Knowledge Integration Inside Multitask Network for Analysis of Unseen ID Types

Timothée Neitthoffer, Aurélie Lemaitre, Bertrand Coüasnon, Yann Soullard, Ahmad Montaser Awal

Knowledge Integration inside Multitask Network for Analysis of Unseen ID Types

Timothée Neitthoffer[1,3], Aurélie Lemaitre[1], Bertrand Coüasnon[1], Yann Soullard[1,2], and Ahmad Montaser Awal[3]

[1]Univ Rennes, CNRS, IRISA, Rennes, France
[2]Université Rennes 2, CNRS, LETG, IRISA, Rennes, France
[3]IDNow, AI & ML Center of Excellence, Cesson-Sévigné, France

Abstract. Identity Document recognition is a key step in Know Your Customer applications where identity documents (IDs) are verified. IDs belonging to the same type share the same field structure called template. Traditional ID pipelines leverage this template to guide the localisation of the fields and then the text recognition. However, they have to be tuned to the different templates to correctly perform on those. Thus, such pipelines can not be directly used on new types of IDs. In this work, we address the task of text localisation and recognition in the context of new document types, where only the template is available with no labeled samples from the new ID type. To that end, we propose the use of Context Blocks (CB) performing template self-attention to guide the features of the network by the template. We propose three ways to leverage CB in a multitask architecture. To evaluate our approach, we design a new public task for the MIDV2020 database from rectified in-the-wild photos. Our method achieves the best results for two datasets including an industrial one composed of real examples.

Keywords: Knowledge Integration, Multitask Learning, OCR, Text localisation, Identity Documents, Deep Network

1 Introduction

Optical Character Recognition (OCR) is a major research topic in computer vision where text is recognized from captured images. Although it first appears in the 80's, text recognition remains a challenging problem with highly variable documents and capture conditions. In this work, we are interested in Know Your Customer applications where a service provider extracts user's information from their identity documents (IDs). Countries issue IDs for different uses (id cards, driving licenses, etc) while following the international norms (such as ICAO).

Each kind of ID contains specific information and thus present different template, background and security elements. IDs belonging to the same type and the same version (for example "2015 French ID card") share the same template as shown in Fig. 1. Although the template possesses strong structural knowledge, it is not enough to localize the fields. Indeed, as shown in Fig.2, the images are

Fig. 1: a,b,c,d) Greek passports from MIDV2020 e) Greek template

Fig. 2: Pipeline of information extraction in IDs.

captured in real conditions, mainly with smartphones, and then rectified from their detected quadrilateral to a plane document. However, the uncontrolled environment and the variable capture conditions with projections, introduce shifts and deformations: the fields are distorted and different in size from the template. Thus, it is not possible to apply the template directly onto an image.

Traditional pipelines (Fig. 2) deal with in-the-wild ID images captured from various devices. The document image is first localised and classified in the input image and then rectified. The fields are detected by aligning the result of text localisation and the known template. The text fields are extracted from the rectified image and then recognized by a generic OCR. The quality of the predictions highly depends on the alignment process, that may propagate errors and must be trained for each kind of document.

In this work, we consider as input a rectified document image. We focus on field localisation and recognition. We especially focus on dealing with new ID types from new countries or newer versions. To correctly handle new types, the traditional methods need to be hand-tuned in an *ad hoc* manner on annotated data. However, it is difficult to obtain labeled data for those new ID types, due to the sensitive information they contain. In that context, we only benefit from

the known template as an *a priori* knowledge, without the need of any labeled example from the new ID types. We propose the following contributions:

- define core concepts for knowledge integration in neural networks;
- propose template self-attention using a Context Block that guides the predictions and allows to deal with new ID types (templates not seen in training);
- design a new dataset and task on the MIDV2020 database for field localisation and recognition, available to the community [1].

This paper is organized as follows. In section 2, we introduce the works related to ID recognition, joint text localisation and recognition and knowledge integration. Then, we present proposals on knowledge integration inside neural networks through the use of an auxiliary branch and template self-attention. In section 4, we describe the new dataset and task we designed for MIDV2020 [5]. Finally, we present the result of our experiments on the new MIDV2020 dataset and on a private industrial dataset before concluding.

2 Related works

In this section, we first introduce the methods for information extraction on IDs. We then present the different works on joint text localisation and recognition, and on knowledge integration in models, which can be outside of the ID domain.

2.1 Identity Document Analysis

ID analysis involves the localisation of the fields and their recognition. Mustafina *et al.* [14] and Van *et al.* [16] perform text localisation using neural network architectures. However, due to the complexity of the background and the variability of capture conditions and of the document templates, the models tend to overdetect text fields. As discussed before, IDs of the same type share the same template. This template is sometimes used to guide the prediction of text field localisation. For instance, Attivissimo *et al.* [1] uses the template as a crop mask to directly extract the fields from the image. However, this method makes the assumption that the ID is perfectly localised and rectified. This is rarely the case due to inaccuracies from the previous steps. In addition, the authors do not evaluate the CER when the localisation is not correct. Also, ID printing process incorporates field position variations from one document to another, making the direct application of the template impossible. Bulatovich *et al.* [3,4] perform text detection via morphological analysis and then apply a template alignment algorithm to extract the fields. Nevertheless, this alignment balances between the different kinds of templates and need to be hand-tuned in an *ad hoc* manner, using labelled samples for dealing with new ID types. This is particularly difficult as, in case of new document types, only the template may be available. Text recognition in the cropped field images is usually performed using a Convolutional Recurrent Neural Network (CRNN) [1,14,16].

[1] `https://gitlab.inria.fr/tneittho/midv2020-rectified-photo`

A two step approach propagates the errors from the localisation to the recognition task. To address this problematic, we seek to perform jointly the localisation and the recognition. asa well as leveraging the templates in this joint context. In the next sections, we present works outside of the ID domain for joint text localisation and recognition as well as knowledge integration.

2.2 Joint Text Localisation and Recognition

Recent works have addressed this joint tasks by performing implicit localisation using attention mechanism. Indeed, Bluche *et al.* and Coquenet *et al.* [2,8] perform paragraph level and page level recognition on ancient documents by leveraging the attention mechanism to deal with line breaks or to select the current line to read. Furthermore, methods from Yousef *et al.* [17] and Coquenet *et al.* [7] guide the network to unfold the multiple lines of the document into a single sequence. However, these methods deal with single column documents where IDs present complex layouts and backgrounds on the entire document.

Coquenet *et al.* [9] addresses this problem by learning the reading order of the text blocs in the document with the use of Transformers. However, this method heavily relies on the templates seen at training time, from real data and generated synthetic data. Generating real samples of synthetic IDs is a very hard task which is out of the scope of this work. The lack of real examples limits the use of such kind of methods, especially when dealing with new ID types.

Another way to address the problematic is the design of a multitask framework. In [6] and [15], the authors leverage a multitask network to perform ancient document recognition. A first part of the network is dedicated to text localisation as object detection. The detected areas are used to pool the feature maps of the backbone which are then given in input to the second part, dedicated to text recognition. The backbone is shared by the two parts. This kind of structure allows to focus the recognition only on the relevant parts of the document.

2.3 Knowledge Integration

Administrative documents, similarly to IDs, usually follow a structure that can be used to guide the recognition. We can distinguish two kinds of approaches for using *a priori* knowledge. The template approach is commonly used in current industrial pipelines. A mask or a graph [11] is created and adjusted to the document image. The full-text approach defines a number of logical rules based on detection and recognition results and the content is associated to the related semantic value (Couasnon *et al.* [10] and Guerry *et al.* [12]). However, these methods are not optimized based on the available knowledge in a end-to-end manner.

Neural networks can benefit from this knowledge to obtain better performances. In [13], a feature vector representing the possible classes is concatenated to the features of the network to guide disease classification. Such proposition is interesting as it allows to condition the predictions by the input knowledge.

Currently, no acceptable proposition have been made to answer text localisation and recognition in the context of new ID types. Traditional solutions in the ID domain rely on a two step approach that easily propagates errors from the localisation to the recognition. In addition, the alignment algorithms proposed to leverage the template at disposition need to be fine-tuned using labelled samples of the new ID type. Propositions from Carbonel *et al.* [6] and Kushibar *et al.* [13] are interesting in our context but need to be further explored.

3 Proposed Methods

Based on the State of the Art, we aim to integrate the knowledge inside a multitask network. We further develop those ideas and adapt them to the analysis of new ID types. In this section, we formalize the paradigm of knowledge integration in neural network by defining its core concepts. We then present our proposals to leverage them to address new ID types recognition without any additional data. Finally, we give details about our architectures.

3.1 Definition of Knowledge Integration inside Neural Networks

We propose to divide Knowledge Integration inside neural networks into three core concepts that complements one another: the **knowledge representation**, the **knowledge connection** and the **integration operation**.

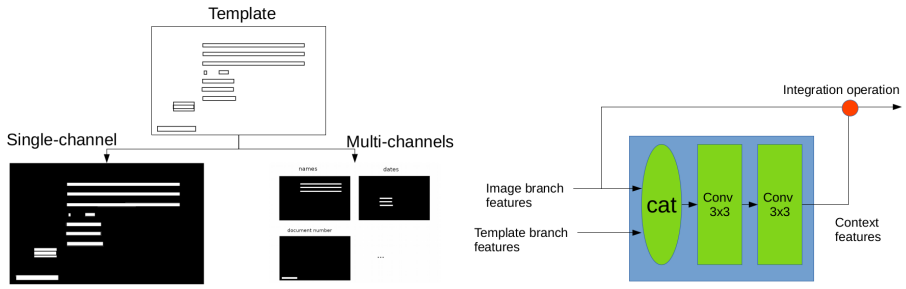
The **knowledge representation** corresponds to the form of the knowledge used as input of the network. For example the template is integrated as a graph or as a mask and what kind of graph or mask (section 3.2). We call **knowledge connection** the way the knowledge representation is processed, and where it connects to the feature maps from the input (section 3.3). The **integration operation** is the method used to mix the features from the template and the ones from the input (section 3.4).

3.2 Knowledge Representation

We propose to give as input to the model an ID image and the corresponding template. The template contains the knowledge about the fields localisation and type (first name, emission date, ...). To remain in the same modality as the image, we consider the template as a mask of the same dimensions as the input image. We explore two representations (Fig. 3a)). A **single-channel** mask, which is a binary mask to discriminate between the text fields and background and a **multi-channel** mask where each channel corresponds to one type of field.

3.3 Knowledge Connection

We choose to integrate the template inside the backbone in order to extract features combining the template and the input image. As the backbone is shared by the localisation and recognition heads, the knowledge integration (i.e. the



a) b)

Fig. 3: a) Template representation as a single-channel or a multi-channel mask. b) Context Block (CB) for template integration. The features of both image and template branches are concatenated. Two convolutions are used to build feature maps that are then given to the integration operation (see section 3.4).

template) will impact the predictions of the whole system. Thanks to that, the network is able to focus on the important features based on the template and it can be easily used with new document types, that we called **unseen** ID types.

Knowledge Connection as an Additional Channel First, an intuitive way is to consider the template as another channel of the image. That way, the network can extract correlations between the channels and process directly the template in the input representation. Here, the backbone is the same than the one presented in Fig. 4. We refer to this architecture as **Naive**. However, doing so might lead the neural network to consider the template as part of the graphical structure of the image and thus misuse it.

Knowledge Connection as an Auxiliary Branch A second approach is to process the template and the image in two parallel branches that are then connected one to another. To connect the two branches, we introduce a **context block** (CB) presented in Fig. 3b). This block is composed of two convolutions and takes as input the concatenation of the image branch features and the template branch features. Thus, a CB is used to combine the two sets of feature maps and to output context features that will be mixed with the image branch. In the following paragraphs, we explore three variants based on this method.

In a first architecture (Fig.5), the template branch follows the descending stage of the image branch, so that the *a priori* knowledge guides the construction of more complex features. We refer to this architecture as **Multitask with Knowledge Integration (MKI)**. Nevertheless, the reconstruction stage might benefit from the template branch, this is why we propose two other variants.

The second architecture **MKI-S** (Fig.6) uses skip connections from the template branch in each up-sampling block in the reconstruction stage. Thus, the representations of the template at different resolutions guide the reconstruction.

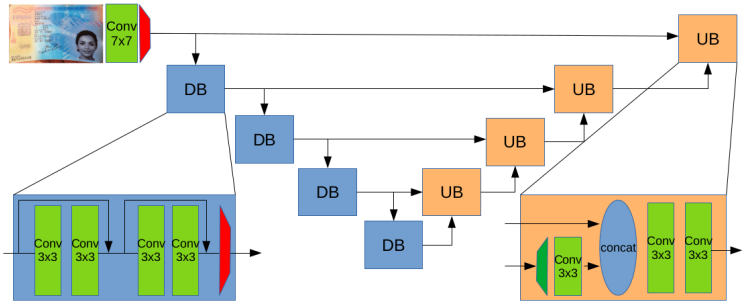


Fig. 4: Our multitask network backbone based on a UNet-like architecture. The descending stage is made of Down Blocks (DBs) and the ascending stage of Up Blocks (UBs). UBs take the feature maps of the previous stage (lower arrow) and the skip connection from the DB (upper arrow). The red trapezoids are max-pooling operations and the green ones are upsampling with cubic interpolation.

The last variant **MKI-D** (Fig. 7) extends the template branch to match the image one, creating a double UNet structure. This structure makes the template branch to compute specific features for the reconstruction stage. However, it makes the branch harder to train as the template branch depth increases.

3.4 Mixing the knowledge features into the image branch

As introduced in section 3.1, the knowledge integration is also determined by **integration operations**. They specify the way to mix the feature maps from the context blocks into the image branch (Fig. 3b)). In our context, the system should focus on the important elements according to the template. It is indeed important in case of IDs due to large and various background parts that don't refer to text fields. We propose a template self-attention operation on the image features, based on the features coming from the context block (see Fig. 3b)).

3.5 Implementation details

The different architectures follow the figures presented in Section 3.3. The backbone is a U-Net like structure starting with a 7×7 convolution with a stride of 2 and 64 output channels. It is composed of four stages down and four stages up. After each down stage a maxpooling operation reduces by two the dimension of the feature maps and the number of output channels is multiplied by two. The up stages take as input the previous stage feature map as well as a skip connection from the stage of the same resolution. The previous stage feature maps are upsampled following a bilinear policy before going through a 3×3 convolution. The two inputs are then concatenated and processed by two 3×3 convolutions.

In this work, field localisation is performed by semantic segmentation. Our localisation head is composed of three 3×3 convolutions with a sigmoid activation

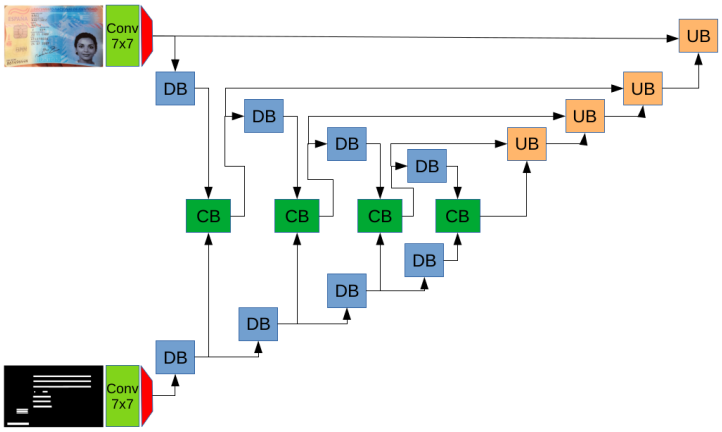


Fig. 5: Backbone of the MKI architecture. The upper branch is the document image branch and the lower branch the template branch. The Context Blocks (CB) in between allow the information of the template branch to flow to the document image branch. The DBs and UBs are the same as in Fig.4.

to discriminate each pixel as either background or text field. Our localisation head is trained using a binary mask as label and the balanced cross-entropy (BCE) loss to take into account the class imbalance. Then, we compose the bounding boxes by reducing the connected predicted pixels to rectangular boxes. We filter out boxes with an area under a threshold. The boxes are then used to pool the corresponding areas in the feature maps produced by the backbone.

The recognition head is a traditional CRNN-like module. It takes as input the list of pooled features corresponding to the detected fields. It is composed of two 3×3 convolutions, two BLSTM and finally a linear layer to output the character probabilities. This recognition head is trained using the well-known CTC loss. As we deal with both localisation and recognition tasks, we associate the target differently from a traditional text recognition task. If a predicted bounding box overlaps with a ground truth box, then the expected label is the one associated to the ground truth box. If the predicted bounding box does not overlap with any predicted ground truth box, the expected label is the empty string.

We balance the two losses by linearly interpolating the BCE and the CTC loss. The system is trained for 150 epochs on different GPUs with the Adam optimizer and a learning rate of 1×10^{-4}.

4 Databases

We now present another contribution: a new dataset and task on MIDV2020 database, that evaluates both field localisation and text recognition.

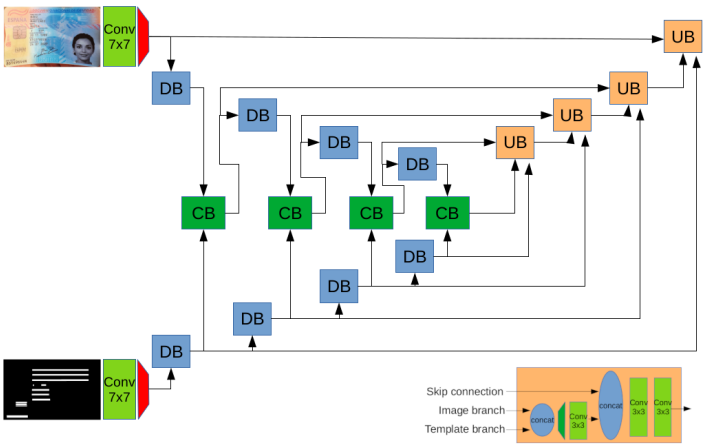


Fig. 6: Backbone of the MKI-S architecture. The skip connections propagate template information to the reconstruction stage. The UB is modified to take the incoming skip connections from the template branch in input.

4.1 New "*rectified photos*" dataset and new task for MIDV2020

The main public databases for IDs are the MIDV databases with the latest version being MIDV2020 [5]. The dataset "*templates*" of MIDV2020 contains the images of the different documents in full page with the annotations of the position of the fields as well as their corresponding text label and field type. However, this dataset does not reflect real capture conditions, such as lightning conditions or deformations. Furthermore, MIDV2020 contains a "*photos*" dataset, made of ID in real capture conditions, but that does not contain the annotations at field level. To that end, we propose a new dataset for MIDV2020 and a new task to evaluate Text Localisation and Recognition on Unseen ID Types (TLR-UIDT).

To create the new dataset, each document image, cropped from the "photos" dataset, is rectified to the dimensions of the same document image from the "templates" dataset. The annotations of text field positions coming from the "templates" dataset are then rectified to match the localisation of text fields from the resized document. We also design the template of each ID. We obtain a new dataset called "*rectified photos*", composed of the same 1000 document images as in the MIDV2020 dataset. It contains 10 ID types with their template and 100 document images per type. The text field positions and transcriptions for each document are known for training and test.

Using this "*rectified photo*" dataset, we propose a new task: TLR-UIDT that addresses text localisation and recognition of new ID types that we call *unseen ID types* : this refers to images from ID types (and templates) never seen in training. The rectified document images can be done in input of a network which is trained to output the text localisation, the name of the field and the textual content. Table 1 illustrates the training, validation and test set. To evaluate systems on

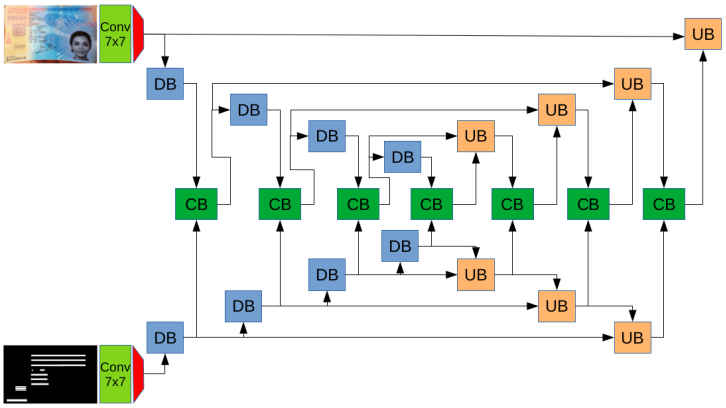


Fig. 7: Backbone of the MKI-D architecture. The template branch follows the reconstruction stage to guide the reconstruction step.

Table 1: Organisation of the *TLR-UIDT* task, for the public "*Rectified Photo*" dataset and the private dataset.

"*rectified photos*"	Types of ID	Nb of images	Image distribution
Train	8 ID types	640	80 images/ID type
Validation	8 seen ID types + 1 unseen ID type	180	10 images/seen ID type + 100 images from unseen val ID type
Test	8 seen ID types + 1 other unseen ID type	180	10 images/seen ID type +100 images from unseen test ID type
Private			
Train	7 ID types	6629	947 images/ID type
Validation	4 seen ID types + 4 unseen ID types	4912	614 images/ID type
Test	5 seen ID types + 13 unseen ID types	2147	113 images/ID type

unseen IDs types, we reserve one type of document for validation and another one for test. We also add examples in validation and test of the types seen in training to evaluate the model prediction and prevent overfitting. We propose to perform a 10-fold cross-validation, following the experimental setting shown in Table 1 so that each ID type is considered unseen once. For example, fold 0 would keep the *Serbian passport* for the unseen ID type in validation and the *Slovak id card* for the unseen ID type in test, fold 1 the *Albanese id card* and the *Greek passport*, and so on. Both the new dataset "*rectified photo*" and the splits for the new task TLR-UIDT are publicly available.

4.2 Metrics

To evaluate the performance of field localisation, we use common metrics: the recall, the precision, and the Intersection over Union (IoU). A value of IoU equal to 0.8 is used in order to evaluate the quality of text field localisation (for

computing both the recall and precision). Using such an high value of IoU for extracting boxes and computing the Character Error Rate (CER) may not reflect the quality of text recognition due to box rejection based on the threshold. Thus, for computing the CER, we select an IoU threshold equal to 0.1. We distinguish three cases: i) true positives, when the system should predict the label of the associated field; ii) false positives, when the system should predict an empty string as no field is present; iii) true negatives, when the system did not predict the box then it is considered to have predicted an empty string. Each metric is computed for ID types (and templates) seen during training (*seen* case) and for those not seen at training time (*unseen* case).

4.3 Private Identity Document dataset

In the following experiments, we also apply our work on a private dataset (Table 1). It is made of real world samples. We study a subset of 24 ID types. The dataset is split into a *train_set*, a *val_set* and a *test_set*. Similarly to the new task on MIDV2020, we keep unseen ID types in validation and test. For each set, the number of images per ID type is balanced following Table 1:

- *train_set* seen ID types:Albanian Passport front, Austrian Passport front, Czech Identity Card front, Czech Identity Card back, Swiss Identity Card front, Swiss Identity Card back, Swiss Passport front;
- *val_set* unseen ID types: Belgian Residence Permit front, Belgian Residence Permit back, Belgian Passport front, Bulgarian Passport front;
- *test_set* unseen ID types:Austrian Driving License front, Belgian Identity Card front, Belgian Driving License front, Bosnian Passport front, Beninese Passport front, Croatian Identity Card front, Croatian Passport front, German Residence Permit front, Greek Passport front, Lithuanian Passport front, Polish Identity Card front, Polish Identity Card back, Ukrainian Passport front.

This dataset is harder than the MIDV2020 dataset as the samples present more capture variations and some of them have ID localisation errors. Indeed, as the ground truth of the ID localisation is not available, perfect rectification is hard and can distort the field structure. As this is an industrial dataset that contains sensitive information, we cannot provide it to the community.

5 Results

We now evaluate our proposals for knowledge integration: the different architectures for **knowledge connection** and the **knowledge representations** on performances. Finally, we compare our best method to the state of the art approaches. In our experiments, we do not use any data augmentation strategy nor synthetic data as we focus on evaluating the impact of the different approaches.

Table 2: Comparison of knowledge connection architectures on the TLR-UIDT task of MIDV2020. The mean and deviation are computed over the ten folds.

	Architecture	Naive	MKI	MKI-S	MKI-D
Seen ID types	cer ↓	0.46 ± 0.14	0.17 ± 0.02	$\mathbf{0.14} \pm 0.02$	0.17 ± 0.03
	recall ↑	0.94 ± 0.01	0.94 ± 0.01	0.94 ± 0.02	$\mathbf{0.96} \pm 0.01$
	precision ↑	0.93 ± 0.02	$\mathbf{0.95} \pm 0.01$	0.94 ± 0.01	0.94 ± 0.01
	IoU ↑	0.87 ± 0.01	0.87 ± 0.01	0.87 ± 0.02	$\mathbf{0.88} \pm 0.01$
Unseen ID types	cer ↓	0.81 ± 0.16	0.34 ± 0.06	$\mathbf{0.31} \pm 0.07$	0.47 ± 0.11
	recall ↑	0.80 ± 0.12	$\mathbf{0.86} \pm 0.09$	0.84 ± 0.08	$\mathbf{0.86} \pm 0.08$
	precision ↑	0.84 ± 0.08	0.84 ± 0.10	$\mathbf{0.87} \pm 0.09$	$\mathbf{0.87} \pm 0.10$
	IoU ↑	0.77 ± 0.09	0.79 ± 0.06	0.79 ± 0.05	$\mathbf{0.80} \pm 0.05$

5.1 Comparison of the Knowledge Integration Architectures

We evaluate the different architectures for **knowledge connection** (see section 3.3) on the new MIDV2020 TLR-UIDT task (Table 2). We observe that the Naive variant is not efficient. For both seen and unseen ID types, it presents a really high CER (0.46/0.81). This shows that the use of an additional channel is not the best way to integrate the template.

For the seen ID types, we note that the localisation performances are similar for every architecture with a recall/precision of 0.94/0.94. MKI-S obtains the lowest CER (0.14) although the localisation performances are equally good. This may indicates that the skip connections from the template branch help produce more generic features and better propagate gradients in the template branch. For the unseen ID types, the localisation is improved when the template is connected to the reconstruction stage.

To evaluate the capacity of the recognition, the recognition head was trained on the ground-truth segmented text fields. We obtain a CER of 0.33 for the unseen ID types which is higher than the MKI-S architecture (0.31). This shows that MKI-S is able to extract relevant features in the backbone for text recognition. In the next experiments, we use the MKI-S architecture.

5.2 Selection of the Knowledge Representation

We study the impact of knowledge representations defined in section 3.2 on the MKI-S architecture. Results are presented in Table 3. On the seen ID types, the multi-channel approach performs better on the localisation and present similar performances for text recognition. This representation seems to hold more information to help separate the fields. However, regarding the unseen ID types, the single channel presents a lower CER.

Taking into account the standard deviation, selecting the best representation is not obvious. The single-channel version reaches a lower CER and should be able to handle low representation of some field types. Thus, the single-channel representation is used to design our multitask model for knowledge integration.

Table 3: Comparison between the different knowledge representations on the TLR-UIDT task of MIDV2020 for MKI-S{att}.

	Seen ID types		Unseen ID types	
Template	single	multi	single	multi
cer ↓	**0.14** ± 0.02	0.15 ± 0.03	**0.31** ± 0.07	0.35 ± 0.11
recall ↑	0.94 ± 0.02	**0.96** ± 0.01	0.84 ± 0.08	**0.88** ± 0.10
precision ↑	0.94 ± 0.01	**0.96** ± 0.02	**0.87** ± 0.09	0.86 ± 0.11
IoU ↑	0.87 ± 0.01	**0.89** ± 0.02	**0.79** ± 0.05	**0.79** ± 0.05

5.3 Comparison against existing methods

We now compare our method to three existing systems presented in Section 2 on the MIDV2020 dataset (Table 4). As the source code is not available for some of them, we have re-implemented the state-of-the art methods. The **Chained** system is inspired from the sequential approaches of Mustafina et al. [14] and Van et al [16]. To make the comparison fair, we use our backbone for the text field localisation; in a second step, our CRNN, used in the recognition head, is applied on the detected text line images for text recognition. We call **Adapted-[6]** the adaptation of Carbonell et al. [6]: it refers to our multitask network without knowledge integration. **Similar-[1]** is our implementation of [1]. This system directly applies a large template on the image as a crop mask. We extend the margins of the template of TLR-UIDT task to match their experimental conditions. We then apply our CRNN for text recognition.

Table 4 shows that Similar-[1] suffers from poor localisation performances that heavily impact the recognition. Indeed, this system is designed for IDs without deformations. This shows that the simple application of a template is not sufficient to perform text localisation and recognition.

Regarding the seen IDs types, MKI-S significantly outperforms the other approaches in localisation and presents a better recognition. The localisation performances of the **Chained** and **Adapted-[6]** are equivalent but the last one is better than the **Chained** system for text recognition. This shows that the text recognition head benefits from shared features in the backbone. The gap between our MKI-S model and the other ones is even more significant on the unseen IDs types. The Chained and Adapted-[6] localisation scores drop by around 20 points of percentage which also impact the text recognition. By training with the templates, our MKI-S model is able to adapt to new templates.

5.4 Generalisation to the private dataset

Finally, we perform the same comparison as in section 5.3 on the industrial private database (Table 5). This database contains real world samples and a higher number of documents and types. Thus, those results give a better overview of the performance of each system in the real world context.

On the seen ID types, the MKI-S architecture reaches better performances than on the MIDV-2020 dataset with a recall/precision of 0.98/0.97 and CER

Table 4: Comparison of methods on the TLR-UIDT task of MIDV2020. The line "template" indicates whether the template is used.

		Chained	Adapted-[6]	Similar-[1]	MKI-S (ours)
	template	-	-	✓	✓
Seen ID types	cer ↓	0.21 ± 0.04	0.18 ± 0.02	0.98 ± 0.04	$\mathbf{0.14} \pm 0.02$
	recall ↑	0.93 ± 0.01	0.93 ± 0.01	0.40 ± 0.03	$\mathbf{0.94} \pm 0.02$
	precision ↑	$\mathbf{0.94} \pm 0.01$	0.93 ± 0.01	0.41 ± 0.03	$\mathbf{0.94} \pm 0.01$
	IoU ↑	0.86 ± 0.01	0.86 ± 0.01	0.57 ± 0.01	$\mathbf{0.87} \pm 0.01$
Unseen ID types	cer ↓	0.70 ± 0.18	0.72 ± 0.14	1.01 ± 0.04	$\mathbf{0.31} \pm 0.07$
	recall ↑	0.72 ± 0.14	0.75 ± 0.06	0.41 ± 0.14	$\mathbf{0.84} \pm 0.08$
	precision ↑	0.67 ± 0.17	0.73 ± 0.11	0.42 ± 0.14	$\mathbf{0.87} \pm 0.09$
	IoU ↑	0.63 ± 0.11	0.69 ± 0.08	0.57 ± 0.07	$\mathbf{0.79} \pm 0.05$

Table 5: Evaluation of our system on the private dataset.

	Seen ID types				Unseen ID types			
	Chained	Adapted-[6]	Similar-[1]	MKI-S (ours)	Chained	Adapted-[6]	Similar-[1]	MKI-S (ours)
cer ↓	0.17	0.24	1.23	**0.03**	1.15	0.87	1.22	**0.17**
recall ↑	0.88	0.89	0.05	**0.98**	0.36	0.38	0.08	**0.79**
precision ↑	0.86	0.88	0.05	**0.97**	0.30	0.42	0.08	**0.81**
IoU ↑	0.77	0.83	0.42	**0.91**	0.39	0.41	0.40	**0.76**

of 0.03. Despite the dataset being harder, our method takes advantage of the higher number of training samples. This shows how important is the template integration even on seen ID types. On the unseen ID types, MKI-S achieves the best results and the gap with the other methods is even greater. MKI-S heavily benefits from the integration of the template to deal with new ID types.

6 Conclusion and perspectives

In this paper, we address the task of text localisation and recognition in the context of unseen ID types, i.e. when a type has not been used for training. The template of ID types, describing the field structure, is always known even on new types. We leverage this knowledge by integrating it inside the neural network. We define the core concepts of knowledge integration in neural networks and propose template self-attention to guide the predictions in the context of unseen ID types with no annotated data. We compare three different architectures to leverage template self-attention and integrate the template. We show that the use of an auxiliary branch is crucial. Besides, skip connections from the template branch helps to compute relevant features and to generalize better. To evaluate our methods, we design a new dataset for the MIDV2020 database as well as a new task to perform recognition on unseen IDs types and make them public. We show that our approach benefits from IDs template integration to achieve better results than state-of-the-art methods on the new dataset as well as on a private industrial database composed of real world samples. Contrary to state-of-the-art works, our approach is able to deal with new ID types. Going ahead,

we plan to improve the unseen IDs types analysis by integrating the template into Transformer-based architectures via attention and querry conditionning.

References

1. Filippo Attivissimo, Nicola Giaquinto, Marco Scarpetta, and Maurizio Spadavecchia. An automatic reader of identity documents. In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 3525–3530, 2019.
2. Théodore Bluche. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. *Advances in neural information processing systems*, 29, 2016.
3. KB Bulatov, PV Bezmaternykh, DP Nikolaev, and VV Arlazarov. Towards a unified framework for identity documents analysis and recognition. *Computer Optics*, 46(3):436–454, 2022.
4. Konstantin Bulatov, Vladimir V Arlazarov, Timofey Chernov, Oleg Slavin, and Dmitry Nikolaev. Smart idreader: Document recognition in video stream. In *ICDAR*, volume 6, pages 39–44. IEEE, 2017.
5. Konstantin B. Bulatov, Ekaterina Emelianova, and Daniil V. Tropin et al. MIDV-2020: A comprehensive benchmark dataset for identity document analysis. *CoRR*, abs/2107.00396, 2021.
6. Manuel Carbonell, Alicia Fornés, Mauricio Villegas, and Josep Lladós. A neural model for text localization, transcription and named entity recognition in full pages. *Pattern Recognition Letters*, 136:219–227, 2020.
7. Denis Coquenet, Clément Chatelain, and Thierry Paquet. Span: a simple predict & align network for handwritten paragraph recognition. In *ICDAR*, 2021.
8. Denis Coquenet, Clément Chatelain, and Thierry Paquet. End-to-end handwritten paragraph text recognition using a vertical attention network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):508–524, 2022.
9. Denis Coquenet, Clément Chatelain, and Thierry Paquet. Dan: a segmentation-free document attention network for handwritten document recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
10. Bertrand Coüasnon. Dmos, a generic document recognition method: Application to table structure analysis in a general and in a specific way. *IJDAR*, 2006.
11. Vincent Poulain d'Andecy, Emmanuel Hartmann, and Marçal Rusinol. Field extraction by hybrid incremental and a-priori structural templates. In *DAS Workshop*, pages 251–256. IEEE, 2018.
12. Camille Guerry, Bertrand Coüasnon, and Aurélie Lemaitre. Combination of deep learning and syntactical approaches for the interpretation of interactions between text-lines and tabular structures in handwritten documents. In *ICDAR*, 2019.
13. Kaisar Kushibar, Sergi Valverde, and Sandra Gonzalez-Villa *et al.* Automated subcortical brain structure segmentation combining spatial and deep convolutional features. *Medical image analysis*, 48:177–186, 2018.
14. Venera Mustafina and Sergey Ivanov. Identity document recognition: Neural network approach. In *Intern. Russian Automation Conference*, pages 806–811, 2021.
15. Mohammad Reza Sarshogh and Keegan Hines. A multi-task network for localization and recognition of text in images. In *ICDAR*, pages 494–501, 2019.
16. Duc Phan Van Hoai, Huu-Thanh Duong, and Vinh Truong Hoang. Text recognition for vietnamese identity card based on deep features network. *IJDAR*, 2021.
17. Mohamed Yousef and Tom E Bishop. Origaminet: weakly-supervised, segmentation free, one-step, full page text recognition by learning to unfold. In *CVPR*, 2020.