Precise Segmentation for Children Handwriting Analysis by Combining Multiple Deep Models with Online Knowledge

Simon Corbillé[1(✉)], Éric Anquetil[2], and Élisa Fromont[1]

[1] Univ Rennes, IRISA, 35000 Rennes, France
{simon.corbille,elisa.fromont}@irisa.fr
[2] INSA Rennes, IRISA, 35000 Rennes, France
eric.anquetil@irisa.fr

Abstract. We present a strategy, called Seq2Seg, to reach both precise and accurate recognition and segmentation for children handwritten words. Reaching such high performance for both tasks is necessary to give personalized feedback to children who are learning how to write. The first contribution is to combine the predictions of an accurate Seq2Seq model with the predictions of a R-CNN object detector. The second one is to refine the bounding box predictions provided by the detector with a segmentation lattice computed from the online signal. An ablation study shows that both contributions are relevant, and their combination is efficient enough for immediate feedback and achieves state of the art results even compared to more informed systems.

Keywords: Handwriting Recognition and Segmentation · R-CNN object detector · Seq2Seq network · French Children Handwriting

1 Introduction

The paradox of Sayre [1] is a famous problem in the handwriting recognition domain. This dilemma claims that a handwritten word cannot be recognized without being segmented in letters and at the same time cannot be segmented in letters without the word being recognized. To tackle the handwriting recognition task, the systems use an analytic or a holistic approach. The analytic approach segments the handwriting and tries to recognize letters, while the holistic approach tries to recognize the whole word without explicit segmentation. State-of-the-art methods use holistic approaches based on deep learning model. They are designed for recognition only and are efficient in solving this task. However, in a context of learning spelling, the letter segmentation provided by these approaches is not precise enough to provide a useful spatial feedback on spelling mistake to a user.

We aim at designing a support system for **learning cursive handwriting at school** and more particularly in a dictation context. Tackling both the challenges of **recognition and segmentation of children handwriting** may

G. A. Fink et al. (Eds.): ICDAR 2023, LNCS 14190, pp. 236–252, 2023.
https://doi.org/10.1007/978-3-031-41685-9_15

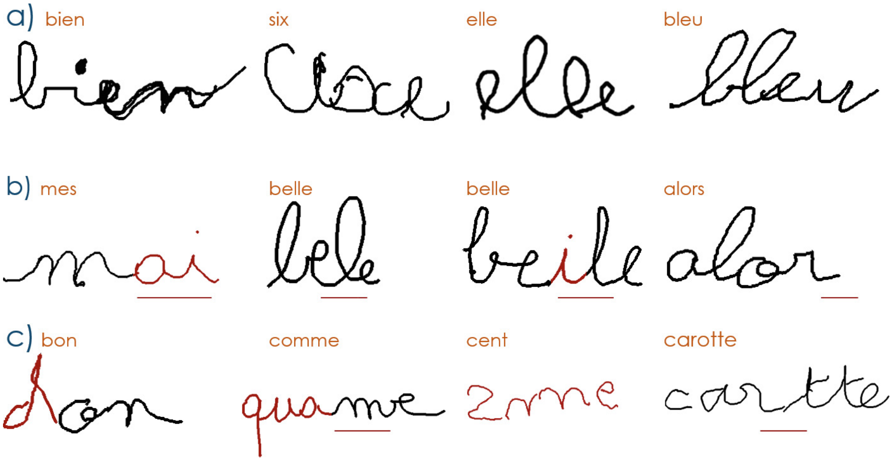

Fig. 1. Examples of cursive children handwriting. The oral French instruction given to the children is provided in *orange* and examples of feedback are drawn in *red*. Line *a* shows some degraded handwriting, line *b*, phonetic errors and line *c* shows other types of errors in a context of learning spelling. (Color figure online)

allow a system to provide a fine-grained analysis on the handwritten words and to **deliver immediate spelling feedback** to children. The children are in a learning process and therefore, their handwriting is degraded and contains misspelling errors. Line *a* of Fig. 1 illustrates several examples of degraded handwriting. We can see that a distortion of the letter "e" can be interpreted as a letter "l" and vice versa for the word "elle" in the third position of this line. Line *b* shows several examples of phonetic errors. In the first example, where the instruction is "mes" [mɛ], the child writes "mai" [me], which sounds very similar in French. These homophonic errors can be anticipated in automatic systems using a language model that would take into account the contextual information. However, other types of errors in a context of learning spelling illustrated in line *c* such as dyslexia and out-of-vocabulary words cannot. The first example of line *c* shows a common mistake in French where the child confuses the letters "b" and "d" which are phonetically close.

To provide an accurate recognition and segmentation of the children handwritten words, we propose the two following contributions included in the Seq2Seg system:

- We present an original combination strategy using a model dedicated to recognition and an object detector dedicated to segmentation. The recognition model is used to recognize a word and to select the segmentation predictions of the object detector corresponding to the letters of the recognized word.
- We use an segmentation lattice [2–4] which encodes expert knowledge to refine the letter segmentation provided by the object detector and thus improve the precision of the segmentation.

This paper is organized as follows. The related works are described in Sect. 2. The contributions are detailed in Sect. 3. Section 4 presents an ablation study of our approach and compares it with the state of the art. We conclude in Sect. 5.

2 Related Work

This section presents works related to the **recognition and segmentation** of handwritten words. The first part introduces the latest methods in the handwriting recognition domain. The second part sets out the limits of these methods for a segmentation task and presents the state of the art in children handwriting recognition and segmentation. Finally, we provide a brief presentation of object detection models to show their relevance in an handwriting segmentation context.

2.1 Handwriting Recognition

The state of the art in handwriting recognition is achieved by the **Sequence-to-Sequence** (Seq2Seq) [5,6] and Transformer [7,8] networks. Seq2Seq use an encoder-decoder paradigm enhanced by an attention mechanism, while Transformers are based on a feature extractor followed by multi-head attention mechanisms. Transformers are slightly more accurate but need much more data to be optimized than Seq2Seq. This is often dealt with data generation and data augmentation techniques.

In our work, a rather small dataset is available compare to adult handwriting ones than can be found in [9,10] due to the cost associated with data collection in schools and degraded handwriting annotations. We thus decided to rely upon a Seq2Seq network for word recognition because of its good compromise accuracy/need of labeled data.

2.2 Handwriting Segmentation

To our knowledge, there is no (reasonable sized) public dataset for handwriting (semantic) segmentation, *i.e.,* handwriting words with annotations at the letter pixel level. This task is particularly tedious and time-consuming but is not necessary, nowadays, to achieve excellent recognition results for the architectures mentioned above. For this reason, handwriting letter segmentation methods are difficult to compare quantitatively. For the networks designed for handwriting recognition, the letter segmentation can be computed from the position of the receptive field associated to the letter prediction. The width and height of the receptive field being fixed, this approach, which lacks flexibility, does not provide a precise segmentation. Furthermore, most networks are trained with the connectionist temporal classification (CTC) [11] approach. CTC manages the alignment between an input data sequence and an output sequence of frames of variable size. CTC is known to have a peaky behavior [12] *i.e.,* it predict one frame per letter. This impacts the segmentation performance since a frame has a fixed size while a handwritten letter has a variable one. In [13,14], the authors modified CTC to enforce a better alignment between the frames and the real letters. However, despite these efforts, the segmentation was still lacking precision.

The authors of [4] and its extensions [15,16] use an analytic approach to reach the state of the art in children handwriting recognition and segmentation. The **letter recognition** is made from letter splitting hypotheses coming from a **segmentation lattice** [2–4]. Then, the method selects the best path of the lattice where his associated word is closest to the instruction or to a phonetically close word. However, this system uses a **language model** to guide the analysis of children handwriting using assumptions of probable phonetic errors. It is thus specific and dedicated to the French language and cannot be easily adapted to other languages. Moreover, as already shown in the Introduction in Fig. 1, some children errors cannot be prevented using a language model. In this work, we want to achieve results on par with [15] without relying on a language model.

2.3 Object Detection

We rely on an existing, very successful, two-stage deep learning-based object detector [17] to perform a precise localization of the letters in the handwritten words. **Two-stage detectors** [17–19] are known to be a little bit more precise for localization than their one-stage counterparts [20–23] even though they are usually slower. Object detectors provide a joint classification of objects into classes and a regression of the bounding boxes that best localize each object in an image (or in a video frame). In a two-stage detector, candidate regions are generated by a RPN (region-proposal network) and processed to perform the detection task.

Based on the output of the object detector (*i.e.* labeled bounding boxes), the final segmentation is obtained using all the handwriting pixels within the predicted bounding box. Note that in our work, the image of the handwriting comes from an online signal, therefore, this image is noise-free both on the background and on the handwriting pixels. This makes it possible to extract the letter segmentation from the bounding box coordinates. Also note that we could have used the semantic segmentation output of an instance-based semantic segmentation network such as Mask-RCNN [17] to directly segment the letters. However, the complexity in terms of parameters of such segmentation networks (the semantic segmentation part of the network is usually independent from the object detection part), and the limited number of realistic labeled children words to train it, made bounding boxes of traditional object detectors better candidates to tackle our segmentation problem when the segmentation target is too ambiguous.

3 Methods

We propose Seq2Seg, a method to combine two deep learning models (Seq2Seq and R-CNN) and expert online knowledge to accurately segment and recognize children handwritten words. Seq2Seg, illustrated in Fig. 2 leverages each method to provide a precise semantic segmentation of the children words. The first level (**Level A**) uses a model dedicated to the recognition task as an oracle to filter out the bounding box's predictions of the object detector. **Level B** uses an expert segmentation lattice [2–4] to refine the letter segmentation associated to the bounding boxes predicted by the object detector. The segmentation lattice use online data, while the object detector and the recognizer use online data converted to offline.

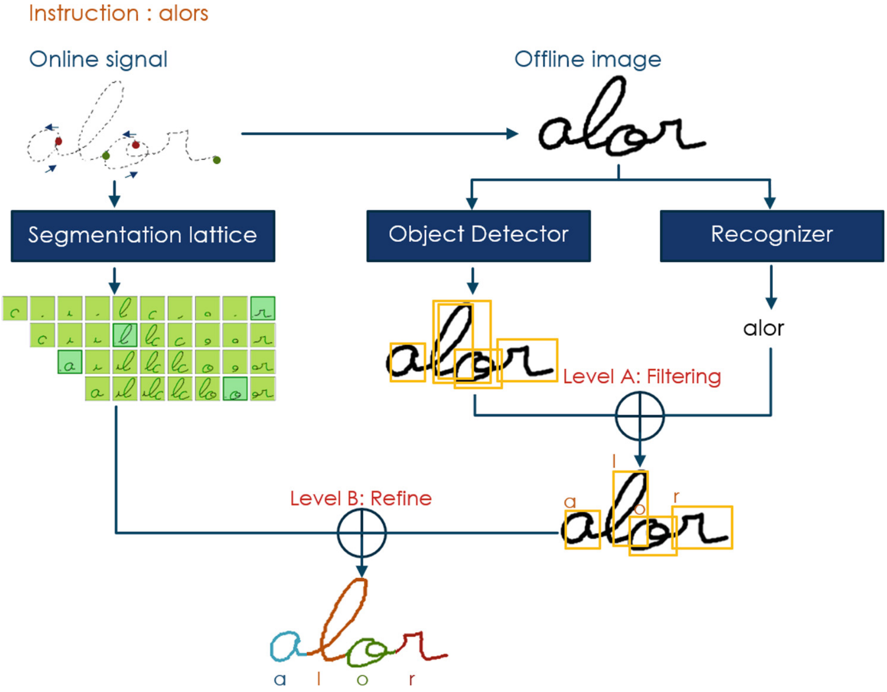

Fig. 2. Summary of levels A and B contributions.

In our work, we use the Seq2Seq architecture defined in [15] as the text recognition model and the R-CNN architecture defined in [17] as the object detector. The Seq2Seq performs well on the recognition task but provides an imprecise segmentation while R-CNN performs well on the segmentation task but is less accurate in recognition than the Seq2Seq (see Table 2 in Sect. 4 for the detailed results).

3.1 Level A: Filtering Bounding Boxes Predictions with an Accurate Recognition Model

The recognition model is trained solely on the recognition task and outputs a word. From this word, one can deduce, in particular, the number of letters to be segmented. This information makes it possible to select a fixed number of object detector segmentation predictions **during the inference** and use the more accurate recognition result of the recognition model. The process of selecting the predictions from the object detector can be difficult and ambiguous in certain cases, as illustrated in Fig. 3: *e.g.*, in a letter “m”, two letters “n” can be recognized but they cannot be both true at the same time so here, a more global view is necessary to choose the right segmentation. The use of the precise recognition model, providing that it does not introduce other errors, makes it possible to remove these ambiguities.

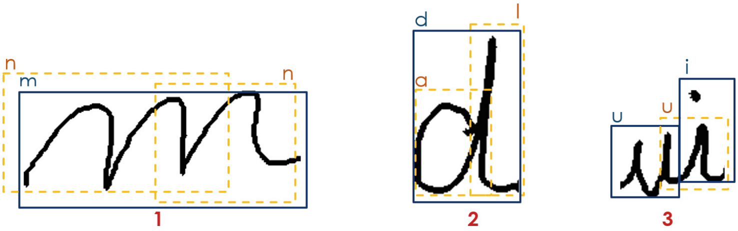

Fig. 3. Examples of ambiguity in object detector predictions: the correct prediction is in full line and the wrong ones in dash.

The object detector has several output x-ordered predictions. The goal is to select the object detector prediction corresponding to the letter segmentation. This method illustrated in Fig. 4 is broken down into three steps: (Step 1) we **compute all object detector prediction sequences**; (Step 2) we **filter the sequences according to the length of the word recognized by the recognition model**; (Step 3) we **compute the score associated with each sequence**. The final selected sequence is the one with the highest score. R-CNN natively includes two Non-Maximum Suppression (NMS) phases to filter out its predictions. The first is applied to the regions proposals to reduce the number of proposals to consider, while the second is applied to predictions (bounding boxes and labels) to keep the best prediction for the objects predictions with the same label. In a letter-in-word detection context, there is little overlap between letters unlike a classic COCO-style object detection. In order to handle the cases where several letter predictions are nested as emphasized in Fig. 3, we have added an NMS on the predictions of the model which is **independent of the class**. Our method uses the predictions **before** the last NMS to have a wide variety of prediction to filter with segmentation ambiguities. The purpose of the method is to remove these ambiguities.

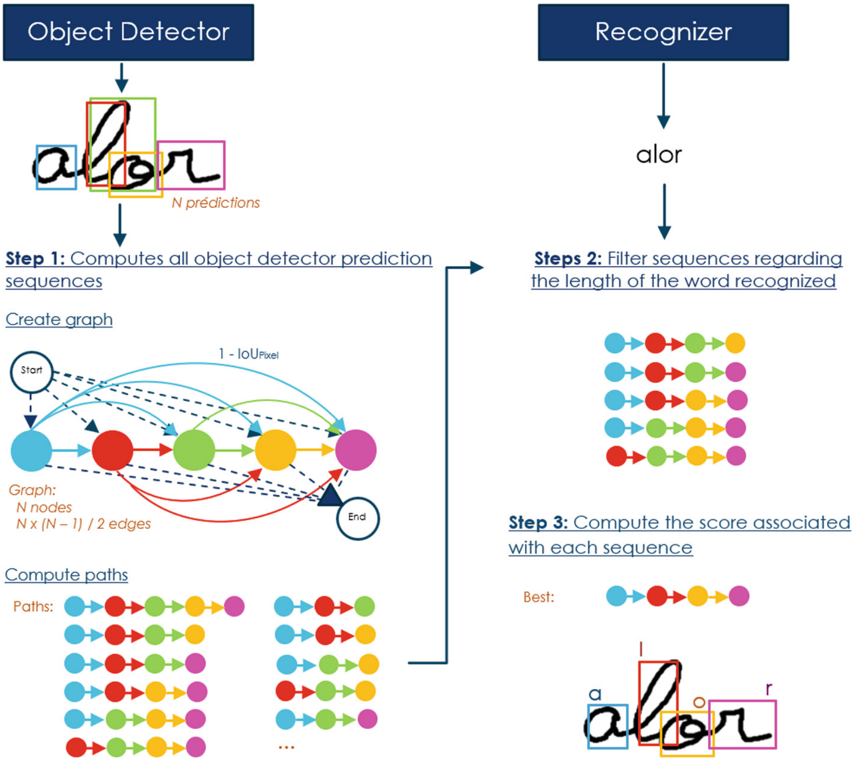
Object Detector
N prédictions
Step 1: Computes all object detector prediction sequences
Create graph
1 - IoUPixel
Start
End
Graph:
N nodes
N x (N – 1) / 2 edges
Compute paths
Paths:
...
Recognizer
alor
Steps 2: Filter sequences regarding the length of the word recognized
Step 3: Compute the score associated with each sequence
Best:
a
l
o
r

Fig. 4. Level A: Example the three steps of the process of filtering object detector predictions with the result of a recognition model.

(1) Compute All the Prediction Sequences: consider a directed graph $G(V, E)$, where V and E correspond to the sets of vertices and edges. For each prediction of the object detector ordered by x_{min} from the bounding box coordinates, a vertex is added in G as illustrated in Fig. 4. The weight of an edge $e_{ij} = (v_i, v_j) \in E$ is computed as $e_{ij} = 1 - \text{IoU}_{Pixel}$ between the predictions ordered by x_{min} associated to the vertices. IoU_{Pixel} stands for the Intersection Over Union of the handwriting pixels contained in the two bounding boxes corresponding to the two vertices: the predictions with the higher overlap have a weaker link. A sequence of predictions ordered by x_{min} corresponds to a graph path, *i.e.* a list of connected vertices in the graph.

(2) Filter the Sequences According to the Length of the Word Predicted by the Recognition Model: there are three selection scenarios (Table 3 in Sect. 4 details the result of each type of scenario):

- **Perfect matching:** The number of predicted letters of the object detector and of the recognition model is equal. In this case, we expect our filtering to only improve the recognition part of the object detector (that we do not use explicitly).

- **Matching:** The number of predicted letters of the object detector and of the recognition model is different but there is at least one possible matching in the solutions. In this case, we expect that the use of the word classifier as an oracle will help to remove some ambiguities for the object detector. This may improve both the recognition and the segmentation.
- **No matching:** The number of predicted letters of the object detector and of the recognition model is different and there is no possible matching in the predictors' solutions. In this case, both the classification and the segmentation of the object detector are used (the Seq2Seq is ignored). In practice, in this case, we noticed that the Seq2seq was either predicting an additional letter or was missing one. It is thus important for the object detector to be able to ignore the oracle prediction when there is a strong conflict between both models. This filtering might thus improve the overall recognition results since the object detector will take over the Seq2Seq but only for the most difficult predictions.

(3) Compute the Score Associated with Each Sequence: the score of a sequence of size N_a takes into account the degree of overlap between all the bounding boxes involved in the sequence. In particular, it minimizes the inter-letter overlap and also includes a coverage criterion to ensure a good coverage of the entire handwritten text. The overlapping score, $s_{overlap}$, is the product of all edge weights *weight v* in the path of the graph $G(V, E)$ corresponding to a sequence:

$$s_{overlap} = \Pi_{i=1}^{N_a} weight\ v_i \tag{1}$$

The larger the overlap, the lower the score is. On the contrary to classic COCO-style object detection contexts [24], in the handwriting context, there is almost no overlap between objects to detect except for the ligature area between the letters. To compute the coverage score and to count each pixel only once, we add the number of pixels contained in each prediction and the number of pixels contained in the intersection of the two predictions is subtracted from the number of pixels contained in each prediction. Then, the predicted number of pixels is divided by the total number of pixels as follows:

$$s_{cover} = (\Sigma_{i=1}^{N_a} N_p\ pred_i - \Sigma_{i=2}^{N_a} N_p\ inter(pred_{i-1}, pred_i))/N_p\ total \tag{2}$$

The final $s_{alignment}$ score is defined as:

$$s_{alignement} = s_{overlap} + s_{cover} \tag{3}$$

The output of the Seq2Seg model is the semantic segmentation computed from the bounding boxes of the sequence with the highest $s_{alignment}$ score together with the predictions of the Seq2Seq model for each letter. We can note that the efficiency of this method in terms of computation time depends on the size of the generated graphs. In our children handwriting context, the graphs associated with the words are small because the words are smaller than 10 letters. The computation time of this method is therefore low enough to provide immediate feedback.

3.2 Level B: Use of a Segmentation Lattice Based on Online Handwriting

The **online handwriting** can be split *a priori* into different segmentation hypotheses grouped in a **segmentation lattice** using heuristics (and without letter recognition). This process is detailed in [2] and consolidated in [4]. Furthermore, the online signal makes it possible to obtain a first automatic **semantic segmentation** for each hypothesis where two classes are considered: background and handwriting. Our goal is to use this lattice to find the "nearest" hypotheses of the segmentation lattice associated to the bounding boxes predicted by the object detector as illustrated in Fig. 5.

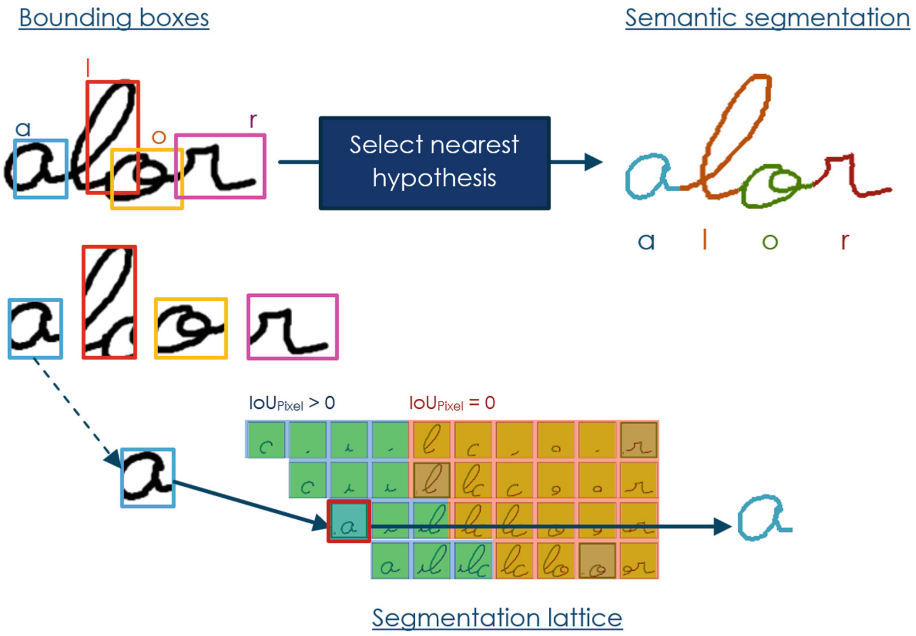

Fig. 5. Example of bounding boxes refinement with the online segmentation lattice. IoU_{Pixel} is used to select the best lattice hypothesis.

The similarity between the lattice nodes and the bounding boxes is computed with an IoU_{Pixel} *i.e.* an Intersection-over-Union between the handwriting pixels contained in a bounding box (easily accessible as explained before) and the ones in a node of the segmentation lattice. By associating the hypotheses of the lattice with the bounding boxes, this method **refines the coordinates of the**

bounding boxes and thus increase the precision of the segmentation of the object detector. Moreover, this approach also provides a better segmentation for slant handwriting than the "bounding box to segmentation" trivial correspondence proposed in Sect. 2.3. This is illustrated in Fig. 5 for the letter "l" and "o".

4 Experiments

4.1 Dataset

Children cursive handwriting is acquired on pen-based tablet at schools. The **French** handwritten words are acquired as an **online signal** encoded by a sequence of points represented by two coordinates (x, y), a pressure and a timestamp. The online signal is used to compute the segmentation lattice presented in the previous section and is converted into an image with linked points and a thickness of 2 on the links. The input images are padded (x axis) and resized (y axis) at $1\,280 \times 128$ pixels to fit the used deep learning models. Table 1 details the datasets used to train/test the deep learning models (which are all variants of the acquired children words). The original dataset is composed of 8 054 French handwritten cursive words annotated at the word level that are useful to train the Seq2Seq network. Besides, 2 126 words are annotated (*i.e.* segmented) at the letter level to train the R-CNN object detector. The number of letter-annotated data being limited, we redefined the splits compared to [15] and between the two models, to better train the object detector. The children writers are different for the training and the testing and the test set is the same for all models. Due to GDPR restrictions on children's private data, this dataset is not public.

Table 1. Details on the data used to train the deep learning models.

Models	Annotation type	Training	Validation	Test	Total
Seq2Seq	Words	6 022	1 000	1 032	8 054
R-CNN	Letters	918	176	1 032	2 126
R-CNN with synthesis	Letters	27 540	176	1 032	28 748

To better train the R-CNN model (that is data greedy), we perform data augmentation only on the training set (called "with synthesis" in the table). Among a list of usual offline deformations (stretching, slant) and more recent ones (stroke stretching, curvature [25]), each word is augmented 30 times with random parameters.

4.2 Implementation and Evaluation Metrics

Implementation: The Seq2Seq model follows the same architecture and training protocol as in [15]. The model performs poorly with only children handwriting dataset and to our knowledge, there is no other children handwriting dataset available. Therefore the model is pre-trained on an adult handwriting dataset [10] and then fine-tuned on the children handwriting dataset. The model is trained during 200 epochs with a batch size of 16. The RMS prop optimizer is used with a learning rate of 0.001. Since the test set is different, we reevaluate the method from [15] on our dataset. The R-CNN with a ResNet-FPN backbone is trained during 60 epochs with a batch size of 4. The AdamW [26] optimizer is used with a learning rate of 0.0001. The R-CNN parameters are indicated in the original article [17] except that we ignore the Mask branch, we use the Complete Intersection Over Union (CIoU) [27] criterion to match the ground truths and the predictions during the training phase. We also add an NMS filtering independent of the class on the outputs to handle nested predictions as explained in Sect. 3.1. We set all parameters of the R-CNN model on the validation set using the Mean Average Precision (MAP) performance score before evaluating the best model on the test set.

Metrics: To evaluate the performance of our Seq2Seg approach, we use the usual Character Error Rate (CER) and Word Error Rate (WER) with a Damerau Leveinshtein [28] distance for the recognition performance. We use Intersection Over Union (IoU) and IoU at pixel level to evaluate the segmentation performance. As explained before, the IoU$_{Pixel}$ focuses on the handwriting lines and ignores the (mostly white) background.

4.3 Quantitative Results

We first perform an ablation study to measure the impact of our different contributions as well as the choice of features extractor backbone in object detector. Then, we compare our approach to the state-of-the-art models on our data. All experiments are evaluated in terms of **recognition, segmentation and computing speed** on the test set. While the networks are trained on GPU, the processing time is computed on a laptop with an Intel Core i7-8665U CPU. Indeed, education applications are run on a pen-based tablet, where an internet connection is not always available. Therefore, timing analysis is more relevant on a CPU-equipped laptop. We consider acceptable an analysis time lower than 2 s to deliver immediate feedback to the children.

Table 2 shows the results of the ablation study, where Level A corresponds to the filtering method of the object detector predictions with the result of a recognition model presented in Sect. 3.1 and Level B corresponding to the refinement of the bounding boxes coordinates of the object detector with a segmentation lattice presented in Sect. 3.2. We denote Seq2Seq as the result of the encoder part and use only the encoder result in this work, as recommended in [15]. We can see in the table that the choice of a deeper backbone (we tried 18 to 101

Table 2. Ablation study of the object detector backbone and impact of our contributions. Recognition is evaluated with Character Error Rate (CER) and Word Error Rate (WER) (lower values are better). Segmentation is evaluated with Intersection Over Union (IoU) and IoU$_{Pixel}$ (higher values are better). The *Average time* is the averaged number of seconds for a method to analyze a word.

		Recognition		Segmentation		Time
Method	Backbone	CER (%)	WER (%)	IoU (%)	IoU$_{Pixel}$ (%)	Average Time (s)
Seq2Seq [15]		**5.3**	**19.4**	48.4	59.4	0.12
R-CNN [17]	ResNet-18 FPN	12.0	36.4	78.6	80.3	1.25
	ResNet-34 FPN	12.2	37.7	79.6	81.3	1.29
	ResNet-50 FPN	11.4	34.7	80.5	81.5	1.61
	ResNet-101 FPN	10.7	34.2	81.0	82.2	2.17
Level A	ResNet-18 FPN	5.2	19.0	81.7	83.6	1.43
	ResNet-34 FPN	**5.0**	**18.6**	82.3	84.0	1.47
	ResNet-50 FPN	5.2	18.9	82.8	84.0	1.79
	ResNet-101 FPN	5.1	19.0	82.0	83.5	2.35
Level B	ResNet-18 FPN	12.0	36.4	82.6	85.0	1.40
	ResNet-34 FPN	12.2	37.7	83.3	85.9	1.44
	ResNet-50 FPN	11.4	34.7	83.8	86.3	1.77
	ResNet-101 FPN	10.7	34.2	84.4	87.1	2.32
Seq2Seg: Levels A + B	ResNet-18 FPN	**5.2**	**19.0**	**85.9**	**88.3**	**1.58**
	ResNet-34 FPN	**5.0**	**18.6**	**86.1**	**88.9**	1.62
	ResNet-50 FPN	**5.2**	**18.9**	**86.3**	**89.0**	1.95
	ResNet-101 FPN	**5.1**	**19.0**	**85.6**	**88.4**	2.50

layers) in the object detector (R-CNN) improves the performance in recognition and segmentation (-1.3% of CER from ResNet 18 to ResNet 101; +1.9% of IoU$_{Pixel}$ from 18 to 101 layers). On the other hand, the computing time increases of more than 2 s. The Seq2Seq model remains much more accurate in recognition (CER/WER) than all versions of the R-CNN. The choice of the backbone had no significant impact on our contribution (see bottom part of the table). We chose the backbone ResNet-34 FPN for the next experiments due to its speed and slightly better performance in recognition.

Level A: filtering the object detector's predictions with the results of the Seq2Seq allows us to obtain slightly better results in recognition (CER of 5%) than the Seq2Seq alone (CER of 5.3%). The reasons for this are given in the"no matching case" of the second step of the first contribution presented in Sect. 3.1. Furthermore, this method selects the bounding boxes to maximize the coverage and minimize the overlap of the handwriting and thus improves the segmentation of the object detector. Table 3 details the different scenarios of filtering and their contributions to the performance compared to the object detector and the Seq2Seq performance alone:

- In the scenario where the number of predictions of the two models is equal, the Level A improves only the recognition performance as expected. This scenario concerns most of the words.
- In the scenario where the number of predictions is different and a matching exist, the gain is the highest. Indeed, the strategy makes it possible to filter the bad predictions of the object detector.
- For a few words, nothing is filtered out and thus this contribution does not improve the object detector performance. In practice, this corresponds to words for which the recognizer makes more mistakes than the object detector.

Table 3. Number of words by scenario of filtering between R-CNN and Seq2Seq. Performance of models alone and level A contribution. R-CNN uses ResNet34-FPN backbone.

		R-CNN		Seq2Seq		Level A	
Filtering type	#Words	CER (%)	IoU (%)	CER (%)	IoU (%)	CER (%)	IoU (%)
Perfect Matching	857	8.0	**85.6**	**3.8**	50.2	**3.8**	**85.6**
Matching	164	34.7	47.9	**11.5**	40.0	**11.5**	**64.6**
No Matching	11	**1.8**	**87.2**	36.6	31.3	**1.8**	**87.2**

Level B: refining the bounding boxes coordinates by the use of a segmentation lattice improves the R-CNN segmentation performance for a small computing cost.

The results of the competitors are shown in Table 4. The best recognition and segmentation performance on our dataset are given by [15] with a small margin compared to Seq2Seg (+0.1% CER, +1.6% IoU$_{Pixel}$), a high computation cost (5,07s, +3,45s compared to Seq2Seg) and using a language model. To overcome this computation cost, the authors of [15] have proposed a pruning strategy (shown in the second line). This strategy degrades the recognition performance as well as the segmentation one which makes it **significantly lower than Seq2Seg for recognition and segmentation** (-2.6% CER, -4.3% WER, +1.3% IoU, +2.6% IoU$_{Pixel}$).

The following section presents a qualitative analysis of the results obtained and shows the limits associated with the children handwriting recognition and segmentation tasks.

4.4 Qualitative Results

This section presents a qualitative analysis of the results obtained by Seq2Seg. The goal is to visualize the effect of each contribution, *i.e.* the impact of Level A and Level B contributions. In these visualization examples, the output of the object detector corresponds to the **predictions before the last NMS** was performed independently of the class of the predicted bounding box.

Table 4. Comparison to state-of-the-art approaches. Recognition is evaluated with Character Error Rate (CER) and Word Error Rate (WER) (lower values are better). Segmentation is evaluated with Intersection Over Union (IoU) and IoU$_{Pixel}$ (higher values are better). The *Average time* is the averaged number of seconds for a method to analyze a word. LM stand for "Language Model".

		Recognition		Segmentation		Time
Method	LM	CER (%)	WER (%)	IoU (%)	IoU$_{Pixel}$ (%)	Average Time (s)
Fusion competition [15]	Yes	**4.9**	**16.1**	**89.2**	**90.5**	5.07
Fusion competition (pruning) [15]	Yes	7.6	22.9	84.8	86.3	**0.72**
Seq2Seg (Our)	No	**5.0**	18.6	86.1	**88.9**	**1.62**

Figure 6 emphasizes the relevance of Level A contribution. The filtering process by the recognition model selects the correct number of letters by minimizing the overlap and maximizing the coverage rate. Moreover, the use of the segmentation lattice in Level B contribution produces a precise segmentation of the handwriting words especially in example 1 where the bounding boxes of the letters “i” and “t” overlap.

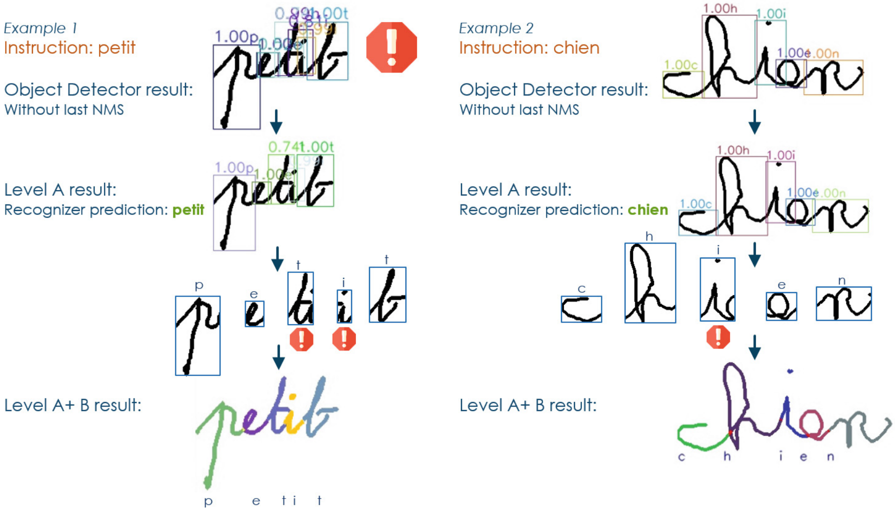

Fig. 6. Examples with an accurate recognition and a precise segmentation.

Figure 7 illustrates examples where the recognition model makes errors. In example 1, there is no matching between the prediction of the recognition model and the object detector. We can see that the Seq2Seq makes recognition errors and therefore its associated filtering would be wrong. In this case, the bounding boxes and labels predicted by the object detector are used and provide an accurate result in recognition and segmentation. Example 2 shows a case where the filtering by the recognition model leads to a segmentation error. In addition, we can note the omission of the drawing of the point of the “i” in example 2 which is quite common in a context of learning how to write.

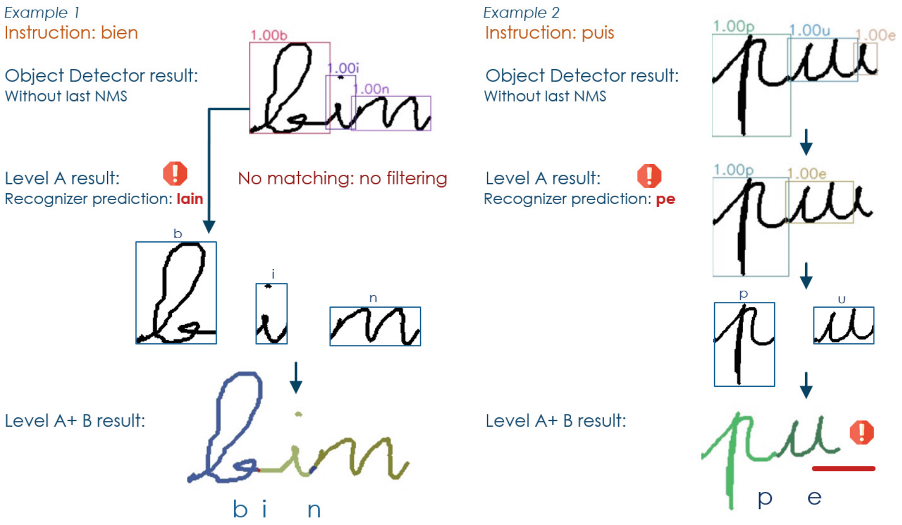

Fig. 7. Examples where the recognition model makes errors: in example 1, there is no matching between the recognition model and the object detector. In example 2 the filtering leads to an under-segmentation error.

Note that evaluating the quality of the handwriting segmentation with the currently used metrics is difficult. Indeed, it is not easy to define an absolute segmentation ground truth for some letters due to the ligature area between letters. Thus, a prediction can have an IoU lower than 100% with the ground truth while the associated segmentation is correct. Moreover, the ground truth class associated with a degraded letter can vary according to the annotator (confusion between the letter “e” and the letter “l”, “a” and “o” ...). Taking into account the uncertainty in the predictions might be helpful to know when a (human) teacher should take over the automated system to provide a more useful advice to the children.

5 Conclusion

We presented Seq2Seg, an original combination strategy which uses a model dedicated to recognition as an oracle to filter out the segmentation predictions of an object detector and then refines the segmentation using an expert segmentation lattice. Seq2Seg produces the best of both worlds: the accurate recognition of a Seq2Seq and the precise segmentation provided by an R-CNN object detector. Seq2Seg is efficient enough to provide immediate feedback to children learning how to write and it outperforms the state of the art results on this task without the use of a language model. This last point makes Seq2Seg much more flexible to other learning contexts. Our future work will focus on evaluating and improving the quality of the feedback in school contexts. In particular, we plan to better leverage the uncertainty of the decisions (both for the Seq2Seq and the object detector), for example by allowing the system to reject hypotheses, to prevent giving erroneous feedback to the children.

References

1. Sayre, K.M.: Machine recognition of handwritten words: a project report. Pattern Recognit. **5**(3), 213–228 (1973)
2. Anquetil, E., Lorette, G.: Perceptual model of handwriting drawing application to the handwriting segmentation problem. In: 4th International Conference Document Analysis and Recognition (ICDAR 1997), 2-Volume Set, 18–20 August 1997, Ulm, Germany, Proceedings, p. 112. IEEE Computer Society (1997)
3. Anquetil, E., Lorette, G.: On-line handwriting character recognition system based on hierarchical qualitative fuzzy modelling. In: Progress in Handwriting Recognition, pp. 109–116 (1997)
4. Simonnet, D., Girard, N., Anquetil, É., Renault, M., Thomas, S.: Evaluation of children cursive handwritten words for e-Education. Pattern Recogn. Lett. **121**, 133–139 (2019)
5. Michael, J., Labahn, R., Grüning, T., Zöllner, J.: Evaluating sequence-to-sequence models for handwritten text recognition. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, 20–25 September 2019, pp. 1286–1293. IEEE (2019)
6. Coquenet, D., Chatelain, C., Paquet, T.: End-to-end handwritten paragraph text recognition using a vertical attention network. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
7. Kang, L., Riba, P., Rusiñol, M., Fornés, A., Villegas, M.: Pay attention to what you read: Non-recurrent handwritten text-line recognition. Pattern Recognit. **129**, 108766 (2022)
8. Barrere, K., Soullard, Y., Lemaitre, A., Coüasnon, B.: Transformers for Historical Handwritten Text Recognition. In: Doctoral Consortium - ICDAR 2021, Lausanne, Switzerland (2021)
9. Marti, U.-V., Bunke, H.: A full English sentence database for off-line handwriting recognition. In: Fifth International Conference on Document Analysis and Recognition, ICDAR 1999, 20–22 September 1999, Bangalore, India, pp. 705–708. IEEE Computer Society (1999)
10. Liwicki, M., Bunke, H.: IAM-OnDB - an on-line English sentence database acquired from handwritten text on a whiteboard. In: Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), 29 August - 1 September 2005, Seoul, Korea, pages 956–961. IEEE Computer Society (2005)
11. Graves, A., Fernández, S., Gomez, F.J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, 25–29 June 2006, vol. 148 of ACM International Conference Proceeding Series, pp. 369–376. ACM (2006)

12. Zeyer, A., Schlüter, R., Ney, H.: Why does CTC result in peaky behavior? CoRR, abs/2105.14849 (2021)
13. Liu, H., Jin, S., Zhang, C.: Connectionist temporal classification with maximum entropy regularization. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., (edn.), Advances in Neural Information Processing Systems vol. 31, pp. 839–849 (2018)
14. Li, H., Wang, W.: Reinterpreting CTC training as iterative fitting. Pattern Recognit. **105**, 107392 (2020)
15. Krichen, O., Corbillé, S., Anquetil, E., et al.: Combination of explicit segmentation with Seq2Seq recognition for fine analysis of children handwriting. IJDAR 25, pp. 339–350 (2022). https://doi.org/10.1007/s10032-022-00409-4
16. Krichen, O., Corbillé, S., Anquetil, E., Girard, N., Nerdeux, P.: Online analysis of children handwritten words in dictation context. In: Barney Smith, E.H., Pal, U. (eds.) ICDAR 2021. LNCS, vol. 12916, pp. 125–140. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86198-8_10
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017, pp. 2980–2988. IEEE Computer Society (2017)
18. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., (edn.) Advances in Neural Information Processing Systems. vol. 28, pp. 91–99 (2015)
19. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra R-CNN: towards balanced learning for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 821–830. Computer Vision Foundation/IEEE (2019)
20. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016, pp. 779–788. IEEE Computer Society (2016)
21. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: Optimal speed and accuracy of object detection. CoRR, abs/2004.10934 (2020)
22. Li, C., et al.: YOLOv6: A single-stage object detection framework for industrial applications. CoRR, abs/2209.02976 (2022)
23. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. CoRR, abs/2207.02696 (2022)
24. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
25. Mouchère, H., Bayoudh, S., Anquetil, E., Miclet, L.: Synthetic on-line handwriting generation by distortions and analogy. In: 13th Conference of the International Graphonomics Society (IGS2007), pp. 10–13, Melbourne, Australia, November 2007
26. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019. OpenReview.net (2019)
27. Wang, X., Song, J.-Y.: ICIoU: improved loss based on complete intersection over union for bounding box regression. IEEE Access **9**, 105686–105695 (2021)
28. Damerau, F.: A technique for computer detection and correction of spelling errors. Commun. ACM **7**(3), 171–176 (1964)