Multimedia Tools and Applications (2024) 83:24339–24359
https://doi.org/10.1007/s11042-023-15382-1

Improving concave point detection to better segment overlapped objects in images

Miquel Miró-Nicolau[1,2] · Gabriel Moyà-Alcover[1,2] · Manuel González-Hidalgo[2,3,4] · Antoni Jaume-i-Capó[1,2]

Received: 14 January 2022 / Revised: 28 March 2023 / Accepted: 15 April 2023 /
Published online: 14 August 2023

Abstract

This study presents a method to improve state-of-the-art concave point detection methods as the first step towards effectively segmenting overlapping objects in images. The approach relies on analysing the curvature of the object contour. This method comprises three main steps. First, the original image is preprocessed to obtain the curvature value at each contour point. Second, the regions with higher curvatures are selected and a recursive algorithm is applied to refine previously selected regions. Finally, a concave point is obtained for each region by analysing the relative position of their neighbourhood. Furthermore, the experimental results indicate that improving the detection of concave points leads to better division of clusters. To evaluate the quality of the concave point detection algorithm, a synthetic dataset was constructed to simulate the presence of overlapping objects. This dataset includes the precise location of concave points, which serve as the ground truth for evaluation. As a case study, the performance of a well-known application, such as the splitting of overlapping cells in images of peripheral blood smears samples from patients with sickle cell anaemia, was evaluated. We used the proposed method to detect concave points in cell clusters and then separated these clusters by ellipse fitting.

✉ Gabriel Moyà-Alcover
gabriel.moya@uib.es

Miquel Miró-Nicolau
miquel.miro@uib.es

Manuel González-Hidalgo
manuel.gonzalez@uib.es

Antoni Jaume-i-Capó
antoni.jaume@uib.es

[1] UGiVIA Research Group, University of the Balearic Islands, Dpt. of Mathematics and Computer Science, 07122 Palma, Spain

[2] Laboratory for Artificial Intelligence Applications (LAIA@UIB), University of the Balearic Islands, 07122 Palma, Spain

[3] SCOPIA Research Group, University of the Balearic Islands, Dpt. of Mathematics and Computer Science, 07122 Palma, Spain

[4] Health Research Institute of the Balearic Islands (IdISBa), E-07010, Palma, Spain

Keywords Sickle cell disease · Image processing · Computer vision · Overlapped objects · Segmentation · Concave points

1 Introduction

Segmenting overlapped objects in images is a process that can be used as a first step in various biomedical and industrial applications, ranging from the study of blood cells [11] and grain analysis [28] to the segmentation of magnetic resonance breast images [6].

In these applications, the analysis of individual objects using their singular features is typically required. The existence of image areas with overlapping objects reduces the available information for some of the objects in the image, that is, occlusion zones. These occlusion zones introduce enormous complexity into the segmentation proces, making it a challenging problem that can be solved from multiple points of view and is still open.

Several strategies have been proposed for addressing this problem. Watershed algorithms [18, 22, 25] and level sets [4] are widely used in scenarios with well-defined boundaries between overlapping objects. However, these methods cannot separate overlapping objects with homogeneous intensity values because the detection of the initial markers is inaccurate, resulting in a diffuse boundary.

To circumvent these difficulties, an alternative segmentation method based on concave point detection can be employed. These points denote the positions where the contours of the different objects overlap and, simultaneously, are the locations where the overlapped object changes from one of its sub-objects to another. Once the concave points are detected, multiple techniques can be used to divide the objects. The advantage of concave points detection is that it is invariant to scale, color, rotation, and orientation.

Furthermore, methods based on the detection of concave points can perform good segmentation without using a large dataset or input image size constraints, unlike deep learning methods. Moreover, the detection of concave points for segmenting overlapping objects can be considered transparent because it presents simulatability, decomposability, and algorithmic transparency [1].

The better the concave-points detection, the better is the cluster division. Therefore, we propose a new method that increases the precision concave-point detection compared to state-of-the-art methods. This method is based on the analysis of contour curvature. To evaluate our proposed method, we constructed a synthetic dataset to simulate overlapping objects and provided the position of concave points as ground truth. We used this dataset to compare the detection capacity and spatial precision of the proposed method with those of the state-of-the-art methods. As a case study, we evaluated the proposed concave point detector with a well-known application: the splitting of overlapping cells in microscopic images of peripheral blood smear samples of red blood cells (RBC) of patients with sickle cell disease (SCD). Note that this algorithm is not limited to cell segmentation; it can be used for other applications in which the separation between overlapping objects is required.

The remainder of this paper is organized as follows: in the next section, we describe the related work. In Section 3 we explain the proposed method for efficient detection of concave points. Section 4, specifies the experimental environment and describes the datasets used. Section 5 discusses the results and comparison experiments obtained after applying the proposed method to synthetic and real images of object clusters. Finally, in Section 6 we give the conclusions of our work.

2 Related work

In state-of-the-art methods, we can find multiple approaches based on concave-point detection. Following the taxonomy proposed by Zafari et al. [31]. However, by adding a category for other methods, we classified these methods into four categories: skeleton, chord, polygon approximation, curvature, and others.

2.1 Skeleton

Skeleton-based approaches utilize information about the boundary and its medial axis to detect concave points. Song and Wang [24] identified concave points as local minima of the distance between the boundary of an object and its medial axis. The medial axis was obtained using an iterative algorithm based on a binary thinning algorithm, followed by a pruning method. Samma et al. [23] identified concave points by intersecting the boundary of an object obtained by applying the morphological gradient skeleton of a dilated background image.

These methods require a large change in curvature to detect existing concave points. Therefore, skeleton-based methods tend to fail on contour objects with smooth curvatures.

2.2 Chord

Chord methods use the boundary of the convex hull area of overlapping objects. This boundary consists of a finite union of curved segments, each of which is an arc of the object's contour C, or a line with its endpoints on C called a chord. If L_i is one of those lines (a chord) and C_i is a segment of C with the same endpoints, the union of L_i with C_i generates a simple closed curve, that determines a convexity defect. Chord methods use the convex hull contour and the convexity defects to detect concavities. The primary idea of these approaches is to identify the furthest points between the contour and convexity defect.

Multiple solutions used chord analysis to extract convex points. Farhan et al. [9] proposed a method for obtaining concave points by evaluating a line fitted to the contour points, where a concave point is detected if the line that joins two contour points does not reside inside the object. Kumar et al. [15] proposed boundaries of concave regions and their corresponding convex hull chords. The concave points were defined as the points on the boundaries of the concave regions that maximized the perpendicular distance from the convex hull chord. Similarly, Yeo et al. [29] and LaTorre et al. [16] applied multiple constraints to the area between the convexity defect and the contour to determine its quality.

Chord methods consider assume that only one concave point exists for each convexity defect, but this assumption may not always hold in clusters containing more than two objects, therefore, some concave points are missdetected.

2.3 Polygon approximation

Polygon approximation is a set of methods that represents the contours of objects through a sequence of dominant points. These methods aim to remove noise by approximating the contour of a simpler object.

Bai et al. [2] developed a new class of algorithm to follow this approximation. Their algorithm analysed the difference between a set of contour points and a straight line connecting their extremes. Points at a large distance from this previously defined line were considered dominant points. Chaves et al. [5] used the well-known Ramer–Douglas–Peucker (RDP)

[8] algorithm to approximate the contour, and concave points were detected using conditions on the gradient direction at the contour. A similar approach for detecting these points was presented in [32] by Zafari et al., in which the authors proposed a parameter-free concave point detection to extract dominant points. They selected the concave points using a condition based on the scalar product of consecutive dominant points described in [33].

The same authors [30] used a modified version of the curvature-scale space proposed in [14] to find interest points. Finally, they discriminated them between concave and convex points.

Zhang et al. [35], similarly to [30], used a modified version of the curvature-scale space to approximate an object. By applying the previously described object approximation algorithms, they obtained a set of dominant points, from which the concave points were identified. The concave points were detected by evaluating the angular changes in these dominant points, and these angular changes were evaluated using the *arctangent* criteria. The points with an angular change higher than a threshold were classified as concave points.

These methods are highly parametric and are not robust changes in object size. Another limitation of these methods is that they deform the original silhouette to simplify it. The approximation is a tradeoff between the lack of precision in the position of the concave points and the smoothness applied to the contour. This tradeoff affected the final results.

2.4 Curvature

Methods that fall into this category identify the concave points as the local extremes of the curvature. The curvature κ at each point $q_i = (x_i, y_i)$ on the contour is computed as:

$$\kappa(q_i) = \frac{x_i' \cdot y_i'' - y_i' \cdot x_i''}{(x_i'^2 + y_i'^2)^{3/2}}, \tag{1}$$

where x_i and y_i are the contour points coordinates.

Wen et al. [27] calculated the derivative by convolving the boundary using Gaussian derivatives. González-Hidalgo et al. [11] used the k-curvature and the k-slope to approximate the value of the curvature. The dominant points, the ones with the highest curvature, can be located in both the concave and convex regions of the contours. In [31] three different heuristics were described to detect the concave points.

These methods tend to fail when multiple concave points are located in small areas. The main reason for this is the loss of precision caused by the approximating the curvature value. Another problem is the existence of noise, which tends to be identified as changes on the curvature, one way to fix this problem is to use a coarser approximation.

2.5 Other methods

Despite the taxonomy proposed by Zafari et al. [31], other techniques exists for determining concave points that do not fall into any of the previous categories. Some of these studies are described below.

Fernández et al. [10] defined a sliding window for the contour and calculated the proportion of pixels that belongs to the object and the pixels that belong to the background on this window. This proportion determines the likelihood of concavity existing at the evaluated point. He et al. [13] adapted this method for use in three dimensions. The optimum results were obtained in scenarios with high concavity. This method is highly sensitive to changes in the size of objects, and its accuracy decreases in the presence of noise. These two problems are a consequence of the lack of generalizability of the method.

Wang et al. [26] proposed a bottleneck detection method. They defined a bottleneck as a set of two points that minimized the Euclidean distance and maximized the distance on a contour. The set of points that defined the bottleneck were concave points. A cluster may contain multiple bottlenecks. This algorithm was unable to determine the number of elements belonging to a cluster, and the number of elements was a hyperparameter of the algorithm. Another limitation is that they did not consider clusters with an odd number of concave points.

Zhang and Li [34] proposed a method for determining concave points using two-step algorithm. First, they detected a set of candidate points by using a Harris corner detector [12]. Second, they selected concave points using two different algorithms: one for obvious concave points and the other for uncertain concave points. Their algorithm has a high number of parameters. This higher number of parameters has two different consequences: on one hand the method is highly adjustable to the features of the overlapping objects; on the other hand, this number of parameters results in a high complexity of the algorithm.

3 Methodology

Motivated by the performance of state-of-the-art methods and with the aim of improving the results on the challenging task of separating overlapped objects, we propose a new method based on the work of González-Hidalgo et al. [11] for detecting concave points. The entire process is illustrated in Fig. 1.

3.1 Pre-procesing

We applied a segmentation algorithm to obtain objects from the images. Depending on the complexity of the data, we can use different techniques: from Otsu [20], a simpler, non-parametric, thresholding technique, to a more complex approach such as Chan-Vese [3] that needs to be parameterized. Once the object was segmented we obtained its contour and removed the noise generated by the segmentation technique using the RDP algorithm proposed by [8]. Finally, we compute the curvature at each point using a well-known technique called k-curvature [21]. This technique considers the curvature of each point as the difference in its slope. The k-curvature is separable, which allows us to perform the calculation independently for each direction.

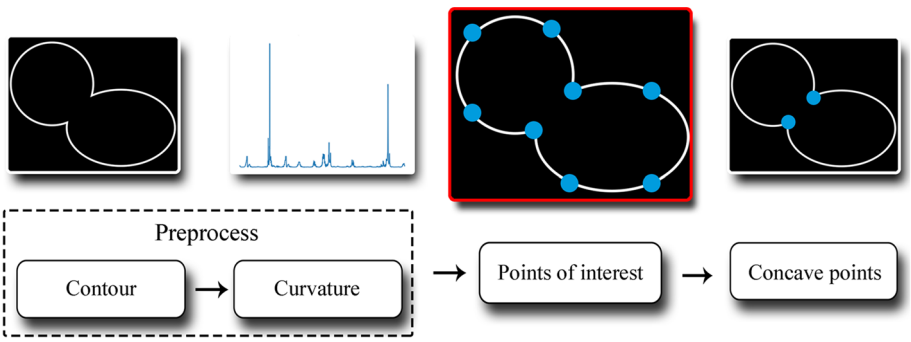


Fig. 1 Flow chart of the proposed method to obtain concave points from a contour

3.2 Points of interest

In this section, we define a methodology to find points of interest by analysing the curvature of each contour point, which can be either a concave or convex point, in next step, we will identify the concave points.

The first step in obtaining the points of interest involved determining the subsets of contiguous points that included those with the highest curvature. We referred to these subsets as regions of interest. All points belonging to a region of interest (ROI) have a curvature greater than a certain threshold and are defined by their start and end points. The process to determine the regions of interest is a recursive procedure described below, which provides a list of regions of interest and ensures the presence of a point of interest within each ROI.

Let C and t be the set of contour points and initial threshold for the curvature value, respectively. We also defined two other thresholds to control the length of the regions of interest, namely l_{min} and l_{max}. The l_{min} value aims to avoid an excessive number of regions and reduce the effect of noise on the object contour. The l_{max} value is useful for preventing the point of interest from being located in an excessively large region, because we were interested in extracting only one concave point from each region. The optimal values of both parameters are closely related to the size of the object handled by the algorithms. The process of determining these values is highly dependent on the specific problem. Section 4.3 describes the process of determining these values for two different problems. The recursive procedure that allowed us to detect the regions of interest is as follows:

Step 1: Taking the contour points C, we constructed a list of regions of interest `l_regions` selecting all non-overlapping sets of adjacent points which curvature was greater than t. If necessary we updated t. The original image is shown in Fig. 2a. Figure 2b depicts the list of regions of interest detected, and the different regions of interest are marked in different colors.

Step 2: Let r be a region of interest in `l_regions`, and we denote in length by $\ell(r)$. Three cases are considered in this study:

- Case 1: If $l_{min} \leq \ell(r) \leq l_{max}$, we continued with the next region of interest.
- Case 2: If $\ell(r) > l_{max}$, we returned to Step 1 with r and $t+\delta t$. We updated `l_regions` with the regions where r is divided, and r is removed from the list.
- Case 3: If $\ell(r) < l_{min}$, we combined the region r with its closest region. That is, we identified $r_{closest}$ such that $d(r, r_{closest}) < k$, where k was the displacement allowed to calculate the k-curvature. Let $r_{new} = r \cup r_{closest}$ be the new region.

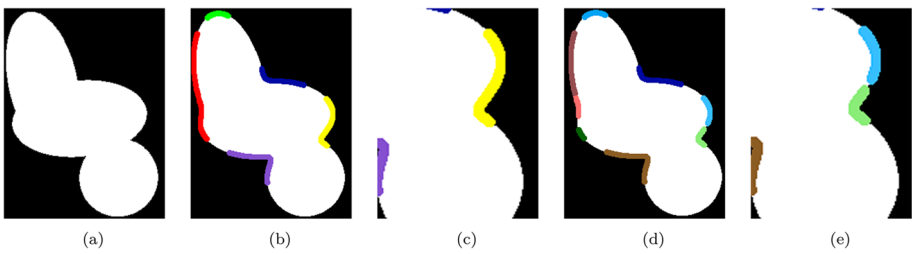


Fig. 2 Successive steps to identify regions of interest. (a) Original image. (b) Regions of interest detected in Step 1. Different regions of interest are marked with different colors. (c) Detail of a region of interest detected in Step 1. (d) Regions of interest detected by the recursive procedure, Step 2. (e) Detail of Step 2

a) If $\ell(r_{new}) > l_{min}$, we returned to Step 1 with r_{new} and $t + \delta t$. If necessary, we updated `l_regions`.
b) If $\ell(r_{new}) < l_{min}$, we moved to Case 3 with r_{new} instead r.

The process ended when all the regions of interest in the `l_regions` had a length between l_{min} and l_{max}. In Fig. 2d, the final list of detected regions is presented. As mentioned previously, different regions of interest are marked in different colors. Moreover, in Fig. 2c, a zoomed view of an initial region of interest is shown, and in Fig. 2e, the final state is depicted. As observed, the initial region of interest is divided into two new regions, each containing a point of interest.

After this recursive procedure, we obtained a list of regions of interest, where each region contained a point of interest, that is, a concave or convex point. Finally, we identified one interest point for each region of interest. We used the weighted median of the curvature to locate them because it is a well-known technique that assumes that the point of interest is located near the center of the region; however, this central position is not a perfect location, and can be improved.

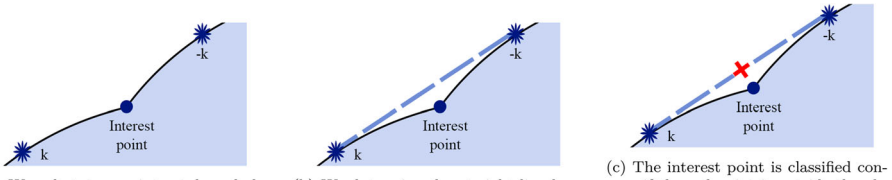
3.3 Selection of concave points

After identifying the set of points of interest, we proceeded to detect the concave ones. This part of the algorithm is based on an analysis of the relative position of the neighbourhood of each point [11], we include this for the sake of completeness with a more detailed explanation. The classification phase consists of the following three steps:

1. **Determine two k-neighbour points**: We selected two points on the contour, which were located at a distance of k and $-k$ relative to the interest point. This step is described in Fig. 3a.
2. **Definition of a line between the k-neighbours**: We constructed a straight line between the points selected in the previous step, see Fig. 3b.
3. **Middle point of the line**: We classified a point as concave if the middle point of the previously defined line was outside the object; otherwise, the point was classified as convex. See Fig. 3c.

4 Experimental setup

The experimental setup was designed to evaluate the performance of the concave point detector compared to state-of-the-art methods and demonstrate that better concave point detection implies a better segmentation of overlapping objects.



(a) We select two points at k and -k distance from the interest point.

(b) We determine the straight line between previously selected points.

(c) The interest point is classified concave if the red point is outside the object.

Fig. 3 Steps to discriminate between concave and convex points

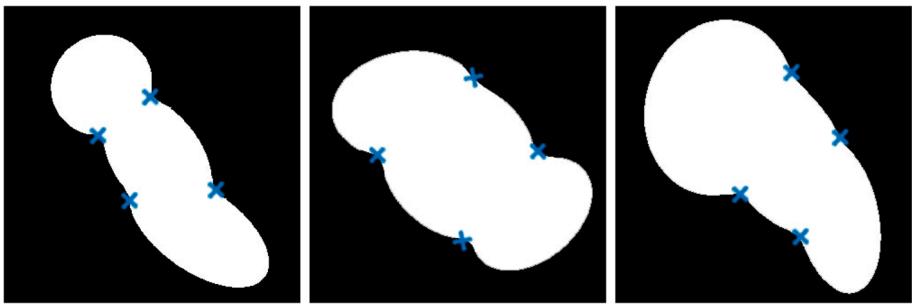


Fig. 4 Three examples from the OverArt dataset. In blue color the concave points, in white color the clusters defined by three overlapped ellipses

4.1 Datasets

Two sets of images were used in this study. On the one hand, we created a set of synthetic images, that we called OverArt dataset. It contains 2000 images, each with three overlapping objects, with annotated concave points as the groundtruth. On the other hand, we used the *ErythrocytesIDB2* dataset of real images from González-Hidalgo et al. [11], it contains 50 images of peripheral blood smears samples of patients with sickle cell anaemia. We used this dataset to check whether the spacial precision of the concave point detector method affects the results of overlapping object segmentation.

4.1.1 OverArt dataset

We generated the OverArt dataset to obtain the ground truth of the concave points on overlapped objects. Each image in the dataset contains a cluster of three overlapped ellipses. We placed three ellipses in order to simulate the real case of the red blood cells in microscopic images.

The code is available at https://github.com/expainingAI/overArt. The data set of images used in this study was created choosing nusing 42 as the random seed.

Figure 4 shows three different examples from this dataset.

The ellipses of each image are defined by three parameters: rotation, *feret* diameter size, and centre. These values are randomly generated using the set of constraints listed in Table 1.

Table 1 Parameters we used to generate the OverArt dataset

Parameter	Value
Minimum *feret*	45 px
Maximum *feret*	100 px
Minimum distance between centers	45 px
Maximum distance between centers	85 px
Minimum rotation	0°
Maximum rotation	360°

The table detail the range of the values that defines each ellipse. The distance metrics are defined in pixels (px). Angles are defined in degrees.

To construct each cluster, we located the first ellipse at the center of the image. The positions of the other two ellipses are related to this first ellipse. We randomly selected the location of the second ellipse inside the area defined by the minimum and maximum distances to the center of the first ellipse. Finally, we followed the same process as for the third ellipse, which was randomly placed inside the area defined by the minimum and maximum distances to the center of the first and second ellipses.

To compare the precision of the different methods in finding the concave points, we needed the ground truth of its location. We calculated this value and added it to the dataset. A concave point is defined as the position where two or more ellipses intersect and must be located over the contour that defines the overlapping region.

Overlapping objects are defined by the ellipse equation, see (2) and (3). For each image of the dataset, we obtained the positions of all concave points by analytically solving (4).

$$\lambda_1 = \frac{((x - center_x)\cos(\phi) + (y - center_y)\sin(\phi))^2}{(a^2)}, \tag{2}$$

$$\lambda_2 = \frac{((x - center_x)\sin(\phi) - (y - center_y)\cos(\phi))^2}{(b^2)}, \tag{3}$$

$$\lambda_1 + \lambda_2 = 1, \tag{4}$$

where x and y are the unknown variables, $center_x$, $center_y$ defines the central point of the ellipse, and ϕ is the angle between the horizontal axis and the ellipse feret. Finally, a and b represent the semi-axis.

4.1.2 ErythrocitesIDB2 dataset

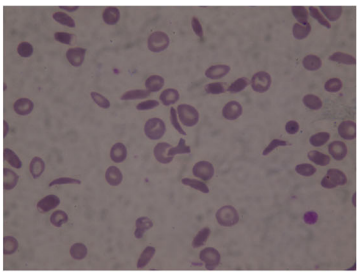
In this study, we used microscopic images of blood smears collected from *ErythrocytesIDB2* [11], available at http://erythrocytesidb.uib.es/. The images consist of peripheral blood smears samples of patients with sickle cell anaemia classified by a specialist from "Dr. Juan Bruno Zayas" Hospital General in *Santiago de Cuba*, Cuba. Specialist criteria were used as an expert approach to validate the results of the classification methods.

Patients with sickle-cell disease (SCD), are characterized by red blood cells (RBCs) with a sickle or half-moon shape, instead of the smooth, circular shape of normal cells. To confirm the SCD diagnosis, peripheral blood smear samples were analysed by microscopy to check for the presence of sickle-shaped erythrocytes and compare their frequency to that of normal red blood cells. Peripheral blood smear samples always contain overlapping or clustered cells, and the sample preparation process can affect the number of overlapping erythrocytes in the images studied. Clinical laboratories typically prepare blood samples for microscopic analysis using the dragging technique, in which more cell groups are apparent in the samples because of the spreading process [11].

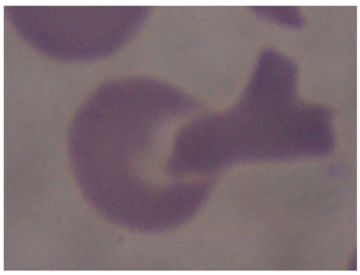
Each image was labeled by a medical expert. There are 50 images with different number of cells (see Fig. 5), this set of images contains 2748 cells. These cells belong to three classes defined by medical experts. These are circular, elongated, and other, as shown in Fig. 6.

4.2 Performance measures

The results of our algorithm should be measured using multiple numerical and well-defined metrics to ensure quality. The objective of these metrics is to evaluate the precision of predicting the position of a concave point and the resultant impact on the splitting of overlapped objects. We used five metrics: MED, F1-Score, SDS_Score, MCC and CBA.



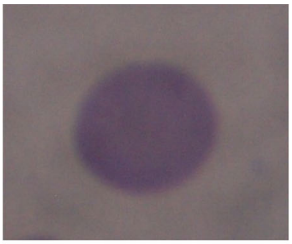
(a) Example of image of the ErythrocitesIDB2 dataset.



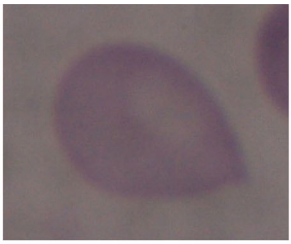
(b) Detail of the image containing a cluster.

Fig. 5 Sample of patient with sickle cell anemia from ErythrocitesIDB2 dataset

- **Mean of the Euclidean distance (MED).** Let $f = \{C_i\}_{i=1}^{p}$ be the detected concave points for the proposed method for a given image, and $GT = \{GC_i\}_{i=1}^{l}$ be the ground truth concave points of that image, we know that it may be that $l \neq p$. However, for each point GC_j exists $C_{i_j} \in f$ such that $d(GC_j, C_{i_j})$ is minimum, then we define the MED performance measure by (5).
- **F1-Score**. It is a standard and widely used measure. It is the harmonic mean of precision and recall, see (8). The precision and the recall depends on the number of false positives (FP), true positives (TP) and false negatives (FN). We also included the precision and the recall to the results in order to explain the F1-Score.
- **Matthew's Correlation Coefficient (MCC)**. Introduced in [17], is a correlation measure between the prediction and observation. We used the adaptation proposed by Mosley et al. [19] for multi class problems, see (9). This metric lies in the [-1,1] range, where -1 represents perfect missclasification, 1 a perfect classification, and 0 a random classification. It is designed to deal with unbalanced data.
- **Class Balance Accuracy (CBA)**. Introduced by Mosley et al. [19]. It represents the overall accuracy measure built from an aggregation of individual class metrics. This measure is designed to deal with unbalanced data. See (10).
- **Sickle cell disease diagnosis support score (SDS-Score)**. Proposed by Delgado-Font et al. [7], the SDS-Score indicates the usefulness of the method for the diagnosis of



(a) Example of healthy *circular* cell.



(b) Example of *other* cell.

(c) Example of *elongated* cell.

Fig. 6 Examples of the three types of cells present in the ErythrocitesIDB2 dataset. *Elongated* cells are also known as sickle cells

patients with sickle cell disease. This metric does not consider as a mistake a misclassification between elongated and other cells (or vice versa), due to the nature of the disease. See (11).

$$MED = \frac{\sum_{j=1}^{l} d(GC_j, C_{i_j})}{l}, \tag{5}$$

$$Precision = \frac{TP}{TP + FP}, \tag{6}$$

$$Recall = \frac{TP}{TP + FN}, \tag{7}$$

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \tag{8}$$

$$MCC = \frac{\sum_{i,l,m=1}^{z} c_{ii} \cdot c_{ml} - c_{li} \cdot c_{im}}{\sqrt{\sum_{z=1}^{n}(\sum_{l=1}^{n} c_{lz})(\sum_{\substack{f,g=1 \\ f \neq z}}^{z} c_{gf})} \sqrt{\sum_{l=1}^{z}(\sum_{i=1}^{z} c_{il})(\sum_{\substack{f,g=1 \\ f \neq l}}^{z} c_{fg})}}, \tag{9}$$

$$CBA = \frac{1}{z} \cdot \sum_{i=1}^{z} \frac{c_{ii}}{\max(\sum_{j=1}^{3} c_{ij}, \sum_{j=1}^{3} c_{ji})}, \tag{10}$$

$$SDS - Score = \frac{\sum_{i=1}^{3} c_{ii} + c_{23} + c_{32}}{\sum_{i=1}^{3}\sum_{j=1}^{3} c_{ij}}, \tag{11}$$

where c_{ij} is the number of elements of class i predicted as the class j and z the number of classes. In particular, c_{23} represents the cells predicted to be other when they are elongated and c_{32} represents the other cells predicted to be elongated.

We used a paired t-test to check the difference between the F1-Score of our results and those of state-of-the-art methods. The null hypothesis was that our results were greater. Previously, the normality of the data distribution was checked, by using the Shapiro-Wilk test.

4.3 State-of-the-art methods

In the Introduction, we study state-of-the-art methods that separate overlapping objects by finding concave points. To perform our experiments, we selected a representative subset, excluding the methods we could not reproduce because of the absence of information in the original paper and the lack of access to the source code.

We used the original code of Zafari et al. [30] and González-Hidalgo et al. [11]. In addition to the two previously mentioned methods, we considered the following methods: LaTorre et al. [16], Fernández et al. [10], Song and Wang [24], Chaves et al. [5], Bai et al. [2], Wang et al. [26] and Zafari et al. [32].

As we did not have the values of the hyperparameters of the methods and in order to make a fair comparison, we performed an exhaustive search to obtain the hyperparameters of each method for each experiment, see Table 2, even if we had their original values.

4.4 Experiments

We conducted two experiments to study the two different characteristics of the proposed method. The first is the precision of the concave points detection. Second, how the detection precision affects the posterior segmentation of overlapping objects.

4.4.1 Experiment 1

This experiment aimed to compare the detection capacity and spatial precision of the proposed method with those of the state-of-the-art methods. We used the generated OverArt dataset that because it contains the position of each concave point. The training and test sets were constructed by randomly selecting 1000 images, and it is important to note that the intersection between both sets was empty.

To evaluate the performance of each method, we used two different performance measures from Section 4.2: the Mean Euclidean Distance (MED) and the F1-Score. To compute the F1-Score we matched each detected concave point with a ground truth point. We matched two points if the distance was smaller than the integer threshold, θ, we set it experimentally. If there were more than one candidate, the nearest candidate was selected. We considered a false positive as a predicted point that did not match a ground truth point. A false negative was considered when there was no candidate for a ground-truth point.

Table 2 First column summarizes the original parameters of each method

Method	Original hyperparameters	OverArt hyperparameters	ErythrocitesIDB2 hyperparameters
Proposed method	-	k: 7 px, l_{min}: 2 px, l_{max}: 11 px, ϵ: 0.2	k: 9 px, l_{min}: 6 px, l_{max}: 25 px, ϵ: 0.1
LaTorre et al. [16]	Min. concavity area: 20 px, Min. distance to BB: 5 px, Degree: 10%	Min. concavity area: 55 px, Min. distance to BB: 1 px, Degree: 10%	Min. concavity area: 40 px, Min. distance to BB: 15 px, Degree: 10%
Fernández et al. [10]	Environment size: 5 × 5, Concavity threshold: -	Environment size: 7 × 7, Concavity threshold: 1.9	Environment size: 5 × 5, Concavity threshold: 1
González-Hidalgo et al. [11]	k: 17 px, sT:0.14	k: 13 px, sT: 0.3	k: 17px, sT:0.14
Zafari et al. [30]	k: 15 px	k: 5 px	k: 20 px
Chaves et al. [5]	ϵ: -, k: 2 px, Concavity threshold: $\pi/2$	ϵ: 0.1, k: 11 px, Concavity threshold: 0.	ϵ: 0.5, k: 2 px, Concavity threshold: 0.5
Bai et al. [2]	dTh: 3px, Point distance: 1 px, nStep: 2 px	dTh: 1 px, Point distance: 7 px, nStep: 3 px	dTh: 1 px, Point distance: 2 px, nStep: 2 px

Second and third columns summarize the set of hyperparameters for *Experiment 1* and *Experiment 2* respectively.

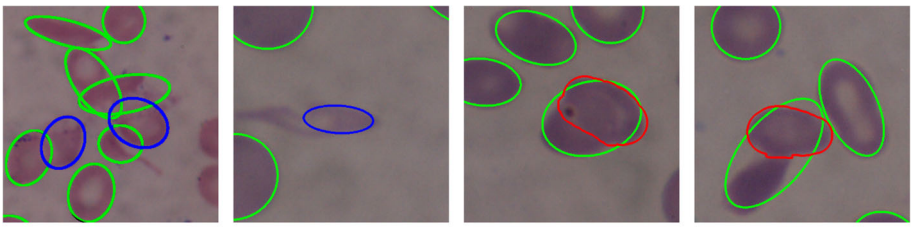


Fig. 7 Examples of false positives and false negatives using the proposed algorithm. Figure 7a and b depict the detection of none existing cells in blue (false positives). Figure 7c and d shows in red cells from the ground truth that are not detected (false negatives). Correct detections (predictions overlapped to a ground truth cell) in all sub-figures are shown in green

To evaluate the performance of each method, we searched for the best set of hyperparameters to maximize the F1-score. We performed this process for each θ value between 1 and 20, allowing us to observe the evolution of the performance of each method when different thresholds were set.

4.4.2 Experiment 2

This experiment was designed to determine how the precision of concave point detection affects the division of overlapping objects in a real-world scenario. We used the ellipse fitting method proposed by Gonzàlez et al. [11] to divide the overlapped objects from the detected concave points. After completing this step, the ground truth was compared with the predicted objects. The *ErythrocytesIDB2* dataset was used in this experiment. As a training set, we randomly selected 70% of the images, and there are 34 images that contain 1825

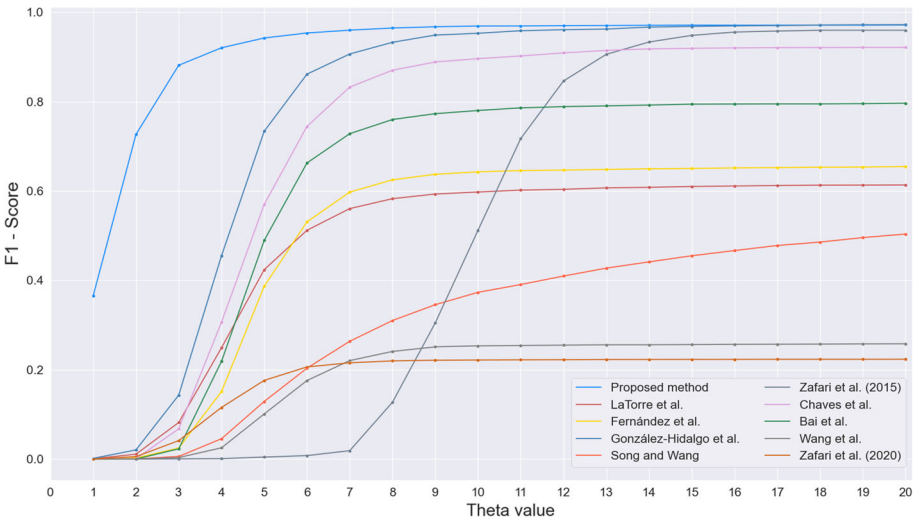


Fig. 8 F1-score on the test set of OverArt dataset with different values of θ, that is the maximum allowed distance to make a match between a detection and a ground-truth point. The proposed method outperforms the methods described in Section 4.3 for all θ values

cells. The remaining 30% of the images (16 images containing 980 cells) were used as test set.

The problem addressed in this experiment was a multi-class problem; therefore, we used the CBA, MCC, and SDS-Score and the adapted version of the F1-Score averaging the results for each class. We considered the prediction of a non-existing cell in the ground truth as a false positive, and the omission to predict an existing cell in the ground truth as a false-negative. Figure 7 depicts examples of false-positives and false-negatives detections.

5 Results and discussion

In this section, we analyse and discuss the results of experiments on the datasets described in Section 4.1.

5.1 Experiment 1

Figure 8 depicts the F1-score we obtained using the test set for each value of θ and the best hyperparameters for each method. We can observe that the proposed method outperforms the state-of-art methods, and the difference is larger when θ is lower than 10 pixels, when it is more difficult to match a detection and a ground-truth point.

Tables 3 and 4 list the results obtained for the detection of the concave points in *Experiment 1* when $\theta = 15px$. We selected this value according to the results depicted in Fig. 8, where all methods had a stable F1-score. The tables summarize the precision, recall, F1-Score and MED values of the concave-point detection methods. We also added the standard deviation (STD) of the MED measure in order to provide complementary information. In our evaluation, it was important to obtain a small value of MED; however, it was also important to ensure that this measure was not scattered.

Table 3 summarizes the results obtained for the training set. We can observe that the proposed method achieved the best value for the F1-Score and it is almost tied with González-Hidalgo et al. method. The methods of Chaves et al. and Zafari et al. [30] also yielded satisfactory results for this measure. The other methods had a strong imbalance

Table 3 Results of the *Experiment 1* using 1000 images of the training set from synthetic dataset

Method	Precision	Recall	F1-Score	MED	STD
Proposed method	**0.998**	0.943	**0.969**	**1.777**	**1.551**
LaTorre et al. [16]	0.445	0.928	0.604	9.270	20.226
Fernández et al. [10]	0.601	0.701	0.646	23.823	41.759
González-Hidalgo et al. [11]	0.982	0.953	0.967	4.544	3.682
Song and Wang [24]	0.481	0.421	0.449	41.213	51.764
Zafari et al. [30]	0.978	0.911	0.943	10.193	4.337
Chaves et al. [5]	0.991	0.847	0.913	5.404	5.795
Bai et al. [2]	0.992	0.646	0.783	5.0948	3.057
Wang et al. [26]	0.275	0.255	0.264	83.742	62.325
Zafari et al. [32]	0.128	**0.984**	0.226	4.470	3.120

MED is the mean of the euclidean distance from a detected point to the closest ground truth point, *STD* is its standard deviation. In bold, the best results for each measure

Table 4 Results of the *Experiment 1* using 1000 images of the test set from synthetic dataset

Method	Precision	Recall	F1-Score	MED	STD
Proposed method	**0.995**	0.948	**0.971**	**1.762**	**2.023**
LaTorre et al. [16]	0.454	0.931	0.610	9.129	20.223
Fernández et al. [10]	0.597	0.714	0.650	22.305	39.853
González-Hidalgo et al. [11]	0.977	0.959	0.968	4.504	3.002
Song and Wang [24]	0.511	0.461	0.484	38.156	51.613
Zafari et al. [30]	0.979	0.919	0.948	10.176	5.326
Chaves et al. [5]	0.987	0.859	0.919	5.414	5.707
Bai et al. [2]	0.989	0.664	0.794	5.338	6.276
Wang et al. [26]	0.239	0.221	0.229	96.446	70.646
Zafari et al. [32]	0.125	**0.986**	0.223	4.462	3.443

MED is the mean of the euclidean distance from a detected point to the closest ground truth point, *STD* is its standard deviation. In bold, the best results for each measure

between the results from precision and recall; higher precision usually provokes a lower recall value and vice versa. When this situation occurs, the methods obtain a low F1-Score performance. For the MED measure, the proposed method obtained the best result with the lowest standard deviation. This indicates that the values were close to the mean and less scattered than those of the others.

Table 4 summarizes the results obtained for this test set. These results were very similar to those obtained using the training set. In addition, the proposed method achieved the best F1-Score and MED with a very low STD.

From the previous analysis, we can state that the proposed method identifies the concave points with the highest balance between precision and the recall, which implies lower rates of false positives and negatives. Lower values of MED and STD metrics indicate that the detected points are close to the ground-truth points, indicating a high degree of spatial precision.

Table 5 Results of the *Experiment 2* using the 34 images of the train set from *ErythrocytesIDB2*

Method	Precision	Recall	F1-Score	SDS-Score	CBA	MCC
Proposed method	0.862	**0.882**	**0.872**	**0.908**	0.659	**0.739**
LaTorre et al. [16]	0.813	0.806	0.808	0.873	0.637	0.642
Fernández et al. [10]	0.777	0.789	0.781	0.846	0.569	0.587
González-Hidalgo et al. [11]	0.847	0.852	0.850	0.902	0.649	0.703
Song and Wang [24]	0.669	0.849	0.746	0.842	0.545	0.595
Zafari et al. [30]	0.787	0.871	0.826	0.881	0.636	0.687
Chaves et al. [5]	0.848	0.873	0.860	0.901	0.648	0.723
Bai et al. [2]	**0.871**	0.870	0.870	0.907	**0.667**	0.731
Wang et al. [26]	0.729	0.842	0.781	0.856	0.594	0.621
Zafari et al. [32]	0.704	0.783	0.724	0.822	0.509	0.563

SDS is the sickle cell diagnosis support score. *MCC* is the Matthew's Correlation Coefficient and *CBA* is the Class Balance Accuracy. In bold, the best results for each measure

Table 6 Results of the *Experiment 2* using the 16 images of the test set from *ErythrocytesIDB2*

Method	Precision	Recall	F1-Score	SDS-Score	CBA	MCC
Proposed method	**0.856**	**0.861**	**0.858**	**0.888**	**0.665**	**0.724**
LaTorre et al. [16]	0.813	0.783	0.798	0.865	0.602	0.631
Fernández et al. [10]	0.766	0.752	0.758	0.843	0.598	0.601
González-Hidalgo et al. [11]	0.838	0.821	0.829	0.877	0.625	0.677
Song and Wang [24]	0.665	0.825	0.735	0.829	0.592	0.595
Zafari et al. [30]	0.766	0.842	0.801	0.845	0.650	0.655
Chaves et al. [5]	0.826	0.846	0.836	0.873	0.634	0.684
Bai et al. [2]	0.842	0.836	0.838	0.879	0.638	0.693
Wang et al. [26]	0.711	0.821	0.762	0.829	0.618	0.592
Zafari et al. [32]	0.696	0.728	0.697	0.806	0.532	0.536

SDS is the sickle cell diagnosis support score. *MCC* is the Matthew's Correlation Coefficient and *CBA* is the Class Balance Accuracy. In bold, the best results for each measure

5.2 Experiment 2

We summarize the results for *Experiment 2* in Tables 5 and 6 obtained with the images in ErythrocitesIDB2 dataset. As in the previous experiment, we separated the results in two different tables, for the training set and the test set, respectively. The results of the training set are presented in Table 5. The proposed method surpasses the CBA measure using the method described by Bai et al. but achieved the best results for all other metrics. Table 6 presents the results obtained from the test set. It should be noted that when the proposed method was faced with unseen data, it achieved the best values for all metrics. Figure 9 presents the results of the *Experiment 2* for an image in the test set. In the original image, we observed two different clusters of cells. The proposed method and the one by González-Hidalgo et al. were the only ones capable to segment both clusters correctly, that shows the difficulty to solve this problem as we had the overlapping zones that did not provide any information. As shown in the figure, the application of different concave point detection

Table 7 Results of applying the t-test between our proposed method and the rest of state-of-the-art methods

Methods	p-value	t stadistic
LaTorre et al.	$\mathbf{2.46 \times 10^{-3}}$	3.051
Fernández et al.	$\mathbf{3.62 \times 10^{-5}}$	4.682
Gonzàlez-Hidalgo et al.	$\mathbf{6.41 \times 10^{-2}}$	1.565
Song and Wang	$\mathbf{1.11 \times 10^{-4}}$	4.378
Zafari et al.	**0.01546**	2.297
Chaves et al.	0.1693	0.973
Bai et al.	0.1824	0.922
Wang et al.	$\mathbf{5.42 \times 10^{-4}}$	3.707
Zafari et al.	$\mathbf{7.15 \times 10^{-7}}$	5.516

The alternative hypothesis is: the mean of our method is greater than the others. In bold, the methods that are significantly worse than the proposed method

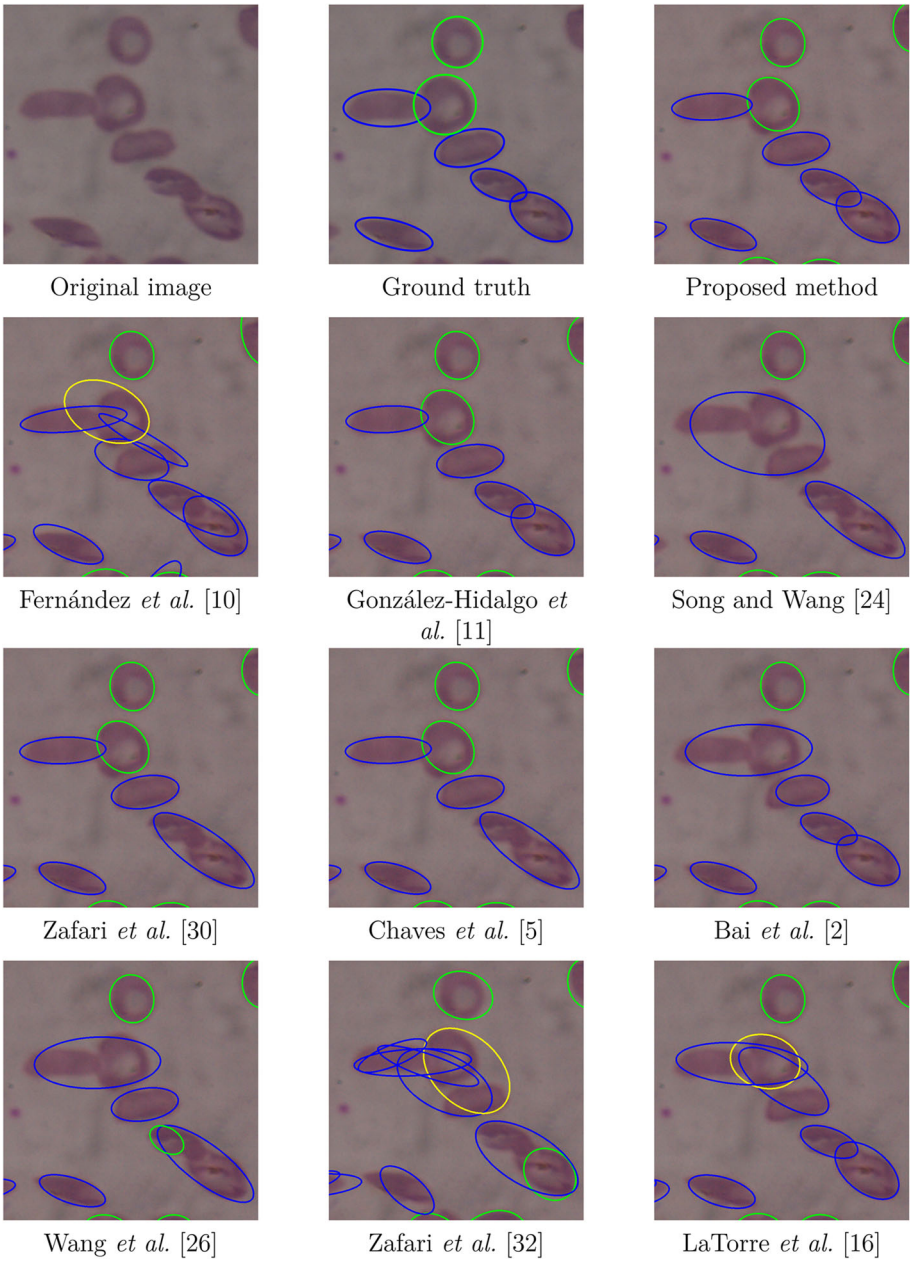


Fig. 9 Example of results obtained from the test set of ErythrocitesIDB2 dataset for each method. Colors used in this figure: blue for the elongated cells, green for the circular cells, yellow for the other cells. Only two methods can segment both clusters correctly

methods led to different object segmentation results, and it should be noted that we used the same algorithm to segment the clusters for all approaches.

We used a paired t-test to compare the methods and determine if the proposed method significantly outperformed the other methods. The numerical results of the algorithm comparison for the F1-score performance measure are summarised in Table 7. Using the F1-score, we compared our method with one of the other methods using a t-test with a confidence level of 95%. The result of the application of this test is that the proposed method outperformed significantly most of the other proposals, except the ones based on curvature estimation as González-Hidalgo et al. [11], Chaves et al. [5] and Bai et al. [2], where the improvement was not statistically significant.

From the analysis of the results obtained with *Experiment 2*, we can determine that the increase in precision in the detection of concave points implies an improvement in the results of the segmentation of the overlapping objects.

From the results obtained in the second experiment, we can conclude that an increase in the precision of concave-point detection improves the result of a later segmentation method.

Finally, note that the proposed method can be considered transparent [1], because it has the ability of simulatability (being simulated or thought about strictly by a human), decomposability (explaining each part of the method), and exhibit algorithmic transparency (the user can understand the process followed by the method to produce any given output from its input data). This is particularly important in health, to trust the behaviour of intelligent systems.

6 Conclusions

Concave point detection is the first step in segmenting overlapping objects in images. The existence of these clusters reduces the information available in some areas of the image, making it still a challenging problem.

The methodology proposed in this study is based on the curvature approximation of each point on the contour of the overlapping objects. First, we selected regions with higher curvature levels because they contained at least one interest point. Second, we applied a recursive algorithm to refine the previously selected regions. Finally, we obtain a concave point for each region.

We included the implemented code, confusion matrices with the raw data of all studied methods, and images used as training and test sets to allow researchers to compute other metrics more easily (see https://github.com/expainingAI/overlapped-objects). As an additional contribution we constructed and opened to the scientific community a synthetic dataset to simulate overlapping objects, we provided the position of the concave points as a ground-truth, as far as we know, is the first public dataset containing overlapping objects with annotated concave points. We used this dataset to compare the detection capacity and spatial precision of the proposed method with those of state-of-the-art methods (see https://github.com/expainingAI/overArt). For scientific progress, it would be beneficial if the authors published their raw data, code, and image datasets.

Finally, as a case study, we evaluated the proposed concave-point detector and state-of-the-art methods with a well-known application, such as the splitting of overlapping cells in microscopic images of peripheral blood smear samples of RBC of patients with sickle-cell disease. The goal of the case study was to determine whether the spatial precision of the concave point detector method affects the results of a classification algorithm for the morphology of RBC in a real-world scenario.

The experimental results demonstrated that the proposed method yielded better results for both synthetic and real datasets, in using multiple standard metrics. We can conclude that the proposed methodology detects concave points with the highest exactitude, as it obtain lower values in the *MED* metric and a small standard deviation. In the same experiment, the best F1-Score, that means a good balance between precision and recall for detecting concave points. We designed a second experiment to determine how the precision of concave point detection affects the division of overlapping objects in a real-world scenario. From the results we can conclude that a method with higher precision for finding concave points, such as the method proposed in this paper, helps achieve better cell classification. Finally, it is important to note that this method is not limited to the case study performed in this work; it can also be used for other applications where separation between overlapping objects is required. Furthermore, methods based on the detection of concave points can provide good segmentation without large dataset or input image-size constraints unlike deep learning methods. Moreover, the detection of concave points for segmenting the overlapping objects can be considered transparent because presents simulatability, decomposability as well as algorithmic transparency [1].

As future work, we have identified one key area for improvement: the definition of parameter values. Currently, the values of these parameters are highly dependent on the problem, and their successful utilization requires extensive knowledge of the problem at hand. This can limit the applicability of the proposed method, particularly in cases in which the user does not have the necessary expertise. To address this limitation, future research should focus on developing an automatic method for defining these parameter values. Such an approach would enable the algorithm to adapt more easily to new problems without requiring significant domain knowledge.

Acknowledgments This work is part of I+D+i Project PID2019-104829RA-I00 "EXPLainable Artificial INtelligence systems for health and well-beING (EXPLAINING)" funded by MCIN/AEI/10.13039/501100011033, and is part of I+D+i Project PID2020-113870GB-I00 - "Desarrollo de herramientas de Soft Computing para la Ayuda al Diagnóstico Clínico y a la Gestión de Emergencias (HESOCODICE)", funded by MCIN/AEI/10.13039/501100011033/. Miquel Miró also benefited from the fellowship FPI/035/2020 (Govern de les Illes Balears)

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Author Contributions **Miquel Miró-Nicolau**: Software, Visualization, Formal analysis , Writing - Original Draft.
Gabriel Moyà Alcover: Conceptualization, Validation, Project administration, Review & Editing.
Manuel González-Hidalgo: Methodology, Formal analysis, Writing - Review & Editing.
Antoni Jaume-i-Capó: Methodology, Writing - Review & Editing, Re sources.

Declarations

Conflict of Interests The authors declare no conflict of interest.

References

1. Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R et al (2020) Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Inf Fusion 58:82–115
2. Bai X, Sun C, Zhou F (2009) Splitting touching cells based on concave points and ellipse fitting. Pattern Recognit 42(11):2434–2446
3. Chan T, Vese L (1999) An active contour model without edges. In: International conference on scale-space theories in computer vision, pp 141–151. Springer
4. Chang H, Yang Q, Parvin B (2007) Segmentation of heterogeneous blob objects through voting and level set formulation. Pattern Recognit Lett 28(13):1781–1787
5. Chaves D, Trujillo M, Barraza JM (2015) Concave points for separating touching particles. In: 6th International Conference on Graphic and Image Processing (ICGIP 2014), vol 9443, pp 94431W. International Society for Optics and Photonics
6. Comelli A, Bruno A, Di Vittorio ML, Ienzi F, Lagalla R, Vitabile S, Ardizzone E (2017) Automatic multi-seed detection for mr breast image segmentation. In: Image analysis and processing-ICIAP 2017: 19th International Conference, Catania, Italy, September 11-15, 2017, Proceedings, Part I 19, pp 706–717. Springer
7. Delgado-Font W, Escobedo-Nicot M, González-Hidalgo M, Herold-Garcia S, Jaume-i Capo A, Mir A (2020) Diagnosis support of sickle cell anemia by classifying red blood cell shape in peripheral blood images. Medical & Biological Engineering & Computing
8. Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica Int J Geogr Inf Geovis 10(2):112–122
9. Farhan M, Yli-Harja O, Niemistö A (2013) A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search. Pattern Recogn 46(3):741–751
10. Fernández G, Kunt M, Zrÿd J-P (1995) A new plant cell image segmentation algorithm. In: International conference on image analysis and processing, pp 229–234. Springer
11. González-Hidalgo M, Guerrero-Pena F, Herold-Garcia S, Jaume-i Capó A, Marrero-Fernández PD (2014) Red blood cell cluster separation from digital images for use in sickle cell disease. Ieee J Biomed Health Inf 19(4):1514–1525
12. Harris CG, Stephens M et al (1988) A combined corner and edge detector. In: Alvey vision conference, vol 15, pp 10–5244. Citeseer
13. He Y, Meng Y, Gong H, Chen S, Zhang B, Ding W, Luo Q, Li A (2014) An automated three-dimensional detection and segmentation method for touching cells by integrating concave points clustering and random walker algorithm. PloS one 9(8)
14. He XC, Yung NHC (2004) Curvature scale space corner detector with adaptive threshold and dynamic region of support. In: Proceedings of the 17th International conference on pattern recognition, 2004. ICPR 2004., vol 2, pp 791–794 vol. 2
15. Kumar S, Ong SH, Ranganath S, Ong TC, Chew FT (2006) A rule-based approach for robust clump splitting. Pattern Recogn 39(6):1088–1098
16. LaTorre A, Alonso-Nanclares L, Muelas S, Peña J, DeFelipe J (2013) Segmentation of neuronal nuclei based on clump splitting and a two-step binarization of images. Expert Syst Appl 40(16):6521–6530
17. Matthews BW (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure 405(2):442–451
18. Mosaliganti KR, Noche RR, Xiong F, Swinburne IA, Megason SG (2012) Acme: automated cell morphology extractor for comprehensive reconstruction of cell membranes. PLoS Comput Biol 8(12):e1002780
19. Mosley L (2013) A balanced approach to the multi-class imbalance problem. PhD thesis, Iowa State University
20. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Systems, Man, Cybern 9(1):62–66
21. Pavlidis T (1980) Algorithms for shape analysis of contours and waveforms. IEEE Trans Pattern Anal Mach Intell 1(4):301–312
22. Rodríguez R, Alarcón TE, Pacheco O (2005) A new strategy to obtain robust markers for blood vessels segmentation by using the watersheds method. Comput Biol Med 35(8):665–686
23. Samma ASB, Talib AZ, Salam RA (2010) Combining boundary and skeleton information for convex and concave points detection. In: 2010 7th International conference on computer graphics, imaging and visualization, pp 113–117. IEEE

24. Song H, Wang W (2009) A new separation algorithm for overlapping blood cells using shape analysis. Int J Pattern Recognit Artif Intell 23(04):847–864
25. Wählby C, Sintorn I-M, Erlandsson F, Borgefors G, Bengtsson E (2004) Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. J Microsc 215(1):67–76
26. Wang H, Zhang H, Ray N (2012) Clump splitting via bottleneck detection and shape classification. Pattern Recogn 45(7):2780–2787
27. Wen Q, Chang H, Parvin B (2009) A delaunay triangulation approach for segmenting clumps of nuclei. In: 2009 IEEE International symposium on biomedical imaging: From Nano to Macro, pp 9–12. IEEE
28. Yan L, Park C-W, Lee S-R, Lee C-Y (2011) New separation algorithm for touching grain kernels based on contour segments and ellipse fitting. J Zhejiang Univ Sci C 12(1):54–61
29. Yeo T, Jin X, Ong S, Sinniah R et al (1994) Clump splitting through concavity analysis. Pattern Recogn Lett 15(10):1013–1018
30. Zafari S, Eerola T, Sampo J, Kälviäinen H, Haario H (2015) Segmentation of partially overlapping nanoparticles using concave points. In: International symposium on visual computing, pp 187–197. Springer
31. Zafari S, Eerola T, Sampo J, Kälviäinen H, Haario H (2017) Comparison of concave point detection methods for overlapping convex objects segmentation. In: Scandinavian conference on image analysis, pp 245–256. Springer
32. Zafari S, Murashkina M, Eerola T, Sampo J, Kälviäinen H, Haario H (2020) Resolving overlapping convex objects in silhouette images by concavity analysis and gaussian process. J Vis Commun Image Rep 73:102962
33. Zhang W-H, Jiang X, Liu Y-M (2012) A method for recognizing overlapping elliptical bubbles in bubble image. Pattern Recogn Lett 33(12):1543–1548
34. Zhang W, Li H (2017) Automated segmentation of overlapped nuclei using concave point detection and segment grouping. Pattern Recogn 71:349–360
35. Zhang Q, Wang J, Liu Z, Zhang D (2020) A structure-aware splitting framework for separating cell clumps in biomedical images. Signal Process 168:107331