

Arrow detection in biomedical images using sequential classifier

K. C. Santosh¹  · Partha Pratim Roy²

Received: 27 May 2016 / Accepted: 1 December 2016 / Published online: 3 January 2017
© Springer-Verlag Berlin Heidelberg 2016

Abstract Biomedical images are often complex, and contain several regions that are annotated using arrows. Annotated arrow detection is a critical precursor to region-of-interest (ROI) labeling, which is useful in content-based image retrieval (CBIR). In this paper, we propose a sequential classifier comprising of bidirectional long short-term memory (BLSTM) classifier followed by convexity defect-based arrowhead detection. Different image layers are first segmented via fuzzy binarization. Candidate regions are then checked whether they are arrows by using BLSTM classifier, where Npen++ features are used. In case of low confidence score (i.e., BLSTM classifier score), we take convexity defect-based arrowhead detection technique into account. Our test results on biomedical images from imageCLEF 2010 collection outperforms the existing state-of-the-art arrow detection techniques, by approximately more than 3% in precision, 12% in recall, and therefore 8% in F_1 score.

Keywords Arrow detection · Document images · Biomedical publications · Image region labeling · Content-based image retrieval

1 Introduction

Biomedical images play a crucial role in educational and medical research purposes. In addition, they are valuable in establishing clinical decision support system (CDSS) benefiting from content-based image retrieval (CBIR). To make an efficient CBIR system, one needs to label regions-of-interest (ROIs). Since biomedical images comprise of several different regions, detecting an annotated arrow could help segment ROIs. ROIs can be used in indexing images or in analyzing the content [1, 2]. In Fig. 1, we provide a complete scenario of the project where the importance of the arrow is highlighted. Biomedical images are often annotated with pointers such as arrow and asterisk to highlight ROIs (see Fig. 2) and this way, pointers minimize the distractions from other image regions. In addition, ROIs are often referred to article text and figure captions. This paper improves on prior work in arrow detection toward meeting this goal in image content analysis. Detecting arrows is not straightforward. Arrows (in Fig. 2 appear with either high or low intensity to enhance their visibility in the image. This means that their intensities vary with respect to the background. In addition, in many cases arrows are blurred, overlapped or surrounded by textured areas. Arrow types can be just a triangle (i.e. a regular arrowhead) or with straight and curved tail.

1.1 Related work

Few techniques are reported in literature for detection of arrows overlaid in biomedical images. These techniques depend upon segmenting text like and symbol like objects, sparse pixel vectorization and local or global thresholding.





In [3], Dori et al, proposed a technique to detect arrows based on previously reported work on sparse pixel




✉ K. C. Santosh
santosh.kc@usd.edu

Partha Pratim Roy
proy.fcs@iitr.ac.in

¹ Department of Computer Science, University of South Dakota, 414 E Clark St, Vermillion, SD 57069, USA

² Indian Institute of Technology Roorkee, Department of Computer Science, Roorkee, India

OPEN  traumatic brain injury  view as   API ?

Selected Limits: Clear All  Image Type: X-ray  Exclude Graphics 

limits: Rank By Article Type Image Type Subsets Collections License Type Specialties Search In Query By Image Recent Searches

Acquired heterotopic ossification in hips and knees following encephalitis: case report and literature review.
Zhang X, Jie S, Liu T, Zhang X - *BMC Surg* (2014)

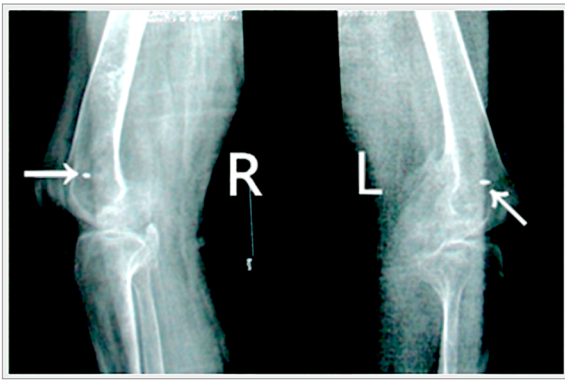
Bottom Line: He developed severe pain and significantly impaired range of motion of bilateral hips and bilateral knees. Heterotopic ossification in the bilateral hip joints and bilateral knee joints associated with encephalitis have never been reported previously. Different patient should be managed with different appropriated protocol based on the risk of individual patient and the institutional experience.

View Article: [PubMed Central](#) - [HTML](#) - [PubMed](#)

Affiliation: Department of Orthopedics, the Second Xiangya Hospital, Central South University, 139 Renmin Road, Changsha, Hunan 410011, P.R, China. liutang1981@126.com.

ABSTRACT

Background: Heterotopic ossification (HO) is a rare and potentially detrimental complication of soft-tissue trauma, amputations, central nervous system injury (traumatic brain injuries, spinal cord lesions, tumors, encephalitis), vasculopathies, arthroplasties and burn injury, characterized by lamellar bone growth in non-osseous tissues such as the muscle and the joint capsule. Heterotopic ossification associated with encephalitis is rare and the occurrence of excessive, symptomatic heterotopic ossification around bilateral hips and bilateral knees is rarely described in the literature.

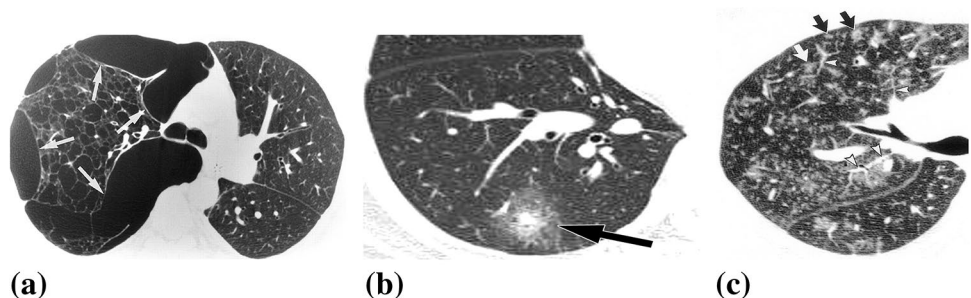


© Copyright Policy - open-access License 1 - License 2 Related In: [Results](#) - [Collection](#) [Show All Figures](#)

Figure 5: Postoperative radiographs of lateral radiographs of knees shows no loosening of rivets and no recurrence 13 months after the excision of the ossific mass.

Fig. 1 Using US National Library of Medicine's (NLM's) Open-*i*SM image retrieval search engine (<https://openi.nlm.nih.gov>), the illustration highlights the importance of using *arrow* in biomedical images (i.e., its location pointing ROI and relationship between the texts and ROI)

Fig. 2 Examples showing different types of *arrows* pointing specific image regions. These are taken from published biomedical articles



vectorization [4]. The concept relies on the cross sectional runs (or width runs) of black image regions (assuming arrow in black). The technique utilizes an interesting application but, is never applied on biomedical images, as it is limited to machine printed images such as electrical wiring diagrams, drawings and graphical symbols. Other techniques used features such as eccentricity, convex area and solidity [5]. These features can define regular arrows (i.e., straight arrows showing left, right, top and bottom). Since overlaid arrows in biomedical images can be distorted, computing straightforward geometrical features cannot differentiate arrows from other regions. Cheng et al used text-like and arrow-like objects separation, assuming that arrows are shown in either black or white color with respect to the background [6]. As mentioned earlier (see last paragraph of Sect. 1), arrows are not appeared in just either black or white pixels, their work cannot fit into the target. From the

binary image, arrow-like object separation employs a fixed-sized mask (after removing the small objects and noise as in [5]), which are then used for feature computation such as major and minor axis lengths, axis ratio, area, solidity and Euler number. Removing small candidate is not a solution, since overlaid arrows can also be just a triangle (that can be small too). Further, arrows can have texture similarity with the regions they are connected/pointed to. This will produce distorted arrow candidates at the time of segmentation. A recent study uses a pointer region and boundary detection to handle distorted arrows [7], which is followed by edge detection techniques and fixed thresholds as reported in [8, 9]. These candidates are used to compute overlapping regions, which are then binarized to extract the boundary of the expected pointers. Fundamentally, edge-based arrow detection techniques are limited by the weak-edge problem [5–7]. Weak-edge happens in case intensities

vary a lot on a single arrow candidate, and as a consequence, part of the edges will be missed. In addition, the techniques rely on hard thresholding has to be empirically designed from one dataset to another. For edge detection in binary or grayscale images, most state-of-the-art methods use classical algorithms like Roberts, Sobel and Canny edge detection. Template-based methods are limited, since they require new templates to train new images. Also, it may be necessary to re-evaluate the threshold values when new images are used. Edge-based techniques are still considered, since sampling points can be remarkably compact compared to solid regions, especially when broken boundaries are recoverable (as reported in [10]). In biomedical images, one of the major issues for a broken boundary is non-homogeneous intensity distribution, where pointers overlap with content. This is one of primary the reasons, hard thresholding (at the time of binarization, for instance) may not work.

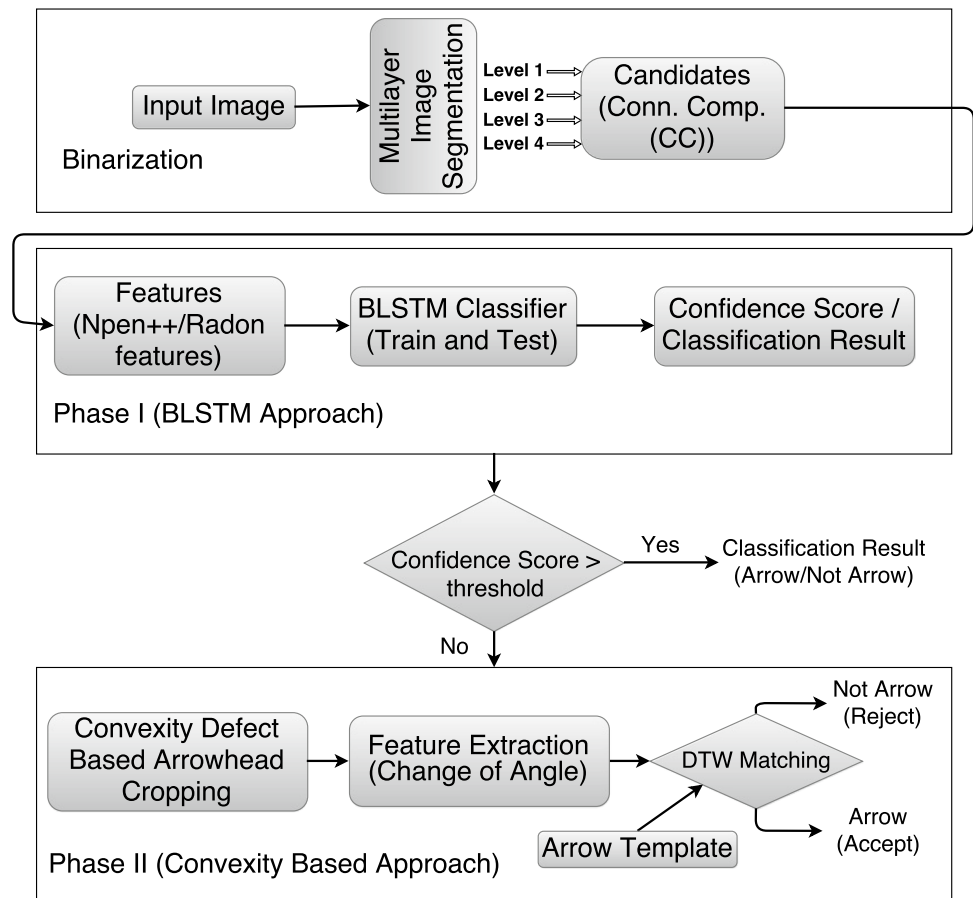
1.2 Contribution outline

Our method can be summarized as shown in Fig. 3. It relies on a grayscale fuzzy binarization process at different levels [11], where candidates are segmented based on

connected component (CC) principle. Unlike the common state-of-the-art methods, we use four different levels of fuzzy binarization. This ensures that overlaid arrow candidates are not missed (see Fig. 4). Our system comprises of two classifiers, analyzing the candidates in sequence to determine whether these arrow candidates are arrows. It consists of a neural network based bidirectional long short-term memory (BLSTM) classifier followed by convexity-defect based arrowhead detection. Npen++ and the Radon features are computed for these candidates and are validated with BLSTM-trained arrow model. BLSTM classifies each candidate as arrow (and non-arrow) candidates with some cross-entropy error (CER) score. This CER defines how confident BLSTM classification is, which is inversely proportional to the the confidence score. If the confidence score of the BLSTM classifier crossed the threshold, the candidate is classified as an arrow. Otherwise, the candidate is passed through convexity defect-based technique. The latter step prunes artefacts (i.e., unwanted noisy object and/or image regions) and stores arrowhead-like candidates, since it deals with just the arrowhead.

The remainder of the paper is organized as follows. In Sect. 2, we explain the binarization technique. We explain BLSTM arrow detection in Sect. 3 that includes feature

Fig. 3 Overall system workflow in block format. Block-wise explanation can be found in Sect. 1.2



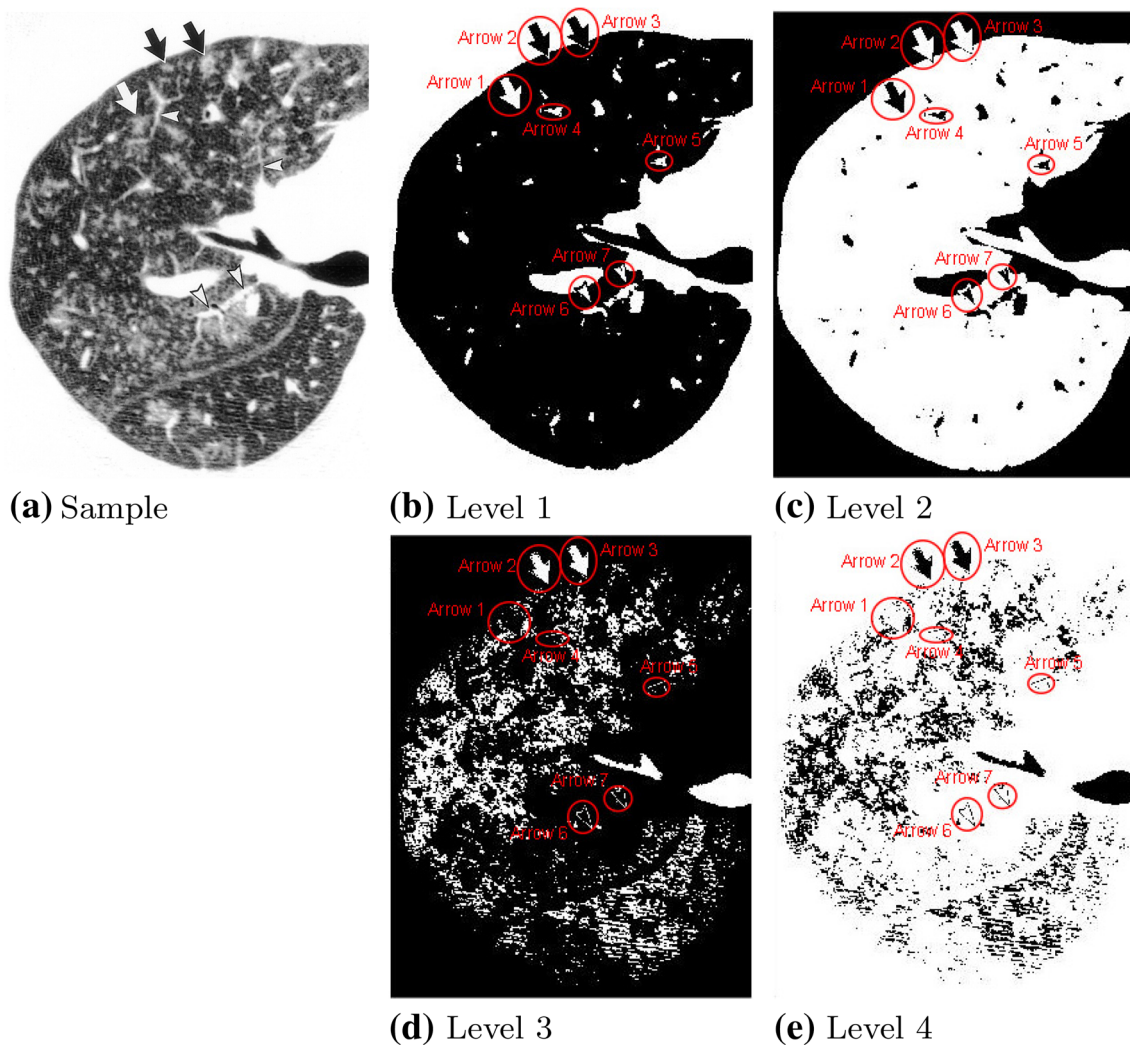


Fig. 4 Fuzzy binarization (of Fig. 2c): four different levels (levels 1–4), where the *arrows* are encircled both in *red* and *black* with respect to the *background color*

extraction and classification. In Sect. 4, we discuss convexity defect-based techniques for arrowhead detection. Experimental setup and results are reported in Sect. 5. We also extend our evaluation by taking a comprehensive and comparative study with state-of-the-art techniques in Sect. 6. Section 7 concludes the paper.

2 Multilayer image segmentation

In biomedical images (see Fig. 2), arrows appear with either high or low intensity to enhance their visibility in the image. In addition, in many cases arrows are blurred, overlapped or surrounded by textured areas. In such contexts, typical binarization tools that are based on fixed threshold values are unable to segment candidate regions. Therefore, we focus on an adaptive binarization tool, which is based

on a fuzzy partition of a 2D histogram of the image, taking into account the gray level intensities and local variations [12]. 2D Z-function criteria based on the optimization of fuzzy entropy are then computed from this histogram to automatically set the threshold. Z-function employs two kernels: low-level and high-level cuts. In addition, we take their inversions, and altogether, four different binarized levels are processed, as illustrated in Fig. 4. In Fig. 4, arrow candidates are encircled in both red and black (with respect to the background color). The main idea of using four different levels of binarization is not to miss the overlaid arrows. Furthermore, deformed and/or distorted arrows can be discarded since the arrows are repeated in other levels of binarization. Note that image regions are segmented based on the 8×8 connected component (CC) principle. In general, CCs, in 2D image, are clusters of pixels with the same value, which are connected to each other through 8-pixel

connectivity. CCs are referred to as candidate regions. From a pool of several candidates, we are required to select arrow-like candidates.

3 Arrow detection using BLSTM classifier

To train and test a neural network based BLSTM classifier, we compute features: (1) Npen++ and the Radon transform. Using both features, we classify arrow head candidates based on the confidence score of the BLSTM.

$$\cos\alpha(i) = \frac{\Delta x(i)}{\Delta s(i)}, \quad \text{and} \quad \sin\alpha(i) = \frac{\Delta y(i)}{\Delta s(i)} \tag{1}$$

where Δs , Δx , and Δy are defined, respectively, as follows:

$$\begin{aligned} \Delta s(i) &= \sqrt{\Delta x^2 + \Delta y^2}, \\ \Delta x(i) &= x(i - 1) - x(i + 1), \quad \text{and} \\ \Delta y(i) &= y(i - 1) - y(i + 1). \end{aligned}$$

Curvature The computation of curvature at a point $(x(i), y(i))$ can be considered as any consecutive points along the trajectory (or writing direction), and is described as follows:

$$\begin{aligned} \cos \beta(i) &= \cos \alpha(i - 1) * \cos \alpha(i + 1) + \sin \alpha(i - 1) * \sin \alpha(i + 1) \quad \text{and} \\ \sin \beta(i) &= \cos \alpha(i - 1) * \sin \alpha(i + 1) + \sin \alpha(i - 1) * \cos \alpha(i + 1). \end{aligned} \tag{2}$$

3.1 Features

The performance of any neural network classifier depends on the features that are used to represent the candidates. In our study, we have tested the performance with two different feature descriptors: Npen++ [13] and the Radon feature [14]. Npen++ features are expected to be worked for well defined geometric patterns (such as arrows), including curvature, curliness and orientation. Similarly, the Radon feature is well suited for the patterns since projection in the Radon space changes 2D arrow image into 1D signal that can be considered as strokes. BLSTM classifier can be considered as a well-known for strokes and gesture recognition [15, 16].

3.1.1 Npen++ feature descriptor

Npen++ features were originally introduced for handwriting recognition. It actually comprises a number of features computed along the handwriting trajectory. The normalized sequence of the captured coordinates $(x(i), y(i))$ forms the input to the system. It computes a sequence of features along this trajectory. But, not all of these features are relevant for our arrow detection approach. Most of the features of Npen++ depend on the baseline. In original Npen++ recognizer, the baseline $b(x(i))$ corresponds to the original writing line on which a word or text was written. In our study, we consider the lowest line parallel to x-axis passing through the contour of arrow as baseline.

Vertical position The vertical distance between $y(i)$ and $b(x(i))$ of a point $(x(i), y(i))$ is the vertical position of the point, where $b(x(i))$ is the y-value of the baseline for i th point on the contour.

Orientation At point $(x(i), y(i))$, the local writing direction is described as

Note that this sequence does not represent curvature but, angular difference.

Aspect The aspect $A(i)$ of the contour in the vicinity of a point characterizes the height-to-width ratio of the bounding box containing the preceding and succeeding points of $(x(i), y(i))$. It is computed as:

$$A(i) = \frac{\Delta y(i) - \Delta x(i)}{\Delta y(i) + \Delta x(i)} \tag{3}$$

Curliness Curliness $C(i)$ feature describe the deviation from a straight line in the vicinity of $(x(i), y(i))$. It is computed as the ratio of the length of the contour and maximum side of the bounding box,

$$C(i) = \frac{L(i)}{\max(\Delta x, \Delta y)} - 2, \tag{4}$$

where $L(i)$ denotes the length of the contour in the vicinity of the point computed as the sum of all line segments, and Δx and Δy are width and height of the bounding box, respectively.

We have applied Npen++ feature for selection of arrow-like candidates. Figure 5 shows how do the Npen++ features look like for both arrow and non-arrow candidates, and therefore, it allows to realize its discriminative property.

3.1.2 The Radon transform

The radon transform computes projections of an image matrix along specified directions [14]. A projection of a two-dimensional function $f(x, y)$ is a set of line integrals. It is computed by calculating the length of line integrals from multiple sources along parallel paths, or beams, in a certain

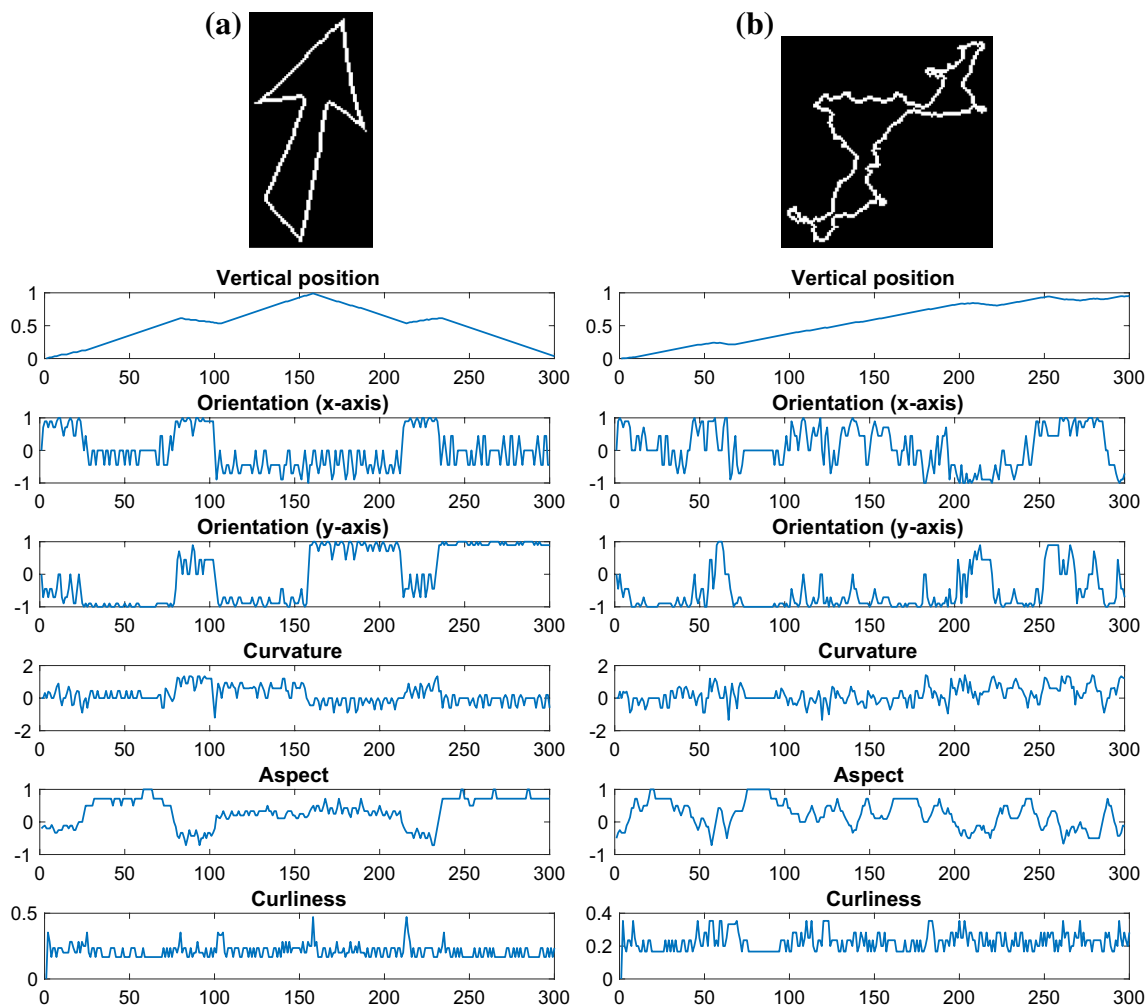


Fig. 5 Npen++ features, representing **a** arrow and **b** non-arrow candidates. We provide them side-by-side so that one can compare features find how discriminate they are

direction. In general, the Radon transform of $f(x, y)$ is the line integral of f parallel to the y -axis is

$$R_{\theta}(x') = \int_{-\infty}^{\infty} f(x' \cos\theta - y' \sin\theta, x' \sin\theta + y' \cos\theta) dy' \quad (5)$$

where $x' = x \cos\theta + y \sin\theta$ and $y' = y \cos\theta - x \sin\theta$. The beams are spaced 1 pixel unit apart. To represent an image, the Radon function takes multiple, parallel-beam projections of the image from different angles by rotating the source around the center of the image. Since arrow is a regular geometric shape, the radon transform of arrow tend to have regularities.

3.2 BLSTM classifier

Again, we are motivated by the use of recently introduced bidirectional long short-term memory (BLSTM) [17].

In simple words, the BLSTM is a recurrent neural network having connections between the nodes so as to form a directed cycle, thus, providing a ‘memory’ of network’s previous internal state.

3.2.1 Long short-term memory (LSTM) layer

A specific architecture known as memory block forms the LSTM network nodes. Each memory block contains a memory cell and it interact with rest of the network with the help of three gates, viz., an input gate, an output gate and a forgot gate [17]. The forget gate determines when the input is important enough to keep in memory and when the block can forget the values. This helps memory cells retain their state for a long time and to model the context at feature level. The input signal is processed in both directions: forward and backward by two different

layers, thus improving ID sequence recognition. Next layer combines the output of both the layer to form the feature map. Like convolutional neural network, multiple forward and backward layer in each LSTM layer, and multiple feature maps at the output layer are possible. Further, it is possible to stack multiple LSTM layers using method like max-pooling subsampling.

3.2.2 Candidate selection

To train the BLSTM model, the aforementioned features (see Sects. 3.1.1 and 3.1.2) can be either separately applied or combined.

Individual feature performance in BLSTM

The features: Npen++ and the Radon, are separately used to train BLSTM models for the arrows (and not arrows). For testing, a test candidate is passed tested through the trained BLSTM models. As a reminder, LSTM layer has been discussed in Sect. 3.2.1.

Integrating features in BLSTM

This could be another option to realize how good will be the BLSTM after feature integration. To handle this, we propose two different setups: (1) weighted average score, and (2) confidence scores (in parallel).

Setup 1: Weighted average score Since we do not know which feature is performed well, in this setup, we start with providing two different weights: α and $1 - \alpha$. This can be formalized as follows:

$$x_{avg} = \alpha \times x_{Npen++} + (1 - \alpha) \times x_{Radon}, \tag{6}$$

where the values of α ranges from 0 to 1, and x_F represents confidence score from BLSTM for that particular feature, F . In general, we have

$$x_{avg} = \begin{cases} x_{Radon}, & \text{if } \alpha = 0 \\ x_{Npen++}, & \text{if } \alpha = 1 \\ \alpha \times x_{Npen++} + (1 - \alpha) \times x_{Radon}, & \text{otherwise.} \end{cases}$$

Therefore, we do not allow biasing any particular feature. The average score, x_{avg} is then used to classifying arrow-candidates

Setup 2: Confidence score in parallel fashion As in *setup 1*, the candidate image is passed through both Npen++ trained classifier and the Radon trained classifier

in parallel. At the result, we made the BLSTM classification decision, from which we received high confidence score. It can be generalized as,

$$x_{decision} = \begin{cases} x_{Radon}, & \text{if } x_{Radon} \geq x_{Npen++} \\ x_{Npen++}, & \text{otherwise,} \end{cases} \tag{7}$$

where x_F represents confidence score from BLSTM for that particular feature, F .

Like the state-of-the-art works, the candidate classified by BLSTM with high confidence score is accepted. If not, remaining candidates are passed though second phase of screening (i.e., convexity defect-based arrowhead detection, and also refer to Fig. 3). The latter phase (Sect. 4) aims to eliminate the false positives that are coming from BLSTM.

4 Arrowhead detection using convexity defect-based algorithm

Unlike earlier section and previously reported works [11, 18], in this paper, we do not take arrow tail into account because arrow tail structures vary a lot. As a consequence, the geometric signature computed from extreme points of a triangle (i.e., triplet). Such a change can affect the overall appearance of the arrow (see Fig. 6). This limits the performance of the previous technique, both in computational complexity and in detection rate. In this section, we limit our work and detect an arrowhead that includes following steps: (1) convexity defect-based arrowhead candidate cropping; and (2) arrowhead candidate matching via dynamic time warping (DTW).

The convexity defect-based technique is based on the characteristics of the arrowhead that can be represented by a triangle [19, 20]. Once candidate arrowheads are selected, we confirm them by matching with the templates via DTW.

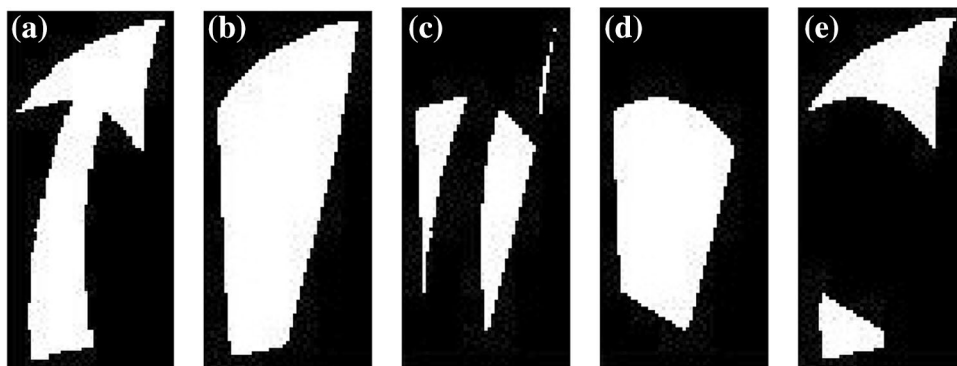
4.1 Convexity defect-based arrowhead candidate cropping

We apply hull convexity defect concept to select arrow-like candidates (see Fig. 7). If a set of points along the contour of the binary CC contains the line segments connecting each pair of its points, it is said to be convex. In a convex combination, each point $x_i \in S$ is assigned a

Fig. 6 Examples showing the changes in tail structure. Further, an absence of the tail is also possible



Fig. 7 Arrowhead candidate cropping: **a** an arrow, **b** convex hull, **c** convexity defect, **d** a complete convexity defect region, and **e** arrowhead candidates



weight or coefficient w_i in such a way that the coefficients sum to one. These weights are, at the same time, used to compute a weighted average of the points. In general, the convex hull, expressed in a single formula, is the set: $\{\sum_{i=1}^{|S|} w_i x_i | (\forall i: w_i \geq 0) \wedge \sum_{i=1}^{|S|} w_i = 1\}$. Thus, the convex hull of a finite point set $S \in \mathbb{R}^2$. An example is shown in fig. 7b. A convex shaped silhouettes on both sides of the arrow can be computed by subtracting an original candidate from the convex hull (see Fig. 7c). This removes tail, and the convexity defect region is shown in Fig. 7d, which is just a convex hull of both convex shaped silhouettes. At the end, arrowhead candidate(s) is(are) selected by subtracting an original image with the convexity defect region shown in Fig. 7e. All connected components (after subtraction) are taken as the potential arrowhead candidates.

4.2 Arrowhead candidate selection

We apply a template matching technique to confirm arrowhead candidates (see Fig. 7). We extract a feature along the contour and match with the predefined templates using dynamic time warping (DTW) technique. The arrowhead candidate is confirmed when the similarity score crosses the empirically designed threshold.

We have a set of coordinate points along the contour, $P = \{p_i\}_{i=1, \dots, n}$. We compute change in angle with respect to x-axis from any consecutive pair: p_i and p_{i+1} , $\alpha_i = \arctan\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$, and therefore, we can represent a whole sequence as a feature vector, $f = \{\alpha_i\}_{i=1, \dots, n}$. We then represent a digital curve using fewer points through polygonal approximation [21–23], such that the properties of the curvature of the digital curve are retained. Continuous redundancy of α_i can be possible in our feature vector, $\alpha_i = \alpha_{i+j}, j = 1, \dots, m$, where $m \leq n$. In our implementation, we compute the difference between the angles and check whether it crosses the threshold: α_i if $|\alpha_i - \alpha_{i+1}| \leq \epsilon$, where ϵ is user-defined. Figure 8 shows three examples, where the changes in angles are shown at all dominant points.

For matching, we use DTW algorithm, since it allows to find the dissimilarity between two non-linear sequences potentially having different lengths [24, 25]. In Fig. 8, one can notice the variations in feature vector size from one arrowhead to another. Consider two feature sequences: $f_1 = \{\alpha_i\}_{i=1, \dots, n}$ and $f_2 = \{\beta_j\}_{j=1, \dots, m}$ of size n and m , respectively. The aim of the algorithm is to provide the optimal alignment between both sequences. At first, a matrix of size $n \times m$ is constructed. Then for each element, local distance metric $\delta(i, j)$ between the events e_i and e_j is computed i.e., $\delta(i, j) = (e_i - e_j)^2$. Let $D(i, j)$ be the global distance up to (i, j) ,

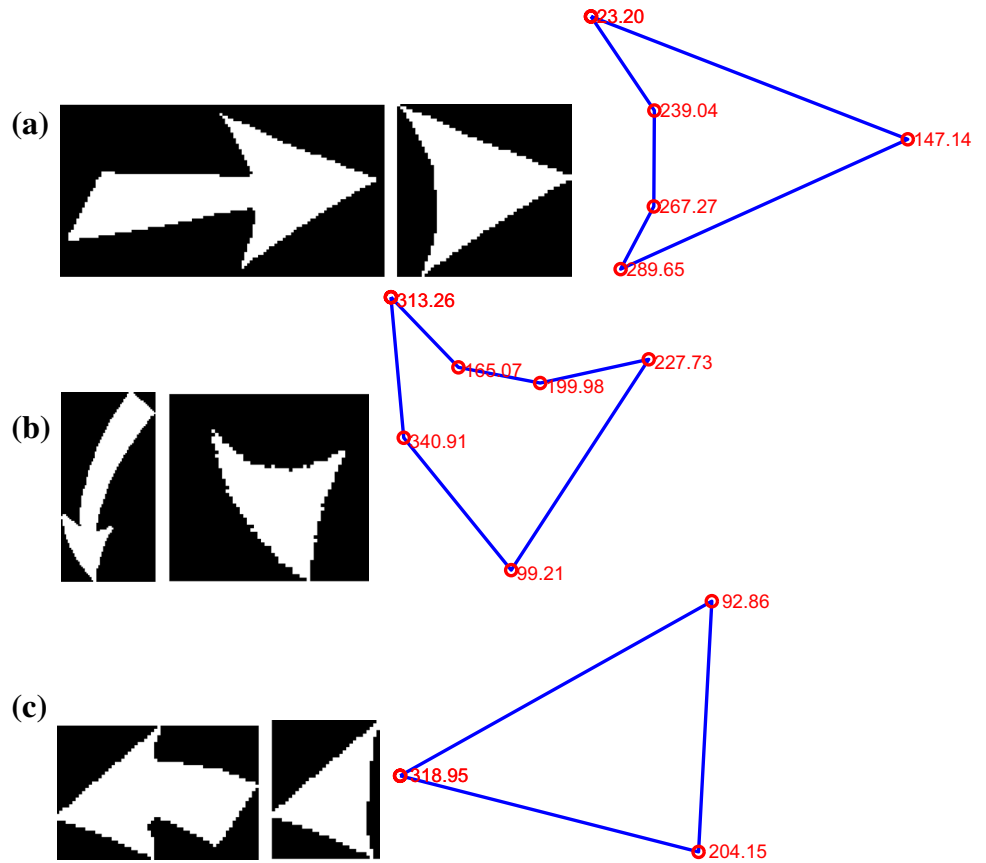
$$D(i, j) = \min \begin{bmatrix} D(i-1, j-1), \\ D(i-1, j), \\ D(i, j-1) \end{bmatrix} + \delta(i, j)$$

with an initial condition $D(1, 1) = \delta(1, 1)$ such that it allows warping path going diagonally from starting node $(1, 1)$ to end (n, m) . The main aim is to find the path for which the least cost is associated. The warping path therefore provides the difference cost between the compared signatures. Formally, the warping path is, $W = \{w_k\}_{k=1, \dots, l}$, where $\max(i, j) \leq l < i + j - 1$ and k th element of W is $w(i, j)_k \in [1:n] \times [1:m]$ for $k \in [1:l]$. The optimised warping path W satisfies the following three conditions: boundary condition, monotonicity condition and continuity condition. We then define the global distance between f_1 and f_2 as,

$$\Delta(f_1, f_2) = \frac{D(n, m)}{l}. \tag{8}$$

The last element of the $n \times m$ matrix gives the DTW-distance between f_1 and f_2 , which is normalised by l i.e., the number of discrete warping steps along the diagonal DTW-matrix.

Fig. 8 Examples showing a complete process (from *left to right*) starting from an original candidate (resulting from fuzzy binarization—see Fig. 4), *arrowhead* cropping (see Fig. 7) to feature extraction after *polygonal* approximation



5 Experiments

5.1 Datasets, ground-truth and evaluation protocol

The well-known imageCLEF dataset [26] is used for testing. It is composed of 298 chest CT images. Each image is expected to have at least one arrow, and there are 1049 arrows, in total. For all images in the dataset, groundtruths of the arrows were created and each ground-truth includes information like arrow type, color, location, and direction.

For validation, for any given image in the dataset, our performance evaluation criteria are precision, recall and F_1 score,

$$\text{precision} = \frac{m_1}{M}, \quad \text{recall} = \frac{m_1}{N} \quad \text{and} \quad (9)$$

$$F_1 \text{ score} = 2 \left(\frac{(m_1/M) \times (m_1/N)}{(m_1/M) + (m_1/N)} \right),$$

where m_1 is the number of correct matches from the detected set M and N is the total number of arrows (in the ground-truth) that are expected to be detected.

Table 1 Performance of the proposed system (in %)

Classifier		Precision	Recall	F_1 score
BLSTM classifier (1)	Npen++	13.96	98.43	24.45
	the Radon	06.14	97.31	11.55
Convexity defect-based algorithm (2)		88.50	93.80	90.09
Sequential classifier: (1) + (2)	Npen++	95.39	99.21	97.26
	the Radon	92.69	99.10	95.79

Bold-face numbers indicate the best score

5.2 Our results and analysis

5.2.1 Results

In Table 1, the performance evaluation in terms of precision, recall and F_1 score, can be taken as follows.

1. BLSTM classifier with two different features that are separately applied;
2. Convexity defect-based algorithm (without BLSTM classifier); and
3. Sequential classifier, by integrating both: BLSTM and convexity defect-based technique (see Fig. 3).

Based on the aforementioned schema of the results, we provide results in Table 1. In Table 1, we observe the following:

- BLSTM alone has a very good recall value but, with large large false positives. It holds true (i.e., high recall) for both features: Npen++ (98.43%) and the Radon (97.31%) that are applied separately.
- Convexity defect-based algorithm performs better than BLSTM in terms of precision.

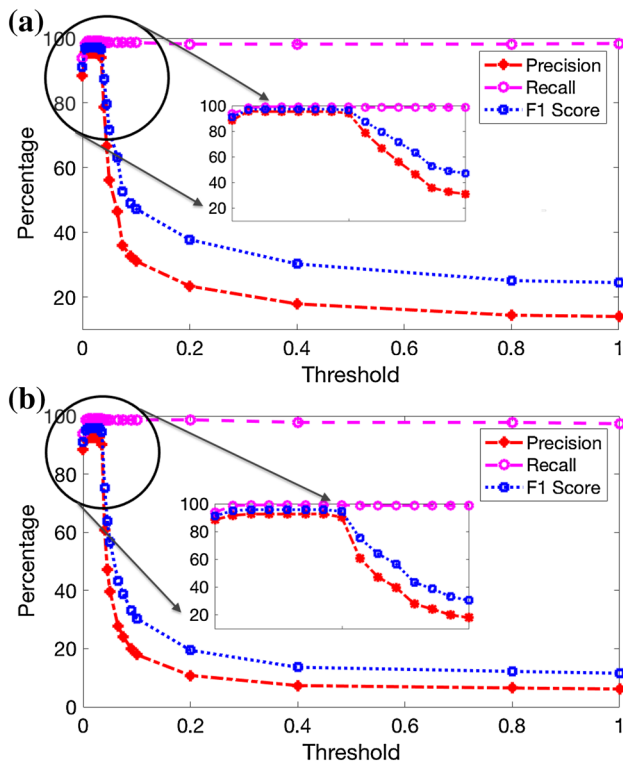


Fig. 9 Graph showing the change in accuracies with change in the threshold value of cross entropy error for **a** Npen++ feature and **b** Radon feature

- *Integrating both (sequential classifier)* BLSTM and convexity defect-based technique provides interesting results. The candidates having low confidence scores (at the BLSTM end) are now correctly be detected/classified at the latter phase, thanks to convexity defect-based arrowhead cropping (see Sect. 4). On the whole, the sequential classifier that combines BLSTM and convexity defect-based arrowhead cropping provides better results. In our results (Table 1), we provide results for separate features. For example, from the sequential classifier, we received F_1 score of 97.26% when BLSTM (with Npen++) is combined with convexity defect-based algorithm, which is better than when the Radon features are used in BLSTM (refer to the last two rows of Table 1).

Note that our BLSTM classifier uses a threshold on the cross entropy error value to decide whether we should further test the candidate with convexity defect-based algorithm. If not, we accept as it is classified by BLSTM classifier. In Fig. 9, we detail an idea of how BLSTM performances have been changed in accordance with the change in this threshold values. We note that best results (and almost equal) are provided when the normalized threshold values are in the range: [0.01–0.03], in both features. With this validation, we use the threshold value of 0.025 for all tests (see Table 1). Besides, in Fig. 9, we observe the following:

- for small threshold values, all arrow candidates will be above the threshold and will be passed through convexity defect-based algorithm (i.e., only BLSTM is active); and
- for large threshold values, all arrow candidates will be below the threshold and will not be passed through convexity defect-based algorithm (only convexity defect-based algorithm is active).

Table 2 Performance (in %): integrating features in BLSTM (setup 2) and with convexity defect-based algorithm

Classifier	Precision	Recall	F_1 score
BLSTM classifier (1) (setup 2: integrating features)	26.52	99.88	41.86
Convexity defect-based algorithm (2)	88.50	93.80	90.09
Sequential classifier: (1) + (2)	96.75	99.88	98.29

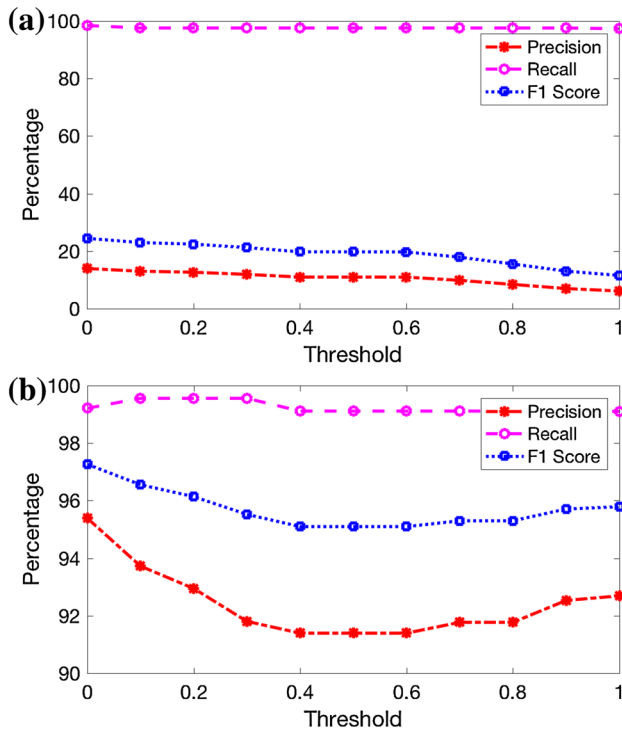


Fig. 10 Graph showing the changes in performance with different values of threshold, α : **a** without and **b** with convexity defect-based algorithm

5.2.2 Integrating features in BLSTM

As mentioned in Sect. 3.2.2, we used two different setups: 1) weighted average score, and 2) confidence scores (in parallel).

Using the Eq. 6, Fig 10 shows the changes in performance with change in the value of threshold, $\alpha \in [0, 1]$. In Fig 10, we also provide the performance of sequential classifier (i.e., integrating features in BLSTM plus convexity defect-based algorithm). In our test, performance is found to be maximum for Npen++ feature alone, and tends to linearly decrease first, which is followed by slight advancement with increase in the weight of the Radon's score. On the whole, we observe that integrating the Radon feature trained classifier with Npen++ feature trained classifier does not add to the accuracies (i.e., no change in performance).

In the latter setup, the candidate image is passed through both Npen++ trained classifier and the Radon trained classifier in parallel. At the result (as described in Eq. 7), we made the BLSTM classification decision, from which we received high confidence score. This setup boosted our results by more than 1%. In Table 2, precision, recall and F_1 score are shown. We remind

that we take feature integration account in BLSTM first, and then combine with convexity defect-based algorithm. Compared to Table 1, the performance has been increased by more than 1%, thanks to feature integration. For a comparison, we will take these best scores in Sect. 6 (Table 2).

5.3 Processing times

Processing times, on average, for both Npen++ and the Radon features are almost identical. For each candidate image, Npen++ feature took approximately 12.7 and 14.4 ms, respectively, without and with convexity defect-based approach. On the other hand, with the Radon feature, each candidate image took 15.5 and 16.2 ms, respectively, without and with convexity defect-based approach. We have used Unix Environment (Ubuntu 16.04) with 8 GB RAM and Intel Core i7 processing power PC with Matlab R2015a.

5.4 Example outputs

In Fig. 11, we provide some example outputs illustrating arrow detection. In these outputs, our aim is to show the importance of using multilayer image segmentation concept (see Sect. 2). In both examples of Fig. 11, arrows are detected from using two different layers, since they are appeared in both black and white pixels. This means that straight forward image binarization does not work (for example, the works reported in [6–8]). For a comparison, we refer to Sect. 6. For better understanding, we provide more outputs in Fig. 12.

6 State-of-the-art comparison

Further, the comparative study with state-of-the-art methods has been made. In this comparison, our benchmarking methods are categorized into two groups:

1. State-of-the-art methods that are specially designed for arrow detection; and
2. Common template-based method by using well-known state-of-the-art shape descriptors.

6.1 Arrow detection methods

Four well-known methods from the state-of-the-art that are specially designed for arrow detection are used:

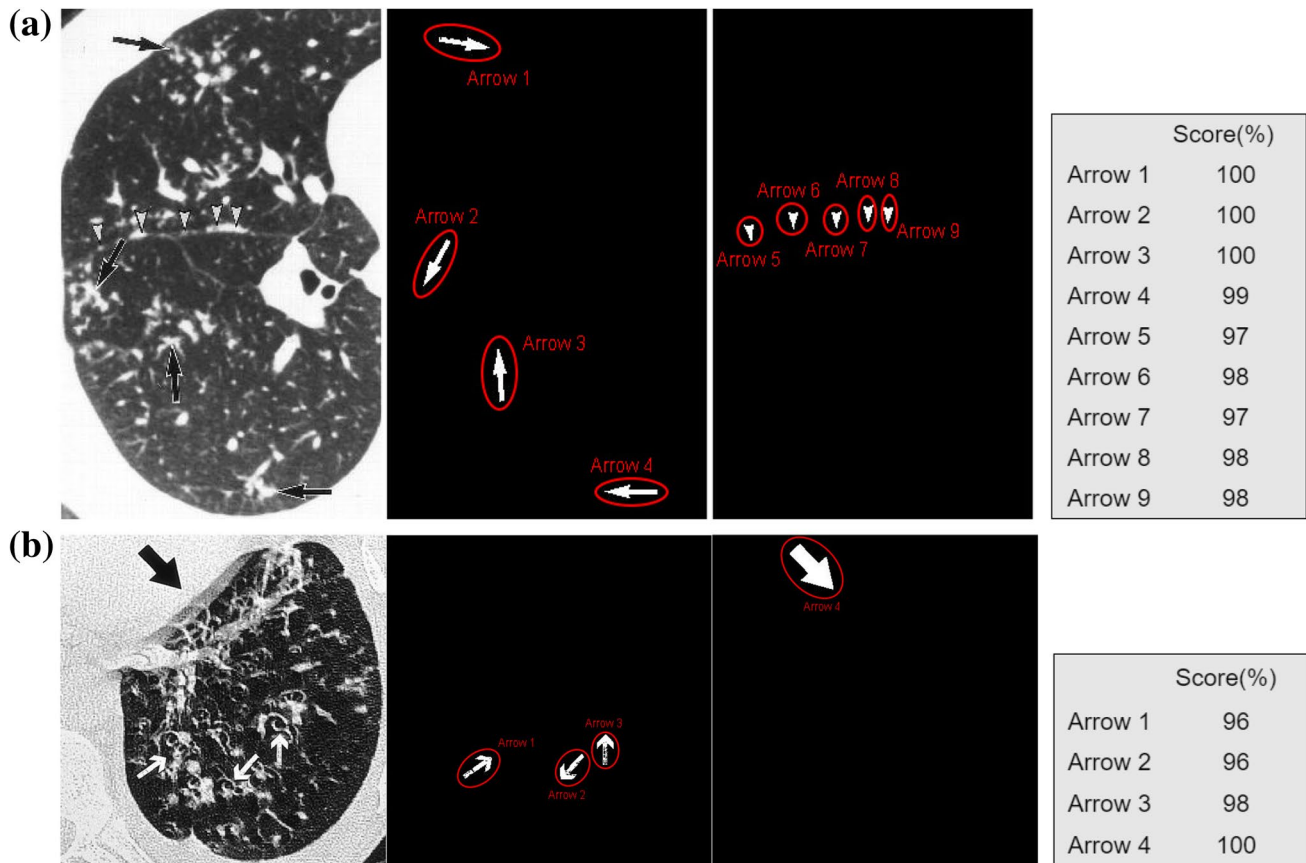


Fig. 11 Examples showing different binarization levels are used to detect *arrows*. This demonstrates the idea of image inversion used in binarization since *arrows* are not just *black-filled*

1. Global thresholding-based method (M1) [6];
2. Two edge-based methods (M2:M3) [7, 8]; and
3. A template-free geometric signature-based method (M4) [11].

The results are provided in Table 3, where method 4 (M4) performs the best with precision, recall and F_1 score of 93.14, 86.92 and 89.94%, respectively.

6.2 Template-based methods

In case of template-based method, we created 11 templates (arrows) having different shapes (including sizes). The template size can further be extended in accordance with the dataset. To extract shape features, we took the most frequently used shape descriptors (in computer vision) from the state-of-the-art. They are

1. Generic Fourier descriptor (GFD) [27],
2. Shape context (SC) [28],

3. Zernike moment (ZM) [29],
4. Generic Radon transform (G-RT) [30] and
5. DTW-Radon [31].

As before, results (precision, recall and F_1 score) are provided in Table 4. Among all shape descriptors, GFD provides the best performance.

6.3 Comparison summary

From all reported methods (see Tables 2, 3 and 4), we take best results for a comparison with the proposed method. A complete comparison study is provided in Table 5. Considering the dataset, the proposed method outperforms the best state-of-the-start arrow detection method by more than 8% F_1 score, and the template-based (shape descriptor) method by more than 20% F_1 score.

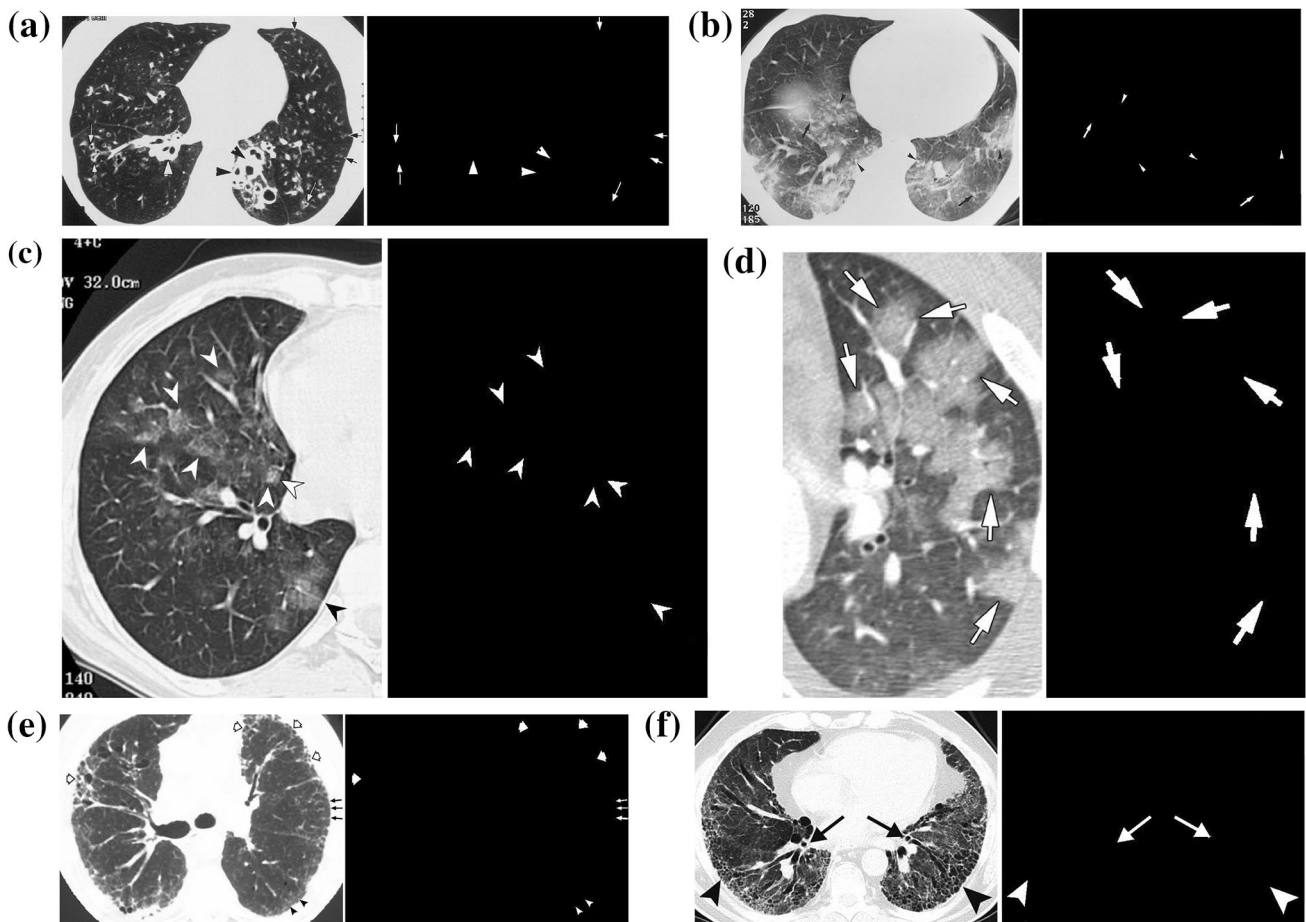


Fig. 12 Examples illustrating *arrow* detection, on the *right* of the input image

Table 3 Performance comparison (in %) of previously reported methods

Methods	Precision	Recall	F ₁ score
M1: Cheng et al. [6]	81.10	74.10	77.00
M2: You et al. [8]	22.80	77.80	35.00
M3: You et al. [7]	84.20	81.60	83.00
M4: Santosh et al. [11, 18]	93.14	86.92	89.94

Bold-face numbers indicate the best score

Table 4 Performance comparison (in %) of template based method

Methods	Precision	Recall	F ₁ score
Generic Fourier descriptor (GFD) [27]	75.10	78.33	76.68
Shape context (SC) [28]	68.30	71.40	69.82
Zernike moments (ZM) [29]	55.20	57.70	56.40
Generic Radon Transform (G-RT) [30]	59.50	63.60	61.48
DTW-Radon [31]	62.10	65.30	63.65

Bold-face numbers indicate the best score

Table 5 Performance comparison (in %) of our method with the best previously reported works

Methods	Precision	Recall	F ₁ score
Generic Fourier descriptor (GFD) [27]	75.10	78.33	76.68
Santosh et al. [11, 18]	93.14	86.92	89.94
Our method	96.75	99.88	98.29

Bold-face numbers indicate the best score

This attests the usefulness of our method for further image region labelling problem, in biomedical images.

7 Conclusion and future work

We have presented a sequential classifier to detect overlaid arrows in biomedical images, comprising of bi-directional long short-term memory (BLSTM) classifier

followed by convexity defect-based arrowhead detection. Our test results on biomedical images from imageCLEF 2010 collection outperforms the existing state-of-the-art arrow detection techniques, by approximately more than 3% in precision, 12% in recall and therefore 8% in F_1 score.

To the best of our knowledge, this is the first time a sequential classifier combining both neural network and geometrical shape of the arrow has been used. Our immediate step would be labeling image regions with the use of the arrows detected by the current work.

Acknowledgements The authors would like to acknowledge Mr. Naved Alam for his work during his stay (Research Work) at the Department of Computer Science, Indian Institute of Technology (IIT) Roorkee.

References

- Demner-Fushman D, Antani S, Simpson M, Rahman M (2010) Combining text and visual features for biomedical information retrieval and ontologies. Tech. rep, LHCBC Board of Scientific Counselors, National Institutes of Health, Bethesda
- Demner-Fushman D, Antani S, Simpson MS, Thoma GR (2012) Design and development of a multimodal biomedical information retrieval system. *J Comput Sci Eng* 6(2):168–177
- Dori D, Wenyin L (1999) Automated cad conversion with the machine drawing understanding system: concepts, algorithms, and performance. *IEEE Trans Syst Man Cybern Part A Syst Humans* 29:411–416
- Dori D, Member S, Liu W (1999) Sparse pixel vectorization: An algorithm and its performance evaluation. *IEEE Trans Pattern Anal Mach Intell* 21:202–215
- Park J, Rasheed W, Beak J (2008) Robot navigation using camera by identifying arrow signs. In: *International Conference on Grid and Pervasive Computing—Workshops*, pp 382–386
- Cheng B, Stanley RJ, De S, Antani S, Thoma GR (2011) Automatic detection of arrow annotation overlays in biomedical images. *Int J Healthc Inf Syst Inf* 6(4):23–41
- You D, Simpson MS, Antani S, Demner-Fushman D, Thoma GR (2013) A robust pointer segmentation in biomedical images toward building a visual ontology for biomedical article retrieval. In: Zanibbi R, Coiasnon B (eds) *Document recognition and retrieval*, vol 8658 of *SPIE Proceedings*. SPIE
- You D, Apostolova E, Antani S, Demner-Fushman D, Thoma GR (2009) Figure content analysis for improved biomedical article retrieval. In: Berkner K, Likforman-Sulem L (eds) *Document recognition and retrieval*, vol 7247 of *SPIE Proceedings*, SPIE, pp 1–10
- You D, Antani S, Demner-Fushman D, Rahman MM, Govindaraju V, Thoma GR (2010) Biomedical article retrieval using multimodal features and image annotations in region-based cbr. In: Likforman-Sulem L, Agam G (eds) *Document recognition and retrieval*, vol 7534 of *SPIE Proceedings*. SPIE, pp 1–10
- Hori O, Doermann DS (1995) Robust table-form structure analysis based on box-driven reasoning. *Int Conf Document Anal Recogn* 1:218–221
- Santosh KC, Wendling L, Antani S, Thoma G (2014) Scalable arrow detection in biomedical images. In: *International Conference on Pattern Recognition*. IEEE Computer Society, Stockholm, pp 3257–3262
- Cheng H, Chen Y-H (1999) Fuzzy partition of two dimensional histogram and its application to thresholding. *Pattern Recogn* 32:825–843
- Jaeger S, Manke S, Reichert J, Waibel A (2001) Online handwriting recognition: the npen++ recognizer. *Int J Document Anal Recogn* 3(3):169–180
- Deans SR (1983) *The Radon transform and some of its applications*. A Wiley-Interscience publication, Wiley, New York
- Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. *Pattern Anal Mach Intell IEEE Trans* 31(5):855–868
- Liwicki M, Graves A, Bunke H, Schmidhuber J (2007) A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In: *Proc. 9th Int. Conf. on Document Analysis and Recognition*, vol 1, pp 367–371
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Santosh KC, Wendling L, Antani SK, Thoma GR (2016) Overlaid arrow detection for labeling regions of interest in biomedical images. *IEEE Intell Syst* 31(3):66–75
- Santosh KC, Alam N, Roy PP, Wendling L, Antani S, Thoma G (2016) Arrowhead detection in biomedical images. In: *Electronic imaging, document recognition and retrieval XXIII*, vol 7. Society for Imaging Science and Technology, pp 1–7
- Santosh KC, Alam N, Roy PP, Wendling L, Antani SK, Thoma GR (2016) A simple and efficient arrowhead detection technique in biomedical images. *Int J Pattern Recogn Artif Intell* 30(5):1–16
- Ramer U (1972) An iterative procedure for the polygonal approximation of plane curves. *Comput Gr Image Process* 1(3):244–256
- Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Can Cartogr* 10(2):112–122
- Prasad DK, Leung MK, Quek C, Cho S-Y (2012) A novel framework for making dominant point detection methods non-parametric. *Image Vision Comput* 30(11):843–859
- Sakoe H (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans Acoust Speech Signal Process* 26:43–49
- Keogh EJ, Pazzani MJ (1999) Scaling up dynamic time warping to massive dataset. In: *European PKDD*, pp 1–11
- Müller H, Kalpathy-Cramer J, Egge I, Bedrick S, Radhouani S, Bakke B, Kahn CE Jr, Hersh W (2010) Overview of the clef 2009 medical image retrieval track. *Multilingual information access evaluation II*. *Multimedia Experiments*. Springer, New York, pp 72–84
- Zhang D, Lu G (2002) Shape-based image retrieval using generic fourier descriptor. *Signal Process Image Commun* 17:825–848
- Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. *IEEE Trans Pattern Anal Mach Intell* 24(4):509–522
- Kim W-Y, Kim Y-S (2000) A region-based shape descriptor using zernike moments. *Signal Process Image Commun* 16(1–2):95–102
- Hoang TV, Tabbone S (2012) The generalization of the r-transform for invariant pattern representation. *Pattern Recogn* 45(6):2145–2163
- Santosh KC, Lamiroy B, Wendling L (2013) Dtw-radon-based shape descriptor for pattern recognition. *Int J Pattern Recogn Artif Intell* 27(3):33