Historical Map Understanding Using End-to-end Methods

Ladislav Lenc$^{1,2[0000-0002-1066-7269]}$, Jiří Martínek$^{1,2[0000-0003-2981-1723]}$, and Pavel Král$^{1,2[0000-0002-3096-675X]}$

[1] Dept. of Computer Science & Engineering, University of West Bohemia, Plzeň, Czech Republic
[2] NTIS - New Technologies for the Information Society, University of West Bohemia Plzeň, Czech Republic
{llenc,jimar,pkral}@kiv.zcu.cz

Abstract. This work deals with processing historical cadastral maps. It is a part of a complex system for seamless map creation. The main goal is to detect and recognize so-called nomenclatures, text information composed of several components that specify the position of a map sheet in the coordinate system. This information will then be used to create a large seamless map, allowing for better online presentation.
The main contribution of this work is the utilization of a modern end-to-end approach and its comparison with the currently used two-step approach combining text detection and OCR techniques. We chose a visual document understanding model, concretely Donut, for our experiments. The results prove that such models can be successfully trained for our task and outperform the traditional methods.

Keywords: Historical maps · Visual document understanding · Text recognition · Donut.

1 Introduction

Processing historical documents is an essential step in preserving our cultural heritage. Abundant volumes of various historical materials are stored in archives. It is crucial not only to digitize them but also to provide efficient ways to utilize them and extract the appropriate information. In this work, we present methods for information extraction from historical cadastral maps. We focus on the maps of the so-called stable cadastre, which was used in the former Austro-Hungarian empire in the first half of the 19th century. More concretely, we are working with maps covering the area of the Czech Republic owned by the Czech Office for Surveying, Mapping and Cadastre (CUZK)[3].

The maps of the Stable cadastre [2] are stored as individual map sheets covering a small part of the cadastre area. Our main goal is to identify the position of the map sheet in the coordinate system. The position is indicated by a so-called

[3] https://www.cuzk.cz/

nomenclature which is a set of several pieces of information describing the map sheet location. Therefore, we must recognize the text of the nomenclature and localize its components. The obtained nomenclature components are utilized for placing the map sheet in the correct position and allow us to create a seamless map which is our final goal.

This task can be viewed as a special case of Visual Document Understanding (VDU), respectively as one of its sub-tasks, Key Information Extraction (KIE). VDU has been traditionally performed in two steps. The first step is text detection and recognition and the second one is the information extraction itself. However, recently, the trend has gone towards so-called end-to-end methods. The input of the end-to-end methods is an image and the output is an editable format such as JSON or XML containing structured output with the desired information. Such methods perform the whole task in one step, which can eliminate the chaining of errors caused by the individual steps.

The main contribution of this work is an in-depth evaluation and comparison of the traditional two-step method with the modern end-to-end approach based on the Donut [10] model for our task. The more appropriate approach will be then integrated into the experimental system for seamless map creation. We first present a two-step method utilizing text detection and OCR. Next, we experiment with the Donut end-to-end model trained for the direct prediction of nomenclature components in the form of a structured JSON file. Both methods are evaluated on our newly created Nomenclature dataset, and we assess their advantages and drawbacks when utilized for seamless map creation.

2 Task Background

As mentioned in the previous section the maps of Stable cadastre are presented as individual map sheets. Each sheet contains a label (so-called nomenclature) defining the global and local positions in the Gusterberg system. The nomenclature is usually located in the top-right corner (as illustrated in Figure 1) and is composed of four components:

1. Name of the cadastral area;
2. Colonne – relative position to the Gusterberg meridian:
   (a) eastern ("O.C." – *Oestliche Collone*);
   (b) western ("W.C." – *Westliche Collone*);
3. Global coordinates (e.g. VII.29.);
4. Local coordinates (Section, e.g. a.f.).

Figure 2 then shows the Gusterberg coordinate system with both global and local positioning. The blue line indicates the meridian on which the village of Gusterberg lies and defines the origin of the coordinate system. The Roman and Arabic numbers then specify the position in the grid. Each cell in this upper-level grid is further divided into sections defined by a pair of small letters (see the right part of Figure 2).

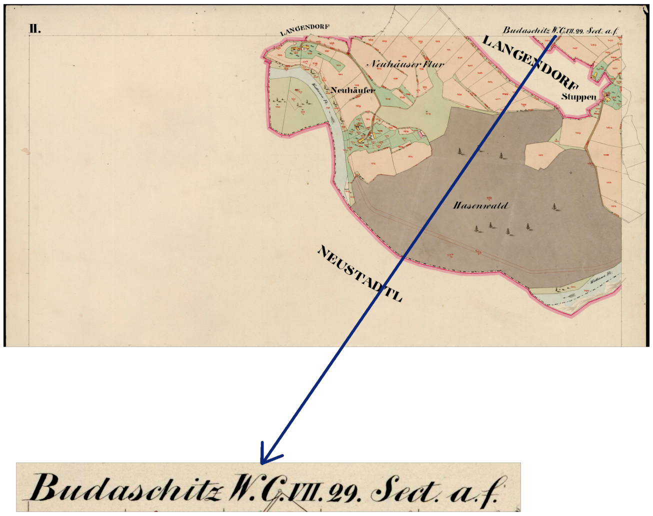

Fig. 1. Example map sheet with a zoomed nomenclature region

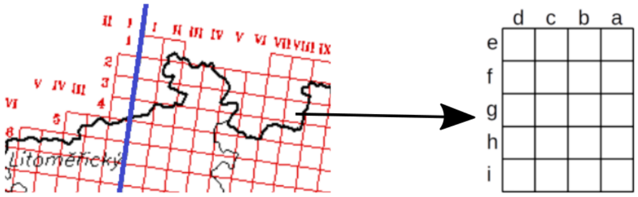

Fig. 2. Gusterberg coordinate system

The main goal is to recognize all nomenclature components to allow the correct positioning of the map sheets and their connection into one seamless map. There are some challenging cases where a single map sheet spans over two local areas (sections) and the section definition is more complicated (for example: “e.c. & f.c.”). A similar case can occur also for the global positions (for example: “XI.10.11.”)

3 Related Work

To the best of our knowledge, there is limited work on the analysis of cadastral maps using deep learning techniques. Ignjatic et al. [7] present an overview of the use of Deep Neural Networks (DNNs) for cadastral map digitization. Deep learning was also utilized for the analysis of unmanned aerial vehicle images that can provide an efficient way of cadastral mapping compared to classical techniques [24]. Detection of road types in historical Austrian-Hungarian maps can be found in [3]. Although these studies are very interesting, they differ from the focus of our task.

Therefore, we focus next on text detection and OCR/HTR algorithms in general. Most text detection algorithms are currently based on some variants of Convolutional Neural Networks (CNNs) [6]. The HTR task has been often solved by a combination of CNNs and recurrent neural networks [18, 21]. More recently, this task has been addressed using attention networks [17, 16] or using Transformers [9, 23, 12].

Recent document image understanding methods combine computer vision and natural language processing approaches. These approaches rely on OCR engines and feed their output into the neural language model [13, 25]. Many such methods use the popular BERT architecture as a language model.

DocFormer [1] integrates visual and textual features using multi-modal self-attention layers, resulting in state-of-the-art performance on various VDU benchmarks. Another efficient approach introduces a model based on contrastive learning. This model enhances visual representation in text-rich scenarios by aligning document object features with visual features generated by vision encoders in large visual-language models. This method has been shown to improve the performance of VDU tasks by focusing on fine-grained features [14].

Last but not least, we can mention the work of Zhu et al. [26] that includes rationale distillation methods, which aim to distill the reasoning process of large models into smaller, more efficient ones.

4 Nomenclature Detection and Recognition

We present two methods for nomenclature detection and recognition. The first one is a two-step approach combining neural networks for nomenclature detection and an OCR engine for text recognition. The second one is an end-to-end method utilizing the Donut [10] model.

4.1 Two-step Method

The first step is the nomenclature region detection and extraction. Follows the recognition of the nomenclature text. The final stage is the identification of the nomenclature components and their post-processing. We employ Mask R-CNN [5] and YOLO [22] networks for the detection.

Mask R-CNN Mask R-CNN has been developed on top of the R-CNN (Region-Based Convolutional Neural Network) network and its successor Faster R-CNN [20]. It was designed specifically for object detection and recognition. The Mask R-CNN combines semantic and instance segmentation and thus outputs also a segmentation mask for each Region of Interest (RoI). The original R-CNN model consists of region proposal, feature extraction and classification modules. The proposed regions are processed separately and classified for the presence of the desired objects.

The next step in the development was the Fast R-CNN [4] model, with several innovations that dealt with the R-CNN drawbacks and sped up the process. Despite the enhancements, computational efficiency was still decreased due to the number of RoIs. The situation emerged into proposing the Faster R-CNN [20] model and the Mask R-CNN that has been developed on top of the Faster R-CNN.

YOLO YOLO (You Only Look Once) [19] architecture family already has a significant number of members (YOLv11 is the last one at the time of writing). Most versions, starting from YOLOv3, were developed by the Ultralytics company.

The original YOLO architecture was proposed as a real-time object detector. The emphasis was placed on the inference time so that it can be used in time-critical applications. Every new version of the model comes with several improvements over the older YOLO models. For example, YOLOv8 [8] brought a new anchor-free detection system. Also, Convolutional blocks were changed and improved, and a mosaic augmentation was applied during the training phase. Generally, every new model slightly improves the performance while reducing the number of parameters and thus improving computational efficiency. The experiments were performed with the YOLOv10 [22] version.

Text Recognition The detection results are processed by the Tesseract OCR engine. This engine was selected based on the results of our preliminary experiments. We have trained a model on the training part of the nomenclature dataset. Since the nomenclature texts are in German, we utilized a German model (`deu_best`) as a starting point and used our data for fine-tuning. This decision was made mainly due to an insufficient number of annotated images.

4.2 Donut

Document Understanding Transformer (Donut) [10] is an OCR-free method designed for the VDU task in an end-to-end manner. It performs a direct mapping from raw input to the desired output.

The architecture is based on transformer networks. In the pre-training phase, Donut learns how to read the input image by making predictions of the following words. The predictions are made by conditioning the image and the text contexts together. Then, it learns how to understand the whole document according to the downstream task. The process starts by dividing the image into rectangular patches. The encoder extracts features from the given input image by converting it into a set of embedding values calculated from the number of image patches and the dimension of the encoder's latent vectors.

The encoder starts by splitting the image into patches, followed by applying the Swin transformer [15] blocks to the patches. After that, the patch layers are merged and passed to the textual decoder. The decoder uses the BART [11] architecture, which is responsible for generating a token sequence. Together, the transformers can convert the input images into editable files as illustrated in Figure 3.

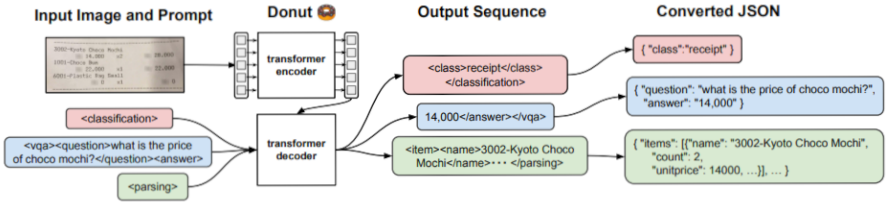
Input Image and Prompt
Donut 🍩
Output Sequence
Converted JSON
3002-Kyoto Choco Mochi
14,000 x2 28,000
1001-Choco Bun
22,000 x1 22,000
6001-Plastic Bag Small
0 x1 0
transformer encoder
transformer decoder
<classification>
<vqa><question>what is the price of choco mochi?</question><answer>
<parsing>
<class>receipt</class></classification>
14,000</answer></vqa>
<item><name>3002-Kyoto Choco Mochi</name>··· </parsing>
{ "class":"receipt" }
{ "question": "what is the price of choco mochi?", "answer": "14,000" }
{ "items": [{"name": "3002-Kyoto Choco Mochi", "count": 2, "unitprice": 14000, ...}], ... }

Fig. 3. Donut model architecture

Donut can be used for KIE as well as for other VDU tasks such as visual question answering or document classification.

5 Experiments

This section first presents the created Nomenclature dataset and metrics used for the evaluation. Then we present experiments performed with both the two-step method and the Donut model.

5.1 Nomenclature Dataset

We have created an annotated dataset from data provided by CUZK. The ground-truths contain information about the position of the nomenclature text bounding box and a complete transcription. We have also added JSON files containing all nomenclature components as shown in Listing 1.1.

Listing 1.1. Nomenclature annotation in JSON format

```
{"area": "Trzebeschitz", "colonne": "OC", "roman": "VII",
"arabic": "20", "first_sector": "b", "second_sector": "h"}
```

We picked 10 different areas that cover a significant part of the Gusterberg coordinate system to ensure enough variability. In each area, we have manually annotated 80 map sheets. Our dataset thus contains 800 annotated map sheets in total. The dataset is divided into train (650 samples), validation (50 samples) and test (100 samples) parts. It is freely available for research and educational purposes[4].

[4] https://corpora.kiv.zcu.cz/nomenclature/

5.2 Evaluation Metrics

For the evaluation of nomenclature detection experiments, we use a standard Average Precision metric (AP). We also report precision and recall. The text recognition results are reported in terms of Word Error Rate (WER) and Character Error Rate (CER). To evaluate the nomenclature components recognition we utilize the accuracy metric. Exact Match (EM) is the accuracy that all the nomenclature components are correct.

5.3 Two-step Method

In this section, we evaluate the individual components of the two-step method as well as the overall results achieved on the nomenclature components recognition. Table 1 presents the results of the nomenclature detection experiments. We show a comparison of the results of two neural network architectures, namely Mask R-CNN and YOLO.

Table 1. Nomenclature detection results; IoU threshold=0.5

Model	AP@50	Conf. Thresh.	P	R
Mask R-CNN	0.963	0.9	0.897	0.980
YOLO	0.992	0.55	0.970	0.990

The best results have been obtained by selecting the confidence threshold of 0.55 for YOLO, considering only predicted regions with a score of at least 0.55. the resulting values of precision and recall are 97.0% and 98.9% respectively.

On the other hand, confidence scores obtained by Mask R-CNN are consistently higher than the ones from YOLO. The vast majority of predicted regions obtained scores above 0.95. Thus, we have chosen the confidence threshold of 0.9 resulting in a very high recall=98%. Due to the relatively high amount of false positive regions (11 FPs) contrary to the YOLO model, the precision decreased below 90% (89.7%).

Next, we evaluate the text recognition performance on the detected nomenclatures obtained by the YOLO network. We use character error rate (CER) and word error rate (WER) metrics to see the performance of our trained model. We obtained 63.3% and 14.8% for WER and CER respectively.

The obtained values indicate that the text recognition is far from perfect. However, a significant part of the text is constituted by the cadastre area name which is not crucial for us in this task. Moreover, there are specified rules for the individual components of the nomenclature and we thus can apply some heuristics and post-process the OCR result. The final results of the two-step method including post-processing are summarized in Table 2

Table 2. Accuracies of the two-step method (in %) ; Col. = Colonne, R = Roman coordinates, A = Arab coordinates, S1 = Sector 1, S2 = Sector 2

	Col.	**R**	**A**	**S1**	**S2**	**EM**
Two-step method	91	82	88	92	91	74

The accuracies of all nomenclature components are around 90%, except the Roman numbers with an accuracy of 82%. The exact match is then 74%, which means that 74% of the nomenclatures were identified perfectly.

5.4 Donut End-to-end Method

This section presents the experiments with the Donut model. In the first experiment, we train Donut directly on the whole map sheets and let it predict the nomenclature components as a JSON file (see Listing 1.1). We experiment with several feature sizes (size of the patches used in the Donut encoder) and with two sizes of the input images. Due to the large size of the original images, we used resizing with 1/2 and 1/4 ratios. We experiment with the pre-trained Donut base model (`naver-clova-ix/donut-base`) and fine-tune it for our task. In all cases, we use the learning rate `2e-5` and train the model for up to 20 epochs. Early stopping is applied based on the value of the nomenclature exact match metric measured on the validation part of the dataset. Batch size varies from 1 to 4 depending on the feature size. All experiments are computed on GPU Nvidia A4000 with 16GB of memory.

Table 3. Accuracies of Donut model trained on whole map sheets (in %); Col. = Colonne, R = Roman coordinates, A = Arab coordinates, S1 = Sector 1, S2 = Sector 2, EM = Exact match, Area = Area Levenshtein distance

Image size	**Feature size**	**Col.**	**R**	**A**	**S1**	**S2**	**EM**	**Area**
1/4	**480x640**	60	17	9	27	21	0	9.21
	720x960	69	12	16	31	51	0	5.93
	960x1250	98	25	67	94	80	12	3.22
	1200x1600	99	79	93	95	90	65	1.40
	1320x1760	99	79	90	95	94	67	1.28
1/2	**480x640**	31	17	9	19	23	0	8.96
	720x960	70	17	17	54	66	3	5.19
	960x1250	94	8	66	89	82	5	2.74
	1200x1600	97	35	81	94	90	25	1.94
	1320x1760	97	80	88	96	98	69	1.95

Table 3 shows the results of Donut trained on whole map sheets to recognize the nomenclature components. The upper part is trained on sheets resized with a ratio of 1/4 while the lower part contains results resized with a ratio of 1/2. The results indicate that the smaller dimensions of input features are insufficient for this task. However, the bigger ones can achieve results comparable to the original two-step method. We can see that, similarly as in the two-step method, the lowest accuracy is achieved for Roman numbers.

The second evaluated scenario expects that the nomenclatures were already detected and cropped. We train Donut to recognize directly all nomenclature components as in the first scenario. In this case, the examined area is much smaller which should be beneficial for the recognizer. The settings of the training procedure are the same as in the case of training on the whole images.

Table 4. Accuracies of Donut model trained for nomenclature parts recognition on cropped nomenclature images (in %); Col. = Colonne, R = Roman coordinates, A = Arab coordinates, S1 = Sector 1, S2 = Sector 2, EM = Exact match, Area = Area Levenshtein distance

Feature size	**Col.**	**R**	**A**	**S1**	**S2**	**EM**	**Area**
480x640	94	94	95	95	95	84	1.82
600x800	97	94	94	98	95	83	1.54
720x960	99	93	97	98	96	86	1.65

Table 4 shows the results of this experiment. We can see a significant improvement mainly for Column and Roman and Arabic numbers. The sector (S1 and S2) recognition accuracy is more or less the same. Better recognition of the parts brought logically a significant improvement in the overall accuracy (EM). On the other hand, the recognition of the Cadastre area name remained comparable.

6 Conclusions and Future Work

In this paper, we presented, evaluated and compared two methods for the analysis of historical cadastral maps within the frame of a system for seamless map creation. We developed a two-step method for nomenclature detection and recognition and compared it with an end-to-end model Donut with the goal to select the more appropriate method for integration into our system.

The experiments showed that Donut trained for predicting the nomenclature components from the whole map sheet, omitting the detection step, can achieve an exact match of 69% which is comparable to the two-step method. In the second evaluated scenario, we were able to achieve significant improvement of the exact match reaching 86%. It is a very good result which can be utilized in the target application. However, we cannot consider it as a pure end-to-end approach as it expects the nomenclature detection by external detector. In comparison with the two-step method, we have to mention that Donut proved a very good performance in the cadastre area name recognition. It indicates that its implicit OCR capabilities are better than those of Tesseract.

In future work, we would like to explore ways to improve the results. One possibility is to pre-train Donut on historical documents. We would also like to experiment with other map analysis tasks because nomenclature is not the only text content in the cadastral maps.

Acknowledgments. This work was created with the support of the project "R&D of Technologies for Advanced Digitization in the Pilsen Metropolitan Area (DigiTech)" No.: CZ.02.01.01/00/23_021/0008436 co-financed by the European Union.

References

1. Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., Manmatha, R.: Docformer: End-to-end transformer for document understanding. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 993–1003 (2021)
2. Čada, V., Vichrová, M.: Horizontal control for stable cadastre and second military survey (franziszeische landesaufnahme) in bohemia, moravia and silesia. Acta geodaetica et geophysica Hungarica **44**(1), 105–114 (2009)
3. Can, Y.S., Gerrits, P.J., Kabadayi, M.E.: Automatic detection of road types from the third military mapping survey of austria-hungary historical map series with deep convolutional neural networks. IEEE Access **9**, 62847–62856 (2021)
4. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
6. He, T., Huang, W., Qiao, Y., Yao, J.: Accurate text localization in natural image with cascaded convolutional text network. arXiv preprint arXiv:1603.09423 (2016)
7. Ignjatić, J., Nikolić, B., Rikalović, A.: Deep learning for historical cadastral maps digitization: overview, challenges and potential (2018)
8. Jocher, G., Chaurasia, A., Qiu, J.: Yolo by ultralytics. `https://github.com/ultralytics/` (2023)
9. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Pay attention to what you read: Non-recurrent handwritten text-line recognition. arXiv preprint arXiv:2005.13044 (2020)
10. Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S.: Ocr-free document understanding transformer. In: European Conference on Computer Vision. pp. 498–517. Springer (2022)
11. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
12. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: Trocr: Transformer-based optical character recognition with pre-trained models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 13094–13102 (2023)
13. Li, P., Gu, J., Kuen, J., Morariu, V.I., Zhao, H., Jain, R., Manjunatha, V., Liu, H.: Selfdoc: Self-supervised document representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5652–5660 (2021)

14. Li, X., Wu, Y., Jiang, X., Guo, Z., Gong, M., Cao, H., Liu, Y., Jiang, D., Sun, X.: Enhancing visual document understanding with contrastive learning in large visual-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15546–15555 (2024)
15. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
16. Ly, N.T., Nguyen, H.T., Nakagawa, M.: 2d self-attention convolutional recurrent network for offline handwritten text recognition. In: International Conference on Document Analysis and Recognition. pp. 191–204. Springer (2021)
17. Poulos, J., Valle, R.: Character-based handwritten text transcription with attention networks. Neural Computing and Applications pp. 1–11 (2021)
18. Puigcerver, J.: Are multidimensional recurrent layers really necessary for handwritten text recognition? In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 67–72. IEEE (2017)
19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
21. Simistira, F., Ul-Hassan, A., Papavassiliou, V., Gatos, B., Katsouros, V., Liwicki, M.: Recognition of historical greek polytonic scripts using lstm networks. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 766–770. IEEE (2015)
22. Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: Yolov10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458 (2024)
23. Wick, C., Zöllner, J., Grüning, T.: Transformer for handwritten text recognition using bidirectional post-decoding. In: International Conference on Document Analysis and Recognition. pp. 112–126. Springer (2021)
24. Xia, X., Persello, C., Koeva, M.: Deep fully convolutional networks for cadastral boundary detection from uav images. Remote sensing **11**(14), 1725 (2019)
25. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., et al.: Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. arXiv preprint arXiv:2012.14740 (2020)
26. Zhu, W., Agarwal, A., Joshi, M., Jia, R., Thomason, J., Toutanova, K.: Efficient end-to-end visual document understanding with rationale distillation. arXiv preprint arXiv:2311.09612 (2023)