Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20162-6

Template-based text field segmentation for ID documents using dynamic squeezeboxes packing

Michael Zingerenko[1,2] · **Elena Limonova**[2,3] · **Vladimir V. Arlazarov**[2,3]

Received: 16 May 2024 / Revised: 23 July 2024 / Accepted: 23 August 2024

Abstract
In this paper, we focus on the problem of text field segmentation in identity documents. These documents, characterized by their fixed layouts, present an opportunity to apply computationally efficient template-based algorithms. We consider the Dynamic Squeezeboxes Packing method and demonstrate its integration into document recognition systems, utilizing a single sample per document type. We benchmark text field segmentation on the MIDV-2019 public dataset using standard intersection-over-union and our custom intersection-over-template metrics, while also measuring processing time. We demonstrate that Dynamic Squeezeboxes Packing maintains competitive quality compared to text in the wild methods (EAST, CRAFT) and named-entity recognition method (LayoutLMv2). A significant advantage of this method is its processing speed, averaging 9 ms per image on the x86_64 platform, which is substantially faster than EAST (980 ms), CRAFT (2030 ms), and LayoutLMv2 (2210 ms). The obtained results suggest that the considered method has strong potential as a method in document image analysis, particularly for processing identity documents.

Keywords Text segmentaion · Identity documents · Dynamic squeezeboxes packing · Template matching · MIDV-2019

Elena Limonova and Vladimir V. Arlazarov contributed equally to this work.

✉ Michael Zingerenko
zingerenko.mv@phystech.edu

Elena Limonova
limonova@smartengines.com

Vladimir V. Arlazarov
vva@smartengines.com

1 Moscow Institute of Physics and Technology, 9 Institutskiy per, Dolgoprudnii 141701, Russia

2 Smart Engines Service LLC, Prospekt 60-Letiya Oktyabrya, 9, Moscow 117312, Russia

3 Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Ulitsa Vavilova, 44, Moscow 119333, Russia

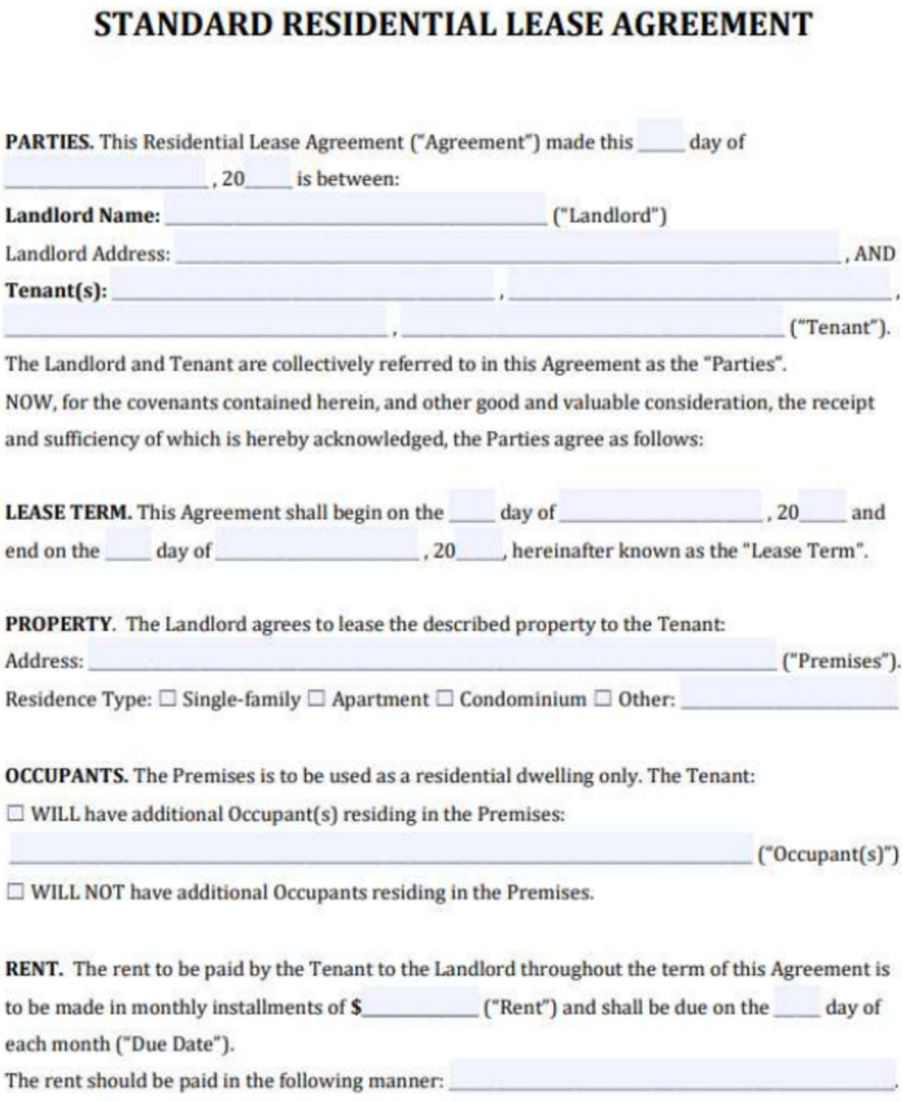
Published online: 18 September 2024

1 Introduction

The importance of document recognition is increasing across various sectors, notably in banking, booking, and public services. The demand for sophisticated solutions for efficient and accurate information extraction is underscored by a plethora of recent studies [1–4].

The process of document recognition can benefit from a pre-existing knowledge about the document's layout and attribute features. Document structures exhibit a wide array of variability and, for analytical purposes, can be categorized into three distinct types:

- Flexible forms, which adhere to changeable templates, encompassing documents like questionnaires, notifications, declarations, and bank plastic cards, exemplified in Fig. 1.



STANDARD RESIDENTIAL LEASE AGREEMENT

PARTIES. This Residential Lease Agreement ("Agreement") made this _____ day of _____, 20____ is between:

Landlord Name: _____ ("Landlord")

Landlord Address: _____, AND

Tenant(s): _____, _____, _____, _____ ("Tenant").

The Landlord and Tenant are collectively referred to in this Agreement as the "Parties".

NOW, for the covenants contained herein, and other good and valuable consideration, the receipt and sufficiency of which is hereby acknowledged, the Parties agree as follows:

LEASE TERM. This Agreement shall begin on the _____ day of _____, 20____ and end on the _____ day of _____, 20____, hereinafter known as the "Lease Term".

PROPERTY. The Landlord agrees to lease the described property to the Tenant:

Address: _____ ("Premises").

Residence Type: ☐ Single-family ☐ Apartment ☐ Condominium ☐ Other: _____

OCCUPANTS. The Premises is to be used as a residential dwelling only. The Tenant:

☐ WILL have additional Occupant(s) residing in the Premises:

_____ ("Occupant(s)")

☐ WILL NOT have additional Occupants residing in the Premises.

RENT. The rent to be paid by the Tenant to the Landlord throughout the term of this Agreement is to be made in monthly installments of $_____ ("Rent") and shall be due on the _____ day of each month ("Due Date").

The rent should be paid in the following manner: _____.

Fig. 1 Flexible form

- Rigid forms, characterized by standard template, include identity documents such as passports, ID cards, driver's licenses, etc. (see Fig. 2).
- Non-template documents, devoid of a standardized format, covering various types of business correspondence, demonstrated in Fig. 3.

In this paper, our focus is directed towards identity documents, which are classified under the category of rigid forms. These documents can be broadly classified into two types: single-page and multi-page. However, multi-page ID documents can be efficiently processed sequentially page by page. Consequently, our study primarily emphasizes the processing methodologies applicable to single-page identity documents. We also consider the case of only one document image available which correnspponds to scanner or single photo setup. To provide a comprehensive understanding, we outline possible recognition pipeline.

1. **Document detection:** The initial stage of recognizing the document within an image, involving document localization and dewarping (see example in Fig. 4a).
2. **Document type identification:** The subsequent step of classifying the document type, such as ID cards, driver's licenses, etc.
3. **Zone segmentation:** Segmenting the document into regions of interest, which we designate as zones, like text segments or photographs (see example in Fig. 4b).
4. **Data extraction:** Identifying key information within located zones, such as name in a passport's text zone (see example in Fig. 4c).
5. **Information conversion:** Converting obtained information into a user-defined format (see example in Fig. 4d).

This study examines various approaches for identifying text fields within documents, drawing from both neural networks and classical algorithms. While neural network-based

Fig. 2 Rigid form

Date: [Date letter is sent]

Subject: [The subject of your letter]

Greeting: Dear [recipient],

Purpose of your letter: I am writing to [reason for your letter].

Explanation: I have [number] vacation days available and would like to use [number] days for [type of vacation]. I will be gone from [date leaving] to [date of return home] and will return to work [date of return to work]. [Additional details]

Status update: My current projects [status of current projects and duties].

Plan for coverage: I have asked [name] to take over [specific responsibilities] while I am gone.

Availability: I [will have access to] or [will not have access to] communication. I [will respond to messages] or [will not be able to respond to messages].

Final request for approval: Please respond by [date] for me to finalize my travel arrangements.

Closing: Sincerely,

Signature: [Your name]

Title: [Your title]

Phone: [Your phone number]

Fig. 3 Non-template document

methods offer flexibility, they face limitations due to the scarcity of public datasets for diverse ID documents. Also, they do not utilize spatial information available for ID documents type. Conversely, template-based methods can provide more computationally efficient and robust solutions for practical applications. However, they face challenges in incorporating different document templates, which can take considerable time and require human interaction. In this paper, we consider template-matching approach to text field segmentation in identity documents based on Dynamic Squeezeboxes Packing (DSP) [5] method.

Our contribution is as follows.

- We demonstrate a way to efficiently incorporate DSP into document recognition system using only one sample per document type.
- We introduce a baseline for text field segmentation in identity documents on MIDV-2019 dataset [6] and demonstrate standard intersection-over-union (IoU), problem-specific intersection-over-template (IoT) metrics to access segmentation quality and final text recognition accuracy.
- We compare the mainstream text-in-the-wild EAST [7], CRAFT [8] methods, named-entity recognition LayoutLMv2 [9] method and considered DSP-based approach.
- We also compare inference time on different computational platforms, specifically ARM typical for edge devices and desktop x86_64 and show that DSP-based approach is by about two orders of magnitude faster.

(a) Document Detection

(b) Zone Segmentation

(c) Data Extraction

(d) Information conversion

Fig. 4 Sample images from document recognition pipeline

2 Related work

The area of data extraction from images encompasses multiple crucial steps, one of the most vital being the segmentation of text fields within an image through bounding boxes, followed by their conversion from image to text. Over the past two decades, text segmentation algorithms have undergone significant evolution. The earliest methods employed basic morphological operators [10, 11]. These techniques evolved with the incorporation of heuristic rules [12]. Contemporary successors of these algorithms utilize predetermined knowledge about the document's layout and fields, commonly referred to as template-based methods [13, 14].

In this context, the Dynamic Squeezeboxes Packing method (DSP) [5] is a notable template-based method. DSP significantly reduces computational complexity compared to previous methods [15], lowering it from $O(NW^2)$ to $O(NW)$, where N represents the number of text fields, and W the number of feasible locations for one field. This advancement marks a substantial leap in computational efficiency. However, the DSP method lacks comparative benchmarks against public datasets and its practical application in real-world scenarios, such as document recognition systems. Furthermore, the accuracy assessments in [5] were conducted post-OCR and limit the ability to gauge the direct impact of the template application.

On the other hand, modern neural networks [16, 17] have emerged as highly accurate alternatives, especially for visually rich documents like identity cards. These methods leverage

the abundant visual features present in such documents, with object detection and semantic segmentation being the predominant techniques employed for field extraction [18, 19].

A remarkable development in this field is the EAST algorithm [7] – Efficient and Accurate Scene Text Detector. This algorithm represents a significant departure from traditional multi-stage OCR frameworks. Introduced in 2017, EAST employs a single neural network that is adept at predicting text in various orientations and shapes within images. Authors focused on optimizing loss functions and neural network architecture, thus establishing EAST as a benchmark in text segmentation tasks.

Another notable advancement is the CRAFT algorithm [8] – Character Region Awareness for Text Detection. CRAFT introduces an innovative dual-dataset strategy, utilizing character-level annotations from synthetic images and estimated character-level ground truths from real images. This approach significantly improves character recognition, especially in texts exhibiting complex attributes like curvature and orientation variations. A key feature of CRAFT is its novel representational technique, which has notably enhanced the precision of text detection.

In contrast, methods based on Natural Language Processing [20, 21] transform documents into 1D sequences of tokens and employ named entity recognition models to classify each word. While effective for documents with simple layouts like books or articles, they are less efficient for documents with complex visual structures, such as invoices and tax notices, where structural and visual information are crucial for performance.

A recent and significant contribution to this field is the LayoutLMv2 [9]. This method utilizes visual information in the training process, incorporating not just text embeddings of each token, but also the relative positions of tokens and the corresponding image feature maps. Drawing inspiration from the BERT model [22], LayoutLMv2 was pre-trained on a scanned document classification task to learn the interplay between text and layout information. While LayoutLMv2 has extended its capabilities to sophisticated entity recognition, it faces limitations in terms of the range of its pretrained models and computational efficiency, being currently restricted to only four document types.

In summary, image text extraction advancements have progressed from basic morphological processing to modern neural network-based approaches like EAST and CRAFT algorithms and current template-based approaches like DSP. These methods demonstrate significant improvements in accuracy and efficiency, particularly for visually complex documents. Additionally, approaches rooted in Natural Language Processing, such as LayoutLMv2, show promising results in incorporating both textual and visual information.

3 Proposed approach

3.1 Document model overview

Let us introduce a model of an identity document. We consider a distortion-free image of a document, devoid of projective transformations and other alterations. The document encompasses three primary types of elements, exemplified in Fig. 5a: background, static and content elements.

Background elements which may include patterns or designs intrinsic to the document. Background can contain complex features, for instance, the classic "guilloché" pattern (yellow zone in Fig. 5b). It is commonly found in identity or registration documents, providing protection against forgery. At the same time, bank plastic cards often exhibit dynamic

(a) (b)

Fig. 5 a) Identity document elements: static (red), personal data content (green) and parts of background (yellow); b) "guilloché" pattern in the background (yellow). Images taken from MIDV-2019 [6]

backgrounds, which vary significantly from one series to another. These backgrounds, while visually appealing, generally do not convey informational attributes and are primarily decorative.

Static elements frequently appearing as textual segments within the document. These often encompass headings and field labels, such as "name" or "address", providing structural context to the document. Other static elements in the document can include boundary lines and checkboxes, which play a crucial role in defining the document's structure.

Content elements comprising data fields that contain information that changes from document to document. Such fields typically contain textual data; however, in some instances, alternative forms like barcode fields are utilized. It is noteworthy that barcodes can occasionally function as static elements, thus aiding in the classification of the document.

The arrangement and interplay of these elements define a document's template.

3.2 Template parametrization

We introduce a key element in for our document templates: zones. Each zone corresponds to a specific area of interest within the document, such as zones designated for photographs, signatures, data, ID numbers, etc., as illustrated in Fig. 4b. In the context of text segmentation, our focus is primarily on text zones. Each of these zones is conceptualized as a single-column structure, which can be divided into lines with one or more text fields and text-free space between them. We suppose that such image regions are provided by document detector.

Let us introduce a parametric document layout description. Each line within the document is characterized by a set of parameters designed to specify its properties. These parameters are:

- **Height** (h). This parameter represents the maximum potential height of a line, measured in pixels. It sets the upper boundary for the line's height.
- **Delta height** (Δh). This parameter defines the maximum allowable reduction in the line's height from its maximum value, thereby establishing the possible height range for the line as $[h - \Delta h, h]$.
- **Type**. We distinguish between two line types: 'gap', referring to areas devoid of content fields with data, and 'text', indicating areas containing content fields.

Furthermore, lines classified as 'text' type are structured through the combination of 'field' entities, which encode interleaving gaps and text areas. The left and the right entities must always be gaps. All the fields have the following parameters:

- **Width** (w). This parameter specifies the maximum width that a field can span, measured in pixels.
- **Delta width** (Δw). This parameter determines the maximum reduction in the field's width from its maximum value, thus defining the possible width range as $[w - \Delta w, w]$.
- **Type**: 'gap' or 'text'.

Text fields also have:

- **Name**. This parameter serves as an identifier for the field, facilitating its recognition and processing.
- **Is optional** ($is\_optional$). A boolean indicator specifying whether the presence of the field is mandatory or optional.

These parameters define the structural and content-related aspects of lines and fields within a document template. It is noteworthy to mention that a 'text' line may encompass multiple 'text' fields, each of them surrounded by 'gap' fields. An example of these template parameters and their practical application is depicted in Fig. 6. These templates are created by human operators for each type of document, allowing for multiple templates for a single document to enhance the representation of various scenarios within the document's layout.

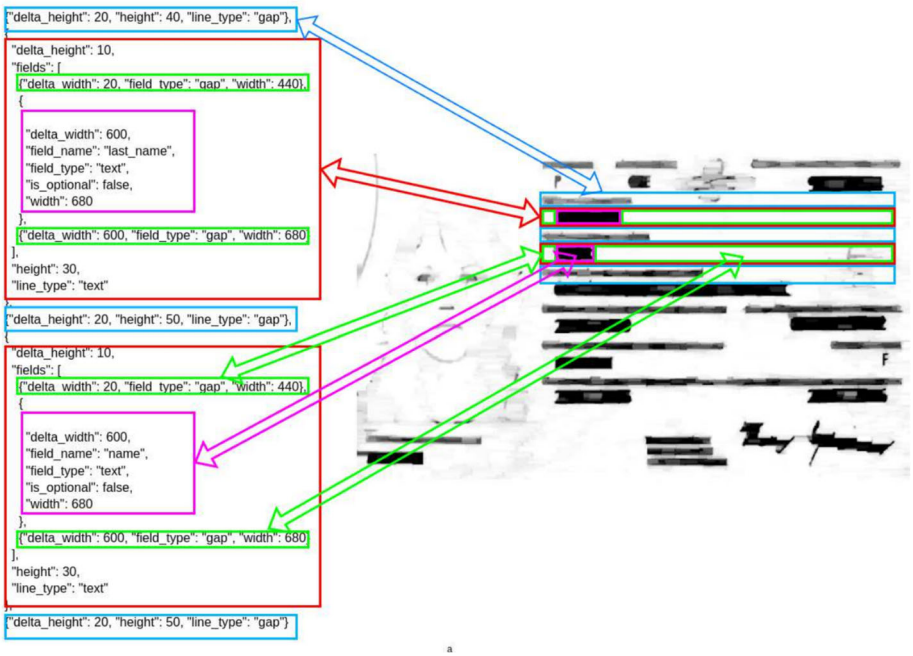


Fig. 6 Example of template parameters and their functions

3.3 Template-based segmentation using DSP

The input of text field segmentation is dewarped greyscale document zone $I(x, y)$ as shown in Fig. 7a and a list of document templates T_i, $i \in 1, \ldots, K$, where K is the number of templates for current zone. Then, for each template T_i, we apply Dynamic Squeezeboxes Packing method to processed zone image to gather exact text fields locations and we pick the result with the highest score.

In this method, both the size and position of fields can vary within template limits, introducing adaptability for different document types. This adaptability is a key feature, as it allows the algorithm to be fine-tuned for a variety of document formats, ensuring robustness and versatility in text segmentation. DSP method expects that text in images has lower brightness compared to the background, which is normally true for identity documents.

Before applying DSP algorithm, authors of [5] used a sequence of image processing techniques, including morphological closing and opening, subtraction, and auto-contrast, tailored to the dimensions of the document's structural elements. This preprocessing aims to increase intra-class variance and, to some degree, remove static elements while keeping content fields. These operations are illustrated in Fig. 7b. Such an approach does not work very well when the size of static elements is the same as the size of the content or when there is a rich background. In these cases, one can use a simple neural network for preprocessing. This stage can be performed only once, as it is the same for all zone templates.

The DSP method starts with determining the approximate locations of the text fields. All the fields are considered as mandatory and having fixed heights and widths of $h - 4/5\Delta h$ and $w - 4/5\Delta w$, where h and w are the field sizes from the current template, and Δh and Δw are their possible variations. Then a two-class segmentation task (text "fields" and background "gaps") is solved via maximization of an inter-class variance function [23] using dynamic programming and coarse grid:

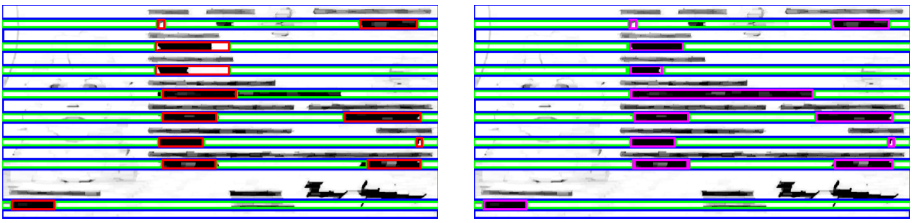
$$V = \omega_0\omega_1(\mu_0 - \mu_1)^2 \text{sign}(\mu_0 - \mu_1) = \omega_0\omega_1(\mu_0 - \mu_1)|\mu_0 - \mu_1| \longrightarrow \max, \quad (1)$$

where ω_0 and ω_1 are fractions of the text and the background classes from all the image pixels, μ_0 and μ_0 are mean intra-class brightness values. The result is shown in Fig. 8a. We can see, that the field positions are determined correctly, but need some fine-tuning as well as field sizes.

Finally, the field position and size adjustment is performed through a coordinate descent method. This process continues until no significant improvement is observed, or a maximum iteration limit is reached, culminating in the accurate extraction of text fields, as shown in Fig. 8b. Obtained inter-class variance is used as a score for current template.

(a) Input zone image (b) Preprocessed input zone

Fig. 7 Image preprocessing



(a) Dynamic programming output

(b) Coordinate descent output

(c) Final result after rejector

Fig. 8 Example of template-based segmentation using DSP

3.4 Rejector for optional and empty fields

The DSP algorithm considered all the fields mandatory, but templates can include fields marked as optional. So, after the DSP algorithm finishes, we move to the rejection phase to remove absent optional fields. We compute a novel metric upon the bounding box of a potential field and reject it if it is below the threshold:

$$\frac{\mu_{gap} - \mu_{field}}{\mu_{gap}} < T, \tag{2}$$

where μ_{field} is the mean brightness inside the identified bounding box for a field, μ_{gap} is the mean brightness for a gap area outside the field, and T is the rejection threshold. Figure 9 illustrates an example of the field and gap areas.

4 Experiments

4.1 Experimental setup

We use the MIDV-2019 dataset [6] to evaluate the quality of text field segmentation in identity documents. Today, it is the most diverse publicly available dataset of identity document images. It comprises 50 types of documents from various countries, each annotated with bounding boxes for every data text field present in the document. It contains 6000 sample

Fig. 9 The green box represents the field, and the rest of the image is the gap

images with different positions, rotations and complex backgrounds behind the documents. Thus, MIDV-2019 perfectly suits our task because:

- it contains rigid-form documents and their distortion-free templates,
- and allows us to evaluate the quality of field segmentation using bounding boxes.

We consider four different methods:

- template-based DSP approach customly implemented in C++;
- text-in-the-wild EAST detector, pre-trained model implemented in PyTorch;[1]
- text-in-the-wild CFART detector, pre-trained model implemented in PyTorch;[2]
- named entity recognition-based LayoutLMv2, pre-trained model implemented in PyTorch.[3]

To assess the quality of the algorithms, we employ two specific metrics. The first metric is the standard measure for bounding boxes, known as Intersection over Union (IoU):

$$IoU = \frac{S(B \cap B_{gt})}{S(B \cup B_{gt})}, \tag{3}$$

where B is obtained bounding box for a field, B_{gt} is a bounding box for ground truth and $S(\cdot)$ is an area of the input figure.

The second metric, tailored to the specific requirements of our task, is Intersection over Template (IoT):

$$IoT = \frac{S(B \cap B_{gt})}{S(B_{gt})} \tag{4}$$

The rationale behind IoT is that it is acceptable for bounding boxes to be larger than the ground truth for effective text extraction (see example in Fig. 10). Ideally, the IoT value should approach one, indicating precise bounding box placement, while maintaining an IoU value greater than 0.5 for reliable detection.

4.2 Experimental evaluation

For the accurate assessment of processing time in real-world scenarios, our study encompassed two distinct types of computational devices: ARM Cortex A78AE with ARMv8 architecture, representing mobile and edge device architectures, and AMD Ryzen 7 1700 with x86_64, exemplifying a desktop platform.

Note that we evaluated text-in-the-wild methods under two distinct scenarios. The first scenario involves applying these methods to full images. It corresponds better to real life; however, in this approach, it is necessary to implement a filtering process to isolate text segments that align with the ground truth template. So, it requires additional post-processing steps to extract only the relevant information specific to each document type. Our experiment considered detected text fields with IoU > 0.1 with the ground truth bounding boxes for each field. This way, we determined that the text refers to a given field.

The second scenario focuses exclusively on the document data zone as DSP-based and LayoutLMv2 methods. In practical applications, this approach is more direct, as it targets specific areas of the document known to contain relevant information. However, there still

[1] https://github.com/PaddlePaddle/PaddleOCR

[2] https://github.com/JaidedAI/EasyOCR

[3] https://github.com/huggingface/transformers

Fig. 10 Example bounding box bigger then ground truth

should be some previous detection step to retrieve this zone. In our work, we used document location and type from ground truth and extracted zones according to pre-defined coordinates to eliminate detector and classifier errors. In practice, one can use, for example, a method proposed in [24]. For EAST, CRAFT and LayoutLMv2 we also considered as presented only fields with IoU > 0.1 with the ground truth.

The averaged results for images from the MIDV-2019 dataset are systematically presented in Table 1. All the images were processed separately. An examination of the table reveals that EAST and CRAFT on full images and LayoutLMv2 and DSP yield comparable outcomes concerning mean IoU and IoT metrics. If we consider only zone part, EAST and CRAFT have surpassed DSP method by quality metrics. However, the DSP method distinguishes itself by offering a remarkably superior processing time, outperforming the other methods by a factor of 100, even for the zone part of the image. In our analysis, we also evaluated the accuracy of text obtained by the PaddleOCR. To estimate text recognition accuracy in a unified manner, we extracted text fields from the document images using obtained bounding boxes and recognized them using the default recognition algorithm from the PaddleOCR framework.

The comparative analysis of zone image processing demonstrates that EAST and CRAFT algorithms on zone images outperform LayoutLMv2 and DSP regarding recognition quality.

Table 1 Text field segmentation quality and time

Method	mIoU	mIoT	Recognition Quality (PaddleOCR)	x86_64 mean time, ms	ARMv8 mean time, ms
EAST (full image)	0.56	0.59	34.07%	980	8840
EAST (zone)	**0.62**	**0.71**	**46.17%**	827	6690
CRAFT (full image)	0.48	0.62	33.11%	2038	5675
CRAFT (zone)	0.57	0.67	45.38%	2006	3967
LayoutLMv2 (zone)	0.35	0.61	19.83%	2211	2390
DSP (zone)	0.50	0.52	43.93%	**9**	**12**

The bold entries represent the best metrics and times in each column

LayoutLMv2 often generates too big bounding boxes, so when these boxes are passed to PaddleOCR, it captures additional text and gives an incorrect result. So, in its current form, LayoutLMv2 is unsuitable for extracting information from identification documents of the types considered. EAST and CRAFT launched on the full image showed noticeably worse results than for the document zone. It means that detecting regions of interest in advance can improve the results. The DSP method works significantly better than LayoutLMv2, EAST (full image), and CRAFT (full image). However, its accuracy is slightly lower than that of EAST and CRAFT launched on the document zone. However, while accepting a marginal increase in error rates, the DSP method offers a computational speed that is approximately 100 times higher. This makes it an exceptionally advantageous tool for real-time document recognition systems, where rapid data processing is of paramount importance.

5 Discussion and limitations

In this paper, we introduced a computationally efficient template-based approach for document text segmentation using Dynamic Squeezeboxes Packing (DSP). Our method demonstrates very fast inference and competitive accuracy. In contrast to modern approaches, our method does not utilize neural networks, instead, it is based on classical algorithms like dynamic programming. However, it is important to acknowledge that the generation of templates within this method necessitates manual intervention by a human operator, which introduces a potential bottleneck in terms of scalability and segmentation accuracy.

Furthermore, our strategy of relying on a single sample for each document type presents limitations in capturing the full spectrum of variability observed in real-world document types. While document templates are based on static fields that tend to remain consistent over time, identity documents can still vary depending on factors such as the issue year and region.

As illustrated in Table 1, there is a noted deficiency in recognition accuracy for our method compared to the EAST detector, which is two orders of magnitude slower. We analyzed segmentation errors and identified that they are primarily attributed to the limitations of our morphological preprocessing technique. Figure 11 demonstrates the preprocessed image of a driving license, showing how inaccurate preprocessing leads to incorrect bounding boxes and results in a decrease in overall accuracy. These challenges are posed by complex, heavily structured backgrounds.

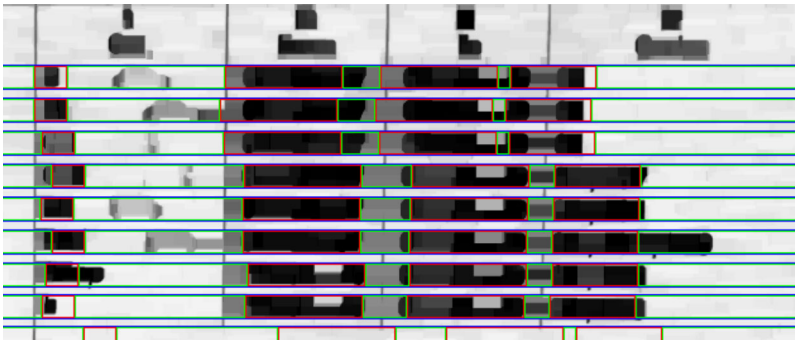


Fig. 11 Example of erroneous morphological preprocessing

Our future research will focus on:

- Automating the template generation process,
- Enhancing the method's robustness by exploring and implementing potential solutions to improve document preprocessing,
- Using more diverse datasets with identity documents and testing different OCR engines.

6 Conclusion

This paper presents a computationally efficient template-based method for text field segmentation in identity documents, employing the Dynamic Squeezeboxes Packing (DSP), proving that one sample per document is sufficient for accurate matching. We presented benchmarking results for the public MIDV-2019 dataset and established a baseline for text field segmentation in identity documents using standard IoU, task-specific Intersection-over-Template metric and text recognition accuracy. DSP method maintained competitive quality in segmentation when compared with advanced text-in-the-wild methods (EAST, CRAFT) and the named-entity recognition method (LayoutLMv2). A noteworthy finding of our research is the exceptional processing speed of the DSP method, averaging 9 ms per image on the x86_64 platform, significantly outpacing other contemporary methods. In summary, document text segmentation using DSP method has demonstrated significant potential for further development and practical application.

Funding This research received no external funding.

Data Availability Data sharing is not applicable.

Declarations

Competing Interests The authors declare no conflicts of interest.

References

1. Baviskar D, Ahirrao S, Potdar V, Kotecha K (2021) Efficient automated processing of the unstructured documents using artificial intelligence: a systematic literature review and future directions. IEEE Access 9:72894–72936
2. Khan A, Baharudin B, Lee LH, Khan K (2010) A review of machine learning algorithms for text-documents classification. J Adv Inf Technol 1(1):4–20
3. Liu L, Wang Z, Qiu T, Chen Q, Lu Y, Suen CY (2021) Document image classification: progress over two decades. Neurocomputing 453:223–240
4. Arlazarov V, Andreeva EI, Bulatov K, Nikolaev D, Petrova O, Savelev BI, Slavin O (2022) Document image analysis and recognition: a survey. Comput Opt. https://doi.org/10.18287/2412-6179-co-1020
5. Povolotskiy MA, Tropin DV (2019) Dynamic programming approach to template-based ocr. In: Eleventh international conference on machine vision (ICMV 2018), vol 11041. SPIE, pp 485–492
6. Bulatov K, Matalov D, Arlazarov VV (2020) Midv-2019: challenges of the modern mobile-based document ocr. In: Twelfth International Conference on Machine Vision (ICMV 2019), vol 11433. SPIE, pp 717–722
7. Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, Liang J (2017) East: an efficient and accurate scene text detector, pp 2642–2651. https://doi.org/10.1109/CVPR.2017.283
8. Baek Y, Lee B, Han D, Yun S, Lee, H (2019) Character region awareness for text detection. In: 2019 IEEE/CVF Conference on computer vision and pattern recognition (CVPR), pp 9357–9366. https://doi.org/10.1109/CVPR.2019.00959

9. Xu Y, Xu Y, Lv T, Cui L, Wei F, Wang G, Lu Y, Florêncio DAF, Zhang C, Che W, Zhang M, Zhou L (2020) Layoutlmv2: multi-modal pre-training for visually-rich document understanding. CoRR. abs/2012.14740 2012.14740
10. Hirata NST, Barbera J, Terada R (2000) Text segmentation by automatically designed morphological operators. In: Proceedings 13th Brazilian symposium on computer graphics and image processing (Cat. No.PR00878), pp 284–291. https://doi.org/10.1109/SIBGRA.2000.883924
11. Slugin DG, Arlazarov VV (2017) Text fields extraction based on image processing. Trudy ISA RAN (Proceedings of ISA RAS). 67(4):65–73. FRC CSC RAS
12. Ibrahim Z, Isa D, Rajkumar R (2008) Text and non-text segmentation and classification from document images. In: 2008 International conference on computer science and software engineering, vol 1, pp 973–976. https://doi.org/10.1109/CSSE.2008.1516
13. Rusiñol M, Benkhelfallah T, dAndecy VP (2013) Field extraction from administrative documents by incremental structural templates. In: 2013 12th International conference on document analysis and recognition, pp 1100–1104. https://doi.org/10.1109/ICDAR.2013.223
14. Sun Y, Mao X, Hong S, Xu W, Gui G (2019) Template matching-based method for intelligent invoice information identification. IEEE Access 7:28392–28401
15. Felzenszwalb PF, Zabih R (2010) Dynamic programming and graph algorithms in computer vision. IEEE Trans Pattern Anal Mach Intell 33(4):721–740
16. El Bahi H, Zatni A (2019) Text recognition in document images obtained by a smartphone based on deep convolutional and recurrent neural network. Multimed Tools Appl 78. https://doi.org/10.1007/s11042-019-07855-z
17. Andreeva E, Arlazarov V, Gayer A, Dorokhov E, Sheshkus A, Slavin O (2019) Document recognition method based on convolutional neural network invariant to 180 degree rotation angle. Journal of information technologies and computing systems (JITCS). https://doi.org/10.14357/20718632190408
18. Hao L, Gao L, Yi X, Tang Z (2016) A table detection method for pdf documents based on convolutional neural networks. In: 2016 12th IAPR Workshop on document analysis systems (DAS), pp 287–292. https://doi.org/10.1109/DAS.2016.23
19. Yang X, Yumer E, Asente P, Kraley M, Kifer D, Lee Giles C (2017) Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5315–5324
20. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition
21. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. 1802.05365
22. Devlin J, Chang M, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. 1810.04805
23. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
24. Skoryukina N, Arlazarov VV, Nikolaev DP (2020) Fast method of id documents location and type identification for mobile and server application. In: ICDAR 2019, pp. 850–857. The Institute of Electrical and Electronics Engineers (IEEE), Manhattan, New York, U.S. https://doi.org/10.1109/ICDAR.2019.00141