Fast Insect Damage Detection in Wheat Kernels Using Transmittance Images

Zehra Cataltepe
Siemens Corporate Research Inc.
Intelligent Vision and Reasoning Dept.
755 College Road East
Princeton, NJ 08540
E-mail: zehra@scr.siemens.com

Tom Pearson
U.S. Department of Agriculture
GMPRC
1515 College Ave
Manhattan, KS 66502
E-mail: tpearson@gmprc.ksu.edu

Enis Cetin
Bilkent University,
Dept of E.E. Engineering,
TR-06533 Bilkent, Ankara, Turkey,
E-mail: cetin@ee.bilkent.edu.tr

Abstract— We used transmittance images and different learning algorithms to classify insect damaged and un-damaged wheat kernels. Using the histogram of the pixels of the wheat images as the feature, and the linear model as the learning algorithm, we achieved a False Positive Rate (1-specificity) of 0.12 at the True Positive Rate (sensitivity) of 0.8 and an Area Under the ROC Curve (AUC) of 0.90 ± 0.02. Combining the linear model and a Radial Basis Function Network in a committee resulted in a FP Rate of 0.09 at the TP Rate of 0.8 and an AUC of 0.93 ± 0.03.

I. Introduction

Infested wheat kernels cause loss of quality in the wheat products. They also cause a lot more damage if they are put into storage with other kernels. It is important to be able to identify insect damaged kernels so proper decisions can be made about them.

Current methods of insect detection such as cracking and flotation [1], infrared CO_2 analysis [2], immunological methods [3], NIR [4], and x-ray inspection [5] can be laborious, slow, expensive, and ineffective at distinguishing a sound kernel from a kernel that is internally infested. It is possible that the use of acoustics [6] to detect insects may serve as an alternative which would allow for recognition of kernels where the insect has already emerged as well as those in which the insect is still living inside the kernel. In this paper we describe a method to identify insect damaged kernels based on transmittance images. This method is fast and inexpensive compared with the other methods.

We first segmented the individual wheat kernels from the original transmittance images. Then we used the histogram of pixel intensities from each kernel to decide if it was insect damaged or not. We used a number of different algorithms, namely the linear model, quadratic model, k-nearest neighbor, linear model with weight decay and Radial Basis Function Network. Linear model was the best of all the algorithms with a False Positive Rate (1-specificity) of 0.12 at the True Positive Rate (sensitivity) of 0.8 and an Area Under the ROC Curve (AUC) of 0.90 ± 0.02. Although the radial basis function network performed worse than the linear mode (an AUC of 0.77 ± 0.05), a committee of a linear model and a radial basis function resulted in an improved FP Rate of 0.09 at the TP Rate of 0.8 and an AUC of 0.93 ± 0.03. We also

experimented with K-nearest neighbor model, quadratic model and linear model with weight decay (ridge regression). All of these learning methods resulted in worse performance than the linear model.

II. Wheat Images and Features

Hard red winter wheat (H2) was used to obtain the images. The insect damaged kernel images were taken from wheat infested with rice weevil and kept at about a moisture of 11%. Transmittance images were taken as 800 pixels/inch tif images using an Epson Expression 1680 scanner. The exposure was set to 20 and gamma to 1.22.

The original un-damaged and insect damaged wheat kernel images were taken all together in two different shots. First we segmented each single kernel out of the original pictures using the blue component of the RGB. We obtained 355 good and 364 insect damaged kernels. We rotated each image so they had the maximum height and minimum width. Please see figure 1 for some sample images. The background color was white, so we determined the borders of each wheat image based on the background color. The reflectance along the borders of the image were affecting the features, so we cropped 10 pixels from each pixel row on each side of the wheat.

The histogram of red component of the pixels colors over each wheat image was used as the input feature for the learning algorithm. The 256 different Red components were put into bins as follows: If the red value was less than or equal to 80 the pixel was added into bin 0. If it was larger than 250 it was added into the last bin. Otherwise, the pixel was added into a bin in-between, each bin being responsible for 5 different red values resulting with 36 input features. In addition to these histogram values, we also used a feature similar to the gradient and histogram combination feature of [7]. For each horizontal pixel stripe on the kernel image, we counted the pixels that had a gradient on the red component (current point - the 10th previous) greater than a certain value $(10, 20, 40)$ and red component also greater than a certain value $(200, 220, 240, 250)$. These $3 \times 4 = 12$ additional features resulted in a final feature set of size 49. Figure 2 shows the mean and standard deviation of features for all the available data. The gradient histogram features

Fig. 1. A sample of good and insect damaged kernel pictures.

Fig. 2. The mean and standard deviation of the input features for good and insect damaged kernels.

have more variability than the normal histogram features. We assigned output 0 to the good kernels and 1 to the insect damaged kernels.

In addition to the histogram features, we tried two other features: the minimum, maximum and majority over 3x3 rectangles and the mean on the center of the wheat. However, the results didn't improve, so we don't report them here.

III. LEARNING ALGORITHMS

We used two different learning algorithms, the linear model and the radial basis function (RBF) network [8]. The input features for all the algorithms were $\underline{x} \in \mathcal{R}^{49}$ and the corresponding outputs were $y \in \{-1, 1\}$.

In order to get reliable figures on algorithm performance, we used cross validation. We randomly partitioned all the available data into a training and a test set. The training set used 90% of data from each class and the test set used the remaining 10%. We repeated the partitioning 10 times.

We estimated the model performance using the ROC (Receiver Operating Characteristics) [9], [10], [11] and the Area Under the ROC curve (AUC) [12] on the test set. In order to obtain different False and True Positive rates on the ROC curve, we varied the threshold of each learning algorithm.

- **Linear Model**: Let $A_{N \times (36+1)}$ contain normalized training inputs preceded by 1 and $\underline{b}_{N \times 1}$ contain the outputs y_i for all the N training examples. The linear model is obtained by solving for $\underline{w}_{37 \times 1}$ in the equation $A\underline{w} = \underline{b}$. In order to solve this equation we need to invert $A^T A$.
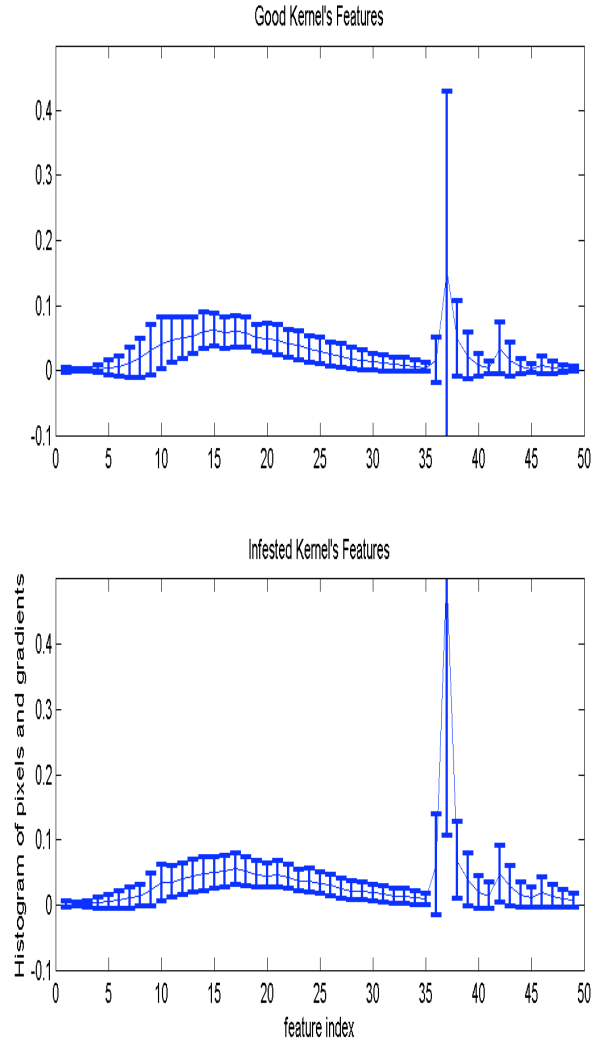
Since A was not full rank, $A^T A$ was not invertible. We used singular value decomposition [13] with $\epsilon = 0.001$. In order to get different points on the ROC curve, we varied the threshold for the output from -2 to 2 in steps of 0.1. If the output for a test case was smaller than the threshold we classified it as good and otherwise we classified it as insect damaged.

- **Radial Basis Function (RBF) Network**: We used the RBF network described in [8]. We chose the first layer weights through K-means clustering of the normalized training data. We computed the basis unit output for basis (first layer) weights $\underline{v}$ and normalized input $\underline{x}$ as $e^{\frac{-\alpha}{36}|\underline{v}-\underline{x}|}$, where $|.|$ denotes the absolute norm of the vector. We computed the output of the RBF network for an input as the linear combination of the basis unit outputs appended by 1. We used singular value decomposition to

Algorithm	AUC
Linear	0.90 ± 0.02
RBF	0.77 ± 0.05
RBF and Linear Committee	0.93 ± 0.03

TABLE I

AREA UNDER ROC CURVE (AUC) FOR DIFFERENT LEARNING ALGORITHMS

Fig. 3. Performance of Different Learning Algorithms.

compute the output layer weights.

Since the three parameters of the RBF network, number of basis units, the scaling factor α of the basis unit function and ϵ of the singular value decomposition made a difference in the performance, we chose the parameters to be used for each training set as follows. We first divided the training set into a new training (90%) and validation (10%) set and we repeated this division 5 times. For each parameter combination, we found the mean AUC over 5 different training-validation partitions. We chose the parameter combination with the maximum mean AUC to be used for the overall training set.

We used thresholds as in the linear model to get different ROC curve points.

- **Linear Model and RBF Network Committee**: We used a linear combination[14] of the RBF network and the linear model outputs as the output of the committee and the same thresholds to get ROC curve points.

IV. RESULTS

For each of the 10 training-test set partitioning of the available data, we used the training set to train the learning algorithm. We then used the test set to compute the ROC (Receiver Operating Characteristics) [9], [10], [11] curve for each partitioning.

We interpolated the ROC curve for each partitioning and reported the mean and standard deviation of the True Positive Rate (sensitivity) for each False Positive Rate (1-specificity) value for each learning algorithm [10]. The mean and the standard deviation on the ROC curve gives us a better idea on the performance of an algorithm. In order to get a reliable mean, we discarded the ROC curve with the maximum and minimum AUC and computed the average ROC curve using the 8 remaining ROC curves. Please see table IV and figure 3.

Because of its simplicity and performance linear model seems to be the best single algorithm. The RBF and linear model committee performed the best. In figure 4 we show the AUC performance of the combined classifier for different weights given to the linear model. The weight of the RBF model is always 1.

Although we experimented with different implementations of the RBF algorithm, such as determining the input weights one by one according to the training example with the worst error, using not all the inputs but a portion of them, we could not improve the performance of the RBF network. We think
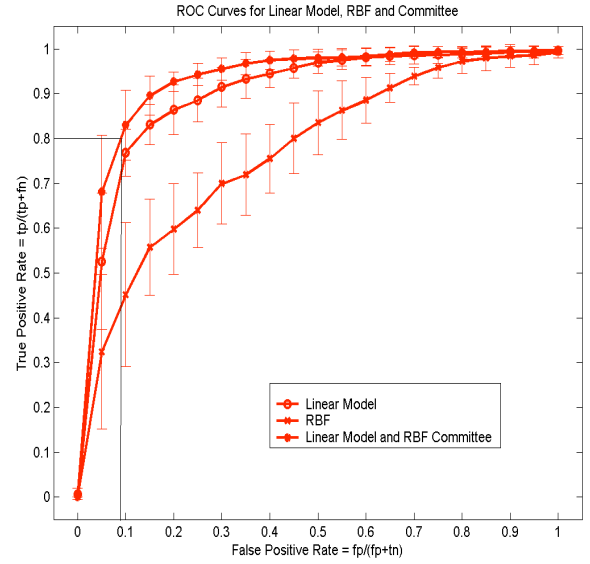
that the high correlation of the input features may be the reason and ICA, PCA or another approach, such as [15] may be helpful.
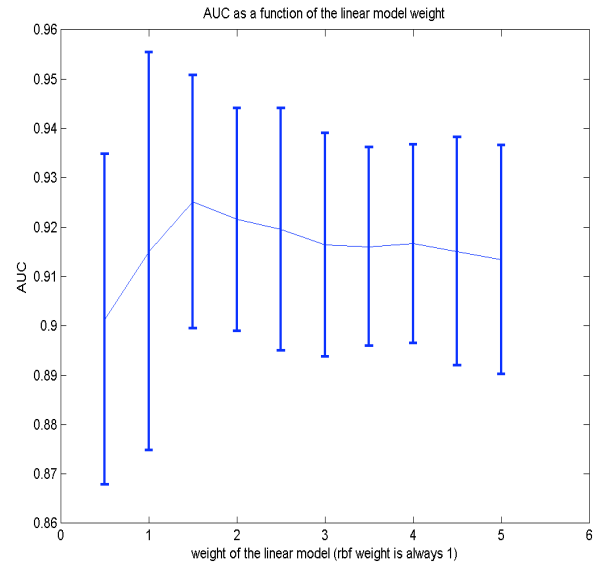

Fig. 4. Performance of the Combined Classifier as the Weight of the Linear Model Changes.

V. DISCUSSION

We used a number of learning algorithms to classify good and insect damaged wheat kernels and we found out that thelinear model performed the best. Additional information about the kernels such as reflectance images or compression force or conductance measurements [6] could be used to

improve performance of a single classifier. Another approach is to train different classifiers with each of these features and then combining them [16].

Since the machines that automatically detect damaged kernels must work very fast, it is important to determine which few features are the most important to determine the damaged kernels [7]. Approaches such as Independent Component Analysis (ICA) or Principal Component Analysis (PCA) could be used to select important features. It is worth noting, however, that on a 1.33 GHz Power PC G4 running MacOS X, linear or RBF network models take less than 1 milisecond to classify a sample. Since feature extraction for the samples used in this paper was performed manually, this figure does not include the time required for feature extraction.

ACKNOWLEDGEMENTS

REFERENCES

[1] G.E. Russell, "Evaluation of four analytical methods to detect weevils in wheat: granary weevil, sitophilus granarius in soft white wheat," *Journal of Food Protection*, vol. 51, pp. 547–553, 1988.

[2] W.A. Bruce, M.W. Street, R.C. Semper, and D. Fulk, "Detection of hidden insect infestations in wheat by infrared carbon dioxide gas analysis," *ARS bulletin*, vol. July, AAT-S-26, 1982.

[3] F.A. Quinn, W. Burkholder, and G.B. Kitto, "Immunological technique for measuring insect contamination of grain," *Journal of Economic Entomology*, vol. 85(4), pp. 1463–1470, 1992.

[4] F.E. Dowell, J.E. Throne, and J.E. Baker, "Automated nondestructive detection of internal insect infestation of wheat kernels by using near-infrared spectroscopy," *Journal of Economic Entomology*, vol. 91(4), pp. 899–904, 1998.

[5] R.A. Stermer, "Automated x-ray inspection of grain for insect infestation," *Transactions of the ASAE.*, vol. 15, pp. 1081–1085, 1972.

[6] T. Pearson, D.L. Brabec, and C.R. Schwartz, "Automated detection of internal insect infestations in whole wheat kernels usign a perten skcs 4100," *Applied Engineering in Agriculture*, vol. 19(6), pp. 727–733, 2003.

[7] T. Pearson and R. Young, "Automated sorting of almonds with embedded shell by laser transmittance imaging," *Applied Engineering in Agriculture*, vol. 18(5), pp. 637–641, 2002.

[8] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[9] C. E. Metz, "Basic principles of roc analysis," *Seminars in Nuclear Medicine*, vol. 8, pp. 283–298, 1978.

[10] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," in *Proc. 15th International Conf. on Machine Learning*. 1998, pp. 445–453, Morgan Kaufmann.

[11] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine," *Clinical Chemistry*, vol. 39, pp. 561–567, 1993.

[12] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, No. 7, pp. 1145–1159, 1997.

[13] W. H. Press, W. T. Vetterling, B. P. Flannery, and S. A. Teukolsky, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[14] G. Fumera and F. Roli, "Performance analysis and comparison of linear combiners for classifier fusion," *Lecture Notes in Computer Science, Structural, Syntactic and Statistical Pattern Recognition*, vol. 2396/2002, pp. 424, 2003.

[15] R.B. Huang, Y.M Cheung, and L.T. Law, "A divide-and-conquer based radial basis function network with application to recurrent function modeling," in *Proceedings of 2003 International Joint Conference on Neural Networks (IJCNN'03)*, Portland, Oregon, USA, 2003, pp. 512–516.

[16] J. Kittler M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20(3), pp. 226–239, 1998.