UDEL CIS at ImageCLEF Medical Task 2016

Pengyuan Li, Scott Sorensen, Abhishek Kolagunda, Xiangying Jiang, Xiaolong Wang, Chandra Kambhamettu*, Hagit Shatkay*

Department of Computer and Information Sciences,
University of Delaware, Newark, DE US
{pengyuan,sorensen,chandrak,shatkay}@udel.edu

Abstract. Figures play an important role within biomedical publications. A typical and essential first step toward using images is the detection of compound figures and their separation into panels. In ImageCLEF'16 our team has participated in the compound figure detection and separation tasks, where we utilized a method based on connected component analysis (CCA) to detect and to separate compound figures, while extending CCA in several ways to improve correct detection of subfigures while avoiding over-fragmentation. We have also participated in the Subfigure Classification task, where we employed an array of global image characteristics and a merge-split strategy coupled with neural network classifiers. We describe here the methods used in each task and analyze the performance of our system.

Keywords: Compound figure detection, Compound figure separation, Visual Information, Biomedical image classification, Connected Component Analysis, Neural Networks, ImageCLEF 2016.

1 Introduction

Our team has participated in the ImageCLEF'16 Medical Compound Figure Detection, Figure Separation and Subfigure Classification tasks [8]. For the first two, our approach is based on Connected Component Analysis (CCA), extending and significantly improving upon our work from ImageCLEF'15 [20]. For the classification task, we employed primarily global features based on color and gradient statistics along with a merge-split strategy, aiming to improve performance on classes that tend to be confused with one another. In this report we focus primarily on the compound figure detection and separation tasks, and briefly go into the classification task. The details of our approach and discussion of performance are provided in the following sections. In Section 2 we discuss our approach to compound figure separation; Section 3 presents our method for compound figure detection, while Section 4 discusses our approach to subfigure classification; Section 5 summarizes the work.

[*] These authors contributed equally to the work.

2 Compound Figure Separation

About 40% to 50% of figures in biomedical documents are compound figures consisting of multiple panels [6]. Segmenting compound figures into constituent panels is an essential first step for harvesting information from different panels. Current compound figures separation techniques can be classified into two main methods: The first is based on finding gaps in the figure [13, 1, 3, 2, 4, 21, 18]. Gaps are identified by finding sharp peaks in axis projections along the figure. As this can lead to both over- and under-fragmentation, figure caption analysis and separation analysis has been used as an aid for determining the correct number of panels (fragments).

In contrast, rather than look for separators we employ Connected Component Analysis (CCA) [16], which aims to directly identify large well-connected regions within the figure. However, small connected objects that are not connected to the rest of the image (such as legends or labels) may also lead to over-fragmentation. To address this issue, we "adopt" small fragments in a post-processing step.

The above methods do not address stitched compound figures, that is, figures in which panels directly touch one another and are not separated by gaps. Santosh, et al [15] first proposed a method to separate stitched compound figure based on Line Segment Detector. However, their method requires first manually separating stitched compound figures from the rest of the dataset, as opposed to a fully automated classification process. In this paper, we propose a figure separation scheme based on connected component analysis. To avoid over-fragmentation, we develop a separation quality assessment step. Furthermore, we employ the SUSAN edge detector [17] to separate stitched compound figures.

2.1 Methods

Compound figures consist of several panels, typically separated by gaps, which appear as vertical or horizontal light/dark bands; however, such gaps may be blurry or too thin to recognize. We first preprocess compound figures by resizing, adjusting, and cropping them to make the gaps in the images clearer and broader. We then apply Connected Component Analysis (CCA) to separate compound figures into constituents panels. As part of this step, (to which we refer throughout the rest of this report as CCA), we set thresholds to eliminate small objects and keep only the main components as individual panels, as we have done before [20].

Notably, this process may not be effective in several cases, namely: individual panels whose contents may not be well-connected, very blurry panels, and stitched compound figures. For the first two, which often occur in displayed graphics, we apply SUSAN edge detection[1] on the original compound figure, thus sharpening the blurry components and keeping the connectivity within the panels. For stitched compound figures, we apply SUSAN edge detection to find the

[1] We have experimented with several edge-detection methods and found SUSAN to have the best performance in this context.

positions where adjacent pixels are sharply changed. The boundary can then be detected by finding the peak value of the sum projections that are calculated by summing the pixel values along the horizontal and the vertical directions. Some compound figures are separated by partial gaps, where boundaries do not cross the entire image; we separate such figures by finding the projection with the highest peak value along only one direction. The projections are then calculated for the separated sub-figures, and the separation is recursively repeated until the highest peak of projections falls below a threshold.

As a last step, we employ a separation quality assessment step that we have developed to prevent over- and under-segmentation. It consists of five steps: 1) Merge overlapping panels; 2) Temporarily eliminate small disconnected components; 3) Recover missing panels; 4) Separation of potentially overlapping regions; 5) Small component recovery. The complete framework is shown in Fig. 1, and is further described below.

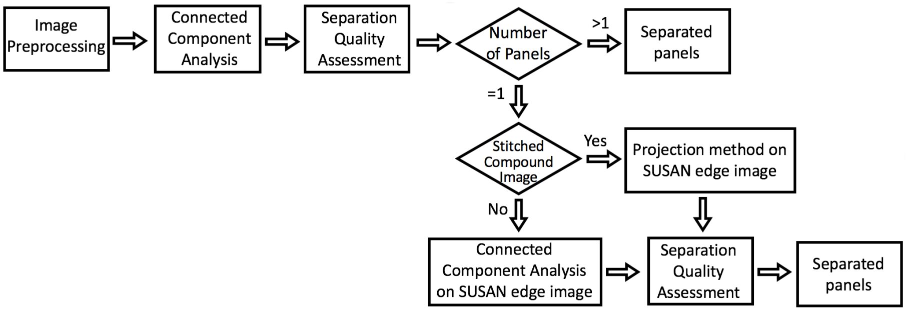
Image Preprocessing
Connected Component Analysis
Separation Quality Assessment
Number of Panels
>1
Separated panels
=1
Stitched Compound Image
Yes
Projection method on SUSAN edge image
No
Connected Component Analysis on SUSAN edge image
Separation Quality Assessment
Separated panels

Fig. 1. An outline of the figure separation approach

Image Preprocessing Gaps in compound figures typically separate sub-panels into clear individual components. However, some panels may be positioned too close to one another, or a thin gap may be noisy or blurred, making separation hard. To address this issue, we first scale-up the original image I, of size $m \times n$, to $2m \times 2n$, using nearest neighbor interpolation, which broadens the separating gaps. Notably, gaps in compound figures are not always white or black, that is, the intensity of pixels in the gaps is non-binary. To make the gap clearer, we adjust the intensity of figures, remapping pixel intensity that is below a lower threshold T_{low} or above an upper threshold T_{high} in the resized image $I_{resized}$ to 0 and 1 values, respectively. This enhances contrast in the image so that gaps, which are the lightest or the darkest bands in the figure, will be clearer. In our experiments, we set T_{low} to 0.05 and T_{high} to 0.95.

Connected Component Analysis Compound figure segmentation implies identifying panels that can be separated by boundaries and contents. We assume that the gaps among panels are white (which can be reversed later by inverting pixel values). We thus generate a mask of image foreground M by setting a threshold t and mapping all grayscale pixels whose value is below t to 0 and all those whose value exceeds t to 1. In our experiments the threshold t is set to 0.95. We then detect the connected components in M by using the Connected Component Labeling method [9]. This method is based on labeling gray values of similar intensity of adjacent pixels using the same label. A connected component is a set of pixels in which each pair of adjacent pixels share the same label. In the labeling process we employ here, we use 4-way adjacency to calculate connectivity. A panel bounding box is then set around the smallest rectangle that contains all pixels in each connected component. To detect panels in images with black gaps we reverse the gray value of every pixel in the image.

Using connected component analysis may generate some small bounding-boxes due to small and unconnected objects in the image, such as text. We thus set two thresholds to initially eliminate bounding boxes of very small height or width: $t_{height} = height/20$, $t_{width} = width/20$, where *width* and *height* are the total figure width and height.

Connected Component Analysis on SUSAN edge image If the figure cannot be separated through the step above, we employ a classifier that labels it as either a *stitched compound figure* or a *compound figure with gaps*. We define here a gap as a row or a column whose minimum gray value is higher than 0.95. If a gap is found in the figure, the latter is labeled as a *compound figure with gaps*. Otherwise, it is labeled as *stitched*.

Two kinds of images with gaps cannot be directly separated by the CCA method: images that have very blurry components and images that have components with very low internal connectivity. To address the first issue, we apply SUSAN edge detector to the preprocessed image, which sharpens blurry components in the image. Still, the corresponding SUSAN edge image, denoted I_{susan}, may have poor connectivity. To enhance connectivity of components in I_{susan}, we dilate the connected regions within the SUSAN edge image using the minimum gap-width in the image as the dilation factor. After dilating, the connectivity of connected-regions within the dilated SUSAN edge image is increased. We then apply the CCA method to the dilated image.

Projection on SUSAN edge image For stitched compound figures separation, identifying panel boundaries is the main challenge. To detect these boundaries we propose a separation method based on the application of SUSAN edge detector to the figure. The edge detector is first applied to the preprocessed image, which highlights the boundary between panels. Thus the objective becomes that of detecting boundaries shown in the resulting SUSAN-edge image. Given a SUSAN-edge image I_{susan}, if $I(x, y)$ is a detected edge we have $I_{susan}(x, y) = 1$.

Therefore, summing the pixel value along the horizontal and the vertical directions gives rise to two projections: $Proj0°$ and $Proj90°$. The panel separation is done along the horizontal or the vertical line that goes through the highest projection position. For figures with complex layout, the boundary between panels may not cross the whole figure; in this case we recursively separate the figure along one direction at a time, where the projection peak value is at least 0.7 of the height or the width of the region currently considered for separation.

Separation Quality Assessment

a. Merge Overlapping Panels: Because CCA is based on the connectivity of components in the image, some bounding boxes may overlap. For example, the bounding box of legends may overlap with the bounding box of corresponding line graphs. Therefore, we choose to merge two bounding boxes when their overlap ratio is larger than 0.1.
b. Temporarily Eliminate Small Disconnected Components: In the CCA separation we try to eliminate noise and text by setting a threshold for the size of bounding boxes. In this step we eliminate the subfigures that are too small when compared with the biggest bounding box from the separation result. When the height or width of a subfigure is less than the 1/5 of maximum height or width of all subfigures, this subfigure will be removed.
c. Recover Missing Panels: Due to blurriness in some of the compound figures, some panels will be recognized while others may be omitted/missed in our separation process. Therefore, we propose a missing subfigure recovery method. We assume that the missing panel is similar in size and symmetric in position to present panels. We thus check for each present panel whether there is enough space for another bounding box to its left, right, above or below. The candidate panels are expected to have similar black area ratio, and the boundary of candidate panels are verified by checking if all pixels in the boundary have a variance less than 5.
d. Separation of Overlapping Regions: We calculate the dark area overlap. If the overlap-rate is less than 0.5 we consider the separation to be incorrect.
e. Small Component Recovery: While we have eliminated text and some relatively small regions from our separation result, there may be some essential parts to be eliminated. We will merge those bounding boxes to their closest qualified bounding box. To do this we employ several rules.

1. If merging changes both height and width of a qualified bounding box – do not merge.
2. If merging changes more than 20% of height or width of a qualified bounding box – do not merge.
3. If the change of height or width for a qualified bounding box is more than 20%, this qualified bounding box keeps its original size.
4. Each small bounding box can only be merged once.

Last, we experiment with an automatic cropping scheme that simply subtracts the position of the top-left corner of the top-left bounding box, from the coordinates of all bounding boxes. This ensures that all top-left panels are anchored at position $[0, 0]$, which reflects the bounding boxes position in the training data.

2.2 Runs

Here we describe each of our submitted runs for the Figure Separation task:

- FS.run1: Connected Component Analysis.
- FS.run2: Combination of the image preprocessing and Connected Component Analysis.
- FS.run3: Combination of separation quality assessment and Connected Component Analysis.
- FS.run4: Combination of image processing, separation quality assessment and Connected Component Analysis.
- FS.run5: Combination of image preprocessing, separation quality assessment, Connected Component Analysis and Connected component analysis method over results of SUSAN edge image.
- FS.run6: Combination of image preprocessing, separation quality assessment, Connected Component Analysis and projection on SUSAN edge image.
- FS.run7: The framework proposed in Fig. 1 without adopting small objects (small component recovery).
- FS.run8: The framework proposed in Fig. 1 with small component recovery.
- FS.run9: The results from FS.run8 with automatic cropping employed.

Table 1. Compound Figure Separation (FS) results, shown per run

Run	Run Type	Correctly Classified (%)	# of unseparated
FS.run1	Visual	73.57	162
FS.run2	Visual	74.30	146
FS.run3	Visual	75.27	202
FS.run4	Visual	74.38	243
FS.run5	Visual	81.23	85
FS.run6	Visual	84.03	13
FS.run7	Visual	84.08	9
FS.run8	Visual	83.04	9
FS.run9	Visual	**84.43**	9

2.3 Results and Discussion

Results from all our runs are shown in Table 1. The simple Connected Component Analysis approach (originally used in [20]) achieves a relatively low accuracy of 73.57%. Preprocessing slightly improved the accuracy, at the cost of fewer images being successfully separated. Adding separation quality assessment further improves accuracy. Combining all our methods improves results, where the highest accuracy is achieved using the complete framework, along with automatic cropping, with a total accuracy of 84.43%. As we were the only team participating in this task, these results are not comparable to other groups.

3 Compound Figure Detection

This task was first introduced in 2015 [7], aiming to identify whether a figure is compound or not. We use a text-based method applied to the figure-caption, image-based methods, and combination methods for compound figure detection. We first introduce the methods, followed by a description of the 10 runs submitted to ImageCLEF 2016 Compound Figure Detection task. We then show and discuss the results from each run.

3.1 Methods

Text-based compound figure detection Captions in compound figures often contain specific delimiters, separating the description of each panel. That is, figure caption that contain at least two of these delimiters are often captions of compound figures. Therefore, we employ a text-based compound figure detection method by extracting these delimiters from the captions. To select the delimiters, we manually analyzed captions of figures from the training set, and identified a list of delimiters that frequently occur in compound figure captions. These delimiters are used to detect compound figures, that is, a figure is classified as compound if the corresponding caption has more than two of these delimiters. A list demonstrating the typical delimiters used is shown in Table 2.

Table 2. Representative delimiters used for text-based compound figure detection

Delimiter Kind	Examples
Arabic numeric delimiters	1, 2, 3, 4, or 1), 2), 3), 4),
Roman numeric delimiters	I, II, III, IV, or i, ii, iii, iv,
Alphabetic delimiters	a, b, c, d, or (A), (B), (C), (D),
Direction delimiters	top, bottom, middle, upper

Image-based compound figure detection The image-based compound figure detection method we employ uses the result from the figure separation method (see Section 2) to classify the figures as compound vs. non-compound. If a figure can be separated into multiple panels, it is labeled *compound*, otherwise, it is labeled *non-compound*. In addition to the separation-based decision, we extract three features for compound figure detection:

- First, the bounding box with maximum area divided by the area of the image (Maximum Bounding box Area f_1).
- Second, the sum of the number of solid columns and rows in the image (Number of Separating Lines f_2).
- Last, we form two binary-vectors: a *row-based* vector where the value 1 is placed at each position in which a solid column starts (0 in all other positions), and a *column-based* vector where the value 1 is placed at each position in which a solid row starts (0 in all other positions). We multiply each of

these vectors by a Gaussian whose standard deviation is proportional to the image size, thus giving more weight to solid rows and columns that are at the center of the figure (as opposed to the margins). We sum all elements of the weighted vectors and take the result as a third feature, (which we call the *solid axis matrix*, denoted f_3).

Combined methods As both caption and image provide important information for deciding whether a figure is compound, we use several combination-methods based on features extracted from both captions and images.

One combined method is a simple logical 'OR' (union) of the two predictions. In this case, if either the text-based detection method or the image-based method labels a figure as *compound*, the figure is assigned the *compound* label.

The second combined method employs a Decision Tree classifier on the text-based and image-based prediction results. Each figure is represented as a two-dimensional binary vector $\langle r_1,\ r_2\rangle$, in which r_1 is the label assigned by the text-based method (where 0 denotes *non-compound* and 1 *compound*). Similarly, r_2 is the label assigned by the image-based method. All figures in the training set are used by the J48 algorithm on Weka [10] to build a decision tree model.

The third combined method employs a Decision Tree classifier on the text-based prediction results, image-based prediction results along with the three additional image features listed before. Thus, each figure is represented as a 5-dimensional vector $\langle r_1, r_2, f_1, f_2, f_3\rangle$. r_1 and r_2 are as in the above decision tree, while f_1 , f_2 and f_3 are the three additional features described earlier. All figures in the training set are used by the J48 algorithm (Weka implemntation) to build a decision tree model.

3.2 Runs

We have submitted 10 runs in total for the compound figure detection subtask, as follows:

- FD.run01: Uses only the text-based detection method.
- FD.run02: Uses the figure separation result from FS.run4 (see Sec. 2.2).
- FD.run03: Uses the figure separation result from FS.run7 (see Sec. 2.2).
- FD.run04: Represents each figure in the test-set as a 5-dimensional vector based on the results of FD.run2, and the three features mentioned in Sec. 3.1 (Image-Based detection), then employs the Decision Tree classifier built over the training set.
- FD.run05: Represents each figure in the test-set as a two-dimensional vector based on the results of FD.run1 and FD.run2, then employs the Decision Tree classifier built over the training set.
- FD.run06: Uses logical OR (Union) over the results from FD.run1 and FD.run2.
- FD.run07: Represents each figure in the test-set as a two-dimensional vector based on the results of FD.run1 and FD.run3, then employs the Decision Tree classifier built over the training set.

- FD.run08: Uses the logical OR relationship (union) over the results from FD.run1 and FD.run3.
- FD.run09: Represents each figure in the test-set as a five-dimensional vector based on the results of FD.run1, FD.run2 and the three additional features, then employs the Decision Tree classifier built over the training set.
- FD.run10: Represents each figure in the test-set as a five-dimensional vector based on the results of FD.run1, FD.run3 and the three additional features, then employs the Decision Tree classifier built over the training set.

3.3 Results and Discussion

Our results are given in Table 3. We achieve an accuracy of 90.74% using our combined textual and visual approach that represents images as 5 dimensional vectors using feature extraction, showing that the combined method improves upon the purely visual methods. This result is among the very top achieved on this task this year.

Table 3. Compound Figure Detection (FD) results, shown per run

Run	Run Type	Correctly Classified (%)
FD.run01	Textual	85.47
FD.run02	Visual	89.29
FD.run03	Visual	69.82
FD.run04	Visual	89.64
FD.run05	Mixed	90.39
FD.run06	Mixed	52.25
FD.run07	Mixed	85.47
FD.run08	Mixed	69.06
FD.run09	Mixed	90.39
FD.run10	Mixed	**90.74**

4 Subfigure classification

We have used two techniques to address subfigure classification.

- Training a hierarchical classifier using merge-split approach.
- Fine tuning a pre-trained deep-network.

As our deep learning approach has not proven effective in this setting, we will not discuss it further in the interest of focusing on the hierarchical approach.

4.1 Methods

The complexity of a classifier is typically proportional to the number of classes in the data and the amount of training data required is proportional to the complexity of the classifier. Due to limited training data available, and classifier limitations, some of the classes are not easily distinguished from one another. To combat this we have augmented the ImageCLEF training dataset with images collected from the web. Furthermore we have developed an approach to target specific cases of easily mis-classified images. Given a classifier and training data we use a *merge-split* approach, in which classes that tend to be easily confused are merged into a single class, and the classifier is re-trained accordingly. Another classifier is then trained to classify (split) each merged class into its constituents. Details of the image features and the classifier used are discussed in the following subsections.

Image Features The global image feature used is a combination of 4 types of feature-vectors that capture color distribution, contrast, gradient orientations and distribution of local patterns in an image, as follows:

- The first type of feature vectors consist of information regarding the mean and co-variance of the color histogram. We compute histograms with 10 bins for each color channel and compute the mean and co-variance across these histograms. This results in a 10 and a 9-dimensional vector, respectively, both of which are normalized.
- The second set of components is based on a Histogram of Oriented Gradients [5] computed over the entire image. This forms a 31-dimensional vector.
- The third type of feature-vector consists of properties derived from gray level co-occurrence matrix (GLCM) [11] with 8 gray levels on the image. The properties include contrast, correlation, energy and homogeneity. This forms a 4-dimensional vector.
- The last set of features is the histogram of local binary pattern (LBP) [14] computed over the entire image, giving rise to a 58-dimensional vector.

The VLFeat library was used for feature extraction [19]. Each of these feature vectors is normalized and they are all concatenated to form a global feature representation for the given image.

Before feature extraction, the images are preprocessed to remove bright and dark borders with constant intensity as they do not contribute to image information content. The cropped images are then re-sized to be of size 64×64.

Classifier and Training We use a neural network multi-class classifier [12]. The network has a hidden layer with 10 neurons, the input layer has 112 units (same as the size of the feature vector). Stochastic gradient descent is used for training the network. The target error is set to 0.01 and the maximum iterations number to 1000. Training a classifier using the merge-split approach involves the following steps:

- Train the neural network over the given training data.
- Use the confusion matrix to merge classes that are easily confused: For each class A if more than 10% of its instances are classified as class B, then A and B are merged. Note that more than 2 classes can be merged into a combined class. Also note that there may be classes that are not merged or merged classes consisting of 2 or more individual classes.

- Train a base-classifier to distinguish between the new set of classes, obtained by the merging process above.
- Train a subordinate classifier to split each of the merged-classes, that is, classify their instances into the original constituent classes.

4.2 Results and Discussion

The merge-split training approach was used to train a set classifiers. 10 different classifiers were trained using random initialization and the one that did best on the training data was used on the ImageCLEF test data. We used two approaches for doing this:

- Sequential: the training steps described in Section 4.1 were used sequentially to train 10 different classifiers and the best among them was picked;
- Parallel: the merge step was repeated 10 times and the best classifier was chosen before performing the split step 10 times with the selected set of classes. We trained models on both the original training data and our augmented training data. The results on imageCLEF 2016 test data are shown below.

We submitted 4 runs with classifiers trained using the merge-split method:

- Run 1: Sequentially trained classifier on ImageCLEF training data.
- Run 2: Sequentially trained classifier on augmented training data.
- Run 3: Parallel trained classifier on ImageCLEF training data.
- Run 4: Parallel trained classifier on augmented training data.

We also submitted 3 runs with classifiers trained on the augmented dataset using our deep learning approach (Runs 5-7). These are different models that were trained in the same manner, with varying number of iterations. Table 4 shows results on ImageCLEF test data.

Table 4. Subfigure Classification results

Run	Correctly Classified (%)
Run 1	**72.46**
Run 2	71.53
Run 3	68.17
Run 4	68.79
Run 5	15.62
Run 6	53.16
Run 7	53.24

5 Conclusion

We have presented our approach for addressing the Compound Figure Separation, Compound Figure Detection, and Subfigure Classification tasks at ImageCLEF'16 Medical. Our approach for figure separation is based on connected component analysis, and extends it via preprocessing steps, SUSAN edge detection for improved segmentation and for addressing stitched-figure segmentations, and separation quality assessment. For compound figure detection, we have developed and employed text-based, visual and combined methods. The text based method leverages common delimiters typically found in captions of compound figures. The visual separation techniques employ our figure separation approach to identify images that can or cannot be split into panels. We use decision trees to combine the methods. We were the only team to submit an entry to the Figure Separation task, and our results on the Compound Figure Detection task are among the very top this year.

For subfigure classification we developed an approach based on global features extracted from the images. The features include statistics derived from color, gray level, edge orientation and local patterns. Once images were represented through these features, we used a merge-split scheme and trained neural network classifiers. All the techniques presented throughout this paper offer important capabilities to document analysis approaches that seek to extract information from figures.

Acknowledgment

This work was partially supported by NIH grant R56LM011354A.

References

1. S. Antani, D. Demner-Fushman, J. Li, B. V. Srinivasan, and G. R. Thoma. Exploring use of images in clinical articles for decision support in evidence-based medicine. In *Proc. of SPIE Document Recognition and Retrieval (DRR'08),* 68150Q, 2008.
2. E. Apostolova, D. You, Z. Xue, S. Antani, D. Demner-Fushman, and G. R. Thoma. Image retrieval from scientific publications: Text and image content processing to separate multipanel figures. *Journal of the Association for Information Science and Technology*, 64(5):893–908, 2013.
3. B. Cheng, S. Antani, R. J. Stanley, and G. R. Thoma. Automatic segmentation of subfigure image panels for multimodal biomedical document retrieval. In *Proc. of SPIE Document Recognition and Retrieval (DRR'11),* 78740Z, 2011.
4. A. Chhatkuli, D. Markonis, A. Foncubierta-Rodrguez, F. Meriaudeau, and H. Müller. Separating compound figures in journal articles to allow for subfigure classification. In *Proc. of SPIE Medical Imaging,* 86740J, 2013.
5. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05),* 886-893, 2005.

6. A. G. S. de Herrera, J. Kalpathy-Cramer, D. Demner F., S. Antani, and H. Müller. Overview of the ImageCLEF 2013 medical tasks. In *Working Notes of CLEF 2013 (Cross Language Evaluation Forum)*, 2013.
7. A. G. S. de Herrera, H. Müller, and S. Bromuri. Overview of the ImageCLEF 2015 medical classification task. In *Working Notes of CLEF 2015 (Cross Language Evaluation Forum)*, 2015.
8. A. G. S. de Herrera, R. Schaer, S. Bromuri, and H. Müller. Overview of the ImageCLEF 2016 medical task. In *Working Notes of CLEF 2016 (Cross Language Evaluation Forum)*, 2016.
9. R. C. Gonzalez and R. E. Woods. *Digital image processing.* Prentice-Hall, 2002.
10. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
11. R. Haralick and L. Shapiro. *Computer and robot vision.* Addison-Wesley Pub. Co., 1993.
12. M. F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.
13. R. F. Murphy, M. Velliste, J. Yao, and G. Porreca. Searching online journals for fluorescence microscope images depicting protein subcellular location patterns. In *Proc. of IEEE International Symposium on Bioinformatics and Bioengineering (BIBE'01),* 119-128, 2001.
14. T. Ojala, M. Pietikinen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on pattern analysis and machine intelligence*, 24(7):971–987, 2002.
15. K. C. Santosh, S. K. Antani, and G. R. Thoma. Stitched multipanel biomedical figure separation. In *Proc. of the IEEE International Symposium on Computer-Based Medical Systems (CBMS'15),* 54–59, 2015.
16. H. Shatkay, N. Chen, and D. Blostein. Integrating image data into biomedical text categorization. In *Proc. of the Int. Conf. on Intelligent Systems for Molecular Biology (ISMB'06), Bioinformatcs*, 22(14):446–453, 2006.
17. S. M. Smith and J. M. Brady. SUSAN—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.
18. M. Taschwer and O. Marques. Compound figure separation combining edge and band separator detection. In *Proc. of International Conf. on Multimedia Modelling (MMM'16),* 162–173, 2016.
19. A. Vedaldi and B. Fulkerson. VLFeat: an open and portable library of computer vision algorithms. In *Proc. of ACM Int. Conf. on Multimedia (MM'10),* 1469-1472, 2010.
20. X. Wang, X. Jiang, A. Kolagunda, H. Shatkay, and C. Kambhamettu. CIS UDEL working notes on imageclef 2015: Compound figure detection task. In *Working Notes of CLEF 2015 (Cross Language Evaluation Forum)*, 2015.
21. X. Yuan and D. Ang. A novel figure panel classification and extraction method for document image understanding. *International Journal of Data Mining and Bioinformatics*, 9(1):22–36, 2014.