

An automatic algorithm for detection of infestations in X-ray images of agricultural products

R. P. Haff · T. C. Pearson

Received: 4 June 2007 / Accepted: 26 June 2007 / Published online: 18 July 2007
© Springer Science+Business Media, LLC 2007

Abstract An automatic recognition algorithm was developed and tested for detection of certain defects or contaminants in X-ray images of agricultural commodities. Testing of the algorithm on wheat kernels infested with larvae of the granary weevil, *Sitophilus granarius* (*L.*) yielded comparable results to those obtained by human subjects evaluating digitized X-ray film images (14.4% overall error vs. 15.6% for human subjects). Further tests on X-ray images of olives infested with the Olive Fly, *Bactrocera oleae* (*L.*), yielded a total error of 12% for large infestations and over 50% for the smallest infestations with false positive results below 10%. Testing of alternate training strategies showed that for this type of algorithm, which uses a form of discriminant analysis with a generally “fuzzy” decision boundary, best results are obtained when training with samples that map far away from the boundary, then applying the derived decision function to all samples to be classified.

Keywords X-ray · Imaging · Detection · Infestations · Sorting

Introduction

X-ray imaging has become a well-established method for the inspection of certain agricultural products for defects and contaminants. In particular, many packaged products are inspected using linescan X-ray equipment for the presence of metal, plastic, or bone, as these are high-density targets and easily recognized in an X-ray image. For many other detection problems in agriculture, X-ray imaging has not been well established in the processing plant environment. Many contaminants or defects, such as insect infestation, are much more difficult to detect with X-ray imaging due to the small size of the object of interest and the small difference in X-ray density between the target material and its surroundings.

Detection of insects, as well as other defects, is a high priority for the food industry for quality assurance and food safety reasons. Beyond the direct loss of product, stringent import controls by many countries aimed at preventing the introduction of alien species make insect infestation a potential threat to U.S. export markets. Wheat producers alone in the U.S. can lose as much as one billion dollars in a bad year [1, 2]. Insect infestation is a major component of USDA grading standards, but only account for live insects and do not address the problem of insects growing inside the kernels. Since many of these defects are internal and invisible to most detection methods, X-ray imaging is often the only feasible recourse. X-ray examination has been the official method of the Association of Analytical Chemists (AOAC) for inspection for internal insect infestation in grain or seeds since 1961 [3]. Consequently, improving X-ray technology for the purpose of food inspection has been an area of active research for many years. Within this field of research there are generally two active areas. The first seeks to improve X-ray equipment to generate images

R. P. Haff (✉)
USDA ARS WRRC, 800 Buchanan St., Albany, CA 94710,
USA
e-mail: ron@pw.usda.gov

T. C. Pearson
USDA ARS GMPRC, 1515 College Ave., Manhattan, KS
66502, USA
e-mail: thomas.pearson@gmprc.ksu.edu

of higher resolution and lower noise, so that little areas of small density differences can be detected. The evolution of high resolution and low noise CCD arrays for the digital camera market has allowed a great deal of progress in this area, as CCD chips are commonly used as detectors in X-ray equipment. Second is the development of algorithms to automatically identify features of interest within the X-ray images for sorting purposes. This area of research is vital, as any detection method must operate at the high speeds typical of a food processing plant environment.

There has been some work reported on algorithms to detect insect infestation in X-ray images of wheat kernels. Keagy and Schatzki convoluted masks mimicking larvae with X-ray images of infested wheat kernels and were able to identify large infestations [4]. Karunakaran et al. reported a digital X-ray system, including an automatic classification algorithm, for the inspection of wheat for rice weevils that was over 95% accurate [5]. The algorithm was limited by the fact that the images used were acquired with a fine focus X-ray microscope arrangement, yielding high quality images but impractical for operating at high speed. Haff and Slaughter used Bayesian classifiers to detect Granary weevil infestations in wheat kernels [6]. This method was up to 98% effective for larger larvae but could not reliably detect smaller larvae. Other work has been reported to develop algorithms to identify defects in X-ray images of a variety of crops, including watercore damage in apples [7], insect damage in almonds [8], and Naval Orange Worm in pistachios [9].

Discriminant analysis using Bayesian classifiers can be well suited for high speed image analysis, depending on the number of features to be extracted from the images and the computation required deriving those features. The general principle involves mapping features derived from the images into “feature space” and computing a decision boundary in that feature space that separates the classes of images with minimum error. One objective is to identify features that can be extracted with minimal computation. Another is to restrict the number of features to as small a set as possible while still retaining the essential information that allows proper separation into classes. The form of the derived discriminant function is partially dependant on the *a priori* probability of a kernel being infested in the first place. Manipulation of the *a priori* probability can be used to influence the sensitivity of an algorithm, allowing control over the amount of non-infested kernels falsely identified as infested (false positives) at the expense of decreasing the amount of infested kernels correctly classified.

There has been a significant amount of work reported in using statistical approaches in selecting features for use in detection algorithms, although for the most part this work applies to selection of spectral bands from digitized spec-

tra, primarily for NIR or hyperspectral inspection. One approach is to use stepwise discriminant analysis to select a small number of spectral bands from generalized spectra [10]. Genetic algorithms have been developed in recent years which select features using improvements to randomly selected subsets of features [11, 12, 13]. Principle component analysis (PCA) has been used to select spectral bands with high discriminating power [14, 15]. While these statistical methods are useful in selecting optimal spectral bands from spectra, they do not address the problem of extracting features from images. They are therefore suited to spectroscopic methods of inspection and sorting, such as with NIR and hyperspectral analysis, but not helpful in detecting defects or contaminants in X-ray images.

The objective of this research was to develop a computer algorithm based on discriminant analysis that can detect defects or contaminants in X-ray images of agricultural commodities. The algorithm was to be tested on existing databases of X-ray images of wheat kernels infested with larvae of the granary weevil, and on images of olives infested with the Olive Fly. Results should compare favorably with those obtained by human subjects viewing the same digitized images. Finally, training strategies meant to deal with the inherent fuzziness at the boundary between classes that is inherent in the X-ray images should be conceived and tested.

Materials and methods

Algorithm

The algorithm extracts an arbitrary number of features from each X-ray image for use in an iterative discriminant analysis routine, which tests each combination of an arbitrary number of the features for the best performance in distinguishing the classes to be sorted. For real-time sorting, the number of features extracted is mostly dependant on the available time for image processing. Generally, using more features yields better results up to a point, beyond which performance diminishes. For this research, 64 features were extracted from each image, and the discriminant analysis routine was used to test every possible combination of three features. From this best combination of features, the routine computes a boundary condition of the form shown in equation 1 for use in real-time sorting:

$$AF_1 + BF_2 + CF_3 \geq 0 ? \quad (1)$$

where F_1 , F_2 , and F_3 are the numerical values of the three features extracted and A, B, and C are the decision boundary coefficients. For each sample to be sorted, if equation 1 is true then the sample is classified in a certain group, otherwise it is classified in the other.

Creating a sorting algorithm using discriminant analysis requires a large number of X-ray images of samples from each class. In this research, the two classes are infested and non-infested (clean). The images were separated into two sets, one for training and the other for testing, or validation. Half of the images from each class were assigned to the training set and half to the validation set. The images in the training set are used to compute the boundary condition, which is then used to classify each image in the validation set. Since the true condition of each validation set image (infested or clean) is known, this allows determination of the accuracy of the computed decision boundary. This process was repeated using each possible combination of three out of the 64 chosen features to determine which combination gives the best sorting accuracy. Once this “optimal” decision boundary is determined, all that is required for classifying samples in real-time sorting is to extract the three chosen features and apply the decision boundary. The procedure for selecting the best combination of features is outlined in the shaded portion of the flow chart shown in Fig. 1.

The feature selection portion of the algorithm is derivative based and maps the differences in pixel values in the image to a set of 64 numbers, or features, each of which represents the number of instances in the image where a particular derivative (range) occurs in a given direction. First, the step was defined as the distance between pixels over which each derivative was computed and the spread as the distribution of levels into which the gray scale (0–255) was divided for the mapping. For each pixel location in the image, excluding a border region consisting of the number of pixels in the step, the derivative was computed in eight directions corresponding to the eight points of the compass. The eight derivatives are defined in Table 1, where $I_{x, y}$

the pixel intensity at pixel location (x, y) and ss (short step) is defined as $step * \cos(45) = step * 0.7071$. Taking the derivative in different directions is meant to compensate for differences in images that occur because of the orientation of either the commodity or the direction of insect tunnels. It is expected that the best results would be obtained as the number of directions tested increases. However, as the number of directions increases so does the size of the feature histogram and the computational requirements in deriving the algorithm.

The step defines the radius of a circle around the current pixel location, and the derivatives are the differences in pixel intensities between the center of the circle and pixel locations on the circumference at angles separated by 45 degrees. The spread defines the segmentation of the gray scale between 0 and 255 for eight bit applications (Table 2).

The derivative directions (Table 1) and the derivative magnitudes (Table 2) form the axis of an 8×8 feature histogram (Fig. 2). For each pixel location in the image (excluding the edges at a depth equal to the step) there are eight derivative values, thus eight of the bins in the feature histogram are incremented. The total number of entries (Σ) in the histogram should therefore be:

$$\Sigma = 8(H - 2s)(W - 2s) \quad (2)$$

where H is the image height in pixels, W is the image width in pixels, and s is the step size. The example in Fig. 2 represents a feature histogram for a 90×130 image with a step of 5. The sum of the entries should therefore be 76,800. Note that for 8 bit images the maximum spread is 36, so that $7 * \text{spread} < 255$.

Image generation, wheat

Grain infested with Granary Weevils was imaged on X-ray film using a Faxitron X-ray cabinet (20 KeV, 3 mA, 90 sec). Film images were used for creation of the algorithm because of their high quality. In applying the algorithm for high speed sorting, real-time digital X-ray equipment with a much shorter exposure time on the order of a few milliseconds would be used. The resulting film was inspected with a microscope (3x) to determine which kernels were infested. The infestations were classified by measuring the width of the tunnels with an objective micrometer and following established criteria [16]. Fifty of each life stage were selected as well as 450 non-infested images. A digital image of each selected kernel was created from the film with a film scanner (ScanPremio ST, Acer Peripherals, San Jose, CA, with Adobe Photoshop 5.5) at a resolution of 174 pixels per cm (57.6μ per pixel). The corresponding image size was 90 pixels by 130 pixels. The

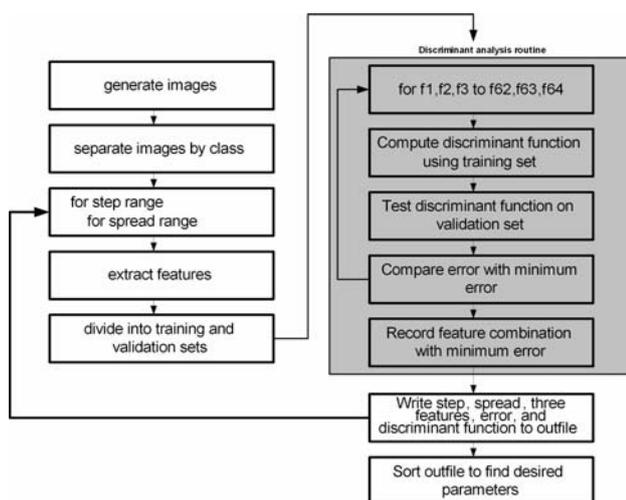


Fig. 1 Flow chart describing the selection of optimal parameters and features for sorting images by class

Table 1 Definitions of the eight derivatives that generate the extracted features

Derivative #	Direction (°) ^a	Value
1	0	$I_{x, y} - I_{x, y\text{-step}}$
2	45	$I_{x, y} - I_{x+ss, y-ss}$
3	90	$I_{x, y} - I_{x+step, y}$
4	135	$I_{x, y} - I_{x+ss, y+ss}$
5	180	$I_{x, y} - I_{x, y+step}$
6	225	$I_{x, y} - I_{x-ss, y+ss}$
7	270	$I_{x, y} - I_{x\text{-step}, y}$
8	315	$I_{x, y} - I_{x-ss, y-ss}$

^a 0°= Up (North)

Table 2 Division of eight bit gray scale for histogram binning

Derivative magnitude bin #	Derivative values (D)
1	$0 \leq D < \text{spread}$
2	$0 \leq D < 2^* \text{ spread}$
3	$2^* \text{ spread} \leq D < 3^* \text{ spread}$
4	$3^* \text{ spread} \leq D < 4^* \text{ spread}$
5	$4^* \text{ spread} \leq D < 5^* \text{ spread}$
6	$5^* \text{ spread} \leq D < 6^* \text{ spread}$
7	$6^* \text{ spread} \leq D < 7^* \text{ spread}$
8	$7^* \text{ spread} \leq D < 225$

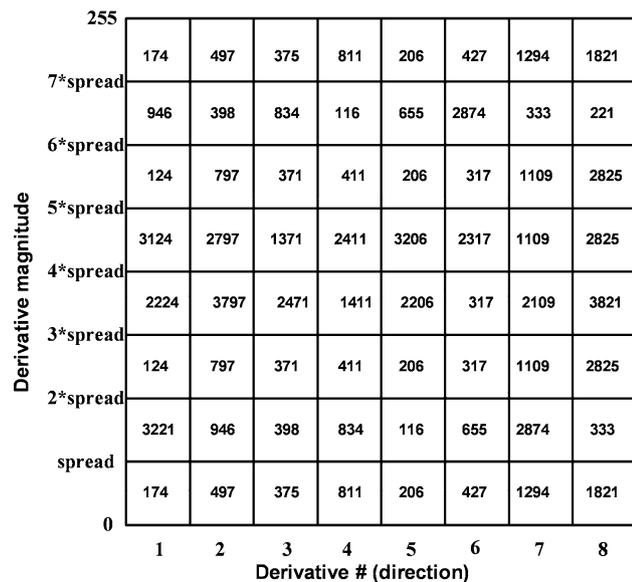


Fig. 2 Binning of derivative directions and magnitudes into the 64-bin feature histogram. Each instance of a particular derivative magnitude and direction increments the appropriate bin

data set thus consisted of 900 images, half of which were from infested kernels. Each image was subjected to a threshold so that the background was uniform with a pixel intensity of zero. The distribution of the images by life

stage of the insect in the infested kernels is shown in Table 3. Figure 3 shows sample images of each of the life stages, as well as images of non-infested kernels.

Image generation, olives

As a second trial for the algorithm, olives infested with the Olive Fly (*Bactrocera Oleae L.*) were collected from California olive orchards and imaged on film using an X-ray cabinet (43804N, Faxitron X-ray Corporation, Wheeling, IL) at 28 KeV, 3 mA for 180 sec. The film was visually inspected to determine which olives were infested. The damage due to infestation was subjectively classified as slight, moderate, or severe. Figure 4 shows sample X-ray images for each classification.

The data set consisted of 410 images, 249 of which were infested and 161 non-infested. Of the infested samples, 31 were classified as having slight damage, 46 moderate, and 172 severe. The uneven distribution of samples was a consequence of the limited number of samples available. A digital image of each olive was generated from the film as previously described at a resolution of 59 pixels per cm (169.5 μ /pixel). Images were divided between training and validation sets and the algorithm applied following the same methodology used for wheat kernels.

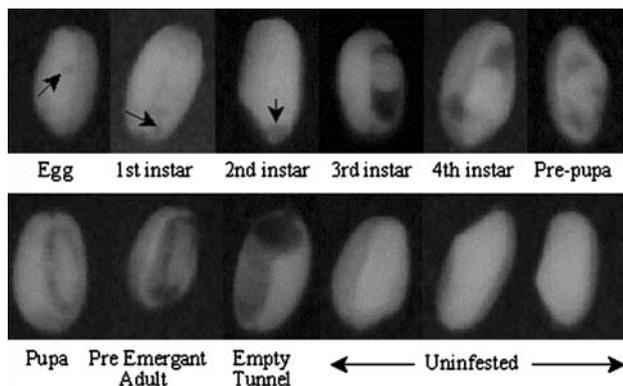
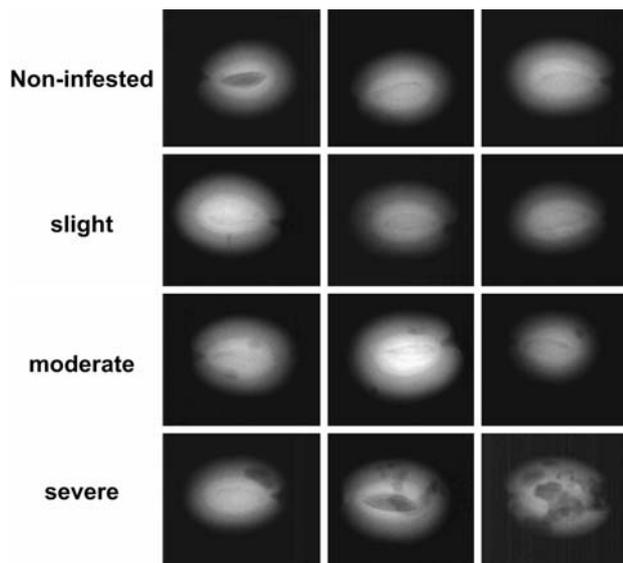
Algorithm testing, wheat

The images of wheat kernels were used to train the algorithm as previously described and the resulting discriminant functions were applied to the validation set to obtain classification error rates. Two different strategies were used for training the algorithm. The first strategy was to use all 900 images, thus including each life stage for both training and validation. Second, images of infested kernels were divided into two sets based on insect size. Eggs through the 3rd instar stage were classified as small, while the remainder were classified as large. With this strategy, only the large set was used for training. Separate training strategies were devised because in many cases the smaller infestations, particularly at the egg stages, are not discernible in the digital images as some loss of image quality occurs in the digitization process. Also, the original identification and classification of infestations was done under a microscope. The result is that the algorithm is presented with images from different classes that are not really different in certain cases, thus confusing the training process. One of the goals, therefore, is to determine the preferred training strategy.

Classification error rates were tabulated for the best combination of step, spread, and three optimal features for each of the training strategies described. Error rates were also compared with those obtained by human subjects

Table 3 Distribution of images by age of the insect

Life Stage	# of images	Life Stage	# of images
Egg	50	prepupa	50
1st instar	50	pupa	50
2nd instar	50	adult ^a	50
3rd instar	50	Tunnel ^b	50
4th instar	50	Tot infested	450
		Non-infested	450

^a pre-emergent^b post-emergent**Fig. 3** Digitized X-ray film images of infested and non-infested wheat kernels**Fig. 4** Sample X-ray images of olives with various levels of infestation damage caused by the olive fruit fly

inspecting film with a magnifying glass, the most common method currently used for quality control inspection [6]. When sorting the output file from the algorithm, it is possible to select the parameters (step, spread, features)

that yield the lowest overall error, the lowest false positive error (good product classified as bad) or the lowest false negative error (bad product missed). Which of these results is most desirable depends on a cost–benefit analysis of the product being sorted. Generally, the lowest false positive rate is desirable, as in real-time sorting the false positive percentage represents the portion of the product that is being discarded. However, if the defect or contaminant is of great concern, generally for food safety reasons, then the lowest false negative error rate would be desired. The objective here is mainly to detect insect infestations, which are generally a food quality issue. Therefore, for the sake of brevity, we report only the results that yield the lowest overall error rates and the lowest false positive error rates.

Scatter plots were generated in feature space showing the best three features as well as the decision boundary derived from the discriminant function for certain training instances. The decision boundary was derived by equating the discriminant functions for the two classes. In other words, the decision boundary consists of all points in feature space where the distance to the centroid (location of average image of a particular class) for each class is equal and defines a plane in three-dimensional space. The plots were rotated to give the appearance of looking along the edge of the decision plane, which therefore collapses into a line allowing a clear visualization of the separation of the two classes in feature space.

Algorithm testing, olives

Algorithm testing for the images of olives was conducted similarly to the procedure for the images of wheat kernels. For the case of olives, however, there is no previous work with which to compare results, either for machine recognition or human recognition. Training was conducted using all available images in the training set because the data set is much smaller than that for the wheat kernels.

Results and discussion

Wheat

Classification results for minimal false positive classification are shown in Table 4. The first column represents using all images in the training set for training, and applying the derived decision function to the entire validation set (first training strategy). In the second column, only images with large infestations (as well as clean) were used, both for training and validation. Finally, the third column represents training with only images of large infestations, and applying the decision function to the entire validation set. Note that the decision function is the

same for the second and third columns, and so the validation results will be identical for stages four through eight. These results are shown in a separate column because the difference between the false positive rates is of interest. The results indicate that training with only large infestations gives the best results, with false positive rates reduced to 1.8%, or 4 incorrectly classified out of 225 images of non-infested kernels in the validation set. As expected, false positives were higher for classification of all images in the validation set vs. only the large infestations (1.8% vs. 1.3%). For large training, the best step was 17, the best spread was 22, and the three best features were 7, 38, and 50. The eight derivatives computed for each pixel were therefore taken as the difference in pixel

Table 4 Classification error rates for parameters selected to minimize false positive rates. (row 9). Column 1 all/all refers to the instance where images of all stages were used for both training and validation. Columns two and three have similar notation

Stage	All/all % error	Large/large % error	Large/all % error
Egg	84.0	x	64.0
1st Instar	60.0	x	52.0
2nd Instar	52.0	x	44.0
3rd Instar	48.0	x	38.0
4th Instar	32.0	40.0	40.0
Prepupa	28.0	20.0	20.0
Pupa	24.0	20.0	20.0
Adult	24.0	16.0	16.0
Empty Tunnel	28.0	24.0	24.0
Total infested	42.0	24.0	35.3
Non-Infested	3.6	1.3	1.8
Total	22.9	8.8	18.7

All/all step = 26 spread = 34 large/large step = 17 spread = 22

intensity between the current pixel and the intensity at eight points of the compass lying on a circle of radius 17 pixels with the current pixel at the center. The derivatives were mapped into the feature histogram with bin boundaries of 0, 22, 44, 66, 88, 110, 132, 154, and 255. The features (7, 38, 50) correspond to the number of pixel locations at which: the derivative between current pixel and NW along circle described above is less than 22; the derivative between current pixel and W is between 88 and 110; the derivative between current pixel and E is between 132 and 154. Figure 5 shows the scatter plots for the two training scenarios for minimal false positive training, with the plot on the left corresponding to training with all images and that on the right with only the large images. Black dots on the gray side of the boundary represent false positives.

Classification results for minimal total error classification are shown in Table 5. Each column represents strategy as described above for minimal false positive classification. Again the results indicate that training with only large infestations gives better classification, with total error reduced to 14.4%. False positives were again higher for classification of all images in the validation set vs. only the large infestations (4.9% vs. 3.1%). For large training, the best step was 18, the best spread was 27, and the three best features were 6, 47, and 61. Figure 6 shows the scatter plots for the two training scenarios for minimal total error training, with the plot on the left corresponding to training with all images and that on the right with only the large images. As above, black dots on the gray side of the boundary represent false positives.

Table 6 shows percent recognition for human observation of digitized film for wheat kernels infested with the granary weevil [6]. Overall results between the algorithm and human recognition are comparable. Humans are much

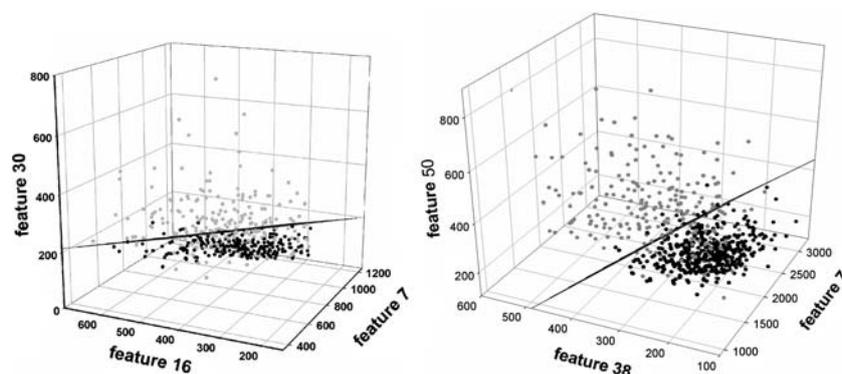


Fig. 5 Scatter plots of mapping the features into the feature space for minimizing false positive classification. The plot on the left is the result of training with all images, while that on the right is the result of training with just the large images. While the decision boundary is a plane in the feature space, the plots have been rotated so that the

plane collapses into a line, allowing visualization of the separation of the two classes. Black represents non-infested kernels and grey represents infested. Black dots on the grey side represent the false positives

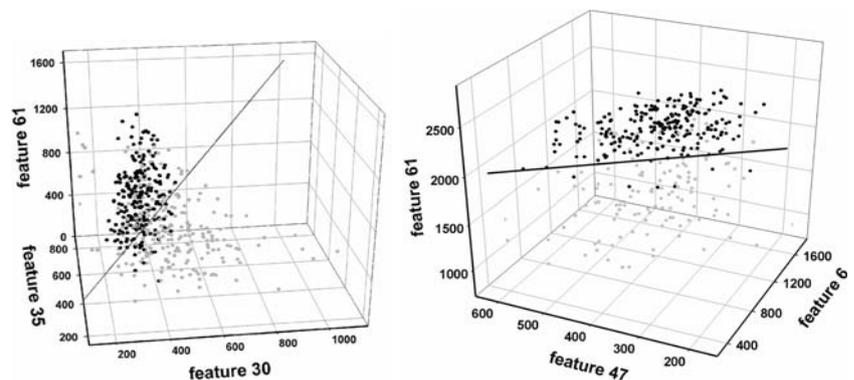
Table 5 Classification error rates for parameters selected to minimize total error rates

Stage	All/all % error	Large/large % error	Large/all % error
Egg	68.0	x	56.0
1st Instar	60.0	x	48.0
2nd Instar	40.0	x	38.0
3rd Instar	28.0	x	24.0
4th Instar	24.0	18.0	18.0
Prepupa	8.0	8.0	8.0
Pupa	4.0	4.0	4.0
Adult	4.0	4.0	4.0
Empty tunnel	8.0	16.0	16.0
Total infested	27.1	9.6	24.0
Non-infested	12.4	3.1	4.9
Total	19.8	5.6	14.4

All / all step = 17 spread = 34 large/large step = 18 spread = 27

more reliable at correctly classifying advance infestations, while the algorithm is more reliable for distinguishing small infestations from non-infested kernels. Another fac-

Fig. 6 Scatter plots of mapping the features into the feature space for minimizing total error. The plot on the left is the result of training with all images, while that on the right is the result of training with just the large images. Black represents non-infested kernels and grey represents infested. Black dots on the grey side represent the false positives

**Table 6** Recognition results for digital observations[6]

Stage	Percent Error				
	Subject 1	Subject 2	Subject 3	Subject 4	Avg.
Egg	86.4	96.4	82.9	92.1	89.4
1st Instar	72.7	90.1	64.4	72.7	75.0
2nd Instar	44.3	69.3	40.7	40.0	49.6
3rd Instar	21.6	48.6	20.3	23.6	28.5
4th Instar	3.4	9.5	0	3.4	4.1
Prepupa	2.0	4.0	3.0	0	2.2
Pupa	2.2	2.0	1.2	0	1.3
Adult	1.7	3.4	1.7	1.0	1.9
Empty tunnel	2.0	6.7	2.0	2.0	3.2
Total infested	26.3	36.7	24.0	26.1	28.3
False positive	5.7	0.7	4.0	1.3	2.9
Total	16.0	18.7	14.0	13.7	15.6

tor to be considered is the variability in the human data between the four subjects, a problem that computers do not share.

Olives

The results for testing the algorithm with all levels of infestation damage are shown in Table 7. As expected, the recognition was low for slight and moderate damage, at 50% and 52% respectively. For severe damage the recognition rose to 87%. False positives were 16%. Total error rate for infested images was 24%. As stated earlier, the poor results and high false positives are believed to be a consequence of the small number of slight and moderate classifications in the training set, hindering the effort to derive a proper decision boundary between the two regions.

The results obtained testing the algorithm with just the severely damaged olives are shown in Table 8. As expected, the recognition of severe damage remained almost unchanged, while false positives dropped from 16% in the previous case to 10%. We are predominantly interested in the severe damage caused by infestation, although

Table 7 Classification results using images for all stages of infestation

Infestation Damage	# in validation set	# Correctly classified	% Recognition
Slight	16	8	50
Moderate	23	12	52
Severe	86	75	87
Non-infested	80	67	84
Total	205	162	79

Table 8 Classification results using only images of large infestations

Infestation damage	# in validation set	# Correctly classified	% Recognition
Severe	86	74	86
Non-infested	80	72	10
Total	166	146	88

detection of all damage would be ideal. However, when considering the desire to detect all damage with the need to reduce the false positive rate, the latter is far more important. Future work will concentrate on reducing the false positive rate, which presumably can be accomplished by training with a larger number of samples so that a more precise decision boundary can be achieved. Unfortunately for our research, but fortunate for the California olive industry, infestations in the last olive season occurred at lower levels than previous years and infested samples were difficult to obtain. This was probably due to the hot, dry conditions experienced last season, which tend to hamper the buildup of olive fruit fly populations [17].

One of the benefits of the feature selection process used here is that it requires no prior knowledge of what features in the image are significant, as the process will sort them out. The technique should work just as well for any sets of images that can be sorted into distinct classes. There may be more suitable criteria for setting up the feature histogram than derivatives and pixel intensities, and the spacing of the histogram bins is an important factor that must be considered.

Conclusion

An automatic recognition algorithm has been developed for detection of inclusions in X-ray images of certain agricul-

tural commodities. Testing of the algorithm on wheat kernels infested with larvae of the granary weevil yielded comparable results to those obtained by human subjects evaluating digitized X-ray film images. The low success of properly classifying images containing small infestations by either the algorithm or human subjects suggests that improved results may be dependant on the development of X-ray equipment capable of generating higher quality images with higher resolution and less noise. Testing of alternate training strategies showed that for this type of algorithm, which uses a form of discriminant analysis with a generally “fuzzy” decision boundary, best results are obtained training with samples that map far away from the boundary, then applying the derived decision function to all samples to be classified.

References

1. R.A. Stermer, *Trans. ASAE* **15**, 1081–1085 (1973)
2. F.E. Dowell, J.E. Throne, J.E. Baker, *J. Econ. Entomol.* **91**(4), 899–904 (1998)
3. Association of Analytical Chemists (AOAC), *Official methods of analysis* 13th edn., (sec. 44.038, 1980) p.790
4. P.M. Keagy, T.F. Schatzki, *Cereal Chem.* **70**(6), 696–700 (1993)
5. C. Karunakaran, D.S. Jayas, N.D.G. White, *Trans. ASAE* **46**(3), 739–745 (2003)
6. R.P. Haff, D.C. Slaughter, *Trans. ASAE* **47**(2), 531–537 (2004)
7. S. Kim, T.F. Schatzki, *Trans. ASAE* **43**, 1695–1702 (2000)
8. S. Kim, T.F. Schatzki, *Trans. ASAE* **44**(4), 997–1003 (2001)
9. P.M. Keagy, B. Parvin, T.F. Schatzki, *Lebensm.-Wiss. u. Technol.* **29**, 140–145 (1996)
10. F.E. Dowell, T.N. Boratynski, R.E. Ykema, A.K. Dowdy, R.T. Staten, *Plant Disease* **86**(9), 1011–1013 (2002)
11. T.A. Lestander, R. Leardi, P. Geladi, *J. Near Infrared Spectrosc.* **11**(4), 433–446 (2003)
12. B.L. Steward, A.L. Kaleita, R.P. Ewing, D.A. Ashlock, *Genetic algorithms for hyperspectral range operator selection*, ASAE paper No. 053063. St. Joseph, Mich.: ASAE 2005
13. S.G. Bajwa, P. Bajcsy, P. Groves, L.F. Tian, *Trans. ASAE* **47**(3), 895–907 (2004)
14. K.C. Lawrence, W.R. William, B. Park, R.J. Buhr, *J. Near Infrared Spectrosc.* **11**, 269–281 (2003)
15. P.M. Mehl, K. Chao, M. Kim, Y.R. Chen, *App. Eng. Agric.* **18**(2), 219–226 (2002)
16. R.L. Kirkpatrick, D.A. Wilbur, *J. Econ. Entomol.* **58**(5), 979–985 (1965)
17. F.G. Zalom, R.A. Van Steenwyk, H.J. Burrack, *Olive fruit fly. Pest notes publication 74112*, University of California Agricultural and Natural Resources (2003)