Electronic Letters on Computer Vision and Image Analysis 24(2):183-217, 2025

Extraction and Separation of Touching Character in Arabic Manuscripts

Nabil AOUADI*

* *LaTICE-ENSIT, 5 Avenue Taha Husseïn, BP 56 Bab Mnara-Tunis, Tunisia*
Received 20th of october, 2024; accepted 2nd of december 2025

Abstract

This paper addresses the problem of touching characters (TCs) extraction and separation in Arabic manuscripts. It proposes a recognition-based method to separate them and reassign each component piece to its corresponding word. Whether the TCs are located within a single text line or between adjacent ones, the proposed method retrieves them from the document. We enhanced an existing technique for the extraction of vertically aligned (up-down) TCs. We suggested a novel extraction technique for horizontally/left-right TCs that is based on an examination of the morphology of Arabic word terminal letters. Next, TCs are identified by template matching, which makes use of an interpolation method based on Thin Plate Spline transformation and a Shape Context descriptor. Finally, the separation is performed according to the distance from the centroids of the identified template parts. Experiments were conducted on a large dataset using three metrics: Manhattan, Euclidean, and Canberra distances. The performance reached 94.2% for the extraction of horizontally/left-right TCs and 94% for the extraction of vertically/up-down TCs, together with a 94% accuracy for the separation method. The proposed approach surpasses existing techniques and effectively handles the diversity, variability, and structural complexity of TCs.

Key Words: Arabic manuscript, Touching characters, Extraction, Separation, Shape Context descriptor, Interpolation function, Midpoint.

1 Introduction

Arabic handwriting recognition systems face numerous challenges, including character decomposition, skewed text lines, ligatures, variable word spacing, diacritical marks that can be positioned above or below the character body and alter the lexical meaning of the word, overlapping sub-word segments within the same vertical region, among other issues.

Arabic script is cursive, which adds to these challenges, especially when characters touch (known as touching characters, or TCs). TCs occur when adjacent characters connect either horizontally or vertically, often due to narrow spacing between lines (see Fig. 1(a)) or stylistic calligraphy (Fig. 1(b)). This challenge is particularly prevalent in unconstrained Arabic manuscripts, where most characters (21 out of 28) possess ascenders, descenders, and multiple diacritical marks (see Fig. 1(h)).

Correspondence to: $nabil.aouadi@utic.rnu.tn$

Recommended for acceptance by $by Angel D. Sappa$
https://doi.org/10.5565/rev/elcvia.2019
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

Segmenting Arabic manuscripts ideally should occur without touching characters (TCs). However, when TCs are present, a more detailed analysis is needed. TCs can be categorized into three types:1) Simple Touching (see Fig. 1(c)), 2) Multi-Touching (see Fig. 1(d)): more than two characters are connected and 3) Overlapping (see Fig. 1(e) for inter-lines overlapping and Fig. 1(f) and (g) for inter-words overlapping): The connection extends beyond the characters, which can happen between lines (inter-lines) or between words (inter-words).

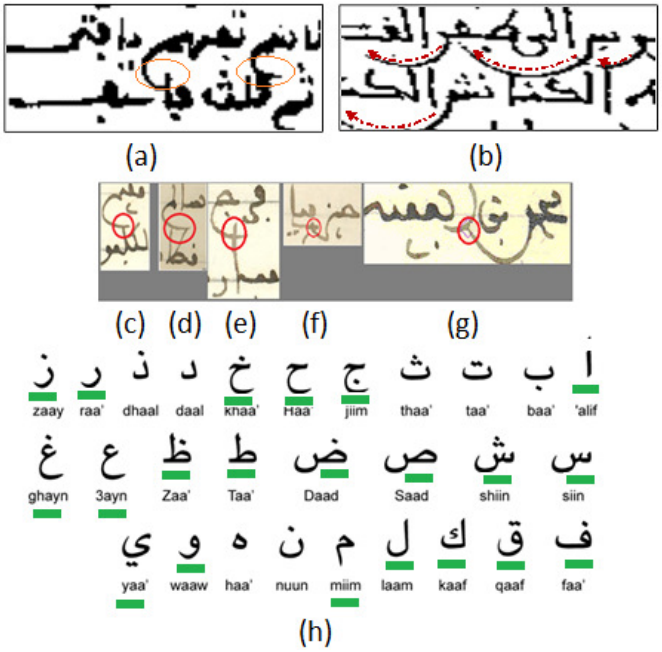


Figure 1: Examples of TCs: (a) TCs that happen when interlines are closely spaced, (b) TCs because of large descender, (c) simple TC, (d) multi-touching character, (e) Overlapping character, (f) An illustration of a TC that is horizontally inter-word, (g) Horizontally inter-word TC that is not consecutive, (h) Arabic characters with underlining for ascenders and descenders.

TCs can also take two forms: horizontal (left-right), occurring between successive characters or words within the same text line, or vertical (up-down), appearing between consecutive text lines.

Kang and Doermann [1] state that there are two primary methods for TC separation: recognition-based and recognition-free methods. The first method makes use of the TCs' structural data, such as projection profiles, skeletons, and contours. Therefore, a TC can be divided using a few guidelines without the need for a recognition phase. This method might effectively address specific touching issues, but it may not be robust enough to handle significant variability or unpredictability. Using a recognition step or supplementary evaluation parameters, the second method produces multiple candidate separation hypotheses and chooses the best one, but it works effectively only if the candidates are well separated.

The objective of this work (The full pipeline is summarized in Algorithm 3 and Fig. 2) is to extract and separate TCs in unconstrained Arabic handwritten texts. We firstly proposed two methods to extract TCs that respectively occur between vertically adjacent text lines and words of the same text line. These methods are mainly based on morphological analysis of Arabic characters. We then proposed a new recognition-based method for separating them. The rest of this paper is arranged as follows. Section 2 discusses related work, and section 3 details the proposed method, organized into two main stages. The first stage focuses on the detection and extraction of ambiguous connected components referred to as Touching Characters (TCs) that may occur either vertically or horizontally within Arabic handwritten lines. The second stage aims to separate these extracted TCs using a recognition-based strategy. This involves: (i) computing a similarity measure between the TC and the dictionary templates, (ii) selecting the Most Similar Template (MST) that maximizes this similarity, and (iii) transferring the known separation structure of the MST, into the TC using the estimated

TPS parameters. Section 4 presents the results and their evaluation, while section 5 concludes the paper.

2 Taxonomy of Touching Character Separation Approaches

Following a literature review, the methods are classified into four categories: recognition-free, recognition-based, hybrid, and deep learning approaches.

2.1 recognition-free approach

Methods within this category are the most commonly employed. They exploit statistical and structural characteristics like TC's density values in its different regions, the TC's histogram projection profile's peaks and valleys, the TC's type, the TC's neighborhood (size and membership of the nearest connected components), and the TC's position (relative to text lines, page columns, baseline, etc.). To separate TCs, the methods use hypotheses, thresholds, and parameters in relation to the handwritten text as line spacing and connected components size, etc. Typically, separation techniques are carried out in the region of the TC with the lowest density or are predicated on the baseline, projection profile, contour, or skeleton analysis of the TC.

According to [1], these methods are empirical and may address some types of TCs, yet it may not be resilient to large variations or uncertainty. To us, they can be designed ad hoc for only some particular layouts and document types. Guided by the document's structure, the recognition-free approach is arduous to generalize to other manuscripts or scripts.

The hypothesis used by the methods that separate TC on the low-density region, considers that the junction part linking two overlapping characters is located in the zone with the minimum stroke width [2, 3]. In case of overlapping or multi-touching connections, such methods are usually inadequate, especially for character recognition tasks. Projection profile analysis methods consider the valleys between consecutive peaks in the TC's projection profile histogram, along with the spaces separating successive text lines or baselines, as optimal locations for cutting the touching components [4, 5, 6, 7, 8, 9, 10]. However, since the projection profile analysis does not produce significant peaks [11]], especially in the case of a manuscript, we believe that in this way, separate characters risk being broken, and their morphology is not preserved.

Methods, based on analyzing TC's contour, follow contour pixels lying in the upper or lower parts relative to adjacent text lines [12, 13]. In [14, 15, 16, 17, 18], TC's skeleton was used for the fragmentation of these connections.

In Arabic words, text lines may connect at the level of terminal letters, as noted by Ouwayed and Belaïd [14]. Therefore, in order to split TCs into two parts, the first step is to 1) identify the point at which lower-line ascenders and upper-line descenders can intersect (S_p), and 2) locate the beginning point (B_p) of the ligature, which is typically the highest point, close to the top line's baseline. The angular variance and the skeletal pixel follow-up are used to distinguish the TC. Beginning at B_p, the follow-up proceeds to S_p. At this stage, the process continues along the curve exhibiting the smallest angular variation. The proposed method treated only four types of connections.

Chen and Leedham in [19], used a blind source separation matrix within the ICA framework to break down mixed pixel information into statistically independent components in order to separate overlapping regions.

2.2 recognition-based approach

Within this approach, two subclasses of methods can be distinguished. The first subclass considers multiple generated candidates separation assumptions and seeks the best one based on recognition. The second subclass of methods uses cutting pathways and other outcomes inferred from a classifier training step (neural networks, support vector machines, etc.).

In the first category, we report the method of Kang and Doermann [1], where a dictionary including templates with their correct separations is created. The method used the shape context [20] similarity is calculated

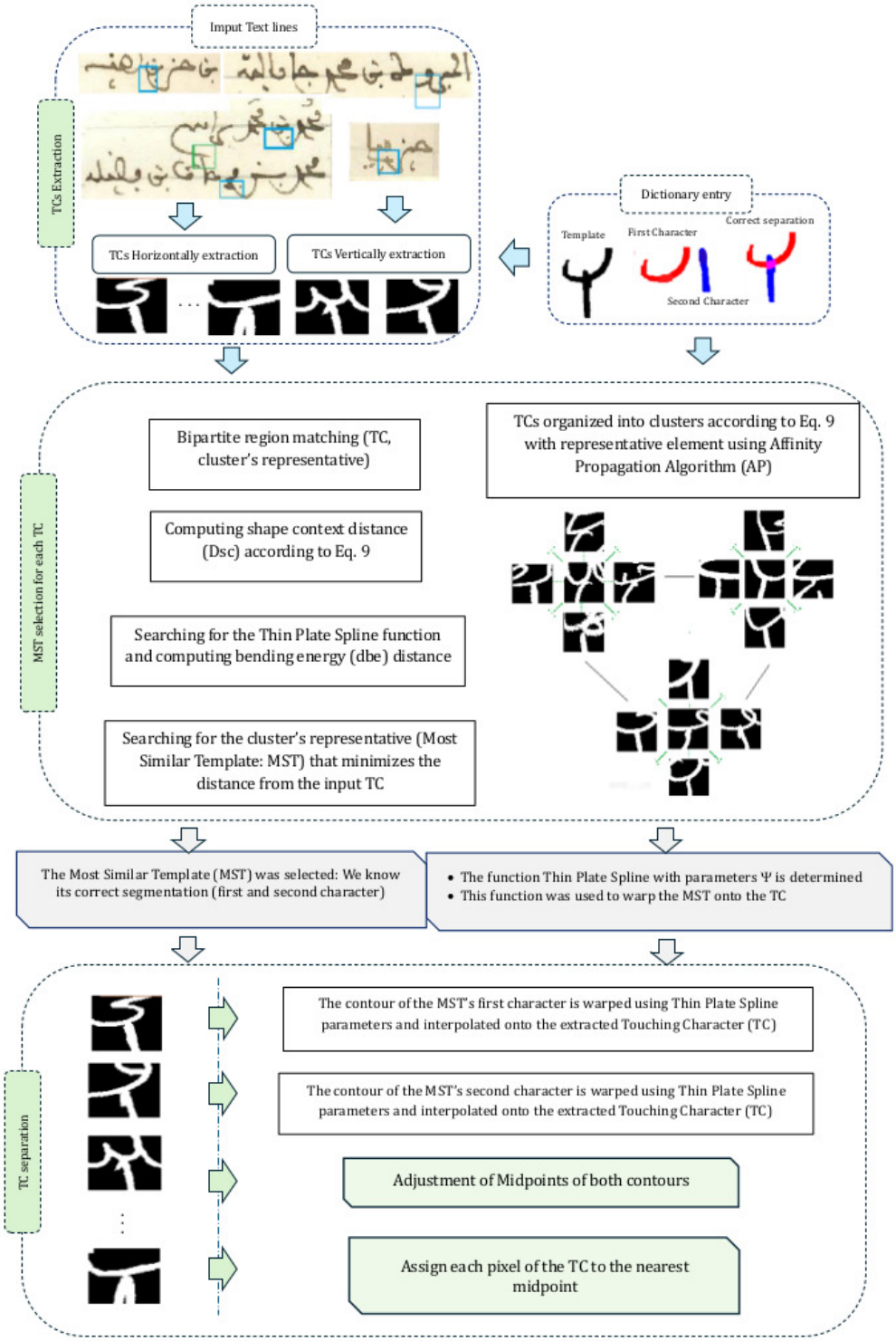


Figure 2: Overview of the proposed system for separating TCs in Arabic handwritten manuscripts.

between the input TC and stored templates to determine the best match, after which the closest template is superimposed on the input TC for separation. Following the same methodology, Ikeda et al. [21] look for the cutoff point, where TC needs to be separated. The TC is then divided into columns, and the region resulting from combining the columns on the left and right is examined. The procedure searchs for another break point in a pre-made graph separation hypothesis if the combination of two created characters is illogical.

2.3 hybrid approach

Methods within this approach segment an input TC using a corresponding template that has already been divided. In other cases, they employ a free separation strategy that exploits the structural or statistical characteristics of the TC. In the work of Piquin et al. [22], a decision tree is employed as a symbolic representation of the TC skeleton. The method searches a dictionary for a matching TC; if found, the component is divided into two parts according to the corresponding graph. Otherwise, the relationship between arcs is analyzed by comparing their arctangent values to determine whether one arc extends from another.

2.4 machine learning approach

Machine learning approaches process the connection as an integrated entity without relying on prior structural knowledge. Some of these methods leverage outcomes derived from classifier training (e.g., neural networks, support vector machines) to determine cutting paths for separating touching digit strings, thereby eliminating the need for additional post-processing [23, 24]. In [25], A U-Net model was trained to segment touching characters at the pixel level. The images were annotated using four colors: (i) one for the background, (ii) one for the first segment, (iii) one for the second segment, and (iv) one for the touching region. Dutta (2021) employed a deep learning architecture for text line segmentation. Structural features were extracted to identify touching or overlapping components that intersect the separation line.

2.5 discussion

The studied approaches are summarized in two comparative tables (Tables 1 and 2). Table1 presents the document type (skewed, cursive, binary, or gray scale), the dataset used, the recorded results, and whether the method handles all TC categories or only specific ones. Table 2 provides an overview of their classes, subclasses, extraction techniques, and segmentation context. For each class, we will highlight its strengths and weaknesses.

Touching character separation is a key step in handwriting recognition. Its success depends on accurately extracting the TC while considering its surroundings (text line, separating line, baseline, neighboring columns, connected components, etc.). However, the literature places more emphasis on TC detection than on separation.

Most separation techniques are tailored to specific document types and address only certain TC cases, leaving other cases for alternative methods. Many are tested on limited datasets, making evaluation and generalization to other manuscripts or scripts is challenging.

Many of the methods reviewed rely on document layout, using parameters like position, size, height, ratio, threshold, TC intersection with text lines or baselines, column width, crossing lines, and bounding box. While these factors may work well for some documents, scripts, or writing styles, they may not be suitable for others.

Some methods classify touching characters (TC) into categories: simple touching, multi-touching, and overlapping, applying specific treatments to each. This classification relies on geometric analysis but can be misleading, risking incorrect treatment (e.g., applying an overlapping method to a multi-touching case).

Methods that separate TCs by horizontal cutting often struggle to determine the precise cut position, as character connections are usually more complex than a simple horizontal overlap between an ascender and a descender.

Methods that identify junction points in the skeleton image can be error-prone, as they often assume a single unique junction point (as in [3] and [14]). However, this point may be missing, incorrectly detected, or

not unique, especially when loops are involved or due to skeletonization limitations. Thinning can also lose important details, making these methods less effective for ancient documents.

Other methods analyze contour information using angles and curves, offering better adaptability by considering connection irregularities and generating multiple separation options. However, they still struggle with different types of TCs.

For separation based on low-density regions, the cutoff point between two connected characters is assumed to lie in the narrowest region. However, this assumption is often incorrect, especially for overlapping TCs. In these cases, the cutoff point is typically located in the largest part.

The study of recognition-based methods showed that they are most effective when used with a large template dataset, as they can handle all types of TCs. However, methods in the first subclass require the closest template or an advanced recognition system to select the correct one; otherwise, TC separation may be poorly performed. Methods in the second subclass depend heavily on choosing the right features to avoid unnecessary cuts and prevent over-separation.

Only one machine-learning technique, proposed in [25], was briefly mentioned. It employs a modified U-Net architecture (AR2U-Net) to separate touching characters through pixel-wise classification.

Other deep learning methods [26] and [27] focus on text line segmentation, particularly for overlapping lines and connected characters. For character separation, they use statistical rules based on the separator line and the sizes of the detected connected components.

Even while machine learning method using deep architectures have shown remarkable performance in text line segmentation, they still face several challenges, particularly in handling touching characters (TCs). These methods often struggle with generalization, requiring extensive post-treatment training datasets that encompass diverse TC scenarios. Additionally, their need for extensive labeled datasets renders their implementation both labor-intensive and time-consuming.

Learning-based approaches are naturally able to manage the issues brought on by complex and heterogeneous TCs. These approaches have two drawbacks: 1) The choice of the training, so that the model can separate a wide range of TC types, and 2) The machine learning algorithm require extensive labeling, which is time-consuming.

2.6 Contribution

This work makes the following key contributions:

- **Two complementary methods for extracting vertically (up-down) and horizontally (left-right) Touching Characters (TCs):**

    1. *Vertical TC extraction:* An enhanced version of the method proposed by Ouwayed and Belaïd [14], adapted and improved to detect and localize TCs occurring between successive text lines in Arabic manuscripts. Enhancements include better preprocessing and localization steps tailored to ancient scripts.
    2. *Horizontal TC extraction:* A newly designed method dedicated to extracting TCs occurring between adjacent words on the same text line: an aspect not addressed in previous works. This method uses geometric and structural features specific to horizontal connections in handwritten Arabic text.

- **A novel recognition-based method for TC separation:**

    1. *Recognition phase:* A new bipartite graph matching technique based on regions is used to identify the Most Similar Template (MST) from a dictionary of pre-segmented character templates.
    2. *Approximation phase:* The transformation between the selected MST and the input TC is estimated using a region-based alignment strategy.

Table 1: Techniques for touching character separation proposed in the literature.

S: Simple touching, O: Overlapping, M: Multi-touching.

Ske: distorted TCs in manuscripts; Cur: handwritten cursive TCs.

Ar: Arabic, La: Latin, Bg: Bengali, Jp: Japanese, Ch: Chinese, Pr: Persian, Gr: Greek, Num: Numeric string.

ln: between lines, wl: within the line, dig: between digits.

Ref.	**Script**	**TCs position**	**Nature**	**Ske**	**Cur**	**Dataset (Name,TCs)**	**Rates** (%)
[1]	Ar	ln	S+C+M	+	+	744	74.4
[2]	La	ln	S, C	+	-	68	82.4
[3]	Ar	wl	S	+	+	IFN-ENIT's image word	-
[4]	La	ln	S+C	+	+	postal addresses	-
[5]	La	ln	S, C	+	-	many	77
[6]	Bg	ln	S	+	+	many	-
[7]	Ar	ln	S+C	+	+	ancient Arabic documents	-
[9]	Ar	ln	S	+	+	many	-
[10]	La	ln	S	+	-	ICDAR'09	-
[14]	Ar	ln	S, C	+	+	256	96.88
[15]	La	ln	S+C	-	-	ICDAR'07 database	-
[16]	La	ln	S	+	+	ICDAR'07 database	-
[17]	La	ln	S+C	-	-	IAM database	-
[18]	La	ln	S, C	+	+	ICDAR'09, ICDAR'13, UMD, GW, Barcelona database Marriage	-
[21]	Jp	ln	S	+	+	many	88.2
[22]	La	ln	S+C+M	+	+	20	90
[23]	Num	dig	S	+	-	303 for test and 1284 for training	82.8
[24]	Num	dig	S	+	+	10000	SVM:98.9, MLP:99.4
[25]	Ar	ln and wl	S+C+M	+	+	744 for training, and 744 for testing	94.6
[28]	La	ln	S+C	+	+	744 for training, and 744 for testing	94.6
[29]	Ch	ln	S	+	-	many	95.58
[30]	Pr , La , Gr , Bg	ln	S, C	+	+	52	-
[32]	Ar	ln and wn	S+C+M	+	+	1500	94

3. *Separation phase:* The final separation is performed by computing the midpoints of the aligned template parts, enabling accurate splitting of the TC (improvements over previous work cited in [31] and [32]). This method
   - is robust against template mismatch: it can successfully separate TCs even if the exact matching template is not present in the dictionary, thanks to the exploitation of internal pixel distribution and points in template parts.
   - does not require a dictionary that covers all possible template configurations, nor does it need a very close template to separate the input TC.

- **Creation of a new benchmark dataset:** We introduce a dataset named Touching Characters for Arabic Manuscripts (TCFAM1), consisting of 6000 carefully curated and annotated samples of Arabic TCs. This dataset provides detailed ground truth and covers a diverse range of character and connection styles, aiming to support future research and benchmarking in Arabic manuscript analysis.

3 Proposed System

The proposed system is designed to locate and separate touching characters (TCs). It is composed of two main components. The first two components are responsible for detecting and extracting TCs: one focuses on vertically/up-down TCs, while the other targets for horizontally/left-right TCs. The second component performs the separation of each extracted TC using a recognition-based strategy. Specifically, it retrieves the Most Similar Template (MST) from a reference dictionary and transfers its known character separation to the input

Table 2: Classification of techniques for separating touching characters.

PP: projection profile, HPP: horizontal projection profile, CC: Connected Component, SL: Separating line.
TL: text lines, PAW: parts of Arabic word, Ch: characters, Dig: digit

Approach	TC Extraction	TC Separation	Text Segmentation Context
Recognition-Free	- When detecting junction pixels in skeleton image.	Skeletonization [14].	into TL.
	- By calculating the smallest distance between two component text lines.	Tracking the tangent line that connects the component convex hulls on either side of the TC [9].	into TL.
	- When a CC is detected to intersect the boundary between two neighboring text lines.	In the low density region [2, 30].	into TL.
	- When detecting junction pixels in skeleton image below the baseline.	In the low density region [3].	into PAW.
	- Based on the vertical size of the component and the analysis of its histogram-based PP.	Analysis of HPPs [5].	into TL and PAW.
	- When a Component belongs into two text-lines.	Analysis of HPPs [4].	into TL.
	- When a component touches both adjacent text lines.	Between the two baselines [10].	into TL.
	- Based on the classification of CCs using K-means.	Inter-space classification by K-means [7].	into TL.
	- In case of absence of two SLs in two adjacent regions.	Analysis of HPPs [6].	into TL, PAW and Ch.
	- TCs are already localized.	contour decomposition [13].	into Ch.
	- When two successive text lines intersect a CC.	Contour tracking [12].	into TL.
	- When two successive text lines intersect a CC.	Skeletonization [16].	into TL.
	- According to the analysis of the component's vertical extent.	Skeletonization [15].	into TL.
	- When observing discontinuity in background skeleton image.	Skeletonization [18].	into TL.
	- When a CC spans between two neighboring baselines.	Skeletonization [17].	into TL.
	- TCs are already localized.	Independent component analysis algorithm (ICA) and a blind source matrix [19].	into Ch.
	- TCs are generated in a semi-automated way.	Assigning pixels based on statistical information [26] and [27].	into TL.
Recognition-Based	- Based on Viterbi algorithm.	Recognition of separation result by OCR [29].	into Ch.
	- Assumption that TCs are already localized.	Overlaying the separation template on the input TC [1].	into Ch.
	- Based on the analysis of the CC's height.	Merging parts followed by recognition step [21].	into Ch.
	- TCs are assumed to be already localized.	Neural network with run-length coding scheme-based recognition [23].	into Di.
	- TCs are assumed to be already localized.	Multilevel concavity analysis-based recognition [24].	into Di.
Hybrid	Based on the analysis of the CC's height.	Skeletal analysis and dictionaries [22].	into TL.
Machine Learning	- Assumption that TCs are already localized.	A modified U-Net that classifies pixels of images of TCs into four groups: background pixels, Character-touching pixels, first, and second character pixels [25].	into Ch.

TC through geometric alignment and adjustment of the contour midpoints of the two character components. The following subsections detail each stage of the system.

3.1 A Methodology for Vertical and Horizontal TCs Extraction

According to the literature review on TC extraction in Arabic manuscripts, no existing method automatically extracts TCs occurring between words of the same text line (i.e., horizontally/left-right TCs).

State-of-the-art methods have primarily focused on TCs occurring between successive text lines (i.e., vertically/up-down TCs). These methods often rely on the following conditions to detect TCs:

- Any component is taller than average, belongs to two alignments, or satisfies specific size and ratio requirements [4, 5, 7, 10, 14, 16, 21]. The uniform skew of the manuscripts, the distinctive writing style,

the easily recovered baselines, and the regular spacing of the text lines all contribute to the ability of most of these methods to localize the TCs.

- A buildup of black pixels along the separator line is detected. Baselines, histogram projection profiles, or the Viterbi method can be used to determine the separation line [2, 21].
- A junction point is identified within the skeleton between two adjacent text lines. This junction pixel is a point that connects disjoint letters [3].
- The vertical distance between components in neighboring text lines decreases [9].
- A discontinuity is found in the background skeleton image. Valleys in the skeleton were utilized in [18] to determine the optimal pathways. The pathways were represented as graphs, where regular pixels served as the connecting edges and specific pixels as the vertices. When a discontinuity between pathways was noticed, TCs were found.

Table 2 (column 2), displays the majority of previous works proposed for vertically/Up-Down TCs extraction. Hereafter, we present a detailed description of our proposed methodology for extracting TCs in both vertical and horizontal directions. These two closely related processes constitute the core of the initial phase of our system.

3.1.1 Vertically Interconnected Touching Character Extraction.

We enhanced the Ouwayed and Belaïd [14] approach, which has been successful in handling this assignment, to extract TCs between consecutive text lines. Here are the following steps:

– Baseline positions are computed based on the input image: Extracting a baseline is particularly challenging when dealing with Arabic handwriting, as opposed to Latin script, which typically features a significant concentration of black pixels along a line. The primary reason for this is the presence of letter extensions or horizontal ligatures [33]. The line support present in the processed texts (blue line in Fig. 3) was used as the baseline, as histogram projection profiles fail to accurately identify it in our manuscripts.

The technique looks for gray-level pixels with locally minimum intensity values that are different from the background by scanning them along a vertical line on the manuscript's left margin. Then, inside a 3×3 window, this pixel and its right-hand neighbors with comparable intensity form the support line.

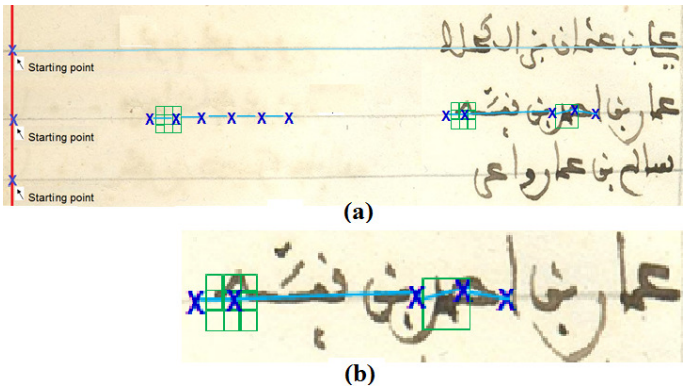


Figure 3: (a) Starting point detection, (b) Baseline extraction.

– As seen in Fig. 4(c)), all related components (*ccx*) are extracted: The process of labeling is completed (see Fig. 4(d)), and the *ccx* that are localized to both vertically adjacent text lines at the same time are further processed accordingly,

– As seen in Fig. 4(e), the interconnected text lines skeletons are retrieved,

– The intersection points close to the minima axis, representing the line midway between the two baselines (see Fig. 4(f)) are localized (see Fig. 4(g)): Note that intersection or junction points are pixels having at least three neighbors in the skeleton images where surrounding connected components are probably TCs. Unlike the Ouwayed and Belaïd's method [14], where junction points research is only made at the horizontal projection histogram profile's valley (minima axis), we have expanded the search to include the junction points in the vicinity of the middle line. Junction points in diacritic or noisy components are discarded,

– When using the junction point as the center, the TC is extracted where the TC's width and the TC's height are 120×120 pixels (see Fig. 4(h)).



Figure 4: Vertical TC detection and extraction: (a) Image with the detected first pixel used to initiate baseline computation, (b) Pre-processing stage, (c) Extraction of connected components, (d) Labeling of connected components, (e) Skeleton representation, (f) Computation of the middle line, (g) Detection of junction points, and (h) Extraction of the target TC.

3.1.2 Horizontally Interconnected Touching Character Extraction

The challenge is how to extract words when there are horizontally touching characters (TCs). Currently, there are no existing solutions addressing this issue. Word extraction in Arabic manuscripts is particularly difficult because words can be split into sub-words, some characters may not connect to the next one (see Fig. 5), and TCs are common. Fig. 5 illustrates, through three examples, how the presence of TCs affects the segmentation of sub-words.

Most previous methods for word extraction analyze the geometric relationships between adjacent components [34, 35]. First, they measure the white space between related portions that are identified as parts of Arabic words (PAWs). They then categorize these gaps as either gaps between words or gaps between word segments. Lastly, using this classification, they combine the recognized components into whole words. Therefore, it is essential to separate TCs in Arabic manuscripts before attempting word extraction, especially for methods that rely on inter- and intra-word spaces.

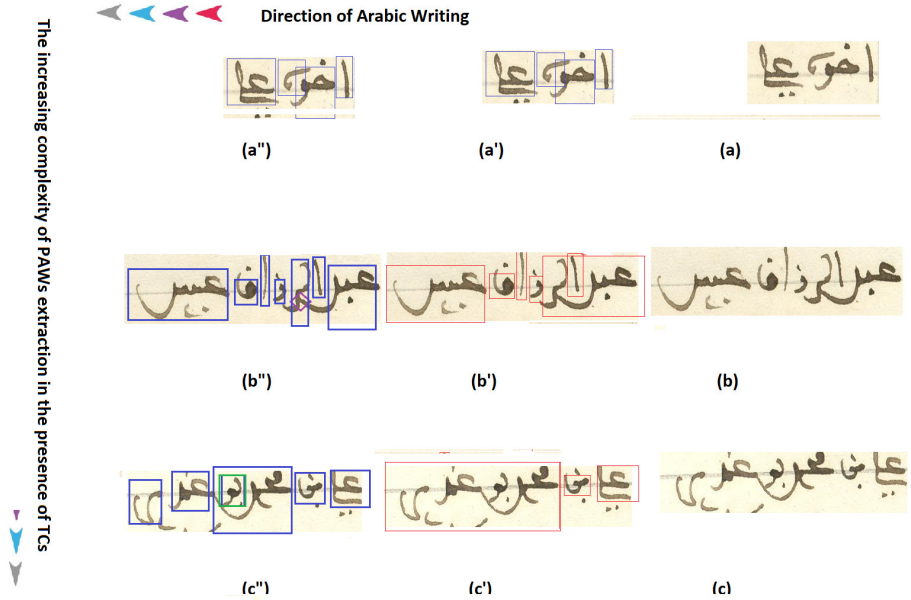


Figure 5: Impact of TCs on PAW Segmentation in Arabic Text Lines: (a) An input text line without TCs; (a') correct segmentation into 4 PAWs; (a") corrected segmentation yields 4 PAWs, as no TCs are present. (b) An input text line with one TC; (b') incorrect segmentation into 6 PAWs, where 2 PAWs are wrongly grouped due to the presence of a TC; (b") corrected segmentation yields 7 PAWs after TC separation. (c) An input text line with multiple TCs; (c') incorrect segmentation into 3 PAWs due to multiple mergings caused by TCs; (c") corrected segmentation yields 6 PAWs after TC separation.

For horizontally inter-word TCs, we propose a new method. Our observations of the Arabic manuscripts reveal that these TCs typically occur below the baseline. In ancient Arabic calligraphy, terminal letters often have a significant downward curve from left to right. Consequently, inter-word TCs usually form a convex curve that ends at the baseline. The suggested strategy comprises three phases, based on examining the convexity of the curve in the word's skeleton image: (i) baseline extraction, (ii) skeletonization of text lines, and (iii) detection of junction points.

- Baseline Extraction: As with vertically interconnected TCs, we used the line support found in the handled documents to determine the baseline.
- Text line skeletonization: Zhang and Suen's [36] thinning procedure is used to recover the lowest portion of the text line skeleton (see Fig. 6(b)). Around the junction point, a number of branches with a thickness of one pixel make up the skeletonized image. Following the identification of the intersection points (see Fig. 6(c)), the neighborhood can be investigated, and eventually the horizontally related TCs can be extracted, as will be discussed below.
- Junction Point Detection: Many existing TC extraction approaches rely on junction points in the skeleton image, assuming that each detected junction corresponds to a TC. However, this assumption is not always valid. As shown in Fig. 7, some junction points may belong to characters within the same word part (the

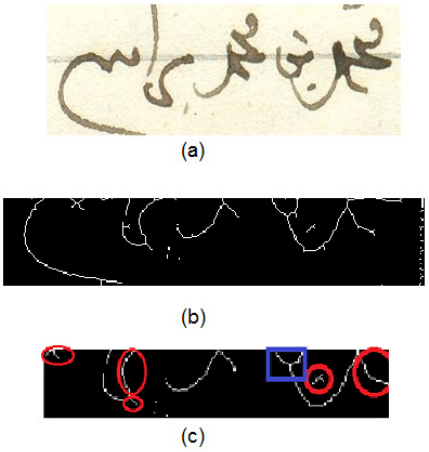


Figure 6: (a) Input image, (b) Bottom of the text line skeleton with multiple junctions, (c) Junction point detection: valid points in blue and invalid ones in red, based on branch convexity or length.

point farthest to the left in Fig. 7(c)) or even occur inside a single character (the point farthest to the right in Fig. 7(c)).

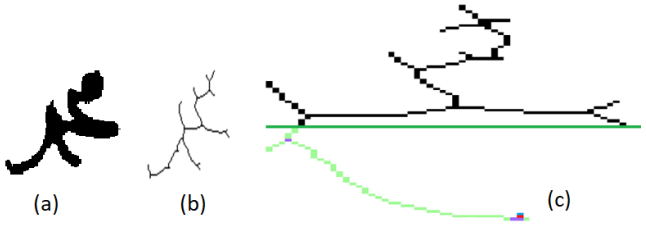


Figure 7: (a) Image of the word, (b)skeleton of the word image, (c) Two identified junction points.

To this end, we propose discarding invalid junction points by analyzing the terminal characters of Arabic words. Each branch connected to a junction point on the right is tested as follows:

1. Identifying every branch that is appropriately joined to every junction point,
2. Small branch removal: only branches of a specific size are kept, so noise and junction points in diacritical marks will be removed (see Fig. 6(c)),
3. Analysis of semi-convex branches: given that certain junction points connected to actual TCs are linked to branches that are slightly non convex (refer to Fig. 8), we determined that a branch, composed of a set of points, is related to a genuine TC only if a sufficient number of pairs (x, y) (empirically set at 310) satisfy the convexity equation (see Eq. 1). To examine its convexity, each branch B_r will be detailed and depicted as a continuous and bijective (we retain only a single pixel for each column) mathematical function.

   Let E represent a convex real vector space, and consider a function $Br : E \rightarrow R$. The function B_r is defined as convex if:

   $\forall (x, y) \in E \times E$ and $\lambda \in [0, 1]$:

$$B_r(\lambda x + (1 - \lambda)y) \leq \lambda B_r(x) + (1 - \lambda)B_r(y) \tag{1}$$

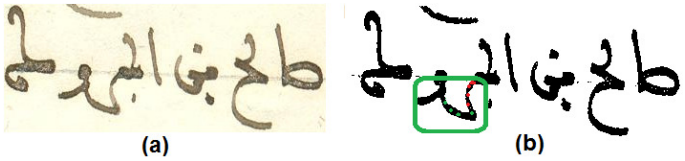


Figure 8: (a) Input image, (b) Example of a valid TC featuring a branch with both convex and concave shapes.

4. Only branches that meet the semi-convexity condition and terminate close to the baseline are retained (see Fig. 7(c), where a distant branch is ignored). A junction point is regarded as valid when at least one of its left-side branches reaches or approaches the baseline. The corresponding TC is then identified as the connected component centered around this junction point,
5. If needed, semi-convex branches are merged (see Fig. 9(c), (f)). The TC corresponds to the connected component centered on the leftmost junction point.

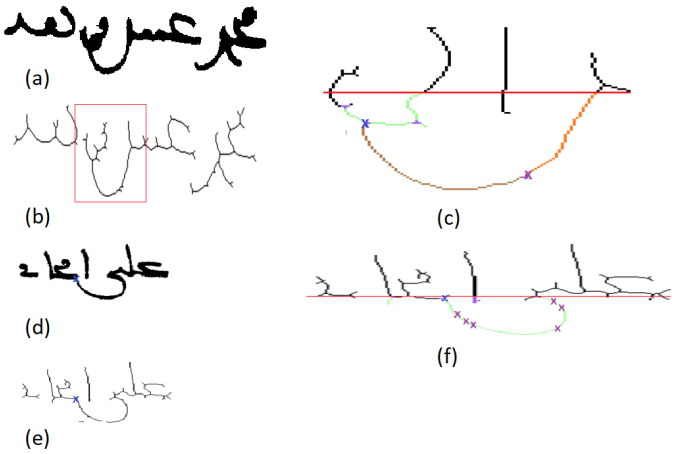


Figure 9: (a) and (d) Images of the text line. (b) and (e) Detection of junction points. (c)and (f) Resulting fusion curves.

In the subsequent discussion, we will concentrate on the procedures following TC extraction, presuming that the TC extraction techniques outlined in sections 3.1.1 and 3.1.2 are capable of extracting these ambiguous elements.

3.2 A Recognition-Based Method for TC Separation

Once the TCs have been accurately extracted, the second component of our system focuses on their actual separation. This is achieved through a recognition-driven strategy: for each TC, the system retrieves the Most Similar Template (MST) from a reference dictionary and transfers the corresponding separation information to the input TC using geometric alignment and refinement of the contour midpoints between character components.

3.2.1 Most Similar Template Selection:

The template that best matches a TC is selected by comparing it to all representative templates stored in the dictionary (Algorithm 1). The dictionary consists of templates with predefined separations, extracted at the

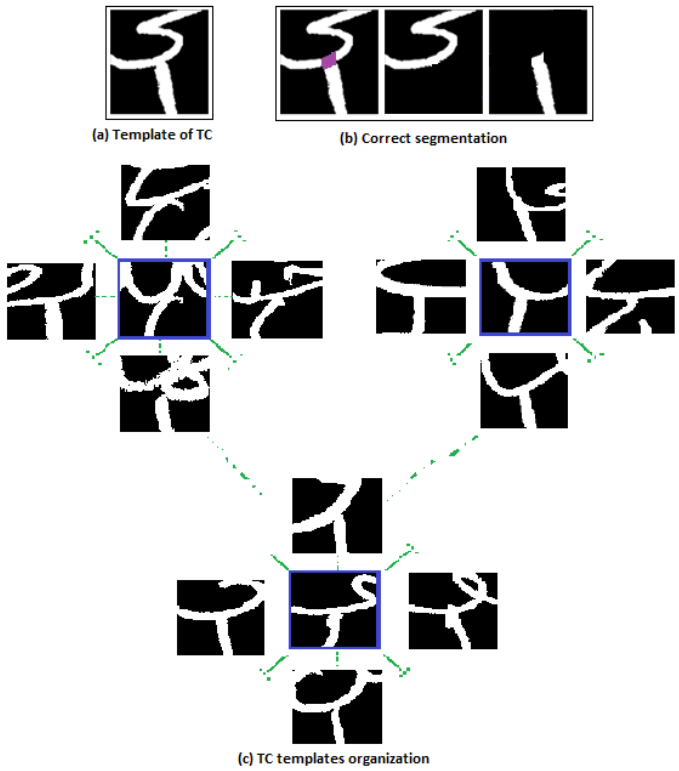


Figure 10: An entry of dictionary: (a) template of TC, (b) Correct separation (below figure), (c) TCs organized into clusters with a centroid template (in blue) using affinity propagation algorithm.

touching point (Fig. 10(a) and (b)). Template clustering is performed using the Affinity Propagation (AP) algorithm [37, 38], with a centroid representing each cluster (Fig. 10(c))

We approach the similarity computing between a template and a given TC, as a three-step process:

1. Make a bipartite graph matching between them using shape context descriptor [20].
2. Filter the result of correspondences to eliminate outlier couples by introducing region correspondences.
3. Estimate the overall distance as the sum of 1) Misalignment between corresponding points, 2) the alignment transform energy between matched points and 3) a term including the quality of the matching (see our proposed pattern similarity in Eq. 9).

a- bipartite graph matching: Each point p_i on the TC contour is paired with the most similar point q_j on the template contour by evaluating their shape context histograms (Fig. 11). The Canny edge detector [39] is used, and a predetermined number of edge pixels is chosen to simplify the matching.

For every pair (p_i, q_j), the matching cost is determined via the χ^2 test (Eq. 2), yielding a square cost matrix C of size N. The values $h_i(k)$ and $h_j(k)$ are obtained from the K-bin normalized distributions of points p_i and q_j using 5 bins for $log(r)$ and 12 bins for θ.

$$C_{ij} = C(p_i, q_j) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k) - h_j(k)]^2}{[h_i(k) + h_j(k)]} \tag{2}$$

The goal is to discover a permutation π that minimizes the total matching cost (Eq. 3) given the cost matrix C with entries C_{ij}. The Jonker and Volgenant algorithm [40] solves this in $O(N^2)$ time, assigning pixels between the TC and template.

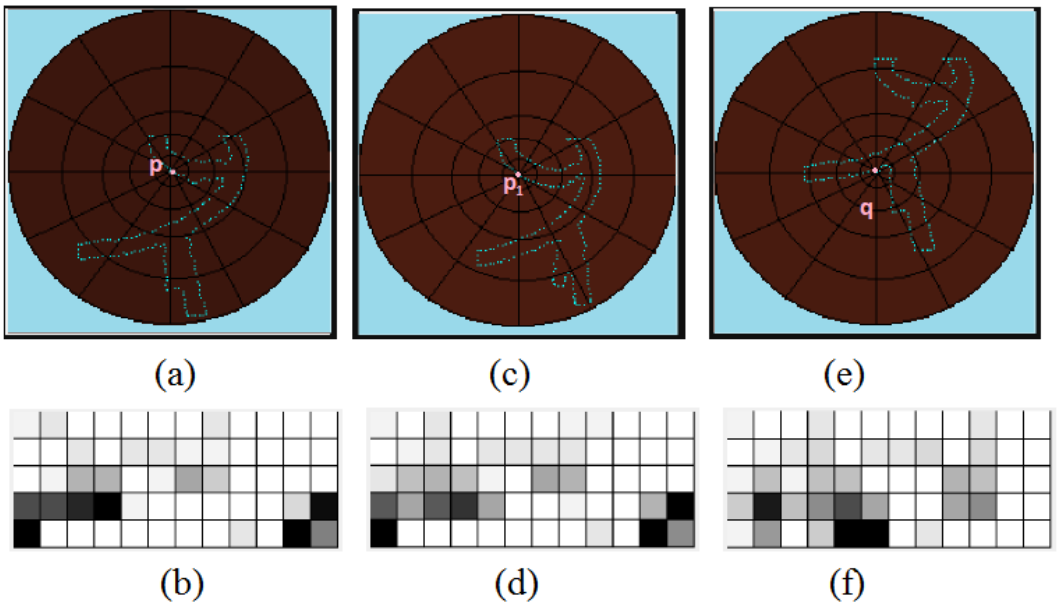


Figure 11: SC-based bipartite matching: (a) SC of p in TC_1,(b) log-polar histogram of p, (c) SC of $p1$ in TC_2, (d) histogram of $p1$, resembling that of p in (b), but differing from q in (f). The best matching is between p and $p1$.

$$C = \sum_i C(p_i, q_{\pi(i)}) \tag{3}$$

When working with small and intricate shapes like TCs, outliers can affect the matching result (Eq. 3). To address this, we use a new technique for pattern matching based on shape context descriptors in regions. This approach helps avoid the problem of outliers and provides a more accurate similarity measure between shapes.

b- bipartite graph matching based on regions: Belongie [20] and Chui [41] highlighted the issue of outlier pixels, proposing two solutions: 1) inserting fake nodes with a fixed matching cost, when the shapes have different numbers of contour points, and 2) introducing a mechanism to handle cases where the matching cost in the matrix C exceeds a predefined threshold. Outliers degrade matching quality and similarity accuracy. To address this, our approach relies on matching regions instead of individual points, considering each point's neighbors to remove doubtful matches.

To introduce region correspondences with the shape context descriptor, let E_1 denote the set of N matched point pairs (Eq. 2, [20]), and $G_{belongie}$ denote a bijection mapping each TC point p_i to its corresponding template point q_j, in accordance with Eq. 2. The Well-Matched Points (WMP) in E_2, used to establish region-level associations, form a subset of E_1 and include pairs (p_i, q_j) selected based on their neighborhood relationships.

In fact, for each matched pair (p_i, q_j) from E_1, we consider their respective k-nearest neighbors k-NN: $N_{TC}(p_i, k)$ and $N_{template}(q_j, k)$. A pair (p_i, q_j) belongs to E_2 if a subset of their corresponding neighbors (at least n out of k) are also matched together. E_2 can thus be defined as follows:

$$\begin{aligned}
&N_{TC}(p_i,k) \times N_{template}(q_j,k) \subseteq E_1,\\
&\text{with} \quad 1 \le i \le N,\ 1 \le j \le N,\\
&\text{and} \ \ \text{Card}\Big(G^{-1}_{belongie}(N_{TC}(q_j,k)) \cap N_{template}(p_i,k)\Big) \ge n,\\
&\text{where} \quad N_H(h,k) = \{\, h' \neq h \mid h' \in k\text{-NN}(h) \,\}.
\end{aligned} \tag{4}$$

In contrast to the matching approach proposed by Belongie [20], we consider well-matched points in E_2 (Eq. 4), based on a neighborhood relation. This introduces a constraint to filter out outlier points, ensuring that pairs (p, q) identified as outliers are discarded when determining the aligning transform function.

c- Estimating the Aligning Transform: The plane transformation between the template and the TC is estimated based on the bipartite graph matching result E_2. This transformation is then used to compute D_{be} (see Eq. 6) and D_{sc} (see Eq. 7). We used the Non-rigid Thin Plate Spline Warping [42].

Starting from the correspondence result E_2, the function f is computed over the entire R^2 domain according to:

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{N} w_i U(||P_i - (x, y)||) \tag{5}$$

where $U(r) = r^2 \log r^2$ and $\sum_{i=1}^{N} w_i = \sum_{i=1}^{N} w_i x_i = \sum_{i=1}^{N} w_i y_i = 0$, so that f(x,y) possesses square-integrable second derivatives.

The interpolating TPS is computed as the smoothest mapping that minimizes the bending energy term I_f.

$$I_f = \int\int_{\Re^2} ((\frac{\partial^2 f}{\partial x^2}) + 2(\frac{\partial^2 f}{\partial x \partial y}) + (\frac{\partial^2 f}{\partial y^2}))d_x d_y \tag{6}$$

d- Distance computing: Three parameters are taken into account in order to assess how similar two shapes are:

1. D_{sc}: the cost of the Shape Context (see Eq. 7);
2. D_{be}: this term quantifies the deformation energy required to align the template with the TC (see Eq. 6, where $T(.)$ denotes the estimated TPS).
3. N_{match}: the ratio between ($|E_2|$) and N (the total count of contour points) (see Eq. 10).

Let n and m denote the numbers of edge points of the two shapes P and Q, respectively:

$$\begin{aligned} D_{sc} = & \frac{1}{n} \sum_{p \in P} \arg\min_{q \in Q} C(p, T(q)) \\ & + \frac{1}{m} \sum_{q \in Q} \arg\min_{p \in P} C(p, T(q)) \end{aligned} \tag{7}$$

where

We modified the similarity metric proposed by Belongie (Eq. 8) by introducing an algebraic distance that takes into account the cardinality of the set E_2, as expressed in Eq. 9. The percentage of well-matched points, N_{match} (Eq. 10), is incorporated through an empirically coefficient (coeff), which was experimentally set to 0.2. Consequently, the most similar template corresponds to the one yielding the smallest distance from the input TC.

$$\text{Similarity} = -D \text{ and } D = D_{sc} + 0.3 D_{be} \tag{8}$$

$$\text{Similarity} = -D \text{ and } D = D_{sc} + 0.3 D_{be} - \text{coeff} * N_{match} \tag{9}$$

$$N_{match} = \frac{|E_2|}{N} \tag{10}$$

$|E_2|$ indicates the overall quality of the matching process. $N_{match} * coeff$, enhances the accuracy of the distance metric. A large E_2 indicates strong similarity between the template and the TC, resulting in a smaller distance. Conversely, when E_2 is small, the TPS transformation and bending energy have a limited impact, as they apply to fewer matching pairs. The factor $N_{match} * coeff$ is therefore used to balance the equation.

The curve in Fig. 12 compares the proposed similarity metric (Eq. 9) with that of Belongie et al. (Eq. 8). Several examples of TCs were tested with both equations (Eqs. 8 and 9), and in each case, they selected the same closest similar templates.

We remark that distances obtained using Eq. 9 (curve in blue) are less than those of Eq. 8 (curve in red), what marks the accuracy of the proposed equation. When observing the trend of the blue curve, according to the ratio N_{match} (see axis of abscissa), we conclude that if the ratio N_{match} approaches 1, the distance tends to -0.2 (as it means that the template and the TC are very close (D_{sc} and D_{be} approach zero). If N_{match} is small, the distance generally increases. But, sometimes it decreases, which means that the distance depends not only on the number of pixels to be interpolated (E_2's size), but also on their positions.

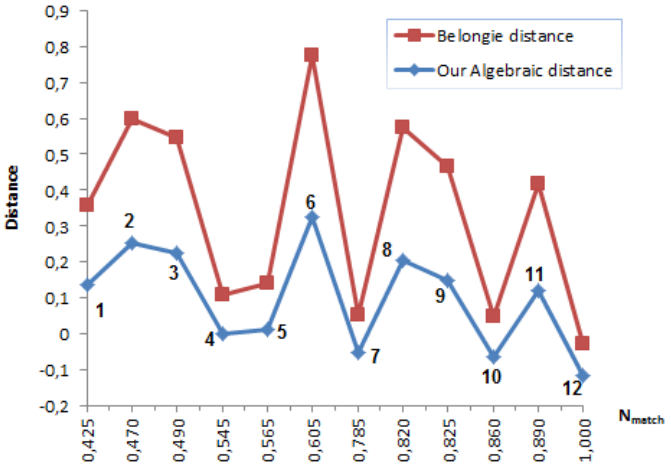


Figure 12: Distance comparison for some couples of TCs and their most similar template using Eq. 8 and 9.

With this modification, we observe that, when using Eq. 9, the similarity between TC and its most similar template rises, as opposed to when using Eq. 8. One example is seen in Fig. 13, where Eq. 8 of [20] is unable to find the most similar template, but Eq. 9, as we suggested, is successful in doing so.

By this adjustment, we note that the similarity between TC and its most similar template increases using Eq. 9, compared to its value using Eq. 8. In Fig. 13, we display one of some examples, where the use of Eq. 8 of [20] fails to identify the most similar template, whereas Eq. 9, as we proposed, succeeds to selecting it.

Table 3 lists the parameters employed in selecting the most similar template and Fig. 14(a) and (b) illustrate a matching relation between a template and a TC using both methods (Belongie et al. as a point-to-point matching and our method as a region-to-region matching) to calculate similarity.

Table 3: Region-based matching: parameters used.

Parameter	Value	Understanding
N	200	Cardinality of contour pixels for both the TC and the template.
k	30	Cardinality of nearest neighbor points for each p_i and q_j.
n	10	Least number of matched pairs within the neighborhoods of p_i and q_j required to be included in E_2.

We set N to 200, instead of 100 (value chosen by Belongie et al. in [20]) for two reasons: 1) it faithfully reflects the structure of the TCs and 2) When employing Eq. 4, $|E_2|$ drops while maintaining its significance.

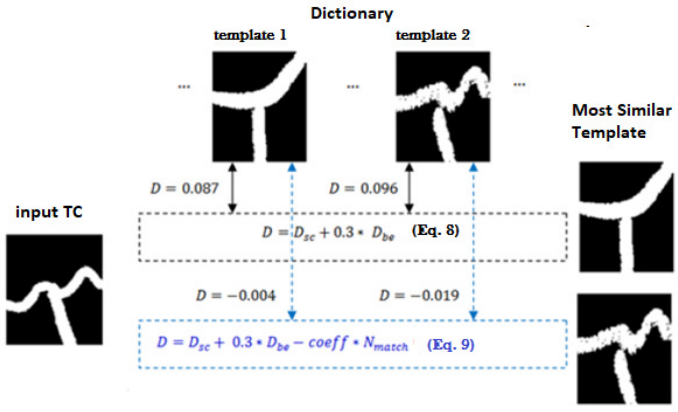


Figure 13: Distances comparison when selecting the most similar template.

We fixed k to 30 after comparing a number of neighbor values. This is because, as Fig. 14 illustrates, if k below 30, $|E_2| \rightarrow$ to 0, and if k above 30, $|E_2| \approx |E_1|$). n is set at 10 since, below this amount, the idea of area matching loses its relevance. If n rises, E_2 may be empty and $|E_2|$ may become closer to zero.

3.2.2 Touching Character Separation:

a- Shape description: Let $P = \{p_1, p_2, \ldots, p_N\} \subset E \subset R^2$ be a set of points in a two-dimensional space, where each point p_i is represented by its coordinates. To quantify distances between points, we introduce a metric D, which can be Euclidean, Manhattan, or Canberra distance. A central point can summarize the spatial distribution of P. The *centroid* C minimizes the sum of squared distances to all points:

$$C = \arg\min_{x \in E} \sum_{i=1}^{N} D(x, p_i)^2.$$

In contrast, the *midpoint* M (or geometric median) is defined as the point minimizing the sum of distances (not squared distances) to the points in P:

$$M = \arg\min_{x \in E} \sum_{i=1}^{N} D(x, p_i).$$

Computing M generally requires a search over the space E, as no closed-form solution exists for arbitrary metrics.

b- Separation using Midpoints: We currently have 1) the most closest template (MST), with its proper division (parts A and B), and 2) the inferred TPS parameters Ψ used to warp the template into the TC. We can conclude that parts A and B can be warped into their corresponding positions in the TC using the same TPS parameters Ψ, as shown in Eq. 11.

$$\begin{aligned} TPS(\Psi, MST._{Part_A}) &\approx TC._{Part_A} \\ TPS(\Psi, MST._{Part_B}) &\approx TC._{Part_B} \end{aligned} \tag{11}$$

After the transformation with the inferred TPS parameters, let's look at (P_a, P_b) the contours of the template parts and (C_a, C_b) their corresponding central points. These core locations roughly correspond to the portions of TC. The suggested approach assigns pixels to the nearest central point in order to recreate the TC's components. Pixels in the touching area may be wrongly allocated as shown in Fig. 16(a). To avoid that, we needed

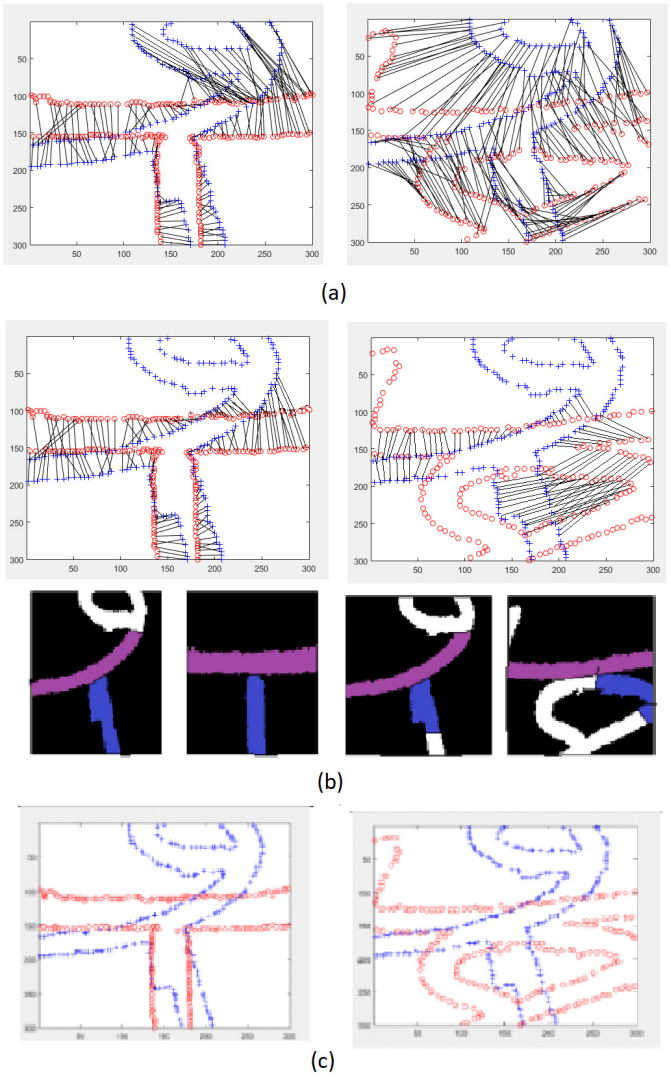


Figure 14: behavior of E_2 depending on the neighborhood region k for 2 samples of TCs and their most similar templates: (a) the matching result for k above 30 (we see that $|E_2|$ tends to $|E_1|$, (b) the matching result for k between 25 and 30 (we see that $|E_2|$ is acceptable and significant), (c) the matching result for k below 25 (we see that $|E_2|$ tends to zero.

to adjust central points before using them. The intersection points (I_a, I_b) between the segment L, which connects the central points C_a and C_b of the two parts (P_a, P_b) are firstly identified as indicated in Eq. 12. For each intersection point, the method computes a new central point as the symmetric of the closest one to the intersection (see C'_b in Fig. 15(a) or C'_a in Fig. 15(b)) (C_a is mapped to C_a': the reflection of C_b across I_b on L) and (C_b is mapped to C'_b: the reflection of C_a across I_a on L). Based on that, pixels are assigned to TC's parts A and B will be performed (see Fig. 16(b) and Algorithm 2).

$$I_a = L \cap P_a \text{ and } I_b = L \cap P_b \tag{12}$$

Algorithm 1 MST selection

Input:

- The input TC
- Dictionary organized into p clusters with AP algorithm according to the similarity matrix (Eq. 9)
- $template_i$: the centroid of cluster i.

Output:

- MST with its correct separation ($part_A$ and $part_B$)
- Ψ: the estimated parameters that warps the most similar template into the TC

$Min_{distance} \leftarrow +\infty$;
for $i \leftarrow 1$ **to** p **do**
  $D \leftarrow D_{Eq.9}(TC, template_i)$;
  if $D < Min_{Distance}$ **then**
    $Min_{distance} \leftarrow D$;
    $MST \leftarrow template_i$;
    $\Psi \leftarrow TPS_{parameters}(template_i, TC)$;
  end if
end for

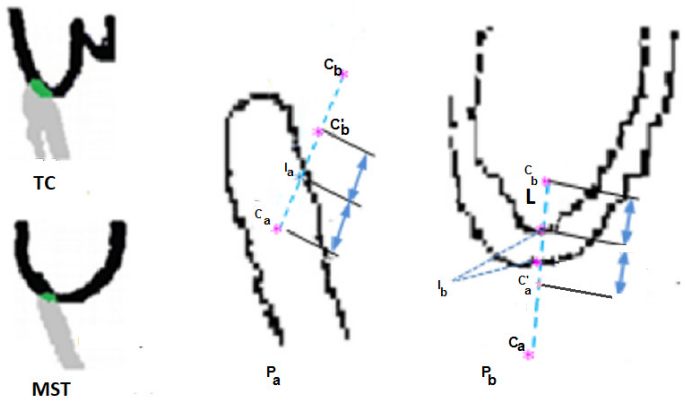


Figure 15: (a) the contour of the template's part A transformed with TPS parameters, (b) the contour of the template's part B transformed with TPS parameters.

4 Experiment Results and Discussion

4.1 Used Dataset

The TC dataset was extracted from ancient and degraded manuscripts from the 19th century, as shown in Fig. 17(a). Each text line consists of sequences of Arabic personal names. The manuscripts were binarized using several algorithms:

- Bernsen's technique (Fig. 17(b)) produces good results but has noise and high computational time.

Algorithm 2 TC Separation

Input:

- The input TC
- The MST with its correct separation ($Part\text{-}A$ and $Part\text{-}B$)
- Ψ: the parameters that warp the MST into the TC.

Output:

- TC's $Part\text{-}A$
- TC's $Part\text{-}B$
- TC's Shared-zone

$P_a \leftarrow TPS(\Psi, Part\text{-}A\text{'s contour})$
$P_b \leftarrow TPS(\Psi, Part\text{-}B\text{'s contour})$
$C_a \leftarrow CentralPoint(P_a)$
$C_b \leftarrow CentralPoint(P_b)$
$L \leftarrow Segment(C_a, C_b)$
$I_a \leftarrow L \cap P_a$
$I_b \leftarrow L \cap P_b$
for each pair (I_a, I_b) **do**
  Search for the nearest couple (C'_a, C'_b) to (C_a, C_b)
  $C'_a \leftarrow C_a$
  $C'_b \leftarrow C_b$
  Extract $TC.Part\text{-}A$
  if $Distance(C'_a, I_b) < Distance(C'_b, I_b)$ **then**
    $C'_b \leftarrow symetric(C'_a, L, I_b)$
  else
    $C'_a \leftarrow symetric(C'_b, L, I_b)$
  end if
  Place C'_a and C'_b on the TC
  $TC.Part\text{-}A \leftarrow$ {pixels near C'_a compared to C'_b}
  Extract $TC.Part\text{-}B$
  if $Distance(C_b, I_a) < Distance(C_a, I_a)$ **then**
    $C''_a \leftarrow symetric(C_b, L, I_a)$
  else
    $C''_b \leftarrow symetric(C_a, L, I_a)$
  end if
  Place C''_a and C''_b on TC
  $TC.Part\text{-}B \leftarrow$ {pixels near C''_b compared to C''_a}
  $TC.Shared\text{-}zone \leftarrow TC.Part\text{-}A \cap TC.Part\text{-}B$
end for

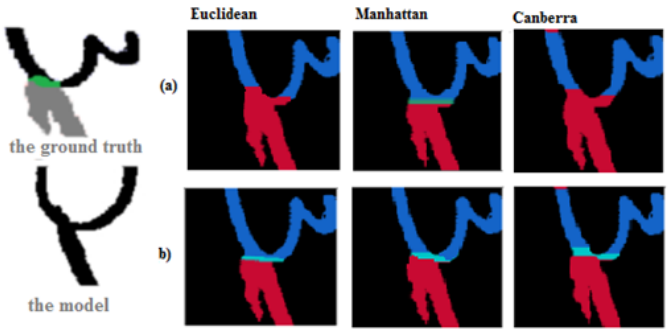


Figure 16: TC separation results: (a) Without centers adjustment, (b) With centers adjustment.

Algorithm 3 Separation of Touching Characters

Input: Manuscripts, an input TC to separate
Output: TC's Part A, TC's Part B, and TC's Shared Zone
Begin:
Vertically TCs ← Apply Vertically Interconnected Touching Character Extraction
Horizontally TCs ← Apply Horizontally Interconnected Touching Character Extraction
dataset TCs ← Vertically TCs ∪ Horizontally TCs
Similarty-Matrix ← Computing-Distance between all pairs of Dataset with Eq. 9
Clustered Dictionary ← Apply Affinity Propagation (Similarity-Matrix)
MST ← Apply MST Selection(Clustered Dictionary, TC)
TC's Part A, TC's Part B, Shared Zone ← Apply TC Separation(MST, TC)
End

- Mean technique (Fig. 17(c))
- The technique proposed by Niblack (Fig. 17(d))
- The technique proposed by Otsu (Fig. 17(e))
- The technique proposed by Sauvola [43]

The highest quality results were obtained using Sauvola's method, combined with a morphological operation to recover thinned characters (Fig. 17(f)).

We created a TC dataset using TCs and their correct separations. Each correct separation splits a TC into two parts: part A and part B, which belong to the first and second characters, respectively. It is important to note that no Arabic ground truth dataset exists, apart from that of Kang and Doermann [1]. This dataset consists of a localized bitmap (size 120×120) with touching strokes, covering only the connected components shared by two disjoint characters (Fig. 18(a)). To expand this database, we:

- added real connections from the manuscripts (Fig. 18(b)),

- applied two algorithms: one randomly substitutes parts A with different parts B to create new TCs, and the second performs rotations and deformations of part B while associating it with part A and vice versa (Fig. 18(c)),

- reduced the interlines and added the resulting TCs (Fig. 18(d)). We built the template dictionary from these TCs and their correct separation.

4.2 Performance Evaluation and Discussion

The experiments are threefold, focusing on: 1) TC extraction, 2) most similar template selection, and 3) TC separation. However, testing the performance of the most similar template selection is subjective, as the goal

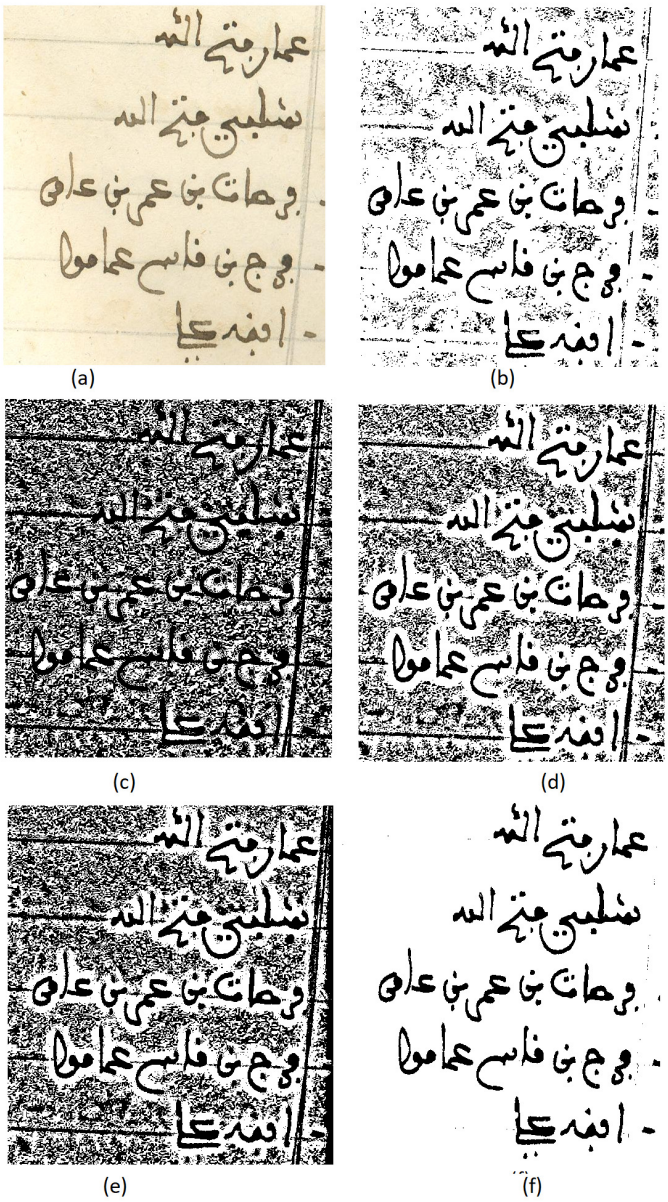


Figure 17: Document binarization using different techniques: (a) Input sequences of Arabic personal names, (b) Bernsen, (c) Mean, (d) Niblack, (e) Otsu, (f) Sauvola

is to find the closest template to a TC rather than an identical one. The notion of "closest template" can only be assessed in a specific context, which in this case is TC separation. Thus, the effectiveness of the recognition system heavily relies on the evaluation of the separation step. To evaluate the extraction of vertically interconnected TCs, we manually selected 1500 handwritten Arabic text lines that include 500 TCs. We achieved a TC extraction rate of 94%. The 6% error rate is primarily due to limitations in the junction point detection algorithm. In some cases, the touching region does not lie near the minima axis, or the detected junction points may be invalid or non-unique.

To evaluate horizontally interconnected TCs, we manually selected a separate set of 1,500 handwritten text lines, distinct from those used for vertical TCs. This subset includes 600 horizontally connected TCs, and the extraction method achieved an accuracy of 94.2%.

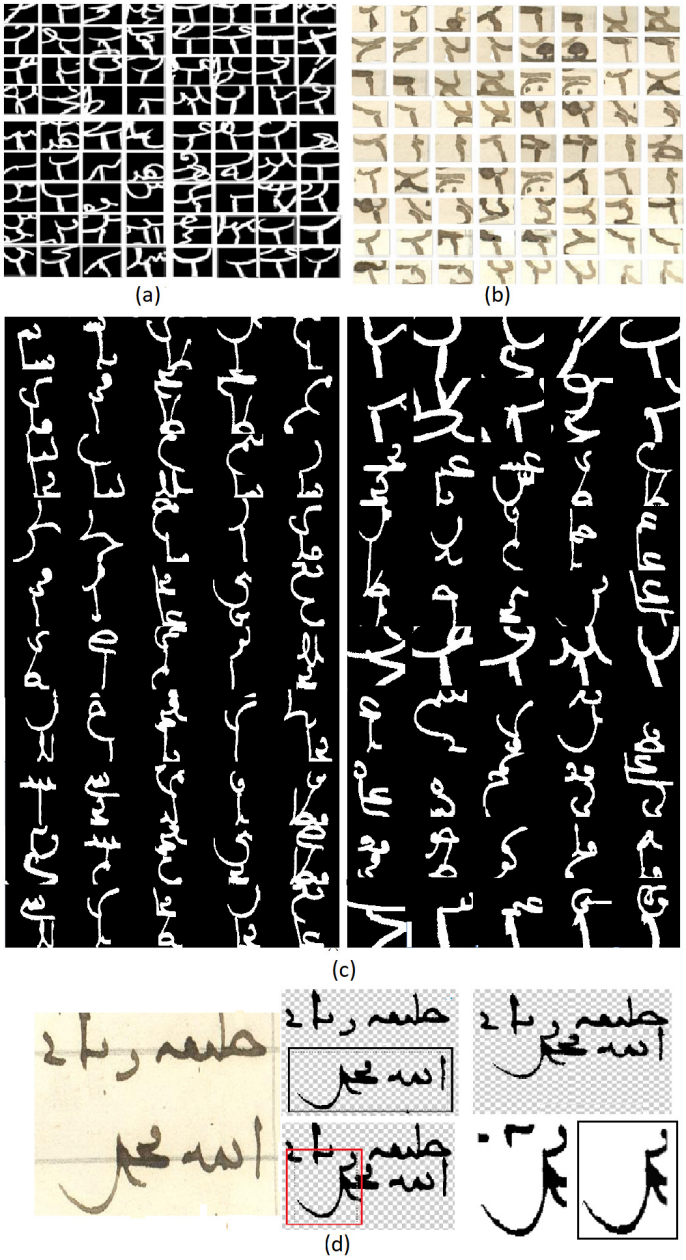


Figure 18: Samples from the used database: (a) samples from Kang and Doermann dataset [1], (b) TCs before binarization, (c) TCs obtained by applying 1) algorithms and 2)A techniques based on convexity curve analysis, (d) The process of TC's extraction by reducing inter-lines space.

The error rate arises from:

- Issues arising in the detection of junction points.
- The fixed cutoff used to evaluate the semi-convexity curves at junction points. The method only considers semi-convex right branches of junction points, discarding left-side branches that may correspond to actual TCs.

We have not been able to compare our results to other works, as there are no existing studies on horizontally

interconnected TC extraction in Arabic manuscripts. In this study, we focused on documents that are clearly non-oriented and produced by a single writer, while evaluation on multi-writer and multi-orientation collections remains an avenue for future work. For the closest template selection (MST), each TC is compared with the representative element of each cluster, utilizing a thread library [44] to enhance efficiency and reduce runtime. The computational complexity of the AP used to create the template clusters is anticipated to grow like $O(N^2)$. It suggests the pair distance matrix S. However, when some AP related parameters are fixed, such as the cluster number, maximum iteration number, and damping factor, the runtime for a 500×500 similarity matrix is about 33 seconds. Note that the clustering with AP is done only once, using a dataset of 500 samples, the most representative ones.

Recall that to match a TC to a template, we have used the Jonker and Volgenant heuristic [40], which takes almost $10ms$ for a 200×200 costs C_{ij} matrix. Now to organize 500 templates into clusters, we need 21.56 min to make clusters. This time involves 21.23 min to establish matrix similarity.

In the TC separation step, the runtime is proportional to cluster number and turns around 23.33 seconds.

For the TC separation accuracy, the evaluation protocol is the same as [32] where a separation with a Match-Score [45] above 0.80 is considered correct.

Pixels within the ground truth segments are denoted by (G_A, G_B), while (R_A, R_B) denotes the sets of points within the result segments. The function $area(s)$ returns the number of elements in the set s. The matching result for the i-th ground truth and result region is denoted by MS_i.

$$\text{MatchScore} = \frac{2 * MS_A * MS_B}{MS_A + MS_B} \tag{13}$$

$$MS_i = \frac{area(G_i \cap R_i)}{area(G_i \cup R_i)} \tag{14}$$

The suggested separation technique is tested using the Euclidean, Manhattan, and Canberra distances, where the gravity center and the midpoint, respectively, characterize the distribution of the point. We used 1100 TCs for that, with a distribution of 600 for TCs that are horizontally connected and 500 for TCs that are vertically connected. Based on the three metrics, the results are regarded as satisfactory (see Table 4). To distinguish these distances, we note that:

- The segments' shapes, as determined by the Manhattan distance, are closer to their ground truth.
- Using the Canberra distance, one of the two components is split into two fragments (see the red fragment above the blue region in the Canberra column of Fig. 16).
- Separating the TC using the Euclidean distance is similar to slicing along the line joining the two centroids of the template's component parts

It seems that Manhattan distance is the most appropriate because the TC's image looks like a grid plan (see Fig. 16 Manhattan column). Furthermore, a comparative analysis with existing methods summarized in Table 5 confirms that our system offers robust and reliable performance, making it a compelling approach for separating both inter-line and inter-word TCs. It is important to highlight several key points when comparing with related work: 1) Few studies address TC separation in Arabic manuscripts, 2) Most existing research focuses on specific manuscripts or categories of TCs, making comparisons impractical, especially for TCs that appear between ascending and descending text lines, 3) Methods for vertically interconnected TCs differ significantly from those for horizontally interconnected TCs, 4) Separation accuracy is not always assessed consistently, particularly regarding the Match Score function [45], and 5) There is currently no standard or publicly available TC database for meaningful comparison.

In this work, we just compared our work with the most related ones: the Kang and Doermann's method [1] and the Ouwayed and Belaïd's method [14] for vertically interconnected TCs separation. For horizontally interconnected TCs separation, we compared with Boukerma and Farah's method [3]. To this end, we have

implemented these methods. Carried experiments on the used dataset and according to Match Score function [45], illustrate the effectiveness of the proposed method. In fact:

1. The accuracy of the Kang and Doermann approach [1] was around 71.4%. There are four types of errors: 1) bad template, 2) ambiguity, 3) disturbing components, and 4) the TPS's incapacity to mark corresponding components very accurately.

   These issues are caused by the steps that precede the TC separation, particularly the TC extraction and the most similar template selection (the recognition and transformation steps).

2. the Ouwayed and Belaïd's method [14] reached an accuracy of about 96.88% . Errors are due to junction point missing and when dealing with complicated TCs.

3. our method reached an accuracy about 94%. Errors are due to bad template and to the aligning process (see Fig. 21). Our method effectively addresses the issues highlighted by Kang and Doermann [1], as illustrated in Fig. 20. Errors due to bad template are solved when well exploiting the pixel distribution of the most similar template parts and their central points. The issues arising from ambiguity and transformation deviations are largely addressed, with only a few errors observed (see Fig. 20(d), resulting outputs). This is achieved through region-to-region matching, which enhances the comparison by detecting similarities between the constituent parts of the two shapes.

4. The proposed deep learning separation method [25], is applicable for connections between successive text lines and between text line words.This method achieved an accuracy rate of 94.6% identical to our method.

When testing with by Boukerma and Farah's method [3], we find that our method achieves a superior accuracy: 92% in comparison to 60%.

Due to the absence of a ground truth TC dataset, our approach was additionally evaluated on a small set of 200 Latin TCs (see example in Fig. 19). The results appear satisfactory; however, testing on a proper ground truth dataset remains necessary.

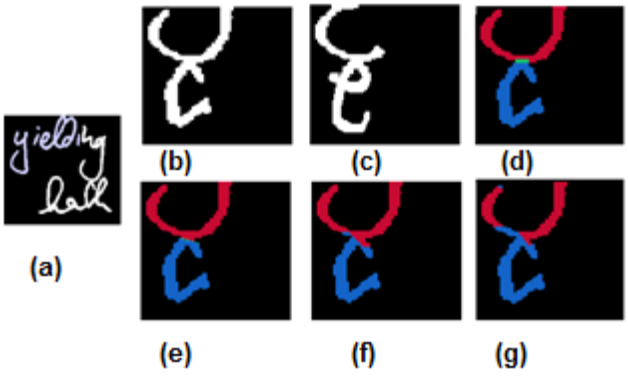


Figure 19: (a) Manuscript used in [15], (b) TC, (c) most similar template, (d) Ground truth, (e), (f) and (g) are separation rate with Euclidean (MS=0.98), Manhattan (MS=0.95) and Canberra (MS=0.92) distances.

Examples of TCs where our method and that of Kang and Doermann [1] achieved a very good score are shown in Fig. 22. The suggested method's failure case is illustrated in Fig. 22 (line 4: our results). The TC was poorly separated even though the separation rate was over 85% because the character in below s¿ (sin) appears to be closer to m¿ (mim).

In sum, the key points of our method are:

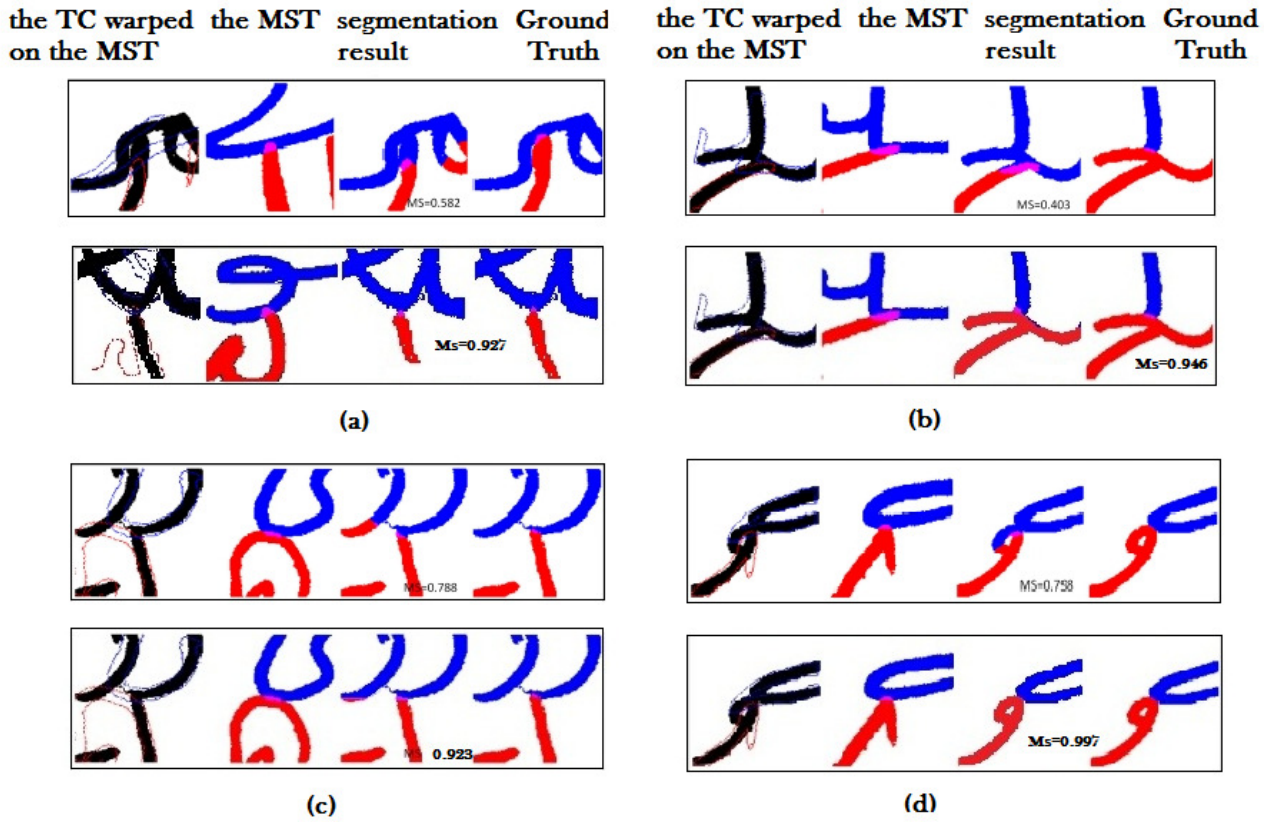


Figure 20: The four kinds of problems related to [1] method followed by our obtained results are: (a) Bad template selection, (b) Ambiguity, (c) Disturbing components, (d) Transform deviation.

Table 4: Test Results with MatchScore Threshold upper than 0.80.

Metric	Euclidean	Manhattan	Canberra
Correctly separations rate	92.6%	94%	92%

- region matching: the most similar template is selected based on region matching. This would avoid us from choosing a farther away one.

- a simple research: In order to improve run time, we compare the input TC only to the cluster centroid, to find the most similar template. We hope to find only the closest template, but not a very closest one,

- a robust separation method: the separation method proposed here, is not so sensible to the shape recognition because, even if the most similar template for an input TC was not found; the method generally succeeds to correctly separating it. Thanks to centers adjustment, where each part seeks its pixels in the other one.

In Fig. 23, we summarized all the followed steps to separated the TC in Fig. 23(a) using Euclidean distance with gravity center and Manhattan distance with midpoint.

As our work is inspired by that of Kang and Doermann [1], where both are recognition-based methods using a template dictionary, we summarized their differences in table 6 as follows:

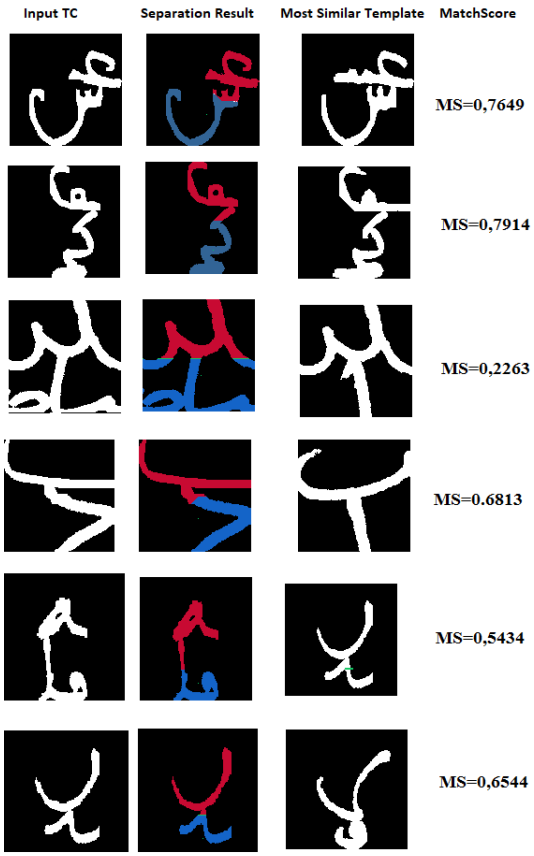


Figure 21: Some Failure Cases of Touching Character Separation.

5 Conclusion

In this work, we proposed novel methods for the extraction and separation of touching characters (TCs) in Arabic manuscripts. The main contributions of our approach are summarized as follows:

- We introduced a novel method for extracting vertically and horizontally interconnected TCs, relying on morphological analysis of terminal Arabic word letters.
- A region-to-region matching approach was introduced for template selection, addressing the issue of outlier pixels and ensuring better identification of the most similar template.
- To optimize runtime, the TC was compared only to cluster representatives, and parallel processing was employed to further reduce execution time.

Table 5: Performance comparison of TC separation methods.

Method	**Database**	**Total of TCs**	**Correct Separations Rate(%)**	
			Vertical TCs	**Horizontal TCs**
[14]	TCs images extracted from 100 private documents	640	96.8	–
[46]	TCs images extracted from AHDB and other Datasets	220	97.2	–
[25]	TCs images extracted from Kang Dataset	744 for training and 744 for testing	94.6	–
[1]	Kang Dataset	744 for training and 744 for testing	71.4	–
Our Method	Our Dataset	500 inter-line TCs and 600 inter-word TCs	94.0	94.2

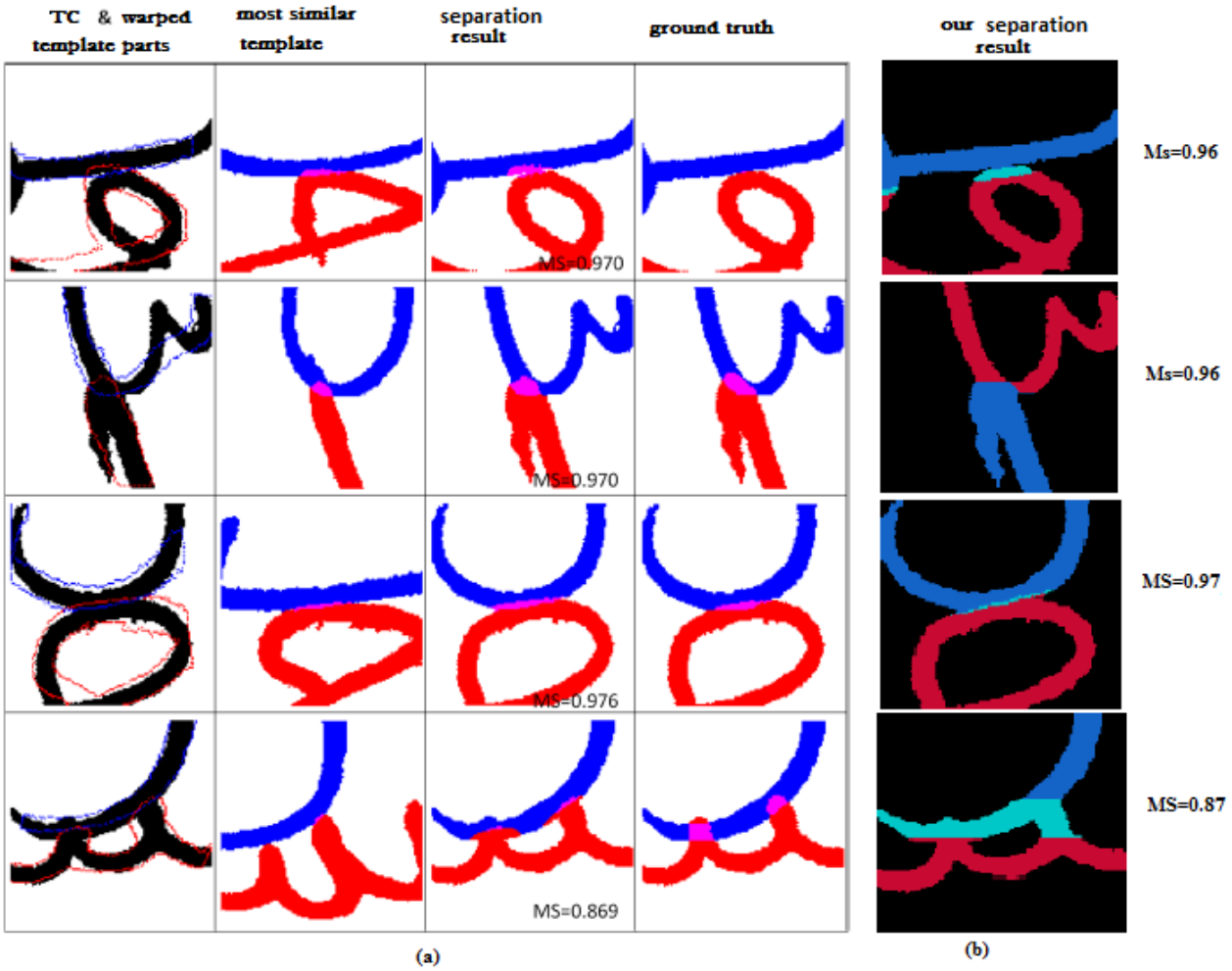


Figure 22: (a) Examples of TCs with MatchScore ≥ 0.8 separated with Kang's method, (b) Results with our method.

- We proposed a recognition-based separation method, selecting the most similar template for a TC and using it for separation.
- The proposed separation method does not require a perfectly matching template. Even without identifying the most similar template, the method successfully separates the TC.
- We adjusted the center points of the template parts to accurately determine where the TC should be separated.
- We demonstrated the script independence of our method by testing it on Latin TCs, achieving promising results.

The proposed TC separation method offers several advantages, including improved run-time performance and the ability to handle complex multi-touching cases, which are particularly challenging. This work provides a comprehensive discussion of the methodologies involved, with the aim of offering clear guidelines for future research in the field.

Although the TC separation approach is tested on Latin TCs, further validation on other scripts with diverse characteristics is necessary to ensure broader applicability.

Table 6: Differences between Kang and Doermann's method [1] and our method

Kang and Doermann's method	**Our Method words**
MatchScore used	
0.8	0.8
Dictionary's structure	
A static dictionary.	A dictionary that automatically grows by adding uncommon TCs whose distance from all existing templates is greater than 0.22.
TC's type	
Only vertically interconnected words	Horizontally and vertically interconnected words
TC's script	
Only Arabic script.	Arabic and Latin script.
The most similar template	
TC is compared to all clusters centroid and then in the cluster itself. Method needs a perfect alignement	the TC is only compared to the cluster centroid.
Matching method	
Point-to-point matching as proposed by Belongie et al.	Region-to-region matching
Outliers pixels	
All pixels are matched	Only well matched pixels. Outliers pixels are discarded.
Similarity metric	
Belongie's similarity metric: Eq. 8.	Eq. 9.
Separation method	
The Kang and Doermann method relies on aligning the most similar template to the input TC, but poor alignment often leads to incorrect separation results.	The key idea is that, after transformation, the two parts of the most similar template share approximately the same midpoints as the TC. This alignment is achieved through the proposed region-to-region matching, which emphasizes corresponding components during transformation. The TC is separated by assigning each pixel to the nearest midpoint. For pixels in the shared region, the midpoints of the two template parts are first adjusted before assignment.
Qunatitive Evaluation	
740 inter-line TCs.	500 inter-line TCs and 600 inter-word TCs.
Correctly Separations Rate (%)	
71.4 for inter line.	94 for inter line (vertically connected characters and 94.2 for inter word (horizontally connected characters.
Failure	
1) no proper template can be found, 2) the most similar template may not be the correct one, 3) Disturbing components interfere with matching, and 4) Although a similar template is found, the TPS transformation may fail to accurately align the corresponding components due to outlier pixels.	1) No proper template can be found for unusual input TCs, and 2) when the TC is a complex multi-touching component.

Several avenues for future work could further improve the performance and applicability of the proposed method:

- Enhancing the robustness of the extraction step by incorporating deep learning-based techniques, which may offer more adaptive features for different scripts and degraded documents.

- Expanding the method's evaluation on a wider range of historical and modern manuscripts from various languages and scripts to confirm its scalability and effectiveness in real-world applications.

We hope that the insights presented in this study will be beneficial for researchers working on similar problems and inspire further improvements in manuscript separation techniques.

References

[1] L. Kang, D. Doermann, "Template-based segmentation of touching characters in handwritten text lines", *in Proc. of the International Conference on Document Analysis and Recognition*, Beijing, China, 417-422,

2011. https://doi.org/10.1109/ICDAR.2011.120.

[2] A. Alaei, U. Pal, P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation", *Pattern Recognition*, 44(4):917-92, 2011. https://doi.org/10.1016/j.patcog.2010.10.014

[3] H. Boukerma, N. Farah, "PAW-IFN/ENIT: une nouvelle base de pseudo-mots arabes pour une approche de reconnaissance pseudo analytique", *AR IMA*, 1-9, 2012.

[4] L. Likforman-Sulem, C. Faure, "Une méthode de résolution des conflits d'alignements pour la segmentation des documents manuscrits", *Traitement du Signal*, 12(6), 541-549, 1995.

[5] K. Takru, G. Leedham, "Separation of touching and overlapping words in adjacent lines of handwritten text", *Proc. of the Eighth International Workshop on Frontiers in Handwriting Recognition*, Ontario, Canada, 496-501, 2002. https://doi.org/10.1109/IWFHR.2002.1030959.

[6] U. Pal, S. Datta, "Segmentation of Bangla unconstrained handwritten text", *Proc. of ICDAR* 1128-1132, 2003. https://doi.org/10.1109/ICDAR.2003.1227832.

[7] A. Zahour, B. Taconet, S. Ramdane, "Contribution aĺa segmentation de textes manuscrits anciens", *Proc. of CIFED*, La Rochelle, 2004.

[8] N. Tripathy, U. Pal, "Handwriting segmentation of unconstrained Oriya text", *Proc. of the Ninth International Workshop on Frontiers in Handwriting Recognition, Kokubunji*, Tokyo, Japan, 306-311, 2004. https://doi.org/10.1109/IWFHR.2004.50.

[9] J. Kumar, L. Kang, D. S. Doermann, W. Abd-Almageed, "Segmentation of handwritten text lines in presence of touching characters", *in Proc. of the International Conference on Document Analysis and Recognition*, Beijing, China, 109-113, 2011. https://doi.org/10.1109/ICDAR.2011.31.

[10] A. Lemaitre, J. Camillerapp, B. Couâsnon, "A perceptive method for handwritten text segmentation", *Proc. of Document Recognition and Retrieval XVIII - Electronic Imaging*, San Francisco, United States, 100-108, 2011. https://doi.org/10.1117/12.873037.

[11] L. Likforman-Sulem, A. Zahour, B. Taconet, "Text-line segmentation of historical documents: a survey", *International Journal on Document Analysis and Recognition*, 10(2), 123-138, 2007. https://doi.org/10.1007/s10032-006-0023-z.

[12] E. Bruzzone, M. C. Coffetti, "An algorithm for extracting cursive text-lines", *Proc. of International Conference on Document Analysis and Recognition*, 749-752, 1999. https://doi.org/10.1109/ICDAR.1999.791896

[13] L. Kang, D. S. Doermann, H. Cao, R. Prasad, P. Natarajan, Local segmentation of touching characters using contour-based shape decomposition, *in Proc. of the 10th IAPR International Workshop on Document Analysis Systems*, Gold Coast, Queensland, Australia, 460-464, 2012. https://doi.org/10.1109/DAS.2012.53.

[14] N. Ouwayed, A. Belaïd, "Separation of overlapping and touching lines within handwritten Arabic documents", *in Proc. of the 13th International Conference on Computer Analysis of Images and Patterns*, Mˆunster (North Rhine-Westphalia), Germany, 237-244, 2009. https://doi.org/10.1007/978-3-642-03767-2_29

[15] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, "Text-line and word segmentation of handwritten documents", *Pattern Recognition*, 42(12), 3169-3183, 2009. https://doi.org/10.1016/j.patcog.2008.12.016.

[16] P. Vassilis, S. Themos, K. Vassilis, C. George, "Handwritten document image segmentation into text lines and words", *Pattern Recognition*, 43(1), 369-377, 2010. https://doi.org/10.1016/j.patcog.2009.05.007.

[17] S. Rohini, R. S. Uma Devi, S. Mohanavel, "Segmentation of touching, overlapping, skewed and short handwritten text lines", *International Journal of Computer Applications*, 49(19), 24-27, 2012. https://doi.org/10.5120/7877-1163.

[18] D. Fernandez-Mota, J. Lldos, A. Fornes, "Graph based approach for segmenting touching lines in historical handwritten documents", *International Journal on Document Analysis and Recognition (IJDAR)*, 17(3), 193-312, 2014. https://doi.org/10.1007/s10032-014-0220-0.

[19] Y. A. Chen, G. Leedham, "Independent component analysis segmentation algorithm", *in Proc. of the 8th International Conference on Document Analysis and Recognition*, 680-684, 2005. https://doi.org/10.1109/ICDAR.2005.140.

[20] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4), 509-522, 2002. https://doi.org/10.1109/34.993558.

[21] H. Ikeda, Y. Ogawa, M. Koga, H. Nishimura, H. Sako, H. Fujisawa, "A recognition method for touching Japanese handwritten characters", *Proc. of International Conference on Document Analysis and Recognition* 641-644, 1999. https://doi.org/10.1109/ICDAR.1999.791869.

[22] P. Piquin, C. Viard-Gaudin, D. Barba, "Coopération des outils de segmentation et de binarisation de documents", *in Proc. of Colloque National sur l'Écrit et le Document*, Rouen, 283-292, 1994.

[23] D. You, G. Kim, "An approach for locating segmentation points of handwritten digit strings using a neural network", *Seventh International Conference on Document Analysis and Recognition*, 142-146, 2003. DOI:10.1109/ICDAR.2003.1227648.

[24] E. Vellasques, L. S. Oliveira, R. Sabourin, A. S. Britto, A. L. Koerich, "Modeling segmentation cuts using support vector machines", *Tenth International Workshop on Frontiers in Handwriting Recognition*, 41-46, 2006.

[25] T. B. A. Gader, A. K. Echi, "Deep learning-based segmentation of connected components in Arabic handwritten documents", *in Proc. of the International Conference on Intelligent Systems and Pattern Recognition*, Springer, 2022. https://doi.org/10.1007/978-3-031-08277-1_8.

[26] K. Barakat, B. Berat, R. Cohen, A. Droby, I. Rabaev, J. El-Sana, "Learning-free text line segmentation for historical handwritten documents", *Applied Sciences*, MDPI, 10(22), 8276, 2020. https://doi.org/10.3390/app10228276.

[27] K. Barakat, B. Berat, A. Droby, R. Alasam, B. Madi, I. Rabaev, R. Shammes, J. El-Sana, "Unsupervised deep learning for text line segmentation", *25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2304-2311, 2021. https://doi.org/10.1109/ICPR48806.2021.9413308.

[28] Dutta, Arpita and Garai, Arpan and Biswas, Samit and Das, Amit Kumar, "Segmentation of text lines using multi-scale CNN from warped printed and handwritten document images", *International Journal on Document Analysis and Recognition (IJDAR)*, 24(4), 299-313, 2021. https://doi.org/10.1007/s10032-021-00370-8.

[29] Y. H. Tseng, H. J. Lee, "Recognition-based handwritten Chinese character segmentation using a probabilistic Viterbi algorithm", *Pattern Recognition Letters*, 20(8), 791-806, 1999. https://doi.org/10.1016/S0167-8655(99)00043-4.

[30] A. Alaei, P. Nagabhushan, U. Pal, "Piece-wise painting technique for line segmentation of unconstrained handwritten text: a specific study with Persian text documents", *Pattern Analysis and Applications*, 14(4), 381-394, 2011. https://doi.org/10.1007/s10044-011-0226-x.

[31] N. Aouadi, A. Kacem, A. Belaïd, "Segmentation of touching components in Arabic manuscripts", *Proc. of the IEEE 14th Int. Conf. on Frontiers in Handwriting Recognition*, 452–457, 2014. https://doi.org/10.1109/ICFHR.2014.82.

[32] N. Aouadi, A. Kacem, A. Belaïd, "A recognition-based approach for segmenting touching components in Arabic manuscripts", *Proc. of the 13th Int. Conf. on Document Analysis and Recognition*, Tunis (relocated to Nancy, France), 2015. https://doi.org/10.1109/ICDAR.2015.7333718.

[33] A. Kacem, N. Aouïti, A. Belaïd, "Structural features extraction for handwritten Arabic personal names recognition", *Proc. of IEEE Transactions on Frontiers in Handwriting Recognition*, 268-273, 2012. https://doi.org/10.1109/ICFHR.2012.276.

[34] H. Ghaleb, P. Nagabhushan, U. Pal, "Segmentation of overlapped handwritten Arabic sub-words", *IJCA Proceedings on National Conference on Digital Image and Signal Processing, DISP*, 24-29, 2015.

[35] AlKhateeb, Jawad H and Jiang, Jianmin and Ren, Jinchang and Ipson, Stan, *Interactive knowledge discovery for baseline estimation and word segmentation in handwritten Arabic text*, Recent advances in technologies,2009. DOI: 10.5772/7428.

[36] T. Y. Zhang, C. Y. Suen, "A fast parallel algorithm for thinning digital patterns", *in Research Contributions in Image Processing and Computer Vision, ACM*, New York, 27(3), 236-239, 1984. https://doi.org/10.1145/357994.358023.

[37] S. E. Schaefer, "Graph clustering", *Computer science review*, 1(1), 27-64, 2007. https://doi.org/10.1016/j.cosrev.2007.05.001.

[38] B. J. Frey, D. Dueck, "Clustering by passing messages between data points", *American Association for the Advancement of Science*, 315(5814), 972-976, 2007. DOI: 10.1126/science.1136800.

[39] J. Canny, "A computational approach to edge detection", *IEEE Transactions on pattern analysis and machine intelligence* 8(6),679-698, 2009. DOI: 10.1109/TPAMI.1986.4767851.

[40] R. Jonker, A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problem", *Journal of Computing*, v38(4), 325-340, 1987. https://doi.org/10.1007/BF02278710.

[41] H. Chui, *Non-rigid point matching: algorithms, extensions and applications*, PhD dissertation, Yale University, 2001.

[42] F. L. Bookstein, "Principal warps: thin-plane spline and the decomposition of deformations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6), 567-585, 2002. https://doi.org/10.1109/34.24792.

[43] J. Sauvola, T. Seppanen, S. Haapakoski, M. Pietikainen, "Adaptive document binarization", *Proceedings of the fourth international conference on document analysis and recognition*, 147-152, 1997. https://doi.org/10.1109/ICDAR.1997.619831.

[44] Lee, Edward A, https://doi.org/10.1109/ICDAR.1997.61983139(5), 33-42, 2006.

[45] B. Gatos, N. Stamatopoulos, G. Louloudis, "ICDAR2009 handwriting segmentation contest", *International Journal on Document Analysis and Recognition (IJDAR)*, 14(1), 25-33, 2011. https://doi.org/10.1007/s10032-010-0122-8.

[46] Ullah, Inam and Azmi, Mohd Sanusi and Desa, Mohamad Ishak and Alomari, Yazan M, ”Segmentation of touching Arabic characters in handwritten documents by overlapping set theory and contour tracing”, *International Journal of Advanced Computer Science and Applications, Science and Information (SAI) Organization Limited*, 10(5), 2019. https://doi.org/10.14569/IJACSA.2019.0100519.

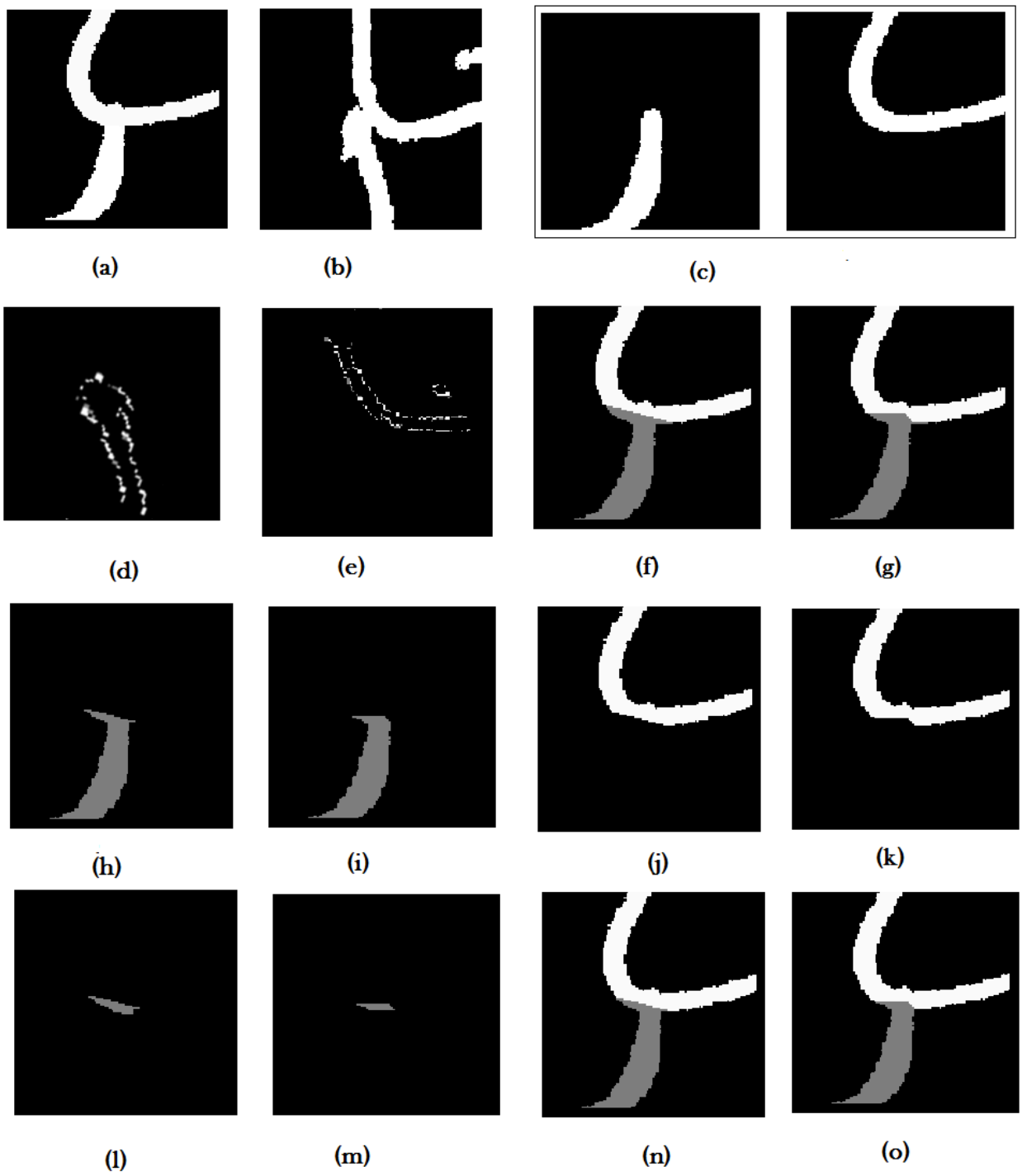


Figure 23: Illustration of the proposed separation method applied to Fig. 23(a): (a) Input TC. (b) The most similar template: similarity=0.013, WMP=147. (c) Ground truth (the TC's part A and B). (d) Template's part A contour after TPS transformation. (e) Template's part B contour after TPS transformation. (f) Result of separation without center adjustment using Euclidean distance and gravity center: $MS = 0.90312$. (g) Separation result without center adjustment using Manhattan distance and midpoint: $MS = 0.92652$. (h) Extraction of the TC's part A using Euclidean distance after gravity centers adjustment. (i) Extraction of the TC's part A using Manhattan distance after midpoints adjustment. (j) Extraction of the TC's part B using Euclidean distance after gravity centers adjustment. (k) Extraction of the TC's part B using Manhattan distance after midpoints adjustment. (l) The shared zone extraction after gravity center adjustment. (m) Extraction of the shared zone after midpoints adjustment. (n) Final separation using Euclidean distance after gravity centers adjustment: $MS = 0.91310$. (o) Final separation using Manhattan distance after midpoints adjustment: $MS = 0.93439$.