OCR-free Document Understanding Transformer

Geewook Kim[1*], Teakgyu Hong[4†], Moonbin Yim[2†], Jeongyeon Nam[1], Jinyoung Park[5†], Jinyeong Yim[6†], Wonseok Hwang[7†], Sangdoo Yun[3], Dongyoon Han[3], and Seunghyun Park[1]

[1]NAVER CLOVA [2]NAVER Search [3]NAVER AI Lab
[4]Upstage [5]Tmax [6]Google [7]LBox

Abstract. Understanding document images (*e.g.*, invoices) is a core but challenging task since it requires complex functions such as *reading text* and a *holistic understanding of the document*. Current Visual Document Understanding (VDU) methods outsource the task of reading text to off-the-shelf Optical Character Recognition (OCR) engines and focus on the understanding task with the OCR outputs. Although such OCR-based approaches have shown promising performance, they suffer from 1) high computational costs for using OCR; 2) inflexibility of OCR models on languages or types of documents; 3) OCR error propagation to the subsequent process. To address these issues, in this paper, we introduce a novel OCR-free VDU model named **Donut**, which stands for **Do**cume**n**t **u**nderstanding **t**ransformer. As the first step in OCR-free VDU research, we propose a simple architecture (*i.e.*, Transformer) with a pre-training objective (*i.e.,* cross-entropy loss). Donut is conceptually simple yet effective. Through extensive experiments and analyses, we show a simple OCR-free VDU model, Donut, achieves state-of-the-art performances on various VDU tasks in terms of both speed and accuracy. In addition, we offer a synthetic data generator that helps the model pre-training to be flexible in various languages and domains. The code, trained model, and synthetic data are available at https://github.com/clovaai/donut.

Keywords: Visual Document Understanding, Document Information Extraction, Optical Character Recognition, End-to-End Transformer

1 Introduction

Document images, such as commercial invoices, receipts, and business cards, are easy to find in modern working environments. To extract useful information from such document images, Visual Document Understanding (VDU) has not been only an essential task for industry, but also a challenging topic for researchers, with applications including document classification [27,1], information extraction [22,42], and visual question answering [44,57].

* Corresponding author: gwkim.rsrch@gmail.com
† This work was done while the authors were at NAVER CLOVA.

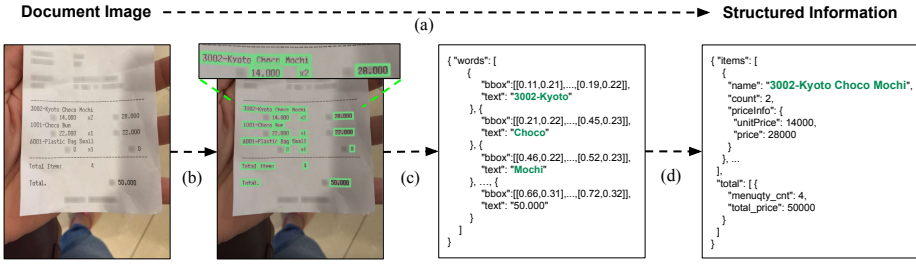
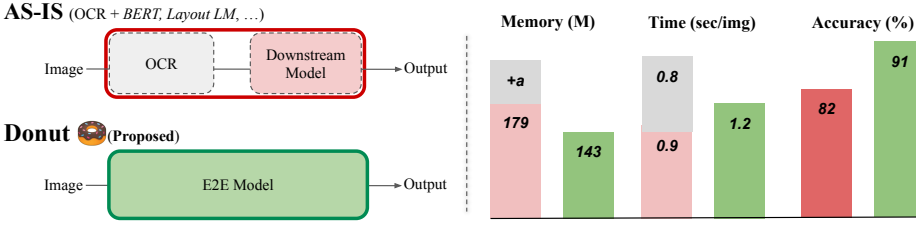


Fig. 1. The schema of the conventional document information extraction (IE) pipeline. (a) The goal is to extract the structured information from a given semi-structured document image. In the pipeline, (b) text detection is conducted to obtain text locations and (c) each box is passed to the recognizer to comprehend characters. (d) Finally, the recognized texts and its locations are passed to the following module to be processed for the desired structured form of the information



(a) Pipeline Overview. (b) System Benchmarks.

Fig. 2. The pipeline overview and benchmarks. The proposed end-to-end model, **Donut**, outperforms the recent OCR-dependent VDU models in memory, time cost and accuracy. Performances on visual document IE [45] are shown in (b). More results on various VDU tasks are available at Section 3 showing the same trend

Current VDU methods [22,24,65,64,18] solve the task in a two-stage manner: 1) reading the texts in the document image; 2) holistic understanding of the document. They usually rely on deep-learning-based Optical Character Recognition (OCR) [4,3] for the text reading task and focus on modeling the understanding part. For example, as shown in Figure 1, a conventional pipeline for extracting structured information from documents (a.k.a. document parsing) consists of three separate modules for text detection, text recognition, and parsing [22,24].

However, the OCR-dependent approach has critical problems. First of all, using OCR as a pre-processing method is expensive. We can utilize pre-trained off-the-shelf OCR engines; however, the computational cost for inference would be expensive for high-quality OCR results. Moreover, the off-the-shelf OCR methods rarely have flexibility dealing with different languages or domain changes, which may lead to poor generalization ability. If we train an OCR model, it also requires extensive training costs and large-scale datasets [4,3,39,46]. Another problem is, OCR errors would propagate to the VDU system and negatively influence subsequent processes [54,23]. This problem becomes more severe in

languages with complex character sets, such as Korean or Chinese, where the quality of OCR is relatively low [50]. To deal with this, post-OCR correction module [51,50,10] is usually adopted. However, it is not a practical solution for real application environments since it increases the entire system size and maintenance cost.

We go beyond the traditional framework by modeling a direct mapping from a raw input image to the desired output without OCR. We introduce a new OCR-free VDU model to address the problems induced by the OCR-dependency. Our model is based on Transformer-only architecture, referred to as **Do**cume**n**t **u**nderstanding **t**ransformer (**Donut**), following the huge success in vision and language [8,9,29]. We present a minimal baseline including a simple architecture and pre-training method. Despite its simplicity, **Donut** shows comparable or better overall performance than previous methods as shown in Figure 2.

We take pre-train-and-fine-tune scheme [8,65] on **Donut** training. In the pre-training phase, **Donut** learns *how to read the texts* by predicting the next words by conditioning jointly on the image and previous text contexts. **Donut** is pre-trained with document images and their text annotations. Since our pre-training objective is simple (*i.e.*, reading the texts), we can realize domain and language flexibility straightforwardly pre-training with synthetic data. During fine-tuning stage, **Donut** learns *how to understand the whole document* according to the downstream task. We demonstrate **Donut** has a strong understanding ability through extensive evaluation on various VDU tasks and datasets. The experiments show a simple OCR-free VDU model can achieve state-of-the-art performance in terms of both speed and accuracy.

The contributions are summarized as follows:

1. We propose a novel OCR-free approach for VDU. To the best of our knowledge, this is the first method based on an OCR-free Transformer trained in end-to-end manner.
2. We introduce a simple pre-training scheme that enables the utilization of synthetic data. By using our generator SynthDoG, we show **Donut** can easily be extended to a multi-lingual setting, which is not applicable for the conventional approaches that need to retrain an off-the-shelf OCR engine.
3. We conduct extensive experiments and analyses on both public benchmarks and private industrial datasets, showing that the proposed method achieves not only state-of-the-art performances on benchmarks but also has many practical advantages (e.g., *cost-effective*) in real-world applications.
4. The codebase, pre-trained model, and synthetic data are available at GitHub.[1]

2 Method

2.1 Preliminary: background

There have been various visual document understanding (VDU) methods to understand and extract essential information from the semi-structured documents such as receipts [20,25,18], invoices [49], and form documents [14,6,43].

[1] https://github.com/clovaai/donut.

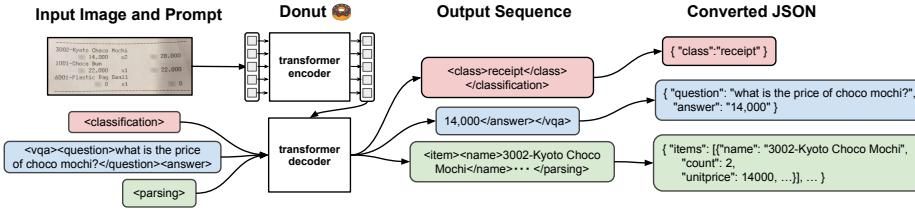


Fig. 3. The pipeline of Donut. The encoder maps a given document image into embeddings. With the encoded embeddings, the decoder generates a sequence of tokens that can be converted into a target type of information in a structured form

Earlier VDU attempts have been done with OCR-independent visual backbones [27,1,15,12,31], but the performances are limited. Later, with the remarkable advances of OCR [4,3] and BERT [8], various OCR-dependent VDU models have been proposed by combining them [22,24,23]. More recently, in order to get a more general VDU, most state-of-the-arts [64,18] use both powerful OCR engines and large-scale real document image data (e.g., IIT-CDIP [32]) for a model pre-training. Although they showed remarkable advances in recent years, extra effort is required to ensure the performance of an entire VDU model by using the off-the-shelf OCR engine.

2.2 Document Understanding Transformer

Donut is an end-to-end (i.e., self-contained) VDU model for general understanding of document images. The architecture of Donut is quite simple, which consists of a Transformer [58,9]-based visual encoder and textual decoder modules. Note that Donut does not rely on any modules related to OCR functionality but uses a visual encoder for extracting features from a given document image. The following textual decoder maps the derived features into a sequence of subword tokens to construct a desired structured format (e.g., JSON). Each model component is Transformer-based, and thus the model is trained easily in an end-to-end manner. The overall process of Donut is illustrated in Figure 3.

Encoder. The visual encoder converts the input document image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ into a set of embeddings $\{\mathbf{z}_i | \mathbf{z}_i \in \mathbb{R}^d, 1 \leq i \leq n\}$, where n is feature map size or the number of image patches and d is the dimension of the latent vectors of the encoder. Note that CNN-based models [17] or Transformer-based models [9,40] can be used as the encoder network. In this study, we use Swin Transformer [40] because it shows the best performance in our preliminary study in document parsing. Swin Transformer first splits the input image $\mathbf{x}$ into non-overlapping patches. Swin Transformer blocks, consist of a shifted window-based multi-head self-attention module and a two-layer MLP, are applied to the patches. Then, patch merging layers are applied to the patch tokens at each stage. The output of the final Swin Transformer block $\{\mathbf{z}\}$ is fed into the following textual decoder.

Decoder. Given the $\{\mathbf{z}\}$, the textual decoder generates a token sequence $(\mathbf{y}_i)_{i=1}^{m}$, where $\mathbf{y}_i \in \mathbb{R}^v$ is an one-hot vector for the i-th token, v is the size of token vocabulary, and m is a hyperparameter, respectively. We use BART [33] as the decoder architecture. Specifically, we initialize the decoder model weights with those from the publicly available[2] pre-trained multi-lingual BART model[38].

Model Input. Following the original Transformer [58], we use a teacher-forcing scheme [62], which is a model training strategy that uses the ground truth as input instead of model output from a previous time step. In the test phase, inspired by GPT-3 [5], the model generates a token sequence given a prompt. We add new special tokens for the prompt for each downstream task in our experiments. The prompts that we use for our applications are shown with the desired output sequences in Figure 3. Illustrative explanations for the teacher-forcing strategy and the decoder output format are available in Appendix A.4.

Output Conversion. The output token sequence is converted to a desired structured format. We adopt a JSON format due to its high representation capacity. As shown in Figure 3, a token sequence is one-to-one invertible to a JSON data. We simply add two special tokens `[START_*]` and `[END_*]`, where $*$ indicates each field to extract. If the output token sequence is wrongly structured, we simply treat the field is lost. For example, if there is only `[START_name]` exists but no `[END_name]`, we assume the model fails to extract "name" field. This algorithm can easily be implemented with simple regular expressions [11].

2.3 Pre-training

Task. The model is trained to read all texts in the image in reading order (from top-left to bottom-right, basically). The objective is to minimize cross-entropy loss of next token prediction by jointly conditioning on the image and previous contexts. This task can be interpreted as a pseudo-OCR task. The model is trained as a visual language model over the visual corpora, i.e., document images.

Visual Corpora. We use IIT-CDIP [32], which is a set of 11M scanned english document images. A commercial CLOVA OCR API is applied to get the pseudo text labels. As aforementioned, however, this kind of dataset is not always available, especially for languages other than English. To alleviate the dependencies, we build a scalable ***Synth****etic* ***D****ocument* ***G****enerator*, referred to as **SynthDoG**. Using the SynthDog and Chinese, Japanese, Korean and English Wikipedia, we generated 0.5M samples per language.

Synthetic Document Generator. The pipeline of image rendering basically follows Yim et al. [67]. As shown in Figure 4, the generated sample consists of

[2] https://huggingface.co/hyunwoongko/asian-bart-ecjk.

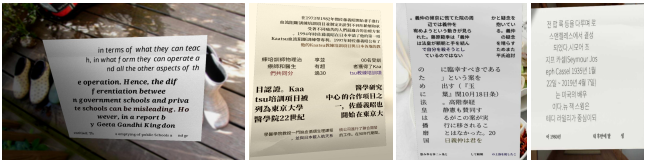


Fig. 4. Generated English, Chinese, Japanese, and Korean samples with SynthDoG. Heuristic random patterns are applied to mimic the real documents

several components; background, document, text, and layout. Background image is sampled from ImageNet [7], and a texture of document is sampled from the collected paper photos. Words and phrases are sampled from Wikipedia. Layout is generated by a simple rule-based algorithm that randomly stacks grids. In addition, several image rendering techniques [13,41,67] are applied to mimic real documents. The generated examples are shown in Figure 4. More details of SynthDoG are available in the code[1] and Appendix A.2.

2.4 Fine-tuning

After the model learns *how to read*, in the application stage (i.e., fine-tuning), we teach the model *how to understand* the document image. As shown in Figure 3, we interpret all downstream tasks as a JSON prediction problem.

The decoder is trained to generate a token sequence that can be converted into a JSON that represents the desired output information. For example, in the document classification task, the decoder is trained to generate a token sequence `[START_class] [memo] [END_class]` which is 1-to-1 invertible to a JSON {"class": "memo"}. We introduce some special tokens (e.g., `[memo]` is used for representing the class "memo"), if such replacement is available in the target task.

3 Experiments and Analyses

In this section, we present **Donut** fine-tuning results on three VDU applications on six different datasets including both public benchmarks and private industrial service datasets. The samples are shown in Figure 5.

3.1 Downstream Tasks and Datasets

Document Classification. To see whether the model can distinguish across different types of documents, we test a classification task. Unlike other models that predict the class label via a softmax on the encoded embedding, Donut generate a JSON that contains class information to maintain the uniformity of the task-solving method. We report overall classification accuracy on a test set.

RVL-CDIP. The RVL-CDIP dataset [16] consists of 400K images in 16 classes, with 25K images per class. The classes include letter, memo, email, and so on. There are 320K training, 40K validation, and 40K test images.

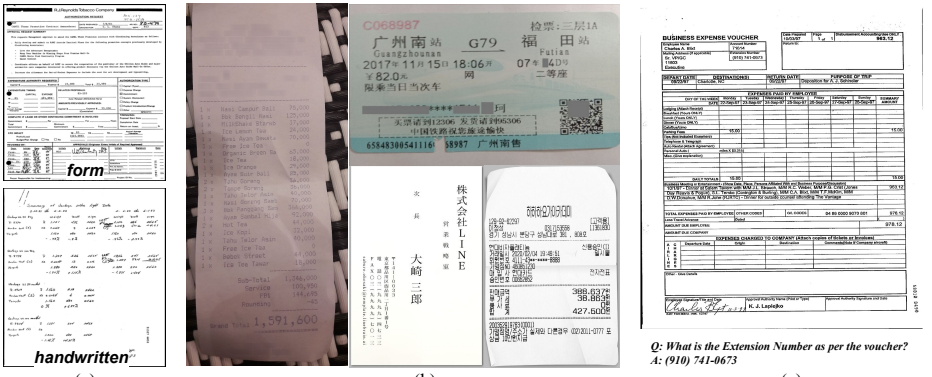


Fig. 5. Samples of the downstream datasets. (a) Document Classification. (b) Document Information Extraction. (c) Document Visual Question Answering

Document Information Extraction. To see the model fully understands the complex layouts and contexts in documents, we test document information extraction (IE) tasks on various real document images including both public benchmarks and real industrial datasets. In this task, the model aims to map each document to a structured form of information that is consistent with the target ontology or database schema. See Figure 1 for an illustrative example. The model should not only read the characters well, but also understand the layouts and semantics to infer the groups and nested hierarchies among the texts.

We evaluate the models with two metrics; field-level F1 score [22,65,18] and Tree Edit Distance (TED) based accuracy [68,70,23]. The F1 checks whether the extracted field information is in the ground truth. Even if a single character is missed, the score assumes the field extraction is failed. Although F1 is simple and easy to understand, there are some limitations. First, it does not take into account partial overlaps. Second, it can not measure the predicted structure (e.g., groups and nested hierarchy). To assess overall accuracy, we also use another metric based on TED [68], that can be used for any documents represented as trees. It is calculated as, $\max(0, 1-\text{TED}(\text{pr}, \text{gt})/\text{TED}(\phi, \text{gt}))$, where gt, pr, and ϕ stands for ground truth, predicted, and empty trees respectively. Similar metrics are used in recent works on document IE [70,23]

We use two public benchmark datasets as well as two private industrial datasets which are from our active real-world service products. Each dataset is explained in the followings.

CORD. The Consolidated Receipt Dataset (CORD)[3] [45] is a public benchmark that consists of 0.8K train, 0.1K valid, 0.1K test receipt images. The letters of receipts is in Latin alphabet. The number of unique fields is 30 containing menu name, count, total price, and so on. There are complex structures (i.e.,

[3] https://huggingface.co/datasets/naver-clova-ix/cord-v1.

nested groups and hierarchies such as `items>item>{name, count, price}`) in the information. See Figure 1 for more details.

Ticket. This is a public benchmark dataset [12] that consists of 1.5K train and 0.4K test Chinese train ticket images. We split 10% of the train set as a validation set. There are 8 fields which are ticket number, starting station, train number, and so on. The structure of information is simple and all keys are guaranteed to appear only once and the location of each field is fixed.

Business Card (In-Service Data). This dataset is from our active products that are currently deployed. The dataset consists of 20K train, 0.3K valid, 0.3K test Japanese business cards. The number of fields is 11, including name, company, address, and so on. The structure of information is similar to the *Ticket* dataset.

Receipt (In-Service Data). This dataset is also from one of our real products. The dataset consists of 40K train, 1K valid, 1K test Korean receipt images. The number of unique field is 81, which includes store information, payment information, price information, and so on. Each sample has complex structures compared to the aforementioned datasets. Due to industrial policies, not all samples can publicly be available. Some real-like high-quality samples are shown in Figure 5 and in the supplementary material.

Document Visual Question Answering. To validate the further capacity of the model, we conduct a document visual question answering task (DocVQA). In this task, a document image and question pair is given and the model predicts the answer for the question by capturing both visual and textual information within the image. We make the decoder generate the answer by setting the question as a starting prompt to keep the uniformity of the method (See Figure 3).

DocVQA. The dataset is from Document Visual Question Answering competition[4] and consists of 50K questions defined on more than 12K documents [44]. There are 40K train, 5K valid, and 5K test questions. The evaluation metric is ANLS (Average Normalized Levenshtein Similarity) which is an edit-distance-based metric. The score on the test set is measured via the evaluation site.

3.2 Setups

We use Swin-B [40] as a visual encoder of **Donut** with slight modification. We set the layer numbers and window size as $\{2, 2, 14, 2\}$ and 10. In further consideration of the speed-accuracy trade-off, we use the first four layers of BART as a decoder. As explained in Section 2.3, we train the multi-lingual **Donut** using the 2M synthetic and 11M IIT-CDIP scanned document images. We pre-train the model for 200K steps with 64 A100 GPUs and a mini-batch size of 196. We use Adam [30] optimizer, the learning rate is scheduled and the initial rate is selected

[4] https://rrc.cvc.uab.es/?ch=17.

Table 1. Classification results on the RVL-CDIP dataset. Donut achieves state-of-the-are performance with reasonable speed and model size efficiency. **Donut** is a general purpose backbone but does not rely on OCR while other recent backbones (e.g., LayoutLM) do. †# parameters for OCR should be considered for non-E2E models

	OCR	#Params	Time (ms)	Accuracy (%)
BERT	✓	110M + $\alpha^{\dagger}$	1392	89.81
RoBERTa	✓	125M + $\alpha^{\dagger}$	1392	90.06
LayoutLM	✓	113M + $\alpha^{\dagger}$	1396	91.78
LayoutLM (w/ image)	✓	160M + $\alpha^{\dagger}$	1426	94.42
LayoutLMv2	✓	200M + $\alpha^{\dagger}$	1489	95.25
Donut (Proposed)		143M	**752**	**95.30**

from 1e-5 to 1e-4. The input resolution is set to 2560×1920 and a max length in the decoder is set to 1536. All fine-tuning results are achieved by starting from the pre-trained multi-lingual model. Some hyperparameters are adjusted at fine-tuning and in ablation studies. We use 960×1280 for Train Tickets and Business Card parsing tasks. We fine-tune the model while monitoring the edit distance over token sequences. The speed of Donut is measured on a P40 GPU, which is much slower than A100. For the OCR based baselines, states-of-the-art OCR engines are used, including MS OCR API used in [64] and CLOVA OCR API[5] used in [24,23]. An analysis on OCR engines is available in Section 3.4. More details of OCR and training setups are available in Appendix A.1 and A.5.

3.3 Experimental Results

Document Classification. The results are shown in Table 1. Without relying on any other resource (e.g., off-the-shelf OCR engine), Donut shows a state-of-the-art performance among the general-purpose VDU models such as LayoutLM [65] and LayoutLMv2 [64]. In particular, Donut surpasses the LayoutLMv2 accuracy reported in [64], while using fewer parameters with the 2x faster speed. Note that the OCR-based models must consider additional model parameters and speed for the entire OCR framework, which is not small in general. For example, a recent advanced OCR-based model [4,3] requires more than 80M parameters. Also, training and maintaining the OCR-based systems are costly [23], leading to needs for the Donut-like end-to-end approach.

Document Information Extraction. Table 2 shows the results on the four different document IE tasks. The first group uses a conventional BIO-tagging-based IE approach [22]. We follows the conventions in IE [65,18]. OCR extracts texts and bounding boxes from the image, and then the serialization module sorts all texts with geometry information within the bounding box. The BIO-tagging-based named entity recognition task performs token-level tag classification upon

[5] https://clova.ai/ocr.

Table 2. Performances on various document IE tasks. The field-level F1 scores and tree-edit-distance-based accuracies are reported. **Donut** shows the best accuracies for all domains with significantly faster inference speed. †Parameters for vocabulary are omitted for fair comparisons among multi-lingual models. ‡# parameters for OCR should be considered. *Official multi-lingual extension models are used

			CORD [45]			Ticket [12]			Business Card			Receipt		
	OCR	#Params	Time (s)	F1	Acc.	Time (s)	F1	Acc.	Time (s)	F1	Acc.	Time (s)	F1	Acc.
BERT* [22]	✓	$86^{\dagger}_{\mathrm{M}} + \alpha^{\ddagger}$	1.6	73.0	65.5	1.7	74.3	82.4	1.5	40.8	72.1	2.5	70.3	54.1
BROS [18]	✓	$86^{\dagger}_{\mathrm{M}} + \alpha^{\ddagger}$	1.7	74.7	70.0									
LayoutLM [65]	✓	$89^{\dagger}_{\mathrm{M}} + \alpha^{\ddagger}$	1.7	78.4	81.3									
LayoutLMv2* [64,66]	✓	$179^{\dagger}_{\mathrm{M}} + \alpha^{\ddagger}$	1.7	78.9	82.4	1.8	87.2	90.1	1.6	52.2	83.0	2.6	72.9	78.0
Donut		$143^{\dagger}_{\mathrm{M}}$	**1.2**	**84.1**	**90.9**	**0.6**	**94.1**	**98.7**	**1.4**	**57.8**	**84.4**	**1.9**	**78.6**	**88.6**
SPADE* [25]	✓	$93^{\dagger}_{\mathrm{M}} + \alpha^{\ddagger}$	4.0	74.0	75.8	4.5	14.9	29.4	4.3	32.3	51.3	7.3	64.1	53.2
WYVERN* [21]	✓	$106^{\dagger}_{\mathrm{M}} + \alpha^{\ddagger}$	1.2	43.3	46.9	1.5	41.8	54.8	1.7	29.9	51.5	3.4	71.5	82.9

the ordered texts to generate a structured form. We test three general-purpose VDU backbones, BERT [8], BROS [18], LayoutLM [65], and LayoutLMv2 [64,66].

We also test two recently proposed IE models, SPADE [24] and WYVERN [23]. SPADE is a graph-based IE method that predicts relations between bounding boxes. WYVERN is an Transformer encoder-decoder model that directly generates entities with structure given OCR outputs. WYVERN is different from Donut in that it takes the OCR output as its inputs.

For all domains, including public and private in-service datasets, Donut shows the best scores among the comparing models. By measuring both F1 and TED-based accuracy, we observe not only Donut can extract key information but also predict complex structures among the field information. We observe that a large input resolution gives robust accuracies but makes the model slower. For example, the performance on the CORD with 1280×960 was 0.7 sec./image and 91.1 accuracy. But, the large resolution showed better performances on the low-resource situation. The detailed analyses are in Section 3.4. Unlike other baselines, Donut shows stable performance regardless of the size of datasets and complexity of the tasks (See Figure 5). This is a significant impact as the target tasks are already actively used in industries.

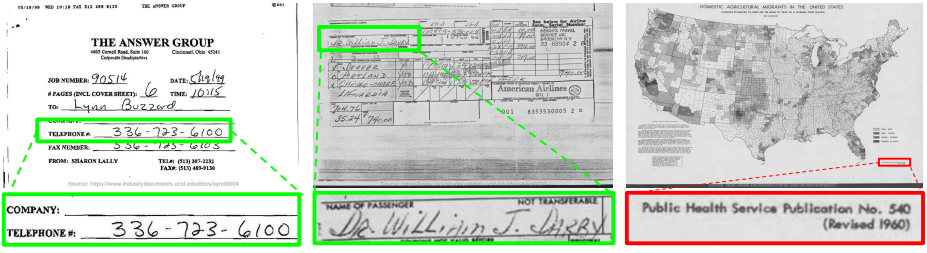
Document Visual Question Answering. Table 3 shows the results on the DocVQA dataset. The first group is the general-purposed VDU backbones whose scores are from the LayoutLMv2 paper [64]. We measure the running time with MS OCR API used in [64]. The model in the third group is a DocVQA-specific-purposed fine-tuning model of LayoutLMv2, whose inference results are available in the official leader-board.[6]

As can be seen, **Donut** achieves competitive scores with the baselines that are dependent on external OCR engines. Especially, Donut shows that it is robust to the handwritten documents, which is known to be challenging to process. In the conventional approach, adding a post-processing module that corrects OCR

[6] https://rrc.cvc.uab.es/?ch=17&com=evaluation&task=1.

Table 3. Average Normalized Levenshtein Similarity (ANLS) scores on DocVQA. Donut shows a promising result without OCR. *__Donut__ shows a high ANLS score on the handwritten documents which are known to be challenging due to the difficulty of handwriting OCR (See Figure 6). †Token embeddings for English is counted for a fair comparison. ‡# parameters for OCR should be considered

	Fine-tuning set	OCR	#Params†	Time (ms)	ANLS test set	ANLS* handwritten
BERT [64]	train set	✓	110M + α‡	1517	63.5	n/a
LayoutLM[65]	train set	✓	113M + α‡	1519	69.8	n/a
LayoutLMv2[64]	train set	✓	200M + α‡	1610	78.1	n/a
Donut	train set		176M	**782**	67.5	**72.1**
LayoutLMv2-Large-QG[64]	train + dev + QG	✓	390M + α‡	1698	**86.7**	67.3



Q: What is the phone number given?
Answer: 336-723-6100
Donut: 336-723-6100
LayoutLMv2-Large-QG: 336-723- 4100

Q: What is the name of the passenger?
Answer: DR. William J. Darby
Donut: DR. William J. Darby
LayoutLMv2-Large-QG: DR. William J. Jarry

Q: What is the Publication No.?
Answer: 540
Donut: 943 (another number in the image is extracted)
LayoutLMv2-Large-QG: 540

Fig. 6. Examples of Donut and LayoutLMv2 outputs on DocVQA. The OCR-errors make a performance upper-bound of the OCR-dependent baselines, e.g., LayoutLMv2 (left and middle examples). Due to the input resolution constraint of the end-to-end pipeline, Donut miss some tiny texts in large-scale images (right example) but this could be mitigated by scaling the input image size (See Section 3.4)

errors is an option to strengthen the pipeline [51,50,10] or adopting an encoder-decoder architecture on the OCR outputs can mitigate the problems of OCR errors [23]. However, this kind of approaches tend to increase the entire system size and maintenance cost. Donut shows a completely different direction. Some inference results are shown in Figure 6. The samples show the current strengths of Donut as well as the left challenges in the Donut-like end-to-end approach. Further analysis and ablation is available in Section 3.4.

3.4 Further Studies

In this section, we study several elements of understanding Donut. We show some striking characteristics of Donut through the experiments and visualization.

On Pre-training Strategy. We test several pre-training tasks for VDUs. Figure 7(a) shows that the Donut pre-training task (i.e., text reading) is the most

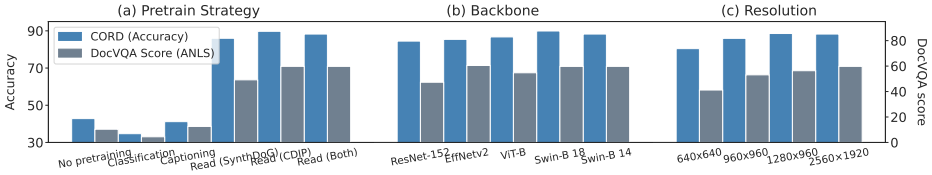


Fig. 7. Analysis on (a) pre-training strategies, (b) image backbones, and (c) input resolutions. Performances on CORD [45] and DocVQA [44] are shown

simple yet effective approach. Other tasks that impose a general knowledge of images and texts on models, e.g., image captioning, show little gains in the fine-tuning tasks. For the text reading tasks, we verify three options, SynthDoG only, IIT-CDIP only, and both. Note that synthetic images were enough for the document IE task in our analysis. However, in the DocVQA task, it was important to see the real images. This is probably because the image distributions of IIT-CDIP and DocVQA are similar [44].

On Encoder Backbone. Here, we study popular image classification backbones that show superior performance in traditional vision tasks to measure their performance in VDU tasks. The Figure 7(b) shows the comparison results. We use all the backbones pre-trained on ImageNet [7]. EfficientNetV2 [55] and Swin Transformer [40] outperform others on both datasets. We argue that this is due to the high expressiveness of the backbones, which were shown by the striking scores on several downstream tasks as well. We choose Swin Transformer due to the high scalability of the Transformer-based architecture and higher performance over the EfficientNetV2's.

On Input Resolution. The Figure 7(c) shows the performance of Donut grows rapidly as we set a larger input size. This gets clearer in the DocVQA where the images are larger with many tiny texts. But, increasing the size for a precise result incurs bigger computational costs. Using an efficient attention mechanism [60] may avoid the matter in architectural design, but we use the original Transformer [58] as we aim to present a simpler architecture in this work.

On Text Localization. To see how the model behaves, we visualize the corss attention maps of the decoder given an unseen document image. As can be seen in Figure 8, the model shows meaningful results that can be used as an auxiliary indicator. The model attends to a desired location in the given image.

On OCR System. We test four widely-used public OCR engines (See Figure 9). The results show that the performances (i.e., speed and accuracy) of the conventional OCR-based methods heavily rely on the off-the-shelf OCR engine. More details of the OCR engines are available in Appendix A.1.

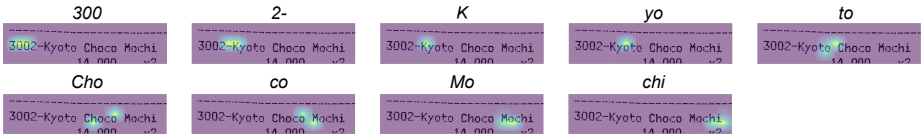


Fig. 8. Visualization of cross-attention maps in the decoder and its application to text localization. Donut is trained without any supervision for the localization. The Donut decoder attends proper text regions to process the image

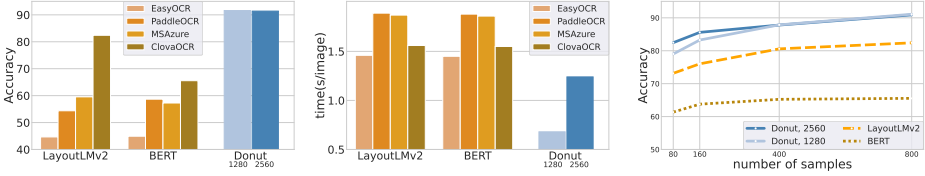


Fig. 9. Comparison of BERT, LayoutLMv2 and Donut on CORD. The performances (i.e., speed and accuracy) of the OCR-based models extremely varies depending on what OCR engine is used (left and center). Donut shows robust performances even in a low resourced situation showing the higher score only with 80 samples (right)

On Low Resourced Situation. We evaluate the models by limiting the size of training set of CORD [45]. The performance curves are shown in the right Figure 9. Donut shows a robust performances. We also observe that a larger input resolution, 2560×1920, shows more robust scores on the extremely low-resourced situation, e.g., 80 samples. As can be seen, Donut outperformed the LayoutLMv2 accuracy only with 10% of the data, which is only 80 samples.

4 Related Work

4.1 Optical Character Recognition

Recent trends of OCR study are to utilize deep learning models in its two substeps: 1) text areas are predicted by a detector; 2) a text recognizer then recognizes all characters in the cropped image instances. Both are trained with a large-scale datasets including the synthetic images [26,13] and real images [28,47].

Early detection methods used CNNs to predict local segments and apply heuristics to merge them [19,69]. Later, region proposal and bounding box regression based methods were proposed [36]. Recently, focusing on the homogeneity and locality of texts, component-level approaches were proposed [56,4].

Many modern text recognizer share a similar approach [37,53,52,59] that can be interpreted into a combination of several common deep modules [3]. Given the cropped text instance image, most recent text recognition models apply CNNs to encode the image into a feature space. A decoder is then applied to extract characters from the features.

4.2 Visual Document Understanding

Classification of the document type is a core step towards automated document processing. Early methods treated the problem as a general image classification, so various CNNs were tested [27,1,15]. Recently, with BERT [8], the methods based on a combination of CV and NLP were widely proposed [65,34]. As a common approach, most methods rely on an OCR engine to extract texts; then the OCR-ed texts are serialized into a token sequence; finally they are fed into a language model (e.g., BERT) with some visual features if available. Although the idea is simple, the methods showed remarkable performance improvements and became a main trend in recent years [64,35,2].

Document IE covers a wide range of real applications [22,42], for example, given a bunch of raw receipt images, a document parser can automate a major part of receipt digitization, which has been required numerous human-labors in the traditional pipeline. Most recent models [25,23] take the output of OCR as their input. The OCR results are then converted to the final parse through several processes, which are often complex. Despite the needs in the industry, only a few works have been attempted on end-to-end parsing. Recently, some works are proposed to simplify the complex parsing processes [25,23]. But they still rely on a separate OCR to extract text information.

Visual QA on documents seeks to answer questions asked on document images. This task requires reasoning over visual elements of the image and general knowledge to infer the correct answer [44]. Currently, most state-of-the-arts follow a simple pipeline consisting of applying OCR followed by BERT-like transformers [65,64]. However, the methods work in an extractive manner by their nature. Hence, there are some concerns for the question whose answer does not appear in the given image [57]. To tackle the concerns, generation-based methods have also been proposed [48].

5 Conclusions

In this work, we propose a novel end-to-end framework for visual document understanding. The proposed method, **Donut**, directly maps an input document image into a desired structured output. Unlike conventional methods, **Donut** does not depend on OCR and can easily be trained in an end-to-end fashion. We also propose a synthetic document image generator, SynthDoG, to alleviate the dependency on large-scale real document images and we show that **Donut** can be easily extended to a multi-lingual setting. We gradually trained the model from *how to read* to *how to understand* through the proposed training pipeline. Our extensive experiments and analysis on both external public benchmarks and private internal service datasets show higher performance and better *cost-effectiveness* of the proposed method. This is a significant impact as the target tasks are already practically used in industries. Enhancing the pre-training objective could be a future work direction. We believe our work can easily be extended to other domains/tasks regarding document understanding.

References

1. Afzal, M.Z., Capobianco, S., Malik, M.I., Marinai, S., Breuel, T.M., Dengel, A., Liwicki, M.: Deepdocclassifier: Document classification with deep convolutional neural network. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 1111–1115 (2015). https://doi.org/10.1109/ICDAR.2015.7333933 1, 4, 14
2. 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 (ICCV). pp. 993–1003 (October 2021) 14
3. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019) 2, 4, 9, 13, 22
4. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9357–9366 (2019). https://doi.org/10.1109/CVPR.2019.00959 2, 4, 9, 13, 22
5. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf 5
6. Davis, B., Morse, B., Cohen, S., Price, B., Tensmeyer, C.: Deep visual template-free form parsing. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 134–141 (2019). https://doi.org/10.1109/ICDAR.2019.00030 3
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 6, 12, 23
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423 3, 4, 10, 14, 27, 29
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=YicbFdNTTy 3, 4
10. Duong, Q., Hämäläinen, M., Hengchen, S.: An unsupervised method for OCR post-correction and spelling normalisation for Finnish. In: Proceedings of the

23rd Nordic Conference on Computational Linguistics (NoDaLiDa). pp. 240–248. Linköping University Electronic Press, Sweden, Reykjavik, Iceland (Online) (May 31–2 Jun 2021), https://aclanthology.org/2021.nodalida-main.24 3, 11
11. Friedl, J.E.F.: Mastering Regular Expressions. O'Reilly, Beijing, 3 edn. (2006), https://www.safaribooksonline.com/library/view/mastering-regular-expressions/0596528124/ 5
12. Guo, H., Qin, X., Liu, J., Han, J., Liu, J., Ding, E.: Eaten: Entity-aware attention for single shot visual text extraction. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 254–259 (2019). https://doi.org/10.1109/ICDAR.2019.00049 4, 8, 10, 22, 24, 25, 26
13. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016) 6, 13
14. Hammami, M., Héroux, P., Adam, S., d'Andecy, V.P.: One-shot field spotting on colored forms using subgraph isomorphism. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 586–590 (2015). https://doi.org/10.1109/ICDAR.2015.7333829 3
15. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 991–995 (2015). https://doi.org/10.1109/ICDAR.2015.7333910 4, 14
16. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 991–995 (2015). https://doi.org/10.1109/ICDAR.2015.7333910 6
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90 4
18. Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., Park, S.: Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. Proceedings of the AAAI Conference on Artificial Intelligence **36**(10), 10767–10775 (Jun 2022). https://doi.org/10.1609/aaai.v36i10.21322, https://ojs.aaai.org/index.php/AAAI/article/view/21322 2, 3, 4, 7, 9, 10, 22, 27
19. Huang, W., Qiao, Y., Tang, X.: Robust scene text detection with convolution neural network induced mser trees. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 497–511. Springer International Publishing, Cham (2014) 13
20. Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., Jawahar, C.V.: Icdar2019 competition on scanned receipt ocr and information extraction. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1516–1520 (2019). https://doi.org/10.1109/ICDAR.2019.00244 3
21. Hwang, A., Frey, W.R., McKeown, K.: Towards augmenting lexical resources for slang and African American English. In: Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects. pp. 160–172. International Committee on Computational Linguistics (ICCL), Barcelona, Spain (Online) (Dec 2020), https://aclanthology.org/2020.vardial-1.15 10
22. Hwang, W., Kim, S., Yim, J., Seo, M., Park, S., Park, S., Lee, J., Lee, B., Lee, H.: Post-ocr parsing: building simple and robust parser via bio tagging. In: Workshop on Document Intelligence at NeurIPS 2019 (2019) 1, 2, 4, 7, 9, 10, 14, 28, 29

23. Hwang, W., Lee, H., Yim, J., Kim, G., Seo, M.: Cost-effective end-to-end information extraction for semi-structured document images. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 3375–3383. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.271, https://aclanthology.org/2021.emnlp-main.271 2, 4, 7, 9, 10, 11, 14, 27
24. Hwang, W., Yim, J., Park, S., Yang, S., Seo, M.: Spatial dependency parsing for semi-structured document information extraction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 330–343. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.findings-acl.28, https://aclanthology.org/2021.findings-acl.28 2, 4, 9, 10
25. Hwang, W., Yim, J., Park, S., Yang, S., Seo, M.: Spatial dependency parsing for semi-structured document information extraction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 330–343. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.findings-acl.28, https://aclanthology.org/2021.findings-acl.28 3, 10, 14, 27
26. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. In: Workshop on Deep Learning, NIPS (2014) 13
27. Kang, L., Kumar, J., Ye, P., Li, Y., Doermann, D.S.: Convolutional neural networks for document image classification. 2014 22nd International Conference on Pattern Recognition pp. 3168–3172 (2014) 1, 4, 14
28. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., Shafait, F., Uchida, S., Valveny, E.: Icdar 2015 competition on robust reading. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 1156–1160 (2015). https://doi.org/10.1109/ICDAR.2015.7333942 13
29. Kim, W., Son, B., Kim, I.: Vilt: Vision-and-language transformer without convolution or region supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 5583–5594. PMLR (18–24 Jul 2021), http://proceedings.mlr.press/v139/kim21k.html 3
30. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980 8, 25
31. Klaiman, S., Lehne, M.: Docreader: Bounding-box free training of a document information extraction model. In: Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I. p. 451–465. Springer-Verlag, Berlin, Heidelberg (2021). https://doi.org/10.1007/978-3-030-86549-8_29, https://doi.org/10.1007/978-3-030-86549-8_29 4
32. Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., Heard, J.: Building a test collection for complex document information processing. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 665–666. SIGIR '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1148170.1148307, https://doi.org/10.1145/1148170.1148307 4, 5

33. 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. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.703, https://aclanthology.org/2020.acl-main.703 5
34. Li, C., Bi, B., Yan, M., Wang, W., Huang, S., Huang, F., Si, L.: StructuralLM: Structural pre-training for form understanding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 6309–6318. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.493, https://aclanthology.org/2021.acl-long.493 14
35. Li, P., Gu, J., Kuen, J., Morariu, V.I., Zhao, H., Jain, R., Manjunatha, V., Liu, H.: Selfdoc: Self-supervised document representation learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5648–5656 (2021). https://doi.org/10.1109/CVPR46437.2021.00560 14
36. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: Textboxes: A fast text detector with a single deep neural network. Proceedings of the AAAI Conference on Artificial Intelligence **31**(1) (Feb 2017). https://doi.org/10.1609/aaai.v31i1.11196, https://ojs.aaai.org/index.php/AAAI/article/view/11196 13
37. Liu, W., Chen, C., Wong, K.Y.K., Su, Z., Han, J.: Star-net: A spatial attention residue network for scene text recognition. In: Richard C. Wilson, E.R.H., Smith, W.A.P. (eds.) Proceedings of the British Machine Vision Conference (BMVC). pp. 43.1–43.13. BMVA Press (September 2016). https://doi.org/10.5244/C.30.43, https://dx.doi.org/10.5244/C.30.43 13
38. Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., Zettlemoyer, L.: Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics **8**, 726–742 (2020), https://aclanthology.org/2020.tacl-1.47 5
39. Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) 2
40. 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 (ICCV). pp. 10012–10022 (October 2021) 4, 8, 12
41. Long, S., Yao, C.: Unrealtext: Synthesizing realistic scene text images from the unreal world. arXiv preprint arXiv:2003.10608 (2020) 6
42. Majumder, B.P., Potti, N., Tata, S., Wendt, J.B., Zhao, Q., Najork, M.: Representation learning for information extraction from form-like documents. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 6495–6504. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.580, https://www.aclweb.org/anthology/2020.acl-main.580 1, 14
43. Majumder, B.P., Potti, N., Tata, S., Wendt, J.B., Zhao, Q., Najork, M.: Representation learning for information extraction from form-like documents. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 6495–6504. Association for Computational Linguistics, Online (Jul

2020). https://doi.org/10.18653/v1/2020.acl-main.580, https://aclanthology.org/2020.acl-main.580 3
44. Mathew, M., Karatzas, D., Jawahar, C.: Docvqa: A dataset for vqa on document images. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2200–2209 (2021) 1, 8, 12, 14
45. Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., Lee, H.: Cord: A consolidated receipt dataset for post-ocr parsing. In: Workshop on Document Intelligence at NeurIPS 2019 (2019) 2, 7, 10, 12, 13, 22, 24, 26
46. Peng, D., Wang, X., Liu, Y., Zhang, J., Huang, M., Lai, S., Zhu, S., Li, J., Lin, D., Shen, C., Jin, L.: SPTS: Single-Point Text Spotting. CoRR **abs/2112.07917** (2021), https://arxiv.org/abs/2112.07917 2
47. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (December 2013) 13
48. Powalski, R., Borchmann, Ł., Jurkiewicz, D., Dwojak, T., Pietruszka, M., Pałka, G.: Going full-tilt boogie on document understanding with text-image-layout transformer. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 732–747. Springer International Publishing, Cham (2021) 14
49. Riba, P., Dutta, A., Goldmann, L., Fornés, A., Ramos, O., Lladós, J.: Table detection in invoice documents by graph neural networks. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 122–127 (2019). https://doi.org/10.1109/ICDAR.2019.00028 3
50. Rijhwani, S., Anastasopoulos, A., Neubig, G.: OCR Post Correction for Endangered Language Texts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 5931–5942. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.478, https://aclanthology.org/2020.emnlp-main.478 3, 11
51. Schaefer, R., Neudecker, C.: A two-step approach for automatic OCR post-correction. In: Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature. pp. 52–57. International Committee on Computational Linguistics, Online (Dec 2020), https://aclanthology.org/2020.latechclfl-1.6 3, 11
52. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 2298–2304 (2017) 13
53. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4168–4176 (2016). https://doi.org/10.1109/CVPR.2016.452 13
54. Taghva, K., Beckley, R., Coombs, J.: The effects of ocr error on the extraction of private information. In: Bunke, H., Spitz, A.L. (eds.) Document Analysis Systems VII. pp. 348–357. Springer Berlin Heidelberg, Berlin, Heidelberg (2006) 2
55. Tan, M., Le, Q.: Efficientnetv2: Smaller models and faster training. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 10096–10106. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/tan21a.html 12
56. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: Leibe, B., Matas, J., Sebe, N., Welling,

M. (eds.) Computer Vision – ECCV 2016. pp. 56–72. Springer International Publishing, Cham (2016) 13
57. Tito, R., Mathew, M., Jawahar, C.V., Valveny, E., Karatzas, D.: Icdar 2021 competition on document visual question answering. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 635–649. Springer International Publishing, Cham (2021) 1, 14
58. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf 4, 5, 12, 24, 27
59. Wang, J., Hu, X.: Gated recurrent convolution neural network for ocr. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/c24cd76e1ce41366a4bbe8a49b02a028-Paper.pdf 13
60. Wang, S., Li, B., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768 (2020) 12, 27
61. Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019). https://doi.org/10.5281/zenodo.4414861 25
62. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural computation **1**(2), 270–280 (1989) 5
63. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020). https://doi.org/10.18653/v1/2020.emnlp-demos.6, https://aclanthology.org/2020.emnlp-demos.6 25
64. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., Zhou, L.: LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 2579–2591. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.201, https://aclanthology.org/2021.acl-long.201 2, 4, 9, 10, 11, 14, 22, 27, 28
65. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 1192–1200. KDD '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3394486.3403172, https://doi.org/10.1145/3394486.3403172 2, 3, 7, 9, 10, 11, 14, 22, 28
66. Xu, Y., Lv, T., Cui, L., Wang, G., Lu, Y., Florencio, D., Zhang, C., Wei, F.: Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. arXiv preprint arXiv:2104.08836 (2021) 10, 27
67. Yim, M., Kim, Y., Cho, H.C., Park, S.: Synthtiger: Synthetic text image generator towards better text recognition models. In: Lladós, J., Lopresti, D., Uchida, S.

(eds.) Document Analysis and Recognition – ICDAR 2021. pp. 109–124. Springer International Publishing, Cham (2021) 5, 6, 23
68. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**, 1245–1262 (12 1989). https://doi.org/10.1137/0218082 7
69. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multi-oriented text detection with fully convolutional networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4159–4167 (2016). https://doi.org/10.1109/CVPR.2016.451 13
70. Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: Data, model, and evaluation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 564–580. Springer International Publishing, Cham (2020) 7

A Appendix

A.1 Details of OCR Engines (MS, CLOVA, Easy, Paddle)

Current state-of-the-art visual document understanding (VDU) backbones, such as BROS [18], LayoutLM [65] and LayoutLMv2 [64], are dependent on off-the-shelf OCR engines. These backbones take the output of OCR as their (one of) input features. For the OCR-dependent methods, in our experiments, we use state-of-the-art OCR engines that are publicly available, including 2 OCR API products (i.e., MS OCR[7] and CLOVA OCR[8]) and 2 open-source OCR models (i.e., Easy OCR[9] and Paddle OCR[10]). In the main paper, Paddle OCR is used for the Chinese train ticket dataset [12] and CLOVA OCR is used for the rest datasets in the document information extraction (IE) tasks. MS OCR is used to measure the running time of the LayoutLM family in document classification and visual question answering (VQA) tasks, following the previous work of Xu et al. [64]. Each OCR engine is explained in the following.

MS OCR MS OCR[7] is the latest OCR API product from Microsoft and used in several recent VDU methods, e.g., LayoutLMv2 [64]. This engine supports 164 languages for printed text and 9 languages for handwritten text (until 2022/03).

CLOVA OCR CLOVA OCR[8] is an API product from NAVER CLOVA and is specialized in document IE tasks. This engine supports English, Japanese and Korean (until 2022/03). In the ablation experiments on the CORD dataset [45] (Figure 9 in the main paper), the CLOVA OCR achieved the best accuracy.

Easy OCR Easy OCR[9] is a ready-to-use OCR engine that is publicly available at GitHub. This engine supports more than 80 languages (until 2022/03). Unlike the aforementioned two OCR products (i.e., MS OCR and CLOVA OCR), this engine is publicly opened and downloadable.[9] The entire model architecture is based on the modern deep-learning-based OCR modules [4,3] with some modifications to make the model lighter and faster. The total number of model parameters is 27M which is small compared to the state-of-the-art models [4,3].

Paddle OCR Paddle OCR[10] is an open-source OCR engine available at GitHub. We used a lightweight (i.e., mobile) version of the model which is specially designed for a fast and light OCR of English and Chinese texts. The model is served on a CPU environment and the size of the model is extremely small, which is approximately 10M.

[7] https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview-ocr.

[8] https://clova.ai/ocr/en.

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

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

Fig. A. Examples of SynthDoG. English, Chinese, Japanese and Korean samples are shown (from top to bottom). Although the idea is simple, these synthetic samples play an important role in the pre-training of Donut. Please, see Figure 7 in the main paper for details

A.2 Details of Synthetic Document Generator (SynthDoG)

In this section, we explain the components of the proposed Synthetic Document Generator (SynthDoG) in detail. The entire pipeline basically follows Yim et al. [67]. Our source code is available at https://github.com/clovaai/donut. More samples are shown in Figure A.

Background Background images are sampled from ImageNet [7]. Gaussian blur is randomly applied to the background image to represent out-of-focus effects.

Document Paper textures are sampled from the photos that we collected. The texture is applied to an white background. In order to make the texture realistic, random elastic distortion and Gaussian noise are applied. To represent various view angles in photographs, a random perspective transformation is applied to the image.

Text Layout and Pattern To mimic the layouts in real-world documents, a heuristic rule-based pattern generator is applied to the document image region to generate text regions. The main idea is to set multiple squared regions to represent text paragraphs. Each squared text region is then interpreted as multiple lines of text. The size of texts and text region margins are chosen randomly.

Text Content and Style We prepare the multi-lingual text corpora from Wikipedia.[11] We use Noto fonts[12] since it supports various languages. SynthDoG samples texts and fonts from these resources and the sampled texts are rendered in the regions that are generated by the layout pattern generator. The text colors are randomly assigned.

Post-processing Finally, some post-processing techniques are applied to the output image. In this process, the color, brightness, and contrast of the image are adjusted. In addition, shadow effect, motion blur, Gaussian blur, and JPEG compression are applied to the image.

A.3 Details of Document Information Extraction

Information Extraction (IE) on documents is an arduous task since it requires (a) reading texts, (b) understanding the meaning of the texts, and (c) predicting the relations and structures among the extracted information. Some previous works have only focused on extracting several pre-defined key information [12]. In that case, only (a) and (b) are required for IE models. We go beyond the previous works by considering (c) also. Although the task is complex, its interface (i.e., the format of input and output) is simple. In this section, for explanation purposes, we show some sample images (which are the raw input of the IE pipeline) with the output of Donut.

In the main paper, we test four datasets including two public benchmarks (i.e., *CORD* [45] and *Ticket* [12]) and two private industrial datasets (i.e., *Business Card* and *Receipt*). Figure B shows examples of *Ticket* with the outputs of Donut. Figure C shows examples of *CORD* with the outputs of Donut. Due to strict industrial policies on the private industrial datasets, we instead show some real-like high-quality samples of *Business Card* and *Receipt* in Figure D.

A.4 Details of Model Training Scheme and Output Format

In the model architecture and training objective, we basically followed the original Transformer [58], which uses a Transformer encoder-decoder architecture and a teacher-forcing training scheme. The teacher-forcing scheme is a model training strategy that uses the ground truth as input instead of model output from a previous time step. Figure E shows a details of the model training scheme and decoder output format.

[11] https://dumps.wikimedia.org.
[12] https://fonts.google.com/noto.

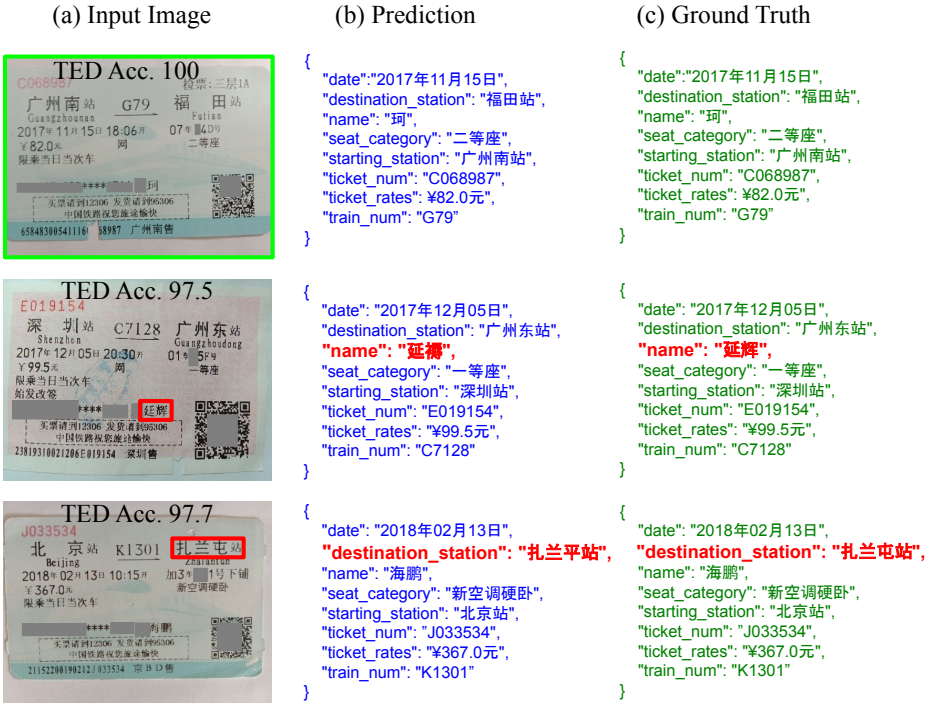


Fig. B. Examples of *Ticket* [12] with Donut predictions. There is no hierarchy in the structure of information (i.e., depth = 1) and the location of each key information is almost fixed. Failed predictions are marked and bolded (red)

A.5 Implementation and Training Hyperparameters

The codebase and settings are available at GitHub.[13] We implement the entire model pipeline with Huggingface's `transformers`[14] [63] and an open-source library `TIMM` (PyTorch image models)[15] [61].

For all model training, we use a half-precision (fp16) training. We train Donut using Adam optimizer [30] by decreasing the learning rate as the training progresses. The initial learning rate of pre-training is set to 1e-4 and that of fine-tuning is selected from 1e-5 to 1e-4. We pre-train the model for 200K steps with 64 NVIDIA A100 GPUs and a mini-batch size of 196, which takes about 2-3 GPU days. We also apply a gradient clipping technique where a maximum gradient norm is selected from 0.05 to 1.0. The input resolution of Donut is set to 2560×1920 at the pre-training phase. In downstream tasks, the input resolutions are controlled. In some downstream document IE experiments, such as,

[13] https://github.com/clovaai/donut.
[14] https://github.com/huggingface/transformers.
[15] https://github.com/rwightman/pytorch-image-models.

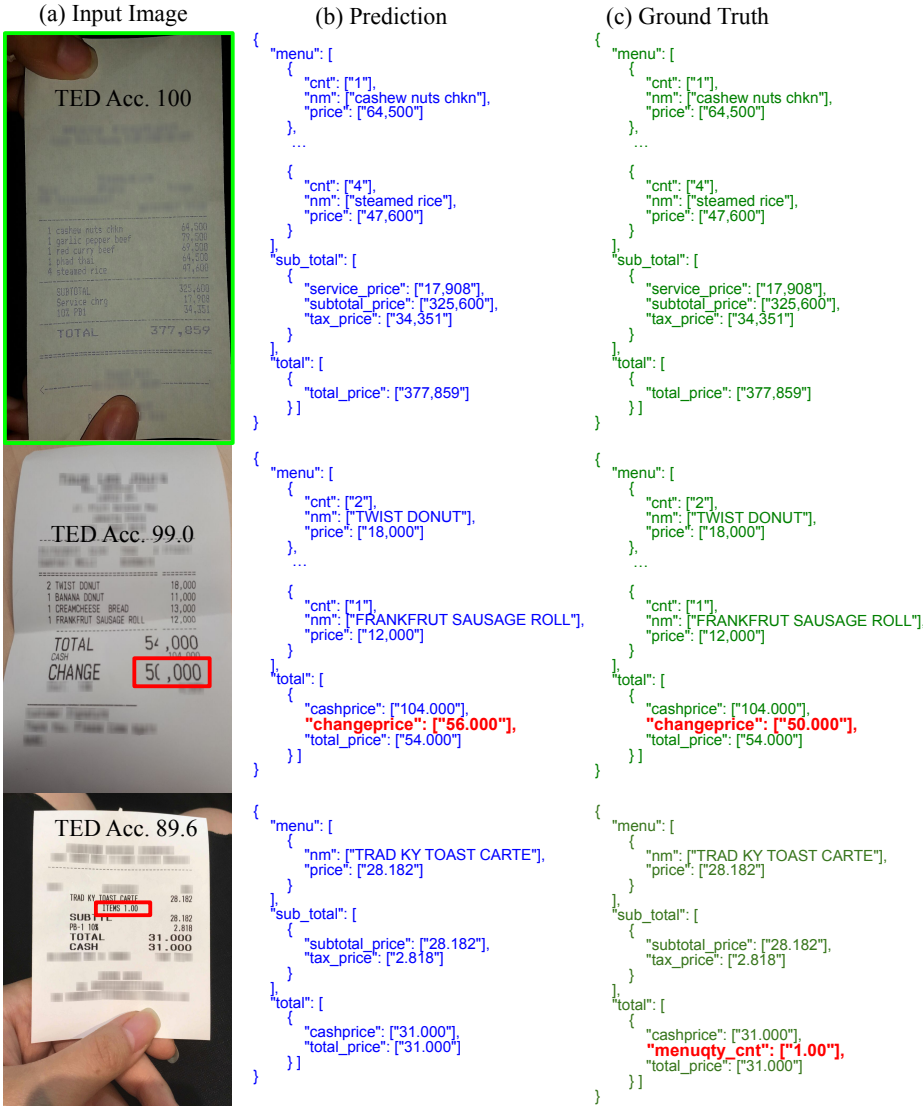


Fig. C. Examples of *CORD* [45] with Donut predictions. There is a hierarchy in the structure of information (i.e., depth = 2). **Donut** not only reads some important key information from the image, but also predicts the relationship among the extracted information (e.g., the name, price, and quantity of each menu item are grouped)

CORD [45], *Ticket* [12] and *Business Card*, smaller size of input resolution, e.g., 1280×960, is tested. With the 1280×960 setting, the model training cost of Donut was small. For example, the model fine-tuning on *CORD* or *Ticket* took approximately 0.5 hours with one A100 GPU. However, when we set the

Fig. D. Examples of *Business Card* (top) and *Receipt* (bottom). Due to strict industrial policies on the private industrial datasets from our active products, real-like high-quality samples are shown instead

2560×1920 setting for larger datasets, e.g., *RVL-CDIP* or *DocVQA*, the cost increased rapidly. With 64 A100 GPUs, *DocVQA* requires one GPU day and *RVL-CDIP* requires two GPU days approximately. This is not surprising in that increasing the input size for a precise result incurs higher computational costs in general. Using an efficient attention mechanism [60] may avoid the problem in architectural design, but we use the original Transformer [58] as we aim to present a simpler architecture in this work. Our preliminary experiments in smaller resources are available in Appendix A.6.

For the implementation of document IE baselines, we use the `transformers` library for BERT [8], BROS [18], LayoutLMv2 [64,66] and WYVERN [23]. For the SPADE [25] baseline, the official implementation[16] is used. The models are trained using NVIDIA P40, V100, or A100 GPUs. The major hyperparameters, such as initial learning rate and number of epochs, are adjusted by monitoring the scores on the validation set. The architectural details of the OCR-dependent VDU backbone baselines (e.g., LayoutLM and LayoutLMv2) are available in Appendix A.7.

A.6 Preliminary Experiments in Smaller Resources

In our preliminary experiments, we pre-trained Donut with smaller resources (denoted as $\text{Donut}_{\text{Proto}}$), i.e., smaller data (SynthDoG 1.2M) and fewer GPUs

[16] https://github.com/clovaai/spade.

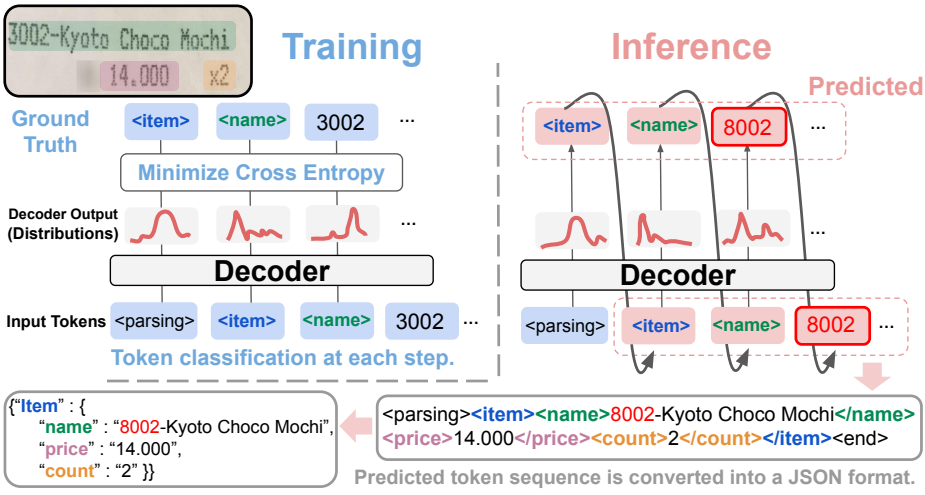


Fig. E. Donut training scheme with teacher forcing and decoder output format examples. The model is trained to minimize cross-entropy loss of the token classifications simultaneously. At inference, the predicted token from the last step is fed to the next

(8 V100 GPUs for 5 days). The input size was 2048×1536. In this setting, $\text{Donut}_{\text{Proto}}$ also achieved comparable results on *RVL-CDIP* and *CORD*. The accuracy on *RVL-CDIP* was 94.5 and *CORD* was 85.4. After the preliminaries, we have scaled the model training with more data.

A.7 Details of OCR-dependent Baseline Models

In this section, we provide a gentle introduction to the general-purpose VDU backbones, such as LayoutLM [65] and LayoutLMv2 [64]. To be specific, we explain how the conventional backbones perform downstream VDU tasks; document classification, IE, and VQA. Common to all tasks, the output of the OCR engine is used as input features of the backbone. That is, the extracted texts are sorted and converted to a sequence of text tokens. The sequence is passed to the Transformer encoder to get contextualized output vectors. The vectors are used to get the desired output. The difference in each task depends on a slight modification on the input sequence or on the utilization of the output vectors.

Document Classification At the start of the input sequence, a special token `[CLS]` is appended. The sequence is passed to the backbone to get the output vectors. With a linear mapping and softmax operation, the output vector of the special token `[CLS]` is used to get a *class-label* prediction.

Document IE With a linear mapping and softmax operation, the output vector sequence is converted to a *BIO-tag* sequence [22].

IE on 1-depth structured documents When there is no hierarchical structure in the document (See Figure B), the tag set is defined as {"B_k", "I_k", "O" | $k \in$ pre-defined keys}. "B_k" and "I_k" are tags that represent the beginning (B) and the inside (I) token of the key k respectively. The "O" tag indicates that the token belongs to no key information.

IE on n-depth structured documents When there are hierarchies in the structure (See Figure C), the BIO-tags are defined for each hierarchy level. In this section, we explain a case where the depth of structure is $n = 2$. The tag set is defined as {"B_g.B_k", "B_g.I_k", "I_g.B_k", "I_g.I_k", "O" | $g \in$ pre-defined parent keys, $k \in$ pre-defined child keys}. For instance, the Figure C shows an example where a parent key is "menu" and related child keys are {"cnt", "nm", "price"}. "B_g" represents that one group (i.e., a parent key such as "menu") starts, and "I_g" represents that the group is continuing. Separately from the BI tags of the parent key (i.e., "B_g" and "I_g"), the BI tags of each child key (i.e., "B_k" and "I_k") work the same as in the case of $n = 1$. This BIO-tagging method is also known as *Group BIO-tagging* and the details are also available in Hwang et al. [22].

Document VQA With a linear mapping and softmax operation, the output vector sequence is converted to a *span-tag* sequence. For the input token sequence, the model finds the beginning and the end of the answer span. Details can also be found in the Section 4.2 of Devlin et al. [8].