International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 8 (2016) pp. 173-183

Segmentation approach of Arabic manuscripts text lines based on multi agent systems

Youssef Boulid[1], Abdelghani Souhar[2], Mohamed Youssfi Elkettani[3]

[1] Department of Mathematics, Faculty of Science, University Ibn Tofail,
PB 242, 14000, Kenitra, Morocco
y.boulid@gmail.com

[2] Department of Computer Science, Faculty of Science, University Ibn Tofail,
PB 242, 14000, Kenitra, Morocco
houssouhar@gmail.com

[3] Department of Mathematics, Faculty of Science, University Ibn Tofail,
PB 242, 14000, Kenitra, Morocco
kettani.y@gmail.com

***Abstract*: Huge amount of handwritten documents can be treated by intelligent systems in order to retrieve information using text queries instead of manually searching in the scanned images. These systems often consist of five stages: pre-processing, segmentation, feature extraction, decision-making (character recognition) and post-processing. This paper deals with the segmentation stage, in which we present an approach inspired by the perception mechanisms involved in the human reading process to automatically extract text lines from Arabic handwritten documents. The proposed approach is based on multi-agent systems to detect and group connected components that belong to the same line. Experimental results on a data-set of Arabic handwritten documents show that this approach is a promising solution for extracting handwritten curved text lines.**

***Keywords*:** arabic handwritten documents, recognition**,** segmentation, text-lines, connected component, multi-agent systems.

I. Introduction

Nowadays, even with the intensive use of new technologies, humans continue to use the paper as a physical medium of communication and information storage. The collection and archiving of these documents is one of the great objectives of nations, as these archives are an inexhaustible mine of information and witnessing the evolution of nations and a lever of social, economic and cultural.

Huge amount of documents are stored in their original format (paper) or as scanned images and remain to be exploited. The conversion of these images into text version will allow users to automatically retrieve information from text queries instead of manually searching in the scanned images.

Systems for converting handwritten document image into text version are called Intelligent Character Recognition systems (ICR) and often consist of five stages: pre-processing, segmentation, feature extraction, character recognition and post-processing.

Segmentation is the process of partitioning the document into homogeneous entities such as lines, words or characters; and it is an essential step in the character recognition process since the recognition rate depends strongly on it. The segmentation of the document into text lines is to assign the same label to units that are spatially aligned (pixels or connected components) [1].

Extracting lines from images of printed documents is easier than from handwritten documents since in the first, the lines are horizontally aligned and the interline spaces are unified, but for manuscripts, the variability of writing styles and the image degradations make the process more difficult.

The Arabic script is naturally cursive, which makes the extraction of text lines from Arabic handwritten documents a real challenge [2].

Among the challenges of the Arabic script:

- The writing styles differ widely from one writer to another.

II. The fluctuation of the base line due to the movements of the pen, results in different variations in the inclination within the same text line.

- The cluttered writing style introduces ligatures between parts of words which makes overlap the adjacent lines.
- The Arabic writing is characterized by ascender and descender which easily introduce connections between successive lines.
- The diacritical points, which lie below and above the words, further complicate the task.

In this paper, we extend our work in [3] and propose an approach for the detection and the extraction of handwritten text lines by exploiting knowledge about the connected components arrangement. The components that participate in the same line are those satisfying some constraints.

Most of the architectures of ICR are linear, i.e. the phases are executed in a sequential manner, and this may cause high amount of errors that accumulate from one stage to another. Therefore, to solve this problem, the proposed system is based on the concept of Multi Agent Systems (MAS).

An agent is an autonomous entity which observes through sensors and acts upon an environment using actuators and directs its activity towards achieving goals[1].

The proposed system it is expected to be integrated into a global architecture for handwriting recognition based on MAS that simulate the process of human reading. Each agent is responsible for a portion of the image and handles a specific task of the recognition process and cooperates with other agents.

This concept will give us more flexibility and simplicity in the modeling and in the implementation of a system that is able to imitate the human reading process [4].

The rest of the paper is organized as follows: Section II examines some related works. Section III describes our approach for segmentation of Arabic handwritten text lines. Section IV presents the experimental results. Finally, Section V concludes the paper.

III. Related works

Generally there are two global approaches for the segmentation of text lines: either, through searching for separating locations between lines, or by searching for physical units such as Connected Component (CC) which are aligned.

As mentioned in [5] the extraction techniques can be divided into three classes: top-down, bottom-up and hybrid methods.

In the top-down methods, the document image is partitioned into regions, often recursively based on the global characteristics of the image. These methods are influenced by the large curvatures of the lines and touching characters that belong to more than one line. For the bottom-up methods, the basic elements such as pixels or CC are grouped to form text lines, for that a lot of calculation and heuristic analysis are needed. Hybrid approaches combine the two classes of techniques to give best results. The works in [6,1], present a review of existing segmentation methods of handwritten text lines. There are also competitions like ICDAR [7] and ICFHR [8] that participate in advancing the state of the art in this field of research.

The most common method is called Projection Profile (PP), it produces a histogram that represents each line of the image by : the number of black pixels in this line [9,10], the number of black/white transitions [11] or the number of CC. Then, the maxima and minima locations in the histogram are determined, and the space between two consecutive minima represents the location of the text line. This method is well suited for printed documents having straight rows. However, for handwritten documents the presence of short and narrow lines as well as many components that overlap with each other, will not produce significant peaks.

Other methods exist, as the k-means algorithm [12], the Hough transform [13,14] and active contour technique [15].

To solve the problems associated with the PP, the image can be divided into vertical strips and the histogram is projected within each band [12,16, 17].

In [12], the document is split vertically into several strips, and text blocks are detected based on the histogram of the PP. The blocks are clustered using K-means in three classes; the large blocks are split horizontally into a number of average text blocks according to the analysis of the neighborhood and interline spacing. With a collection of 100 historical documents of the national Tunisian archives, the authors claim to have 96% accuracy of correct line segmentation.

In the category of CC analysis [18] the method in [19] reaches 93.35% of the overall text line identification accuracy. First the outliers components are removed using a threshold value, then characters belonging to tow lines are detected and divided horizontally at the halve distance. For line detection, a rectangular neighbourhood is centred on a current component and increases to include those that satisfy certain conditions. The filtered components at the beginning are reallocated to the corresponding lines with respect to distance from their bounding box.

In [20], the data set consist of images each containing two pages of handwritten Arabic, after removing small components and linking broken characters, Fourier curve fitting within the horizontal PP is used to locate the point of separation. The contour is used to extract the base line of the CC which allows locating the cut point between different adjacent lines. The components are assigned to the closest line that is approximated by a polynomial curve that fits the pixels in the baselines. Finally, lifted small size components are reassigned to their line according to the closest CCs nearest neighbour in four directions. Using the matching score, the authors announced to have F1-Score for six different handwritten books, respectively 93% and 94% for the MS threshold values of 95% and 90%.

In [21], a smearing method based on adaptive morphological dilation is proposed for Arabic handwritten text lines extraction. The horizontal PP is used to estimate the skew line. Dilation with adaptive structuring element that changes the size and the slop according to the zone is used for the smearing process. In the second stage the big blobs are detected with a recursive function that search for the cut point in order to separate the components and match them with their lines following attraction and repulsion criterion. Touching components are detected and separated using erosion recursively in order to detect the thinness part which corresponds to the cut point. Using the matching score, the authors announced to have F1-Score respectively, of 96% and 97% for MS threshold 95% and 90%, for six different handwritten books.

The work in [22], proposes a generalization of the Adaptive Local Connectivity map (ALCM). The gray-level image produced by the ALCM algorithm is binarized using an adaptive thresholding algorithm to detect the location of the text lines (masks). In the second step, CCs are grouped into location masks. Finally, the text lines are extracted by superimposing the components with text line pattern mask. For touching characters, their contour is divided into pieces of segments which are reallocated to the nearest text line according to the distance from their centre. On a set of 45 handwritten Arabic document images, and based on connected component evaluation, the authors claim to reach 99.5% of

[1] Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, chpt. 2

correct rate of text lines segmentation.

In the same way [22], the method in [23] produces masks of text lines by a progressive filtering and validation process. The method iterates by decreasing the steerable filter height and uses geometric criteria to validate the text line masks. The method stops when every potential line mask have been detected. Finally the connected components are assigned to their respective lines represented by masks regions.

The authors in [24] proposed two hybrid systems that use closing morphological filter to extract lines in Arabic handwritten documents. The first system constructs the Outer Isothetic Cover (OIC) of the corresponding document after applying the Mathematical Morphology (MM), and the second system applies the MM closing on the result of the Hough transform. For both approaches the rate of text line segmentation was medium and that's mainly because the MM are suitable for mono-oriented documents and often fail when there is skew within text lines.

In [5] an approach was proposed for Arabic handwritten multi-oriented text lines extraction. The image is divided into cells and the orientation within each cell is calculated using the Wigner–Ville distribution on the PP histogram. The cells having similar orientation are joined. The method exploits the projection peaks and the orientation within each area to follow the connected components forming text lines. The diacritical components are reassigned to the nearest text line. To separate connected lines, for each touching character and from a starting point, the follow continues beyond the intersection and the strokes of the same component are recovered by analyzing the angular variation corresponding to the curvature of the descending character. On a 100 documents, the rate for text lines segmentation reaches 98.6%.

The algorithm in [25] begins by removing diacritics and then the sparse similarity graph is built based on the local orientation of the component. Text lines are represented by a disjoint set using BreadthFirst Search (BFS). The affinity propagation clustering method is used to assign the blobs to text lines. On a set of 125 Arabic documents, 95.6% accuracy was reported.

The same authors in [25] proposed an approach combining global and local techniques [26]. After removing small components, the direction of text line is detected at each component by locating the region having maximum neighboring components, and then the local orientation is estimated by the least square line passing through the centroid of the components. After that, a graph is constructed where the nodes correspond to components and weights on edges correspond to distance to the estimated orientation line. The shortest path-algorithm is used to complete the graph. Two estimations of text lines are obtained: the Breadth-first-search and the affinity-propagation clustering method. The splitting error are corrected using affinity propagation while merging error are corrected using Expectation Maximization. Touching components are localized by finding the common tangent of the convex-hull of successive components, and then the component is split near the centroid. Finally the diacritics and accent components are reassigned to their closest component. On a collection of 125 Arabic document images, the authors reach 98.76% of F1-score on a MS threshold of 90%.

Among all participants in ICDAR 2013 [7] the best method reaches about 94%, which firstly uses the Gaussian filter and divide the image into blocks which are then binarized using an adaptive threshold with respect to the skew angle estimated in each block. Then, the blocks are concatenated to get the path of text lines, and finally text lines are extracted by thinning the background of the path image.

Concerning the cluttered writing, the presence of touching/overlapping components make text line extraction more challenging since the adjacent lines become connected. These ambiguous components can be detected before, during or after text line extraction. As mentioned in [1] the separation of these components along the vertical direction is hard.

In general there are methods that employ recognition to correctly segment the touching components, where templates stored in a dictionary with their known correct configurations are used to figure out how to segment them [27,28], and other methods based on structural analysis of contour [22, 29], skeleton [30,31], foreground/background [12,17,32,33], convex Hull [26], that locate the connection points and use some rules such as angle and curvature features or distance from centre of gravity [25] to select the optimal configuration of segments.

Among the methods based on contour analysis, the work in [29], suggest a method based on local segmentation of contour of touching characters. It groups a linear representation of the edglelets into boundary fragments and then selects a connection among the boundary fragments that give the minimum global cost to produce the final segmentation of the shape. The authors perform an iterative search for the parameters that yield exactly two resulting components.

In [22] the algorithm draws a reference line between the line patterns and segments the contour of the component into contour segments. These segments are grouped with the corresponding text lines, based on the location of the centre of mass of each contour segment relatively to the reference line.

In the methods based on projection profile, the algorithm in [12] groups the different blocks into three classes. The big blocks include touching and overlapping components and are segmented into several parts based on the estimated heights of average blocks in the neighborhood and inter-line spacing.

In [32] a structural method for the separation of touching and overlapping words was proposed. After detecting touching/overlapping components using average line height. The area around interconnection pixels is detected and once the boundary region between the two lines is established the component is separated in two parts.

The method in [33] analyzes the histogram of the vertical projection profile to detect the presence of ambiguous components, which participate to more than one alignment. The separation of the component is roughly performed by drawing a horizontal frontier segment in middle way from the peaks.

In [17] the contact point in touching component is detected from the area between the baseline of the upper text line and the upper line of the lower text line. Based on the classification of the contact point as cusp contact or continuous contact, the segmentation if performed according to two situations: intersection between loops and intersection between one loop

and one stroke.

Compared to methods based on skeletons, the work in [30] proposes a method based on morphology analysis of Arabic terminal characters. Based on the skeleton of the component, the connection zone is extracted from a rectangle around the intersection point. The starting point is the highest point in the zone near the baseline of the first line, the direction of the curvature of decending letter is determined according to the starting point and the intersecting point. The idea is to follow the skeleton pixels from the starting point and to compute the angular variance crossing the intersection point in order to continue in the right direction that will have a minimum angular variance.

In a post-processing step, the algorithm in [31] begin by estimating the intersecting zone between the lines and the touching component, then compute the skeleton of this component and removes the junction points within these zones and label all the pixel with an id of the zone, finally assign each pixel with the id of the closest skeleton pixel.

Concerning template based methods, the main idea is to separate the connection by approximation to models stored in a dictionary with their known correct segmentations [27].

A method proposed in [27] which is based on recognition for segmentation of touching components by finding similar model stored in a dictionary with its correct segmentation.

In the same way, the work in [28] proposes a training and testing framework based on template matching to segment touching handwritten text. Local shape around touching components are identified and mapped with the best template patch in a constructed dictionary using shape context descriptor, in order to be segmented using the correct segments of the template.

IV. Proposed approach

Our approach is inspired by the results observed in the human reading process and exactly in the detection of text lines. By analysing the shape, size and location of the basic elements, we can collect them into groups representing the lines [34].

The writing is characterized by the notion of order and neighborhood, this means that the words and characters in a local region, are more or less aligned on the same horizontal line, and this is true even for curved writings.

If we take a Bounding Box of a CC which has four sides; top, bottom, left and right. The components above the top side can be considered as participating in the "top **field of vision**" of that component.

The writing order in Arabic script is from right to left and the text is typically represented as portions of words called Pieces of Arabic Words (PAWs) [35]. So for a given PAW, the next one or the successor is definitely located in its left field of vision (Fig. 1).

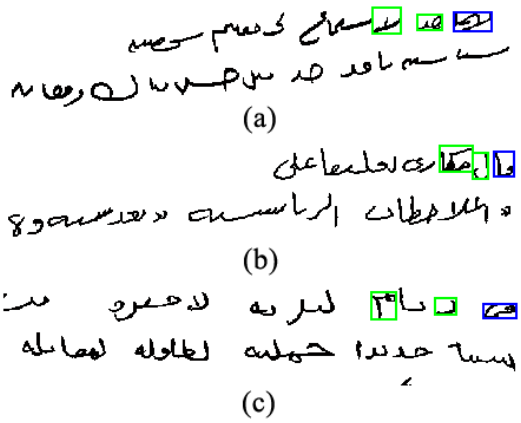


Figure 1. Example of handwritten text lines. The components in green are located in the left field of vision of the components in blue.

The methods which are based on CC analysis use geometric information such as shape, orientation, position, and size of CC for grouping them in rows. As mentioned in [20] These methods are more suitable for complex documents than methods based on PP, even if they could be sensitive to changes in the component structures.

The main idea is to reconstruct the lines of the document by detecting the various components (character, part of the word or full word) of each line, in a successive manner.

To simulate the process of human reading, we use a Multi-Agent System. The agent is designed in a way to move to a CC, to see instantly the other components in its field of vision, evaluates certain characteristics, determines and executes its behavior according to the results of its sensory feedback. For example, if the detected component satisfies the participation conditions, then the agent will move to it and it will do the same for all components participating in the same line.

To have a full visibility of the document, the first Agent called **Document Agent** (DA) is responsible for the extraction of the lines and the calculation of some global settings. The second Agent is called **Line Agent** (LA), it has a partial visibility of the document, and its role is to detect the PAWs (components) of the line. The third Agent is called **Touching Component Agent** (TCA); it is responsible for the detection and the segmentation of touching and overlapping components.

All agents are reactive and goal oriented.

Facing a new document, the DA first removes the diacritics components and calculates some global settings on the document, and orders the LA to begin the search for the first line.

To extract the first line, the LA starts by detecting the first component (considered current) in the line and moves to it, and then detects the next component that becomes current; then again detect the next one and so on, until the end of the line is reached. The DA extracts and suppresses those components from the image and the second line will be considered as the first line, and this process is executed iteratively until there are no more rows to extract.

During the search of the lines, if a touching component is found it is sent to the TCA to be segmented. This latter returns the result to the LA and inform the DA to update the image.

After the detection of the line, the DA recovers the resulted components and extracts their diacritics. (Fig. 2).

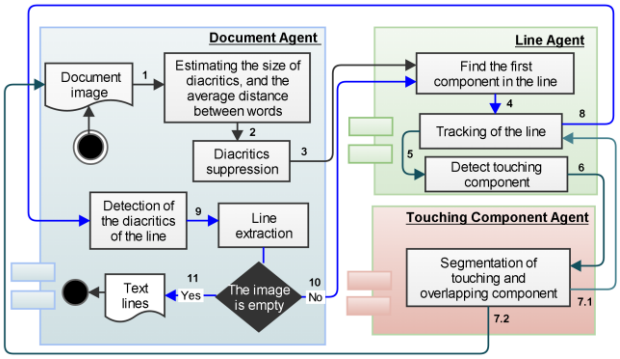


Figure 2. Block diagram illustrating the interaction between the agents.

A. *Sensory parameters*

These settings control the vision of the LA:

- The nature of the field of vision (parallel or perspective): In parallel we specify the height. In perspective we specify the angle.
- The distance vision: is the scope of the vision of the agent.
- The granularity of vision: the amount of visibility and details.

These parameters are estimated and fixed according to a supervised learning on a set of documents.

B. *The characteristics of the environment*

For the DA, the environment is fully observable because it must have a total visibility of the image in order to extract the lines. The environment is partially observable fro the LA since it is able to see only the components in its neighborhood (according to its field of vision) and finally for the TCA, the environment is partially observable too.

- The environment is deterministic since the next state is completely determined by the current state of the environment and the action performed by the agent.
- The choice of the agent to move towards a component has an impact on the next state, therefore the environment is sequential.
- The environment is static because it does not change when the agents do not act.
- All sensory data and actions of the agents are limited and clearly distinct, so the environment is discrete.

C. *Field of vision function*

To detect the components that are in the left field of vision of the component on which the agent is located, it first loops through the image's pixels in rows and columns from the left facade of the that component, and for each line, the index of the first black pixel is added to a list. The goal is to detect all the components that are visible to the agent from its left side.

Thus, we denote $FV_l(C)$ the function of left field of vision for the component C, such as for the example in Fig. 3, we have:

$FV_l(\text{د}) = \{\text{د}, \text{ف}, /\}$

Figure 3. Illustrative example of parallel left field of vision function of a component.

According to Fig. 4, we denote:

- Cc , the current component
- $C_{s/c}$, the successor of the current component
- $C_{p/c}$, The predecessor of the current component

Figure 4. The successor (green) and the predecessor (yellow) for the current component (blue).

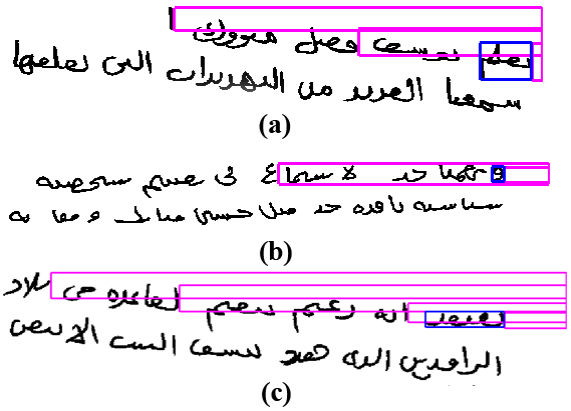
D. *Treatments of Line Agent*

1) *Finding the first component in the text line*

The DA orders the LA to search for the first component in text line, which it proceeds as follows:

The agent loops through the image's pixels in rows and columns from top to bottom, if it finds a component; it reframes a horizontal rectangle having width equals to the distance between that component and the right edge of the image. Then, it searches all the components in this rectangle and moves to the farthest one (the one with the maximum x-coordinate). It again reframes another rectangle from this component, and so on until it gets an empty rectangle.

Among all visited component, the first component in the text line corresponds to the last one where the agent is positioned (Fig. 5).



(a)

(b)

(c)

Figure 5. Illustration of the process of detection of the first component in the line (blue).

Further analyses are made to make sure whether the component is part of a line or not, in order to avoid the case when a part of the character could be considered.

2) *Finding the next component*

Once the agent is positioned on a CC, it searches the successor. It goes thought a set of processes to select the optimal component in its field of vision.

All the components that may be successors are detected by the left field of vision function $FV_l(Cc)$.

Thus, for two components A and B, we denote:

- $S(A)$, The surface of A, which represents the number of black pixels in A.
- $D(A,B)$, The Euclidean distance between A and B.
- $P_{A/B}$, The proportion of component A relatively to component B (Fig. 6).
- $P_{\overline{A}/B}$, The percentage of $\overline{A}$, according to component A. With $\overline{A}$ is the portion of A, which is included in B (Fig. 6).
- $FV_t(A)$, The list of components in the top field of vision of A.
- $FV_b(A)$, The list of components in the bottom field of vision of A.

Figure 6. The value of $P_{A/B}$ is 61% and $P_{\overline{A}/B}$ is 100%. 'A' represents the green rectangle and 'B' represents the blue rectangle, the gray part represents the portion of 'A' included in 'B' denoted $\overline{A}$

Among all the components in the left field of vision of the current component, the agent selects those who significantly participate in its vision, noted $FV_{l'}(Cc)$ as follows:

If we set, $L = FV_l(Cc)$, we have:

$$FV_{l'}(Cc) = \left\{ l_i \in L / P_{\overline{l_i}/Cc} \geq Th, D(l_i, Cc) \leq D_w \right\} \quad (1)$$

Where Th represents the percentage of participation (we find that 40% gives the best results) and D_w represents the average distance between words, which is estimated by:

$$D_w = \frac{\sum_{n=1}^{m} W_n}{m} * 2 \quad (2)$$

Where, W_n represents the width of the Nth component and m represents the total number of components in the image after removing the diacritics.

Let $L = F_{sort}(FV_{l'}(Cc))$ be the list of components that significantly participate in the field of vision of Cc and sorted in ascending order according to their distance from Cc.

The agent can find the current component's successor (C_s / Cc) following the steps below (see Fig. 7):

1. The first element L_1 of the list is a successor of Cc if and only if it does not overlap vertically with another component; this means that there are no components in $FV_t(L_1)$ and $FV_b(L_1)$ which also participate in the vision of Cc.
2. If there is a vertical overlapping, the component having an area greater than that of L_1 is selected as successor, such as:

$$F_{overlap}(L_1 / Cc) = Ct \in \{FV_t(L_1) \cup FV_b(L_1)\}, Ct \in FV_{l'}(Cc) / S(Ct) \geq S(L_1) \quad (3)$$

3. If there is no component that participate significantly in CC ($FV_{l'}(Cc) = \phi$), we set $L = F_{sort}(FV_l(Cc))$ and $C_t = F_{overlap}(L_1 / Cc)$, if $C_t = \phi$ so $Ct = L_1$. Then, the agent calculates the percentage of participation, in order to give more weight to the criterion $P_{\overline{Ct}/Cc}$ as following:

$$V = ((P_{\overline{Ct}/Cc} * 2) + P_{Ct/Cc}) / 3 \quad (4)$$

4. If $V \geq 40\%$ then the successor is Ct.
5. Else if $V \leq 40\%$ we set $C_{p/Ct} = FV_r(Ct)$ indicating the predecessor of Ct. If $C_{p/Ct} \neq Cc$, then Ct will be ignored in the next iteration.
6. If all of the above cases are not satisfied, the agent ignores this time the current component Cc and begins the treatment by considering the last successor found, as current. This algorithm is repeated until reaching the last component in the line for which $FV_l(Cc) = \phi$.

At the end of process, the agent's memory is filled with a list of all visited components, which corresponds to words and characters in the current line.

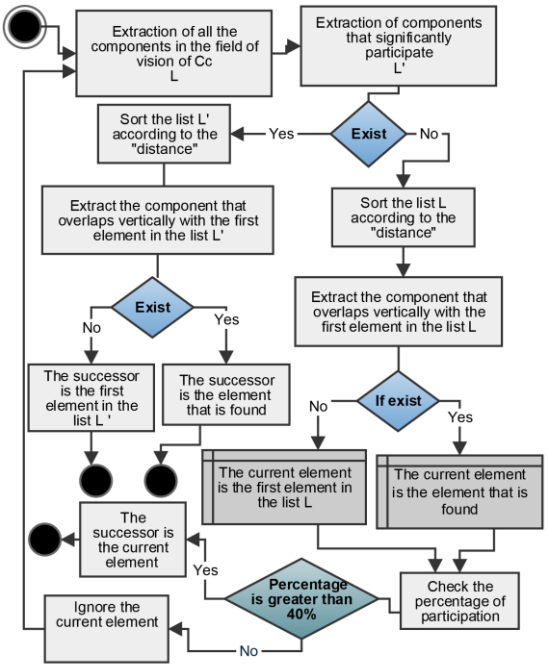


Figure 7. Flowchart of Line Agent treatments for searching the successor.

E. Treatments of Touching Component Agent

During the search of the line, The LA checks each component and verifies whether it touches more than one line, by comparing its predecessor to the components that were already chosen as belonging to the current line.

The predecessors are detected using the right field of vision. Touching or overlapping components often have previous components that belong to more than one line. An example of this is shown in the fig.8.

To make sure that the component participates in more than one line, it is sent to the TCA, which proceeds as follows:

- First, the TCA demands the DA to extract the portion of the image located in the right side of the component. The width and the height of this portion are respectively equal to the height and two times the width of that component.
- Second, the DA sends the histogram of the horizontal projection profile to the TCA. This latter uses a set of rules to analyze the histogram and verifies if the space between peaks (maxima values) represent the interline space. Fig.9 shows the location of these peaks corresponding to the base lines (fictitious line which follows and joins the lower part of the character bodies in a text line [1]).

For the case of touching components located in the first position in the line, the TCA repeats the same treatment, but at this time the operation takes place on the histogram of the portion of the left side of these components.

Figure 8. Detection of touching component (green). The presence of predecessors (magenta) not participate in to the current line.

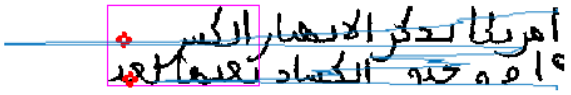


Figure 9. Location of peaks (red) from the analysis of the histogram of horizontal projection profile (blue curve).

Once a component satisfies the above conditions, it is considered as a touching or an overlapping component and then need to be segmented, at this stage, the LA remains inactive and waits the result from the TCA.

According to the nature of the Arabic script which is characterized by the presence of the ascenders and descenders, touching and overlapping of characters often happen in the interline space (Fig.10). Therefore, the region of interest is located between the peaks of the horizontal PP histogram.

In most cases, the touching and overlapping of characters introduce deviation within the character's shape. In order to localize this deviations, the idea is to analyze the continuity of the strokes in the region of interest and to disconnect the parts of stroke that are non linear.

Figure 10. Portions of a document representing some overlapping (red rectangle) and touching (blue rectangle) components.

In order to segment the component and to accurately separate the different strokes belonging to different characters, the TCA executes the following steps:

It begins by extracting the skeleton of the component using a Mathematical Morphology algorithm (Figure 11.b).

After that, it detects the points of intersection on the skeleton which correspond to pixels connected to three or more other pixels and dilate these points with a structuring element disk to group them and represent each group by their centroid.

Using the information of the interline (the space between the first two peaks on the histogram), the agent detects the intersecting points representing deviation (Figure 11.c) and extract a rectangular portion on the skeleton around each intersecting point (Figure 11.d).

From these portions, the end points (pixels with one connection) are located and the agent chooses the first one and connects it with the others, by drawing a segment for each end-point linking it with the first point which passes through the intersecting point.

The Figure 11.f, represent an example of the extracted portion, the pixel in blue is the intersecting point, the pixel in red is the first end-point and the pixels in yellow and green are the second and third end points respectively.

Each line is analyzed separately using the "eccentricity" which is the ratio of the distance between the foci of the ellipse (that has the same second-moments as the region) and its major axis length. The value is between 0 and 1. (An ellipse whose eccentricity is 1 is a line segment).

After removing the rectangular portion from the skeleton, we get a set of connected components (Fig.11.e), and for each intersecting point, the skeleton's parts are reconstructed by linking together the end-points that give the maximum of eccentricity with their corresponding resulted components.

The Figures 11.g and 11.h, represent respectively the constructed segments from the first end-point to the second end-point and to the third end-point. The values of eccentricity are respectively 0.99 and 0.86.

It is to be noted that, in same cases, same end-points remain unconnected (not linked to any other end points), this case corresponds to a touching component and therefore these end-points are linked with the intersecting point.

At the end of the treatment each resulted components are dilated and intersected with the original image to get the segmented characters (Fig. 11.i, 11.j and 11.k).

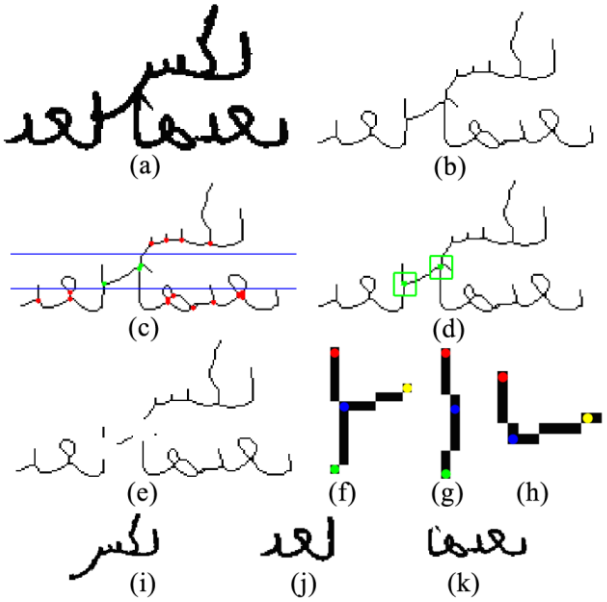


Figure 11. Process of segmentation of touching and overlapping words.

Other example of the segmentation of touching and overlapping characters are shown in Fig.12.

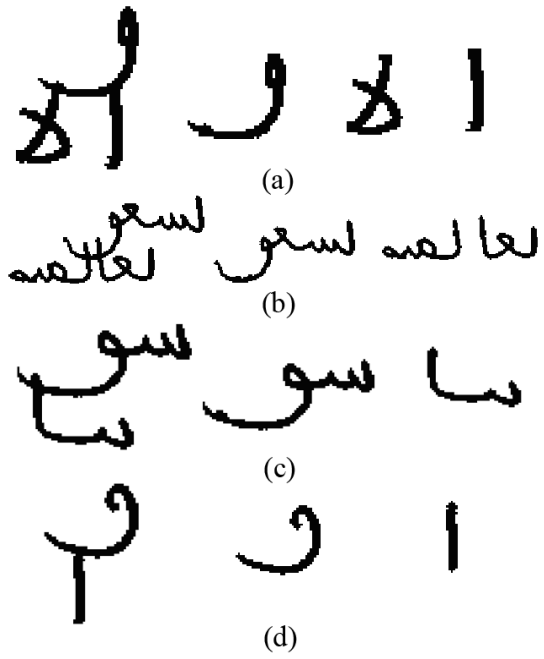


Figure 12. Same example of segmentation of touching and overlapping characters.

Based on the centroid of the resulted separated components, the TCA sends the component belonging to the current line to the LA to continue its treatments. The remaining components are sent to the DA in order to update the document image.

F. Treatment of Document Agent

1) Extraction of the line

The DA receives a list of detected components from the LA. Then it draws a curve that goes through all the pixels of the lower contours of these components and shifts it down vertically in order to include diacritical points that are below the extracted words. The DA then draws a polygon linking the two ends of the curve with the two top corners of the image as shown in Fig. 13.

Then the portion of the image represented by the polygon is extracted from the image and saved in the memory of the agent. This treatment is executed in an iterative manner for each line, until there are no more lines to be extracted.

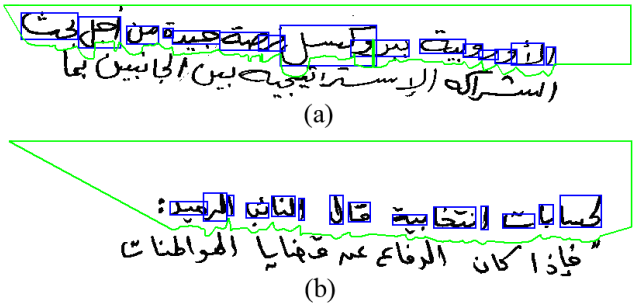


Figure 13. Example of components detection, diacritics assignment and line extraction.

V. Experimentation and evaluation

To assess the accuracy of the proposed approach and to compare it to other methods, we have used a public data set that has 125 images of Arabic handwritten document with 1974 lines, which is available for download at [36].

The method used to evaluate the performance of the system is based on counting the number of matches between the extracted lines and the lines in the ground truth. We used the criterion Match Score (MS) whose values are calculated according to the intersection of the sets of pixels on the results with those of the ground truth [37] which is calculated as follows:

$$MS(r_i, g_j) = \frac{T(P(r_i) \cap P(g_j))}{T(P(r_i) \cup P(g_j))} \quad (5)$$

With $MS(r_i, g_j)$ is a real number between 0 and 1 which represents the matching score between the zone r_i resulted from the algorithm and the zone g_j in the ground truth. P corresponds to pixels that represent the foreground (text) and T is an operator that counts the number of pixels in each zone. Thus, we obtain the scores between all the resulted zones and the zones in the ground truth. If the score for a zone is found above a threshold, it is counted as true positive (TP). If the resulted zones do not match any zone in the ground truth it is counted as false positive (FP), the zones in the data set that are left unmatched are considered false negative (FN). Therefore, we count precision, recall and F1-score as follows:

$$precision = \frac{TP}{TP + FP} \quad (6)$$

$$recall = \frac{TP}{TP + FN} \quad (7)$$

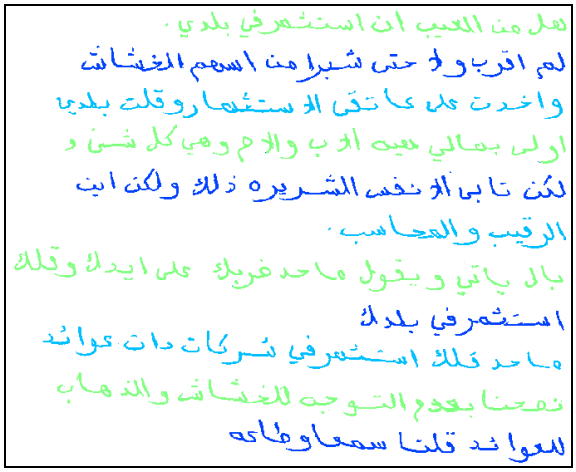
$$F1_{Score} = \frac{2 * precision * recall}{precision + recall} \quad (8)$$

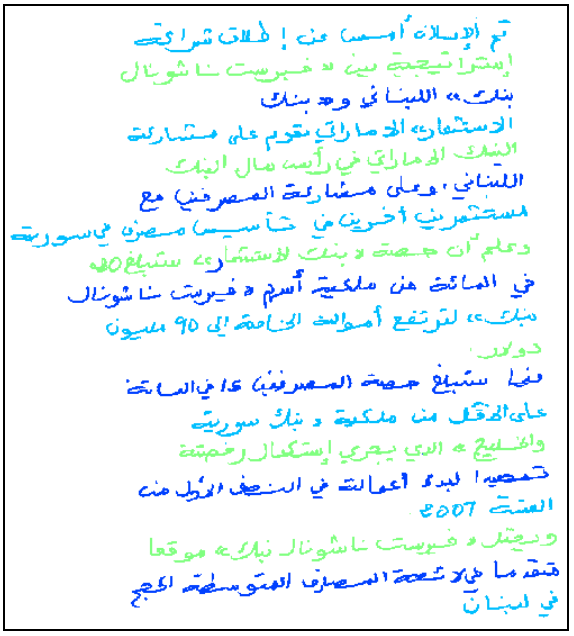
According to the tests we obtained a F1-Score respectively of 97.4% and 94.3% for the MS-threshold values of 90% and 95%. Table 1 compares the score of the proposed method with other methods tested on the same data set.

Methods	F1-Mesure	
	MS=90	MS=95
The method in [25]	95.6	90.9
The method in [26]	98.8	Not reported
The proposed method	97.4	94.3

Table 1. Results obtained on the original data in [36].

Fig. 14 illustrates some visual results of our approach for segmentation of Arabic handwritten documents text lines.
Therefore, we could say that the proposed approach is comparable to the state of the art. Although, it has been adapted here for Arabic handwritten documents, it could be generalized to any other script with linear writing.





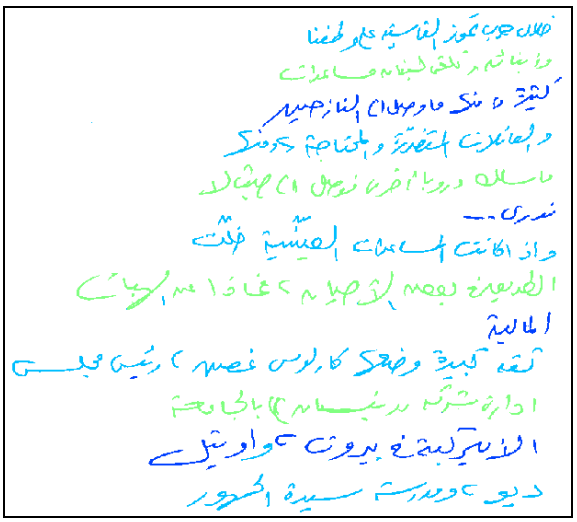


Figure 14. Some results of the proposed approach.

VI. Conclusion

In this paper we have presented a system based on connected components analysis for extracting text lines from the binary handwritten Arabic document images.
The obtained results confirm the effectiveness of the proposed system, although, it is not easy to claim that a handwritten text line extraction system alone will cover all cases. However, the exact segmentation of touching lines in historical manuscripts needs recognition because segmentation and recognition are dependent tasks.
In addition to this, most of character recognition systems execute the phases in a sequential manner, which means that if a step is not properly performed, it may cause errors that accumulate from one stage to another, and therefore, influences the final recognition result.
To solve this problem, we are currently working on an architecture of handwritten character recognition based on collaborative multi-agent systems that incorporate mechanisms and strategies involved in the human reading process. This will keep the traceability of the treatments in each phase and allow returning to a backward step in order to extract further information and to correct errors.
Therefore, we have organized the proposed system as three agents: document agent, line agent and touching component agent that collaborate to segment the document image into text lines.

References

[1] Likforman-Sulem, L., Zahour, A., & Taconet, B. (2007). Text line segmentation of historical documents: a survey. International Journal of Document Analysis and Recognition (IJDAR), 9(2-4), 123-138

[2] Li, Y., Zheng, Y., Doermann, D., & Jaeger, S. (2008). Script-independent text line segmentation in freestyle handwritten documents. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 30(8), 1313-1329

[3] Boulid, Y., Souhar, A., Elkettani, M. Y. (2015, December). Arabic handwritten text line extraction using connected component analysis : from a multi agent perspective.The 15th International Conference on

Intelligent Systems Design and Applications (ISDA), December 14-16 2015, Marrakech, Morroco. IEEE

[4] Boulid, Y., & Elkettani, M. Y. (2014). Approach for Arabic Handwritten Image Processing: Case of Text Detection in Degraded Documents. International Journal of Computer Applications, 101(14)

[5] Ouwayed, N., & Belaïd, A. (2012). A general approach for multi-oriented text line extraction of handwritten documents. International Journal on Document Analysis and Recognition (IJDAR), 15(4), 297-314

[6] Razak, Z., Zulkiflee, K., Idris, M. Y. I., Tamil, E. M., Noor, M. N. M., Salleh, R., ... & Yaacob, M. (2008). Off-line handwriting text line segmentation: A review. International journal of computer science and network security, 8(7), 12-20

[7] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., & Alaei, A. (2013, August). Icdar 2013 handwriting segmentation contest. In Document Analysis and Recognition (ICDAR), 2013 12th International Conference on (pp. 1402-1406). IEEE

[8] Gatos, B., Stamatopoulos, N., & Louloudis, G. (2010, November). Icfhr 2010 handwriting segmentation contest. In Frontiers in handwriting recognition (icfhr), 2010 international conference on (pp. 737-742). IEEE.

[9] Bennasri, A., Zahour, A., & Taconet, B. (1999). Extraction des lignes d'un texte manuscrit arabe. In Vision interface (Vol. 99, pp. 42-48)

[10] Nicolaou, A., & Gatos, B. (2009, July). Handwritten text line segmentation by shredding text into its lines. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on (pp. 626-630). IEEE

[11] Marti, U. V., & Bunke, H. (2001). On the influence of vocabulary size and language models in unconstrained handwritten text recognition. In Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on (pp. 260-265). IEEE

[12] Zahour, A., Likforman-Sulem, L., Boussalaa, W., & Taconet, B. (2007, September). Text line segmentation of historical arabic documents. InDocument Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on (Vol. 1, pp. 138-142). IEEE

[13] Malleron, V., Eglin, V., Emptoz, H., Dord-Crouslé, S., & Regnier, P. (2009, July). Text lines and snippets extraction for 19th century handwriting documents layout analysis. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on (pp. 1001-1005). IEEE

[14] Louloudis, G., Gatos, B., & Halatsis, C. (2007, September). Text line detection in unconstrained handwritten documents using a block-based Hough transform approach. In Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on (Vol. 2, pp. 599-603). IEEE

[15] Bukhari, S. S., Shafait, F., & Breuel, T. M. (2009, July). Script-independent handwritten textlines segmentation using active contours. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on(pp. 446-450). IEEE

[16] Zahour, A., Taconet, B., Mercy, P., & Ramdane, S. (2001). Arabic hand-written text-line extraction. In Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on (pp. 281-285). IEEE

[17] Bruzzone, E., & Coffetti, M. C. (1999, September). An algorithm for extracting cursive text lines. In Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on (pp. 749-752). IEEE

[18] Likforman-Sulem, L., & Faure, C. (1994). Extracting text lines in handwritten documents by perceptual grouping. Advances in handwriting and drawing: a multidisciplinary approach, 117-135

[19] Khandelwal, A., Choudhury, P., Sarkar, R., Basu, S., Nasipuri, M., & Das, N. (2009). Text line segmentation for unconstrained handwritten document images using neighborhood connected component analysis. In Pattern Recognition and Machine Intelligence (pp. 369-374). Springer Berlin Heidelberg

[20] Adiguzel, H., Sahin, E., & Duygulu, P. (2012, September). A Hybrid for Line Segmentation in Handwritten Documents. In Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on (pp. 503-508). IEEE

[21] Khayyat, M., Lam, L., Suen, C. Y., Yin, F., & Liu, C. L. (2012, March). Arabic handwritten text line extraction by applying an adaptive mask to morphological dilation. In Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on (pp. 100-104). IEEE

[22] Shi, Z., Setlur, S., & Govindaraju, V. (2009, July). A steerable directional local profile technique for extraction of handwritten arabic text lines. In Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on(pp. 176-180). IEEE

[23] Swaileh, W., Mohand, K. A., & Paquet, T. (2015, August). Multi-script iterative steerable directional filtering for handwritten text line extraction. In Document Analysis and Recognition (ICDAR), 2015 13th International Conference on (pp. 1241-1245). IEEE

[24] Maddouri, S. S., Ghazouani, F., & Samoud, F. B. (2014, March). Text lines and PAWs segmentation of handwritten Arabic document by two hybrid methods. In Advanced Technologies for Signal and Image Processing (ATSIP), 2014 1st International Conference on (pp. 310-315). IEEE

[25] Kumar, J., Abd-Almageed, W., Kang, L., & Doermann, D. (2010, June). Handwritten Arabic text line segmentation using affinity propagation. InProceedings of the 9th IAPR International Workshop on Document Analysis Systems (pp. 135-142). ACM

[26] Kumar, J., Kang, L., Doermann, D., & Abd-Almageed, W. (2011, September). Segmentation of handwritten textlines in presence of touching components. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 109-113). IEEE

[27] Aouadi, N., Kacem, A., & Belaid, A. (2014, September). Segmentation of Touching Component in Arabic Manuscripts. In Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on (pp. 452-457). IEEE

[28] Kang, L., & Doermann, D. (2011, September). Template based segmentation of touching components in handwritten text lines. In Document Analysis and

Recognition (ICDAR), 2011 International Conference on (pp. 569-573). IEEE
[29] Kang, L., Doermann, D., Cao, H., Prasad, R., & Natarajan, P. (2012, March). Local segmentation of touching characters using contour based shape decomposition. In Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on (pp. 460-464). IEEE
[30] Ouwayed, N., & Belaïd, A. (2009, January). Separation of overlapping and touching lines within handwritten arabic documents. In Computer Analysis of Images and Patterns (pp. 237-244). Springer Berlin Heidelberg
[31] Louloudis, G., Gatos, B., Pratikakis, I., & Halatsis, C. (2008, August). Line and word segmentation of handwritten documents. In Proceedings of the International Conference in Frontiers in Handwritten Recognition (ICFHR-08) (pp. 19-21)
[32] Takru, K., & Leedham, G. (2002, August). Separation of Toching and Overlapping Words in Adjacent Lines of Handwritten Text. In null (p. 496). IEEE
[33] Likforman-Sulem, L., & Faure, C. (1995). Une methode de resolution des conflits d'alignements pour la segmentation des documents manuscrits.Traitement du signal, 12(6), 541-549
[34] Saabni, R., & El-Sana, J. (2011, September). Language-independent text lines extraction using seam carving. In Document Analysis and Recognition (ICDAR), 2011 International Conference on (pp. 563-568). IEEE
[35] Bouafif, F., Maddouri, S., & Ellouze, N. (2012). A hybrid method for three segmentation level of handwritten Arabic script. The international Arab Journal of Information Technology (IAJIT), 9(2), 117-123
[36] Handwritten Arabic Proximity Datasets. Language and Media Processing Laboratory. http://lampsrv02.umiacs.umd.edu/projdb/project.php?id=65
[37] Phillips, I. T., & Chhabra, A. K. (1999). Empirical performance evaluation of graphics recognition systems. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21(9), 849-870

Author Biographies

Youssef Boulid received his M.S. degree in Decision Support Systems and Project Management in 2012 from univesity Ibn Tofail, Faculty of science, Kenitra- Morocco. Currently he is preparing a PhD in the Computer Science Department at the same faculty. His research interests include image processing, handwritten document analysis, Arabic handwritten recognition and Artificial intelligence.

Abdelghani Souhar received M.S. degree in applied Mathematics in 1992, PhD degree in computer science in 1997 from University Mohammed 5 - faculty of science in Rabat - Morocco. Now he is a Professor at university Ibn Tofail - faculty of science in Kenitra - Morocco. His research interests include Computer Aided Engineering, Computer Aided Design and Artificial Intelligence.

Mohamed Elyoussfi Elkettani received M.S. degree in applied mathematics in 1980 and PhD degree in Statistics in 1984 from Orsay Faculty of Science, University of Paris XI. Now he is a Professor at university Ibn Tofail - faculty of science in Kenitra - Morocco. His research interests include Multivariate statistics and Image recognition algorithms.