

Watermark Text Pattern Spotting in Document Images

Mateusz Krubiński^{1,*}, Stefan Matcovići^{2,*}, Diana Grigore², Daniel Voinea² and Alin-Ionut Popa^{2,**}

¹Charles University, Faculty of Mathematics and Physics

²Amazon Inc.

Abstract

Watermark text spotting in document images can offer access to an often unexplored source of information, providing crucial evidence about a record’s scope, audience and sometimes even authenticity. Stemming from the problem of text spotting, detecting and understanding watermarks in documents inherits the same hardships - in the wild, writing can come in various fonts, sizes and forms, making generic recognition a very difficult problem. To address the lack of resources in this field and propel further research, we propose a novel benchmark (**K-Watermark**) containing 65, 447 data samples generated using **Wrender**, a watermark text patterns rendering procedure. A validity study using humans raters yields an authenticity score of 0.51 against pre-generated watermarked documents. To prove the usefulness of the dataset and rendering technique, we developed an end-to-end solution (**Wextract**) for detecting the bounding box instances of watermark text, while predicting the depicted text. To deal with this specific task, we introduce a variance minimization loss and a hierarchical self-attention mechanism. To the best of our knowledge, we are the first to propose an evaluation benchmark and a complete solution for retrieving watermarks from documents surpassing baselines by 5 AP points in detection and 4 points in character accuracy.

Keywords

watermark text patterns, text spotting, text recognition, visual document understanding

1. Introduction

Information retrieval seen through the prism of visual document understanding (VDU) has recently become a mainstream task in the industry plus a hot topic in the computer vision and the natural language processing communities. It is fueled by the constant need to create automated document processing workflows and manifested through use-cases such as named-entity recognition [1, 2, 3, 4, 5, 6], document classification [7, 8], explainability in the context of validation [9, 10] or question answering [11]. In spite of the growing body of research addressing VDU-related problems, several challenges remain unsolved as we are still to attain *holistic understanding of the document*, to name a few - high variability in terms of layout, complexity of the visual information distribution or the multi-lingual / semantically different character of the depicted text. Additionally, since the task requires an efficient combination of visual and textual modalities that would complement each other in case one is incomplete or missing, it is difficult from the modeling perspective.

SDU@AAAI-24: Workshop on Scientific Document Understanding, co-located with AAAI 2024, Vancouver, Canada

*These authors contributed equally.

** Corresponding author.

† Work done while at Amazon.

✉ krubinski@ufal.mff.cuni.cz (M. Krubiński);
stefmatc@amazon.com (S. Matcovići); digrigor@amazon.com
(D. Grigore); dvoinea@amazon.com (D. Voinea);
popaaln@amazon.com (A. Popa)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)



Figure 1: Sample watermarked document image patches. The task of watermark text recovering is challenging (a) from a visual perspective due to resemblance with other document elements (*rightmost image*) and (b) from a textual / language perspective due to high fadedness causing text misinterpretation from overlapping document text (*left upper image*).

In addition to the aforementioned challenges, there are several unexplored areas of equal difficulty. One such topic is the problem of watermark text understanding from document images. Recognizing watermark content can enhance the understanding of the document information beyond the level provided by document text alone. For example, some documents might contain watermark text signaling the status of the file (*e.g.*, confidential, draft) or conveying additional information regarding the discussed content (*i.e.*, hazardous substances). One document analysis domain where this is of critical importance is within compliance-related tasks where in order to demonstrate the validity of a document, one must ensure that there is no watermark text on top, and if there is one, that it follows the required pattern, *e.g.*, denoting the type of document or provenience. The most difficult aspect resides in the fact that we are dealing with two pattern distributions (watermark text vs. document text) that are very similar in terms of visual and textual appear-

ance (see Figure 1) with a high occlusion and overlapping degree between the elements. Moreover, this can be easily mistaken with other forms of text elements which can be placed diagonally inside a document, such as handwritten signatures, stamps or even text-based logos. The main contributions of this work are:

- **Wrender**, a flexible algorithmic procedure for rendering text-based patterns inside a document controlling the orientation, transparency, content and font of the watermark, enabling the creation of diverse training and evaluation datasets,
- **K-Watermark**, a watermark spotting benchmark with a total of 65,447 data samples, that is, to the best of our knowledge, the first of this kind, on which we evaluate different text spotting solutions to establish strong baselines for watermark text detection and recognition,
- **Wextract**, an end-to-end method (see Figure 2) of detecting and recognising watermark text patterns in document images through the use of a novel loss formulation and a hierarchical self-attention encoder-decoder mechanism operating at both local and global level, achieving state-of-the-art results.

2. Related Work

Traditional OCR methods [12, 13] rarely focus on recovering text in occlusion or random orientation scenarios. Recently, a couple of lines of work [14, 15, 16, 17, 18, 19, 20, 21, 22, 23] bring more attention to the task of text-spotting. Their main advantage over standard OCR techniques is their robustness with respect to background appearance variety, unknown or undefined text orientations (*i.e.*, not the usual straight line orientation), unknown or mixed text fonts and even clutter with different scene elements resembling text.

One of the most challenging aspects associated with the task of text-spotting is the ability to determine the necessary connection between the identified individual character instances such that the recovered text has the required semantic meaning. For example, the authors of [19] propose a system which detects individual occurrences of characters as well as their pairwise connections (inspired by the work of [24]) to determine the shape of the word. In our setup, when the background involves sequences of characters, the major challenge is to separate the searched text (watermark) that is occluded by or intersected with the background text.

The object detection methodology [30, 31, 32, 33, 34, 35, 36] can address the issue of localizing visual elements from text forms. The general principle is to localize and classify object regions within the image space by leveraging the local context and the discriminative aspect

Benchmark	Domain	Samples	Supervision Level
K-Watermark	Text Spotting	65,447	Word
DUDE [25]	Question Answering	7,947	Paragraph
Tobacco [26]	Classification & Retrieval	3,482	Document
FUNSD [27]	Named-Entity Recognition	199	Word
PublayNet [26]	Information Retrieval	364,232	Paragraph

Table 1

Common document understanding benchmarks. Our proposed **K-Watermark** dataset falls in the category of large datasets and presents granular-level annotations (*i.e.* word-level).

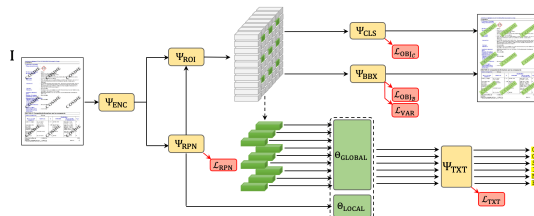


Figure 2: Detailed overview of the proposed Wextract method. Given an input image I , we obtain a set of watermark region proposals via the Ψ_{CLS} and Ψ_{BBx} heads. At the same time, we construct a *global* embedding representation by performing self-attention on top of the sequence of watermark region proposals generated via Ψ_{ROI} and a *local* embedding representation built using self-attention at class-agnostic proposal feature level via Ψ_{RPN} . Lastly, a watermark character-level prediction via Ψ_{TXT} is applied on top of the decoded joint global and local information.

of the searched object classes. Our approach is based on the work of [30] combined with [33] to help us spot watermark instances, by optimally combining modular features at different scales. The majority of document related benchmarks (see Table 1) focus on question answering, NER or general document level reasoning. Text spotting on documents has diverging needs, the most significant being a costly and inefficient world-level supervision process, proving the need for constructing a principled data setup.

3. K-Watermark Dataset

In a real-world scenario, the watermark alteration of a document is performed using specialized document editing tools (*e.g.*, Microsoft Word, Adobe Acrobat). This is a one-way prohibitive process which does not grant access to watermark text details such as size, position, word, font, angle, all of which represent necessary ground-truth information with respect to the task of watermark text spotting. Thus, one is forced to collect these details using human annotators, which turns out to be a complex and time-consuming process, as it requires a high degree of concentration to perfectly align the depicted watermark

text bounding boxes and compensate for the various degrees of transparency and occlusion against visual elements. There are clear advantages of obtaining them in an automated manner as it is more efficient and frugal. Given the lack of available benchmarks on watermark text spotting from document images, we define an image-based data generation procedure called *Wrender*, that augments clear document images with watermark text patterns in a fully programmatic setup allowing for full control over all the aspects of the process. Augmentation is applied in such a manner that the watermark text overlaps with the original text and other visual elements (e.g., tables, figures). The complexity of obtaining these examples stems from the multiple degrees of freedom involved: inserting the text, choosing the font type, the pattern density and visibility of the rotation angle all bring their own difficulty.

Algorithm 1 *Wrender*

Input: $\mathbf{I}_{\text{DOC}} \in \mathbb{R}^{h \times w \times 3}$
Output: $\mathbf{I}, \{b_i\}_{i=1}^{N_{\text{words}}}, \alpha, s$
 where $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}, b_i \in [0, 1]^4, s = \{s_i\}_{i=1}^{N_{\text{chars}}}$ with $s_i \in \{a \dots z\}$

- 1 $\alpha \leftarrow \text{Uniform} \left(-\frac{\pi}{2}, \frac{\pi}{2} \right)$
- 2 $n_{\text{words}} \leftarrow \text{Uniform} (1, 12)$
- 3 $t \leftarrow 0.1 + 0.5 \cdot \text{Random}_{\text{Beta}}(\alpha = 1, \beta = 1.5)$
- 4 $f \leftarrow \text{Uniform}(\text{GoogleFonts})$
- 5 $s \leftarrow \text{Uniform}(\text{NewsWords})$
- 6 $\mathbf{I} \leftarrow \mathbf{I}_{\text{DOC}}$
- 7 **for** $j = 1 \dots n_{\text{words}}$ **do**
- 8 **for** $k = 1 \dots n_{\text{words}}$ **do**
- 9 $pos = \left(\frac{j \cdot w}{n_{\text{words}}}, \frac{k \cdot h}{n_{\text{words}}} \right)$
- 10 $\mathbf{I}, b_i \leftarrow \text{InsertWatermark}(\mathbf{I}, s, t, f, pos)$

3.1. Wrender

In order to make the text insertion procedure as realistic as possible, we studied the typical watermark patterns (based on samples from the web and historical data), and concluded that they are rendered in a grid-like pattern, with the same text, color, font, angular placement and transparency. Their position does not take into consideration other document elements such as text or tables. Thus, our watermark insertion technique *Wrender*, was designed with these real-world observed constraints in mind. Algorithm 1 receives as input a document image \mathbf{I}_{DOC} and returns the watermarked image \mathbf{I} together with a list $\{(b_i, \alpha_i, s_i)\}_{i=1}^{N_{\text{words}}}$ containing the normalized bounding box coordinates $b_i \in [0, 1]^4$ of the inserted watermark text, their angle $\alpha_i \in (-\frac{\pi}{2}, \frac{\pi}{2})$ with respect to the horizontal axis as well as the watermark

text $s = \{s_i\}_{i=1}^{N_{\text{chars}}}$ with $s_i \in \{a \dots z\}$.



Figure 3: Wordcloud visualization encoding a display of the frequency of the words used throughout our training dataset. These are some of the most frequent english words commonly used in newspapers and mass media content.

For each image, the number of watermark text bounding boxes is chosen random from the list of perfect square numbers between 1 and 144. The reason is to have a grid positioning of the inserted words according to the square root of the selected number (e.g., for a value of 16 it will create a grid 4×4 of watermark words). Also, when we position the words on the determined grid, we added a small offset to the entire grid with respect to the page size so that we do not have the exact word positioning for 2 pages with the same grid size. Since based on our observations all the watermarks have the same angular orientation, α and the same text s , we imposed this constraint when rendering the watermarks on the grid. Moreover, the watermark text is usually very faded. Thus, we constrained the visibility of the inserted text to follow a Beta distribution which generates a higher number of values closer to the lower bound of 0.1 and a fewer number of values closer to the upper bound of 0.6. Basically, a value closer to 0.1 implies a higher fadedness and as it gets closer to 0.6, it becomes more visible. For the font type, we randomly sampled from the Google Fonts database [37].

In order to apply our augmentation technique, we use clean document images, not containing watermarks. For illustration purposes, we make use of the publicly available multi-page document datasets Kleister NDA and Kleister Charity [38], using the documents as background images to paste the watermark text patterns. As a consequence, we named the resulting dataset **K-Watermark**. The document data is split at the page level, resulting in the following split: 57,947 images for train, 2,500 images for validation and 5,000 images for test. For the test and the validation data, we have drawn real words from [39] and we have used different fonts than those used for training split, as well as different text densities of the watermark text. Additionally, when generating the test / validation set, we considered the situation of no watermark words, to include document scenarios with

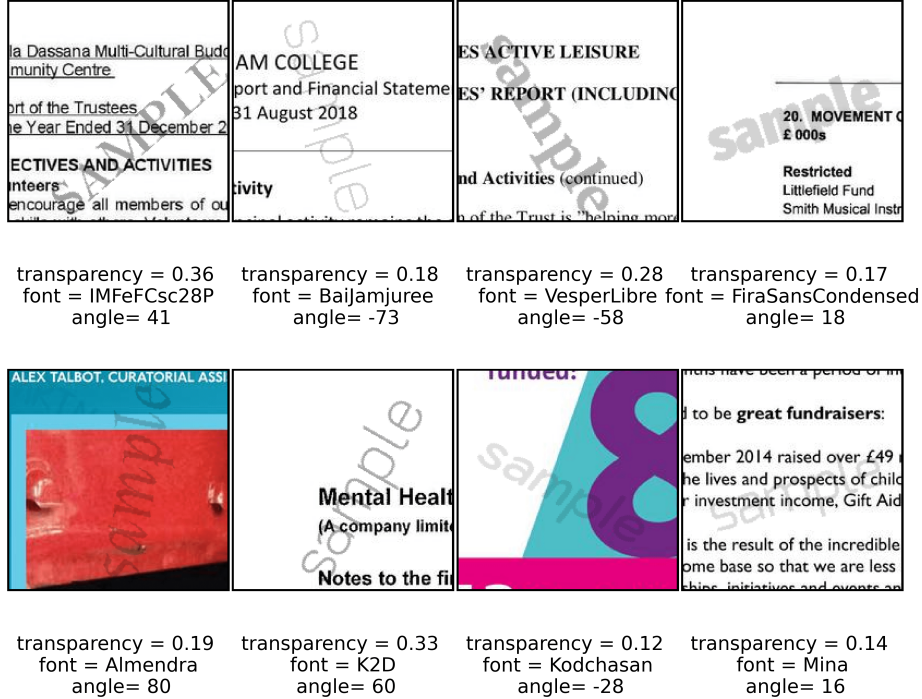


Figure 4: Generated individual watermark text patterns. Using our \mathcal{W} render insertion procedure we have full control over the watermark pattern insertion process similar to off-the-shelf professional document editing tools. Below each image patch, we specify the transparency (*i.e.* on a scale from 0 to 1), font (*i.e.* randomly sampled from [37]) and angle (*i.e.* from -90° to 90° degrees) used.

no inserted watermark to better estimate the false positive rate. We will release the **K-Watermark** dataset together with the code that renders patterns thus encouraging the research on document image watermarks in more diverse scenarios without the explicit need for labeling large amounts of data. Furthermore, the watermark insertion procedure can be used dynamically during training, in order to reduce the possible overfitting, the augmented data might induce.

To prove the validity of our hypotheses about real-world watermarks we conducted a human perception study. We sampled 100 images watermarked using \mathcal{W} render and 100 random watermark documents collected using web search (*i.e.*, pre-generated in the wild). Next, we asked 5 human annotators to rank them as \mathcal{W} or pre-generated. The outcome was 0.51 and 0.49 F1-score for the \mathcal{W} render and pre-generated class, respectively, proving that our procedure is able to mimic the watermark insertion procedure from specialized tools successfully and enabling us to obtain text annotations (bonding box localization and depicted text) with far less resources.

During the training procedure, **K-Watermark** samples are dynamically generated using \mathcal{W} render by

trieving random words from [39] and placing them as watermark patterns on top of the document page. A word-cloud visualization is illustrated in Figure 3 with words used as watermark text patterns during training. These are frequent words from the English vocabulary of various lengths and with different semantic meanings. In Table 2 we illustrate word level statistics for the validation and test sets emphasizing the high density of watermark text patterns at both dataset and page level as well as the high overlap against document text.

In Figure 4 we illustrate samples generated dynamically by using the \mathcal{W} render procedure. It is noticeable how the watermark text augmentation is impacted by the multiple degrees of freedom across the angle, font and fadedness elements. By doing this, we increase the generability potential of our proposed approach.

4. Methodology

Given an image $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$, the objective is to predict a watermark text string $\tilde{s} = \{\tilde{s}_i\}_{i=1}^{N_{chars}}$ with $\tilde{s}_i \in \{a \dots z\}$ and a list of watermark image regions defined by the tuples $\{(\tilde{b}_i, \tilde{\alpha}_i)\}_{i=1}^{N_p}$, where $\tilde{b}_i \in [0, 1]^4$

Split Type	# of Watermarks			Document Text Overlap	
	Total	Mean	Median	Mean	Median
Validation	182,325	72.93 ± 62.564	49.0	53.56 ± 23.76	56.2
Test	360,973	72.19 ± 62.26	49.0	54.4 ± 23.39	57.3

Table 2

Common statistics for validation and test subsets. We emphasize the high density of watermark text pattern content at dataset and page level. Moreover, in columns 5 and 6 we highlight high overlap between document and watermark text which increases the complexity of the proposed task.

Model	mAP	AP@50	AP@75	mAR	AR@50	AR@75
TextSnake [40]	6	20	2	15	36	10
CRAFT [19]	7	26	1	34	46	8
DBNet++ [41]	10	29	4	16	38	11
UNITS [42]	8	15	7	22	38	23
TCM [43]	14	42	5	20	49	12
<i>Fine-tuned</i> TextSnake [40]	32	82	14	44	91	36
<i>Fine-tuned</i> DBNet++ [41]	48	92	44	60	97	67
<i>Fine-tuned</i> UNITS [42]	50	95	48	62	97	68
<i>Fine-tuned</i> TCM [43]	52	95	49	58	96	63
$\mathcal{W}\text{extract}$ w/o \mathcal{L}_{VAR}	56	96	58	61	96	69
$\mathcal{W}\text{extract}$	58	96	65	63	97	72

Table 3

Watermark Text Detection Results on K-Watermark Test Set. We compare against state-of-the-art baselines for text spotting. First baseline (lines 2 – 6) is by running the off-the-shelf version of these approaches on the raw documents (pre-watermark) and the watermarked documents (post-watermark) to retain the watermark text only via prediction set differentiation. The second baseline (lines 7 – 10) is to fine-tune (when possible) on K-Watermark and evaluate directly.

represents the normalized bounding box coordinates with respect to image height and width, h and w , respectively, $\tilde{\alpha}_i \in (-\pi/2, \pi/2)$ represents the angular orientation of the watermark text. In practice, the angles and the watermark text from all the ground truth bounding boxes are the same. As backbone for our proposed framework, $\mathcal{W}\text{extract}$, we use Faster-RCNN [30] combined with feature pyramid networks [33]

$$\Psi_{\text{OBJ}_C}(\cdot) = \Psi_{\text{CLS}} \circ \Psi_{\text{ROI}} \circ \Psi_{\text{RPN}} \circ \Psi_{\text{ENC}}(\cdot) \quad (1)$$

$$\Psi_{\text{OBJ}_B}(\cdot) = \Psi_{\text{BBX}} \circ \Psi_{\text{ROI}} \circ \Psi_{\text{RPN}} \circ \Psi_{\text{ENC}}(\cdot) \quad (2)$$

where Ψ_{ENC} is represented by a common image encoder which in our case is ResNet50 [44], Ψ_{RPN} is a class-agnostic region proposal network (RPN), Ψ_{ROI} is a region of interest (ROI) feature encoder which links the proposals from Ψ_{RPN} . Ψ_{CLS} and Ψ_{BBX} represent classification and regression heads, respectively, designed to align the region proposals received from the Ψ_{RPN} with the watermark class type and their exact corresponding bounding boxes within the image. In our case, the Ψ_{CLS} is a binary classifier stating whether the proposed bounding boxes depict watermark text or not. Moreover, the regression head Ψ_{BBX} not only predicts the bounding boxes of the watermark regions, but also the orientation of the detected watermark text patterns.

Backbone	mAP	mAR
ResNet50 [44]	58	63
SWIN-Base [45]	51	53
ViTDET-Base [35]	53	55

Table 4

Detection Results with Various Image Encoding Backbones. We experimented with different image encoding backbones to validate the best encoding architecture to proceed further with the end-to-end watermark text spotting task.

4.1. Variance Minimization Loss

Traditionally, the Faster-RCNN framework requires losses for the Ψ_{RPN} module and the final $\Psi_{\text{CLS}} / \Psi_{\text{BBX}}$ heads. We propose an additional loss term \mathcal{L}_{VAR} , which minimizes angle, width and height variations across predictions from each single-page document image. This loss is predicated on the observation that across a document, all the watermark text bounding boxes are expected to have the same width, height, and angular orientation. During training, if the cardinality of the watermark sequence is 1, the variance loss term evaluates to 0 (as the variance of a set with a single element is 0), thus not affecting the training process.

$$\begin{aligned} \mathcal{L}_{\text{VAR}} = & \sum_{i=1}^{N_p} [(\tilde{b}_{i2} - \tilde{b}_{i0}) - \frac{1}{N_p} \sum_{j=1}^{N_p} (\tilde{b}_{j2} - \tilde{b}_{j0})]^2 \\ & + \sum_{i=1}^{N_p} [(\tilde{b}_{i3} - \tilde{b}_{i1}) - \frac{1}{N_p} \sum_{j=1}^{N_p} (\tilde{b}_{j3} - \tilde{b}_{j1})]^2 \\ & + \sum_{i=1}^{N_p} (\tilde{\alpha}_i - \frac{1}{N_p} \sum_{j=1}^{N_p} \tilde{\alpha}_j)^2 \end{aligned}$$

where the 3 terms of the sum correspond respectively to the height, the width, and the angle of the predicted list of watermark regions. Additionally, \mathcal{L}_{VAR} is divided by N_p , as a normalization factor with respect to the total number of watermark proposals. As a consequence, the total loss formulation becomes

$$\mathcal{L}_{\text{VAR}} = \mathcal{L}_{\text{OBJ}_C} + \mathcal{L}_{\text{OBJ}_B} + \mathcal{L}_{\text{RPN}} + \mathcal{L}_{\text{VAR}} + \mathcal{L}_{\text{TXT}} \quad (3)$$

where \mathcal{L}_{VAR} represents the variance minimization loss term. The \mathcal{L}_{TXT} term represents the text recognition loss which we will define in the following section.

4.2. Local global self-attention

For the text recognition part, we propose the following hierarchical self-attention mechanism. *Firstly*, we compute local feature representations based on descriptors from the Ψ_{RPN} head.

$$\Theta_{\text{LOCAL}}^i = \text{softmax} \left(\frac{\phi^Q(\Psi_{\text{RPN}}^i) \phi^K(\Psi_{\text{RPN}}^i)^\top}{\sqrt{d_{\text{LOCAL}}}} \right) \phi^V(\Psi_{\text{RPN}}^i)$$

where Ψ_{RPN}^i represents the class-agnostic descriptor corresponding to the detection $i \in \{1 \dots N_p\}$ and $\phi(\cdot) : \mathbb{R}^{d_h \times d_w \times d_{\text{RPN}}} \rightarrow \mathbb{R}^{d_h \times d_{\text{LOCAL}}}$ is a projection function based on a 1×1 convolution, followed by a reshape operation that flattens the width dimension. The intuition is to iterate over each prediction and perform self-attention across a sequence built along the height of the descriptor tensor in order to retrieve meaningful local information with respect to the text recognition task. This is intended to mimic the heuristic of looking across the characters of a word to determine the word itself. In our experimental setup, we chose the following values for the previously listed dimensionalities: $d_h = 7$, $d_w = 7$, $d_{\text{RPN}} = 256$ and $d_{\text{LOCAL}} = 224$. The upper scripts Q, K, V used for ϕ stand for the query, key and value projections and are associated with the attention mechanism. Finally, we obtain a sequence of local embeddings, corresponding to all the retrieved watermark text detections, $\Theta_{\text{LOCAL}} = [\Theta_{\text{LOCAL}}^i]_{i=1}^{N_p}$.

Secondly, we compute global feature representations based on descriptors from the Ψ_{ROI} head.

$$\Theta_{\text{GLOBAL}} = \text{softmax} \left(\frac{\Psi_{\text{ROI}}^Q \Psi_{\text{ROI}}^{K \top}}{\sqrt{d_{\text{GLOBAL}}}} \right) \Psi_{\text{ROI}}^V$$

where $\Theta_{\text{GLOBAL}} \in \mathbb{R}^{N_p \times d_{\text{GLOBAL}}}$. The upper scripts Q, K, V denote the query, key, value projections used within the attention mechanism. The intuition is to perform an implicit statistic by analysing a sequence with all the watermark detections and gather complementary information with respect to the depicted watermark text through the multi-head attention. Thus, we overcome the issues created by occlusion and fadedness of the text by looking simultaneously inside the image. Next, we combine both global and local feature representations into a single tensor $\Theta = [\Theta_{\text{LOCAL}}; \Theta_{\text{GLOBAL}}]$ and project them to $\mathbb{R}^{N_p \times d}$, where d is the embedding size of the sequence elements. By doing this, we enforce the text decoding mechanism to incorporate complementary sources of information originating from bounding box level embeddings (local) and page level embeddings (global).

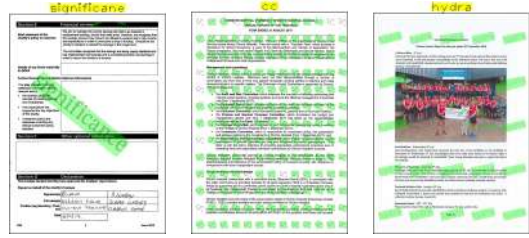


Figure 5: Watermark Text Detection Failure Cases of Wextract on K-Watermark. Detections are highlighted with green bounding boxes and recognized text is written as yellow highlighted text at the top of each image. Failure situations usually occur due to overlap between non-uniform or dark-coloured visual elements, which impacts text recognition (e.g., leftmost image, significance vs. significance). These scenarios are complex, even for humans.

On top of the global and local feature representation Θ we place an auto-regressive character-level transformer decoder [46], $\tilde{s}_i = \Psi_{\text{TXT}}(\Theta, \{\tilde{s}_j\}_{1 \leq j < i})$, to generate the depicted watermark text sequence \tilde{s} .

In practice, the i^{th} character s_i is generated by the decoder Ψ_{TXT} by looking at the visual features Θ and the previously generated characters (i.e., $\forall s_j$ with $j < i$).

During training, we use the reference watermark text s and train with the cross-entropy loss $\mathcal{L}_{\text{TXT}}(\tilde{s}, s)$. During inference, we generate the watermark text, character by character, until the [EOW] symbol is generated or the maximal sequence length (in our experiments is set to

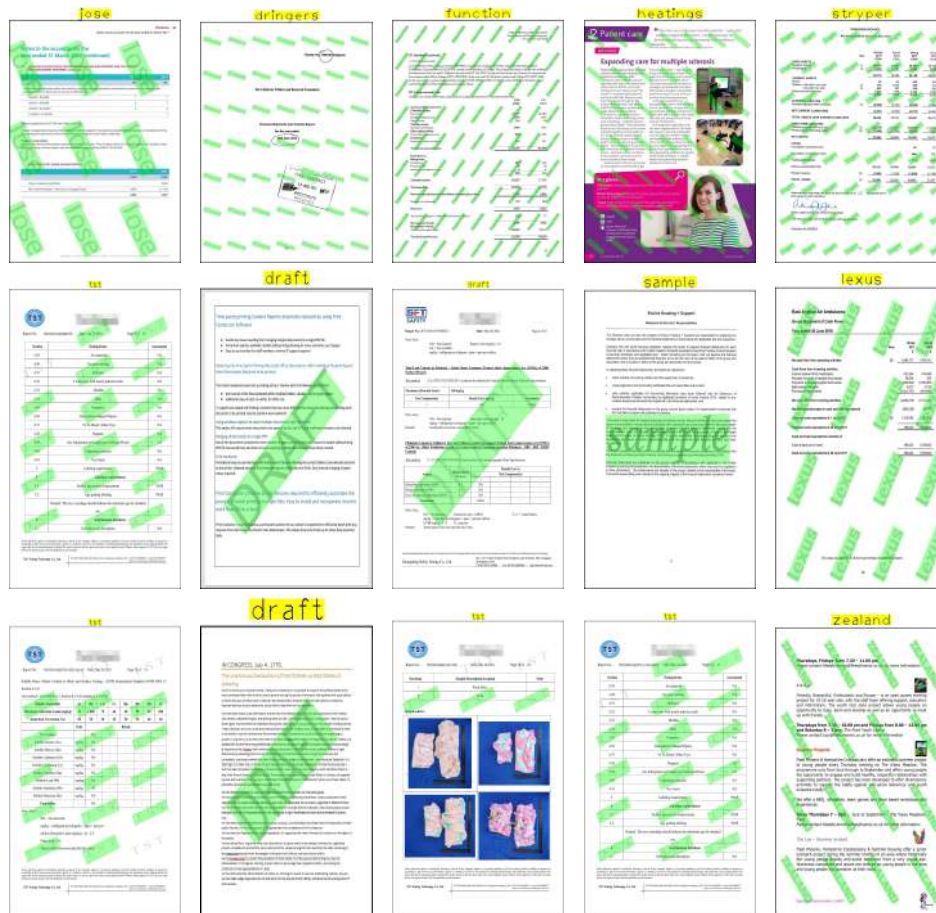


Figure 6: Visual Results of $Wextract$. Row (1) contains K-Watermark test samples and rows (2) and (3) contain web retrieved documents. Detections are highlighted with green bounding boxes and recognized text is written as yellow highlighted text at the top of the document page. Our approach exhibits robustness with respect to the density of the watermark text, the overlap degree with background text as well as images and the fadedness of the text. Additionally, it is robust with respect to vertical text orientation as well as other visual elements with diagonal orientation and generates consistent bounding box predictions in terms of angular orientation, width and height.

15). The idea behind our approach is to combine the information from the visual features (global and local) using the hierarchical encoder, with an auto-regressive decoding step that enables semantic reasoning over the previously generated characters.

5. Experiments

We experimented on our own augmented dataset, **K-Watermark**, created using the $Wrender$ algorithm. The dataset images encode a high amount of variability in terms of visual elements (containing simple text document pages and visually rich document pages depicting

images, logos or table structures), text density and semantics to represent a principled evaluation framework. To the best of our knowledge, we are the first to propose an approach which retrieves exclusively watermark text content. We adapted the text spotting baselines for a fair comparison. We compared against text detection approaches adapted for watermark text detection, combined with text recognition approaches.

5.1. Text Detection

The detection component was trained using an SGD optimiser with a constant learning rate of $5 \times 1e^{-3}$, an epoch of warmup and a batch size of 16. We initialise

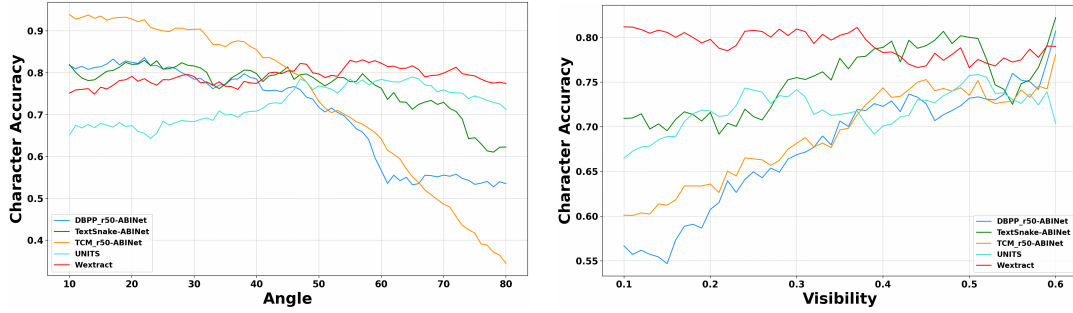


Figure 7: Watermark text spotting error plot against visibility and angular orientation (best seen in color). We plot the character accuracy metric for the **K-Watermark** dataset against different angular orientation of the watermark text with respect to the horizontal axis (*left*) and different text visibility values (*right*). Our method is robust with respect to the visibility and the angular orientation of the watermark text patterns and it is still being able to achieve a high performance even with near vertical angle or extreme fadedness.

Recognition		RobustScanner [22]	MASTER [47]	SATRN [48]	PARSeq [49]	ABINet [21]	UNITS [42]	Wextract
Detection	TextSnake [40]	0.60 ± 0.10	0.60 ± 0.09	0.61 ± 0.11	0.55 ± 0.12	0.75 ± 0.08		
	DBNet++ [41]	0.56 ± 0.13	0.56 ± 0.11	0.56 ± 0.12	0.55 ± 0.14	0.69 ± 0.14	0.70 ± 0.09	0.79 ± 0.02
	TCM [43]	0.51 ± 0.19	0.53 ± 0.17	0.53 ± 0.21	0.58 ± 0.15	0.67 ± 0.21		

Table 5

Watermark Text Spotting Results on K-Watermark Test Set. We report the mean character accuracy metric \pm standard deviation at document level by comparing our proposed **Wextract** (*rightmost column*) against different combinations of state-of-the-art text-spotting methods (*rows*) and text recognition frameworks (*columns*). **Wextract** and UNITS [42] are placed on a stand-alone column as they have end-to-end flows w.r.t. the retrieval of watermark text. Our approach outperforms the overall comparing baselines by a large margin. The second best performing one with close performance involves ABINet [21] recognition. Moreover, our method demonstrates consistent performance at document level as it has the lowest standard deviation *i.e.*, 0.02.

the Faster-RCNN with feature pyramid networks [33] using weights obtained through COCO pretraining [50]. We make use of **Wrenderer** dynamically: at each epoch the same document page has different patterns rendered. To maintain a reasonable training time, we down-scale the images so that the longest edge has between 800 and 1024 pixels. Detailed experimental results on the **K-Watermark** test set are reported in Table 3. We evaluate the watermark text detection component by computing standard rotated object detection metrics (*i.e.*, average precision and average recall).

We report the mean average precision (mAP) and mean average recall (mAR) with an intersection over union (IoU) overlap threshold between 0.5 and 0.95. Additionally, we provide evaluation at fixed IoU thresholds of 0.5 and 0.75. We compare with the state-of-the-art approaches for text detection [40, 19, 41, 43], but also with the detection results of an end-to-end text spotting method [42]. For a fair comparison, we have two baseline comparison setups. The first baseline setup (Table 3 lines 2-6) is to use off-the-shelf weights trained on public scene text detection datasets and recover all detection instances of text from the input image (including *e.g.*, document text, logo text, watermark text). Firstly, we

apply those methods both on the normal document images (pre-watermark insertion) and on the watermarked document images (post-watermark insertion), retaining the set of predictions which only occur on the latter.

Even with this advantageous evaluation setup, our pipeline outperforms these baselines by a large margin (≈ 50 points in mAP). In terms of recall, the gap is smaller due to the fact that the considered detection baselines pick up all types of text patterns from the document page, including watermark text patterns.

The second baseline heuristic (Table 3 lines 7-10) is to fine-tune the baselines on the K-Watermark training set, thus specializing the framework in retrieving watermark text bounding boxes solely. The performance is far superior compared to their off-the-shelf version, however inferior to our **Wextract** approach. We were unable to apply this comparison heuristic on [19] as the authors do not provide the training code. Prior to exploring the text recognition branch, we analysed the impact of using various backbones as alternatives to the ResNet50, however, as it is noticeable in Table 4, other transformer-based [45, 35] alternatives performed poorer and are more memory-intensive due to the increased number of parameters.

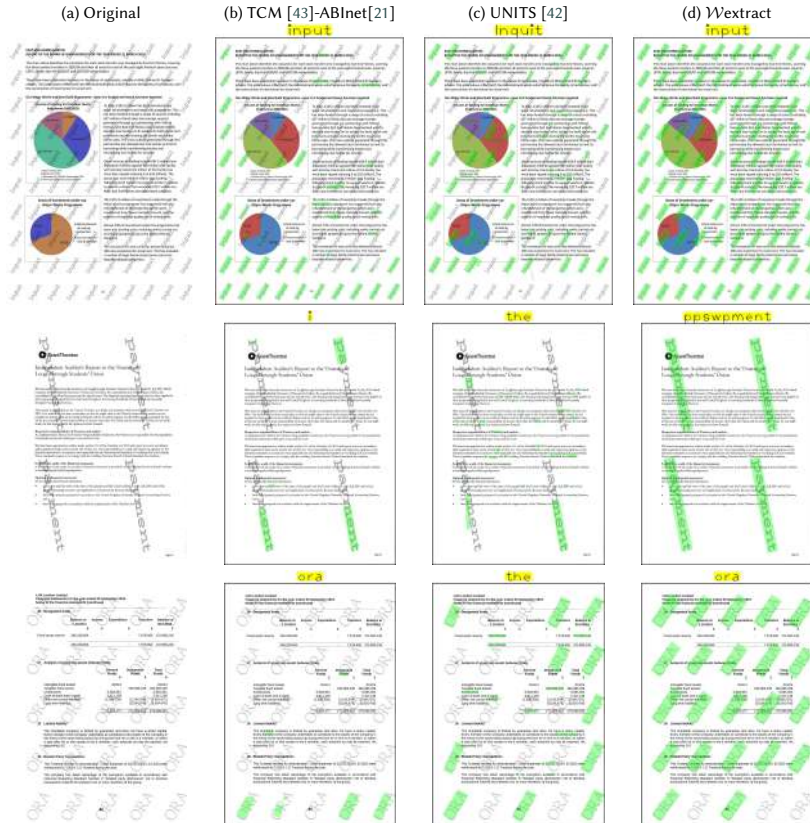


Figure 8: Qualitative assessment of $\mathcal{W}_{\text{extract}}$ against comparing baselines. Notice that our approach provides the highest coverage of watermark text patterns across the document page and generates the closest text prediction w.r.t. the depicted text.

Moreover, we perform an ablation study by training $\mathcal{W}_{\text{extract}}$ with and without the \mathcal{L}_{VAR} (lines 11 and 12 from Table 3). The \mathcal{L}_{VAR} usage results in an improvement of 2 mAP and mAR points, as it leverages the consistency of the watermark patterns in terms of dimension and orientation across the page. In Figure 6, row 1, we illustrate some sample visual results of $\mathcal{W}_{\text{extract}}$ on **K-Watermark** dataset. The method generates consistent watermark predictions across the page, which are in part influenced by the \mathcal{L}_{VAR} loss component. Also, it demonstrates robustness with respect to diagonally placed document text, high occlusion degree against background text or other visual elements such as logos, pictures or document page symbols. In Figure 6, rows 2 and 3, we visually demonstrate the robustness of our approach by testing on pre-generated watermark documents. This validates that our training procedure does not bias towards the $\mathcal{W}_{\text{render}}$ procedure and it is able to detect and recognize watermark text patterns from other sources. In addition, in Figure 5 we showcase examples when

our framework fails to capture all the watermark text instances or when various visual artefacts are recovered as faulty predictions.

5.2. Text Recognition

The text recognition head was trained jointly with a pre-trained watermark detection backbone using AdamW optimiser [51] with an inverse square root schedule that had a starting learning rate of 10^{-5} , $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$, $\gamma = 0.8$ and batch size of 8. Before the joint training, we performed a pre-training step for the Ψ_{TXT} decoder by freezing the detection backbone and using boxes corresponding to ground truth watermarks. It allowed the decoder to learn the semantic distribution of the watermark texts and helped with the stability of training. In order to assess the text recognition performance, we use only the fine-tuned versions of detection baselines, since these are the top scoring ones from Table 3 and apply state-of-the-art text recognition approaches

[22, 47, 48, 21]. Since $\mathcal{W}\text{extract}$ produces a single character sequence prediction per document, and the baselines we work with generate one for each watermark instance, we apply a majority voting aggregation procedure to produce a single sequence. Specifically, we retrieve the character sequence which occurs most frequently among the list of predicted character sequences. The numerical results averaged at document level are illustrated in Table 5. To measure the text recognition performance, we calculate the character accuracy, *i.e.*,

$$1 - \frac{\#\text{CHAR}_{\text{SUBST}} + \#\text{CHAR}_{\text{DEL}} + \#\text{CHAR}_{\text{INS}}}{\#\text{CHAR}_{\text{TOTAL}}}$$

where $\#\text{CHAR}_{\text{SUBST}}$, $\#\text{CHAR}_{\text{DEL}}$, $\#\text{CHAR}_{\text{INS}}$ and $\#\text{CHAR}_{\text{TOTAL}}$ represent the number of character substitutions, number of character deletions, number of character insertions and total number of characters, respectively.

The only baseline with a comparable performance is [21] combined with either [40] or [41]. Our assumption is that the cause of this is the language model component, which adjusts the prediction to resemble a semantically meaningful word entry. One interesting aspect is the fact that although [41] achieves superior detection results compared to [40], in terms of recognition it is inferior. Based on our empirical observations, the method of [41] captures more instances of watermark text patterns. However when combined with the majority voting heuristic, it induces more noise in the recognition pipeline, thus creating a slight performance gap. Although the model of [40] has a poorer detection performance, it captures the most relevant patterns w.r.t. the recognition task, thus less noise and better recognition performance. The method proposed in [42] is the only to provide an end-to-end flow which retrieves the watermark text pattern instances across the document page and their semantic meaning. We were able to fully fine-tune it across our proposed dataset. However, we still had to apply the majority voting procedure, as the method outputs a text prediction for each watermark box.

To better understand the limitations and the advantages of our method, we conduct a detailed analysis of the recognition performance. In Figure 7 we show two performance plots highlighting how the character accuracy is impacted by the fadedness (*right*) and the angular orientation (*left*) of the watermark text. We combined the text recognition pipelines with the best text detection approach for easier visualisation purposes. Our $\mathcal{W}\text{extract}$ is able to achieve the highest performance in situations of very high fadedness (visibility degree of 0.1) or with almost vertical text orientation (angle of ≈ 90 degrees with respect to the horizontal axis). Apart from this, we

manifest constant high performance across all angular orientations and visibility degrees. This observation is in line with the smallest standard deviation showcased by our approach in Table 5. In Figure 8 we showcase visual comparisons between our proposed approach and UNITS [42] and TCM [43]-ABInet[21] baselines. Our approach demonstrates maximum coverage of the watermark text pattern and robustness w.r.t. the orientation, fadedness, font type and high overlap against document text.

Moreover, in Table 6 we perform an ablation study to understand the impact of the local and global feature representations. We re-trained the recognition pipeline with all combinations of global and local feature representations. The global features are primarily specialized for the object detection tasks (classification and bounding box regression), thus using them solely performs poorer with respect to the text recognition task.

$\mathcal{W}\text{extract}$ Feature Type	$[\Theta_{\text{LOCAL}}; \Theta_{\text{GLOBAL}}]$	$[\Theta_{\text{LOCAL}}]$	$[\Theta_{\text{GLOBAL}}]$
Character Accuracy	0.791	0.721	0.479

Table 6

Ablation on importance of *global* vs *local* features. This study proves the importance of using the combined *global* and *local* information when determining the watermark text - they complement each other with respect to the task of text recognition. The usage of *global* features alone underperforms severely, as they are not able to overcome the local noise induced by overlapping with document text and high fadedness factors.

6. Conclusions

We present a principled benchmark for watermark text spotting from documents together with an end-to-end solution for detecting watermark text patterns and recognizing the depicted text. Our key novelty is a hybrid joint watermark text pattern detection and recognition system leveraging a global and local hierarchical attention mechanism robust to occlusion, low visibility and various degrees of document text densities which is constrained by a variation minimisation loss term. This is a niche and important problem with high potential impact, which can be considered as an extension for any OCR. No previous existing methodological approaches exist, thus we compared our proposed watermark text spotting pipeline, $\mathcal{W}\text{extract}$, against strong text-spotting baselines specifically adapted for our problem, outperforming them by a large margin.

References

- [1] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, et al., Layoutlmv2: Multi-modal pre-training for visually-rich document understanding, *arXiv preprint arXiv:2012.14740* (2020).
- [2] P. Li, J. Gu, J. Kuen, V. I. Morariu, H. Zhao, R. Jain, V. Manjunatha, H. Liu, Selfdoc: Self-supervised document representation learning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021*, pp. 5652–5660.
- [3] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie, R. Manmatha, Docformer: End-to-end transformer for document understanding, *arXiv preprint arXiv:2106.11539* (2021).
- [4] C.-Y. Lee, C.-L. Li, T. Dozat, V. Perot, G. Su, N. Hua, J. Ainslie, R. Wang, Y. Fujii, T. Pfister, Formnet: Structural encoding beyond sequential modeling in form document information extraction, *arXiv preprint arXiv:2203.08411* (2022).
- [5] Y. Huang, T. Lv, L. Cui, Y. Lu, F. Wei, Layoutlmv3: Pre-training for document ai with unified text and image masking, *arXiv preprint arXiv:2204.08387* (2022).
- [6] I.-C. Sandu, D. Voinea, A.-I. Popa, Consent: Context sensitive transformer for bold words classification, *arXiv preprint arXiv:2205.07683* (2022).
- [7] S. Garg, T. Vu, A. Moschitti, TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection, in: *AAAI, 2020*.
- [8] M. T. R. Laskar, X. Huang, E. Hoque, Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task, in: *Proceedings of The 12th Language Resources and Evaluation Conference, 2020*, pp. 5505–5514.
- [9] M. Norkute, N. Herger, L. Michalak, A. Mulder, S. Gao, Towards explainable ai: Assessing the usefulness and impact of added explainability features in legal document summarization, in: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021*, pp. 1–7.
- [10] M. L. Hickmann, F. Wurzberger, M. Hoxhalli, A. Lochner, J. Töllich, A. Scherp, Analysis of graphsum’s attention weights to improve the explainability of multi-document summarization, in: *The 23rd International Conference on Information Integration and Web Intelligence, 2021*, pp. 359–366.
- [11] Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang, J. Li, F. Wu, Bertgcn: Transductive text classification by combining gcn and bert, *arXiv preprint arXiv:2105.05727* (2021).
- [12] AWS-Textract, Textract-aws ocr engine, <https://aws.amazon.com/textract> (2019).
- [13] Tesseract ocr engine, <https://github.com/tesseract-ocr/tesseract>, 2006.
- [14] R. Ronen, S. Tsiper, O. Anshel, I. Lavi, A. Markovitz, R. Manmatha, Glass: Global to local attention for scene-text spotting, in: *ECCV, Springer, 2022*, pp. 249–266.
- [15] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, J. Yan, Fots: Fast oriented text spotting with a unified network, in: *CVPR, 2018*, pp. 5676–5685.
- [16] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, X. Bai, Aster: An attentional scene text recognizer with flexible rectification, *PAMI 41* (2018) 2035–2048.
- [17] W. Liu, C. Chen, K.-Y. Wong, Char-net: A character-aware neural network for distorted scene text recognition, in: *AAAI, volume 32, 2018*.
- [18] Y. Zheng, W. Qin, D. Wijaya, M. Betke, Lal: Linguistically aware learning for scene text recognition, in: *Proceedings of the 28th ACM International Conference on Multimedia, 2020*, pp. 4051–4059.
- [19] Y. Baek, S. Shin, J. Baek, S. Park, J. Lee, D. Nam, H. Lee, Character region attention for text spotting, in: *ECCV, Springer, 2020*, pp. 504–521.
- [20] Y. Liu, C. Shen, L. Jin, T. He, P. Chen, C. Liu, H. Chen, Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting, *arXiv preprint arXiv:2105.03620* (2021).
- [21] S. Fang, H. Xie, Y. Wang, Z. Mao, Y. Zhang, Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition, in: *CVPR, 2021*, pp. 7098–7107.
- [22] X. Yue, Z. Kuang, C. Lin, H. Sun, W. Zhang, Robustscanner: Dynamically enhancing positional clues for robust text recognition, in: *ECCV, Springer, 2020*, pp. 135–151.
- [23] M. Jaderberg, A. Vedaldi, A. Zisserman, Deep features for text spotting, in: *ECCV, Springer, 2014*, pp. 512–528.
- [24] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: *Proceedings of the IEEE conference on computer vision and pattern recognition, 2017*, pp. 7291–7299.
- [25] J. Van Landeghem, R. Tito, Ł. Borchmann, M. Pietruszka, P. Joziak, R. Powalski, D. Jurkiewicz, M. Coustaty, B. Anckaert, E. Valveny, et al., Document understanding dataset and evaluation (dude), in: *Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023*, pp. 19528–19540.
- [26] J. Kumar, P. Ye, D. Doermann, Structural similarity for document image classification and retrieval, *Pattern Recognition Letters 43* (2014) 119–126.
- [27] J.-P. T. Guillaume Jaume, Hazim Kemal Ekenel, Funsd: A dataset for form understanding in noisy scanned documents, in: *ICDAR-OST, 2019*.
- [28] I.-S. Stoian, I.-C. Sandu, D. Voinea, A.-I. Popa, Unstructured object matching using co-salient region

- segmentation, in: CVPR Workshops, 2022, pp. 5051–5060.
- [29] I.-C. Sandu, D. Voinea, A.-I. Popa, Large sequence representation learning via multi-stage latent transformers, in: Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 4633–4639.
- [30] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, NIPS 28 (2015) 91–99.
- [31] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: CVPR, 2016, pp. 779–788.
- [32] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in: CVPR, 2020, pp. 10781–10790.
- [33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: ICCV, 2017.
- [34] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: ECCV, Springer, 2020, pp. 213–229.
- [35] Y. Li, H. Mao, R. Girshick, K. He, Exploring plain vision transformer backbones for object detection, in: ECCV, Springer, 2022, pp. 280–296.
- [36] S. Matcovici, D. Voinea, A.-I. Popa, k-nn embedded space conditioning for enhanced few-shot object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 401–410.
- [37] Google inc. google fonts., <https://fonts.google.com/>, 2010.
- [38] T. Stanisławek, F. Galiński, A. Wróblewska, D. Lipiński, A. Kaliska, P. Rosalska, B. Topolski, P. Biecek, Kleister: key information extraction datasets involving long documents with complex layouts, in: ICDAR, Springer, 2021, pp. 564–579.
- [39] K. Lang, T. Mitchell, Newsgroup 20 dataset, 1999.
- [40] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, Textsnake: A flexible representation for detecting text of arbitrary shapes, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 20–36.
- [41] M. Liao, Z. Zou, Z. Wan, C. Yao, X. Bai, Real-time scene text detection with differentiable binarization and adaptive scale fusion, PAMI (2022).
- [42] T. Kil, S. Kim, S. Seo, Y. Kim, D. Kim, Towards unified scene text spotting based on sequence generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15223–15232.
- [43] W. Yu, Y. Liu, W. Hua, D. Jiang, B. Ren, X. Bai, Turning a clip model into a scene text detector, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6978–6988.
- [44] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: NIPS, 2017.
- [45] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: ICCV, 2021, pp. 10012–10022.
- [46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017.
- [47] N. Lu, W. Yu, X. Qi, Y. Chen, P. Gong, R. Xiao, X. Bai, Master: Multi-aspect non-local network for scene text recognition, Pattern Recognition 117 (2021) 107980.
- [48] J. Lee, S. Park, J. Baek, S. J. Oh, S. Kim, H. Lee, On recognizing texts of arbitrary shapes with 2d self-attention, in: CVPR Workshops, 2020, pp. 546–547.
- [49] D. Bautista, R. Atienza, Scene text recognition with permuted autoregressive sequence models, in: European Conference on Computer Vision, Springer Nature Switzerland, Cham, 2022, pp. 178–196. URL: https://doi.org/10.1007/978-3-031-19815-1_11. doi:10.1007/978-3-031-19815-1_11.
- [50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: ECCV, 2014.
- [51] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: ICLR, 2019.