Machine Learning to Identify Defects on Manufactured Workpieces

Ashley Batchelor

Abstract

I used Naïve Bayes classification models to identify manufacturing defects on multiple types of workpieces. The first type included two variations of simulated defects on machined aluminum. The second type included defective integrated circuit leads.

Sec. I. Introduction and Overview

A common method of inspecting industrial workpieces is through machine vision inspection. Typically, an industrial microscope may be configured with a movable stage and sophisticated software for analyzing workpieces. Such systems may be utilized for dimensional metrology, e.g. for dimensional verification by measuring edges, heights from focus, or other features to check manufacturing tolerances. A growing field is inspection for defects. One method is to use machine learning to train a model with defective and non-defective images that may be utilized to inspect for defects on newly manufactured workpieces.

Various techniques have been investigated for using machine vision for detecting and classifying manufacturing defects of metal or fabric. For example, Ding Shumin et al. (Zhongyuan University of Technology) describe using AdaBoost for fabric defect detection and Support Vector Machines for defect classification.[1] Xian Tao et al. (China University of Mining and Technology) describe methods for using neural networks for inspecting metal surfaces for defects.[2] Various patent literature discloses techniques for defect classification. For example, US Patent number 8,315,453 (Applied Materials Israel Ltd.) discloses automatic defect classification techniques using support vector machines. US Patent number 8,660,340 (Hitachi High Technologies Corp.) discloses defect classification techniques using binary classification trees. US Patent Application number 20160163035A1 (KLA-Tencor Corp.) discloses a method for classifying semiconductor wafer defects with neural networks.

Various techniques have been investigated for using wavelet transforms for feature selection for using machine vision for defect detection. For example, Dimitrios Karras describes methods for textile inspection including wavelet transforms for feature selection and Support Vector Machines for textile defect detection using multi-dimensional feature data from multiple wavelet transforms including Haar, Daubechies-9, Coiflets and Symlets.[3]

---

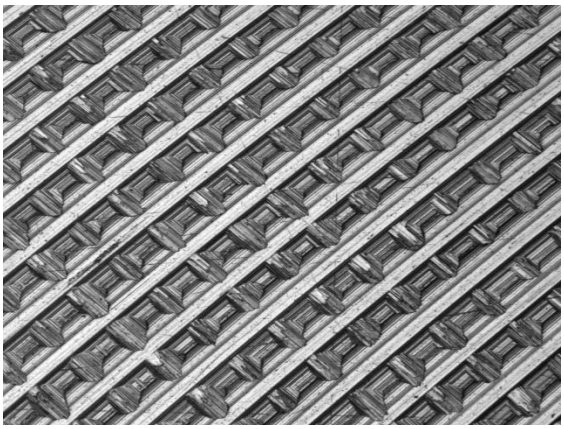
[1] Shumin, Ding & Zhoufeng, Liu & Li, Chunlei. (2011). "AdaBoost learning for fabric defect detection based on HOG and SVM." International Conference on Multimedia Technology (ICMT).

[2] Tao, Xian, et al. "Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks." Appl. Sci. 2018, 8, 1575.

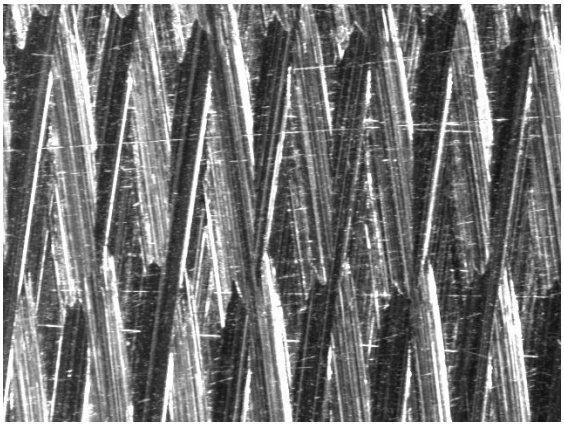
[3] Karras, D.A "Robust Defect Detection Using Improved Higher Order Wavelet Descriptors and Support Vector Machines for Visual Inspection Systems" Proceedings of the 2006 IEEE International Workshop on Imagining Systems and Techniques (IST 2006), 2006, pp.180-185.

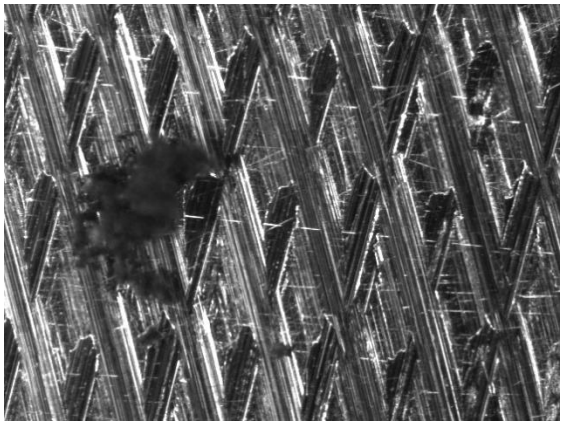
I investigated some simple methods for creating machine learning models to inspect for defects including two types of wavelet transforms for preliminary feature selection, Haar and Symlet 4.

The first type of workpiece defects I analyzed was machined aluminum, including one set with defects simulated with black spray paint droplets, and one set with defects simulated by passing graphite powder through a screen. The first type of workpieces was imaged with a Mitutoyo QVSTREAM machine vision inspection system at Micro Encoder, Inc. in Kirkland, WA. Each image was a 640x480 grayscale image. This is a proprietary data set which is not available to the public, but I have included sample images here.

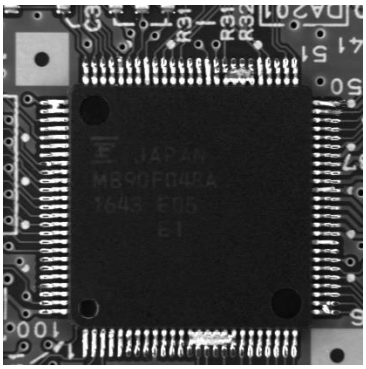


Figures 1 a), and b) – Machined aluminum workpiece with no defect, machined aluminum with spray painted simulated defect.





Figures 2 a), and b) – Machined aluminum workpiece with no defect, machined aluminum with graphite simulated defect.



Figures 3 a), b), and c) – An integrated circuit on a motherboard, a lead leg with no defects, and a defective lead leg.

The second type of workpiece defects I analyzed was lead legs of an integrated circuit on two computer motherboards, as measured by an unspecified model of microscope. One motherboard had defective lead legs. These were from a Kaggle data set.[4] Each image was a 38x180 grayscale image.

For learning and testing, each set of the images described above included a set of images of workpieces with defects and a set without defects.

Sec. II. Theoretical Background

A Naïve Bayes classifier may be defined as follows.[5] According to Bayes' Rule, the probability $p(c|E)$ that an example $E = (x_1, x_2, \dots, x_n)$ belongs to a class c is given by:

$$p(c|E) = \frac{p(c|E)p(c)}{p(E)}$$

For a binary classifier $C = \pm$, E belongs to the class $C = +$ if and only if the Bayesian classifier $f_b(E)$, follows the relation:

$$f_b(E) = \frac{p(C = +|E)}{p(C = -|E)} \geq 1$$

Assuming the attributes are independent and follow the relation:

$$p(E|c) = p(x_1, x_2, \dots, x_n|c) = \prod_{i=1}^{n} p(x_i|c)$$

the Naïve Bayes classifier$f_{nb}(E)$ may then be defined as:

$$f_{nb}(E) = \frac{p(C = +)}{p(C = -)} = \prod_{i=1}^{n} \frac{p(x_i|C = +)}{p(x_i|C = -)}$$

In this exercise, the binary classifier $C = \pm$ may be understood to represent workpieces with defects and workpieces with no defects.

---

[4] https://www.kaggle.com/gagazet/lead-legs-on-chipset/kernels

[5] Zhang, Harry. (2004). The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004. 2.

Sec. III. Algorithm Implementation and Development

For each set of images, I trained a Naïve Bayes model based on several preliminary treatments of the images. These treatments included raw images, Haar wavelet transformed images (i.e. approximation coefficients) of a second level Haar wavelet Transform, first level Symlet 4 wavelet transformed images, principal component amplitudes from singular value decomposition (SVD) of the raw images, principal component amplitudes from SVD of the Haar wavelet transformed images, and principal component amplitudes from SVD of the Symlet 4 wavelet transformed Images.

For the raw images, I reshaped the image data arrays into vectors and stored those vectors in a matrix for processing either with a training model or with SVD and then a training model.

For the Haar Wavelet transformed images, I first performed a second level Haar Wavelet Transform on each image and then reshaped the array of approximation coefficients into vectors and stored those vectors in a matrix for processing either with a training model or with SVD and then a training model.

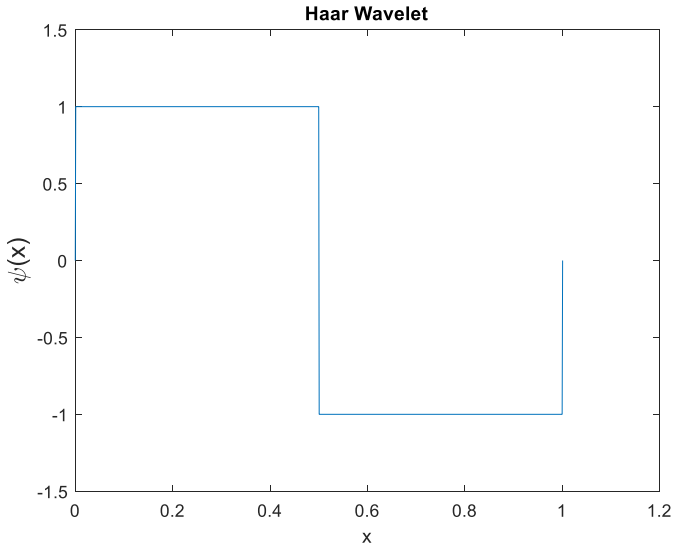
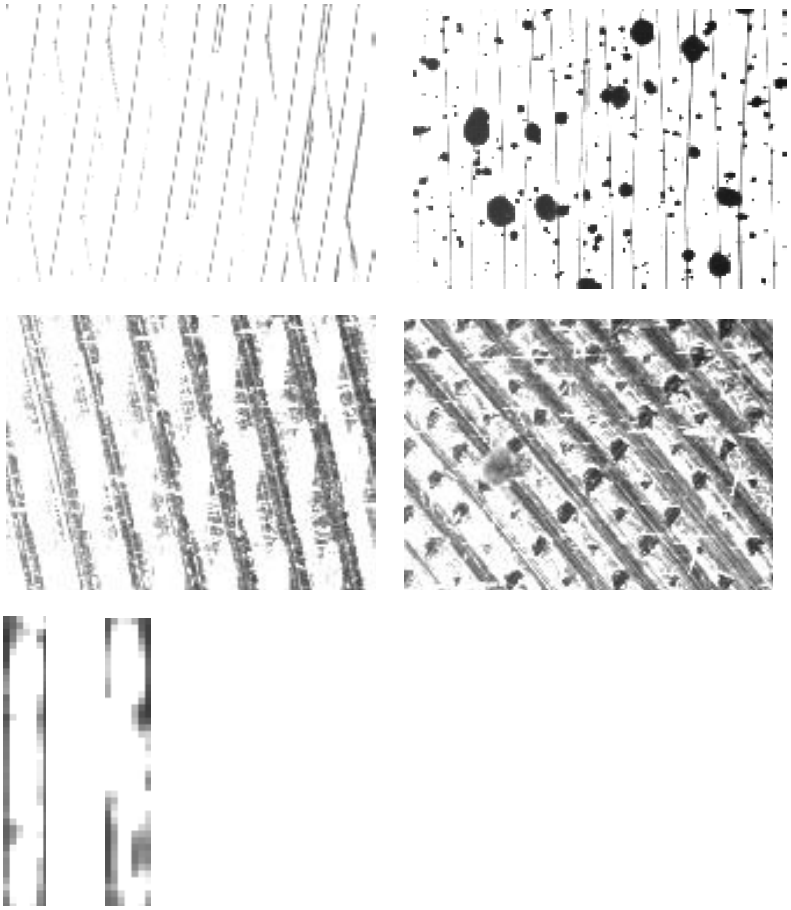


Figure 4 – Plot of Haar Wavelet



Figures 5 a), b), c), e), and f) – Left to right, top to bottom, Haar wavelet transformed images (approximation coefficients) of: a machined aluminum workpiece with no defect, machined aluminum with spray painted simulated defect, a machined aluminum workpiece with no defect, machined aluminum with graphite simulated defect, a lead leg with no defects, and a defective lead leg.

For the Symlet 4 Wavelet transformed images, I first performed a first level Discrete Wavelet Transform on each image with Symlet 4 wavelet and periodic extension. Then for each image, I reshaped each of the arrays of approximation coefficients, the horizontal detail coefficients, the vertical detail coefficients, and the diagonal detail coefficients into vectors and stored those vectors in series in a single column of a matrix for processing either with a training model or with SVD and then a training model.

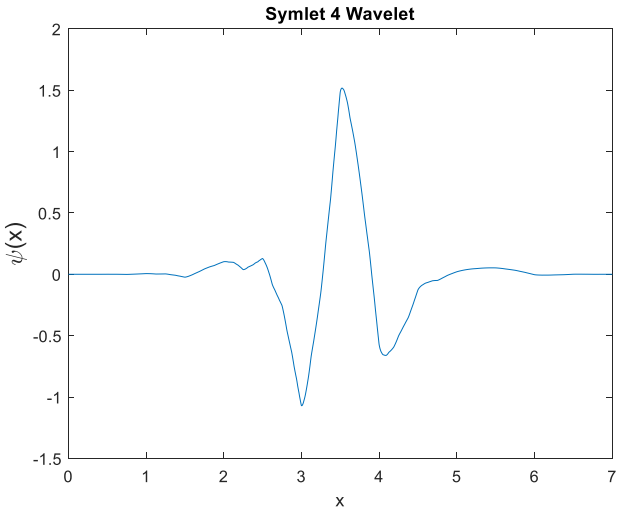
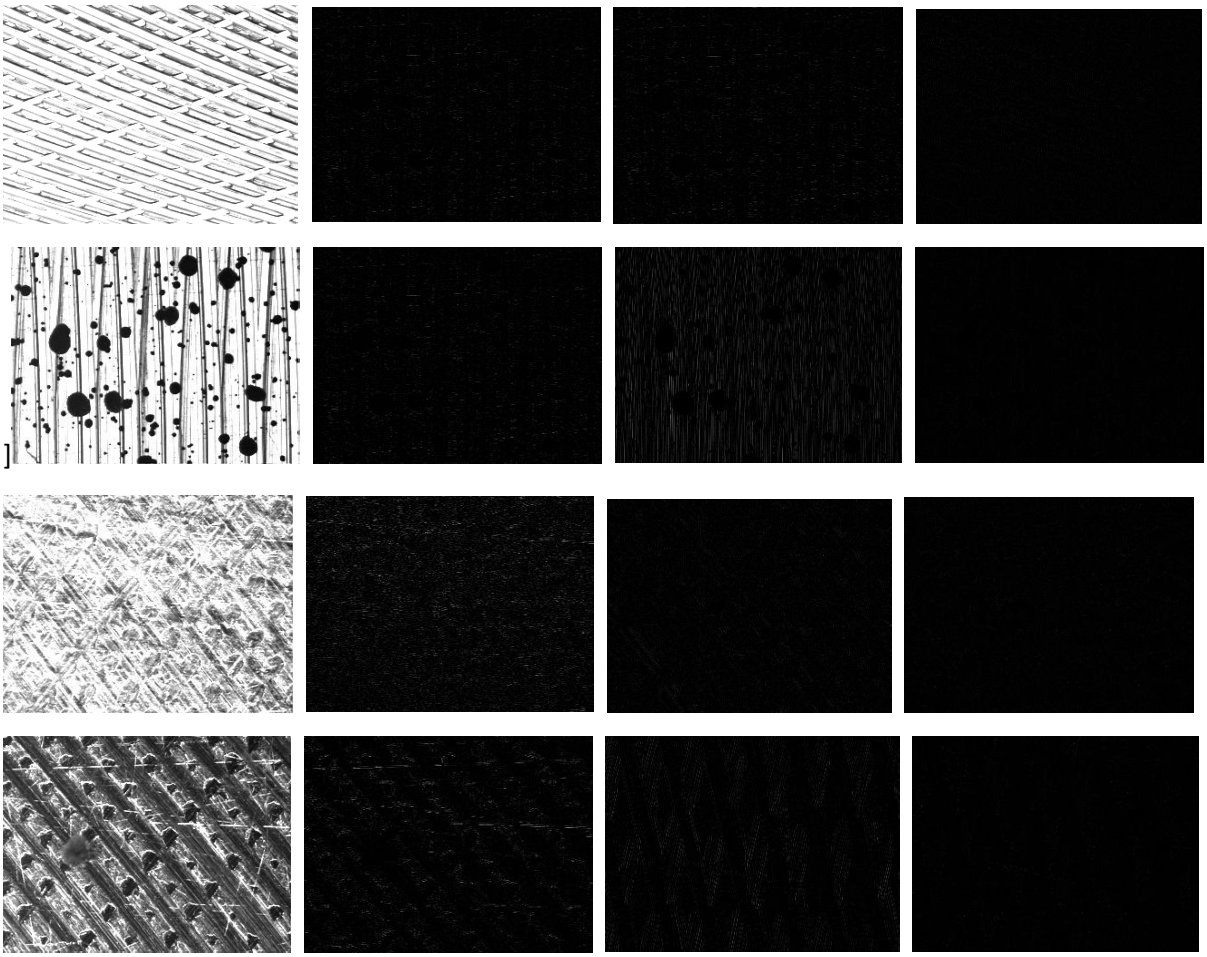
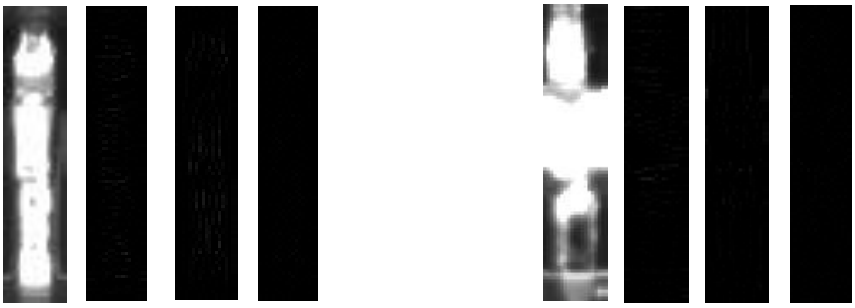


Figure 6 – Plot of the Symlet 4 Wavelet





Figures 7 a) to x) – Left to right, top to bottom, approximation coefficients, horizontal detail coefficients, vertical detail coefficients, and diagonal detail coefficients of level 1 Symlet 4 wavelet transformed images of: a machined aluminum workpiece with no defect, machined aluminum with spray painted simulated defect, a machined aluminum workpiece with no defect, machined aluminum with graphite simulated defect, a lead leg with no defects, and a defective lead leg.

I trained a Naïve Bayes classification model on subsets of the images including 40 images of defective and 40 images of non-defective aluminum workpieces, and 15 images of defective and 25 images of non-defective IC leads. I tested the Naïve Bayes classification model on subsets of the images including 10 images of defective and 10 images of non-defective aluminum workpieces, and 5 images of defective and 5 images of non-defective IC leads. For each case, I cross-validated the model with 20 trials of randomly selected training and test images.

Sec IV. Computational Results

I have included a table in Figure 8 that shows the overall accuracy for each method for each data set. I have included confusion matrices in Figures 9-11 for each method of each data set. No single method provided the most accurate results for each data set. For the spray-paint on aluminum data set, the raw images provided the most accurate results. For the graphite on aluminum data set, the Symlet 4 transformed images provided the most accurate results. For the IC lead legs data set, the raw images subject to SVD and the Symlet 4 transformed images both provided a surprising 100% accuracy.

	Spray-paint on Al	Graphite on Al	IC Lead Legs
Raw	91.8%	57.5%	98.0%
Raw with SVD	75.3%	70.0%	100%
Haar	90.8%	61.3%	98.5%
Haar with SVD	80.3%	73.0%	73.0%
DWT	59.3%	95.0%	100%
DWT with SVD	79.0%	64.3%	78.5%

Figure 8 – Table of overall accuracies of 20 cross-validation tests for each data set and each image treatment.

		Actual	
		Defect	No Defect
Predicted	Defect	188	12
	No Defect	21	179

		Actual	
		Defect	No Defect
Predicted	Defect	157	43
	No Defect	56	144

		Actual	
		Defect	No Defect
Predicted	Defect	161	39
	No Defect	40	160

		Actual	
		Defect	No Defect
Predicted	Defect	184	16
	No Defect	21	179

		Actual	
		Defect	No Defect
Predicted	Defect	118	82
	No Defect	81	119

		Actual	
		Defect	No Defect
Predicted	Defect	161	39
	No Defect	45	155

Figures 9 a), b), c), d), e), and f), left to right, top to bottom - Confusion matrices for machined aluminum with spray-paint: raw images, Haar wavelet transformed images, principal components from singular value decomposition (SVD) of the raw images, principal components from SVD of the wavelet transformed images, Symlet 4 wavelet transformed images, and principal components from singular value decomposition (SVD) of the Symlet 4 wavelet transformed images.

		Actual	
		Defect	No Defect
Predicted	Defect	130	70
	No Defect	76	100

		Actual	
		Defect	No Defect
Predicted	Defect	145	55
	No Defect	100	100

		Actual	
		Defect	No Defect
Predicted	Defect	155	45
	No Defect	86	114

		Actual	
		Defect	No Defect
Predicted	Defect	156	44
	No Defect	76	124

		Actual	
		Defect	No Defect
Predicted	Defect	192	8
	No Defect	12	188

		Actual	
		Defect	No Defect
Predicted	Defect	154	46
	No Defect	97	103

Figures 10 a), b), c), d), e), and f), left to right, top to bottom - Confusion matrices for machined aluminum with graphite: raw images, Haar wavelet transformed images, principal components from singular value decomposition (SVD) of the raw images, principal components from SVD of the wavelet transformed images, Symlet 4 wavelet transformed images, and principal components from singular value decomposition (SVD) of the Symlet 4 wavelet transformed images.

		Actual	
		Defect	No Defect
Predicted	Defect	100	0
	No Defect	4	96

		Actual	
		Defect	No Defect
Predicted	Defect	100	0
	No Defect	0	100

		Actual	
		Defect	No Defect
Predicted	Defect	100	0
	No Defect	3	97

		Actual	
		Defect	No Defect
Predicted	Defect	72	28
	No Defect	26	74

		Actual	
		Defect	No Defect
Predicted	Defect	100	0
	No Defect	0	100

		Actual	
		Defect	No Defect
Predicted	Defect	77	23
	No Defect	20	80

Figures 11 a), b), c), d), e), and f), left to right, top to bottom - Confusion matrices for lead legs of an IC: raw images, Haar wavelet transformed images, principal components from singular value decomposition (SVD) of the raw images, principal components from SVD of the wavelet transformed images, Symlet 4 wavelet transformed images, and principal components from singular value decomposition (SVD) of the Symlet 4 wavelet transformed images.

Sec. V. Summary and Conclusions

For the spray paint on aluminum images, the raw images gave the best results, but required a heavy processing time, i.e. about 13 hours for 20 cross validation trials. The Haar wavelet transformed images gave comparable results, but with considerably less processing time, i.e. about 2 minutes.

For the graphite on aluminum images, the Symlet 4 transformed images gave the best results, with much higher accuracy than any other images. This class of defect was more difficult to model than the spray-paint on aluminum. With the notable exception of the Symlet 4 transformed images, every other method gave lower accuracy compared to the spray-paint on aluminum data set.

For the Kaggle IC lead images, the principal components of the raw images and the Symlet 4 transformed images gave a surprising 100% accuracy, while each of the other methods gave high accuracy. The principal components of the Haar wavelet transformed images gave significantly lower accuracy than the other methods, but it was still more accurate than most of the methods for the aluminum images. This type of defect was much easier to model than the aluminum defects. It is worth

noting that these images were all neatly cropped, and the defects took up a very significant proportion of the field of view of the image compared to the aluminum defects.

It should be appreciated that I only tested two types of wavelet transforms and a thorough comparative study of different wavelets is warranted for various classes of defects. The Haar level 2 wavelet provided high computational speed and strong accuracy for the spray-paint on aluminum and the IC lead legs data sets, while the Symlet 4 wavelet provided strong accuracy for the otherwise difficult graphite on aluminum data set.

Appendix A MATLAB functions used and brief implementation explanations

confusionmat() – This returns a confusion matrix for predicted and actual data.

dwt() = This returns discrete wavelet transform of an image with a specified wavelet type.

fitcnb() – This fits a Naïve Bayes model to the training data.

haar2() – This returns a Haar level 2 wavelet transform of an image.

imread() – This reads an image into a MATLAB array.

predict() – This predicts a set of classifications based on the trained model.

randperm() – This returns a random permutation of a set of numbers.

reshape() – This reshapes an array, e.g. a vector to a matrix or vice versa.

svd() – This calculates the three matrices of SVD based on the input matrix.

Appendix B MATLAB code

```
clear all; close all; clc


%First Part: Prepare an initial data set from a set of images (defective, or
non-defective).  Manually save the variable dataSet as a .mat file for use in
the second part.

z=[];

list= dir('C:\Users\Ashley\Documents\MATLAB\Kaggle No Defect\*');
dataSet = [];
for j=3:length(list)
    currentFileName = strcat('C:\Users\Ashley\Documents\MATLAB\Kaggle No
Defect\', list(j).name);
    y = imread(currentFileName);
    y = imresize(y,[180,28]);
    z = reshape(y,[1,numel(waveletY)]);
    %Optional Haar wavelet transform
    %waveletY = haart2(y,2);
    %z = reshape(waveletY,[1,numel(waveletY)]);
    %Optional Symlet 4 wavelet transform
    %[cA,cH,cV,cD] = dwt2(y,'sym4','mode','per');
    %z1 = reshape(cA,[1,numel(cA)]);
    %z2 = reshape(cH,[1,numel(cH)]);
    %z3 = reshape(cV,[1,numel(cV)]);
    %z4 = reshape(cD,[1,numel(cD)]);
    %z = [z1,z2,z3,z4];

    dataSet = [dataSet;z];
end
```

```
%Second Part: Train and test a machine learning model on images.


clear all; close all; clc

load('Defect1.mat')
defectSet = im2double(dataSet);
load('NoDefect1.mat')
noDefectSet = im2double(dataSet);

totalConfusionMatrix=zeros(2);
for numberOfTrials = 1:9
q1=randperm(50);
q2=randperm(50);
xtrain=[defectSet(q1(1:40),:); noDefectSet(q2(1:40),:)];
xtest=[defectSet(q1(41:end),:); noDefectSet(q2(41:end),:)];
ctrain=[ones(40,1); 2*ones(40,1)];
actual=[ones(10,1);2*ones(10,1)];
%Apply Naive Bayes Classifier and test the model
nb=fitcnb(xtrain,ctrain);
pre=nb.predict(xtest);
currentConfusionMatrix = confusionmat(actual,pre);
totalConfusionMatrix = totalConfusionMatrix + currentConfusionMatrix;
end

totalConfusionMatrix = totalConfusionMatrix + currentConfusionMatrix;

%Sound an alarm when this is done.  Some sets took a very long time to
process.
load handel;
sound(y,Fs);



%Second Part: Train and test a machine learning model on principal components
of images.


clear all; close all; clc

load('Defect1.mat')
defectSet = im2double(dataSet);
load('NoDefect1.mat')
noDefectSet = im2double(dataSet);
allData=[defectSet;noDefectSet];

[u,s,v] = svd(allData.', 'econ');

totalConfusionMatrix=zeros(2);
for numberOfTrials = 1:19
defectSet = v(1:50,:);
noDefectSet = v(51:100,:);
q1=randperm(50);
q2=randperm(50);
xtrain=[defectSet(q1(1:40),:); noDefectSet(q2(1:40),:)];
```

```
xtest=[defectSet(q1(41:end),:); noDefectSet(q2(41:end),:)];
ctrain=[ones(40,1); 2*ones(40,1)];
actual=[ones(10,1);2*ones(10,1)];
%Apply Naive Bayes Classifier and test the model
nb=fitcnb(xtrain,ctrain);
pre=nb.predict(xtest);
currentConfusionMatrix = confusionmat(actual,pre);
totalConfusionMatrix = totalConfusionMatrix + currentConfusionMatrix;
end

totalConfusionMatrix = totalConfusionMatrix + currentConfusionMatrix;
```