Insect Soup Challenge: Segmentation, Counting, and Simple Classification

Katarina Mele
Computational Informatics, CSIRO
Riverside Life Sciences Centre, 11 Julius Avenue, North Ryde, NSW, 2113
Katarina.Mele@csiro.au

Abstract

In the paper we present a method for segmentation of insects from the Insect Soup images. The method enables reliable segmentation of insects of variable size, shape and color. After segmentation, a set of properties are assigned to each segmented insect which enables classification into different categories. The approach was successfully applied on two different types of real life images: images from the Insect Soup Challenge and images acquired from traps in the field using low resolution cameras.

1. Introduction

Insect classification belongs to the same category as image categorization in computer vision. The problem is known for being very challenging, and a general solution does not exist. Categorization is difficult for several reasons: objects that belong to the same category do not look exactly the same, they can be viewed under different angles, they might vary in size, and they can be occluded. Knowing the position of each object of interest is very useful information and can be used to support accurate classification.

In this paper we present a method for segmentation of insects from the insect soup imagery. The segmentation method and its extension to handle small and faint objects are described in the following chapters. Next, examples of simple classification are given. Finally benefits, issues, and potential improvements of the proposed method are discussed.

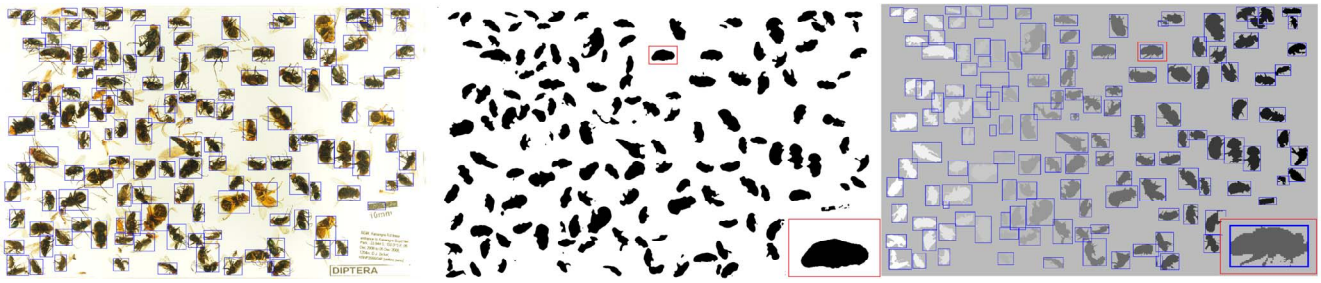
2. Segmentation Method

Images given for Insect Soup Challenge are of high quality. The background is reasonably even on most of the images. Images are mostly noise free and do not require a lot of preprocessing. The only preprocessing step we do is applying a statistical filter to all images. The filter sorts the pixels in a 20x20 patch surrounding each pixel i and replaces i with the 220th pixel value in the sorted patch. In this way we remove small objects that mostly represent noise, we smooth the image and even the background.

Segmentation consists of two steps: coarse-global and fine-local segmentation. In the first step we use a global thresholding method. The result of this simple thresholding is a binary image, where white part represents the background (see Figure 1b) and black parts indicate the foreground (insects, bugs etc). Next we calculate the size and bounding box of each object in the foreground. We eliminate small objects as these are mostly noise. For the remaining objects we extend the bounding box by 30 pixels on each side in order to capture the whole insect's body. Finally we apply Seeded Region Growing Algorithm (SRG) [1] separately for each object, defined by the extended bounding box. SRG is a robust and fast algorithm for segmentation. It does not require any parameter tuning but it does require the input of seeds for background and foreground. Seeds for the patch within the extended bounding box are defined as follows. Pixels on the edge (3 pixels wide) represent samples for the background and the object itself represents the foreground. Pixels on the edge that belong to any other foreground objects are excluded from background seeds. After applying SRG more details are segmented correctly compared to the coarse first step segmentation. Results of both segmentation steps are shown on Figure 1 (b, c). In Figure 1a, each segmented object is indicated by a bounding box. Figure 1b shows results after coarse segmentation, and Figure 1c shows segmentation results after applying SRG, where comparatively more details are segmented correctly.

3. Handling small and bright objects

While the majority of objects are not sensitive to the global threshold value or size of the statistical filter used in preprocessing, small and bright objects are. Statistical filtering in the preprocessing step removes small objects that represent noise. Unfortunately it also removes tiny insects. To identify regions where small objects are present, we rerun the segmentation with settings that are more fitted for small and bright objects. For statistical filtering we use a 10x10 patch size and we replace the middle element with the 70th element. The global threshold is also increased in order to capture bright objects. After coarse segmentation all objects are slightly



(a) (b) (c)

Figure 1: (a) Original image with bounding box around each segmented object. Segmentation is a result of both segmentation steps. (b) Results after first step of segmentation. (c) Results after second step of segmentation. Insect in the right bottom corner is enlarged in order to see the difference between the two steps of segmentation. SRG captures more details than simple thresholding.

enlarged. To enlarge objects we use dilation where the structuring element is disk with radius 10 pixels. Next, we eliminate objects that include insects detected in first run of segmentation. In this way we concentrate only on objects that were not detected in the first iteration of segmentation. The remaining steps are exactly the same as those for detecting bigger objects.

A result of this two iteration process is shown in Figure 2. While parameters used for segmentation in the first iteration fail to identify small insects, parameters for small and faint objects used in second iteration identify the vast majority of them.

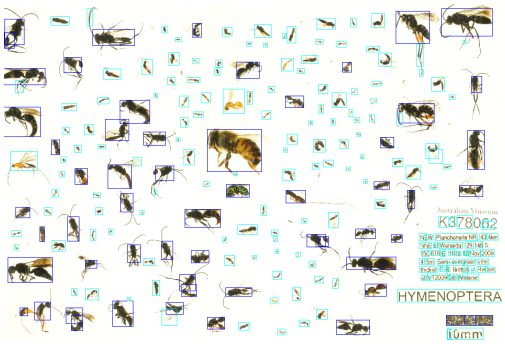


Figure 2: Example of two iteration segmentation. Objects detected in the first iteration are marked with blue bounding box while object detected in second iteration are marked with cyan.

4. Object properties and classification

Each image in a given dataset consists of several different types of insects. If we want to distinguish among them or analyze different visual properties, we have to calculate a set of descriptors for each segmented insect. We have chosen a set of standard descriptors (e.g. size, solidity, minor axis length, major axis length, min pixel value, average pixel value, eccentricity, perimeter etc.) which enable us to classify objects into simple classes. More details about the descriptors can be found in Matlab help [3]. All the descriptors are exported into Weka files. Weka is an open source program that enables further data visualization and exploration using different machine learning techniques [2].

5. Results and discussion

The only parameter that we need to set for the whole segmentation process is the global threshold in the first iteration. It is important that it captures at least part of each object and at the same time does not capture too much which can cause merging of two or more separate objects into one. It turns out that unless we want to capture very small and bright objects in the first segmentation iteration, the value of the global threshold does not need to be set very carefully. The second iteration of segmentation can often correct errors from first iteration adding objects that were missed. Figure 3 shows an example of the two iteration segmentation with different global thresholds used in the first iteration. We used threshold 400 for the left side and 600 for the right. Note that we worked with grayscale images that were obtained from RGB images simple by summing pixel values from all channels. One can notice that fewer objects were detected in the first iteration with lower threshold, and more in the second iteration. While segmentation with the higher threshold detected more objects initially in the first iteration. However, when we combine results from both iterations, the results are almost the same.

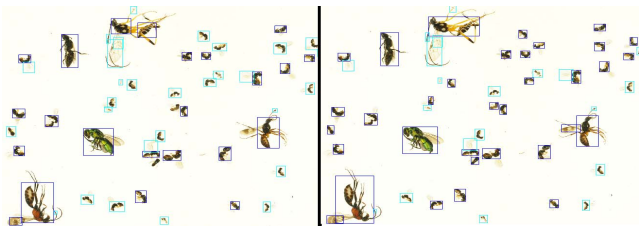


Figure 3: Left side of the image is the result of using lower threshold in the first iteration, and the right side is a result of using higher threshold in the first iteration. Blue bounding boxes are results of first and cyan of second iteration.

The second iteration of segmentation is designed to capture small and faint objects. Its global threshold is fixed to value 700. As insect extremities are usually very faint and thin, they are not captured in the segmentation results of the first iteration. However, they are often picked by second iteration and marked as tiny insects. (For example, see the two insects in the top right corner of Figure 2). These examples represent the majority of false positives and can be eliminated using similar procedures used for neurite detection. In a similar way that neurites are attached to the cell body of a neuron, insect extremities are attached to an insect's body. For example HCA-vision [4] enables reliable analysis of such data. Once extremities are identified we can easily eliminate false positives that represent insect extremities and not small insects.

The plate shown in Figure 4 is a very hard example for automatic image analysis. Insects are close together and there are many examples where they are touching

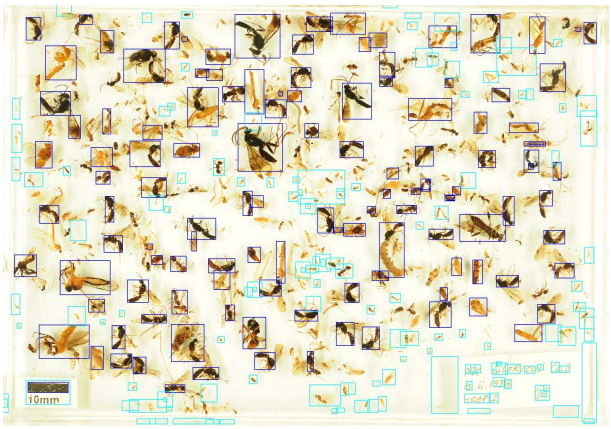


Figure 4: Example of a very challenging plate for automatic image analysis.

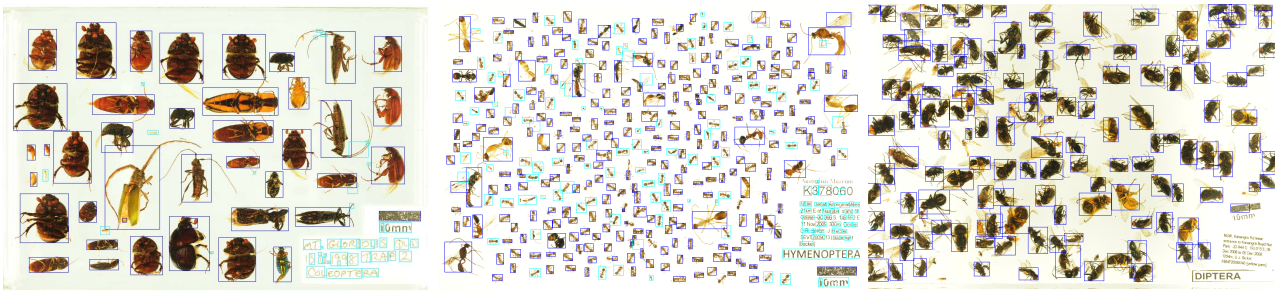


Figure 5: Three examples of two iteration segmentation. The method can successfully deal with vast variety of sizes as well as with certain degree of clutter. In the last image even beetles that are close together are correctly separated.

oroverlapping. A lot of debris is present in the plate, such as broken wings and other insect parts. The trap was most likely in the field for a longer period, and insects probably got damaged when they tried to escape from the trap, or their bodies decayed over time. One of the solutions to this problem is to change the sticky plate before it gets too crowded and the insects' bodies decay, which would make image analysis easier and more accurate. One can notice that the majority of insects with dark bodies are segmented correctly while insects with bright bodies are missed on a lot of occasions. The reason for this is that regions at the global segmentation in the second segmentation iteration are not separated and they merge together. As it is likely that part of the merged region was already detected in the first iteration the whole region is simply discarded. This is not a problem for isolated small/faint insects which are mostly correctly detected in the image.

More results of the two iteration segmentation are given in Figure 5 and supplementary material. Note that accurate segmentation also enables trivial counting of objects in the plate.

Examples of simple classifications are shown in Figure 6. Figure 7 shows segmentation examples from low resolution cameras. It also indicates basic classification where we eliminate false positives (marked with yellow). The lure was also segmented but successfully removed based on color. False positives were removed based on size and shape.

An example of data exploration using Weka is shown in Figure 8. The example shown here is for the image depicted in Figure 5 (middle image). We can see that that solidity separates insects into two groups – two peaks in the graph (Figure 8, left bottom corner). Solidity is a scalar specifying the proportion of the pixels in the convex hull that are also in the region. When we looked into the two classes we noticed that the generated masks for one class are much more precise than masks for the second class. We cannot classify the insects based on this knowledge however we can guide the whole classification process towards more accurate results. Masks with solidity close to 1 are mostly outliers – they fill the whole bounding box and look more like a square than an insect.

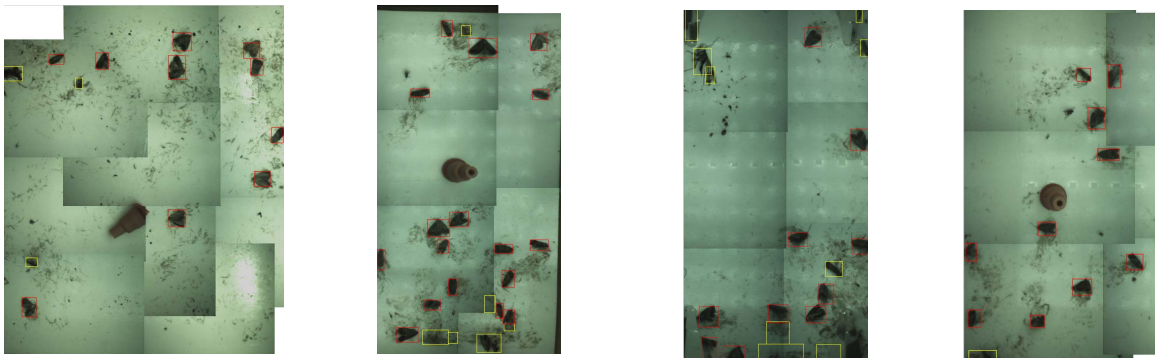


Figure 6: Examples of segmentation and object classification of images from low resolution cameras.

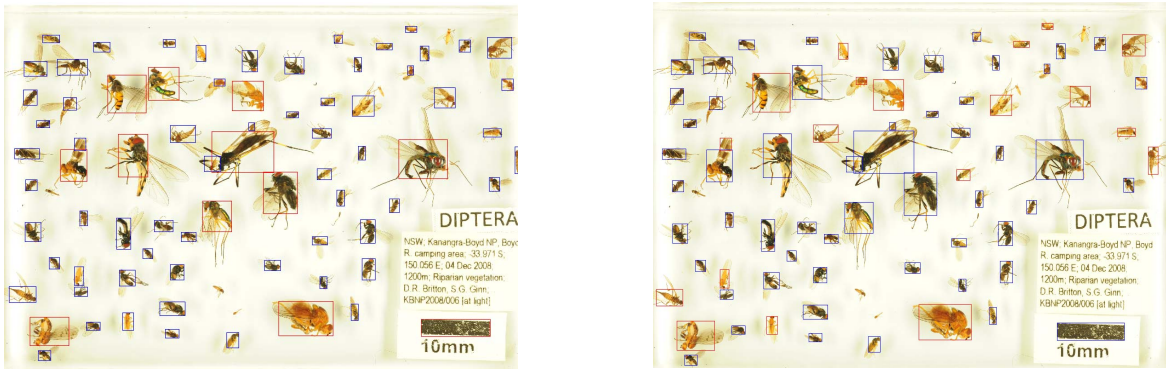


Figure 7: Example of a simple classification based on size (left image) and based on average color (right image). In the left image larger objects are marked with red bounding box, while smaller objects are marked with blue bounding box. In the right image darker insects are marked with blue bounding box and brighter with red bounding box.

6. Conclusion

We presented a method for segmentation and simple classification of insects from insect soup images. While the method is able to reliably segment and count most of the reasonably sized insects in the images, it can be improved when it comes to crowded regions with a lot of small and bright objects. Calculated descriptors enable a basic classification according to visual similarity; however this can be improved by incorporating domain specific knowledge. The system is a great foundation for a sophisticated automated system that would enable an automatic or semiautomatic classification of insects caught in traps and significantly speed up the sorting and cataloguing of insects from the insect soup imagery.

References

[1] R. Adams, L. Bischof Seeded region growing IEEE Trans. Pattern Anal. Machine Intell., 16 (6): 641–647, 1994.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten; The WEKA Data Mining Software: An Update; SIGKDD Explorations, 11(1), 2009.

[3] MATLAB 2013a. The MathWorks, Inc., Natick, Massachusetts, United States.

[4] D. Wang, R. Lagerstrom, C. Sun, L. Bischof, P. Vallotton, and M. Götte, HCA-Vision: Automated Neurite Outgrowth Analysis, Journal of Biomolecular Screening, 15(9):1165-1170, October 2010.

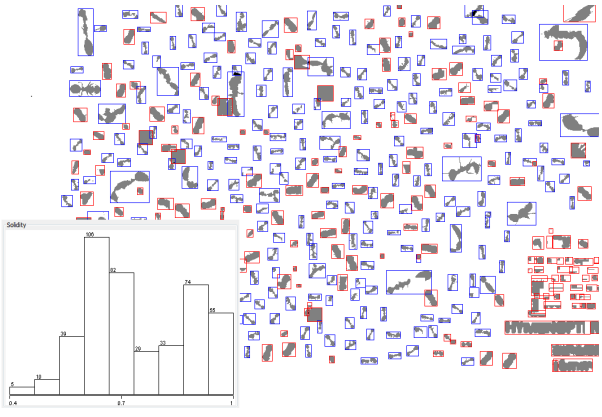


Figure 8: Masks classification based on solidity. Red squares represent masks with high solidity (>0.75) and the blue one masks with low solidity value.