Classification of Pistachio Nuts from X-ray Images

David Casasent and Ashit Talukder
Carnegie Mellon University, Pittsburgh, PA
Thomas F. Schatzki and Pamela M. Keagy
Western Regional Research Center ARS, USDA

Abstract

Classification of X-ray images of randomly oriented touching pistachio nuts is discussed. The ultimate objective is the development of a system for automated non-invasive detection of defective agricultural product items on a conveyor belt. We discuss the selection of statistical features from X-ray images for discrimination between infested and clean items (pistachio nuts). These features are used as inputs to a new modified k nearest neighbor classifier. For this application, a classifier with the best classification rate is not best; rather, it is desired that nearly all good pistachio nuts be correctly classified while locating as many infested nuts as possible. The classifier is modified to provide excellent classification at the desired performance level.

Key Words: Classification, detection, feature extraction, k nearest neighbor classifier (modified), product inspection, X-ray sensors
Introduction

The internal product detail that X-ray images provide allows the presence of worm damage and other defects to be determined by non-destructive(non-invasive) methods in various agricultural products, such as apples (1) and other agricultural products (2,3). Current standard inspection techniques cannot determine many defects that X-ray images can, such as worms, insect damage, etc., since they are typically not seen in standard external visual inspection. Prior work (3) has confirmed that X-ray images can be useful for identification of infected versus clean pistachio nuts.

Prior work on pistachio nut classification (3,4) did not address the problem of segmentation of touching nuts. The image preprocessing developed by us to segment the touching pistachio nuts is discussed in a previous publication (5). In (4) only a hand-picked selected subset of the pistachio nut database was used, and a limited set of histogram features were extracted from each image. In (3), similar features such as the ones used in this paper were used for classification of a smaller version of our database. The current paper uses a much larger database (6).

Fig. 1 shows a typical image of such products on a conveyor belt that would be input to an automatic vision inspection system. Fig. 2 shows typical X-ray images of clean and infected pistachio nuts. As seen, such X-ray images contain useful internal information about the quality of the nut. Infected pistachio nut images (Figs. 2a-h) tend to have large gray scale variations (due to the presence of worm tunnels, infestations, etc.) along with large airgaps (outer dark region between nutmeat and shell), whereas clean nuts typically have smooth nutmeat regions with smaller airgaps (Figs. 2a-d).

This product inspection application is a very difficult classification problem; prior work (6) on classification of a subset of these X-ray images by human experts yielded an average 1% correct classification by five of the best human experts of only 56.2% (standard deviation in 1% was 4.3%). It is desired that an automated classification system produce an equivalent or better performance on the pistachio nuts.

In ready-for-sale nuts such as those used here, 3% of the nuts contain insects or feeding damage, while USDA standards require 1-3% of such damage (6). A 50% reduction in infected nuts, while restricting rejection of good nuts to a commercially acceptable level of 1%, is desirable for quality
and aflatoxin reduction and for commercial reasons. Tests have shown that present automated and manual methods result in rejection of as much as 20% of the good crop (clean nuts) when the amount of infested nuts is reduced to 1%. Therefore, a better and automated classification method is necessary.

In this paper, we discuss rotation and scale invariant histogram features extracted from the the X-ray images that contain better discriminatory information compared to ones used in prior work (4). A new modified k nearest neighbor classifier is used to classify each X-ray image from the extracted features; the modified k nearest neighbor classifier (modified k-NN classifier) is shown to be more robust to the choices of \( k \) than the standard k-NN classifier (7). A new procedure to correctly classify \( \geq 98\% \) of the clean nuts and locate most of the infested nuts at the same time is detailed. We also discuss a new scheme to select the number of features to use from training data \\( P(x) \) performance.

The database used, desired performance, and the new histogram features extracted from the the X-ray images are detailed first. The new modified k nearest neighbor classifier used to classify each X-ray image from the extracted features is discussed next. We discuss the method we use to order the histogram features by their importance or usefulness for discrimination and the selection of the number of the most important histogram features to use.

Materials and Methods

Database used

The database we consider consisted of 25 trays of scanned 2-D X-ray film images of large pistachio nuts (18-20 nuts per ounce). Each image contained about 100 nuts at random orientations. The nuts used were obtained from a processor after sorting and sizing processing and hand-inspection to remove twigs and other non-nut material. These are typical images of such products on a conveyor belt that would be input to an automatic vision inspection system.

Each tray was X-rayed (90 sec. at 25 kV [with an 0.25 mm Be window] with a Faxitron series X-ray system 4380N, Faxitron Corp., Buffalo Grove, IL; Industrex B film, Eastman Kodak,
Rochester, NY). Twelve-bit digital images of these X-ray films were obtained at a resolution of 
(0.173 mm)$^2$/pixel using a Lumi Lane 150 film scanner (Lumina, Sunnyvale, Ca). These images 
were reduced in pixel count by a factor of nine by pixel averaging to produce 12-bit images with 
a resolution of (0.5 mm)$^2$/pixel. Further details of the X-ray imaging system and the digitization 
technique used are discussed in (5).

All of the nuts were dissected and classified into good and insect damaged (bad) by visual 
inspection after dissection (6). Nuts in other categories (immature kernel, large kernel spots, etc) 
were not included. A total of 1,834 nuts were used for classification; these were divided into a 
training set of 942 nuts (600 good and 342 bad) and a test set of 942 nuts (600 good and 342 bad); 
only the training set was used to select the classification parameters (such as the number of features 
to use, and the parameters to normalize each feature before classification). To obtain classification 
performance $P_C$ using only the training set data, we used a leave-one-out testing classification 
scheme on the training data. Leave-one-out classification of a training set of size $S$ involves picking 
one of the training samples in the training set, and using the rest of the $S-1$ training data samples 
as prototypes to classify the selected sample. This is repeated $S$ times, each time using a different 
sample; the overall $P_C$ performance on the training set is then noted.

**Desired performance**

As discussed earlier, for this pistachio nut classification problem, obtaining the best overall clas-
sification accuracy is not desirable; it is commercially desirable to classify most clean pistachiros 
correctly. In many classification problems, it is often necessary to obtain better $P_C$ for one class 
as compared to another class; this is the case in our pistachio nut classification problem. It is 
preferable that $\geq 50\%$ of the infested pistachio nuts in the database be correctly classified, while 
correctly classifying $\geq 99\%$ of the clean nuts in the database.

Receiver operating characteristic (ROC) curves (8) are commonly used for applications where 
true objects need to be classified versus non-objects (clutter). ROC curves involve plotting the 
probability of detection ($P_D$) or the probability of correct classification of an object ($P_C$) versus the 
probability of false alarm ($P_A$), by varying prior probabilities for each class (object versus clutter).
In our current application, clean pistachio nuts are viewed as objects, and infested pistachio nuts are treated as clutter. The ROC curves we use show the variation in the ratio of correctly classified clean nuts versus the ratio of incorrectly classified infested nuts.

**Rotation and scale invariant histogram features**

The features extracted from each input X-ray image should contain useful information that allows discrimination between clean and infested pistachio nuts. These features should be scale-invariant since the size of a pistachio nut can vary from as few as 350 pixels to as many as 900 pixels for large nuts (5). The input pistachio nuts can lie at any orientation; therefore, rotation-invariant features are also needed. Histogram statistical features of an input gray-scale image are rotation-invariant, since they do not contain spatial information. Histogram statistical features are also scale-invariant, if they are calculated by dividing the gray-scale distribution for the pistachio nut pixels by the total area of the nut.

The features we extracted from each pistachio nut X-ray image were the mean, variance, skewness, and kurtosis (histogram features) of four different images of each nut (raw, edge, curvature (8) of raw and curvature of edge images) (10). Variations of these features have been used in prior X-ray pistachio nut classification work (6). In prior work (4) on a hand-picked subset of the database used in this thesis, histogram features were extracted from only the input raw X-ray image and the edge enhanced X-ray image. Our edge images were obtained by calculating the sum of the absolute values of the differences between a given pixel value and its eight neighbors. For each of the resultant four sets of images of the segmented nuts, four sets of histogram features were calculated. This gave a total of 16 possible features. The histogram for each image of each nut was divided by the total number of pixels in each nut image (this provides scale-invariant features). The four histogram features were then calculated separately for the four sets of images. The mean measures the average gray value on the pistachio nut, and variance is a measure of "spread" of gray values within the nut. Skewness is a measure of the symmetry of the distribution; kurtosis is a measure of how sharply peaked the distribution is.

The raw images for infested pistachio nuts tend to be darker with more gray-scale variations,
whereas clean nuts tend to have smoother gray values. Hence, we expect the variance of the raw images to be different for infested pistachio nuts. Edge (high-frequency) information is also expected to be useful, since infested pistachio nuts tend to have rougher texture compared to clean nuts. The edge images for a clean (Figure 3a) and infested (Figure 3b) pistachio nut are shown in Figs. 3h and 3i respectively. The curvature images provide information about the rate of change of gray-values (combination of first and second order differentials in gray-scale) over the entire gray-scale pistachio image. This information is very useful, especially in regions near the airgaps. Infested pistachio nuts tend to have larger and darker airgaps and sharp transition regions (high curvature) between the airgap and nutmeat, whereas clean pistachio nuts tend not to have such high gray-level transition (curvature) regions. Therefore, we expect the curvature image to contain important discriminatory information about each individual pistachio nut. A curvature image for a good (infested) nut is shown in Figure 3c (Figure 3g). The curvature image of the edge-enhanced image for a clean nut and an infested one are shown in Figs. 3d and 3e respectively; no clear differences are immediately apparent.

For the curvature and edge-enhanced images, we morphologically erode the output images with a $3 \times 3$ structuring element prior to using them; this removes the outer boundary between the shell and the background which tends to have very high curvature values due to the gray-scale change in such regions. For only the curvature images, the output images were clipped at $\pm T_c$ (all values $\geq T_c$ are set to $T_c$ and all values $\leq -T_c$ were set to $-T_c$; $T_c=1.5$ was used) and each clipped curvature image was separately normalized to 0-255. This separate normalization reduces the usefulness of the mean of a nut in curvature data; but, it enhances the use of variance, skewness and kurtosis, since good nuts should not have many concave regions (worm tunnel, etc.) while infested nuts should have both concave and convex regions.

**Classifier: Modified k-nearest neighbor classifier**

A new modified nearest neighbor algorithm was used to classify sample data using features extracted from each sample. The standard nearest neighbor classifier (7) classifies test data by comparing the Euclidean distance of each test sample to the nearest prototype (training) sample. The test
sample is assigned to the class of the closest prototype sample. A variation of the nearest neighbor classifier is the k-nearest neighbor (k-NN) classifier (11). In this method, the k nearest prototype samples from a test sample are computed. The test sample is then assigned to that class with the majority (winning class) among the k nearest prototype samples. The k-NN classifier has been proven to provide classification with a maximum classification error that is twice that of the Bayes classifier (7).

However, the k-NN classifier has problems when the amount of overlap between classes is high, and the number of prototypes (training samples) per class is low (12). When the number of prototypes is low, the classification results depend highly on the choice of k (12,13). The distance to the nearest prototype samples is a useful measure, when the number of prototypes in each class is low, and such information could provide a more robust measure for classification. Such modifications have been suggested for the nearest neighbor rule (13-15). A weighted distance k-NN technique (13) weights each of the k-nearest neighbors based on their distances to the test sample; the weight assigned to each of the k nearest samples is inversely proportional to its distance to the test sample. For each class among the k-nearest samples, the sum of the weights for that class is computed. The class with the largest sum of weights is assigned as the winning class. A nearest unlike neighbor scheme has also been suggested (14) in which the relative distance between the winning nearest neighbor class and the closest losing class (nearest neighbor among the other classes) is used as a confidence measure for classification. This approach is used to reject outliers in the test data; however, the nearest unlike neighbor scheme uses only k=1.

An example of the problems with the standard k-NN classifier for test samples that lie close to the boundary between different classes is shown in Figure 4. In such cases, the choice of k in the k-NN classifier is a critical parameter. In Figure 4a, the test sample (+) belonging to class 1 (o) is located in a region of overlap between class 1 (o) and class 2 (x). If k is small (k<3) for this example (Figure 4a), the test sample (+) will be classified wrongly as a class 2 (x) sample. An example where a large value of "k" for the k-NN will result in the test sample from class 1 (+) being wrongly classified as belonging to class 2 (x) is shown in Figure 4b. In this case, the 3 nearest samples to the test sample belong to the correct class 1 (o), but when k=7,9 or 11, the
test sample in Figure 4b will be wrongly classified as a class 2 (x) sample. As shown in these examples, the k-NN can be quite sensitive to the choice of k, especially for samples that lie close to the decision boundary between two classes. The reason for this is that the k-NN does not use the distance of each test sample from the k-nearest prototypes.

We use a modified k-NN using the closest average distance per class for the k-nearest neighbors for each class. This approach appears to be robust to the presence of outliers in the prototype data. Our method is similar to the weighted distance technique (13), but is simpler since we use a linear function of the distance to the closest prototypes. The modified k-nearest-neighbor classifier we use is now detailed. For each test sample \( x_t \), we compute the average distance of each test samples to the closest k samples \( x_{i,t} \) in each class. Thus, we calculate the average distance of the test sample to the k closest prototype samples \( x_{i,1} \) in class 1 \( (d_{\text{avg},i} = (1/k) \sum_{i=1}^{k} (x_t - x_{i,1})^2) \), the average distance to the closest prototypes \( x_{i,2} \) in class 2 \( (d_{\text{avg},i} = (1/k) \sum_{i=1}^{k} (x_t - x_{i,2})^2) \), etc. for all L classes. The test sample is assigned to the class with the closest average distance, \( d_{\text{avg},i} \), i.e., the test sample is assigned to class \( c \) if \( d_{\text{avg},c} \leq d_{\text{avg},i} \) \( \forall i \neq c \). When \( k=1 \), each test sample is classified to the class corresponding to the nearest neighbor and our modified k-NN is the same as the nearest-neighbor classifier. For training data, we use the leave-one-out procedure to obtain the \( R^2 \) performance of the modified k-NN on the training set.

There are two main advantages to our new modified k-nearest-neighbor classifier compared to a k-NN classifier. The k-NN is sensitive to the choice of k for test samples that lie close to class boundaries when the number of prototypes is small (12) (Figure 4). If our modified k-NN is used for the examples in Figures 4a and 4b, the test samples in Figures 4a and 4b will be correctly classified as a class 1 sample \( (o) \) for a reasonable choice of k \( (k=1 \text{ to } 7 \text{ for Figure 4a and } k=1 \text{ to } 7 \text{ for Figure 4b}) \). This occurs because the average distance of the test sample to the class 1 prototypes is smaller than the average distance of the test sample to the class 2 prototypes in both Figures 4a and Figures 4b. Therefore, we expect our modified k-NN to be robust for test samples that lie close to the decision boundary between two classes, since it rejects outliers in the prototype data.

We noted earlier that it is not always preferable to obtain the best overall classification accuracy.
(18) for both classes. The ROC curve of \( \hat{p}_c \) for clean mislabeled inputs with the standard k-NN classifier is obtained by changing the desired “majority” for each class (15). Since the majority measure in the k-NN can only be changed in integer increments, only coarse ROC measurements result; another minor advantage of our modified k-NN is that it provides finer ROC curves. To obtain an ROC curve using our modified k-NN method, we assign a test sample to a class using the following rule: the test sample is assigned to class \( c \) if \( (d_{t, c} - d_{t, \hat{c}}) < T_v \) \( \forall l = 1, 2, \ldots, L, l \neq c \) where we vary the \( T_v \) threshold for class \( c \). If \( T_v = 0 \), each sample is assigned to the class with the closest average distance \( d_{t, c} \). If \( T_v = -\infty \), all test samples are assigned to class \( c \); if \( T_v = +\infty \), then none of the test samples are assigned to class \( c \). Therefore, the thresholds \( T_v \) are analogous to the confidence measure assigned to each class.

Parameter \( (k) \) selection in modified k-NN classifier

Not much prior work has been done on the optimal choice of \( k \) in the k-NN classifier. It has been noted (12) that the choice of the optimal \( k \) in the k-NN classifier is a function of three parameters: feature dimensionality, number of prototypes, and the ratio between the number of samples in both classes. The choice of the optimal \( k \) increases with increasing sample size and increasing feature dimensionality: for a given training sample size and feature dimensionality, the \( \hat{p}_c \) performance typically increases with increasing \( k \) up to some \( k \) value and then decreases (13). This property has been theoretically proven for noise-free Boolean valued features called Boolean threshold functions (12). It is often referred to as the peaking performance of the k-NN by empirical analysis (12, 13). We expect the same trend to hold for our modified k-NN classifier. We determine the optimal choice for \( k \) in the modified k-NN classifier by evaluating its performance with changing \( k \) on the training data (using the leave-one-out test on training data method). We varied \( k \) from 1 to 23 in the modified k-NN, using all 16 input histogram features. The \( \hat{p}_c \) performance of the modified k-NN classifier is seen (Figure 5) to increase steadily with increasing \( k \) until \( k = 17 \), then the \( \hat{p}_c \) performance drops with \( k > 17 \). We also tested the \( \hat{p}_c \) performance of the standard k-NN classifier with varying \( k \), and found that \( \hat{p}_c \) performance of the standard k-NN classifier is “more erratic” with changing \( k \). We feel that this occurs because the standard
k-NN classifier does not perform robust classification of the test samples that lie close to the class boundaries; its performance is therefore sensitive to the choice of k. Therefore, the modified k-NN is preferable in this case in terms of selection of the k value to use.

The 17% performance of the modified k-NN on the training data was found to peak at k=17 (Figure 5). Therefore, we used a value of k=17 in the modified k-NN classifier to classify the pistachio nuts using different feature spaces. Note however that the choice of k in the modified k-NN is not critical and that it gives comparable 17% results for other values of k.

Results and Discussion

The 16 rotation and scale-invariant histogram features extracted from each individual pistachio nut are not all equally useful for discriminating between clean and infested pistachio nuts. Hence, it may be necessary to order these features by their importance for discrimination and to select a subset of these features as inputs to the classifier (classifiers themselves cannot easily find both the features and the best combination of features to use). The selection of the optimal subset of features to use for a specific application is best done by evaluating 17% performance of all possible subsets of input features. It is well known that the problem of selecting the best subset of input features to use for classification is an NP-complete problem (16). We use a sub-optimal solution where we use Statistical Analysis Software (SAS) (17) to order the input histogram features. The number of these ordered features to use is determined from the 17% performance on the training set (using a leave-one-out procedure). The forward-selection technique (17, page 911) in the STEPDISC procedure in SAS (17) was used to order the input histogram features in their order of importance for discrimination. The default parameters (17) of the stepwise discriminant analysis procedure were used.

The first seven histogram features selected by SAS were (in order of importance): variance of curvature of raw images, kurtosis of raw images, variance of curvature of edge images, variance of raw images, variance of edge images, skewness of curvature of raw images, and the mean of curvature of raw images.
We tested the classification performance using the histogram features. The modified k-NN classifier was used. We classified each pistachio nut in the test set using the original 16 histogram features (4 statistical features each from the raw, edge, curvature, and curvature of edge image).

Classification using original histogram features

The sixteen original histogram features were analyzed using SAS; forward-selection was used to order the histogram features. Each feature was normalized to a zero to one range (using only the training set data). Test set data was normalized using the training set parameters. Table 1 shows $I_C$ (percentage of correct classification) obtained with different numbers of features (ordered by SAS) using the modified k-NN classifier with $k=17$. As the number of histogram features used was increased, the percentage of nuts correctly classified $I_C$ generally increased and then decreased (Table 1). From training set data, we determined to use seven histogram features (see bold entries in Table 1). $I_C$ results for the test set follow training set data quite well in Table 1, thus, generalization is good. Therefore, there are differences in the $I_C$ performance when the number of input features vary from two to sixteen.

Preferable $I_C$ measure classifier

Tables 2 and 3 show the confusion matrices for the training and test sets using the modified k-NN with $k=17$, and the seven best original histogram features. Both training and test set scores are similar; this shows good generalization. As noted, the standard $I_C$ performance measure is not the desired one, nor is a classifier with the best $I_C$ (as seen in Table 3, this will reject around 5.5% of the clean crop while rejecting 73.2% of the infested crop). Therefore, if best overall $I_C$ is used, a large number of clean pistachio nuts will be rejected. To achieve preferable performance, we consider detecting only infested nuts with a high degree of confidence.

The algorithm we use is now described. We only classify a nut as infested (bad) if the average distance of each test sample to the k-nearest infested pistachio nut prototypes is less than the average distance to the k-nearest clean (good) nut prototypes by T. Using this technique, only infested nuts with high likelihood of correct classification (large T) are removed and very few clean
nuts are expected to have a large T and be misclassified as bad and be rejected. Note that varying T is similar to varying the ratio of the prior probabilities for each pistachio nut class in a statistical classifier.

Table 4 shows new preferable \( P_C \) results as T is varied. As T is decreased, more bad nuts are detected (\( P_C \) (infested) increases) but more clean nuts are rejected (\( P_C \) (clean) decreases). Thus, a larger T is preferable. T=0.0325 results in classification (rejection) of \( P_C \approx 52\% \) of the infested nuts in the training data (this reduces the infested nuts to 1.5\% of the crop) while rejecting only 1\% of the clean nuts (\( P_C \approx 99\% \) for clean nuts). We use the same threshold T value for the test set. The performance of the test set for this preferred operating point follows the training set; \( P_C \approx 49\% \) of the infested nuts in the test set are rejected (correctly classified), while rejecting only 0.7\% of the clean nuts. It is also possible to select other operating points in Table 4, if it is desired to locate more infested pistachio nuts, at the cost of lower \( P_C \) performance for clean nuts.

Conclusions

We have obtained excellent classification results on segmented agricultural products (pistachio nuts) using rotation and scale invariant features, and a new modified k-NN classifier. We also discussed a method to select the number of features to use, and an efficient scheme to achieve very high classification rates for clean nuts and good classification rates for infested nuts. Use of our histogram features was shown to provide better classification than the average \( P_C \) performance obtained by the five best human experts. An improvement of \( \approx 2.4\% \) was obtained over the average performance of the human subjects. This improvement is commendable since many of the misclassified infested (clean) nuts do not look bad (good) in X-ray images; e.g. nuts with splits can be misinterpreted as having worm tunnels, air gaps could be classified as infested regions, etc. The truth data provided was obtained by dissecting each nut, and visually classifying each dissected nut. Much of this information visible after dissection does not appear in the X-ray imagery. Hence, this is a formidable pattern recognition problem, and even 2-3\% improvements in classification at the 88\% classification level is notable. Despite these problems, the classification achieved here, had
nuts reduced to about 1.5% of the crop with only 1% of the good nuts rejected, appears to be very useful.

Acknowledgments

The authors gratefully acknowledge the work of Lan-Chau Lee for categorization of the pistachio nuts. The classifier work is supported by U.S. Department of Agriculture grants 97-31625 and 98-35508-6340. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the U.S. Department of Agriculture. Reference to a company or product name does not imply approval or recommendation of that product by the U.S. Department of Agriculture to the exclusion of others that may be suitable.
References


List of Figures

1. Typical scanned X-ray image of a tray of pistachio nuts showing internal detail.
2. X-ray images of clean (a-d) and infested (e-h) pistachio nuts.
3. (a) Raw image of clean pistachio nut, (b) corresponding edge image, (c) corresponding curvature image, and (d) corresponding curvature of edge image; (a) Raw image of infested pistachio nut, (f) corresponding edge image, (g) corresponding curvature image, and (h) corresponding curvature of edge image.
4. Examples exhibiting sensitivity of the k-NN to the choice of k: (a) k-NN fails with small k, (b) k-NN fails with large k. In both cases, the test sample 'L' belongs to class 'a'.
5. k vs. training data with changing k in standard k-NN and modified k-NN for all input histogram features.
Figure 1: Typical scanned X-ray image of a tray of pistachio nuts showing internal detail
Figure 2: X-ray images of clean (a-d) and infested (e-h) pistachio nuts.
Figure 3: (a) Raw image of clean pistachio nut, (b) corresponding edge image, (c) corresponding curvature image, and (d) corresponding curvature of edge image; (e) Raw image of infested pistachio nut, (f) corresponding edge image, (g) corresponding curvature image, and (h) corresponding curvature of edge image.
Figure 4: Examples exhibiting sensitivity of the k-NN to the choice of k: (a) k-NN fails with small k, (b) k-NN fails with large k. In both cases, the test sample "+" belongs to class "x".
Figure 5: $R^2_c$ on training data with changing $k$ in standard k-NN and modified k-NN for all input histogram features.
<table>
<thead>
<tr>
<th>No. of Features</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_C$ (Train)%</td>
<td>82.3</td>
<td>87.2</td>
<td>87.2</td>
<td>87.7</td>
<td>87.8</td>
<td>87.9</td>
<td>88.0</td>
<td>87.6</td>
</tr>
<tr>
<td>$I_C$ (Test)%</td>
<td>80.1</td>
<td>86.8</td>
<td>86.6</td>
<td>87.7</td>
<td>87.7</td>
<td>88.4</td>
<td>88.7</td>
<td>87.4</td>
</tr>
</tbody>
</table>

Table 1: $I_C$ for different numbers of histogram features (modified k-NN).
<table>
<thead>
<tr>
<th></th>
<th>Clean</th>
<th>Infested</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>93.7%</td>
<td>6.3%</td>
</tr>
<tr>
<td>Infested</td>
<td>12.5%</td>
<td>77.5%</td>
</tr>
</tbody>
</table>

Table 2: Confusion matrix using modified k-NN (Training set, 7 features).
<table>
<thead>
<tr>
<th></th>
<th>Clean</th>
<th>Infected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>94.5%</td>
<td>5.5%</td>
</tr>
<tr>
<td>Infected</td>
<td>10.8%</td>
<td>79.2%</td>
</tr>
</tbody>
</table>

Table 3: Confusion matrix using modified k-NN (Test set, 7 features).
<table>
<thead>
<tr>
<th>( T )</th>
<th>0.0325</th>
<th>0.0215</th>
<th>0.0110</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{r}_2 ) (Clean) Train</td>
<td>99%</td>
<td>98%</td>
<td>97%</td>
</tr>
<tr>
<td>( \hat{r}_2 ) (Infected) Train</td>
<td>51.8%</td>
<td>58.5%</td>
<td>70.2%</td>
</tr>
<tr>
<td>( \hat{r}_2 ) (Clean) Test</td>
<td>99.3%</td>
<td>98%</td>
<td>97.5%</td>
</tr>
<tr>
<td>( \hat{r}_2 ) (Infected) Test</td>
<td>48.3%</td>
<td>57.3%</td>
<td>68%</td>
</tr>
</tbody>
</table>

Table 4: Classification accuracy with varying \( T \).