boxes, or segmentations specified. In figure 2 below, two COCO JSON formats for separate images can be seen presenting the differences in an image file.


Figure 2: Two COCO JSON instances for seperate images

Every image ID is a unique value. In the datasets, the image IDs are identical to the file names, as can be seen in figure 2. The height and width of the images are set to 1100 and 850 respectively. The category section contains the list of categories and super categories. Due to simplicity, these datasets consist of five super categories and identical categories (text, title, figure, list, and table). For example, "super-category"; list, "category"; list. The last section in the COCO format is the annotation section. The annotation section contains the list of every object annotated in an image. It contains "segmentation", "area", "iscrowd", "imageID", "bbox", "categoryID", and "annotationID". In this research, "segmentation" refers to a list of polygon vertices around the object. The area is measured in pixels. "Iscrowd" refers to whether segmentation is done for a single object or a cluster of objects. In this research "iscrowd" is always set to 0. The "imageID" is the unique id of the image. The "bbox" is the bounding box format. This can be represented as top-left x position, top-left y position, width, and height. The "categoryID" refers to the specific category. Lastly, each annotation has a unique "annotionID".

## 4.2 *Dataset Generation*

This study compares two different datasets. The first dataset is manually annotated, representing real-world PDF documents. The manually annotated dataset is created using Labelbox for image annotations. PDF documents are transformed into images before processing. After the annotation process, the JSON file was converted into the COCO JSON format using Roboflow.

The second dataset is a larger synthetically created dataset established to mimic the multi-structured documents. The synthetic dataset is generated in the following way. Using Gimp software, 7 background images of PDF files were extracted. Next, using Gimp software, 50 titles, 50 texts, 50 tables, 50 figures, and 50 lists including their bounding boxes were created as foregrounds. To generate the images, one of the 7 backgrounds is randomly chosen. Following the background, a minimum of 1 to a maximum of 2 foregrounds are randomly placed over the background, mimicking a PDF file and minimizing the foregrounds overlapping. Slight alterations in size and brightness are created to increase the number of unique instances. The test set of the manually annotated dataset will be used to compare the results of the two datasets.

In table 1, the number of elements of the manually annotated dataset differentiating between training, validation, and test set is presented. It consists of 460 training, 58 validation, and 58 test images including a total of 2916 annotations. The train, validation, test set split used in this research for the manually annotated dataset is 80, 10, 10 percent.

Table 1: Element count for the training, validation, and test set of the manually annotated dataset

| Category | instances train set | instances validation set | instances test set |
|---|---|---|---|
| Table | 250 | 34 | 24 |
| Figure | 140 | 12 | 15 |
| Title | 957 | 124 | 129 |
| Text | 1258 | 150 | 164 |
| List | 311 | 51 | 31 |
| Total | 2916 | 371 | 363 |

Table 2 presents an overview of the synthetic dataset. The synthetic dataset consists of 8000 training and 2000 validation images that are generated. In total, the synthetic dataset has 11958 annotations. In table 2 below, the number of texts, tables, titles, figures, and lists used in the generated training and validation set can be seen. The tables show that

the distribution of the different categories is in balance in the training and validation set. The split for the synthetic dataset is 80, 20 percent.

Table 2: Element count for the training and validation set of the synthetic dataset

| Category | Instances train set | Instances validation set |
|---|---|---|
| Table | 2432 | 610 |
| Figure | 2303 | 587 |
| Title | 2377 | 567 |
| Text | 2403 | 605 |
| List | 2443 | 638 |
| Total | 11958 | 3006 |

Below, two pie charts are presented illustrating the total element distribution of the two datasets. The left pie represents the manually annotated distribution and the right pie represents the synthetic distribution. Noticeable is the difference between the distributions. The synthetic dataset has an even distribution of all elements. Resulting from the random generation and the equal number of annotated elements. The manually annotated dataset has a more real-life distribution based on PDF document contents. Normally, more titles and texts elements can be found in a document contrary to figures, lists, or tables. Hence, the distribution difference.

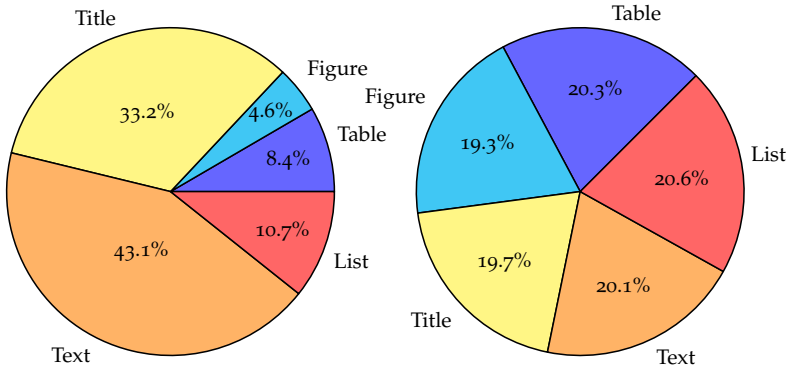


Figure 3: Element distribution manually annotated dataset (left) and synthetic dataset (right)

### 4.2.1 *Annotation*

To generate the datasets, five different elements had to be annotated to create input data for the models. These annotations were done following the method of Li et al. (2020), and Zhong et al. (2019). Although these annotations were done through the XML format, this research manually

annotated elements following the same method. The annotation process for the synthetic dataset was done through foregrounds and backgrounds, whereas the manually annotated dataset was generated using manual annotations. The five elements were annotated using their bounding boxes. This has resulted in the bounding box surrounding the whole element of interest through their maximum points on the image.

### 4.2.2 *Algorithms*

The Mask R-CNN Detectron2 base model as-is will be used as a benchmark model. These include three ResNet variations. The Mask R-CNN Detectron2 model pre-trained on the PubLayNet dataset, after hyperparameter tuning, trained on both the synthetic and manually annotated dataset will be used as the proposed models. These models will include the three ResNet varieties. For comparison, the different architectures of both models are explained. The Mask R-CNN Detectron2 model is evaluated in six ways. The first three evaluations were pre-trained on the PubLayNet and trained using the synthetic dataset using the ResNet50 FPN, ResNet101 FPN, and ResNeXt101x FPN as the backbone. The other three Mask R-CNN Detectron2 models were pre-trained on PubLayNet and trained using the manually annotated dataset including the ResNet50 FPN, ResNet101 FPN, and ResNeXt101 FPN backbones. All six models are tested on the manually annotated test set. This test set represents real-life PDF documents.

### 4.2.3 *Software*

The programming language used in this research is Python 3.9.2 by using Anaconda Navigator and Google Colaboratory. The models have been implemented using PyTorch, Keras, and Tensorflow. During the implementation of the models, the newest version of both Keras and Tensorflow was used. For the data set generation, Visual Studio Code was used whereas Gimp Software, Labelbox, and Roboflow were used for bounding box annotations and JSON conversions. The following libraries and packages have also been used;

- Numpy
- Flask
- Python-dotenv
- flask-wtf
- Opencv-python

- Pillow
- Flask-cors
- Pyyaml
- Gunicorn
- Pdf2image

#### 4.2.4 *Parameter Tuning*

To achieve the highest performance of the models, the correct hyperparameters had to be selected. In this research, multiple parameters specific for Mark R-CNN models have been optimized. These were:

- Intersection over Union

  According to K. He et al. (2017) and Girshick (2015), the RoI is positive when the IoU is at least 0.5. Hence, when the IoU is 0.5, the inference will be correct. Therefore, an IoU of 0.5 will be used.

- Train ROIs per image

  The train ROI per image is the maximum number of ROI the RPN will generate for the image. The image per batch size is set to 4, which is in accordance with K. He et al. (2017), Girshick (2015), and Ren et al. (2015), and the RoI batch size per image is set as default (512).

- Backbone

  The backbone of choice which will be used is the FPN backbone. According to the Detectron2 Model Zoo, the FPN Backbones gives the highest Bounding box average precisions. Hence, the FPN will be used. To compare the performance of different layers, multiple ResNet Backbones will be used. These are ResNet50, ResNet101, and ResNeXt101.

- Loss Weights

  The Mask R-CNN loss function was calculated as the weighted sum of different losses. The loss weights used in this research include the bounding box losses; RPN Bounding Box loss and Mask R-CNN Bounding Box loss. The RPN Bounding Box loss corresponds to the localization accuracy of the RPN. This was the weight to tune in case the object was being detected but the bounding box should be corrected. The Mask R-CNN Bounding Box loss was the loss, assigned on the localization of the bounding box of the identified class.

- Iterations

  The maximum iterations are set to 2000. The validation precision started to increase around iteration 2000. It was also the value where the validation losses mentioned above were at their lowest point or started stagnating. The number of iterations was retrieved by a manual optimization loop. The number of iterations was chosen resulting in the lowest loss weights on the validation set and when the validation precision started to rise.

- Learning Rate

  Training will be done with a learning rate of 0.001, following the Detectron2 pre-trained setup. According to K. He et al. (2017), Girshick (2015), and Ren et al. (2015), this learning rate achieved good training results. Implementing a higher learning rate decreased training precision and gave inaccurate bounding box precision, where the training diverged. Therefore, a learning rate of 0.001 was applied.

### 4.2.5 *Evaluation Metrics*

The last step in the experimental process will be comparing the evaluation results of the model variations with each other and to the baseline. To compare the Detectron2 models with the Baseline model, precision will be used. Eq. (1) shows the calculation of precision by the true positives divided by the sum of the true positives and false positives. Precision will be used because it quantifies the number of positive class predictions belonging to the positive class.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

Specifically, the mean average precision (mAP) will be used. Eq. (2) shows the mAP calculation over i classes. Here, C is the number of classes evaluated and the APi is the average precision for the ith class. The mAP quantifies the accuracy of object detectors. Here, The average precision for an image means the precision averaged over all instances of objects presented in the image. The mAP is the average precision averaged over the IOU of 0.5 to 0.95 with a step size of 0.05 and expressed as a percentage.

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i \tag{2}$$

Besides the mAP, the element average precisions (AP) will also be used to evaluate the performance of the models on individual elements. Eq. (3) presents the AP over different individual elements. The average precision

is calculated by finding the area under the precision-recall curve. The integral is between 0 and 1 because precision has always a value between those numbers.

$$AP = \int_0^1 p(r)dr \tag{3}$$

The following precision metrics will be used to evaluate the models; mAP, mAP50 (mean average precision with an IoU of 50 percent), mAP75 (mean average precision with an IoU of 75 percent), mAPs (mean average precision for instances smaller than $32^2$), mAPm (mean average precision for medium instances between $32^2$ and $96^2$), mAPl (mean average precision for large instances bigger than $96^2$), and the element precision. These precisions are connected to the IoU. For object detection, the IoU is equal to the overlapping area between the ground truth and predicted bounding box (Padilla, Passos, Dias, Netto, & da Silva, 2021). A perfect overlap results in an IoU of one, whereas IoU of zero represents no overlap. For precision evaluation, the IoU threshold will be set to 0.5 to classify whether the prediction is a true positive (one) or false positive (zero). Eq. (4) shows the IoU equation. Which is composed of the Area of Overlap divided by the Area of Union.

$$IntersectionoverUnion = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{4}$$

Because of the importance of the IoU and precision, the mAP with IoU of 0.5 (mAP50) will be the most valuable metric evaluated in this study.

## 5 RESULTS

The results section is organized in the following way. First, the results of the baseline models will be presented. Following these results, the average precision metrics and element precisions of the Detectron2 architectures trained on the synthetic dataset will be presented. Lastly, the results of the Detectron2 architectures trained on the manually annotated dataset will be explained. The models that were evaluated 'as is' on the baseline, the synthetic dataset, and manually annotated dataset are listed below;

- Mask R-CNN Detectron2 pre-trained on PubLayNet dataset using the ResNet50 FPN backbone architecture
- Mask R-CNN Detectron2 pre-trained on PubLayNet dataset using the ResNet101 FPN backbone architecture
- Mask R-CNN Detectron2 pre-trained on PubLaynet dataset using the ResNeXt101 FPN backbone architecture

As mentioned in section 4, the results were evaluated using multiple bounding box mean average precision metrics. These metrics were; mAP, mAP50, mAP75, mAPs, mAPm, mAPl, and element precision.

## 5.1 *Mask R-CNN Baseline Measures*

The baseline models evaluated in this study were the pre-trained PubLayNet Mask R-CNN Detectron2 model 'as is' with the ResNet50, ResNet101, and ResNeXt101 architectures. The different bounding box mAP can be seen in Table 3 below. As can be seen in the table, the mAPs and mAPm both resulted in 0.000 with every ResNet variation, indicating that they are not able to detect elements of smaller and medium sizes. The highest precisions were generated by the ResNeXt101 variation with a mAP50 of 25.131. The lowest results were generated by the ResNet101 variation with a mAP50 of 15.192.

Table 3: Mean average precisions across the baseline model

| Architecture | mAP | mAP50 | mAP75 | mAPs | mAPm | mAPl |
|---|---|---|---|---|---|---|
| PubLayNet ResNeXt50 FPN | 16.441 | 20.614 | 17.060 | 0.000 | 0.000 | 16.484 |
| PubLayNet ResNet101 FPN | 12.340 | 15.192 | 12.733 | 0.000 | 0.000 | 12.340 |
| PubLayNet ResNeXt101 FPN | **19.305** | **25.13** | **19.996** | 0.000 | 0.000 | **19.482** |

The element precisions will also be evaluated to determine the detection power of the individual elements. In table 4, the elements precisions of the baseline model variations are presented. Similar to the precisions above, some elements could not be detected. Here, all models had difficulty detecting texts, titles, and figures, with an element precision of 0.000. All models were able to correctly detect tables, where the ResNeXt101 variation reported the highest element precision of 74.900. The models were also able to detect lists. Again, the ResNeXt101 variation had the highest element precision with 21.624. Just as in table 3, the ResNet101 variation had the lowest precisions contrary to the other two models with element precisions of 48.320 and 13.380 for tables and lists respectively.

## 5.2 *Average Precisions Detectron2 Architectures using the Synthetic Dataset*

In this subsection, the results of object detection on the synthetic dataset are reported. Table 5 shows the bounding box precisions of the baseline model and the synthetic dataset. It shows that the results on the synthetic dataset were lower compared to the baseline model with all variations. As can be seen in table 5, all models had a mAPs of 0.000, whereas the synthetic model had a slightly better precision on the mAPm. The best performing

Table 4: Bounding box element precisions across the baseline model

| Category | Architecture | Bounding Box Element Precision |
|---|---|---|
| Text | PubLay NetResNet50 FPN | 0.000 |
| | PubLayNet ResNet101 FPN | 0.000 |
| | PubLayNetResNeXt101 FPN | 0.000 |
| Table | PubLayNetResNet50 FPN | 63.621 |
| | PubLayNetResNet101 FPN | 48.320 |
| | PubLayNetResNet101x FPN | **74.900** |
| Title | PubLayNetResNet50 FPN | 0.000 |
| | PubLayNetResNet101 FPN | 0.000 |
| | PubLayNetResNeXt101 FPN | 0.000 |
| Figure | PubLayNetResNet50 FPN | 0.000 |
| | PubLayNetResNet101 FPN | 0.000 |
| | PubLaynetResNeXt101 FPN | 0.000 |
| List | PubLayNetResNet50 FPN | 18.584 |
| | PubLayNetResNet101 FPN | 13.380 |
| | PubLayNetResNeXt101 FPN | **21.624** |

model variation on the synthetic dataset is the PubLayNet ResNeXt101 with a mAP50 of 9.304. Although this is the best variation on the synthetic dataset, all baseline variations achieve higher scores on the test set. The worst-performing baseline model outperformed the best-performing model on the synthetic dataset by 3.036 percent. Surprisingly, the ResNet101 was the overall worst performing model in both the baseline and synthetic dataset, outperformed by its ResNet50 and ResNeXt101 counterparts.

Table 5: Mean average precisions across the baseline and improved models

| Architecture | mAP | mAP50 | mAP75 | mAPs | mAPm | mAPl |
|---|---|---|---|---|---|---|
| PubLayNet ResNet50 FPN Baseline | 16.441 | 20.614 | 17.060 | 0.000 | 0.000 | 16.484 |
| PubLayNet ResNet101 FPN Baseline | 12.340 | 15.192 | 12.733 | 0.000 | 0.000 | 12.340 |
| PubLayNet ResNeXt101 FPN Baseline | **19.305** | **25.131** | **19.996** | 0.000 | 0.000 | **19.482** |
| PubLeyNet ResNet50 FPN Synthetic | 7.592 | 10.421 | 7.691 | 0.000 | **0.973** | 7.840 |
| PubLayNet ResNet101 FPN Synthetic | 7.769 | 9.732 | 9.033 | 0.000 | 0.337 | 7.891 |
| PubLayNet ResNeXt101 FPN Synthetic | **9.304** | **10.448** | **10.203** | 0.000 | 0.020 | **9.236** |

## 5.3 *Object Detection using Detectron2 Mask R-CNN comparing Element Precisions using the Synthetic Dataset*

This subsection contains the results for object detection for the elements trained on the synthetic dataset. As can be seen in table 6, each result is listed per element and per architecture. After inspecting the table, it was notable that not all elements could be detected evenly well. For all

three models, titles could not be detected at all, having a precision of 0.000. Contrary to the baseline measures, all three models were able to detect texts and figures, ranging from 0.022 to 1.408 for text and 0.139 to 2.002 for figures. For both elements, the PubLayNet ResNet50 had the highest precision with 1.408 percent and 2.002 percent respectively. Surprisingly, the PubLayNet ResNet101 model does not outperform its ResNet50 counterpart having less accuracy on all elements except for tables. The PubLayNet ResNeXt101 achieved the highest precision for detecting tables, with a precision of 45.541 percent. The models were able to detect lists almost with the same precision, ranging from 0.820 to 2.091, whereas the PubLayNet ResNet50 model had the highest precision percentage with 2.091. Concluding, the PubLayNet ResNet50 model achieved the best overall performance by having the best scores on four of the five elements. However, taking the sum of the precisions, the PubLayNet ResNeXt101 had the highest precision score. Comparing these results from 4, it can be seen that the results of the baseline model were significantly better at predicting tables and lists, with the improved model better at predicting texts and figures. Both the baseline as well as the improved model were not able to detect titles.

Table 6: Bounding box element precision synthetic dataset variations

| Category | Architecture | Bounding Box Element Precision |
|---|---|---|
| Text | PubLayNetResNet50 FPN | **1.408** |
| | PubLayNetResNet101 FPN | 0.485 |
| | PubLayNetResNeXt101 FPN | 0.022 |
| Table | PubLayNetResNet50 FPN | 32.458 |
| | PubLayNetResNet101 FPN | 35.719 |
| | PubLayNetResNet101x FPN | **45.541** |
| Title | PubLayNetResNet50 FPN | 0.000 |
| | PubLayNetResNet101 FPN | 0.000 |
| | PubLayNetResNeXt101 FPN | 0.000 |
| Figure | PubLayNetResNet50 FPN | **2.002** |
| | PubLayNetResNet101 FPN | 0.824 |
| | PubLaynetResNeXt101 FPN | 0.139 |
| List | PubLayNetResNet50 FPN | **2.091** |
| | PubLayNetResNet101 FPN | 1.818 |
| | PubLayNetResNeXt101 FPN | 0.820 |

## 5.4 *Average Precisions Detectron2 Architectures for the manually annotated Dataset*

The last Detectron2 variations were trained on the manually annotated dataset. In table 7, the different average precisions comparing the baseline models, improved synthetic dataset models, and manually annotated dataset models are illustrated. The different average precisions for the manually annotated dataset were significantly higher than the baseline and synthetic dataset variations. The mAP50 ranged from 82.746 to 88.498 for the PubLayeNt ResNeXt101. Similar to the previous models, the manually annotated model variations had a precision of 0.000 on the mAPs, indicating that these variations could not detect elements of a smaller size. The ResNext101 model variation had the highest overall score on the manually annotated dataset on every precision apart from the mAPm. The manually annotated dataset model variation outperformed the baseline and synthetic model variations for every precision with a precision delta – highest precision minus lowest precision – ranging from 48.968 to 63.367 percent for the baseline model and a precision delta ranging from 48.005 to 78.077 percent for the Synthetic dataset variations (i.e. 48.968 - 0.00 for the mAPm).

Table 7: Mean average precisions across the baseline and imrpoved models

| Architecture | mAP | mAP50 | mAP75 | mAPs | mAPm | mAPl |
|---|---|---|---|---|---|---|
| PubLayNet ResNeXt50 FPN Baseline | 16.441 | 20.614 | 17.060 | 0.000 | 0.000 | 16.484 |
| PubLayNet ResNet101 FPN Baseline | 12.340 | 15.192 | 12.733 | 0.000 | 0.000 | 12.340 |
| PubLayNet ResNeXt101 FPN Baseline | 19.305 | 25.131 | 19.996 | 0.000 | 0.000 | 19.482 |
| PubLeyNet ResNet50 FPN Synthetic | 7.592 | 10.421 | 7.691 | 0.000 | 0.973 | 7.840 |
| PubLayNet ResNet101 FPN Synthetic | 7.769 | 9.732 | 9.033 | 0.000 | 0.337 | 7.891 |
| PubLayNet ResNeXt101 FPN Synthetic | 7.769 | 9.732 | 9.033 | 0.000 | 0.337 | 7.891 |
| PubLeyNet ResNet50 FPN Manually | 69.366 | 85.858 | 77.976 | 0.000 | **48.968** | 69.419 |
| PubLayNet ResNet101 FPN Manually | 65.466 | 82.746 | 73.433 | 0.000 | 31.722 | 66.262 |
| PubLayNet ResNeXt101 FPN Manually | **70.942** | **86.898** | **78.927** | 0.000 | 40.326 | **72.946** |

## 5.5 *Object Detection for Detectron2 comparing Element Precisions using the Manually Annotated Dataset*

This subsection presents the element precisions of the model variations trained on the manually annotated dataset. In table 8, the different element precisions are visualized. Contrary to the previous element precisions, these model variations performed significantly better on all elements. For all three model variations, the element table had the highest element precision ranging from 83.516 to 87.858 percent followed by the element text ranging from 76.076 to 78.593 percent. Contrary to the previous models,

these variations can detect titles. With element precisions ranging from 58.552 to 60.767 percent as well as a high element precision for figures, ranging from 62.149 to 76.396. The worst detected elements were lists, with a element precisions ranging from 45.748 to 56.926. Although the ResNext101 variation had the highest average precisions for four of the five precisions, it only had the highest element precisions of two of the five elements.

Table 8: Bounding box element precision manually annotated Dataset Variations

| Category | Architecture | Bounding Box Element Precision |
|---|---|---|
| Text | PubLayNetResNet50 FPN | **78.593** |
| | PubLayNetResNet101 FPN | 77.366 |
| | PubLayNetResNeXt101 FPN | 76.076 |
| Table | PubLayNetResNet50 FPN | **87.858** |
| | PubLayNetResNet101 FPN | 83.516 |
| | PubLayNetResNet101x FPN | 86.754 |
| Title | PubLayNetResNet50 FPN | **60.767** |
| | PubLayNetResNet101 FPN | 58.552 |
| | PubLayNetResNeXt101 FPN | 58.556 |
| Figure | PubLayNetResNet50 FPN | 69.449 |
| | PubLayNetResNet101 FPN | 62.149 |
| | PubLaynetResNeXt101 FPN | **76.396** |
| List | PubLayNetResNet50 FPN | 50.162 |
| | PubLayNetResNet101 FPN | 45.748 |
| | PubLayNetResNeXt101 FPN | **56.926** |

# 6 DISCUSSION

The main goal of this research was to identify which Mask R-CNN models – focusing on the Detectron2 Mask R-CNN models – had the highest performance for DLA and to compare the results of the synthetic dataset with the manually annotated dataset. This was done to identify whether the implementation of synthetic data could account for the manual expensive annotation process. These datasets were created from contract documents, standardization documents, and information documents. The synthetic dataset was generated using cut-out foreground titles, figures, tables, texts, and lists. The manually annotated dataset was generated by manually annotating the five elements. The baseline model variations of this research were the Detectron2 Mask R-CNN model 'as is' with three ResNet variations. To these variations, six model variations of Detectron2 Mask R-CNN architectures had been compared. Half of the variations were trained on the synthetic dataset and the other half on the manually annotated dataset. Prior research, in particular, Lin et al. (2014) and Li et al. (2020), found that Mask R-CNN models were useful in object detection, which can also be used in DLA. This research found that multi-structured documents were more difficult to analyse using the synthetic dataset compared to the manually annotated dataset. This falls in line with the studies from Li et al. (2020) Docbank dataset and Zhong et al. (2019) PubLayNet dataset, although these datasets used structured documents. Although the object detection results of the synthetic dataset were lower than the baseline and manually annotated dataset, this method potentially opened the doors for the use of, or inclusion of, a synthetic dataset. To come to this conclusion, several subquestions have been examined throughout this study.

## 6.1 *Mean average Precisions across the Detectron2 Pre-Trained Models and Architectures*

The first sub-question explored whether different Mask R-CNN Detectron2 architectures differ significantly in performance between the baseline, synthetic, and manual annotated models. The results from the proposed model variations on the synthetic dataset achieved lower results compared to the baseline model. Previous research with object detection baseline methods using Mask R-CNN models had shown a step-wise increase in performance using the ResNet variations. Where the ResNeXt101 architectures outperformed ResNet101 and ResNet50 architectures (Li et al., 2020; Lin et al., 2014). This is partly in line with the results of this study. In some cases, the ResNet50 outperformed the ResNet101 and ResNeXt101

architectures and vice versa. However, this might be due to the ResNet50 architecture being easier to use for simpler datasets (K. He et al., 2016).

It was not expected that the baseline variations outperformed the proposed variations on the synthetic dataset. Every baseline model variation outperformed the proposed synthetic variations. The highest precision of the synthetic model variation (9.304) had a lower precision than the lowest baseline result (19.305). According to (Nikolenko, 2019), this phenomenon can be attributed to the synthetic images not representing exactly the multi-structured documents available, thus resulting in lower test results. However, the implementation of the synthetic dataset functioned as an introduction to explore whether a synthetic dataset could be useful.

The best performing model regarding the different mAP was the ResNext101 model for the manually annotated dataset. This is in line with the Zhong et al. (2019) study. Although their study uses solely structured documents, this manually annoted dataset is more-structured than the randomly placed elements in the synthetic dataset. Also, comparing these results to previous research regarding PubLayNet represents a precision decrease in our study. This might be due to the multi-structural dataset, which makes prediction more difficult. Hence, taking away the structure might result in a decrease in the performance of the model variations.

## 6.2 *Element Performance*

The second sub-question explores the individual element precisions across the different datasets between the model variations. To improve the performance of the Detectron2 model on element detection, multiple different architectures were examined. The six PubLayNet models were evaluated on their element precisions. Results showed that the three synthetic models had difficulty detecting titles. This can be attributed to the annotation differences between texts, titles, and lists. All three elements have the same annotation properties, consisting only of words and spaces. Tables have the highest detecting precision. This can indicate that the shape of the table is easier detected than solely words or different shaped figures. Text, titles, tables, and lists all consist of words, thus having overlap in the way they are structured and annotated. Therefore, finding a distinctive annotation for these elements is difficult. Hence, this could result in difficulties for the model to detect elements. However, since the synthetic dataset was multi-structured, the percentage precisions were not as high as the Zhong et al. (2019) study, which achieved high results (precisions above 90 percent).

The element precisions of the manually annotated dataset resulted in significantly higher results than their synthetic counterpart and baseline

model, with results ranging from 56.926 to 87.858 percent for the best model.

## 6.3 *Real-Life Document Comparison*

The last sub-question investigates whether a conclusion can be made if the results on the synthetic dataset are appropriate enough for real-life documents. To test this, all the model variations were being tested on the manually annotated test set and results were compared. As can be seen in section 5 Results, the synthetic dataset varies greatly from the standard document structure known. This is done because of the central multi-structured aspect in this study. As a result, it is important to know how these trained models performs compared to real-life documents. As mentioned in section 5, the synthetic dataset model variations performed worse than the baseline method. The neglected performance could be influenced by several issues. First, the manually annotated dataset is more-structured than the randomly placed element in the synthetic dataset, which results in less performance Nikolenko (2019). Second, consideration had to be made between the number of elements on a page and the non-overlapping elements on that page. The choice of two elements per page solved the overlapping of the elements. However, this neglected the fact that in real-life documents, a page consists mostly of more than two elements. This could potentially reduce the detection power of the synthetic dataset. Lastly, to increase precision on these multi-structured documents and account for declined precision, more data can be annotated. In this study, only 250 elements instances were annotated (50 titles, 50 figures, 50 texts, 50 titles, and 50 tables). When comparing against the DocBank and PubLayNet datasets, these differences are immense. However, when increasing this number of annotations, the model might perform better on the multi-structured data (Yepes et al., 2021). Therefore, performance can be improved by increasing the annotations (Molloy, 2011). Even though the low results of the variations trained on the synthetic dataset were not expected in this study, it can be argued that there is great potential in using a synthetic dataset that is more tailored to existing documents. This could allow for a clearer representation of real-life PDF documents.

## 6.4 *Limitations*

Due to the computationally expensive models used in this research, consideration had to be made between the length of the training and testing time and the performance of the model. In previous studies with Li et al. (2020) and Zhong et al. (2019), model training time exceeded multiple days

and iterations exceeded multiple ten thousand. Due to time constraints, this was deemed impossible. Therefore, the training maximum was set at three hours and the iterations at 3000 for hyper-parameter tuning.

Another limitation is the time-consuming annotation process. Since the annotations process is manually expensive – and this research was done individually – large amounts of annotations were impossible. Hence, element annotations were set at 50 for the synthetic created dataset which accounted for the low amount of data. The combination of the pre-trained models, as well as the synthetic created dataset, accounted for the relatively low amount of annotations. The annotation limitation of the manual dataset was set at 15 hours, which resulted in 2916 annotations.

Despite its limitations, this study might offer some contributions to the field of DLA and object detection in the following ways. This study made efforts to introduce multi-structured documents with object detection. Also, by comparing the results of a synthetic dataset and manually annotated dataset regarding object detection, improvements have to be made to tailor the synthetic dataset for better results. By doing this, such a synthetic dataset could, in the future, be used to account for the manual annotation process. Also, in the field of document layout analysis, multi-structured documents tend to be more difficult to detect. However, with mimicking the diverse structure of a document, DLA achieved good performance.

## 7 CONCLUSION

The study explores which Mask R-CNN model variation – in particular the Detectron2 model – is useful for DLA for multi-structured information documents using object detection. This was done by answereing the following research question, 'Which is the best performing Detectron2 Mask Region-Based Convolutional Neural Network variation for Document Layout Analysis on Multi-Structured Information Documents?'. It also compares two different datasets, one synthetic created and one manually annotated. Several sub-questions have been formulated to answer this research goal. To try to come to this goal, this research used a pre-trained model with a synthetic created dataset to mimick the multi-structure of the real-life documents which was compared to a smaller manually annotated dataset. In contrast to previous studies from Li et al. (2020), Lin et al. (2014), and Zhong et al. (2019), this study implemented the research direction for multi-structured documents by using a synthetic created dataset and multi-structured documents by comparing the results on multiple datasets.

The results show that the combination of a pre-trained model with a synthetic created dataset achieved lower scores than their baseline model and manually annotated counterpart. However, the results argued that a

synthetic dataset is still a promising way of training data. The baseline model outperformed the proposed model variation of the synthetic dataset. However, the baseline model was outperformed by the manually annotated variations which resulted in high average precisions and element precisions. The results of the first subquestion have shown a distinctively set baseline model to improve. Both the synthetic dataset and baseline model were not able to detect titles, with the synthetic dataset results being low for text figures and lists as well.

The precisions of the three improved Detectron2 models on the manually annotated dataset showed that the these models were good in predicting all elements and that the models – compared to the baseline – were drastically improved. The PubLayNet Detectron2 Models showed great results in detecting every element, with the lowest score still drastically improved compared to the other models.

In conclusion, although the synthetic dataset achieved lower scores than expected, there can still be argued for promising future results. The introduction of a synthetic dataset in DLA can achieve high precision scores when the dataset is created more critically with more pre-determined rules for document re-creation. This can be achieved by annotating more data, placing more elements on a page, and creating fewer random images. Although the dataset was smaller, the manually annotated dataset showed that the models achieved high detection scores and showed that DLA can be applied to multi-structured data. Just like the synthetic dataset, higher results can be achieved by increasing the number of annotations.

In future research, it would be interesting to discover to what extent a very large synthetic dataset can achieve the same results as the pre-trained model or manually annotated dataset. Also, a more advanced generation document can be made to synthetically create the dataset to include more instances on a page, to more closely mimic the multi-structure documents. In this study, only two foregrounds were generated on every page. Including more data in the form of annotation might contribute to a higher performance of multi-structural document understanding.

## REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from https://www.tensorflow.org/ (Software available from tensorflow.org)

Antonacopoulos, A., Pletschacher, S., Bridson, D., & Papadopoulos, C. (2009). Icdar 2009 page segmentation competition. In *2009 10th international conference on document analysis and recognition* (pp. 1370–1374).

Belhadj, D., Belaïd, Y., & Belaïd, A. (2021). Automatic generation of semi-structured documents. In *International conference on document analysis and recognition* (pp. 191–205).

Binmakhashen, G. M., & Mahmoud, S. A. (2019). Document layout analysis: A comprehensive survey. *ACM Computing Surveys (CSUR)*, *52*(6), 1–36.

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 9157–9166).

Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V.-K., & Strintzis, M. G. (2005). Knowledge-assisted semantic video object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(10), 1210–1224.

Deng, L., & Liu, Y. (2018). *Deep learning in natural language processing*. Springer.

Freeman, W. T., Tanaka, K.-i., Ohta, J., & Kyuma, K. (1996). Computer vision for computer games. In *Proceedings of the second international conference on automatic face and gesture recognition* (pp. 100–105).

Gatos, B., Louloudis, G., & Stamatopoulos, N. (2014). Segmentation of historical handwritten documents into text zones and text lines. In *2014 14th international conference on frontiers in handwriting recognition* (pp. 464–469).

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the ieee international conference on computer vision* (pp. 1440–1448).

Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 1–19.

Handschuh, S., & Staab, S. (2003). *Annotation for the semantic web* (Vol. 96). IOS press.

Handschuh, S., Staab, S., & Maedche, A. (2001). Cream: creating relational metadata with a component-based, ontology-driven annotation framework. In *Proceedings of the 1st international conference on knowledge*

*capture* (pp. 76–83).

He, D., Cohen, S., Price, B., Kifer, D., & Giles, C. L. (2017). Multi-scale multi-task fcn for semantic page segmentation and table detection. In *2017 14th iapr international conference on document analysis and recognition (icdar)* (Vol. 1, pp. 254–261).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the ieee international conference on computer vision* (pp. 2961–2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated machine learning: methods, systems, challenges*. Springer Nature.

Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019). Panoptic feature pyramid networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 6399–6408).

Kirillov, A., Wu, Y., He, K., & Girshick, R. (2020). Pointrend: Image segmentation as rendering. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 9799–9808).

Li, M., Xu, Y., Cui, L., Huang, S., Wei, F., Li, Z., & Zhou, M. (2020). Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755).

Molloy, J. C. (2011). The open knowledge foundation: open data means better science. *PLoS biology*, *9*(12), e1001195.

Nikolenko, S. I. (2019). Synthetic data for deep learning. *arXiv preprint arXiv:1909.11512*.

Open Knowledge (Ed.). (2015). *The open data handbook*. Open Knowledge. Retrieved from http://opendatahandbook.org/guide/en/

Padilla, R., Passos, W. L., Dias, T. L., Netto, S. L., & da Silva, E. A. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, *10*(3), 279.

Pletschacher, S., & Antonacopoulos, A. (2010). The page (page analysis and ground-truth elements) format framework. In *2010 20th international conference on pattern recognition* (pp. 257–260).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*, 91–99.

Rigaud, C., Guérin, C., Karatzas, D., Burie, J.-C., & Ogier, J.-M. (2015). Knowledge-driven understanding of images in comic books. *Inter-*

*national Journal on Document Analysis and Recognition (IJDAR)*, *18*(3), 199–221.

Tran, T. A., Oh, K., Na, I.-S., Lee, G.-S., Yang, H.-J., & Kim, S.-H. (2017). A robust system for document layout analysis using multilevel homogeneity structure. *Expert Systems With Applications*, *85*, 99–113.

Trivedi, A., & Sarvadevabhatla, R. K. (2021). Boundarynet: an attentive deep network with fast marching distance maps for semi-automatic layout annotation. In *International conference on document analysis and recognition* (pp. 3–18).

Wu, A. M. F. L. W. G. R., Y.; Kirillov. (2019). *Detectron2: A pytorch-based modular object detection library.*

Wu, X., Ma, T., Li, X., Chen, Q., & He, L. (2021). Human-in-the-loop document layout analysis. *arXiv preprint arXiv:2108.02095*.

Yepes, A. J., Zhong, X., & Burdick, D. (2021). Icdar 2021 competition on scientific literature parsing. *arXiv preprint arXiv:2106.14616*.

Zhang, P., Li, C., Qiao, L., Cheng, Z., Pu, S., Niu, Y., & Wu, F. (2021). Vsr: A unified framework for document layout analysis combining vision, semantics and relations. *arXiv preprint arXiv:2105.06220*.

Zhong, X., Tang, J., & Jimeno Yepes, A. (2019). Publaynet: Largest dataset ever for document layout analysis. In *2019 international conference on document analysis and recognition (icdar)* (p. 1015-1022). doi: 10.1109/ICDAR.2019.00166