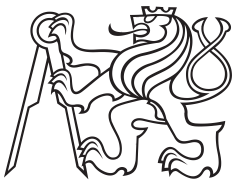
Bachelor Project



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

News Article Layout Extraction from Bitmaps Files

Vít Zeman

Supervisor: Ing. Jan Drchal, Ph.D.
Field of study: Artificial Intelligence
May 2021

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Zeman Vít** Personal ID number: **474549**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

II. Bachelor's thesis details

Bachelor's thesis title in English:

News Article Layout Extraction from Bitmaps Files

Bachelor's thesis title in Czech:

Extrakce rozložení novinových článku z rasterových předloh

Guidelines:

The task is to develop, implement, and evaluate machine learning methods to analyze news page layouts in order to extract separate articles. The input data involve bitmap images of newspaper pages.
1) Explore the state-of-the-art methods of layout extraction.
2) Experiment with selected methods to segment news articles.
3) Evaluate the methods on existing datasets or possibly other datasets supplied by the supervisor.

Bibliography / sources:

[1] Eskenazi, Sébastien, Petra Gomez-Krämer, and Jean-Marc Ogier. "A comprehensive survey of mostly textual document segmentation algorithms since 2008." Pattern Recognition 64 (2017): 1-14.
[2] Zhong, Xu, Jianbin Tang, and Antonio Jimeno Yepes. "Publaynet: largest dataset ever for document layout analysis." 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.
[3] Almutairi, Abdullah, and Meshal Almashan. "Instance Segmentation of Newspaper Elements Using Mask R-CNN." 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, 2019.
[4] Meier, Benjamin, et al. "Fully convolutional neural networks for newspaper article segmentation." 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Vol. 1. IEEE, 2017.
[5] Naoum, Andrew. "Article Segmentation in Digitised Newspapers.", dissertation thesis, Faculty of Engineering and Information Technologies The University of Sydney, (2020).

Name and workplace of bachelor's thesis supervisor:

Ing. Jan Drchal, Ph.D., Department of Theoretical Computer Science, FIT

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **04.02.2021** Deadline for bachelor thesis submission: **21.05.2021**

Assignment valid until:
by the end of summer semester 2021/2022

Ing. Jan Drchal, Ph.D.	prof. Ing. Michael Šebek, DrSc.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. | Date of assignment receipt

Student's signature

Acknowledgements

Firstly, I would like to thank my supervisor Ing. Jan Drchal, Ph.D., whose guidance and advice were very valuable. The access to the computational infrastructure of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" is also gratefully acknowledged. Lastly, I would like to thank my parents for their unending support.

Sincerely, thank you.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, 21. May 2021

Abstract

The aim of this thesis is the extraction of newspaper articles from bitmap files. State of the art of object detection was described with a focus on R-CNN architectures. These methods were implemented via `detectron2`, a Python library.
Additionally, preprocessing of the provided dataset was necessary. The conversion to bitmap files was needed as well as the creation of annotation files from the provided XML files.
Performed experiments are mainly exploring how the model performs with changes on the training dataset. The secondary output of the thesis was the detection of newspaper elements.

Keywords: Convolutional neural network, CNN, newspapers, articles extraction, images, R-CNN

Supervisor: Ing. Jan Drchal, Ph.D.

Abstrakt

Cílem práce je extrakce novinových článků z rasterovaných předloh. Dále byl zpracován přehled moderních metod používaných pro detekci objektů se zaměřením na R-CNN metody. Tyto metody byly implementovány pomocí knihovny `detectron2` programovacího jazyka Python.
Dále, bylo nutné zpracovat poskytnuté novinové datasety. Součástí zpracování bylo převedení do rasterované podoby a vytvoření anotačního souboru z poskytnutých XML souborů.
Provedené experimenty především prozkoumávají chování modelu na různých trénovacích datasetech. Vedlejším výstupem práce byla detekce jednotlivých novinových elementů.

Klíčová slova: Konvoluční neuronové sítě, noviny, CNN, extrakce článků, rasterovaná grafika, R-CNN

Překlad názvu: Extrakce rozložení novinových článku z rasterových předloh

Contents

1 Introduction **1**

1.1 Digital humanities 1

1.2 News Article Layout Extraction . 2

1.3 Task of the thesis 2

1.4 Overview of thesis 3

2 State of the art **5**

2.1 Object detection with convolutional neural networks 6

2.1.1 Regions with CNN feature . . . 7

2.1.2 Fast R-CNN 8

2.1.3 Faster R-CNN 8

2.1.4 Mask R-CNN 9

2.1.5 YOLO 10

3 Used datasets **13**

3.1 Právo dataset 13

3.2 Vector-Based Article Segmentation dataset 14

3.3 PubLayNet 16

3.4 Dataset preprocessing 16

3.4.1 Description of COCO annotation format 17

3.4.2 Creation of datasets 19

4 Aproach **23**

4.1 Evaluation metric 23

4.2 Implementation 26

4.2.1 Conversion of PDFs 26

4.2.2 Conversion of XML files with annotations 26

4.2.3 Library detectron2 27

4.2.4 Main program overview 28

5 Experiments **31**

5.1 Pre-training models 31

5.2 Merging datasets 32

5.3 Size of dataset 33

6 Discussion **39**

6.1 Discussion of the experiments . . 39

6.2 Showcase on never-seen newspapers 40

6.2.1 Article segmetation 40

6.2.2 Element type prediction 42

7 Conclusion **45**

A Bibliography **47**

B List of Abbrevition **51**

C CD Content **53**

Chapter 1

Introduction

Even before the invention of the classical computer, there have been efforts to convert written text into another form of a signal, primarily sound. These techniques are commonly called optical character recognition (OCR). The first works in this field were focused on allowing blind people to read printed text. For example, in 1914, Emanuel Goldberg invented a machine that converted characters to their telegraph codes. Study in this field continues even now.

1.1 Digital humanities

A new field of study emerged in the last decade, which combines digital technologies and humanities studies. It is called digital humanities. And although no formal definition is yet formed or more specifically agreed upon, it is the field of study which intrigues many people. Furthermore, digitization continues in many branches of industry and public services. For example, Czech National Archive works on the National Digital Archive, which aims to preserve and allow easier access to both old and new documents.

All these types of digitization allow new kinds of research, which would be nearly impossible in the past due to mostly time and monetary constraints and complexity of the tasks. The research can now focus on the development of society during years. For instance, analysis and investigation of social media, newspapers, etc. As these media can and often are following and influencing opinions in society.

1.2 News Article Layout Extraction

As mentioned in the previous section, one of the branches of digital humanities is the analysis of newspapers. Knowing the locations of both articles and their elements makes it easier to extract information located in the newspapers. Techniques that allow this are called layout extraction or layout analysis.

Depending on the type of the document, some information can be extracted with deterministic methods. One of, if not the most, used formats for data storage is Portable Document Format (PDF). It is capable of extracting some elements such as text, figures, etc. PDF even allows layering, mainly used for 3D drawings, but its usage is not consistent across all types of documents. Furthermore, PDF and other formats often do not carry information about the grouping of semantics, for instance, articles. Because of this, layout extraction is needed.

One way of implementing it is with convolutional neural networks(CNNs). These networks are now being used in many applications, thanks to the rise of computation power and the decrease of costs. CNNs are most commonly used for analyzing visual imagery, and its input layer is usually made from bitmap files. That leads to an advantage of this implementation, which is that the PDF files do not need to be consistent as PDF format stores data of multiple types. For example, it can store both raster and vector graphics. Vector graphics can be easily transformed into raster graphics, but the opposite transformation is not trivial. Another advantage of this implementation is the capability of usage on scanned pages.

1.3 Task of the thesis

Article segmentation is the main task. Its aim is to find articles on the bitmap file. The main declared output is a bounding box; the secondary one is a segmentation mask. Expected visualized output can be seen in image 1.1

The secondary task is element type prediction. Its goal is to locate and predict the type of newspaper elements such as headlines, body(texts), captions, etc. Expected visualized output can be found in image 1.2

Figure 1.1: Examples of article segmentation on newspapers from Právo dataset. Visualization made out of an annotation.

Figure 1.2: Examples of element type prediction on newspapers from Právo dataset. Visualization made out of an annotation.

1.4 Overview of thesis

In chapter 2, state of the art of layout extraction is introduced. Chapter 3 describes the used datasets and the derived datasets. In the following chapter 4, an approach to the problem and its implementation is described, as well as other needed tools. In chapter 5, multiple approaches are tested and evaluated. Lastly, Chapter 6 is dedicated to a discussion of the experiments.

Chapter 2

State of the art

Multiple types of approaches exist for segmentation algorithms used on newspapers. According to both [6] and [13], the main categories are rule-based approaches and machine learning approaches. The Rule-based approaches can be effective for newspapers with a strict or at least consistent structure. Furthermore, they are not effective on unexpected inputs, such as different brands or periodicals. For example, [3] are mainly detecting white separators with Hough transformation. Because of these reasons, machine learning algorithms are preferred in recent years.

Machine learning algorithms are more general. Furthermore, they are on the rise as the computation power goes up. One of the most common techniques for segmentation is clustering. However, in recent years, CNNs are used for the detection and segmentation of nearly everything, including articles of newspapers.

For example, Benjamin Meier et al.[12] used Fully convolutional neural network (FCN) for segmenting newspapers into articles from bitmap files. The method was trained on a private dataset made from Swiss newspapers, provided by ARGUS DATA INSIGHTS Schweiz AG. Although some pages were discarded because of bad labels, others were discarded intentionally due to the non-rectangular shape of the articles. This makes the task easier.

Another interesting work was done by Abdullah Almutairi and M. Almashan [1]. They used Mask R-CNN for segmenting articles from bitmap files. The network was trained on a dataset that consisted of newspapers written in both Latin script and Arabic. Their main goal was to develop a method which would

not depend on the language of the newspapers. Additionally, three categories were detected, article, advertisement, and page header. Furthermore, the trained model was then used on multiple by model never-seen newspapers. The biggest problem occurred with Japanese newspapers, as Japanese writing system varies from the one used in training.

Other techniques are combining CNNs with other ML methods. For example, Xiao-Hui Li et al. [11] propose a combination of CNNs and Graphical models. Firstly, the primitives are extracted by image processing techniques, then they are used to build a graph with nodes corresponding to the primitives, and the edges represent relations between neighboring primitives. The CNN is then used for the classification of classes for both edges and primitives. The technique was tested on dataset PubLayNet[21].

Even though some methods used the segmentation mask as output [12]. Article extraction could be implemented as a task of object detection.

2.1 Object detection with convolutional neural networks

The object detection task aims to find a bounding box, which is given by four values. These values can be given by two pairs of (x,y) coordinates (for two diagonally opposing corners) or by coordinate (x, y), width w, and height h. In this case, the coordinate marks usually top-left corner, although in some applications, it can mark the center of the bounding box.

Application spectrum of object detection is vast, for example, in medicine (detection of cancerous cells [5]), automotive (detection of objects on the streets [2]), and geography(detection of roads [4]).

The naive approach for object detection is to take an image and perform image classification, where the whole image is used as an input of CNN, and the output is a prediction vector with probabilities for each class. However, this approach only works for the prediction of one instance. In the case of multiple instances, this implementation could be naively used by selecting smaller sections of the image and making predictions based on them. This would, however, lead to the creation of substantial numbers of these sections, as the sought object bounding boxes could differ in both size and location. Therefore, specialized algorithms were created for object detection. These

can be divided into two groups:

1. Region-based algorithms, such as R-CNN
2. One-shot algorithms, such as YOLO

In general, one-shot algorithms are faster and usable in real-time, while region-based are more precise. As the task of this thesis does not explicitly require real-time response, the thesis will deal with region-based algorithms.

2.1.1 Regions with CNN feature

One algorithm which aims to limit the number of sections is R-CNN: Regions with CNN features[9]. This algorithm (overview on figure 2.1) extracts approximately 2 000 regions from the image, chosen by the selective search algorithm [18]. It generates the initial sub-segmentation with graph-based methods[7], combines smaller similar regions into a larger one by a greedy algorithm and, after a specified amount of iterations, uses it to produce the proposal regions. These are then transformed into squares with fixed size and used as an input for CNN, which extract their features. Furthermore, the extracted features are used by the support vector machine (SVM) final classification of the region

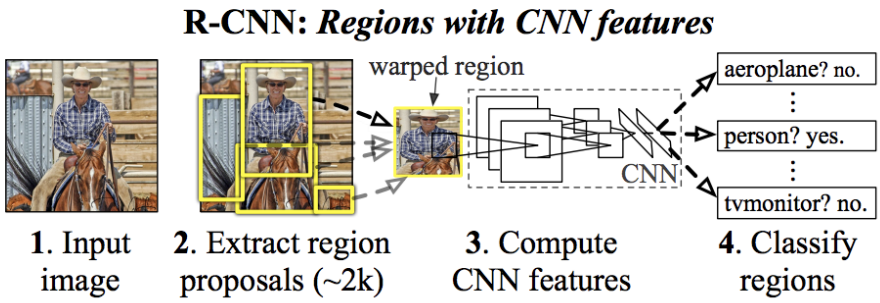


Figure 2.1: R-CNN overview [9]

Even though this approach decreases the number of proposed regions. The classification of 2 000 regions still requires longer time than is acceptable in real-time scenarios. This is because CNN takes the warped region as an input. Therefore, subsequent works try to bypass this obstacle.

2.1.2 Fast R-CNN

One of the derived architectures was Fast R-CNN [8], depicted in figure 2.2. This network saves computation time by processing the whole input image by CNN and generating a convolutional feature map. This CNN is usually referred to as a backbone. The regions of interest (ROIs), which are identical to the proposal regions in R-CNN, are then projected into the feature map. ROIs are then reshaped by pooling into a fixed size, called feature vector. This vector is then used as an input for fully connected layers with two outputs. One output is used for the prediction of the classes, while the other predicts the bounding box parameters.

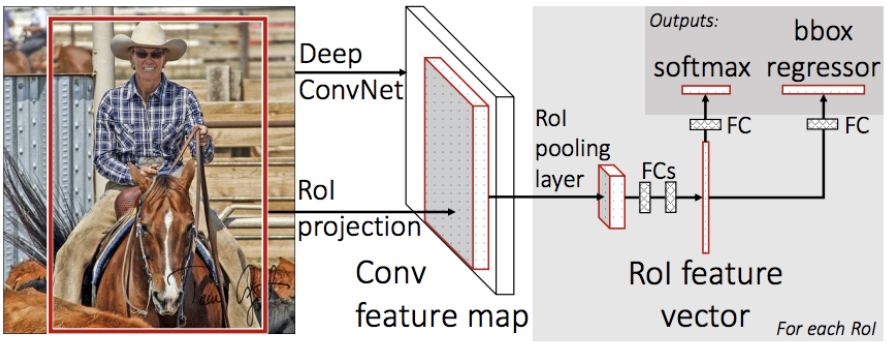


Figure 2.2: Fast R-CNN architecture [8]

2.1.3 Faster R-CNN

While Fast R-RCNN is significantly faster than R-CNN, the improvements revealed a bottleneck: the selective search algorithm used for the creation of proposal regions. Therefore, the next goal of improving the R-CNN based detection algorithm was its replacement. This lead to the creation of Faster R-CNN architecture [16], which consists of two cooperating networks, Fast R-CNN and Region proposal network (RPN).

The main idea behind this network is to replace the selective search algorithm with the RPN, which locates the proposal regions in the feature map and shares them, after adjustment, by pooling with Fast R-CNN. Thanks to this improvement, the prediction is made in terms of tenths of seconds[1] on

[1]This is close to reaction time of humans(200 ms) and therefore it can be used for real time applications.

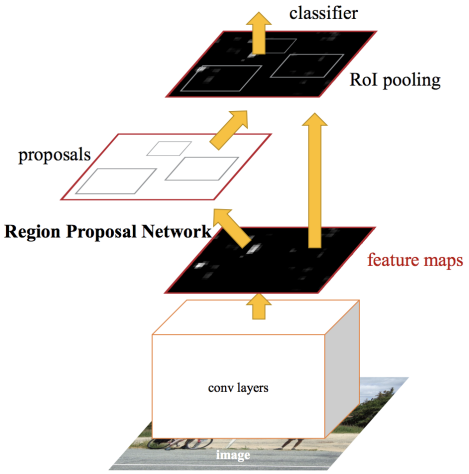


Figure 2.3: Faster R-CNN architecture [16]

NVIDIA Tesla K40 GPU.

2.1.4 Mask R-CNN

Introduced by Kaiming et al. [10], Mask R-CNN expands the capabilities of Faster R-CNN. In addition to the previous network outputs, class label, and bounding box, it adds a segmentation mask. It is computed from the RoI in parallel with the computation of the bounding box values and class label predictions. This computation is made by using an FCN, and its output is a binary mask. Furthermore, the same as the bounding box in Fast R-CNN, the masks are computed for every class.

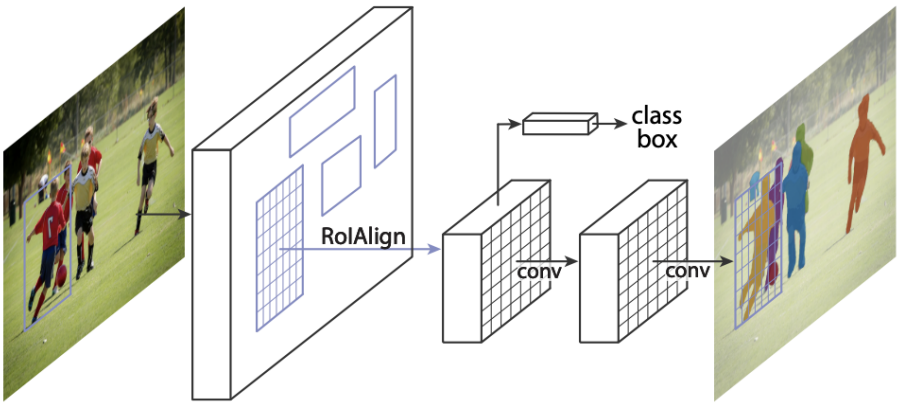


Figure 2.4: Mask R-CNN architecture [10]

2.1.5 YOLO

Algorithm You Only Look Once (YOLO)[15] is built on different premises than those before. As the name suggests, YOLO directly predicts the bounding boxes and their probabilities straight from the image by using a single FCN.

Brief visualization of YOLO is in figure 2.5. The idea behind YOLO is to divide the image into a grid. Additionally, for each cell of the grid is predicted: a specific number of bounding boxes and their confidence score and class probabilities in the cell. If the bounding box has a class probability greater than the set threshold is used for the localization of an object. If more bounding boxes of the same class are overlapping, a non-maximal suppression is used.

Non-maximal suppression is a technique of filtering bounding boxes by checking their overlap. First, the bounding box with the highest confidence is selected. IoU calculation[2] is then done between the selected bounding box and all others of the same class. If IoU is bigger than the selected threshold, the bounding box with smaller confidence is deleted. Then the bounding box with the second highest bounding box confidence is selected, and the process repeats. The algorithm stops when all pairs of bounding boxes with IoU bigger than the threshold were deleted.

YOLO limitation is in the maximal number of detected entities, because each cell can correspond to only one bounding box. On the other hand, it is very popular due to its speed. [15] states that the original base network runs at 45 frames per image on Nvidia Titan X GPU.

[2] See section 4.1 for explanation

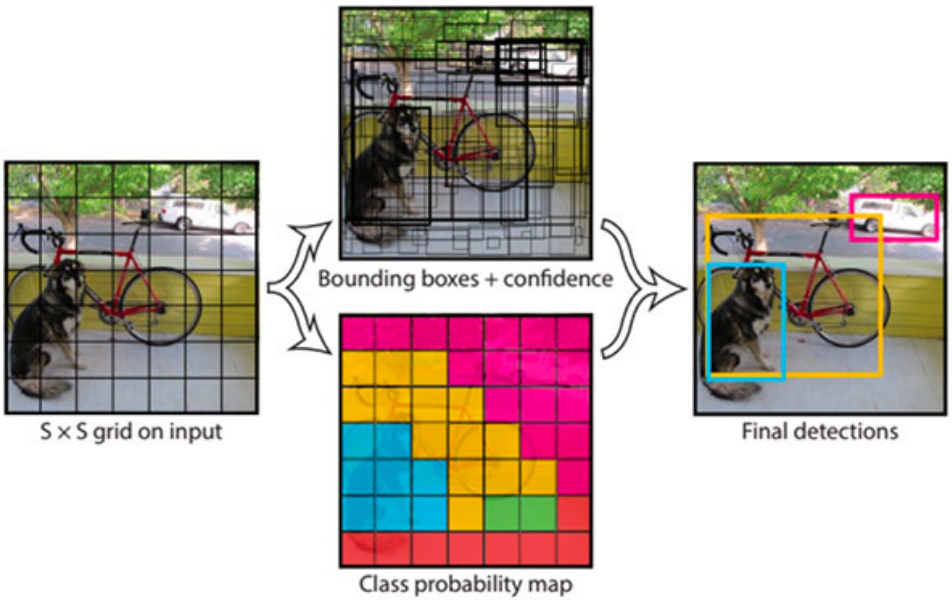


Figure 2.5: Example of YOLO alghorithm[15]

Chapter 3

Used datasets

Probably the most crucial part of every machine learning algorithm are the datasets used for training. For this work, multiple datasets were used. One of them from the public domain and the rest were provided by the supervisor. The problem with newspaper layout extraction is that nearly every encountered paper used a dataset under copyright law. Therefore, they were not made public.

3.1 Právo dataset

This dataset, which was provided by the supervisor, consists of 349 pages from the local Czech newspaper Právo. The newspaper was kindly provided by the publisher of the newspaper Newton Media, as. For every image, an already preprocessed XML file was also provided. These files were created by Ing. Radek Mařík, CSc.[1] who used a ruled-based algorithm to extract data needed for annotations. Additionally, the created files were filtered by supervisor Ing. Jan Drchal, Ph.D., because of the imperfections of the ruled-based algorithm. This led to the creation of a cleaner dataset at the cost of its size.

[1] https://comtel.fel.cvut.cz/en/users/marikr

Syntax of XML files :

```
<pages>
  <page id="1" bbox="0.0,0.0,841.89,1190.551" rotate="0.0">
  <news id="0"  bbox="x_0, y_0, x_1, y_1">
    <title>
        <textbox bbox="x_0, y_0, x_1, y_1" fontSize="f_s"
            tags="headline"> Headline text</textbox>
    <\title>
    <body>
    <textbox id="ID" bbox="x_0, y_0, x_1, y_1"
        fontSize="f_s"  tags="tag"> Text <\textbox>
    ...
    <figurebox id="ID" bbox="x_0, y_0, x_1, y_1" ></figurebox>
    <\body>
  <\news id="1" bbox="x_0, y_0, x_1, y_1">
  <news>
  ...
  <\page>
</pages>
```

One of the used annotation was `news` for newspaper article detection. Their locations were given by bounding boxes declared by coordinates of the bottom-left and upper-right corners. Another problem is that XML file annotations are made from PDF files, meaning that the origin of coordinates is in the lower-left corner. For that reason, a transformation of coordinates was needed. The XML files contained information about multiple categories in the form of `tag`. For this work, some were selected, such as body, headline, photo source, figure, and caption for detection of elements of newspapers. While working on the transformation, randomized splitting of the dataset to train, validation, and test parts were also implemented.

3.2 Vector-Based Article Segmentation dataset

This dataset was also provided by the supervisor. The supervisor also provided this dataset As of this thesis's writing, the Vector-Based Article Segmentation (VBAS) dataset was not completed. Therefore, a pre-release version was used. Dataset is made by Tomáš Zach[20][2] and the dataset is planned to be made public. This version was composed out of 5 British newspapers with 96 pages. Same as the first dataset from the supervisor, it contains annotations within

[2]manuscript in preparation

XML files. However, the XML files are written in another format than that from the previous dataset. Its syntax is:

```
<pages>
  <page bbox ="0, 0,819.213013,1034.645996">
    <textbox bbox="x_0,y_0,x_1,y_1">
    Text outside of articles <\textbox>
    ...
    <news id="0"><textbox bbox="x_0,y_0,x_1,y_1">
    Article text <\textbox>
    ...
    <textbox bbox="x_0,y_0,x_1,y_1"> Article text <\textbox>
    <\news><\page>
  <page>
  ...
</pages>
```

This format leaves for newspaper element extraction only text boxes which include almost all the printed text of each page. Furthermore, each news annotation consists only of multiple text boxes without precise bounding boxes. Another problem is that some of the pages did not contain any articles, such as pages with advertisements.

Figure 3.1: Pages with advertisements and no articles

The provided dataset was in PDF format, which required transformation to bitmap files. As with the previous dataset, a transformation of the coordinates was also needed. In addition, computation of each article bounding box was also required.

3.3 PubLayNet

This dataset was made by Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes [21], and it contains nearly 360,000 images. The dataset is made primarily from biomedical works, and its ratio is approximately 0.94-0.03-0.03 for train, validation, and test splits. This leaves approximately 11,000 images each for the validation and test parts of the dataset.

Annotation files are made in compliance with COCO format. While working on this thesis, only train and validation annotation files were made public. The authors said the reason behind this decision is that the dataset is planned to be used for future competitions. The work on this dataset was not the primary goal of this thesis, and it is used mainly for pre-learning weights. As such, no split for a test part was done. The annotations are composed out of five categories which are text, title, list, table, and figure.

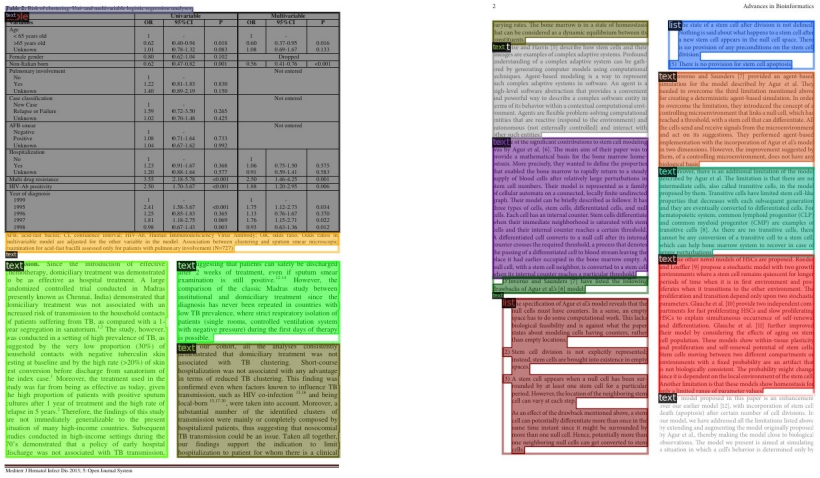


Figure 3.2: Visualization of annotation in PubLayNet dataset

3.4 Dataset preprocessing

As written in the previous section, the provided datasets required additional processing. Such as conversion of annotation XML files and conjunction of datasets.

3.4.1 Description of COCO annotation format

Common object in context (COCO)[3] is a vast dataset used for multiple tasks. This annotation format was chosen for two reasons:

1. it is commonly used by many authors for object detection tasks;
2. it was used for PubLayNet dataset.

It is stored in JavaScript object notation (JSON) format. Its basic structure is:

```
{
"info"          : info,
"licenses"      : [license],
"images"        : [image],
"annotations"   : [annotation],
"categories"    : [category],
}
```

The info section contains information about the dataset, such as contributor, description, and others. Because the dataset provided by the supervisor can not be published, the info section is not essential. The license section is similar to the info section in that way and contains different licensing details. Both license and info sections are not mandatory.

Another major section is `categories`, which is a list of multiple dictionaries of `category`, which are defined as:

```
category{
"supercategory" : str,
"id"            : int,
"name"          : str,
}
```

`Supercategory`[4] can be implemented. However, in this thesis, it was found to be unnecessary. Across all categories, `id` must be unique, as it is then referenced by annotation.

[3] `https://cocodataset.org/`

[4] For example super-category could be a vehicle, and then its categories would be bicycle, car, bus, etc.

The next major section is `images`. Same as `categories`, it is made of a list. The structure of each `image` is:

```
image{
"file_name" : str,
"height"    : int,
"width"     : int,
"id"        : int,
}
```

The most important one is the `id`, which must be unique for each image. This `id` is then referenced by the last section annotation.

The last major section is `annotations`. It is, as well as the two previous two major sections, list The format of the `annotation` depends on the task. In this thesis, each `annotation` follows a structure:

```
annotation{
"segmentation"  : [[polygon]],
"area"          : float,
"iscrowd"       : 0 or 1,
"bbox"          : [x,y,width,height],
"image_id"      : int,
"category_id"   : int,
"id"            : int,
}
```

`Segmentation` is a list of lists[5]. Each list consists of vertices of a polygon, which are stored with x coordinates in odd positions and y in even. The `area` is an area of the mentioned polygon or polygons. `Iscrowd` refers to the possibility that there are many instances of the same object. It is used when instances of the same class are too close to each other(crowds) in an image to be annotated by hand. For usage in this thesis, it was always set as 0. The `bbox`(bounding box) is given by its top-right corner coordinates, width, and length. `Image_id` refers to the image for which the annotation was written. `Category_id` is referring to the category of annotated object. Finally, `id` is a unique id of each annotation.

[5]For cases when an annotated object is interrupted in multiple parts. Though, it was not encountered in this thesis

3.4.2 Creation of datasets

From the provided datasets, multiple preprocessed datasets were created. Both datasets need common techniques for creation. Firstly, the transformation of coordinates was needed. The XML files have origin in the lower-left corner and COCO require the origin located upper-left corner. In addition, the XML file coordinates values do not necessarily correspond to the location of bitmap files. This is due to the PDF origin of the datasets. Therefore, a correction constant k was needed. The transformation formula for the origin of the bounding box is

$$x_C = k \cdot x_0 \tag{3.1}$$
$$y_C = H - (k \cdot y_0 + h) \tag{3.2}$$
$$k = \frac{H}{y_m} \tag{3.3}$$

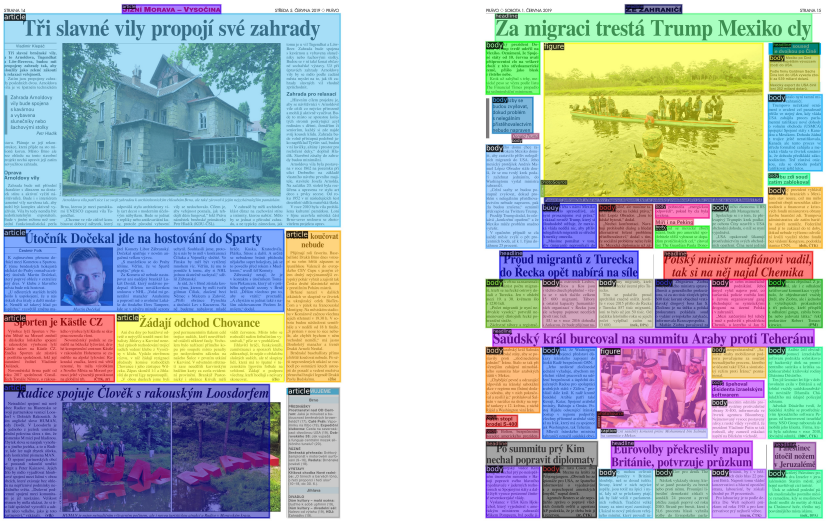
where y_m is the upper y coordinate of the upper-left corner of each page in XML files, x_C and y_C are coordinates needed by COCO (upper-left corner), x_0 and y_0 XML coordinates (bottom-left corner), H is the height of the image, and h is the height of the bounding box. Additionally, width w was also needed for COCO annotation. They are computed as

$$h = k \cdot (y_1 - y_0) \tag{3.4}$$
$$w = k \cdot (x_1 - x_0) \tag{3.5}$$

where x_1 and y_1 are the location of top-right corner of XML bounding box.

The first custom dataset was created out of the Právo dataset for article annotation (figure 3.3a). Additionally, due to the format of XML annotation, another dataset was possible to create. This annotation was made out of elements of newspapers. These were text, headline, caption, photo source, and figures (figure 3.3b).

Another dataset type was created by merging both Právo and VBAS datasets. The problem with the VBAS dataset was that the articles were not given by a bounding box but as a list of text boxes. The easiest solution was chosen. The final article (figure 3.4a) bounding box was created from the maximal and minimal values of these bounding boxes in both axes. A similar solution was used on the Právo dataset. However, due to the incompatible XML annotation format, only one category could be defined for element extraction. In the VBAS dataset, all text was tagged as a body (figure 3.4b). In conjunction, Právo dataset elements, text, headline, caption, and photo source were designated as the body.



(a) : Articles **(b) :** Elements

Figure 3.3: Visualization of possible annotation on Právo dataset

Another encountered problem was segmentation. It was not declared at all in the provided annotations. Due to time restraint and as the primary output declared by the supervisor were bounding boxes[6] segmentation was declared the same as bounding boxes. This is not great for article extraction as part of the segmentation could interfere with another article. For example, "L" shaped articles (figure 3.5) will have the smaller rectangle marked as part of the segmentation mask. Furthermore, this rectangle could be part of another article. With elements, this method is more tolerable as each element category is mostly made from rectangles, their sizes are smaller than articles, and the elements mostly do not overlap.

Multiple derived datasets were created. An overview of them can be seen in table 3.1. The two most common split ratios of training-validation-test datasets are 0.8-0.1-0.1 and 0.7-0.15-0.15. However, depending on the size of the dataset, different ratios can be used[7]. Due to the small size of both provided datasets, 0.8-0.1-0.1 ratio was chosen for many derived datasets. The selection of pages for validation and test parts of the datasets was made at random. Although not completely random, when merging two datasets, the validation and test parts were made with equal proportional representation. Information about the derived dataset are in table 3.1.

[6]This was because outputs of other works, in this task of article extraction, were the bounding boxes

[7]For example, PubLayNet has split ration of 0.94-0.03-0.03

(a) : Articles **(b) :** Elements

Figure 3.4: Visualization of possible annotation on VBAS dataset. As can be seen on the left only some articles has been classified as articles.

Figure 3.5: Example of page with "L" shaped article. The segmentation mask of larger article is covering the smaller one

name	origin	task	split			usage
			No train	No validation	No test	(section)
PrVBAS	Právo+VBAS	AS	279+76	35+10	35+10	5.2
Pr1	Právo	AS	279	35	35	5.2
Pr0.1	Právo	AS, ETP	28	35	35	5.3
Pr0.2	Právo	AS, ETP	56	35	35	5.3
Pr0.3	Právo	AS, ETP	84	35	35	5.3
Pr0.4	Právo	AS, ETP	112	35	35	5.3
Pr0.5	Právo	AS, ETP	140	35	35	5.3
Pr0.6	Právo	AS, ETP	168	35	35	5.3
Pr0.7	Právo	AS, ETP	195	35	35	5.3
Pr0.8	Právo	AS, ETP	223	35	35	5.3
Pr0.9	Právo	AS, ETP	251	35	35	5.3
Pr1.0	Právo	AS, ETP	279	35	35	5.3
VBAS1	VBAS	AS	0	0	96	5.3

Table 3.1: Description of derived dataset. AS in task stands for article segmentation and ETP is element type prediction. All the PrX.X datasets have common validation and test splits, The X.X refers to the proportion of used dataset for training

Chapter 4

Aproach

In this chapter, the implementation will be described, and another technique needed as well as other work, for instance, on datasets. In addition, the metrics used for evaluation will be described.

4.1 Evaluation metric

Many recent papers are showing the results for models with COCO evaluation. COCO uses two main metrics for detection evaluation, firstly average precision (AP) and secondly, average recall (AR). For a better understanding of these metrics, few terms must be defined.

Precision indicates the accuracy of the predictions. The recall is the ratio of detecting how many of all wanted instances were predicted. Precision and recall are calculated as

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

Where TP stands for True Positive, FP is False Positive, and FN is False Negative. Their definition can be understood from table 4.1. As can be seen from equations (4.1) and (4.2), the values of precision and recall lie between 0 and 1. They are then used for precision-recall curve.

		Actual	
		positive	negative
Prediction	positive	TP	FP
	negative	FN	TN

Table 4.1: Binary classification

Another important term is the intersection over union(IoU)[1]. It measures how much the predictions overlap with the ground truth. It is calculated as:

$$\text{IoU} = \frac{A_{GT} \cap A_{Pr}}{A_{GT} \cup A_{Pr}}. \tag{4.3}$$

Where A_{GT} is the area of ground truth, and A_{Pr} is the area of prediction. COCO uses it as a threshold for determining whether a prediction is positive or negative.

The mathematical definition of AP is

$$\text{AP} = \int_0^1 p(r)\, dr \tag{4.4}$$

where p is a function of precision depending on recall r. Meaning, it is an area under the precision-recall curve. This curve is given by pairs of precision and recalls, which are obtained by changing the the threshold of the confidence score of the predictions. In practice, interpolation with rectangles is used as can be seen in figure 4.1. AP is then

$$\text{AP} = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{\text{interp}}(r_{i+1}), \tag{4.5}$$

where $r_1, r_2, ..., r_n$ are recall levels.

COCO evaluation uses these terms to calculate multiple metrics. The main one is AP[2], and it is the mean average over all categories. Additionally, it calculates the mean from 10 IoU thresholds from 0.5 to 0.95 with 0.05 steps. It is the primary metric used for the evaluation of COCO challenges. Metric AP50 is identical to the older PASCAL VOC metric. It has the IoU threshold of 0.5. Depending on the application, the IoU threshold of 0.5 may be too benevolent. Of course, the IoU threshold can be chosen freely by the author. However, for additional comparison between implementations, an additional metric was introduced, AP75. It has, as its name suggests, IoU threshold of 0.75.

[1] Also it can be encountered as Jaccard index

[2] Sometimes can be encountered as mAP (mean average precision) or as AP@[.5:.05:0.95]

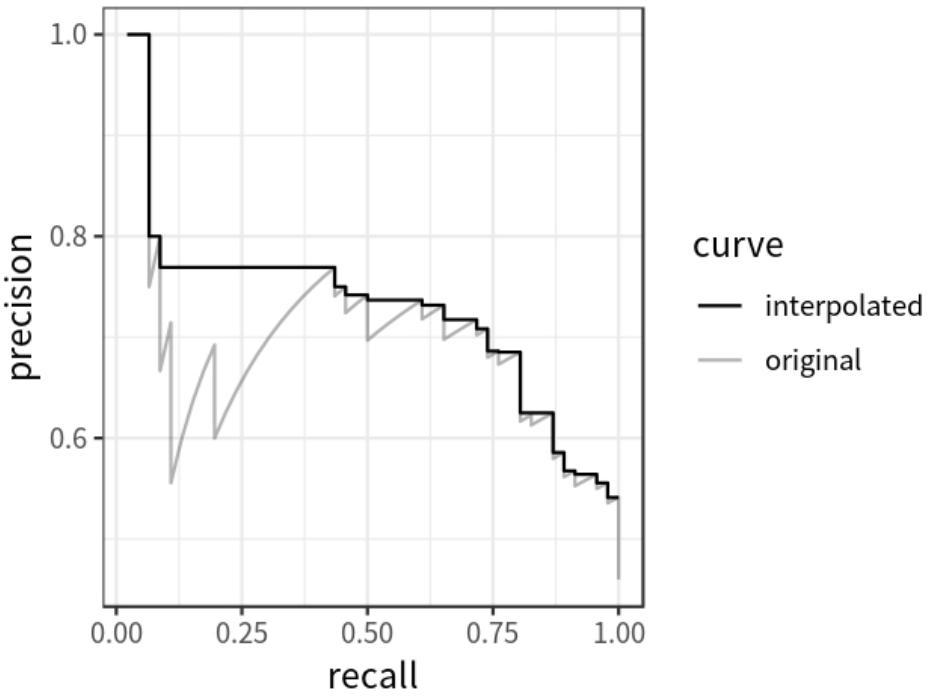


Figure 4.1: Precision-recall curve and its interpolation [14]

Another used metrics by COCO evaluation are AP Across Scales. They are defined as:

- APs for small objects with area $< 32^2$;
- APm for medium objects with $32^2 <$ area $< 96^2$;
- APl for large objects with area $>96^2$.

In this thesis, small objects will probably not be encountered while working on article segmentation. Element type prediction will most likely encounter all sizes.

Furthermore,in the case of element type prediction, AP is calculated for each category. It is then written as AP-C, where C stands for the object category. In this thesis, four will be encountered:

- AP-b : body, meaning mostly paragraphs;
- AP-h : headline;

- AP-c : caption;
- AP-p : photo-source.

As only one class is detected with article segmentation, this metric is not shown due to its redundancy. All AP metric values are in the range of 0-1.

The secondary metric is AR. It is not commonly used for evaluation and comparison, although it can be useful during training.

4.2 Implementation

Code for this thesis was written in Python, as it is one of the most popular programming languages. Additionally, it is commonly used for machine learning. Most of the calculations were made on RCI Cluster[3].

4.2.1 Conversion of PDFs

Although the Právo dataset was provided with the already existing images, the VBAS dataset was originally in PDF format. Therefore, the conversion was needed, for which python library `pdf2image` was used and, more specifically, its function `convert_from_path`. The size of the output images was arbitrarily chosen to be 2276x2875 pixels. The converted images are readable by the human eye while keeping the size of the pictures relatively small.

4.2.2 Conversion of XML files with annotations

As was written in section 3.4.2, the conversion of the annotations was needed. Python script, `XMLtoCOCO.py`, was implemented for this task. It is capable of combining data from both provided datasets, Pr and DP. The main principle of this script is to parse XML files and extract the needed annotation data. The script was implemented with two major functions. One was used for extracting the data from Právo, the other from VBAS. As all used IDs do

[3] `http://rci.cvut.cz/`

not need to be encrypted, the easiest solution was chosen; indexing always started at 0. The exception was made with categories, where the specific indexing was chosen. It can be seen in table 4.2.

Task	object type	ID
Article segmentation	article	1
Element type prediction	body	1
	headline	2
	caption	3
	photo-Source	4

Table 4.2: Indexing of categories

In the Právo dataset, each image had a corresponding XML file. Its structure was described in section 3.1. The function took each line of text as a string, splitted it by space, and checked the line for the wanted strings. These were `textbox`, `figurebox`, and `news`. Additionally, when `page` was detected, the page/image was added to the `images` dictionary. If the desired string was found, the bounding box was searched for and subsequently transformed into the required coordinates by using equations (3.1), (3.2), (3.4), and (3.5). The indexing of categories for element type prediction was possible due to the existence of feature `tag` in each `textbox`.

In contrast to the Právo dataset, XML files for the VBAS dataset were made for each PDF, which consisted of multiple pages. This led to the need to detect additional string `page` for separating annotations for each page. Furthermore, due to the structure of these XML files, element type labels are missing, and therefore element type prediction could only have one class, body.

4.2.3 Library detectron2

Made by the Facebook AI Research group [19], `detectron2` is a Python library distributed under Apache 2 license. It allows relatively easy usage of state-of-the-art networks for object detection. Library requires `PyTorch` version 1.6 or greater and a corresponding version of `torchvision`. Additionally, `OpenCV` is not mandatory. However, it is used for visualization. This library proved to be very valuable as it had many needed utilities.

4.2.4 Main program overview

The main script can be divided into multiple sections, which will be now described.

Parsing arguments

Because the training of models usually takes hours, and because the RCI Cluster has the ability to submit multiple jobs via Slurm[4] batch files, running programs with arguments, which could be changed, was found useful. Implementation of argument parsing was done with Python library `argparse`.

Models and their configuration

The `detectron2` library allows easy usage of pre-made state of the art models. In this thesis, Mask R-CNN was used with varying backbones. These models can then be trained and fine-tuned on the desired dataset.

The main parameters for the configuration were:

- learning rate
- maximal number of epochs
- number of images per batch
- evaluation period
- number of classes

Training

Same as with models, `detectron2` already has predefined trainers. However, existing trainers are basic by design. One of the missing features of

[4] `https://slurm.schedmd.com/`

the default trainer is a computation of validation loss. Therefore, class `CustomTrainer` was implemented as child class of `DefaultTrainer`. It can be seen in `CustomTrainer.py`

This class had a custom hook made, which can be seen in `LossEvalHooks.py`. This hook was inherited from class `HookBase`. Its implemented methods were greatly inspired by `detectron2` files `evaluator.py` from the evaluation part and `train_loop.py` from the engine part.

Evaluation

Evaluation metric was implemented with the usage of `COCOEvaluator`, a function of `detectron2`. The output metrics are in the range of 0-100 instead of mathematically correct 0-1. One way to look at it is as percentages, as almost all encountered works showed results in the same way. This is done probably due to better readability.

Visualization

Visualization also used `detectron2` capability, `Visualizer`. Its two functions `draw_instance_prediction` and `draw_dataset_dict` were used for visualization. The former was used for visualizing annotation, while the latter was used for visualizing the predictions. After visualizing both instances, they were concatenated along the side vertical axis, with the annotation image being always on the left. Implementation of merging images can be found in python script `save_visualization.py` The images to be visualized were normally chosen at random. However, except for images without annotation. These images were usually used for comparison on never seen newspapers. Furthermore, the annotation image was replaced by the original image.

Supporting functionalities

Saving the results of training and evaluation is necessary. For this, a python function `write_txt_output` was written. It was used for saving a dictionary with the data. Aside from the results, input arguments and additional data (e.g., learning time) were also saved to the dictionary.

Chapter 5

Experiments

In this chapter, multiple experiments will be described. The main motivation is the exploration of the capabilities of the models while training on different datasets.

5.1 Pre-training models

Due to the small size of the newspaper datasets, transfer learning was used. This method uses the weights which were trained on a large dataset and uses them for training on a dataset used for application [17]. PubLayNet was chosen for this due to the substantial size and relative similarity of article segmentation and document layout analysis tasks.

Two models were pre-trained. Both models were based on Mask RCNN. One had backbone ResNet-50-FPN and the other ResNet-101-FPN. These models were already implemented in `detectron2`. Training parameters of these models are described in table 5.1. And their last validation AP metrics are in table 5.2

Both models show high AP metrics. Although the second one has higher values for experiments, the first one was chosen due to time restraint.

	Model	
parameter	R50	R101
learning rate	0.00025	0.00025
batch size	512	512
epochs	15000	15000

Table 5.1: Models and parameters for their pretraining

Task		Bounding box		Segmentation	
Model		R50	R101	R50	R101
Evaluation metric	AP	83.5	85.8	81.0	82.9
	AP50	94.8	95.4	94.6	95.3
	AP 75	90.5	91.8	89.3	90.9
	APs	24.1	25.2	23.5	24.4
	APm	56.0	59.8	53.3	56.8
	APl	88.2	90.8	85.3	87.3
	AP-te	90.9	91.5	90.7	90.5
	AP-ta	91.3	93.6	91.8	93.3
	AP-ti	78.3	77.6	76.3	74.5
	AP-f	85.7	89.2	84.5	88.6
	AP-l	71.4	77.1	61.7	66.8
Training time		11:03:05	13:01:23	11:03:05	13:01:23

Table 5.2: Last AP validation metric of models on two backbones ResNet-50-FPN (referred as R50) and ResNet-101-FPN (referred as R101), trained on PubLayNet[21].

5.2 Merging datasets

As two relatively small newspaper datasets were provided, merging them was seen as a tempting possibility. In addition to the size increment, it also introduces a more diverse environment. The disadvantage of the merged dataset is that the VBAS dataset had only some articles annotated, as can be seen on figure 3.4b. Therefore, no model was trained on just the VBAS dataset. However, the VBAS dataset was used for evaluation in the following experiment.

In this experiment, the same model will be trained on two datasets Pr1 and PrVBAS (see table 3.1 for explanation), and then each trained model will be evaluated on test split of both PrVBAS and Pr1. The expected result is that the dataset with more diverse newspapers should outperform the one with less diversity. The model parameters were the same for both trainings. Learning rate 0.00025, number of iterations 1500, batch size 512.

Task	Train	Eval	Evaluation metric AP	AP50	AP75	APs	APm	APl
Bounding box	Pr1	PrVBAS	87.25	93.60	90.67	NaN	100.0	87.2
	PrVBAS	PrVBAS	**87.95**	**95.28**	**90.92**	NaN	100.0	**87.86**
Segmentation	Pr1	PrVBAS	88.19	94.03	91.05	NaN	100.0	88.18
	PrVBAS	PrVBAS	**88.93**	**95.28**	**91.16**	NaN	100.0	**88.85**

Table 5.3: Results of article segmentation for a models trained on Pr1 and PrVBAS dataset. Evaluation computed on test-split of PrVBAS dataset. As can be seen, the model trained on PrVBAS, had better results in all cases. Value NaN for APs means that the computation of metric could not be done due to the lack of small object.

Task	Train	Eval	Evaluation metric AP	AP50	AP75	APs	APm	APl
Bounding box	Pr1	Pr1	**90.71**	**95.71**	**94.35**	NaN	100.0	**90.72**
	PrVBAS	Pr1	89.97	95.70	93.37	NaN	100.0	89.98
Segmentation	Pr1	Pr1	**91.65**	**95.76**	**94.35**	NaN	100.0	**91.67**
	PrVBAS	Pr1	90.91	95.70	93.37	NaN	100.0	90.91

Table 5.4: Results of article segmentation for a models trained on Pr1 and PrVBAS dataset. Evaluation computed on test-split of Pr1 dataset. As can be seen, the model trained on Pr1, had better results in almost all cases. Value NaN for APs means that the computation of metric could not be done due to the lack of small object.

The results for evaluation on PrVBAS are in table 5.3 and for evaluation on Pr1 5.4. As described in section 4.1, AP metric is considered the most important. From the result tables, we can see that both models did better when the evaluation was done on their own test splits.

As expected, it can be seen that the model trained on a more diverse dataset outperformed the model trained on just one dataset. However, the differences are not substantial. Additionally, it can be concluded that both models performed better on their respective test splits. Although for a more conclusive answer, bigger and more diverse datasets would be needed.

5.3 Size of dataset

As it was written in chapter 3, training datasets are important for many ML algorithms. Commonly, the more extensive the dataset is, the better. Furthermore, for CNNs to have wider application, more diverse datasets are usually preferred. Their creation is however costly and time-consuming due

to handcrafted annotation or not exactly trivial algorithms[1]. As so, there is a dilemma for how big the dataset must be. This experiment will try to determine a trend or possible prediction of the needed size. On the other hand, if the datasets are too big, the computation will take more time and allocated resources. This consequently means that the unnecessary cost, which could be avoided.

For this experiment, multiple sub-datasets were created (for the explanation, consult table 3.1). The main methods from 3.4.2 were used but upgraded. After each wanted portion of the dataset, the annotation was saved. This was made because the validation and test parts of the dataset were selected randomly. Moreover, for the correct comparison, the test parts must be the same for all sub-datasets as well as the validation part. The model parameters were the same for both trainings. Learning rate 0.00025, number of iterations 1500, batch size 512. All datasets had the same validation and test split.

Description of the used metrics is in section 4.1. The article segmentation results are in 5.5. No clear leader exists. Although some trends can be seen with the APs and APm, as with the decreasing size of the train split, the model could not properly detect the small and medium-sized objects. This is probably due to the small representation of small and medium-sized objects in this dataset. An additional evaluation was done on dataset VBAS1 and its results can be seen in table 5.6. They were also inconclusive. There, the models had nearly the same values. This is probably due to the lack of annotation in the VBAS dataset

Additionally, the same experiment was done on the element type prediction task. The results are in tables 5.7, and 5.8. Same as before, the differences are minimal.

To summarize, this experiment did not lead to any specific conclusion, as no clear trend was found. It would be possible to continue the work. Adding new smaller datasets (such as Pr0.05, Pr0.01, etc.), training them, and evaluating. For the possibility of finding the breaking point, from which training would lead nowhere.

[1] One such algorithm was used for the creation of PubLayNet[21].

Task	Dataset	Size	Evaluation metric						RunTime
			AP	AP50	AP75	APs	APm	APl	
Bounding box	Pr1.0	279	84.14	89.44	86.76	5.05	**8.77**	87.38	2:47:51
	Pr0.9	251	83.98	89.35	86.93	2.24	3.79	87.39	2:48:10
	Pr0.8	223	84.43	89.5	87.08	4.21	4.03	87.77	2:49:24
	Pr0.7	195	83.86	**89.81**	86.82	5.05	3.57	87.28	2:48:58
	Pr0.6	168	83.75	89.11	85.89	**8.42**	0.64	87.67	2:48:20
	Pr0.5	140	**84.36**	89.04	**87.29**	5.05	1.03	**88.04**	2:49:26
	Pr0.4	112	82.83	87.55	85.64	0.0	0.0	86.98	2:46:55
	Pr0.3	84	83.19	88.33	86.21	0.0	1.72	87.14	2:49:37
	Pr0.2	56	81.66	86.64	83.42	0.0	0.0	85.75	2:45:30
	Pr0.1	28	81.56	86.64	84.77	0.0	0.0	85.68	2:44:34
Segmentation	Pr1.0	279	84.53	89.44	86.48	3.37	**7.87**	87.93	2:47:51
	Pr0.9	251	84.88	89.35	85.9	1.68	3.64	88.35	2:48:10
	Pr0.8	223	**85.11**	**89.97**	**86.69**	4.21	3.82	**88.67**	2:49:24
	Pr0.7	195	84.81	89.81	86.04	**4.49**	3.32	88.27	2:48:58
	Pr0.6	168	84.11	89.11	84.83	**4.49**	0.84	88.15	2:48:20
	Pr0.5	140	84.49	89.04	86.35	2.4	1.36	88.18	2:49:26
	Pr0.4	112	83.77	87.55	85.46	0.0	0.0	88.0	2:46:55
	Pr0.3	84	83.51	88.98	86.03	0.0	2.04	87.37	2:49:37
	Pr0.2	56	82.03	86.64	84.18	0.0	0.0	86.04	2:45:30
	Pr0.1	28	81.64	86.64	83.88	0.0	0.0	85.82	2:44:34

Table 5.5: Results of article segmentation on PrX.X test split, which is the same for all dataset. Trained on datasets with different sizes. No specific model was significantly outperforming the rest. Some trend can be seen with APs and APm, where a smaller train split could not detect objects at all. This is possibly due to small representation of small and medium objects in respective splits dataset.

Task	Dataset	Size	Evaluation metric					
			AP	AP50	AP75	APs	APm	APl
Bounding box	Pr1.0	279	32.91	50.11	35.08	NaN	NaN	32.91
	Pr0.9	251	32.89	50.04	35.06	NaN	NaN	32.9
	Pr0.8	223	32.9	50.11	35.08	NaN	NaN	32.91
	Pr0.7	195	32.89	50.04	35.07	NaN	NaN	32.9
	Pr0.6	168	32.89	50.04	35.06	NaN	NaN	32.9
	Pr0.5	140	32.91	50.11	35.08	NaN	NaN	32.91
	Pr0.4	112	32.89	50.04	35.06	NaN	NaN	32.9
	Pr0.3	84	32.89	50.04	35.06	NaN	NaN	32.9
	Pr0.2	56	32.9	50.04	35.08	NaN	NaN	32.9
	Pr0.1	28	32.91	50.11	35.08	NaN	NaN	32.91
Segmentation	Pr1.0	279	32.2	47.77	35.06	NaN	NaN	32.21
	Pr0.9	251	32.2	47.76	35.05	NaN	NaN	32.21
	Pr0.8	223	32.2	47.77	35.06	NaN	NaN	32.21
	Pr0.7	195	32.15	47.76	35.05	NaN	NaN	32.16
	Pr0.6	168	32.15	47.76	35.05	NaN	NaN	32.16
	Pr0.5	140	32.2	47.77	35.06	NaN	NaN	32.21
	Pr0.4	112	32.2	47.76	35.05	NaN	NaN	32.21
	Pr0.3	84	32.15	47.76	35.05	NaN	NaN	32.16
	Pr0.2	56	32.2	47.76	35.06	NaN	NaN	32.21
	Pr0.1	28	32.2	47.77	35.06	NaN	NaN	32.21

Table 5.6: Results of article segmentation. Trained on datasets with different sizes. Evaluation was done on dataset never seen by these models, VBAS. NaN value means that metric could not be counted. This is due to the absence of the small and medium sized objects in this dataset. As can be deducted outcome of all the trained models is nearly identical after rounding. This is probably due to lack of more annotation in original VBAS dataset. Additionally, this dataset is lacking smaller articles, which, as seen in table 5.5, made the only noticeable difference between model.

Task	Dataset	Size	Evaluation metric						RunTime
			AP	AP50	AP75	APs	APm	APl	
Bounding box	Pr1.0	279	**73.22**	89.08	82.22	24.38	45.72	84.61	3:11:51
	Pr0.9	251	71.93	88.99	77.81	22.54	43.22	84.61	3:19:00
	Pr0.8	223	73.07	**90.13**	80.38	26.08	47.54	84.34	3:19:26
	Pr0.7	195	72.06	88.36	80.3	26.2	42.89	84.39	3:08:08
	Pr0.6	168	71.48	89.07	74.91	24.23	41.59	**85.13**	3:08:31
	Pr0.5	140	71.81	88.36	77.3	**30.14**	45.95	84.42	3:17:34
	Pr0.4	112	72.61	89.09	78.97	24.98	**50.15**	84.43	3:02:05
	Pr0.3	84	72.86	89.01	82.25	19.55	44.6	84.28	3:01:59
	Pr0.2	56	73.03	89.0	**82.47**	21.3	43.56	84.45	3:06:52
	Pr0.1	28	69.59	86.97	75.73	16.41	43.05	83.53	3:13:16
Segmentation	Pr1.0	279	73.75	89.39	82.22	20.09	44.37	84.83	3:11:51
	Pr0.9	251	72.53	89.0	79.49	21.96	44.15	84.3	3:19:00
	Pr0.8	223	73.83	**90.13**	82.79	22.88	47.7	84.3	3:19:26
	Pr0.7	195	73.7	88.37	81.94	22.06	44.39	**84.92**	3:08:08
	Pr0.6	168	72.07	89.01	78.23	24.38	42.36	84.45	3:08:31
	Pr0.5	140	72.49	88.37	78.78	**25.82**	45.69	84.68	3:17:34
	Pr0.4	112	73.05	89.08	79.74	21.16	**49.4**	84.37	3:02:05
	Pr0.3	84	**74.22**	89.24	**83.5**	18.99	43.92	84.55	3:01:59
	Pr0.2	56	73.65	89.27	81.51	21.42	43.87	84.77	3:06:52
	Pr0.1	28	70.65	87.33	76.38	14.99	43.95	83.88	3:13:16

Table 5.7: Results of element type prediction. Training on different sized datasets. All of these trained models had common test and validation splits. The results are very similar to those in table 5.8 for article segmentation, as no clear winner can be chosen. We can see that the trained models are focusing more on detection of larger objects. This is presumably because of the larger representation of these objects.

Task	Dataset	Size	Evaluation metric				RunTime
			AP-b	AP-h	AP-c	AP-p	
Bounding box	Pr1.0	279	**87.65**	76.24	80.46	48.54	3:11:51
	Pr0.9	251	87.24	**77.89**	79.0	43.58	3:19:00
	Pr0.8	223	87.77	77.64	78.53	48.34	3:19:26
	Pr0.7	195	86.19	76.97	80.51	44.59	3:08:08
	Pr0.6	168	**87.65**	76.73	**81.36**	40.16	3:08:31
	Pr0.5	140	87.23	76.98	79.78	43.24	3:17:34
	Pr0.4	112	87.07	77.34	79.97	46.06	3:02:05
	Pr0.3	84	87.05	77.37	77.77	49.27	3:01:59
	Pr0.2	56	87.22	75.14	80.28	**49.48**	3:06:52
	Pr0.1	28	86.18	75.6	78.5	38.08	3:13:16
Segmentation	Pr1.0	279	86.73	76.97	81.01	50.29	3:11:51
	Pr0.9	251	**86.86**	**77.82**	78.57	46.85	3:19:00
	Pr0.8	223	86.84	77.5	79.38	**51.6**	3:19:26
	Pr0.7	195	86.54	77.28	81.1	49.88	3:08:08
	Pr0.6	168	86.66	75.59	**81.45**	44.56	3:08:31
	Pr0.5	140	86.68	76.88	80.74	45.65	3:17:34
	Pr0.4	112	85.99	77.11	81.06	48.02	3:02:05
	Pr0.3	84	86.48	77.3	79.45	53.67	3:01:59
	Pr0.2	56	86.59	75.65	81.26	51.1	3:06:52
	Pr0.1	28	85.38	75.65	79.96	41.61	3:13:16

Table 5.8: Results of each category of element type prediction. This table is continuation of table 5.7. All trained models had model while training on datasets with different sizes. From these values, the best model can not be selected. The smallest metric values are those for category photo-source. This can certainly correlate to table 5.7, as this category is usually the smallest. Metric used is AP for each body, headline, caption and photo-source.

Chapter 6

Discussion

In this chapter, the results of the experiments will be discussed, and future additions and improvements of the experiments will be outlined. Additionally, it will be shown how few selected models behave on newspapers never before seen.

6.1 Discussion of the experiments

The experiment described in section 5.2, where behavior on different variable datasets was explored, did not lead to a conclusive answer. We were only able to confirm that each trained model was better than the other one when the evaluation was done on its own test split. Further research could be made with a more diverse dataset. For example, in the case of having 3 different brands of newspaper, 2 could be merged for training and the third used for evaluation and repeating it with different combinations.

The experiment described in section 5.3, where we were trying to determine the size of the train split on which the model would not be able to learn, also did not lead to a conclusive answer. As no trained model significantly and consistently beat the rest. However, from the experiment in 5.3 we can conclude that transfer learning is a powerful tool. Even on the dataset with 28 images, the trained model did not lack behind. This shows the power of transfer learning approach. This means that in cases when annotation creation is expensive, a small dataset would be found sufficient. The continuation

of this experiment could be in the form of further decreasing the number of images in the training dataset. Additionally, a cross-validation could be implemented for better evaluation. However, this would lead to an increase in computation time.

6.2 Showcase on never-seen newspapers

Besides all the numbers, it is good to know how trained models work on never-seen data. Therefore, some newspapers were scraped from the internet. Among them were Katolický týdeník [1], a Czech catholic newspapers, Pečujeme doma[2], Czech newspapers about social services, and The Sun[3], a British tabloid.

All scrapped newspapers were in PDF format, which led to the need of conversion to images. Because no annotation exists, no metric can be calculated. Therefore, we present a visual check only.

6.2.1 Article segmetation

The used model is the same as in the experiment described in 5.2, the one trained on PrVBAS dataset. Furthermore, only predictions with confidence greater than 0.75 were visualized.

Katolický týdeník was made with a similar newspaper structure as the newspapers from the datasets. Visualization of the predictions can be seen in figure 6.1a. The predictions were reasonable. On Pečujeme doma, the behavior (figure 6.1b) was similar to the previous newspaper. One exception was with headlines; if the headline was made from two lines, only the closer one to the text was detected as the article.

The last and the most interesting newspaper was the tabloid The Sun, as segmenting articles from tabloids is the ultimate goal of multiple works under the supervisor Ing. Jan. Drchal, Ph.D. Tabloids are usually harder for this

[1] `https://www.katyd.cz/`
[2] `https://www.mskruh.cz/publikace/noviny-pecuj-eme-doma`
[3] `https://www.thesun.co.uk/`

(a) : Katolický týdeník **(b) :** Pečujeme doma

Figure 6.1: Visualization of article segmentation on Czech newspapers.

task as their layouts are more complex due to the usage of non-rectangular article boundaries, colorful design, tilted paragraphs, etc.

The results are not surprising, on some pages (e.g. figures 6.2a and 6.2b), the prediction are solid. Although the same problem as with previous newspapers was encountered, the incorrect detection of headlines. In some pages (e.g. figures 6.2c and 6.2d), not all articles were detected. Furthermore, in some pages articles were detected poorly or not at all (figure 6.3).

To conclude, the model can be used on different brands and types of newspapers. Although, few problems were encountered, such as detecting advertisements and pictures as articles and missing the whole or parts of the articles. Additionally, from all predictions on tabloid The Sun, it can be deduced that the model is quite dependent on the headlines of the article. However, this all is understandable, as the training dataset was quite small and made only from two brands of newspapers with a rigid layout. Additionally, the dataset used for pre-training, even though large, was made from scientific documentation, which mostly follows a rigid structure.

(a) : Falsely detected article

(b) : One page article

(c) : Missing article on the left

(d) : Missing article with the swimmer

Figure 6.2: Visualization of article segmentation on British tabloid The Sun.

6.2.2 Element type prediction

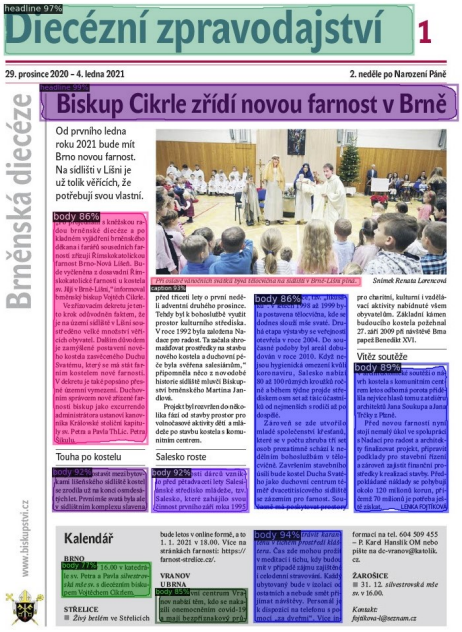
The model used is from section 5.3. The model trained on Pr1.0 dataset was arbitrarily chosen. Additionally, only those objects with confidence greater than 0.75 were visualized (figure 6.4).

(a) : Detecting only part of the article

(b) : Detecting articles, which should be part of the larger article

Figure 6.3: Visualization of article segmentation on British tabloid The Sun.

In the images, where the structure is similar to those from the Právo dataset, the predictions are good. However, in the case of a complex tabloid layout, the model is not doing great. This is probably due to training on only one brand of newspaper, which had a quite rigid layout.



(a) : Czech newspaper Katolický týdeník

(b) : Czech newspaper Pečujeme doma

(c) : Brirish tabloid The Sun

(d) : British tabloid The SUN

Figure 6.4: Visualization of element type prediction on newspapers.

Chapter 7

Conclusion

Firstly, I studied state of the art methods for article segmentation, as well as textual document segmentation. This is due to the application similarities between newspapers and documents. The input of bitmap files partially determined the possible methods of implementation. Implementation via convolutional neural networks was chosen because the ultimate goal of multiple works, under my supervisor Ing. Jan Drchal, Ph.D., is to analyze the tabloid newspaper. As tabloids require more general approaches due to their layout, which is more complex(non-rectangular articles, colorful design, tilted paragraphs, etc.).

The article segmentation task can be interpreted as object detection. For this, two main CNN algorithms exist, one-shot algorithms such as YOLO and region-based algorithm. Region-based algorithms were chosen for their precision. Furthermore, the task does not require real-time prediction, in contrast to, for example, the automotive industry. For this task, Mask R-CNN was used as it allows the detection of the segmentation mask in the bounding box.

The code was written in Python programming language, as it is commonly used for machine learning algorithms. The networks were implemented with the usage of `detectron2`, a python library made by Facebook AI Research.

Additionally, the provided newspaper datasets were in need of preprocessing as their annotation was written in for different problem approaches. The original annotation allowed for the creation of a dataset for secondary task element type prediction. Multiple datasets were created for the experiments.

Models were trained and evaluated, and even though the training datasets were small, the final metrics are promising. This shows the power of a method called transfer learning. In addition, multiple experiments were executed. Firstly, two mask R-CNN implementations were trained on PubLayNet, a large dataset consisting of scientific documents. Two models differed in backbone architecture, and although their evaluation metrics did not differ that much, the one with worse results was chosen for additional experiments due to the training time.

In the second experiment, I tried to determine whether the merging dataset is beneficial for training. This was done by the creation of two datasets. One made out of the Právo dataset, the other by combining both Právo and VBAS datasets. The only thing that can be said is that the models have better evaluation results on the test split of the dataset they were trained on.

The idea behind the last experiment was to find the breaking point of the training split of the dataset. The testing was done on multiple datasets with varying-sized training split and common validation and test splits. This breaking point was not found. However, this led to the conclusion that transfer learning is a powerful utility, as even the small dataset made from 28 pictures was large enough to get the results.

Furthermore, a showcase of predictions on the never-seen-before three brands of newspapers was made. Two of these newspapers were similar in structure to those in the training datasets. The last one was a tabloid, which did not necessarily follow the structure. The predictions by the trained model were reasonable on the newspapers with a more rigid structure. On the tabloids, the predictions were not consistent.

In the future, cross-validation could be implemented. This could lead to a more consistent evaluation result, although cross-validation increases the computation time. Furthermore, to fully use the advantages of CNNs, a larger and more extensive dataset, both in size and diversity, would certainly be appreciated.

Appendix A

Bibliography

[1] A. Almutairi and M. Almashan. Instance segmentation of newspaper elements using mask r-cnn. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1371–1375, 2019.

[2] W. Cai, J. Li, Z. Xie, T. Zhao, and K. LU. Street object detection based on faster r-cnn. In *2018 37th Chinese Control Conference (CCC)*, pages 9500–9503, 2018.

[3] R. S. Chathuranga and L. Ranathunga. Procedural approach for content segmentation of old newspaper pages. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6, 2017.

[4] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 55(6):3322–3337, 2017.

[5] A. Cruz-Roa, H. Gilmore, A. Basavanhally, M. Feldman, S. Ganesan, N. Shih, J. Tomaszewski, F. González, and A. Madabhushi. Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent. *Scientific Reports*, 7:46450, 04 2017.

[6] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier. A comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition*, 64, 10 2016.

[7] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 09 2004.

[8] R. Girshick. Fast r-cnn, 2015.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] X.-H. Li, F. Yin, and C.-L. Liu. Page segmentation using convolutional neural network and graphical model. In X. Bai, D. Karatzas, and D. Lopresti, editors, *Document Analysis Systems*, pages 231–245, Cham, 2020. Springer International Publishing.

[12] B. Meier, T. Stadelmann, J. Stampfli, M. Arnold, and M. Cieliebak. Fully convolutional neural networks for newspaper article segmentation. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 414–419, 2017.

[13] A. Naoum. *Article Segmentation in Digitised Newspapers*. Doctor of philosophy ph.d., 2020-01-01.

[14] Nick Zeng. An introduction to evaluation metrics for object detection. `https://blog.zenggyu.com/en/post/2018-12-16/an-introduction-to-evaluation-metrics-for-object-detection/`, 2018.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2016.

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, Jun 2017.

[17] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A survey on deep transfer learning. In V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham, 2018. Springer International Publishing.

[18] J. Uijlings, K. Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104:154–171, 09 2013.

[19] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[20] T. Zach. Machine learning for news article layout extraction from pdf files. Bachelor's thesis, Czech Technical University, 2021. manuscript in preparation.

[21] X. Zhong, J. Tang, and A. J. Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, Sep. 2019.

Appendix B

List of Abbrevition

Abbrevition	Meaning
AP	Average precision
AR	Average recall
CNN	Convolutional neural network
COCO	Common object in context
FCN	Fully convolutional network
FN	False negative
FP	False positive
ML	Machine learning
OCR	Optical character recognition
PDF	Portable document format
R-CNN	Regions with CNNs features
RoI	Regions of interest
SVM	Support vector machines
TN	True negative
TP	True positive
XML	Extensible markup language
YOLO	You only look once

Appendix C

CD Content

Name	Description
Thesis.pdf	This thesis in pdf format
main.py	Main pipeline for detection
XMLtoCOCO.py	Python script for transforming
CustomTrainer	Customized trainer
LossEvalHooks	Customized hook for evaluation loss
save_vizualization.py	saving images