Implementation of Object Recognition for an Environmental Monitoring System

Final Report for CSE190A: Projects in Vision & Learning

Haili Wang
Department of Computer Science Engineering
University California, San Diego
haw007@ucsd.edu

Abstract

This report is going discuss implementation of an object recognition module for an environmental observation system, particularly for bee population tracking. In principal, the methods discuss here also work for multiple instants of interested objects which appear to be a repeated pattern.

1 Introduction

The goal of this project is to build an object recognition module to solve the interested object counting problem of an ecological image database. Taiwan Forestry Research Institution (TFRI) deployed a remote camera on a bee nest for population tracking. The camera captures one picture every minute in daylight, and transfers those images to the server at night. In order to shorten the data process, this project implements a computer vision approach to count the number of bees in every picture. If the counting result turns out good enough, it can avoid transferring any image but instead just some numbers. The source images of this project come from Archiving Gallery of Shanping Workstation of TFRI [5]. This report is going to present the implementation of this project (section 2), and then talk about the problems of the current approach (section 3), compare the complexity of two implements and open the discussion to further development (section 4).

2 Implementation

Utilizing the rich image toolbox, this project is implemented on Matlab. It uses a typical supervised learning model, and goes through two stages. The in the first stage the algorithm collects user input of bees objects and non-bees objects by mouse click on an image. A series of filters are applied on the selected windows, and stored in a data structure for future reference. In the second stage of the algorithm takes any input image and try to detect bee objects and rejects non-bee objects by sliding a small window over the input image.

2.1 Basic parameters

The source images are taken by a webcam with daylight. Daylight changes drastically over time, and sometimes the camera may stand in rain. The raw source images are big (1600*1200), but a bee usually occupies a 30*30 window. That the bees usually group together becomes one of the most challenging problems in this project. Although the bee nest background is contract to the bees, bee nest usually collects random black dots over time. They may be caused by mud, or loss paint, which hesitate the algorithm. These black dots usually smaller than 10 pixels width, but some bigger region such as bee nest entrance can be 200 pixels width.

2.2 Statistic approach

The first problem of this implementation is that the bee images lack of strong features. Especially when they stand on a dark background, it is quite difficult even for human to distinguish whether there are bees. I experiment with a strong descriptor, SIFT, which can describe sophisticated objects. However the result does not turn out well. An individual bee is so small and dark that SIFT could not have enough information to generate a good descriptor for it.

Fortunately the camera is frontoparallel to the scene with relatively few distracted object. Bees can be viewed as a continuous flow of a unified object. For an individual bee on the background, I expect that the bees regions and non-bee region differ in histogram of color (HOC) and histogram of gradient (HOG). Histogram of gradient gives the description of the edges of the bees, and histogram of color accounts for the color of the bees. For a group of bees, they should have similar HOC and HOG. As RGB color space can be effected by change of lighting, chromatic channels (A* and B*) of LAB color space is used for color histogram. Input RGB color space is converted to grayscale before computing the gradient. In a window of a single bee, the algorithm expects some small region of light colored background and the bee shape object. Any window with similar histogram would be considered as a bee.

The comparison of histogram has to use the same distribution and normalized in order to have a meaningful result. I write my own histogram function, `his_fast.m`, which returns 33 bins for each channel, and significantly improve the performance by shortening a for-loop. The result of the first stage of the algorithm is store in two data structures called `positive.mat` and `negative.mat`, representing histograms of bee's windows and non-bee windows respectively.

At the second stage of the algorithm, an input image is processed by `blkproc.m` (block process) in Matlab, with the same process as the previous stage. `Blkproc()` can divide an image into small windows and apply the given function to each window. In this project the block size is 15 pixels and overlap by 8 pixels (15+8*2 = 31 pixels for each window). Figure 2 shows a representation of windows division. A Euclidian distance is computed between two histograms and any window is selected as positive or negative if the distance falls within a threshold value. Figure 1 shows a general pipeline of the program.

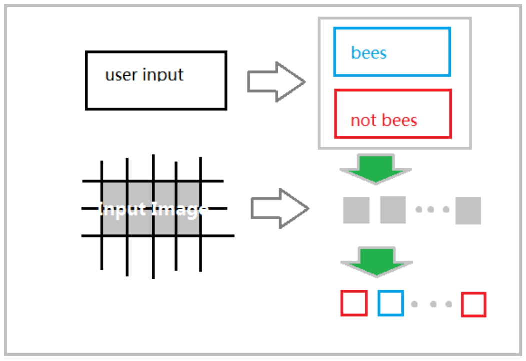

Figure 1 pipeline of the program

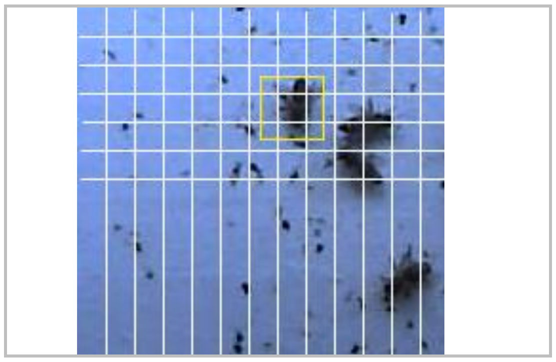

Figure 2 `Blkproc.m` representation of overlapping windows. The yellow box represents the processing region of a window.

2.3 Regression process

After collecting the possible windows, the algorithm turns to a regression process and turns a pixel count or positive window count to a bee count. As limited time of this quarter, the regression process is not yet implemented. A similar approach as crowd monitoring can be taken in the future [1].

2.4 Effect of median filter

A median filter replaces the pixels that are less than the median value with the median value. It can effectively eliminate those small black dots in the images while still let bigger objects, such as bees, get through with some loss of peripheral detail. As a result it reduces some false positives however it also reject some positive objects. Figure 3 shows the different before and after median filter.

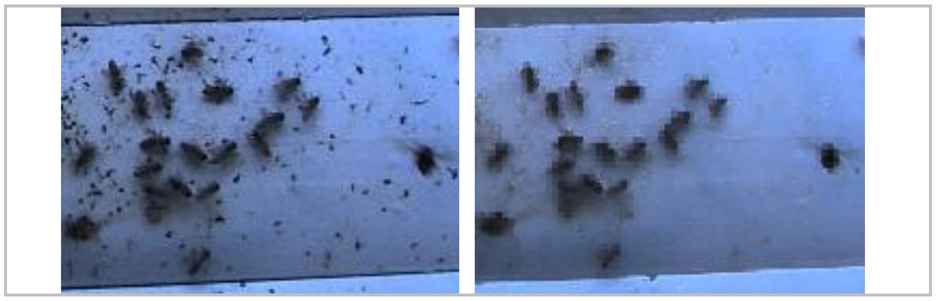

Figure 3 Median filtered
Left: before median filter; right: after median filter

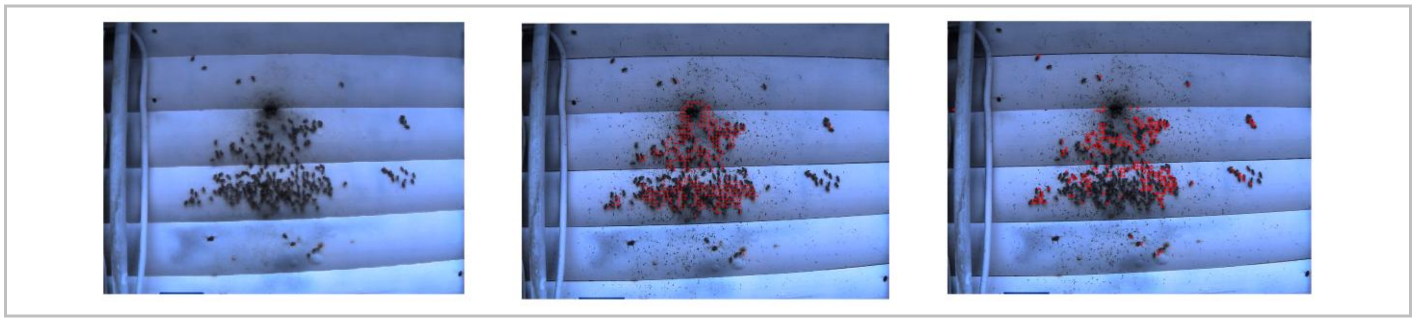

Figure 4 Result of median filter
Left: Image with median filter only; middle: without median filter, plot HOC/HOG result on the original image; right: with median filter, plot HOC/HOG result on the original image

Figure 4 shows the different results with and without median filter. Both results use the same 30 input samples. Without median filter the algorithm falsely select the upper middle region (bee nest entrance) as bees, and select the massively grouped bees at the middle. With median filter the algorithm avoids the bee nest entrance, however, also deselects the grouped bees.

3 Limitation of the current approach and another solution

3.1 The best result of HOC/HOG

When a big group of bees present on an input image, the gap between bees becomes very small and the background is noisy (mud or dark). Since the algorithm expects some light colored background in a positive window, the lacking of background and background color change can impact of the algorithm, and it falsely rejects those windows. In some other case, the edge of muddy regions can be falsely selected as the histograms of those regions are similar to the positive samples. Figure 5 shows a result of 170 positive samples and 170 negative samples with HOC/HOG algorithm, plot positive responses as red, and negative responses as blue. Note that the bee nest entrance region is selected with this sample set.

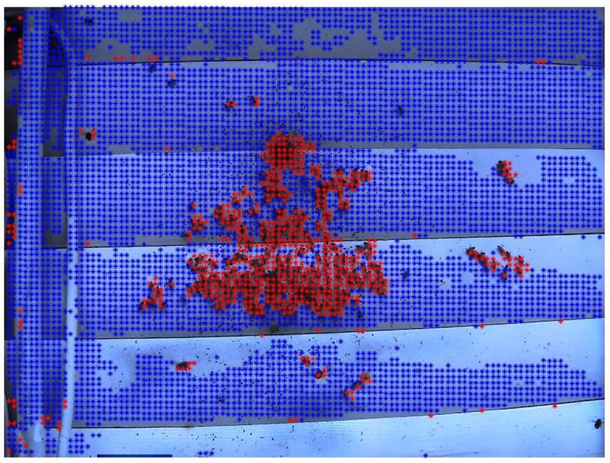

Figure 5 HOC/HOG algorithm with 170 input set results

3.2 Filter bank approach

Filter bank is a series of filters with different shapes, orientations, and sizes. When apply each filter to the image, the response of each filter represents a relation of the filter and the original image, such that the “bee shape” filters will respond better on the bee region, while some other shape filters respond to some other shapes.

When keeping the current framework unchanged, and applies a series of independent filters to an input image and considered those responds as channels. These filters consist of different shapes, orientations, and sizes. Figure 6 shows a typical filter bank used in this project. Index those filters from 1 to 40, among those responded channels; use an index table to store the largest absolute respond index for that pixel. Figure 7 shows the index table as an image. Notice that the upper middle region was a muddy bee nest entrance, which was falsely selected as bee region by the HOC/HOG algorithm. In the index table here it is clear that this region has put on distinguished texture from the real bee region.

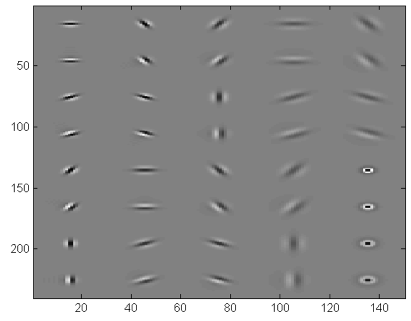

Figure 5 Filter banks

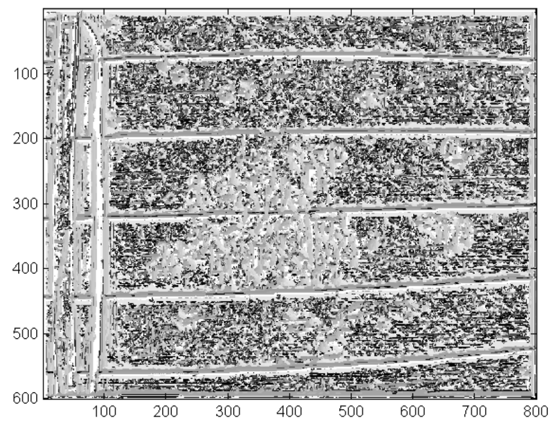

Figure 6 Filter bank max response index table view as an image

4 Algorithm complexity and future integration

As the camera is built in an on field machine with very limited computational power, and one of the goal of this project is to get back a population count instead of transferring all the images, it is important to analyze the complexity of the these two algorithms.

For HOC/HOG approach, the algorithm complexity is follow (assuming image size is n x n, sample size is k, and window size is w).

```
Median filter: O(n^2) +
Histogram: O(4n^2) +
Histogram comparison: O(k*n^2/w^2)
Yields O(k*n^2/w^2 + n^2)
```

For filter bank approach, the algorithm complexity is follow.

```
Convolve each filter, O(n^4) +
Histogram: O(4n^2) +
Histogram comparison: O(k*n^2/w^2)
Yields O(n^4)
```

In practice, HOC/HOG approach takes about 30 seconds including plotting bees, while filter bank approach takes more than 1 minute with only ¼ of the original image resolution, but it is possible to use fast Fourier transform to speed up the convolution process in the filter bank approach. Furthermore, for the propose of population monitoring, it is not necessary to use the original resolution.

The algorithms heavily depend on the supervised user input. The algorithms assume that the bee samples are rich and cover most of the scenarios. Such assumption is reflected on the output. In Figure 4 middle, individual bees have not responded because the sample bees are selected from an unclear background. To avoid this problem, user should accumulate various bees on different lightings and backgrounds across the image database. Especially for HOC/HOG approach, since HOC accounts for chromatic channels, color temperature difference affects the algorithm when daylight changes. In the future integration, the user samples should contain about 200 bees, which spread across different orientations and different density of bees, different kinds of daylight and weather.

Even though HOC/HOG approach has some limitation on different daylights and dark regions, it can be very useful when the interested objects are colorful, such as butterflies.

When filter bank approach can effectively distinguish the “bee texture” and the noisy background, it still needs information of different bee density to collect some sparsely located bees. The window size should be larger than the HOC/HOG approach, such that windows cover different density of bees. Further experiments are needed to decide an exact number.

5 Conclusion

I presented HOC/HOG approach for bee tracking and filter bank approach for bee monitoring. In many cases, for the HOC/HOG approach, it is too sensitive to color changes of the background. For filter bank approach, it requires further performance optimization in order to get a reasonable run time. Although the practice did not exactly follow the milestone I set in the proposal, I addressed most of the problems in this report. The project turned out to be not as easy as I thought. A lot of time was spent on testing some engineering decision. The real life data has greatly challenged my understanding of the theory and design, and I have learned a great deal from this project.

Acknowledgments

I would like to thanks Professor Serge Belongie for his patient advice and all his helps in this project. Also I would like to thank the CSE190a classmates' constructive discussion.

References

[1] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, INRIA Rhˆ one-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, pp. 886–893. [Online]. Available: http://lear.inrialpes.fr/pubs/2005/DT05

[3] S.Bileschi, L.Wolf, "Image representations beyond histograms of gradients: The role of Gestalt descriptors", CVPR, 2007.

[4] Ecological image databases: From the webcam to the researcher, John Porter, Chau-Chin Lin, David E. Smith, Sheng-Shan Lu (2009)

[5] Sensor Archiving Gallery of Shanping Workstation http://srb2.tfri.gov.tw/wsn/