International Journal on Document Analysis and Recognition (IJDAR) (2022) 25:415–427
https://doi.org/10.1007/s10032-022-00408-5

SPECIAL ISSUE PAPER

Textline alignment on the image domain

Boraq Madi[1] · **Ahmad Droby**[1] · **Jihad El-Sana**[1]

Received: 2 April 2022 / Revised: 15 June 2022 / Accepted: 5 August 2022 / Published online: 29 August 2022

Abstract
Editing and publishing a historical manuscript involves a research phase to recover the original manuscript and reconstruct the transmission of its text based on the relations between its surviving copies. *Manuscript alignment*, which aims to locate the shared and the different text among a set of copies of the same manuscript, is essential for this phase. In this paper, we present an alignment algorithm for historical handwritten documents that works directly on the image domain due to the absence of an accurate handwritten text recognition (HTR) system for handwritten historical documents and the necessity to visualize the original manuscripts in parallel to examine features beyond the transcribed text. Our approach extracts subwords, estimates the similarity among these subwords, and establishes an alignment among them. We extract subwords from textlines images and convert them into sequences of subword images. It estimates the similarity between two subwords using a Siamese network model and applies Longest Common Subsequence (LCS) to establish the alignment between two image sequences. We have implemented our algorithm, trained the Siamese model, and evaluate its performance using textline images from historical documents. Our algorithm outperformed the state-of-the-art by large margins. Unlike the state-of-the-art, the framework builds the alignment from scratch without requiring any prior knowledge concern subwords boundaries. In addition, we build a new dataset for textline alignment for historical documents, which include ten pairs of pages taken from two copies of two Arabic manuscripts and annotated at the subword level.

Keywords Alignment · YOLO · Subsequence · Historical documents

1 Introduction

Prior to the introduction of the printing press, written knowledge was spread by copying manuscript by hand. The development of paper production, writing tools, and copy houses led to the wide spread of writing, and millions of manuscripts were inscribed. The reproduction of manuscripts was done by handwriting, which often led to differences between original and copied manuscripts. Most of these changes were done intentionally, but some were done accidentally. These changes include introducing, removing, or altering words or phrases. Often phrases and words in the copied manuscripts were adjusted to geographical regions or era, as languages evolve over time. Many copied manuscripts were perceived as private copies at that time and did not adhere precisely to the original copy, e.g., there are cases where students included the margin notes of their masters' manuscripts within the text of their copies. In addition, the page layout of the original and the copied manuscripts may differ dramatically. This led to the existence of multiple different copies of the same manuscript. Figure 1 shows example pairs of pages, where the pages of each pair are from different copies of the same manuscript.

Nowadays, editing and publishing a historical manuscript involves a research phase to recover the original manuscript from the available copies using external and internal evidences. External evidence can be the geographical, temporal, and spatial dispersion of the variants. Internal evidence can be the writing style, and the mistakes sources such as difficult to read unseals, no word spacing, ink fading, etc. This research is usually done by highly educated professionals within the *manuscript critic* discipline. In addition, the field of *Stemmatology* attempts to reconstruct the transmission of the text in a manuscript based on the relations between its sur-

✉ Boraq Madi
borak@post.bgu.ac.il

Ahmad Droby
drobya@post.bgu.ac.il

Jihad El-Sana
el-sana@cs.bgu.ac.il

[1] Computer Science, Ben-Gurion University of the Negev, Be'er Sheva, Israel

viving copies. *Manuscript alignment*, which aims to locate the shared and the different text among a set of copies of the same manuscript, is essential for these research tasks. Currently, researchers are required to go over the copies (at least two) in parallel, locate the differences, and reason which phrases or words were part of the original manuscript and which were introduced during reproduction. A fundamental approach prefers to recover the original copy by aligning multiple copies because independent scribes are less likely to produce the same changes. An algorithmic approach to address these challenges needs to detect and locate these differences and provide an intuitive visualization scheme that enables researchers to provide the appropriate reasoning, concerning the spotted differences.

The alignment of two historical manuscripts is a challenging task and consumes valuable time of highly educated professionals. Nevertheless, it has received limited attention from the document image analysis community mainly due to lack of awareness of the problem, the absence of annotated datasets, and the complexity of the task.

Manuscript alignment can be addressed in two different domains, text and image. In the first approach, the manuscripts are first transcribed by computer tools or humans, and then the alignment is computed on the text domain. This approach requires a computer transcription tool with high accuracy or a significant human effort. The second approach computes the text alignment directly on the image domain without applying text recognition.

One could solve this problem by applying handwritten text recognition (HTR) and then computing the alignment at the text level using Longest Common Subsequence (LCS). However, there are several limitations to this approach. First, current HTR technology does not provide adequate accuracy when applied to handwritten historical documents. Second, a line-under-line synopsis of manuscripts is a combined presentation of different copies of the same manuscripts, which is vital for philologists to evaluate the most valuable and most original text which exists in multiple manuscripts, assess the value of different manuscripts, and form a stemma of the relationship between manuscripts and the history of the transmission of text. However, projecting the text level alignment back to the image space to help researchers validate and analyze the obtained alignment is challenging. Third, we believe that manuscript alignment is more constrained than the general handwriting text recognition, which may indicate that we need less training data or a smaller model to obtain comparable results.

This paper presents an alignment algorithm that work on the image domain. It inputs a pair of textlines with overlapping content, L_1 and L_2, from two different manuscripts and determine the region where they differ and where they overlap. The alignment of the textlines is performed in three steps: detecting subwords, estimating similarity, and establishing alignment among the subword sequences. In our current implementation, we adopted the YOLO [1] model to detect subwords in textline images and developed a Siamese model to estimate the similarity among subwords. Based on the estimated similarity, we construct a matrix that encodes the distance between the subwords of the two textlines. Finally, we establish the alignment by computing the Longest Common Sub-sequence (LCS) at the subword level using the constructed similarity matrix.

To evaluate our approach, we build a new alignment dataset, *VML-ALGN*, which includes 10 pages from two copies of the same historical Arabic manuscript. The alignment between the two copies is annotated at the subword level on the image domain. The *VML-ALGN* dataset will be publicly available for research purposes. We trained and experimentally evaluated our algorithm using *VML-ALGN* and obtained results that outperformed the state-of-the-art.

The contributions of this paper are:

1. A novel manuscript alignment approach that works directly on the image domain.
2. A historical manuscript alignment dataset, *VML-ALGN*. To the best of our knowledge, this is the first ,publicly available, dataset for Arabic manuscript alignment.
3. Benchmark results for aligning manuscripts on the VML-ALGN dataset.

The rest of this paper is organized as follows: In Sect. 2 we review related works. In Sect. 3, the new alignment dataset is presented. In Sect. 4, we present the alignment algorithm in details. Experimental results are reported in Sect. 5. Finally, conclusion is drawn, and future work is discussed in Sect. 6.

2 Related work

The alignment task appears in various study fields from biological sequences [2] and face images [3,4] to audio tracks [5] and language translation [6]. Document image analysis field uses alignment between the text image and its accompanying transcription, or the text image and its variant's image. Transcription alignment is an example of text-to-image alignment. It aligns the available transcriptions to the corresponding positions in the text image. Transcription alignment helps to automatically generate a large amount of ground truth necessary for training and evaluating effective document image recognition systems [7,8].

Manuscript image alignment is an example of image-to-image alignment. It aligns the positions of layout elements in a manuscript image to the corresponding positions in its variant's image.

Transcription alignment aligns the sequences from different domains. A widely used solution is to convert the image

Arabic Hebrew Latin

Fig. 1 Each column contains a pair of pages that come from different copies of the same manuscript. Pages from these copies often have some differences due to words/sentences altering. In addition, handwriting, text line arrangement, and page layout can be very different; e.g., the main text of the bottom page is similar to the top one in the second column

domain sequence to the text domain sequence utilizing a pattern recognizer. Then a dynamic programming algorithm is used to find the optimal alignment between the text domain sequences [9–11]. The recognition systems are not accurate enough in the case of historical documents or cursive handwriting. Therefore, text domain sequence is converted to image domain sequence by rendering [12–15] or cropping representatives from image domain [16], and a dynamic programming algorithm is used to align the features from the image domain. A hybrid method uses anchors obtained from recognition to constrain the alignment of rendered transcription [17]. Sometimes the manuscript and the transcript might have slightly different content. In such a case, energy minimization can be used to align over-segmented word images to

a single word transcript [18]. While these works rely on segmented word images, the Hidden Markov Models (HMMs) recognizer accompanied by the Viterbi decoding algorithm can implicitly segment and align the words [19–21].

Some other works assume that transcription is perfect and contains the words in the same order they appear in the text image. Accordingly, segmenting the text image into a known number of words, gathered from the perfect transcription, aligns the text domain and image domain without needing to convert one to the other [22,23]. Line segmentation in the image domain and line break information in the text domain are other means that provide sequential order. Prior line segmentation is essential for transcription alignment and possible using manual labeling [21] or textline segmentation algorithms [9,10,14,19]. Line break information can only be included during the manual transcription and increases the alignment accuracy [12]. However, in case the line break information is not provided, the transcription alignment algorithms either place each successive line at the end of the previous line [9,12,13] or split the lines between accurately recognized anchor words [10,24].

In line with state of the art in computer vision, deep learning approaches have been explored for transcription alignment. An object detection network is used for localizing and recognizing the anchor words to constrain the dynamic programming alignment [24]. A Seq2Seq model together with an attention mechanism is used to implicitly segment and align the words [25]. However, these approaches have not addressed the image-to-image alignment problem but only the text-to-image alignment problem.

Manuscript image-to-image alignment has not been studied well. One of the challenges in image-to-image alignment is the absence of line break information. Hence, textlines are segmented before the alignment and concatenated to form a long line. The similarity of the connected components can be computed either by hand-designed features [26] or by automatically extracted features from a deep network [27]. A dynamic programming algorithm is then used to align the features on the image domain.

Fig. 2 Historical document samples: **a**, **b**, **c**, and **d** are taken from the datasets HD_1, HD_2, $ALGN_1$, and $ALGN_2$, respectively

3 Dataset

This section overviews the datasets used in our experimental evaluation and explains their preparation procedure.

VML-HD [28] is a historical document dataset that includes five Arabic manuscripts, where each one has its style. It contains in total of 680 pages with subword level annotation. Each subword is annotated with a bounding box and a transcription label. We experimented with ten pages from two manuscripts from the VML-HD data and we shall refer to them as HD_1 and HD_2. The writing styles of these two manuscripts are exemplified in Fig. 2a and b. Each page in HD_1 and HD_2 has 15 and 14 lines, respectively, on average. The manuscripts in the dataset are challenging for many text recognition algorithms as they contain varying writing styles, diacritics, and touching components.

VML-ALGN is a new dataset that includes pages from two copies of the same manuscript titled *Almll wAlnhal* by *Alshahrastany, Muhammad Ibn Abed-Alkarym*, which was authored around 1127 − 1128. The manuscript presents the religious communities and their philosophies and creeds that had existed up to that time.

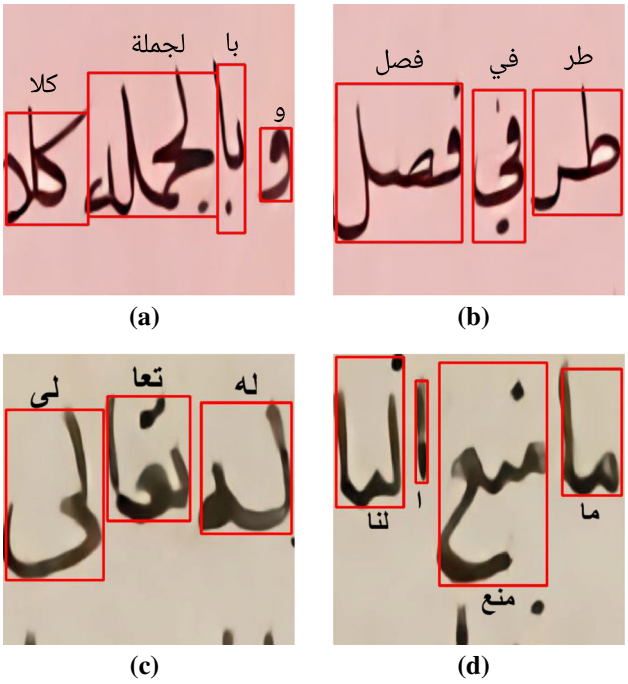


Fig. 3 The bounding boxes and the transcriptions of subwords. The samples **a** and **b** are from $ALGN_1$, while **c** and **d** are from $ALGN_2$

VML-ALGN is divided into two subsets based on the writing style (see Fig. 2c and d). Each subset includes five pages, and we shall refer to them as $ALGN_1$ and $ALGN_2$. The overlap, in terms of text, between the pages of the two subsets is about 60%, mainly due to the difference in the number of textlines each page. Each subword in the pages of VML-ALGN is annotated with a bounding box and a transcription, as shown in Fig. 3. An average page in $ALGN_1$ has about 15 lines, while an average page in $ALGN_2$ has about 25 lines.

We establish the alignment between the datasets pairs ($ALGN_1$,$ALGN_2$) and (HD_1,HD_2) at subword level, using the available transcription for each subword, and project it back to the image space. For example, a subword in $ALGN_1$, is aligned with its respective subword $ALGN_2$ (the two subwords have the same transcription). The alignment identifies a subword using its page and textline numbers and the index of its bounding box with respect to its textline.

4 Method

We present a novel framework for aligning two textlines from two different copies of the same manuscript on the image domain. The framework includes three main steps: automatic subword detection, estimating subword similarity, and establishing alignment among the detected subword sequences. Next, we discuss each one of these steps.

Fig. 4 YOLO inputs a patch of size 320×320 and predicts the bounding boxes of subwords within the patch

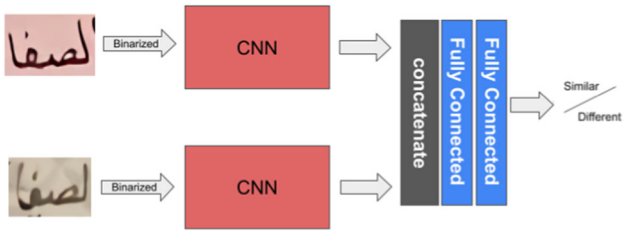


Fig. 5 Illustration of the Siamese Network. The networks take two images as an input and extracts feature vectors using two CNN streams with shared weights. The two vectors are then concatenated and passed to a fully connected network, which outputs a similarity measure

4.1 Subwords detection

In our current implantation, we adopt the YOLO [1] model to detect subwords in textline images. We train YOLO to predict the bounding boxes of subwords in patches, as shown in Fig. 4, where all the subwords are defined as one class, making the recognition procedure irrelevant. The details of training and test sets are explained in Sect. 5.

4.2 Subword similarity estimation

We estimate the similarity of two subwords in the image space using a Siamese network model, M_s. The model M_s is trained to measure the distance between two subword images; i.e., it estimates whether two subwords are the same or not based on their textual content (see Fig. 5). Each branch of the model is the feature extractor part of ResNet34 [29](e.g., the convolutional layers without the classifier part). The output of the two branches is concatenated and passed to a two-layered fully connected network. The loss of the model is computed using binary cross-entropy. The model was trained using subwords extracted from the datasets presented in Sect. 3. The proposed Siamese model achieved an accuracy above 90% using our dataset.

4.3 Establishing subword alignment

One could perceive a sequence of detected subwords of a textline as a string, in which letters are subword images. This scheme enables us to apply string alignment techniques, with some modifications, to align sequences of subwords.

The Longest Common Subsequence (LCS) problem finds the longest subsequence between two strings. Given two strings s_1 and s_2 with the sizes m and n, respectively, we apply dynamic programming to determine their LCS. The dynamic programming involves building a similarity matrix, c, of size $(m+1) \times (n+1)$, where the first row and first column are initialized to zero. The cells of c are assigned values according to Eq. (1), where $s_1(i)$ and $s_2(j)$ is the *ith* and *jth* characters in the sequences s_1 and s_2, respectively. The LCS alignment of s_1 and s_2 is obtained by finding the maximum path in the matrix c.

$$c[i,j] = \begin{cases} 0, & i = 0 \text{ or } j = 0 \\ c[i-1, j-1] + 1, & s_1(i) = s_2(j) \\ max(c[i-1, j], c[i, j-1]), & otherwise \end{cases} \tag{1}$$

The alignment between two strings is based on a binary decision between every two characters, i.e., one for the same characters and zero otherwise. However, alignment between subwords images is estimated by Siamese network, S, which does not output a binary decision but a probability. Therefore, we choose similarity threshold, T_{sim} to decide whether two subwords images: sw_i and sw_j are similar or not, as described in Eq. (2). We set T_{sim} to 0.5 in our experimental evaluation.

$$c[i,j] = \begin{cases} 0, & i = 0 \text{ or } j = 0 \\ c[i-1, j-1] + S(sw_i, sw_j), & S(sw_i, sw_j) \geq T_{sim} \\ max(c[i-1, j], c[i, j-1]) & \text{otherwise} \end{cases} \tag{2}$$

5 Experiments

In this section, we discuss our algorithm's experimental evaluation and explain the dataset's preparation. The experimental evaluation is performed on the components of the algorithm independently, and then we present the evaluation of the entire algorithm.

5.1 Dataset preparation

We prepare the datasets for subword detection and similarity estimation for our experimental evaluation.

Subword-detection datasets: We construct a set of patches from a dataset of textlines by sliding an elastic window over a textline at subword (bounding box) strides, i.e., at each iteration, the sliding window includes the bounding box of the next subword and excludes that of the oldest one. Recall that subword annotation includes its bounding box and these bounding boxes vary. We resize the extracted patches, which are determined by the dimension of the sliding window to 320×320 pixels and update the annotated bounding boxes accordingly, as shown in Fig. 6. Note that the size of the sliding window is dictated by the height of underline textline. This procedure creates patches with three or four subwords on average, where successive patches overlap, as shown in Fig. 6.

It is vital for learning subword localization to generate patches that include multiple subwords. In addition, we realize that multi-subword patches guide the model to generalize better, as they provide broader context than single subword patches. Furthermore, these patches include overlapping and touching subwords, often in typical historical documents.

We extract textlines from the pages of the datasets HD_1, HD_2, $ALGN_1$, and $ALGN_2$, and then apply the patch generation procedure to obtain the datasets det_{HD_1}, det_{HD_2}, det_{ALGN_1}, and det_{ALGN_2}, respectively (Table 1).

Similarity estimation datasets: We utilize the annotated bounding boxes to crop subwords from two overlapping manuscripts. The cropped subwords are grouped according to their transcription (See Fig. 7). The transcription (textual content) of the subwords defines the classes of the dataset. We distinguish between two types of classes: shared and unique. Shared classes exist in both manuscripts, while unique classes exist in one manuscript but not in the other.

We experimented with gray-scale and binary patches and obtained slightly better results, $1-2\%$, using the Otsu-binarized patches. The binarization seems to guide the learning to focus on text features and ignore the background. In addition, we balanced the dataset to include 30 samples for each subword class. To generate samples for classes with less than 30 samples, we augment these samples by applying random rotation with degrees between -10 and 10.

We applied the above procedure, including cropping, grouping, and balancing, to five pages from each of the sets HD_1 and HD_2, to obtain the dataset $subwords_{HD}$. The size of $subwords_{HD}$ and the number of shared and unique classes are reported in Table 2. We performed the same procedure to the datasets $ALGN_1$ and $ALGN_2$ and generated the dataset $subwords_{ALGN}$, which attributes are presented in Table 2.

5.2 Subword detection

We split each subword-detection dataset into 30% and 70% for train and test, respectively. We use the SGD optimizer where learning rate, momentum, and weight decay are set to 0.005, 0.9, and 0.0005, respectively. The YOLO model was trained for 30 epochs with a batch size of 4.

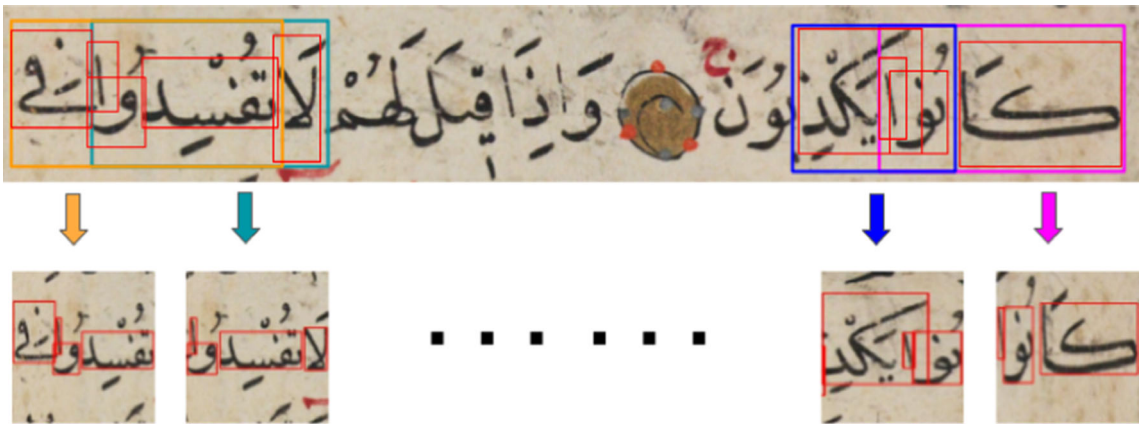


Fig. 6 **a** The first row shows an extracted textline from a page in HD_1 **b** The textline is divided to patches according to the extraction procedure. **c** The extracted patches and their subwords bounding boxes annotations

Table 1 The number of patches in each dataset

Dataset	Patch number
det_{HD_1}	1403
det_{HD_2}	1470
det_{ALGN_1}	2148
det_{ALGN_2}	3389

We experimented with several state-of-the-art text-detection algorithms [30–32] and found out that these models don't perform as well as they do in scene text or modern fonts. The average precision(AP) of those models were very low compared to YOLO. The best model among these algorithms obtains 47.8 AP while YOLO manages to reach 83.2 AP on the worst case using the same dataset. Thus we decide to adopt YOLO for subword detection.

To evaluate the performance of YOLO, we set the Intersection Over Union (IoU) threshold, T_{iou}, to 50% in order to calculate the average precision(AP). The AP is computed as the number of positive detection over the total number of samples. We define a positive detection as a prediction with IoU above T_{iou} with respect to its ground-truth. The YOLO instance is trained from scratch and tested for each dataset. The Average Precision (AP), $AP_{scratch}$, for each dataset is reported in Table 3.

To evaluate the relevance of a pre-trained model, we trained a YOLO instance on 65 pages chosen randomly from different manuscript of VML-HD (Fig. 8 show a sample page). The patches and their labels were prepared from these pages according to the procedure described in Sect. 5.1. We trained the YOLO model for 20 epochs using the above configuration. Let us refer to this instance as $YOLO_{vml}$. An instance of $YOLO_{vml}$, which is denoted $YOLO_{pre}$, is trained for additional 10 epochs using the subword-detection datasets. We tested the two YOLO models on the subword-detection test set and report the obtained results in the second column of Table 3, where $AP_{pre-trained}$ is the average precision.

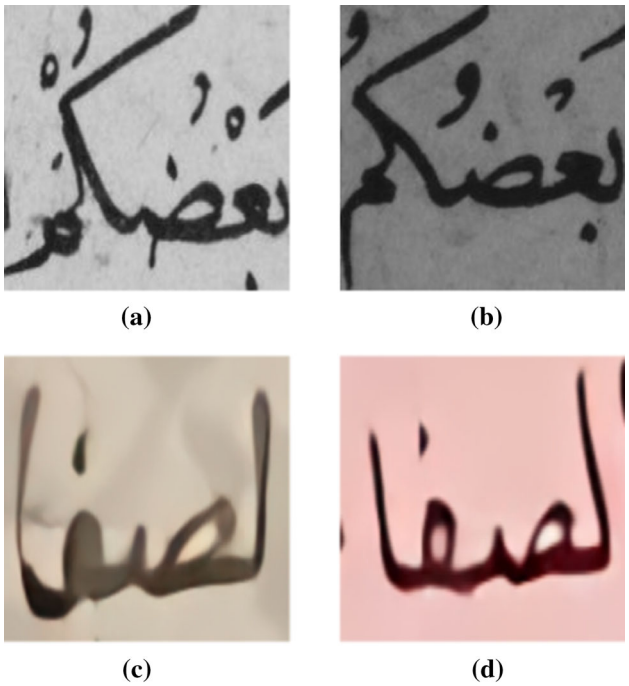


Fig. 7 Subwords with same transcription from $subwords_{HD}$ (top) and from $subwords_{ALGN}$ (bottom)

Table 2 The size and the number of shared and unique classes of each dataset

Dataset	Subword	Shared	Unique
$subwords_{HD}$	$21.3K$	303	104
$subwords_{ALGN}$	$52.83K$	563	635

The pre-trained model, $YOLO_{pre}$, increases the AP by 7% on average, as shown in Table 3. The $YOLO_{vml}$ model

Table 3 The AP of YOLO model over different datasets

Dataset	$AP_{scratch}$	$AP_{pre-trained}$
det_{HD_1}	83.3%	90.91%
det_{HD_2}	84.3%	93.5%
det_{ALGN_1}	81.1%	89.2%
det_{ALGN_2}	83.2%	86.15%

Fig. 8 Samples from manuscript of VML-HD that used in the pre-training experiment

achieves less accuracy, as shown in Fig. 9. The retraining phase guides the model to improve, as illustrated in Fig. 10.

5.3 Similarity estimation

We divide the two datasets $subwords_{HD}$, $subwords_{ALGN}$ to 30% and 70% for train and test, respectively. This division is applied to the shared and unique classes separately and creates two disjoint subsets for each class type(shared and unique). As a result, the train set includes 70% of shared classes and 70% of the unique classes. Numerical details of the train and test sets of $subwords_{ALGN}$ and $subwords_{HD}$ are reported in Table 4. In this division scheme, the model encounters new classes without prior knowledge during the evaluation. The excellent performance in these classes indicates the model's ability to generalize well.

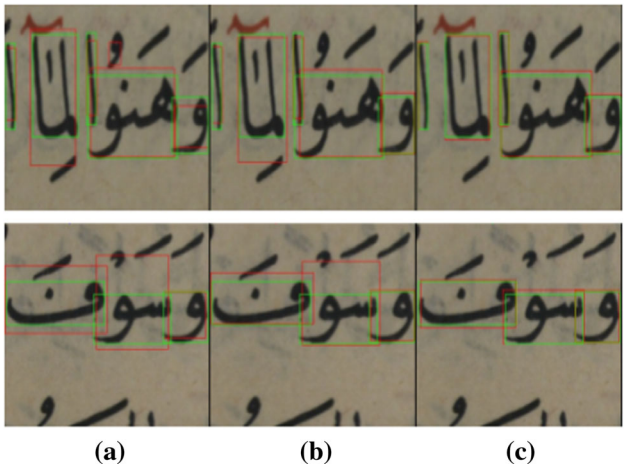


Fig. 9 The red bounding boxes are the predictions and the green are the ground truth. **a** The prediction is based on training on det_{HD_2} only. **b** The results of the pretrained mode without the second phase of training using det_{HD_2}. **c** The combination of pre-trained model followed by training on the training set from det_{HD_2}

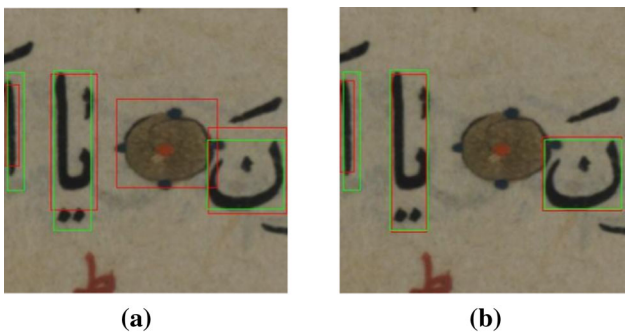


Fig. 10 **a** The pretrained predictions result. **b** The results of pretrained model followed by second phase of training on det_{HD_2}

Table 4 The number of Shared and Unique classes in the train and test sets for $subwords_{HD}$ and $subwords_{ALGN}$

Dataset	Train	Test
$Subwords_{HD}$	*Shared*	*Shared*
	242	61
	Unique	*Unique*
	83	21
$Subwords_{ALGN}$	*Shared*	*Shared*
	450	113
	Unique	*Unique*
	508	127

During training, pairs of subwords are drawn randomly and fed to the Siamese model as shown in Fig. 5. There are two types of subword pairs: similar (See Fig. 11a and b) and different (See Fig. 11c and d). A similar pair includes two subwords drawn from the same class, i.e., part of the shared classes of the dataset. Different pairs are drawn from different

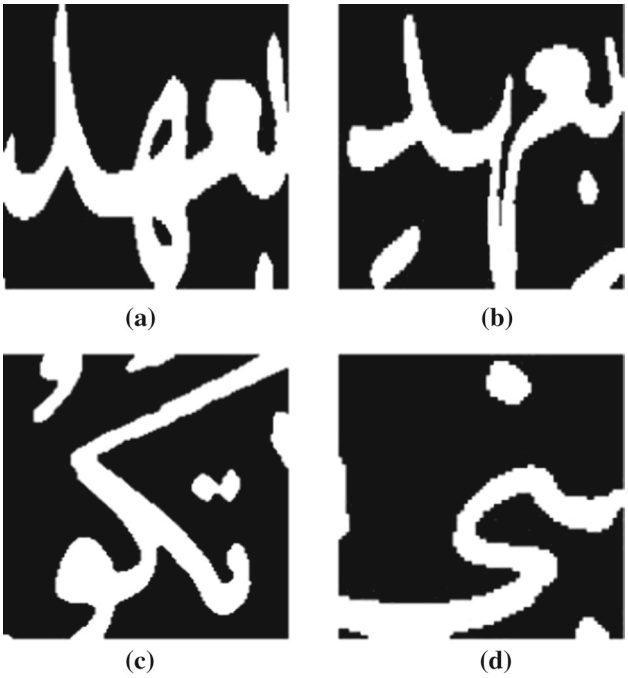


Fig. 11 Subword: **a** and **b** define a similar pair drawn from the same class, while **c** and **b** show subwords drawn from different classes

Table 5 The accuracy of different Siamese networks using the $subwords_{HD}$ and $subwords_{ALGN}$ datasets

Dataset	$Acc_{subwords_{HD}}$	$Acc_{subwords_{ALGN}}$
s_{kassis}	63.7%	68.64%
s_{vgg19}	70.59%	72.9%
s_{vgg16}	73.05%	81.19%
$s_{resnet18}$	86%	85.24%
$s_{resnet34}$	90.1%	88.7%

Fig. 12 Pairs of textlines from different manuscripts with overlapping content

classes without considering whether the class is shared or unique.

We experiment with five Siamese architectures that differ in-branch architecture. The first Siamese, s_{kassis}, is the model architecture proposed in [27], which includes three blocks, where each one consists of convolutional, ReLU, and max-pooling layers. The results of s_{kassis} is the only one reported in this domain, thus, we consider it the state-of-the-art for the similarity estimation. We replace the convolutional branch in s_{kassis} with ResNet34, ResNet18, VGG16, VGG19 [29,33]

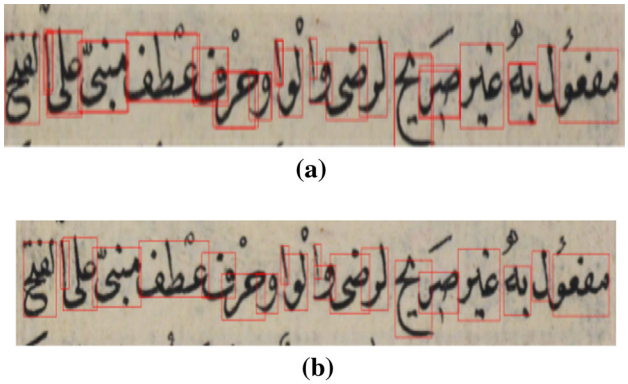


Fig. 13 **a** Bounding box predictions of YOLO for subwords of textline **b** Bounding box after applying NMS for the initial predictions of YOLO

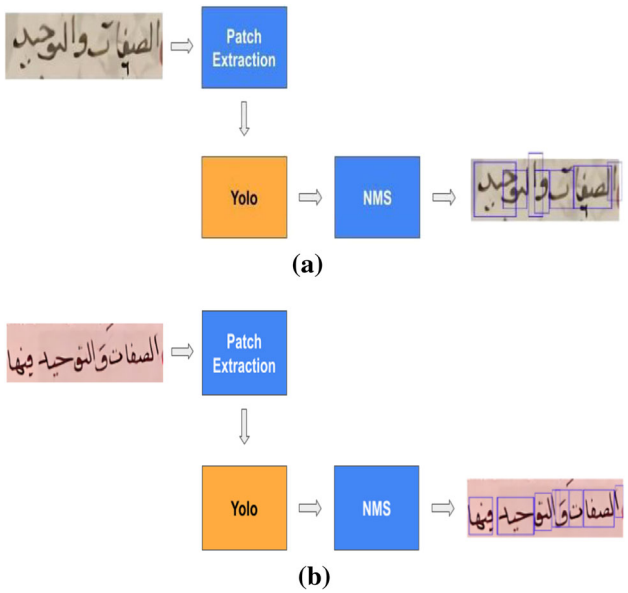


Fig. 14 Algorithm flow: **a** Final bounding boxes, bbs_a, for textline sample from $ALGN_2$. **b** A textline sample from $ALGN_1$ and its final bounding boxes, bbs_b

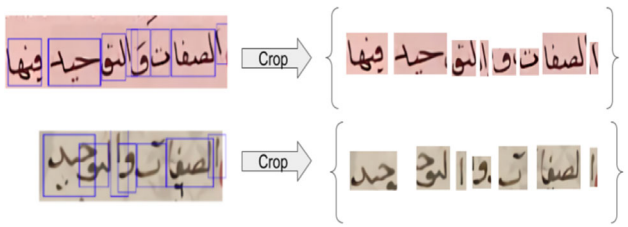


Fig. 15 The two subwords sets sb_a and sb_b generated using the bounding boxes of bbs_a and bbs_b

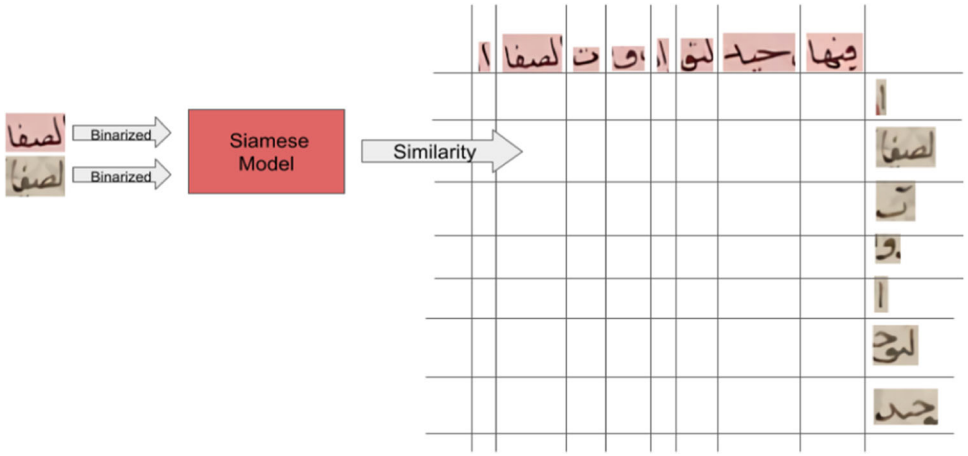


Fig. 16 The cells of the alignment matrix is computed using the trained Siamese network

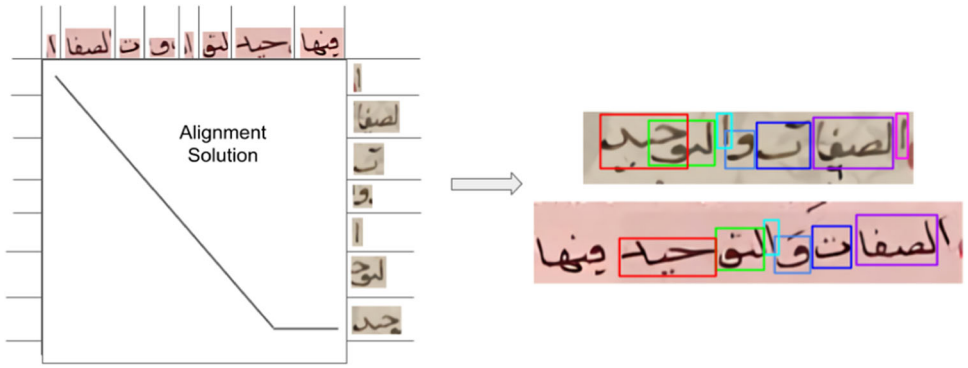


Fig. 17 Each pair(match) in the computed alignment have a unique bounding box colour

to create $s_{resnet34}$, $s_{resnet18}$, s_{vgg16} and s_{vgg19} models, which end with two fully connected layers.

Table 5 shows the accuracy of the five models tested on $subwords_{HD}$ and $subwords_{ALGN}$. The accuracy of $s_{resnet34}$ outperforms all other models, including s_{kassis} model. $s_{resnet34}$ has more parameters than the other models in Table 5. Thus, it can build a better separable feature representation for each class. Therefore, we choose $s_{resnet34}$ to perform the alignment. As seen in Table 5 the $s_{resnet34}$ model performs better on $subwords_{HD}$ than $subwords_{ALGN}$, which reflects the challenges of each dataset. For example, $subwords_{HD}$ has higher resolution, less touching components, and overlapping subwords than $subwords_{ALGN}$.

5.4 Alignment evaluation

We select pairs of overlapping textlines taken from two different copies of the same manuscripts, as shown in Fig. 12. These textlines were selected from HD_1 and HD_2 datasets to produce $textlines_{HD}$ dataset. Similarly, the $textlines_{ALGN}$ dataset was generated from $ALGN_1$ and $ALGN_2$ datasets.

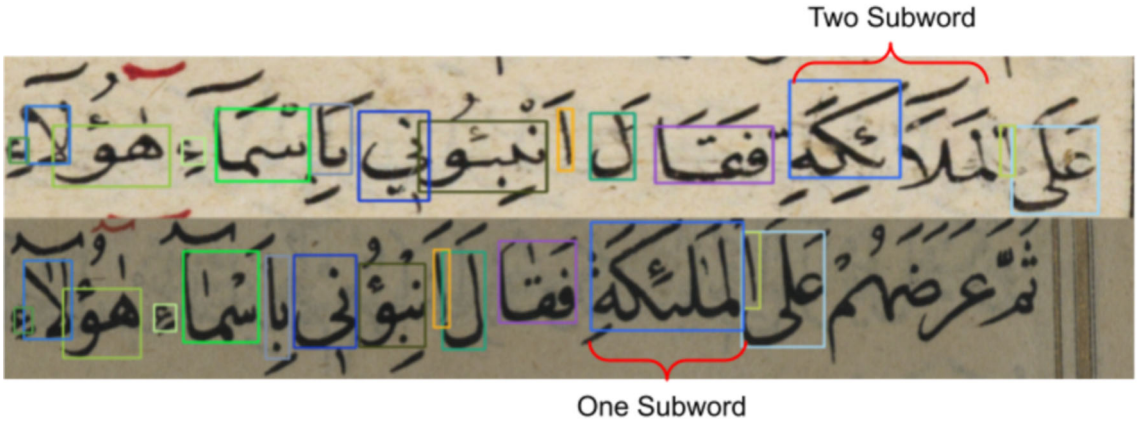
Let L_a and L_b be an extracted pair of textlines. We split L_a and L_b into patches using a sliding window with an overlap factor of 50%. These patches are resized and fed to the trained YOLO. We apply Non-maximum Suppression (NMS) (See Fig. 13) to remove redundant bounding boxes from the YOLO results and obtain the final subword bounding boxes, bbs_a and bbs_b, as shown in Fig. 14a and b, respectively.

We crop the subwords from L_a and L_b according to the bounding boxes bbs_a and bbs_b to generate two subwords sequences, sb_a and sb_b, as shown in Fig. 15.

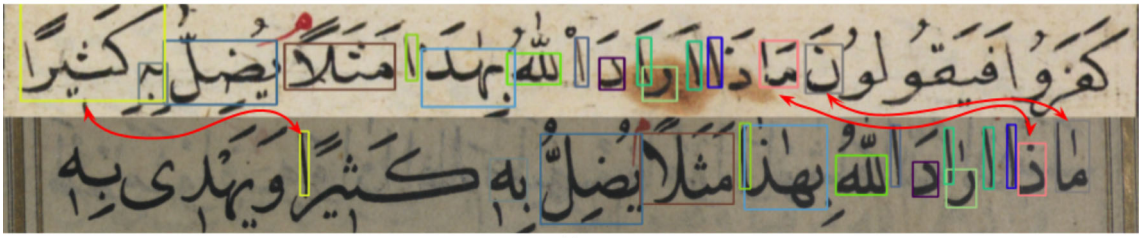
We construct a similarity matrix, M_{align}, of dimension $(\|sb_a\| + 1) \times (\|sb_b\| + 1)$, as illustrated in Fig. 16. The value each cell in M_{align} is computed according to Eq. (2). The similarity between the subwords is calculated using the trained Siamese network, as shown in Fig. 16. We calculate the LCS between L_a and L_b from the matrix and obtain the alignment result as shown in Fig. 17.

Figure 18 shows textline alignment results from $textlines_{HD}$ and $textlines_{ALGN}$. The alignment was evaluated by counting the number of correct matches over the total number of matches. Figure 18a has perfect alignment, while Fig. 18b have one miss-match. This miss results from the difference between writing styles of HD_1 and HD_2. For example, the word *Mlayikh* in Arabic (means angles in English) (see red brace in Fig. 18b is written in HD_1 as two subwords, while it is written as one subword in HD_2. The number of correct matches in the alignment of Fig. 18b is 14 with one miss. Thus the alignment accuracy is 93.3%. Figure 18c shows

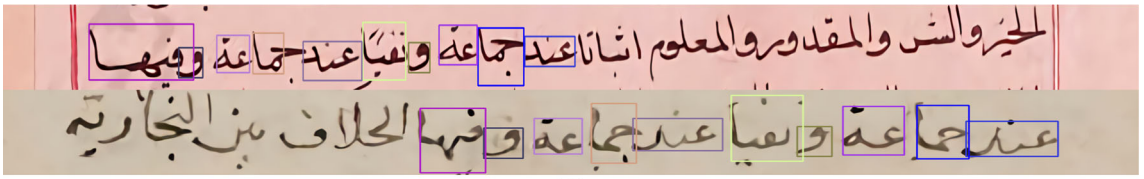
(a)



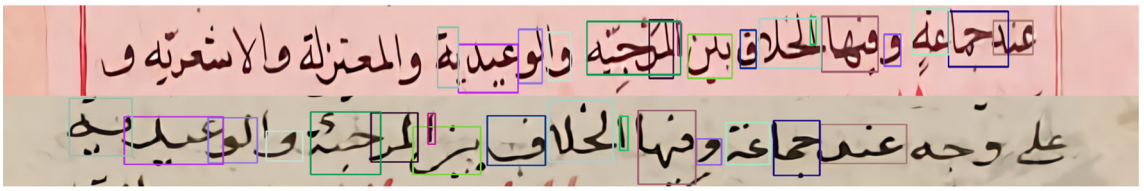
(b)



(c)



(d)



(e)

Fig. 18 The images **a**, **b**, and **c** show the output of the algorithm for samples from $textlines_{HD}$. The rows **d** and **e** are the alignment of textlines from $textlines_{ALGN}$

three miss matches (red arrows) and 12 correct matches lead to an alignment accuracy of 80%.

The alignment accuracy for the sets $textlines_{HD}$ and $textlines_{ALGN}$ were 91.6% and 81.3%, respectively. The higher accuracy of the $textlines_{HD}$ is attributed to the higher resolution of its images and the lack of touching and overlapping subwords.

6 Conclusion

This paper presents a three-step framework for aligning two textlines in the image domain. It detects subwords using YOLO, measures the similarity among subwords using the Siamese network, and establishes the alignment between subword sequences. We prepared subword detection datasets for training YOLO and evaluated its performance. The trained YOLO model achieves excellent accuracy over several datasets. To estimate the similarity among the extracted subword, we prepare two datasets for training and evaluating the Siamese network. In addition, we evaluated our framework end-to-end using pairs of overlapping pages.

In future plan, we will try to eliminate the need to subword extraction and develop a one-shot textline alignment method. In addition, We plan to explore the alignment on page level instead of textline level.

Author Contributions Madi Borak and Ahmad Dropy prepared the figures. All authors reviewed the manuscript. Madi Borak wrote the abstract, experiment, method, and dataset sections. Ahmad Dropy and El-Sana.Jihad wrote the introduction and related work.

Declaration

Conflict of interest The authors declare no competing interest.

References

1. Bochkovskiy, A., Wang, C., Liao, H.M.: Yolov4: Optimal speed and accuracy of object detection. CoRR (2020) arXiv:2004.10934
2. Gao, M., Skolnick, J.: A novel sequence alignment algorithm based on deep learning of the protein folding code. Bioinformatics **37**(4), 490–496 (2021)
3. Jourabloo, A., Liu, X.: Pose-invariant face alignment via CNN-based dense 3d model fitting. Int. J. Comput. Vis. **124**(2), 187–203 (2017)
4. Kowalski, M., Naruniec, J., Trzcinski, T.: Deep alignment network: A convolutional neural network for robust face alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 88–97 (2017)
5. Wang, J., Fang, Z., Zhao, H.: Alignnet: A unifying approach to audio-visual alignment. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3309–3317 (2020)
6. Choi, H., Cho, K., Bengio, Y.: Fine-grained attention mechanism for neural machine translation. Neurocomputing **284**, 171–176 (2018)
7. Al Azawi, M., Liwicki, M., Breuel, T.M.: Wfst-based ground truth alignment for difficult historical documents with text modification and layout variations. In: Document Recognition and Retrieval XX, vol. 8658, p. 865818 (2013). International Society for Optics and Photonics
8. Romero-Gómez, V., Toselli, A.H., Bosch, V., Sánchez, J.A., Vidal, E.: Automatic alignment of handwritten images and transcripts for training handwritten text recognition systems. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pp. 328–333 (2018). IEEE
9. Tomai, C.I., Zhang, B., Govindaraju, V.: Transcript mapping for historic handwritten document images. In: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 413–418 (2002). IEEE
10. Huang, C., Srihari, S.N.: Mapping transcripts to handwritten text. In: 10th International Workshop on Frontiers in Handwriting Recognition (2006). Suvisoft
11. Fischer, A., Indermuhle, E., Frinken, V., Bunke, H.: Hmm-based alignment of inaccurate transcriptions for historical documents. In: 2011 International Conference on Document Analysis and Recognition, pp. 53–57 (2011). https://doi.org/10.1109/ICDAR.2011.20
12. Kornfield, E.M., Manmatha, R., Allan, J.: Text alignment with handwritten documents. In: Proceedingsof the 1st International Workshop on Document Image Analysis for Libraries, 2004, pp. 195–209 (2004). IEEE
13. Kornfield, E.M., Manmatha, R., Allan, J.: Further explorations in text alignment with handwritten documents. Int. J. Document Anal. Recognit. (IJDAR) **10**(1), 39–52 (2007)
14. Lorigo, L.M., Govindaraju, V.: Transcript mapping for handwritten arabic documents. In: Document Recognition and Retrieval XIV, vol. 6500, p. 65000 (2007). International Society for Optics and Photonics
15. Hassner, T., Wolf, L., Dershowitz, N.: Ocr-free transcript alignment. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1310–1314 (2013). IEEE
16. Rabaev, I., Cohen, R., El-Sana, J., Kedem, K.: Aligning transcript of historical documents using dynamic programming. In: Document Recognition and Retrieval XXII, vol. 9402, p. 94020 (2015). International Society for Optics and Photonics
17. Ezra, D.S.B., Brown-DeVost, B., Dershowitz, N., Pechorin, A., Kiessling, B.: Transcription alignment for highly fragmentary historical manuscripts: The dead sea scrolls. In: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 361–366 (2020). IEEE
18. Cohen, R., Rabaev, I., El-Sana, J., Kedem, K., Dinstein, I.: Aligning transcript of historical documents using energy minimization. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 266–270 (2015). IEEE
19. Toselli, A.H., Romero, V., Vidal, E.: Viterbi based alignment between text images and their transcripts. In: Proceedings of the Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2007)., pp. 9–16 (2007)
20. Indermühle, E., Liwicki, M., Bunke, H.: Combining alignment results for historical handwritten document analysis. In: 2009 10th International Conference on Document Analysis and Recognition, pp. 1186–1190 (2009). IEEE
21. Fischer, A., Frinken, V., Fornés, A., Bunke, H.: Transcription alignment of latin manuscripts using hidden markov models. In: Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, pp. 29–36 (2011)
22. Zinger, S., Nerbonne, J., Schomaker, L.: Text-image alignment for historical handwritten documents. In: Document Recognition and

Retrieval XVI, vol. 7247, p. 724703 (2009). International Society for Optics and Photonics
23. Stamatopoulos, N., Louloudis, G., Gatos, B.: Efficient transcript mapping to ease the creation of document image segmentation ground truth with text-image alignment. In: 2010 12th International Conference on Frontiers in Handwriting Recognition, pp. 226–231 (2010). IEEE
24. Ziran, Z., Pic, X., Innocenti, S.U., Mugnai, D., Marinai, S.: Text alignment in early printed books combining deep learning and dynamic programming. Pattern Recognit. Lett. **133**, 109–115 (2020)
25. Torras, P., Souibgui, M.A., Chen, J., Fornés, A.: A transcription is all you need: Learning to align through attention. In: International Conference on Document Analysis and Recognition, pp. 141–146 (2021). Springer
26. Asi, A., Rabaev, I., Kedem, K., El-Sana, J.: User-assisted alignment of arabic historical manuscripts. In: Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, pp. 22–28 (2011)
27. Kassis, M., Nassour, J., El-Sana, J.: Alignment of historical handwritten manuscripts using siamese neural network. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 293–298 (2017). IEEE
28. Kassis, M., Abdalhaleem, A., Droby, A., Alaasam, R., El-Sana, J.: Vml-hd: The historical arabic documents dataset for recognition systems. In: 1st International Workshop on Arabic Script Analysis and Recognition (2017). IEEE
29. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
30. Tian, Z., Huang, W., Tong, H., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network **9912**, 56–72 (2016). https://doi.org/10.1007/978-3-319-46484-8_4
31. Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11474–11481 (2020)
32. Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9336–9345 (2019)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)