# VisualWordGrid: Information Extraction From Scanned Documents Using A Multimodal Approach

KERROUMI Mohamed[0000−0002−2513−8837], SAYEM
Othmane[0000−0003−0311−9384], and SHABOU Aymen[0000−0001−8933−7053]

DataLab Groupe, Credit Agricole S.A, Montrouge , France
kerroumimohamed1997@gmail.com
{othmane.sayem,aymen.shabou}@credit-agricole-sa.fr

[10]-Chargrid-18  [6]-Bertgrid-19

**Abstract.** We introduce a novel approach for scanned document representation to perform field extraction. It allows the simultaneous encoding of the textual, visual and layout information in a 3-axis tensor used as an input to a segmentation model. We improve the recent Chargrid and Wordgrid [10] models in several ways, first by taking into account the visual modality, then by boosting its robustness in regards to small datasets while keeping the inference time low. Our approach is tested on public and private document-image datasets, showing higher performances compared to the recent state-of-the-art methods.

**Keywords:** Information Extraction · Multimodal · Scanned document analysis · WordGrid · Chargrid

## 1 Introduction

In a vast majority of business workflows, information extraction from templatic documents such as invoices, receipts, tax notices, etc. is largely a manual task. Automating this process has become a necessity as the number of client documents increases exponentially. Most industrial automated systems today have a rule-based approach to documents with a certain structure, and can be associated with a finite number of templates. However, documents often have a variety of layouts and structures. In order to understand the semantic content of these documents, the human brain uses the document's layout, as well as the textual and visual information available in its contents.

The challenge is to overcome rule-based systems, and to design end-to-end models that automatically understand both the visual structure of the document and the textual information it contains. For instance, in a document like an invoice, the *total amount to pay* is associated with a numerical value that appears frequently near terms such as *total*, *total to pay* and *net to pay*, and also after fields like *total before taxes*, *taxes*, *cost*, etc. Thus, as Katti et al. showed with Chargrid [10], combining both positional and textual information, was proven to be efficient for this task.

On the other hand, the visual content of a document was proven to improve model accuracy for document classification when combined with textual information [1].

In this article, we prove that adding visual information to textual and positional features improves the performance of the information extraction task. The improvement is more significant when dealing with documents with rich visual characteristics such as tables, logos, signatures, etc. We extend the work of Katti et al. [10,6] with a new approach (called **VisualWordGrid**) that combines the aforementioned modalities with two different strategies to achieve the best results in the task of information extraction from image documents.

The present paper is organized as follows: Section 2 presents related work for information extraction. Section 3 describes the datasets we used for evaluation. Section 4 introduces the proposed approach. Section 5 discusses the obtained results. Finally, Section 6 provides our conclusions regarding the new method.

## 2    Related work

Interest in solving the information extraction task has grown in fields where machine learning is used, from Natural Language Processing (NLP) to Computer Vision (CV) domains. Depending on the representation of the document, different methods are applied to achieve the best possible performance.

For instance, NLP methods transform each document to a 1D sequence of tokens, before applying named entity recognition models to recognize the class of each word [11]. These methods can be successful when applied to documents with simple layout, such as books or articles. However, for documents like invoices, receipts or tax notices, where visual objects such as tables and grids are more common, these textual methods are less efficient. In such cases, structural and visual information are essential to achieve good performance.

Alternatively, computer vision methods can also be very efficient for this task, specifically for documents like Identity Cards which are very rich with visual features. In these approaches, only the image of the scanned document is given as an input. Object detection and semantic segmentation are among the most used techniques for field extraction [8] from these documents. The OCR engine is applied at the end of the pipeline on image crops to extract the text of detected fields. These approaches can be very useful when dealing with documents with normalized templates. For documents with various templates and layouts, these models do not perform well.

Most recent studies try to exploit the textual and the layout aspects of the document by combining both NLP and CV methods in the extraction task. In the Chargrid [10] or BertGrid [6] papers, a document is presented as a 2D grid of characters (or words) embeddings. The idea behind this representation is to preserve structural and positional information, while exploiting textual information contained in the document. Both papers reported significant increase in the performance of information extraction task compared to purely textual approaches. In a more general approach, Zhang et al. [20] recently proposed TRIE,

CharGrid-18
BertGrid-19
TRIE-20
LayoutLM-20-

an end-to-end text reading and information extraction approach, where a module of text reading is introduced. The model mixes textual and visual features of text reading and information extraction to reinforce both tasks mutually, in a multitask approach.

More recently, Yiheng et al. proposed the LayoutLM [16], a new method to leverage the visual information of a document in the learning process. Instead of having the text embedding of each token as the sole input, relative position of tokens in the image and the corresponding feature map of the image crop within the document were added too. Inspired by the BERT model [7], Yiheng et al. used scanned document classification task as a supervised pre-training step for LayoutLM to learn the interactions between text and layout information. Then they enforced this learning by a semi-supervised pre-training using Masked Visual-Lanquage Model (MVLM) as a multi-task learning. The dataset used for pre-training contains 11M documents and the pre-training took 170 hours on 8 GPUs. Hence this approach needs large computational resources.

Other works, focused on solving the document semantic segmentation task, introduced the idea of simultaneously encoding visual, textual and structural modalities into a tensor representation. For instance, Yang et al. [19] proposed a multimodal approach to extract the semantic structure of documents using a fully convolutional neural network. Barman et al. [2] proposed a multimodal segmentation model that combines both visual and textual features to extract semantic structure of historical documents.

Compared to these related works, we propose in this paper two multimodal document representation strategies suited to the information extraction task. The first one is simpler, yet highly effective compared to state-of-the-art multimodal approaches (while improving the extraction scores). The second one is similar to the related works [19,2] but slightly improved and adapted to the field extraction task (i.e. extracting small text regions).

## 3   Data

In order to evaluate our work, we will show our experiments on two datasets showing interesting visual structures that help improving the information extraction task using multimodal strategies.

**RVL-CDIP Dataset** [13] It is a public dataset that was released to help improve and evaluate layout analysis techniques on scanned invoice documents. It contains 520 invoice images (Fig.1) with their corresponding OCR files containing the extracted text, along with XML files containing the ground-truth bounding boxes of all the semantic fields. Each word in a given document of the dataset is classified into a semantic region described by a box. Among the 6 available fields, we will focus on extracting 4 of them: *Receiver*, *Supplier*, *Invoice_info*, *Total*.

**Fig. 1.** Invoice example from RVL-CDIP with fields of interest.

**Tax Notice Dataset** It is an in-house **private** dataset. It contains 3455 tax notices since 2015 (Fig.2). The documents are in French and their templates changed over the years. Hence template matching could not be used as an approach for information extraction. The dataset was annotated by manually putting bounding boxes around fields of interest. There are mainly 6 entities to extract from each document: *Year, Name, Address, Type_of_Notice, Reference_Tax_Income, Family_Quotient*. The dataset contains first and second pages from tax notices, as some fields can appear on both pages, depending on the issue date. Moreover, a single page doesn't necessarily contain all fields.

## 4   Method

In this section, we introduce the **VisualWordGrid** approach, a new 2D representation of documents that extends the Chargrid philosophy by adding the visual aspect of the document to the textual and layout ones. We define two main models that differ on document representation and model architecture : **VisualWordGrid-pad** and **VisualWordGrid-2encoders**.

### 4.1   Document representation

Our main idea is adding the visual information of the image to the textual and structural data used in the *WordGrid* representation. The most direct way for

**Fig. 2.** Tax Notice *fake* example with fields of interest.

doing so is by adding the corresponding RGB channels to each pixel embedding. While this concatenation has no impact on background pixels, it adds a large amount of noise to pre-trained word embeddings. The main challenge here is to adapt the concatenation method to preserve textual embeddings, while adding the background visual information. Our representations of documents extend [10] using two strategies as follows.

Using an OCR, each document can be represented as a set of words and their corresponding bounding boxes. The textual and layout information of each document can be represented in $\mathcal{D} = \{(t_k, b_k)|k = 1, ..., n\}$, with $t_k$ the k-th token in the text of the document and $b_k = (x_k, y_k, w_k, h_k)$ its corresponding bounding box in the image.

**VisualWordGrid-pad**

Our first model representation of the document is defined as follows :

$$W_{ij} = \begin{cases} (e_d(t_k), 0, 0, 0) & if \quad \exists k \quad such \quad as \quad (i,j) \prec b_k \\ (0_d, R_{ij}, G_{ij}, B_{ij}) & \text{otherwise} \end{cases} \tag{1}$$

$$(i,j) \prec b_k \iff x_k \le i \le x_k + w_k \quad \wedge \quad y_k \le j \le y_k + h_k$$

where $d$ is the embedding dimension, $e_d$ is the word's embedding function, $0_d$ denotes an all-zero vector of size $d$, and $(R_{ij}, G_{ij}, B_{ij})$ the RGB channels of the $(i, j)^{th}$ pixel in the raw document's image.

In other words, for each point in the document's image, if this point is included in a word's bounding box $b_k = (x_k, y_k, w_k, h_k)$, the vector representing this point is the word's embedding padded by $\mathbf{0}_3$. Thus, by setting the RGB channels to $\mathbf{0}_3$, we drop the visual information related to this point. However if the point is not included in any word's bounding box, the vector representing this point is the concatenation of $\mathbf{0}_d$ and the RGB channels of this point. In this case, we keep the visual information. Hence, the visual, textual and layout information of the document are encoded simultaneously in a 3-axis tensor of shape $(H, W, d+3)$ as shown in Fig.3, while preserving their original information.
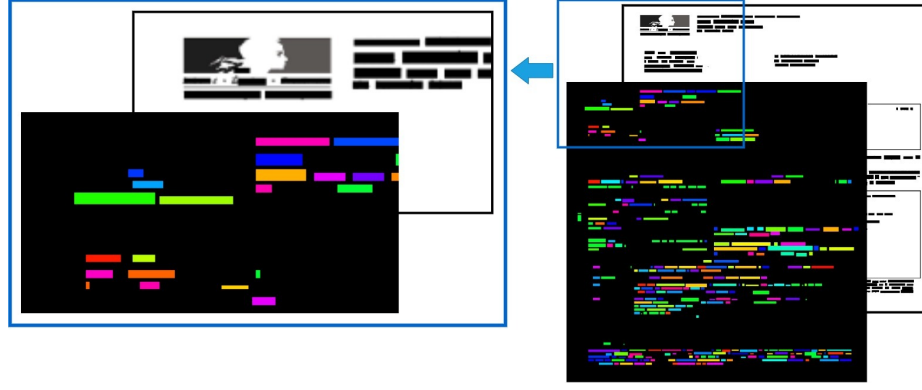


**Fig. 3.** VisualWordGrid encoding of an invoice sample. On the right, the proposed concatenation of a Wordgrid representation and the image of the document. On the left, a zoom of the previous figure.

**VisualWordGrid-2encoders**

Our second model representation is similar to the CharGrid-Hybrid approach presented in [10]. Instead of encoding the document on the character level using a one-hot encoding, we encode the document on the word level using *Word2Vec* [12] or *Fasttext* [4] embeddings. Hence, for each document we have two inputs:

- *WordGrid encoding*: This input encodes the textual and layout information of the document. This approach of encoding is similar to WordGrid presented in [10]. For words encoding, we use *Word2Vec* or *Fasttext* embeddings.

$$W_{ij} = \begin{cases} e_d(r_k) & if \quad \exists k \quad such \quad as \quad (i,j) \prec b_k \\ 0_d & \text{otherwise} \end{cases} \tag{2}$$

- *Image*: The raw image of the document resized to match the WordGrid encoding dimensions.

## 4.2   Model Architectures

In this section, we discuss model architectures related to both strategies.

**VisualWordGrid-pad**

Once the 2D representation of the document is encoded, we use it to train a neural segmentation model. Unlike *chargrid* and *wordgrid* papers, we dropped the bounding box regression block to keep the semantic segmentation block only, since there can be at most one instance of each class in the datasets.

We use the *Unet* [14] as a segmentation model and the *ResNet34* [9] as a backbone for the encoder. The weights of the backbone are initialized using transfer learning from a model pre-trained on the ImageNet classification task. These weights are available in the open-source package *Segmentation Models* [17]. The UNet component extracts and encodes advanced features of the input grid in a small feature map, and the decoder expands this feature map to recover segmentation maps of the same size as the input grid, and thus generates the predicted label masks. We used a *softmax* activation function for the final layer of the decoder. The shape of the decoder's output is $(H, W, K + 1)$, where $K$ is the number of fields of interest, and 1 is the background class.

In the inference step, we iterate over the bounding box of each word in the OCR output, then attribute a single class to the most dominant category pixel-wise, to get the final prediction value for the corresponding field. (see Fig.4).

**VisualWordGrid-2encoders**

This model is composed of 2 encoders. The first one is the classic WordGrid encoder (2), and the second one is the raw image encoder. We keep one decoder, and for each block in the decoder, we concatenate on the skip connections from both encoders (see Fig.4).

## 4.3   Implementation Details

In this section, we provide implementation details.

**Word embedding function**

We use different word embedding functions depending on the dataset. For RVL-CDIP, we propose Word2Vec pretrained embeddings on the Wikipedia corpus, publicly available thanks to *Wiki2Vec* [18]. The choice of *Wiki2Vec* is due to the good quality of the OCR files, since words are correctly recognized and most of them have their related embeddings. Unlike RVL-CDIP dataset, OCR outputs of our Tax Notice dataset are noisy, due to the quality of customer documents scans. We observed frequent misspelling errors in the Tesseract 4.1 [15] ouputs. Our experiments show that a custom *FastText* embedding trained on the corpus of the Tax Notice dataset is the best words embedding function to handle the noise. As explained in [4], Word embedding using this approach is the sum of n-grams subword embeddings. Hence, even in case of a misspelled

**Name: XXXXX**

**Address: XXXXX**

**Reference Tax Income: 22 910**

**Year: 2018**

**VisualWordGrid-2encoders pipeline**

C  Concatenation

Input/ Output

Deconvolution

Skip Connection

**Name: XXXXX**

**Address: XXXXX**

**Reference Tax Income: 22 910**

**Year: 2018**

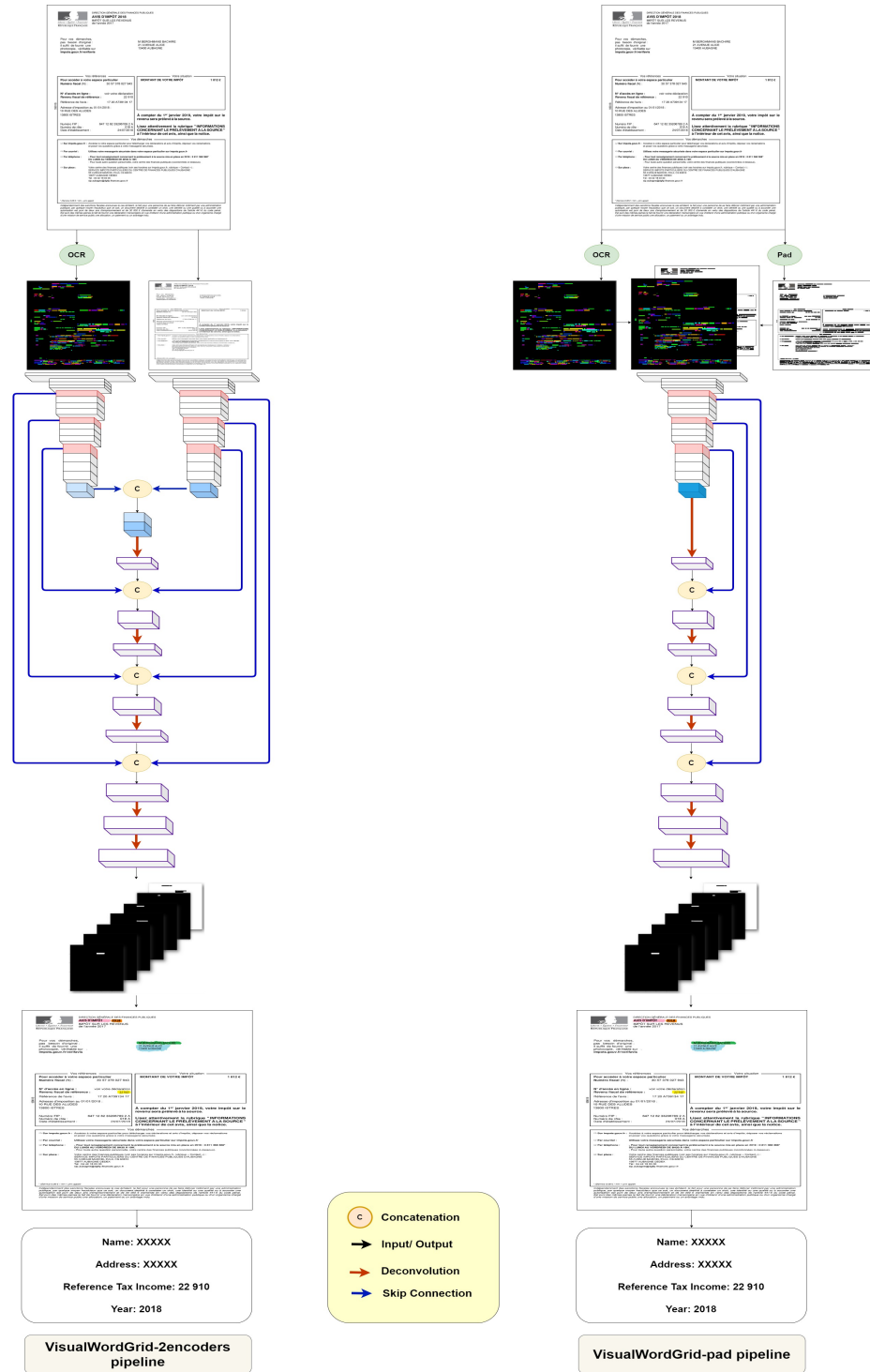**VisualWordGrid-pad pipeline**

**Fig. 4.** VisualWordGrid pipelines.

or dropped character in the token, its embedding wouldn't differ too much from the embedding of the original word.

**Loss function**

The loss function we use for training is the sum of the cross entropy loss for segmentation ($L_{seg}$) and the Intersection over Union loss ($L_{IoU}$).

$$Loss = L_{seg} + L_{IoU} \tag{3}$$

- **Cross Entropy Loss**: The cross entropy loss penalizes pixel mis-classification. The goal is to assign each pixel to its ground truth field.

$$L_{seg} = \sum_{x \in \Omega} -log\big(p_{\hat{l}(x)}(x)\big) \tag{4}$$

with $\hat{l}(x)$ the ground truth label of the point $x$.

- **Intersection over Union Loss**: This loss function is often used to train neural networks for a segmentation task. It's a differentiable approximation of the IoU metric and is the most indicative of success for segmentation tasks as explained in [3]. In our case, it significantly increases performances of the model compared to a model trained only with the cross entropy. The IoU metric is defined as :

$$IoU = \frac{I}{U} = \frac{|T \cap P|}{|T \cup P|} = \frac{|T \odot P|}{|T + P - (T \odot P)|} \tag{5}$$

$$L_{IoU} = 1 - IoU \tag{6}$$

where $T$ is the true labels of the image pixels and $P$ is their prediction labels. We also use the $IoU$ metric to monitor the training of our models.

**Metrics**

To evaluate the performance of the different models, we used two metrics:

- **Word Accuracy Rate (WAR)**: It's the same metric as the one used in [10]. It's similar to the *Levenshtein distance* computed on the token level instead of the character level. It counts the number of substitutions, insertions and deletions between the ground-truth and the predicted instances. This metric is usually used to evaluate speech-to-text models. The WAR formula is as follows:

$$WAR = 1 - \frac{\#[insertions] + \#[deletions] + \#[substitutions]}{N} \tag{7}$$

where, $N$ is the total number of tokens in the ground truth instance for a specific field. The WAR of a document is the average on all fields.

– **Field Accuracy Rate (FAR)**: This metric evaluates the performance of the model in extracting complete and exact field information. A field is the set of words of a same entity. This metric counts the number of exact match between the ground-truth and the predicted instances. It is useful in industrial applications, as we need to evaluate the number of cases where the model succeeds to extract the whole field correctly, for control purposes for example. The FAR formula is as follows:

$$FAR = \frac{\#[\text{Fields exact Match}]}{N_{fields}} \tag{8}$$

where, $\#[\text{Fields exact Match}]$ is the number of fields correctly extracted from the document with an exact match between the ground-truth and the predicted words values, and $N_{fields}$ is the total number of fields. We note that for any processed document in the evaluation set, with no target field in the ground truth, we attribute an empty string to the value of each field, so false positives are penalized too.

In the next section, we will report for each model the average WAR and FAR metrics on the documents in the test set.

## 5  Experiments

In this section, we compare our approaches (VisualWordGrid-pad, VisualWordGrid-2encoders) to two others, on both datasets. We report their average scores ($\overline{FAR}$, $\overline{WAR}$) and inference time ($\overline{InferenceTime}$) on CPU for a single document, and their number of trainable parameters.

The two competing approaches are the following ones:

– **Layout Approach**: This approach is a layout encoding only. Instead of using word embedding or pixel RGB channels to encode a specific document as in (1), it uses a simpler 2D encoding suited to a segmentation task, i.e.:

$$W_{ij} = \begin{cases} (1,1,1) & if \quad \exists k \quad \text{such} \quad \text{as} \quad (i,j) \prec b_k \\ (0,0,0) & \text{otherwise} \end{cases} \tag{9}$$

Then, we use this type of document encoding (Fig. 5) as input to train an information extraction model using the proposed architecture, loss and model hyper-parameters.
– **WordGrid**: This approach is very similar to BertGrid [6]. Instead of using a Bert [7] model to generate contextual embeddings, we use a *Word2Vec* pre-trained embedding for RVL-CDIP dataset, and a custom *Fasttext* embedding for the Tax Notice dataset. Equation (2) introduces the document encoding formula.
We keep the same model architecture, loss function and model hyper-parameters as proposed in the VisualWordGrid model.
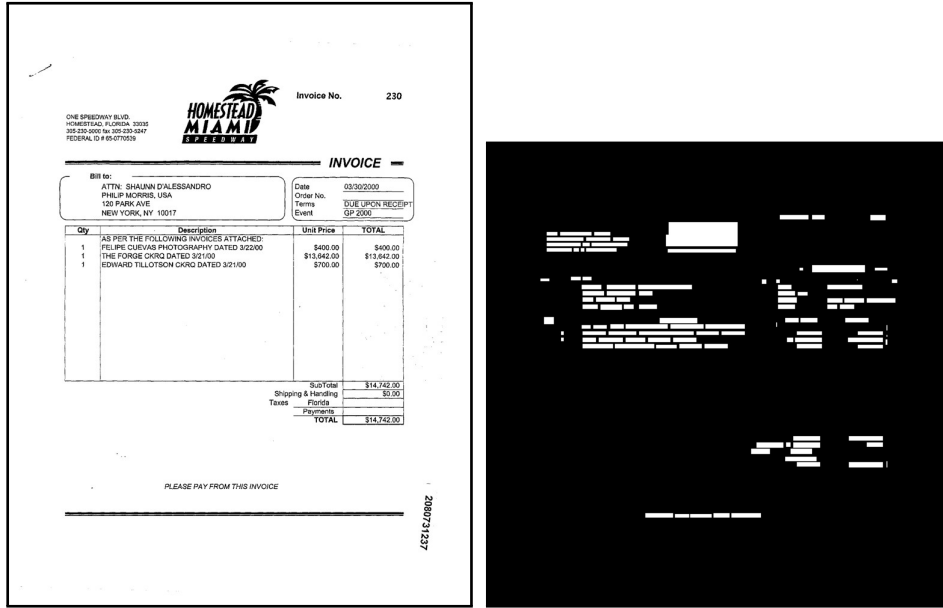
**Fig. 5.** Invoice sample and its encoding using the Layout approach

### 5.1   Datasets

Since the RVL-CDIP dataset volume is very small, we don't use a classic split of the dataset into training set and validation sets. Instead, we use a *k-fold* split of the dataset with $k = 5$. For each experiment, we do 5 tests, each one with a training on 80% of the dataset and the remaining 20% is split equally into validation and test sets. We report the average of the metrics on the 5 tests. This way, the values of the metrics don't depend on the seed of the split, and metrics are a more reliable representation of real model performance.

For the Tax Notice dataset, we assign 80% of the dataset to training, 15% to validation and 5% to test, on which we report our results. The OCR task to extract textual information was performed using the open source OCR engine Tesseract 4.1.

### 5.2   Results

For all experiments , we use *Adam* optimizer with *lr=0.001* and *batch_size*=8. We use a GPU NVIDIA Quadro RTX 6000 with 24GB GPU memory and the *Keras* framework [5] to build and train models. The inference time is measured on Intel Xeon W-2133 CPU (3.60 GHz). The table 1 shows the scores ($\overline{WAR}$, $\overline{FAR}$) of the different approaches on the RVL-CDIP dataset.

We clearly see in the table 1 that **VisualWordGrid-pad** gives the best FAR and WAR scores. Our proposed encoding system improves the WordGrid FAR

**Table 1.** Models performances on the RVL-CDIP dataset.

| Approach | $\overline{FAR}$ | $\overline{WAR}$ | $\overline{InferenceTime}$ | #Parameters |
|---|---|---|---|---|
| Layout Only | 23.0 % | 5.4 % | 2.14 s | 24 439 384 |
| WordGrid | 27.7 % | 10.8 % | 2.22 s | 24 743 673 |
| **VisualWordGrid-pad** | **28.7 %** | **18.7 %** | 3.77 s | 24 753 084 |
| VisualWordGrid-2encoders | 26.9 % | 17.0 % | 6.08 s | 48 003 004 |

and WAR by **1** and **7.9** respectively. Moreover, it exploits all the visual, textual and structural content of documents while keeping the inference time and the number of parameters close to the WordGrid ones.

Unlike Katti et al. [10], we notice an increase in the WAR score when using the two encoders approach (VisualWordGrid-2encoders) to capture the visual and textual aspect of document. It boosts the WordGrid performance, since the WAR goes up by **6.2** . The reasons for this improvement are the modifications we added to make the model more robust in the information extraction task. We used a ResNet34 backbone for the encoder and took advantage of transfer learning to speed up the training of the model. We also changed the cross entropy loss used in [10] by adding the IoU loss to it. Notice that we used the $IoU$ as a metric for the callback.

Similarly, we tested the different approaches on the Tax Notice dataset. We reported the results in table 2.

**Table 2.** Models performances on the Tax Notice dataset.

| Approach | $\overline{FAR}$ | $\overline{WAR}$ | $\overline{InferenceTime}$ | #Parameters |
|---|---|---|---|---|
| Layout Only | 83.3 % | 92.3 % | 5.29 s | 24 439 674 |
| WordGrid | 83.6 % | 92.4 % | 5.70 s | 24 743 963 |
| VisualWordGrid-pad | 83.9 % | 92.9 % | 5.92 s | 24 753 374 |
| **VisualWordGrid-2encoders** | **85.8** % | **93.6 %** | 6.19 s | 48 003 294 |

The VisualWordGrid-padding approach slightly improves the WordGrid scores, while the VisualWordGrid-2encoders gives the best performance but at the expense of a slightly higher inference time.

As in several industrial applications, using information extraction requires the smallest inference time. VisualWordGrid-pad would be the best choice. It leverages the visual/textual/layout information of a document while keeping the number of trainable parameters roughly the same as WordGrid.

## 6   Conclusion

VisualWordGrid is a simple, yet effective 2D representation of documents that encodes the textual, layout and visual information simultaneously. The grid-based representation includes token embeddings and the image's RGB channels. We can take advantage of these multimodal inputs to perform several document understanding tasks. For the information extraction task, VisualWordGrid shows better results than those of state of the art models on two datasets (the public RVL-CDIP dataset and the private Tax Notice dataset), while keeping model parameters and inference time roughly the same (especially when using the padding strategy). In many fields, this approach is suitable for production.

## References

1. Audebert, N., Herold, C., Slimani, K., Vidal, C.: Multimodal deep networks for text and image-based document classification. In: Machine Learning and Knowledge Discovery in Databases. pp. 427–443. Springer International Publishing, Cham (2020)
2. Barman, R., Ehrmann, M., Clematide, S., Oliveira, S.A., Kaplan, F.: Combining visual and textual features for semantic segmentation of historical newspapers (2020)
3. van Beers, F., Lindström, A., Okafor, E., Wiering, M.: Deep neural networks with intersection over union loss for binary image segmentation. In: ICPRAM (2019)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
5. Chollet, F., et al.: Keras (2015), `https://github.com/fchollet/keras`
6. Denk, T., Reisswig, C.: Bertgrid: Contextualized embedding for 2d document representation and understanding. CoRR **abs/1909.04948** (09 2019), `http://arxiv.org/abs/1909.04948`
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
8. Hao, L., Gao, L., Yi, X., Tang, Z.: 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 (04 2016). https://doi.org/10.1109/DAS.2016.23
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2016)
10. Katti, A.R., Reisswig, C., Guder, C., Brarda, S., Bickel, S., Höhne, J., Faddoul, J.B.: Chargrid: Towards understanding 2d documents. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. pp. 4459–4469. Association for Computational Linguistics (2018), `https://www.aclweb.org/anthology/D18-1476/`
11. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics:

Human Language Technologies. pp. 260–270. Association for Computational Linguistics, San Diego, California (Jun 2016). https://doi.org/10.18653/v1/N16-1030, `https://www.aclweb.org/anthology/N16-1030`

12. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)
13. 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)
14. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
15. Smith, R.: An overview of the tesseract ocr engine. In: Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR). pp. 629–633 (2007)
16. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Aug 2020). https://doi.org/10.1145/3394486.3403172, `http://dx.doi.org/10.1145/3394486.3403172`
17. Yakubovskiy, P.: Segmentation models (2019)
18. Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., Matsumoto, Y.: Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. arXiv preprint 1812.06280v3 (2020)
19. Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., Giles, C.L.: Learning to extract semantic structure from documents using multimodal fully convolutional neural network (2017)
20. Zhang, P., Xu, Y., Cheng, Z., Pu, S., Lu, J., Qiao, L., Niu, Y., Wu, F.: Trie: End-to-end text reading and information extraction for document understanding (2020)