Automatic Heliothis Zea Classification Using Image Analysis

Tom Patten, Wenjing Li and George Bebis
Computer Vision Laboratory
University of Nevada, Reno, NV
(patten,wjli,bebis)@cs.unr.edu

Michael Freeman
Verdia Inc.
Redwood City, CA
michael.freeman@verdiainc.com

Abstract

In this paper, we present the design and implementation of an image analysis system for the automatic analysis of Heliothis zea insect images. The Heliothis zea is a corn earworm eating corn crops. Biotech researchers are interested in developing insecticidal bio-toxins with the best performance to kill or stunt the growth of this insects. The automated analysis of Heliothis zea images is imperative for fast and efficient biotech experiments. The proposed system consists of three stages: (i) insect segmentation, (ii) region processing, and (iii) instar and life classification. A probabilistic model (PM) based on mixtures of Gaussians has shown better performance for segmenting the insect images. And a back-propagation neural network (NN) has shown better performance for classifying the insect instar stage. The proposed system has been evaluated on real data using a five-fold cross validation procedure.

1 Introduction

With the increasing use of image analysis algorithms for rapid analysis of experimental data in biological experiments, it has become imperative to be able to analyze the resulting data in an expeditious and reliable manner. The relative paucity of automated techniques for reliable and rapid analysis of biological data has proved to be the rate-limiting step or bottleneck in most high-throughput experiments. Most of the experimental data in biological experiments can be acquired and represented in the form of images. Thus, automated analysis of such data entails the design and implementation of appropriate image analysis algorithms. In this paper, we present the design and implementation of an image analysis system for the automated classification and analysis of insect images.

1.1 Heliothis zea insect

The Heliothis zea, also called the corn earworm, is an insect which is a pest to corn crops. Biotech companies are interested in developing insecticidal proteins to fight pests. Usually, they inject experimental proteins into plants. As the plant grows, the protein becomes part of the plant's tissue. Insects which feed off the plant get a dose of the protein. The protein causes the infected insects to die, or stunts their growth, preventing them from reproducing. A biotech insecticidal protein, therefore, reduces the Heliothis zea population, which in turn reduces the damage to the corn crops.

A successful biotech insecticidal protein should meet certain criteria. It should reduce damage to crops. Also, it must be non-toxic to mammals and birds, because this same protein will be present in the food which comes to the market place. A successful biotech product has some attractive benefits. Crops which use biotech products will not need to be treated with as much chemical insecticide. Chemical insecticides have a negative side effect called fungal toxins, so using less insecticide results in less damage from fungal toxins. Crops raised using biotech products will benefit in these ways.

1.2 Bio-toxin analysis and scoring

To investigate the performance of the insecticidal protein, biotech researchers need to test multiple levels of protein with different concentration. Usually, they perform the testing in assay plates. One assay plate contains a large number of tiny wells. Each well is approximately two centimeters in diameter. Inside each well is a mixture of wheat germ mixed with concentrated a biotech insecticidal protein. These two ingredients are mixed together thoroughly. A set of Heliothis zea insects is then placed into each well. The larvae are delivered on a type of silk, similar to spider silk, which is hard for the lab technician to handle, so they cannot control the quantity of insects per well, but it is usually between one and ten insects. The experiment is run by letting the insects eat the food with the protein and grow over a four-day period. At the end of the experiment, the growth rate of the insects is measured for each well. When they have quantified the growth rate of a sufficient number of sample wells, they can quantify the rating of the biotech insecticidal protein at the concentration which was tested. Fig. 1 shows a 96-well assay plate after a four-day growth period.

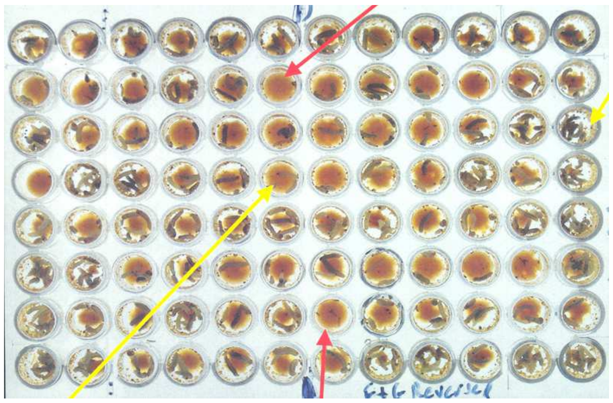


Figure 1: An assay plate with 96 wells.

Entomologists classify insect growth activity into stages. At infancy a Heliothis zea insect is in its first larval instar stage. About twenty-four hours later the Heliothis zea will shed its skin, and enter a second larval instar stage. At the second larval instar stage the insect body size and the size of its spots are distinctly larger than they were in the first larval instar stage. After yet another twenty-four hours the Heliothis zea insect will again shed its skin and enter the third larval instar stage. Heliothis zea insects in the third larval instar stage are considerably large compared to the second instar stage, and as before their spot sizes are also distinctly larger. At the end of the four day experiment the Heliothis zea would normally have reached an instar stage no greater than the third instar stage. The larval instar stage classification is the most important measure which is used to quantify growth rate of insects in the experiments. After a four-day period experiment, an insect which undergoes normal growth (and is not affected by the biotech insecticidal proteins) will have grown to the third instar stage. When the well contains a high degree of biotech insecticidal protein, the insects generally die in the first instar stage. Also the insects may get stunted, or die in the second instar stage, depending on the concentration of the protein.

By monitoring the growth activities of the insects in the assay plate, including number of insects in each well, the instar stages of each insect, the life information(whether the insect is dead or alive), biotech researchers can score each well in the assay plate according to the concentration levels of the protein in that well. Based on these scores, the standard industry measurement, called EC50, can be computed. EC50, or Effective Concentration 50%, is computed for each biotech insecticidal protein. It is a standard form of measurement of biotech insecticidal protein potency.

1.3 Why using image analysis

Currently, the scoring of the Heliothis zea images is done by experienced biotech experts manually. They watch each image of the well and use their vision to perform the analysis. There are usually 5000 images to be analyzed weekly. Therefore, it is imperative to automate this process to reduce the workload of the biotech researchers. The objective of our work is to design and implement algorithms for the automatic classification of the insects' life and instar stages. We do not deal with automating the scoring process in this paper.

The proposed system consists of three stages: (i) insect segmentation from background, (ii) region processing, and (iii) instar and life classification. Two models have been implemented and compared in the segmentation and classification stages. The first is a back-propagation NN, while the second is a PM based on mixture of Gaussians. We have evaluated each model using real images and a five-fold cross validation procedure.

The paper is organized as follows. In Section 2, a literature review of related work is presented. Section 3 deals with the insect segmentation problem. Region processing and instar and life classification are introduced in Sections 4 and Section 5 respectively. The training of the NN and PM are presented in Sections 6 and 7. Experimental results are presented in Section 8. Finally, our discussion of results and conclusions are given in Section 9.

2 Review of Previous Work

There have been several efforts to automate the analysis of image data arising from biological or medical experiments [16, 3, 4, 1, 18, 2]. Recent research has focused on either NNs [8, 10, 12, 4, 13, 18] or PMs [14, 6, 9, 2].

Ghosh and Chinnaiyan [6] have proposed a statistical model based on mixture of Gaussians in the context of hierarchical clustering of gene expression data from DNA microarray experiments. Jain et al. [9] describe a program called UCSF Spot for fully automatic quantification of DNA microarrays. The program automatically locates both, subarray grids and individual spots while requiring no user identification of any image coordinates. Manduchi et al. [14] describe a protocol by which discrete values are used to provide an easily interpretable description of differential expression. Novel statistical methods are proposed to attach confidence levels to the hypothesis that changes in expression levels represents true changes.

In [2], the design and implementation of a computer vision system called DNAScan was presented for the automated analysis of DNA hybridization images. A recursive segmentation procedure was designed and implemented to extract spot-like features in hybridization images in the presence of a highly inhomogeneous background. Positive hybridization signals were extracted from the spot-like features using grouping and decomposition algorithms based on computational geometry. A mathematical model for the positive hybridization patterns and a Bayesian pattern classifier based on the shape-based moments were proposed and implemented to distinguish between the clone-probe hybridization signals.

Lerner [13] applied neural networks to automate the analysis of chromosome images, in all aspects of the analysis including segmentation, feature description, selection and extraction, and classification. He used a new approach called a classification-driven partially occluded object segmentation (CPOOS) to separate clusters of touching chromosomes. And a multi-layer perceptron (MLP) was used to score and verify hypotheses. Such a NN based system classified 5500 chromosomes with a success rate of 83.6%.

In [4], Cheng et al. used a competitive Hopfield neural network for segmenting grey scale medical images. It is a kind of Hopfield network that incorporates the winner-takes-all (WTA) learning mechanism. The image segmentation is conceptually formulated as a problem of pixel clustering based upon the global information of the gray level distribution. Patel et al. [18] used a neural network for automating the process of grading eggs. They reported an accuracy of 92.8% for blood spots, 85% for dirt and stain detection, and 87.8% for crack detection. The features used by Patel et al. were formed by computing a histogram of the RGB color space of each image and using the counts from each histogram bin as a feature. They used 768 (256*3) histogram bins corresponding to 768 nodes in the input layer of the neural network.

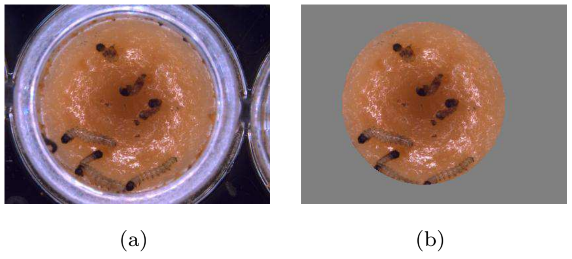

(a) (b)

Figure 2: (a) a typical Heliothis zea image, (b) Hough transform result.

3 Insect Segmentation

Fig. 2(a) shows a typical Heliothis zea insect image. It is a 600*800 color image captured by a digital camera mounted on a robot arm. All the images have one thing in common which is the circular shaped well in which the experiment is taking place. The insects are located inside the well with the colored background which is the biotech insecticidal protein. Our first goal was to segment all the insects from the background. Here, we propose to use a segmentation framework with two phases: training and segmentation. The flow chart is shown in Fig. 3.

3.1 Feature selection

Since the area of interest of the insect image is the area inside the circular well, the first step is to detect the rim of the well and remove the well from the image. Here, the Hough Transform for circle detection was used to extract the circle underlying the rim of the well. Once the well was detected, we removes it from the image. Fig. 2(b) shows the results of the Hough transform applied to the image in (a). Because the camera may not be exactly perpendicular to the assay plate during shooting, we might not get perfect results for every image, however, the results are very satisfactory.

Feature selection is always a crucial problem in pattern classification. The basic rule in feature selection is to choose features that would result in large between-class distance and small within-class variance in the feature vector space. To judge if a pixel in the image belongs to the insect or not, we need to use multiple criteria for each pixel. Our feature selection method is based on a small window or sub-image centered on that pixel. In this way, we compute many features based on this window. After careful investigation and testing through experiments, we adopted the following five features:

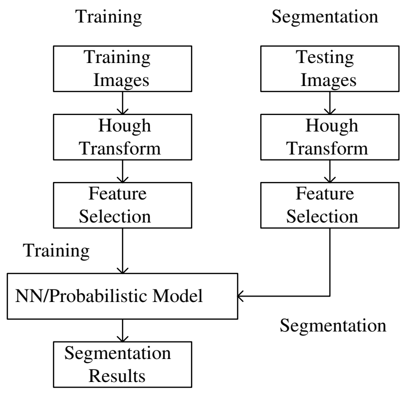


Figure 3: Flow-chart of the segmentation stage.

- Average hue (hue is a numerical representation of a color, e.g. yellow=1.57)
- Average saturation (grey has saturation 0.0, and vivid colors like red, yellow, blue have 1.0)
- Grey-scale percentage (measured by saturation and a threshold)
- Intensity variance
- Count of spots (if the windows contains something that looks like a spot)

Here we convert the image from RGB color space to HSV space [15, 19] to compute the average hue value and the average saturation value.

4 Region Processing

Based on the segmented results of the insect images, in this stage, we apply image processing to refine the segmented regions. Goatsias et al [7] proposed morphological operations as a method of merging pixels into sets of pixels that share the same properties based on their densities and proximities. We use two morphological operations, "morphological opening" and "morphological closing." Each of these operations consists of two processes: dilation and erosion.

The goal of morphological operations is to convert areas of the image which consists of dense points of foreground into solid regions each of which represent the insects. The morphological opening includes two steps, dilation followed by erosion. After morphological opening, the next step is morphological closing, the reverse of morphological opening. It is the same as morphological opening, except the erosion and dilation steps are reversed. The purpose of morphological closing is that the erosion step will eliminate tiny false positive regions which are too small to be considered insect.

After the above morphological operations, we can build our lists of regions in each image. Smaller regions which include less than 1,000 pixels are filtered out. These regions are considered as the noise or peeled skin of the insects. We have also applied the snake algorithm [11] to find a better boundary for each region. Fig. 4 shows the improved boundaries of the insect regions.

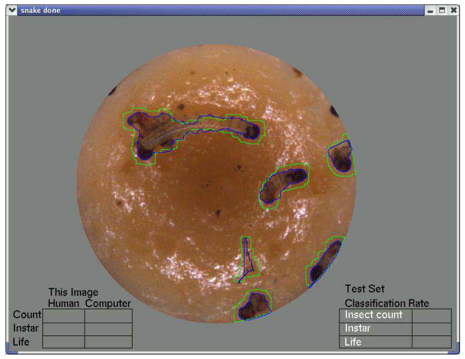


Figure 4: Improved boundaries using snake algorithm.

5 Instar and Life Classification

5.1 Insect spot detection

We start by visiting each pixel in the region. We use the Breadth-First-Search (BFS) algorithm to grow sub-regions, such that each sub-region contains sets of pixels with similar intensity, or brightness. We continue in this way until each region pixel is assigned to one sub-region. This will partition the set of region pixels into pair-wise disjoint subsets, each subset being a sub-region of an image region.

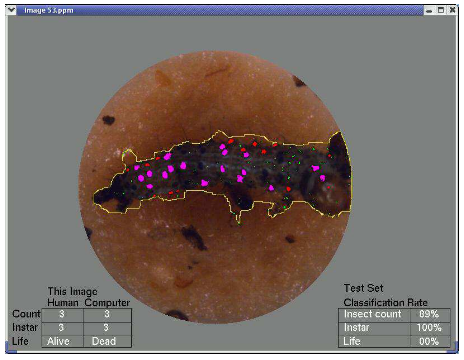
From the sub-regions, we next try to find candidate sub-regions which might contain insect spots. We do this by visiting each sub-region, then visiting its surrounding field of pixels, and evaluating the degree of contrast between the region and its field. When we find sub-regions which are candidate spots, we append all of the sub-region pixels onto a large list of pixels. We use a queue data structure for this list.

We can find the spots by their darkness, or intensity values. Each pixel has an elevation corresponding to its darkness. The dark pixels on the spots form hills. We can find the spot regions by finding these hills. At the present point in processing, we have a list which holds most of the pixels which comprise the hills. We take a pixel off the queue and pass it as a seed to BFS. BFS grows an area of contiguous pixels, and returns the result as a list of pixels. We traverse the list to find its darkest pixel. This pixel is the "top of a hill". We pass this "top of the hill" pixel again as a seed to BFS, and this time the BFS procedure enforces the constraint that as grows the spot, it is not allowed to go uphill. This is exactly how we identify two spots close to each other as two distinct spots.

The resulting list returned by BFS is now a candidate spot, and we put it on a master list of spot regions. We also mark these pixels on our mapped image as "spot", We return to our original list of spot candidate pixels, and retrieve the next pixel. Pixels on this queue already marked on our mapped image as spot are disregarded, and we move on to the next pixel on the list. When we find a pixel not yet marked as a spot pixel, we repeat the procedure described above. When we have processed the entire list of candidate spot pixels in this way, we will have a list of spots on an insect region.

We can validate spots for roundness. First, we find the size of the min-max box around the spot. Then we compute the relation between the count of spot pixels, and the count of pixels in the min-max box. A perfectly round circle will have the relation $circle/square = \pi/4$. Of course, we do not look for perfectly round spots. We use a roundness threshold, which has the effect of filtering out regions which are not round enough to be spots. This allows us to consider only round spots for classification purposes.

The most important goal of the classifier is to identify the larval instar stage of the insects. An interesting thing that we have noticed about the insects is, that even though the insects grow rapidly, the size of their spots seem to remain consistent within each instar stage. One possible explanation of this may come from the fact that they shed their skin each time they change instar stages. We can use the pixel count of the spots as a measure of spot size. We use thresholds for size classifications which distinguish between the classes. We have defined three spot size classifications: small, medium, large. This size gives us a good feature for correctly classifying the instar stage of the insects. Fig. 5 shows an example of the spot detection (i.e., the green colored spots are classified as small; the red spots are classified as medium; the magenta colored spots are classified as large).



This Image

	Human	Computer
Count	3	3
Instar	3	3
Life	Alive	Dead

Test Set
Classification Rate

Insect count	89%
Instar	100%
Life	00%

Figure 5: Spot detection (third instar).

5.2 Feature selection

With the master list of spots completed, we are ready to extract features. We define one feature vector for each region. For each region, we look for the most useful features to extract, and compute the region features. After experimentation, we decided to use the following features:

- count of small spots
- count of medium spots
- count of large spots
- ratio of spot pixels to region pixels
- skin brightness near the spots

The first three features are useful for instar classification, and the last two features are intended to be useful for life classification, however, we use all five features for performing each classification task. As in the segmentation classification task, our feature vectors are five dimensional.

6 PM Training

Mixture models are a type of density model which comprises a number of component functions, usually Gaussian. These component functions are combined to provide a multi-modal density. They have been employed to model the color distribution of objects for real-time segmentation and tracking [15]. The task can be made more robust by generating a mixture model corresponding to background colors in addition to foreground model and employing Bayes' rule to perform pixel classification.

Let the conditional density for the sample data ξ belonging to a multi-colored object O be a mixture with M component densities:

$$p(\xi|O) = \sum_{j=1}^{M} p(\xi|j)\pi(j) \quad (1)$$

where a mixing parameter $\pi(j)$ corresponds to the prior probability that data ξ was generated by component j and where $\sum_{j=1}^{M} \pi(j) = 1$. Each mixture component is a Gaussian with mean μ and covariance matrix Σ, i.e. in the case of a N dimensional space:

$$p(\xi|j) = \frac{1}{(2\pi)^{\frac{1}{2}} |\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(\xi-\mu_j)^T \Sigma_j^{-1}(\xi-\mu_j)} \quad (2)$$

Expectation-Maximization (EM) algorithm provides an effective maximum-likelihood algorithm for fitting such a mixture model to a set of training data [17]. The EM algorithm is iterative with the mixture parameters being updated. It monotonically increases the likelihood in each iteration, converging to a local maximum. Mixture models provide greater flexibility and precision in modelling the underlying statistics of sample data. Once a model is generated, posterior probabilities can be computed according to the Bayes' rule. Given density estimates for both an object O,and the background S,the probability that a sample data, ξ belongs to the object is given by the posterior probability $P(O|xi)$:

$$P(O|\xi) = \frac{p(\xi|O)P(O)}{p(\xi|O)P(O) + p(\xi|S)P(S)} \quad (3)$$

7 NN Training

Instead of explicitly estimating the *pdf* of the data, NNs learn the classification boundary from a set of examples. NNs are computational systems inspired by the structure, processing methods, and learning ability of a biological brain [5].

In general, NNs implement a nonlinear mapping of the from $u = G(x)$. The mapping function G is established during a training phase where the network learns to correctly associate input patterns x to output patterns u (i.e., supervised learning). During training, their free parameters (i.e., weights and biases) are adjusted in a systematic way so as to minimize a cost function. Typically, the cost function is defined on the basis of the mean-square error between a desired network response (i.e., u) and the actual network output. In the context of classification, NNs can learn highly non-linear decision boundaries, without explicitly estimating the probability distribution of the data.

In this paper, we considered two-layer feed-forward NNs (see Fig. 6) with sigmoidal activation functions, trained by back-propagation, a popular learning algorithm that uses gradient descent to adjust the network weights and biases [5].

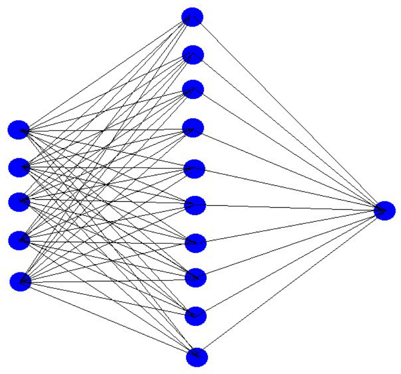


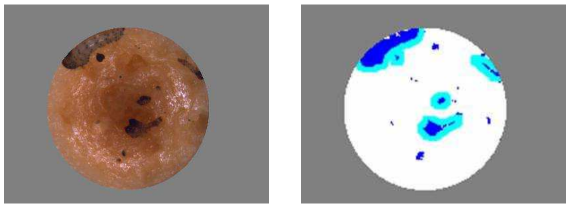
Figure 6: Topology of the neural network model.

8 Experimental Results

A set of 88 images provided by Verdia Corporation were used in our experiments. We used a five-fold cross validation procedure to evaluate the accuracy statistically. In detail, we randomly divided the whole set of images into three subsets. Subset 1 is called the training data set consisting of 50% of the images, which were used to train both the neural network model and the probabilistic model. Subset 2 is the validation set consisting of 25% of the images, which were used for bootstrapping (see below). Subset 3 is the test set consisting of the last 25% of the images, which was used to test the performance of the classifiers. We do the partition of the three sub-sets five times, and we compute the average performance of the classifiers.

8.1 Results for image segmentation

For the insect image segmentation, we first get the ground truth images for each insect image by hand using a graphics program. The set of the ground truth images were used to measure the accuracy of the proposed segmentation methods. We estimate segmentation accuracy as follows:



(a) (b)

Figure 7: (a) original image (b) segmented image.

$$accuracy = \frac{2 * Seg_i}{(Seg_n + Seg_h)} \quad (4)$$

where Seg_h is the number of insect pixels per ground truth image. And Seg_n is the number of insect pixels segmented by the classifier. Seg_i is the intersection of Seg_h and Seg_n.

To reduce the number of false segmentations, we used bootstrapping [5]. Specifically, after training a particular classifier (i.e., NN or PM) using the initial training set, we used the classifier to classify each pixel in the validation set as background or foreground. After classifying each pixel, the ground truth data set was consulted to see if the network made any classification error. We chose 10% from the pixels which were misclassified to generate new windows of data and augment the training data. This process was repeated until the number of false positives dropped below a given threshold.

Table 1 summarizes our segmentation results. The columns "S1" to "S5" indicate the five-fold partition of the images. The accuracy of each classifier on the test set for each partition is shown on each column. The last column shows the average accuracy for the different classifiers. Our results indicate that the PM performs better than the NN for this task. Fig. 7(b) shows the segmented results of the image shown in in Fig. 7(a).

Table 1: Segmentation results.

Model	S1	S2	S3	S4	S5	Avg
NN	90%	58%	90%	91%	87%	83%
PM	89%	90%	89%	90%	86%	89%

8.2 Results for instar and life classification

Tables 2 and 3 summarize the results for the NN and PM classifiers for instar and life classification. In each table, the second column indicates the accuracy for insect counting. We got 77.4% and 84% accuracy on average. There are several reasons for this relatively low accuracy. First, the insect images have much noise, such as the skin and the head cap peeled off from the insects. Second, the insects may overlap with each other. This is the most important factor affecting the accuracy of the insect counting. Currently, the system does not handle overlapped insect regions.

The third column of each table indicates the accuracy for the instar classification. We got 95.2% accuracy on average using NNs and 62% using PMs. The main reason that the PM did not perform as well for instar classification is due to the limited number of training examples. The last column shows the accuracy for the life classification. We believe that we did not get good accuracy for the life classification due to several reasons. First, life classification is difficult. The insects assay plate was frozen before taking pictures (i.e., otherwise the insects would climb out of the well during the shooting). The frozen process might have changed the skin color of the insects. It is even hard for humans to make this decision correctly when the insects are dying. Second, the feature we used to train the classifier might not have been optimal. In any case, compared to the instar classification, life classification plays less important role in the final scoring system to compute the EC50.

Table 2: Classification results using NNs.

Partition	Insect count	Instar	Life
S1	74%	97%	67%
S2	78%	95%	69%
S3	77%	93%	56%
S4	78%	96%	67%
S5	80%	95%	71%
Avg	77.4%	95.2%	66%

9 Conclusions

We presented an image analysis system for processing Heliothis zea insect images. The system includes three main stages: insect segmentation, region processing, and instar and life classification. Two mod-

Table 3: Classification results using PMs.

Partition	Insect count	Instar	Life
S1	81%	59%	69%
S2	89%	83%	63%
S3	83%	67%	60%
S4	85%	29%	80%
S5	83%	75%	71%
Avg	84%	62%	69%

els were implemented and compared in the segmentation and classification stages respectively: a backpropagation NN, and a PM based on mixture of Gaussians. Our results indicate that the PM performs better for the segmentation task while the neural network model performs better for instar classification which is most important for computing the EC50.

Although we demonstrated our system for the case of the Heliothis zea insect, it could be extended to handle other types of insects such as the Spodoptera exigua (beet armyworm) or the Trichoplusia ni (cabbage looper). Future work includes resolving the cases where insects overlap with each other for more accurate classification, optimizing feature selection, and employning more powerful classifier (e.g., Support Vector Machines).

References

[1] G. Agam and I. Dinstein. Geometric separation of partially overlapping nonrigid objects applied to automatic chromosome classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1212–1222, 1997.

[2] Suchendra M. Bhandarkar, Tongzhang Jiang, Nan Li, and Kunal Verma. Automated analysis of dna hybridization images for high-throughput genomics. *submitted to machine vision and applications*, 2004.

[3] A. Carothers and J. Piper. Computer-aided classification of human chromosomes: a review. *Stat. Comput.*, 4:161–171, 1994.

[4] Kue-Sheng Cheng, Jzau-Sheng Lin, and Chi-Wu Mao. The application of competitive hopfield neural network to medical image segmentation. *IEEE Transactions on Medical Imaging*, 15(4):560–566, Aug. 1996.

[5] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* John Wiley, 2 edition, 2001. 20-29.

[6] D. Ghosh and A.L. Chinnaiyan. Mixture modeling of gene expression data from microarray experiments. *Bioinformatics*, 18:275–286, 2002.

[7] John Goatsias and K. Sivakumar. Morphological transformations of image sequences. Master's thesis, Dept. of Electrical and Computer Engineering, John Hopkins University.

[8] J. Graham, P. Errington, and A. Jennings. A neural network chromosome classifier. *Journal of Radiat. Res.*, 33:250–257, 1992.

[9] A.N. Jain, T.A. Tokuyasu, A.M. Snijders, R. Segraves, D.G. Albertson, and D. Pinkel. Fully automatic quantification of microarray image data. *Genome Research*, 12:325–332, 2002.

[10] W.P. Sweeney Jr., M.T. Musavi, and J.N. Guidi. Classification of chromosomes using a probabilistic neural network. *Cytometry*, 16:17–24, 1994.

[11] M. Kass, A. Witkin, and D. Tezopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[12] B. Lerner, H. Guterman, I. Dinstein, and Y. Romen. Human chromosome classification using multilayer perceptron neural network. *International Journal of Neural Systems*, 6:359–370, 1995.

[13] Boaz Lerner. Toward a completely automatic neural-network-based human chromosome analysis. *IEEE Transactions on Systems, Man and Cybernetics*, 28(4):544–552, Aug. 1998.

[14] E. Manduchi, G.R. Grant, S.E. McKenzie, G.C. Overton, S. Surrey, and C.J. Stoeckert. Generations of pattern from gene expression data by assigning confidence to differentially expressed genes. *Bioinformatics*, 16(8):685–698, 2000.

[15] Stephen J. McKenna, Yogesh Raja, and Shaogang Gong. Tracking color objects using adaptive mixture models. *Image and vision computing*, 17:225–231, 1999.

[16] Fernand Meyer. Automatic screening of cytological specimens. *Computer Vision, Graphics and Image Processing*, 35:356–369, 1986.

[17] Todd K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, Nov. 1996.

[18] V.C. Patel, R.W. McClendon, and J.W. Goodrum. Color computer vision and artificial neural networks for the detection of defects in poultry eggs. *Artificial Intelligence Review*, 12:163–176, 1998.

[19] Shamik Sura, Gang Qian, and Sakti Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. *Proceedings of IEEE International Conference on Image Processing*, pages 589–592, Jun. 2002.