Document Analysis
Exercises

a.o. Univ.-Prof. Dr. Robert Sablatnig

Schedule The exercise should be written in Python. Important dates are listed below:

- 26.05.2025 Deadline of assignment 2
- 02.06.2025 Presentation of assignment 2

Submission The assignments must be uploaded to the TUWEL course. Each assignment should be zipped (name.zip) and contain the subsequently enumerated items:

- **main{i}.py** the main routine for each task
- **assignmentX.pdf** the report

All files must contain the name of all group members. The report should explain the assignments and answer all questions. It should be a comprehensive but short (e.g. 6 pages) explanation of the algorithms used, their advantages, and drawbacks. You are also expected to provide intermediate results for all the subtasks.

If you have any questions please refer to the TUWEL forum.

Assignment 2 - Layout/OCR and NLP

The following task is centered on a project recently conducted in our research group[1]. The overall goal is the preparation/indexing of selected museum holdings by the Lower Austrian State Collections. The focus for the indexing of a photo collection we will use in this assignment is text recognition (printed text) and obtaining the location written on the card for georeferencing the respective images. The images are provided by the Lower Austrian State Collections; we kindly ask you not to redistribute them.

Data We consider 63 samples of index cards of the so-called *Machura* photo collection, named after the curator Lothar Machura (1909-1982) who worked at the Lower Austria State Museum for 45 years. An example is shown in Fig. 1. One sample usually consists of German machine-printed text in the upper right corner, indicating meta-data (e.g., location, description, date, film number) about the image preview on the card. Some samples can also include handwritten text, but this should be neglected in this task. In the following, we will develop the pipeline to implement the text detection and transcribing and analyzing the text. In the project, the final output of the named entity recognition (that we do not focus on in our assignment) consists of obtaining the GPS coordinates of the location to create a digital landscape of the photographies.

Tasks The following tasks should guide you through each step of the pipeline. If you come up with ideas for improvement, feel free to try it out.

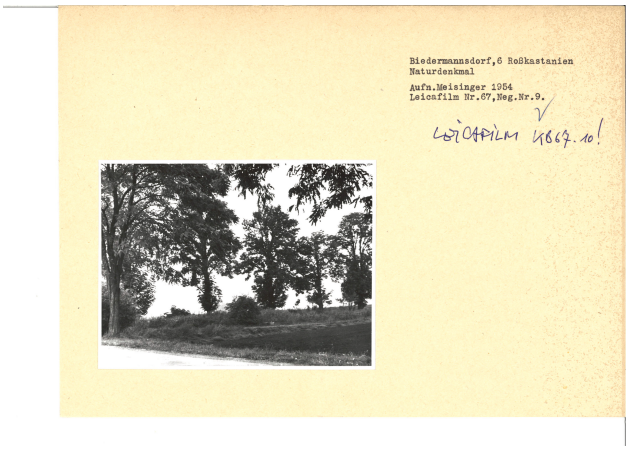

Fig. 1: Sample of the Machura photo collection.

[1] Data: `https://cloud.cvl.tuwien.ac.at/s/BmetMYW8i6qSJ29`

Task A - Layout Analysis

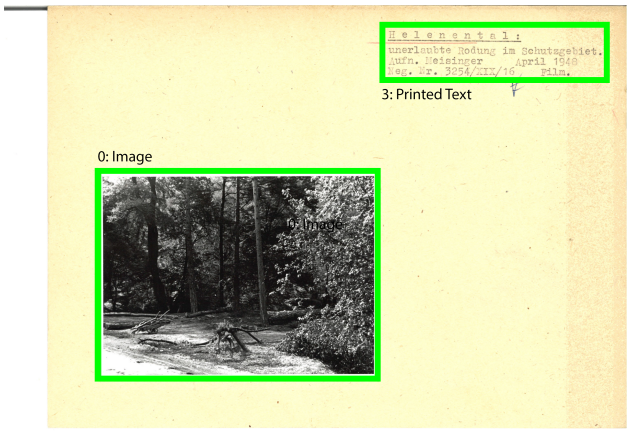

Fig. 2: Layout Analysis

Goal In the first task, we will use a simple pretrained object detection model (YoloV8) to extract the main parts of the images, an example is shown in Fig. 2. The model was trained on 100 manually annotated examples. The goal is to extract the printed text of the images.

Task Firstly, make sure you install the library `ultralytics=8.0.171` to be able to correctly load the YoloV8 checkpoint we provide. The model is trained to predict three classes

1. Image
2. Handwritten Text (HWT)
3. Printed Text (PT)

to detect the different parts of the image. For our assignment, we are only interested in class 3, the machine printed text. Loading the model is done by

```
from ultralytics import YOLO
# Load the checkpoint
model = YOLO("yolov8n.pt")
```

Results Extract the machine-printed text of the images and store it for further use. Include some qualitative examples (e.g., errors, if there are any).

Task B - OCR

Goal For further analysis, we apply a text recognition, to deal with the previously extracted images. An example of the output is shown in Fig. 3. You are free to use any OCR engine of your choice, one possibility is the use of Tesseract[2]. You can find the ground truth data in the corresponding .TXT file.

```
Helenental:
unerlaubte Rodung im Schutzgebiet.
Aufn. Meisinger
April 1948
Neg. Nr. 3254/KIX/16
Film.
```

Fig. 3: OCR output

Task You should evaluate the following options:

- Try to binarize the images on your own (e.g. Otsu, Su) and compare it to the OCR results of the color images. Do we need to binarize the images before or is Tesseract able to deal with color images?
- Directly forward the full images to Tesseract and compare it to the results when only using the crops of our Yolo model. Does the segmentation of Tesseract work as well as our layout analysis?
- Evaluate the influence of the Page Segmentation Mode (PSM) of Tesseract.

Results The results should be reported by calculating the Character Error Rate (CER) and Word Error Rate (WER). Also show some qualitative examples of the comparisons.

[2] https://github.com/tesseract-ocr/tesseract

Task C - Named Entity Recognition

```
{
    "Location": "Helenental",
    "Description": "unerlaubte Rodung im Schutzgebiet",
    "Date": "April 1948",
    "Photographer": "Meisinger",
    "Film": "Neg.Nr. 3254/KIX/16, Film",
}
```

Fig. 4: NLP Output

Goal The goal of the third task is to analyze the text of the Tesseract output using NLP with two approaches: 1) `https://spacy.io/` and 2) a Large Language Model (LLM).

Task Make yourself familiar with spaCy. Try to solve the following tasks with spaCy (the results do not need to match the output of Fig. 4):

1. As text data, use the Tesseract output from the OCR assignment.
2. Apply a Named Entity Recognition, visualize the word vectors using t-SNE, per entity as well as the complete dataset.
3. Find similar words based on the word vectors and present the results. Can you spot any patterns?

Secondly, you will notice that the results of spaCy are poor, in particular if we consider special entities such as person/photographer. Therefore, we want to investigate the use of LLMs for named entity recognition as shown in the example above. For the LLM, the task can be solved in two ways, depending on your hardware resources:

- Running the LLM in a web demo (e.g., ChatGPT, Llama 2 or 3 `https://deepinfra.com/meta-llama/Meta-Llama-3-70B-Instruct`, etc.).
- Downloading a model and directly running inference via a Python script (e.g., `https://huggingface.co/meta-llama`). You will probably only be able to run the LLM on the CPU, requiring that your machine has enough RAM (e.g., 32 GB, depending on the model size). Find a proper prompt to directly obtain a correct JSON given the entities in the example.

Results Evaluate at least five samples and include them in the report. You can also play with different prompts/models and compare the differences. What are the benefits/disadvantages of spaCy and your selected LLM? The effectiveness of the method should be mainly determined by the location entity. Include qualitative examples and plots of your evaluation and the tasks above. Pay attention to a careful description.