

Data Trimming Methods for Image Classification of HYDICE Hyperspectral Data

John A. Marin

**Department of Electrical Engineering and Computer Science
United States Military Academy
West Point, New York 10996
jmarin@usma.edu**

Kirk C. Benson

**Department of Mathematics
United States Military Academy
West Point, New York 10996
kirk-benson@us.army.mil**

**Andrew S. Allen IV, James J. Hickman
United States Corps of Cadets
West Point, New York 10997**

ABSTRACT

This paper presents the preliminary results concerning the applicable and appropriate use of statistical methods versus machine learning methods in band selection and the subsequent classification of HYDICE satellite imagery. Specifically, it compares a genetic algorithm-based approach to a correlation coefficient based approach in the selection of data used for classification of a HYDICE image. Additionally, it uses statistics and an image-rendering tool for quantitative and qualitative analysis of neural network topology and image value respectively. Finally, the generalization of these techniques and tools for other neural network applications is discussed.

1. Introduction

This is an interim report of a multiyear project focused on testing the capabilities of machine learning methods to classify hyperspectral imagery. Data for this endeavor consisted of a HYDICE image of Fort Hood, Texas encompassing 210 bands of spectral bandwidth sliced at 10nm with a 2.5 spatial resolution. The training data represented 2816 pixels.

In this report we explore data trimming in a preprocessing pursuit of greater speed and accuracy of given genetic algorithm and neural network algorithms. In particular we tested the application of a neural network using the software NeuralSim to create a model that can take any given terrain and classify it quickly and accurately for use by a military commander. Our efforts indicate that linear

dependence in excess of 70% is unnecessary for robust classification of terrain. Reference [1] describes the data set and collection process.

The remainder of this paper is organized as follows. In section 2 we discuss previous efforts that this paper builds upon. In section 3 we discuss terrain classification. Section 4 describes the methodology for band selection from the data set. Section 5 provides a summary and recommendations for future work.

2. Previous Efforts

In an endeavor to build on previous efforts, the authors conducted a comprehensive literature research. The goal was to learn how others attacked the problem, identify mistakes made previously, and obtain a better understanding of the problem. Previous researchers at the United States Military Academy explored modification of the genetic algorithmic data selection with focus on end member selection [2]. Additionally, other research such as George Washington University attempts to use neural networks to analyze the sounds an engine made in an attempt to classify potential problems was helpful. A complete listing of these sources is enclosed in the bibliography.

3. Terrain Classification

Several studies attest to the capability of neural networks to outperform traditional classification methods for feature extraction and classification in multispectral satellite images [3,4]. We used the

backpropagation algorithm available with the NeuralSIM software package to train a neural network. This software package builds the network adding nodes to the hidden layer until degradation in performance is observed on the test data set.

A neural network is fundamentally a network of mathematically interconnected input/output elements; these elements, known as nodes, are connected through numerically weighted arcs, and parallel sections of nodes are grouped into sequential layers—groups of nodes that are mathematically manipulated simultaneously. Each individual node (except for the input nodes) will receive input from one or more nodes from a previous layer. For example, a node in layer J+1 will receive input from nodes in layer J. The value of the incoming nodes from layer J is multiplied by their respective arc weights to the node in layer J+1.

The receiving node sums up the products (essentially producing the dot product of the node and arc weight vectors) and sends the result through a transfer function. The transfer function is selected by the programmer—it can be bitwise, discrete, or continuous; one popular transfer function is the sigmoid function because it has an easily calculable derivative. The result of the transfer function is multiplied by new arc weights and sent to nodes in layer J+2. Eventually, the network will reach the last layer (output layer), and the transfer function values from nodes in the output layer become calculated values for the approximated function. This entire process is known as a forward pass.

The calculated output is compared to the target output, and the difference between the two is the error. Using a gradient search and sum of the squared error, the network calculates and applies a weight correction to each of the arc weights, layer-by-layer, in reverse-sequential order. This operation, known as the backward pass, helps adjust the arc weights so that the calculated output will converge upon the target output. The forward-backward passes are continuously repeated until some sort of stopping criterion is met.

4. Band Analysis

Band selection began with a cursory examination of the data to ascertain if similarities existed between the spectral values. In order to select the bands for a subsequent classification, we used both a genetic algorithm approach and software assisted correlation coefficient trimming technique. We utilized a training set of 2,816, (each of which was represented

by 210 scaled, 8-bit integers) pixels divided into seven categories to include bright roads (br), lighter roads (lr), sparse grass (lg), medium grass (mg), dense grass (hg), trees (tr), and water (wa). The following sections discuss both the genetic algorithm and correlation coefficient approach.

Genetic Algorithm Approach

Genetic algorithmic approaches allow machines to simulate nature's process of natural selection and appear to work on a large class of interesting problems for which no reasonably fast algorithms exist. We employed a commercially available software package called NeuralSIM [5].

NeuralSIM employs a genetic algorithm to select variables and uses a backpropagation neural network for the subsequent classification or prediction. NeuralSIM uses multiple regression with a SoftMax shaping function [5,6] as the fitness function for classification problems. NeuralSIM allows the user to control several parameters affecting the genetic algorithm, such as population size, crossover probability, and parent selection (the user can select uniform, fitness, or rank independently for either parent).

The user must select several options to guide NeuralSIM toward a solution. These options include amount of noise in data (clean (1), moderate (2), noisy (3), very noisy (4)). Noise is the behavioral consistency of the data. The next option is the level of data transformation (scale (1), superficial (2), moderate (3), comprehensive). The user then defines the complexity of variable selection (none (1), superficial (2), moderate (3), comprehensive (4), and exhaustive (5)). Lastly, the extent of network search (none (1), superficial (2), moderate (3), comprehensive (4), and exhaustive (5)) is defined. The numbers after each level were used to simplify the naming convention for output.

In this effort, a variety networks were built based upon these changing these variables. Initially, we considered building a network for each possible combination of variables. Since there were four variables, each with four or five possible values, the total number of networks that must be built in this approach is 400. We employed a testing methodology focused on sensitivity analysis to provide both direction and efficiency in this effort.

The first network built was called "3333". Therefore, NeuralSIM treated the data as noisy, performed a moderate data transformation, moderate variable

selection, and moderate network search. Next, the first 3 variables were made constant, and only the level of network search was changed. With networks built in this manner, it was possible to compare them to determine which level of network search best solved the problem of terrain classification. At this point the following networks were built: “3332”, “3333”, “3334”, and “3335”.

The determination of which network provided the best terrain classification involved using several predefined Measures of Effectiveness (MOE). These included comparison of network topology (simple is better), the highest r^2 value, lowest fitness function value, and a subjective opinion on image clarity. In this case, 3335 had a simple topology, the highest r^2 value, lowest fitness function value, and a generally coherent picture compared to the other network outputs.

The fitness function value was determined by applying a scaled penalty for incorrect classification of given terrain. For example, if a road was classified as water it carried a penalty of 5 while low grass being classified as medium grass carried a lower penalty of 1. Multiplying these values with the number of incorrect terrain classifications produced comparable fitness values. Figure 1 details the scaled penalties and Figure 2 shows the confusion matrix for the 3335 network.

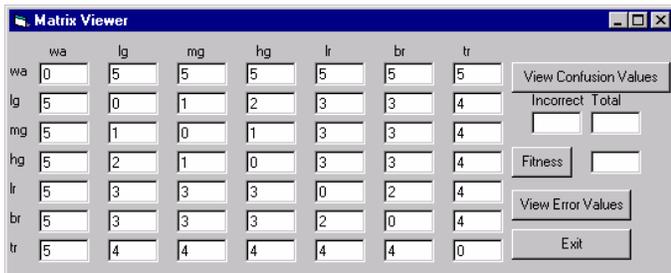


Figure 1: Penalty values for incorrect classification

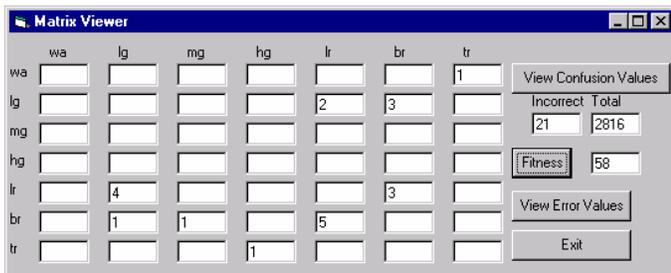


Figure 2: Confusion matrix for 3335 network

Next, having determined the appropriate level of network search, this variable was set constant (5), and new networks were built which varied the level of network search. The next networks built were “3325”, “3335”, “3345”, and “3355”. Applying the same MOEs indicated which of these networks best classified the terrain and, therefore, which level of variable selection was most appropriate. This process was continued for the last two variables.

The fitness function values for the various networks ranged from 58 to 290, and the number of incorrect guesses ranged from 21 to 104. Logically, the chosen network falls at the low end of these numbers. The final decision in the selection process considered three networks. These were 3425, with 25 incorrect guesses out of 2816 possible, a fitness function value of 60, and an r^2 of .9889; 2425, with 26 incorrect, fitness function value of 60, and r^2 of .9889; and 3335, with 21 incorrect, fitness function value of 58, and r^2 of .9819. We selected network 2425 (that is, moderately noisy data, comprehensive data transformation, superficial variable selection, and exhaustive network search) as the best network.

Network	2425	3225	3235	3322	3323	3325	3332
Incorrect	26	87	45	67	57	30	27
Fitness Function	60	230	125	189	161	84	78
Network	3333	3334	3335	3345	3355	3425	4425
Incorrect	24	23	21	31	104	25	62
Fitness Function	70	65	58	92	290	60	165

Table 1: Fitness function and number of incorrect classifications for considered networks

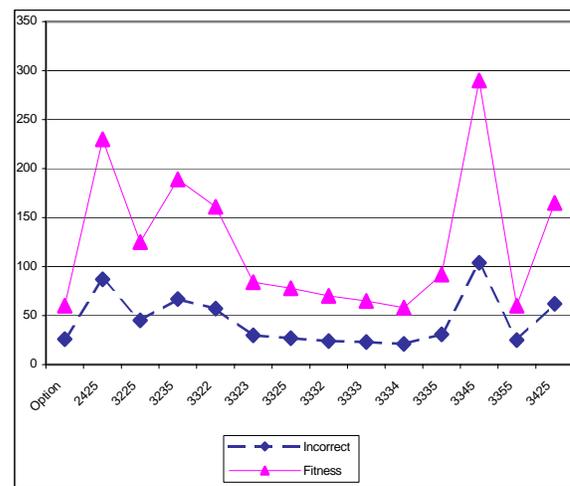


Figure 3: Graph of fitness function and number of incorrect classifications for considered networks

Additionally, based on qualitative analysis, the picture rendered by the 2425-network was more coherent than the pictures created by the other networks. Another discriminating factor was that this network did not predict water incorrectly on the training set, something that is very important to the ultimate goal of determining trafficability. Finally, the topology was simple for this network, with 13 inputs, and 8 nodes in the hidden layer.

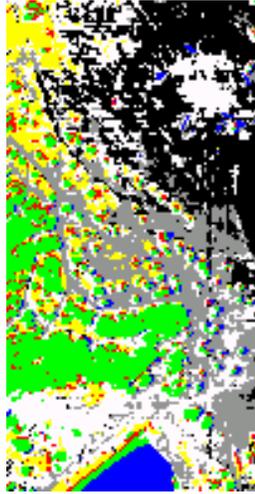


Figure 4: Network 2425 Image

Correlation Coefficient Analysis

During initial analysis of the data set, clear similarities were noticed in some of the spectral bands - some were even identical. Additionally, the data set of 210 spectral bands took over 24 hours in some cases to run on PIII-500 MHz computers. These were two motivating factors to reduce the data set. The computational expense of using similar bands was thought to be inefficient.

The researchers developed a software application to aid in data trimming based on correlation coefficients. This program allows the user to select a particular correlation level (0.85, for example), and then trims the data to discard columns with a correlation greater than or equal to that tolerance. For example, if five columns have a correlation of 0.85, the program will remove all but one. The remaining data consists of columns that have a lower correlation than the user's tolerance input.

The data was trimmed beginning with high correlation values from 0.95 to 0.99 in 0.01 increments. The step size was decreased to 0.05 for correlation values from 0.50 to 0.95. Generally, each new set contained fewer columns than the original

210-band set. Table 2 below details the number of columns included in the data set at set correlation coefficient values.

Correlation	50%	55%	60%	65%	70%	75%	80%
# Columns	14	26	30	40	54	84	108
Correlation	85%	90%	95%	96%	97%	98%	99%
# Columns	140	144	144	148	148	154	162

Table 2: Columns remaining after data trimming

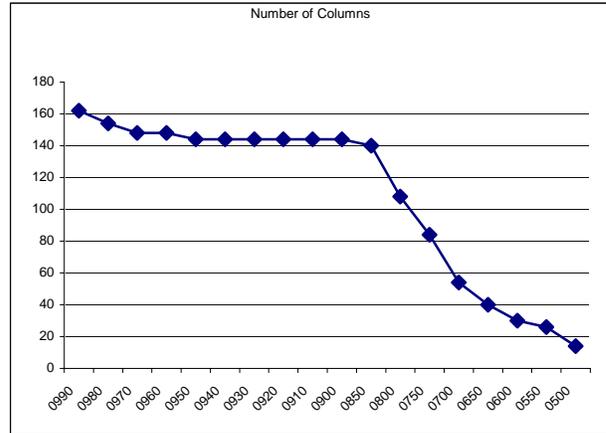


Figure 5: Graph of columns remaining after data trimming

The main goal was to rebuild our chosen network (2425) with the trimmed columns of known-output data set, then run the trimmed columns of the larger data set with unknown output. Further analysis using the aforementioned MOEs will determine how trimming the data affected network error.

At this point, 14 new 2425 networks were built off of the 14 trimmed data sets: 0.99 correlation, 0.98, 0.97, 0.96, 0.95, 0.90, 0.85, 0.80, 0.75, 0.70, 0.65, 0.60, 0.55, and 0.50. Table 3 details the number of incorrect terrain classifications and the fitness function value for these networks.

Correlation	# Incorrect out of 2816	Fitness Function
0500	298	923
0550	73	205
0600	36	107
0650	36	108
0700	13	42
0750	11	33
0800	22	60
0850	28	81
0900	26	69
0960	18	46
0980	11	31

Table 3: Fitness function and number of incorrect terrain classifications for trimmed data networks

5. Summary and Future Work

Key features of this research include the application of genetic algorithms and backpropagation neural networks in the classification of HYDICE satellite imagery. Reduction of the data set through the use of correlation coefficient analysis provides an interesting direction for future endeavors. Specifically, we have shown that some facets of linear dependence can assist in the solution of non-linear type problems. An additional feature of this research is the incorporation of different Measures of Effectiveness that allow for objective comparison of different solutions sets.

Future work for this project includes optimization of the neural network code to speed processing time. This could take the form of writing specific software solving this and associated classification problems.

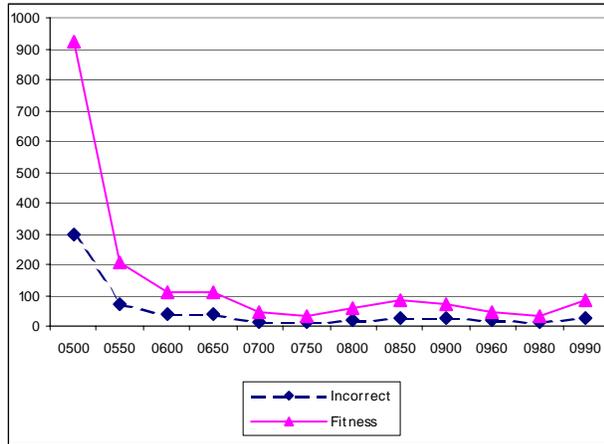


Figure 6: Fitness function and number of incorrect terrain classifications for trimmed data networks

The results from running the smaller data set showed that, in some cases, the fitness function values from the trimmed data were actually lower than the fitness function value from our untrimmed 2425 network. The three data sets that had the lowest fitness function values were 0.70, 0.75, and 0.98 correlation coefficient. The 0.98 correlation data set contained over 150 columns, but the 0.70 and 0.75 correlation data sets had 54 and 84 columns, respectively. R^2 values for all three were above 0.98. We selected the 0.70 trimmed data set because it had only 54 columns of input data.

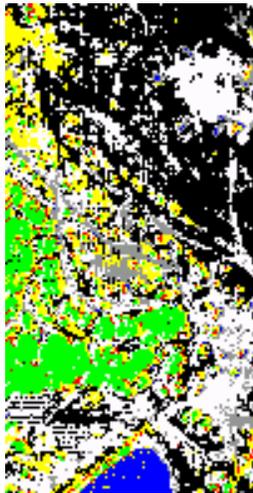


Figure 7: Network image of 0.70 correlation coefficient trimmed data set

References

- [1] McKeown, D., S. Cochran, S. Ford, C. McGlone, J. Shufelt and D. Yaocum, "Fusion of HYDICE Hyperspectral Data with Panchromatic Imagery for Cartographic Feature Extraction," *IEEE Transactions on Geoscience and Remote Sensing*, 37 1999, pp. 1261-1278.
- [2] Marin, John A., J. Brockhaus, J. Rolf, J. Shine, J. Schafer, and A. Balthazor, "Assessing Data Selection and Image Classification Techniques on HYDICE Hyperspectral **Data (Sir – need help to finish this one)**"
- [3] Bischof, H., W. Schneider, and A.J. Pinz. "Multispectral Classification of Landsat-Images using Neural Networks", *IEEE Transactions on Geoscience and Remote Sensing*, 30, 1992, pp. 482-489.
- [4] Marin, J.A., "An Inductive Approach to the Extraction of Roads from Multispectral Satellite Images", *Doctoral Dissertation*, School of Engineering and Applied Science, University of Virginia, Charlottesville, VA, Aug, 1995.
- [5] *NeuralSIM*, Technical Publications Group, Aspen. Inc., Pittsburgh, PA, 1999.
- [6] Bishop, C. *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

APPENDIX A

Microsoft Excel - training

File Edit View Insert Format Tools Data Window NeuralSIM Help

Arial 10 B I U \$ % , +.00 +.0

HA1 = b208

	GO	GP	GQ	GR	GS	GT	GU	GV	GW	GX	GY	GZ	HA	HB	HC	HD
1	b196	b197	b198	b199	b200	b201	b202	b203	b204	b205	b206	b207	b208	b209	b210	Cat
2	206	201	201	192	191	208	195	204	175	191	200	178	169	198	198	lr
3	189	176	187	186	191	185	195	187	207	191	163	255	169	169	113	lr
4	48	24	22	24	53	38	60	59	95	89	72	102	198	226	226	tr
5	43	24	22	37	31	54	60	68	95	102	91	178	169	169	141	tr
6	35	19	13	24	21	46	45	68	95	51	91	127	169	169	198	tr
7	30	9	4	18	21	38	45	51	79	76	91	127	113	198	141	tr
8	43	14	17	24	31	38	52	51	79	89	109	153	169	198	198	tr
9	39	14	17	24	21	46	52	68	95	63	109	153	141	226	169	tr
10	39	14	13	24	10	54	45	59	111	89	109	127	198	141	226	tr
11	39	14	8	18	31	38										
12	70	49	53	55	53	77										
13	74	53	62	68	63	69										

Input Variable Selection Level

Variable Selection Level

Do you really need all of those inputs?
Select the depth of search for key sets of variables. 'Comprehensive' works well for a wide variety of problems.

- comprehensive variable selection
- no variable selection
- superficial variable selection
- moderate variable selection
- comprehensive variable selection
- exhaustive variable selection

Advanced Level Parameters

Network Type: adaptive gradient

Save Last Net

Architecture: 7-2-7/0.9563

Regularization

Hidden: 0.05 Output: 0.01 Auto Tune Wt. Decay

20 1 Noise (Kalman)

Output

SoftMax Function

average classification rate Evaluation

2 MPE Direct Connections

Hidden Architecture

1 2 Min/Max Increment

30 Max Layer Size

Iterate Through Vector

Variable Selection

50 Maximum Generations

7 Patience

100 Quantization Levels

3 Cross-validation Sets

2 Population Factor

Perform Cascade VS

Heuristics

Tolerance

Net Node Hidden

0.003 0.003 0.003

1 Maximum Networks

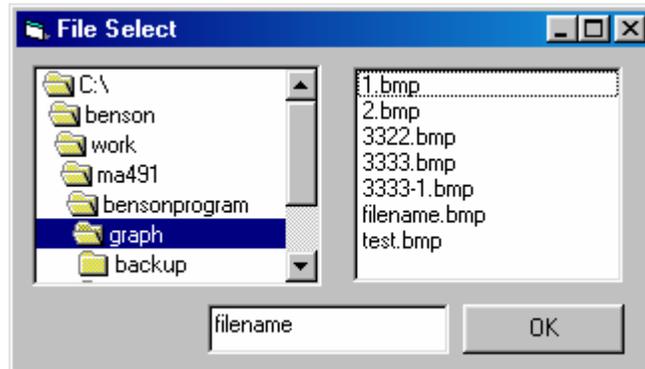
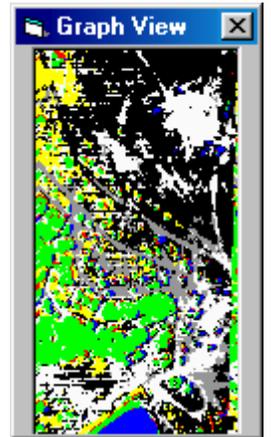
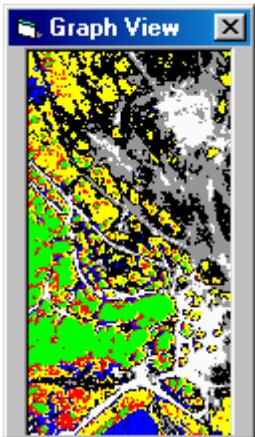
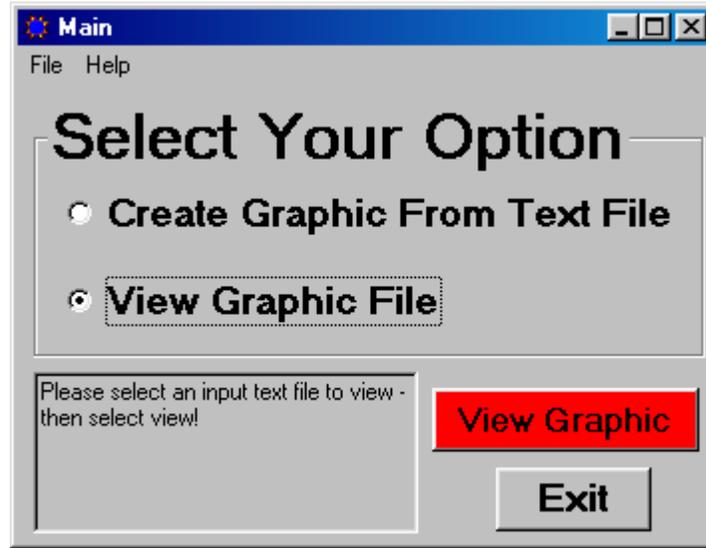
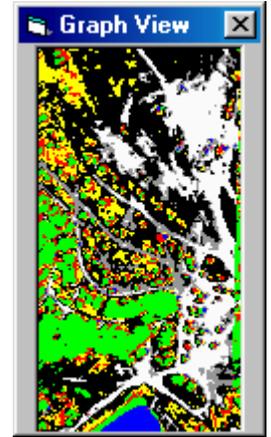
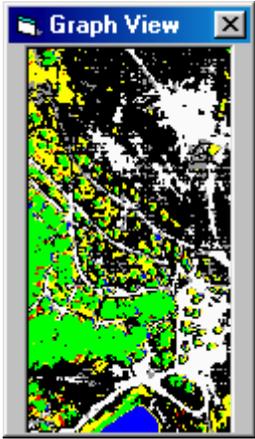
1 Patience

Set Seed... Write Weights...
Training Sets... Transforms...
OK Cancel Help

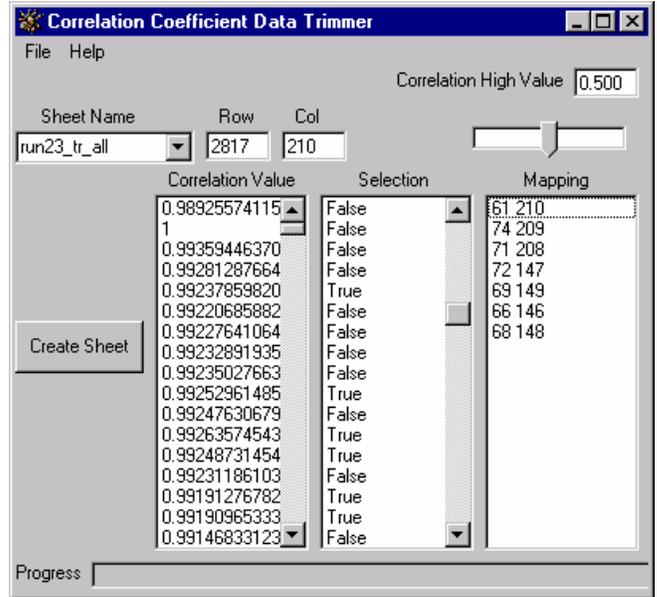
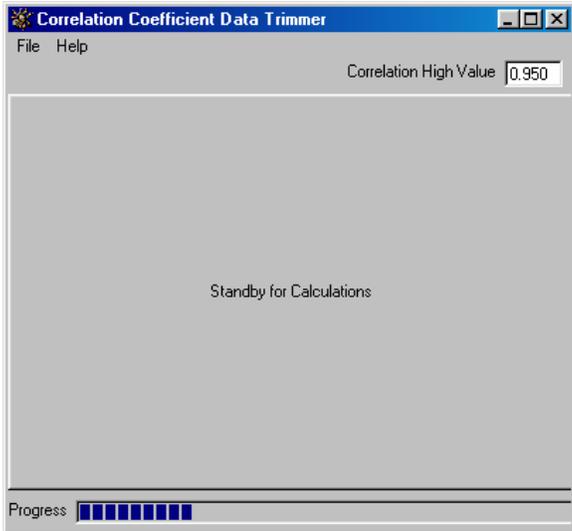
run23_tr_all

Ready

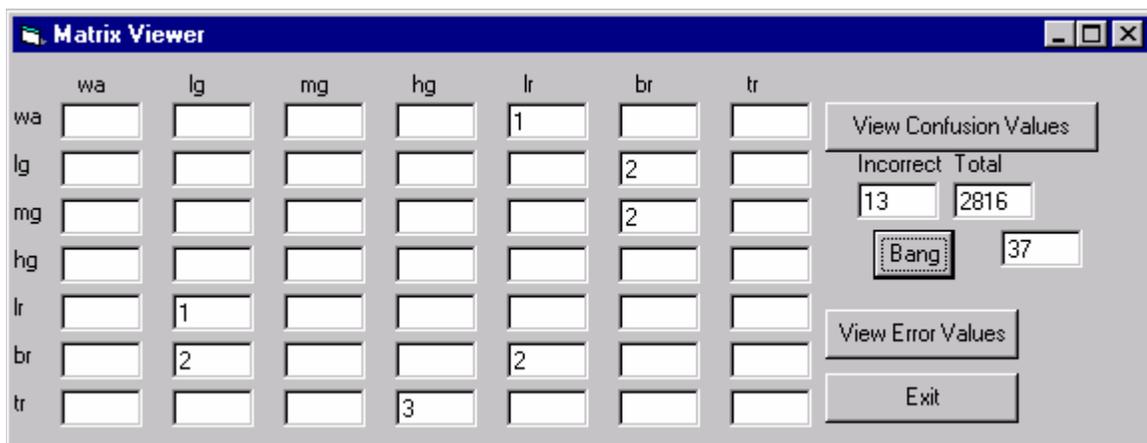
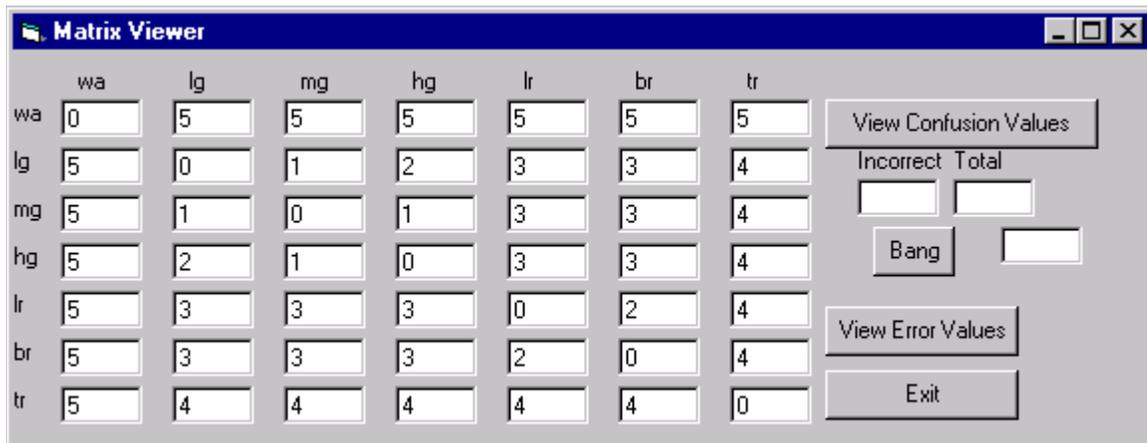
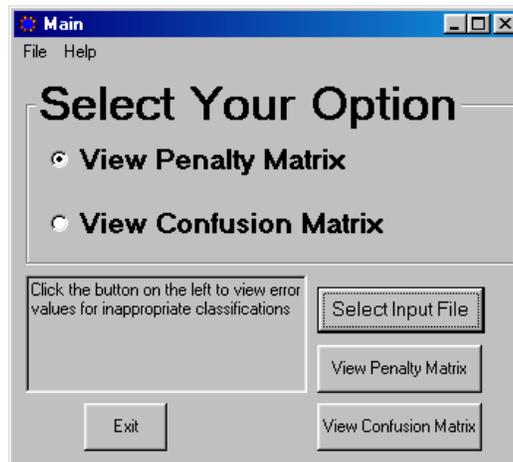
APPENDIX B



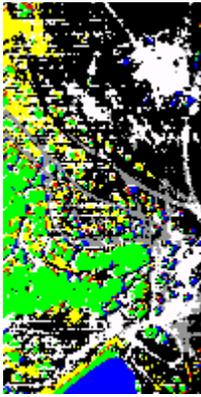
APPENDIX C



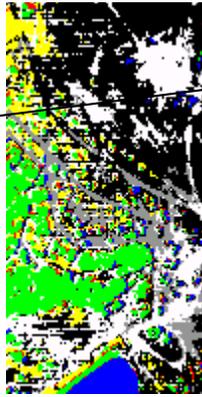
APPENDIX D



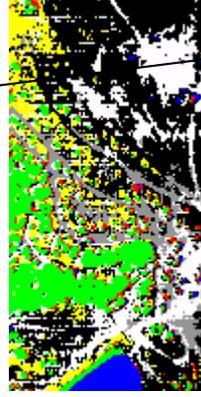
APPENDIX E



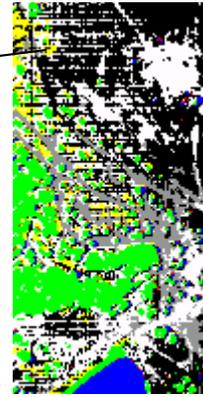
3332



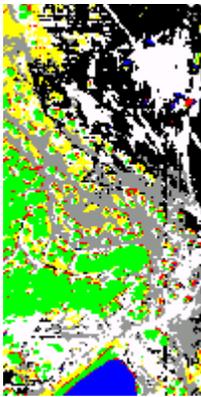
3333



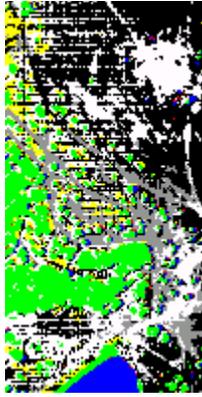
3334



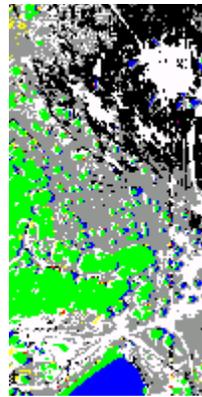
3335



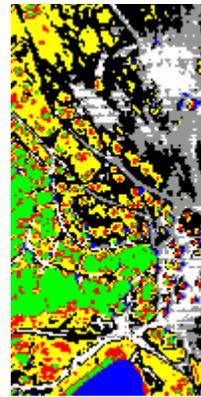
3325



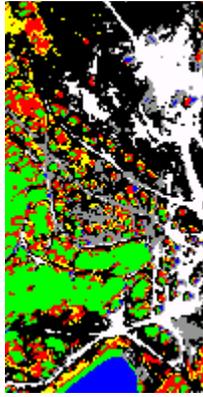
3335



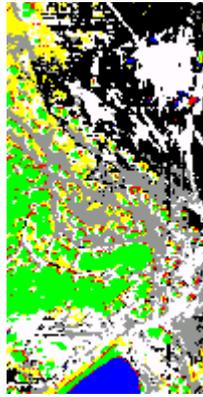
3345



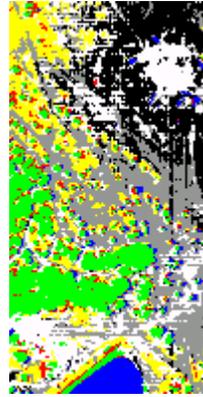
3355



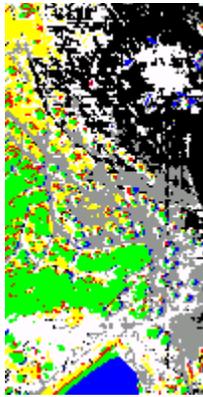
3225



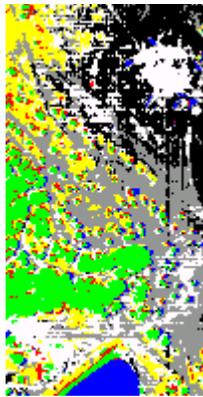
3325



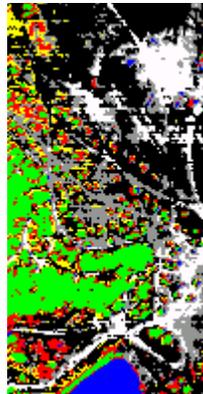
3425



2425



3425

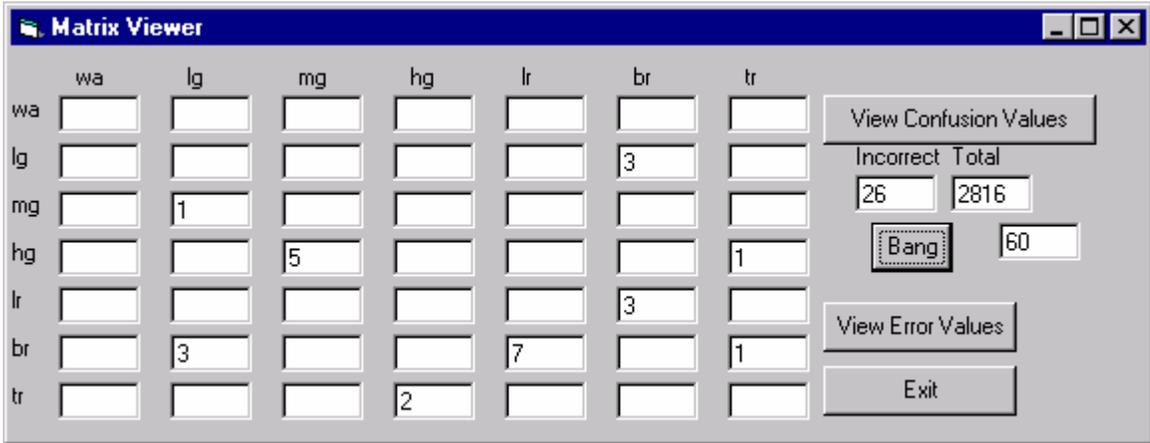


4425

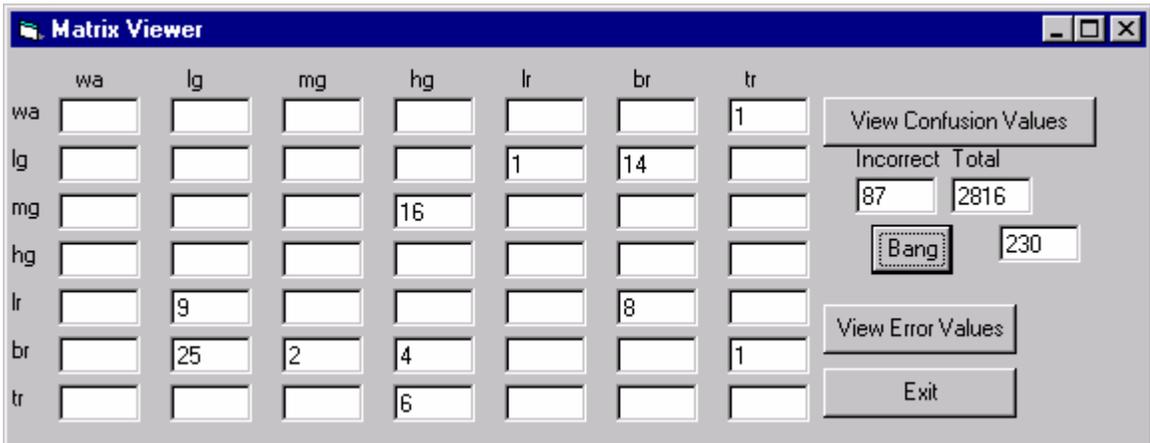


APPENDIX F

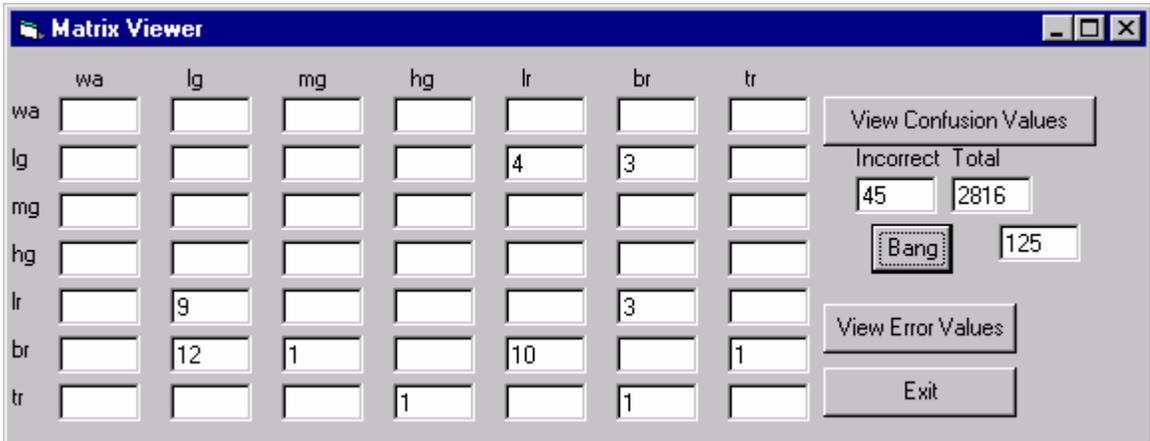
2425



3225



3235



3322

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa							2
lg			2		19	7	
mg				3			
hg							1
lr		3				3	
br		10	6	1	7		1
tr				2			

View Confusion Values

Incorrect Total
67 2816

Bang 189

View Error Values

Exit

3323

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa							2
lg			2		7	9	
mg				3			
hg							1
lr		4				4	
br		11	6	1	4		1
tr				2			

View Confusion Values

Incorrect Total
57 2816

Bang 161

View Error Values

Exit

3325

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa				1			
lg						5	
mg						1	
hg			1				
lr		3				4	
br		7	2		4		1
tr				1			

View Confusion Values

Incorrect Total
30 2816

Bang 84

View Error Values

Exit

3332

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa							1
lg					3	2	
mg							
hg							
lr		9				3	
br		3			4		1
tr				1			

View Confusion Values

Incorrect Total
27 2816

Bang 78

View Error Values

Exit

3333

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa							1
lg					2	2	
mg							
hg							
lr		7				3	
br		2			4		1
tr				1		1	

View Confusion Values

Incorrect Total
24 2816

Bang 70

View Error Values

Exit

3334

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa				1			
lg					2	3	
mg							
hg							
lr		6				3	
br		1			5		1
tr				1			

View Confusion Values

Incorrect Total
23 2816

Bang 65

View Error Values

Exit

3335

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa							1
lg					2	3	
mg							
hg							
lr		4				3	
br		1	1		5		
tr				1			

View Confusion Values

Incorrect Total

21 2816

Bang 58

View Error Values

Exit

3345

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa				1			
lg					3	3	
mg							
hg							2
lr		4				4	
br		5			5		1
tr				3			

View Confusion Values

Incorrect Total

31 2816

Bang 92

View Error Values

Exit

3355

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa				1			
lg			12		44	2	
mg				1			
hg							4
lr		9				3	
br		8	9	2	4		
tr				5			

View Confusion Values

Incorrect Total

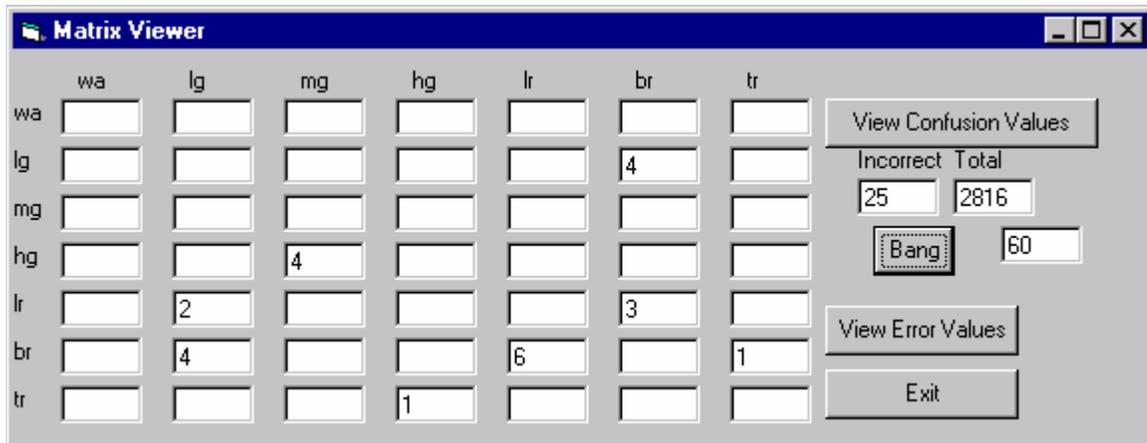
104 2816

Bang 290

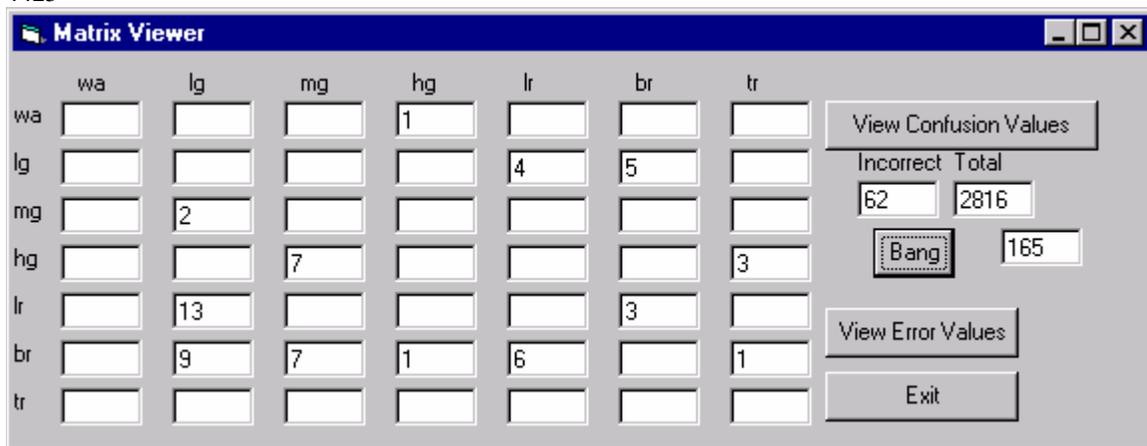
View Error Values

Exit

3425



4425



APPENDIX G

0500

	wa	lg	mg	hg	lr	br	tr	
wa							1	<input type="button" value="View Confusion Values"/> Incorrect Total 298 2816 <input type="button" value="Bang"/> 923 <input type="button" value="View Error Values"/> <input type="button" value="Exit"/>
lg					64	6	1	
mg				33			10	
hg			12				10	
lr		38				2	8	
br		14	1		5		4	
tr	2	34	15	11	16	11		

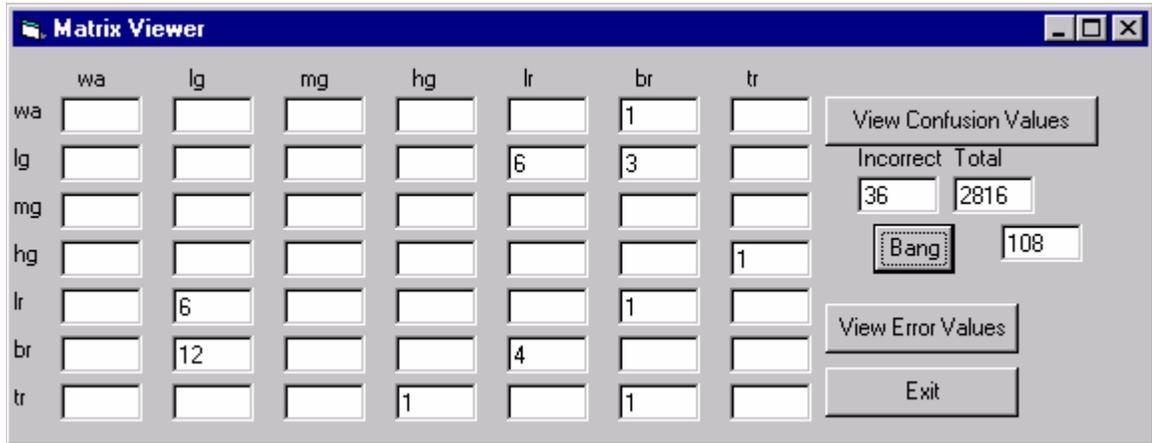
0550

	wa	lg	mg	hg	lr	br	tr	
wa			2					<input type="button" value="View Confusion Values"/> Incorrect Total 73 2816 <input type="button" value="Bang"/> 205 <input type="button" value="View Error Values"/> <input type="button" value="Exit"/>
lg					15	2		
mg				2				
hg			1				2	
lr		9				4		
br		12	4		15		2	
tr				3				

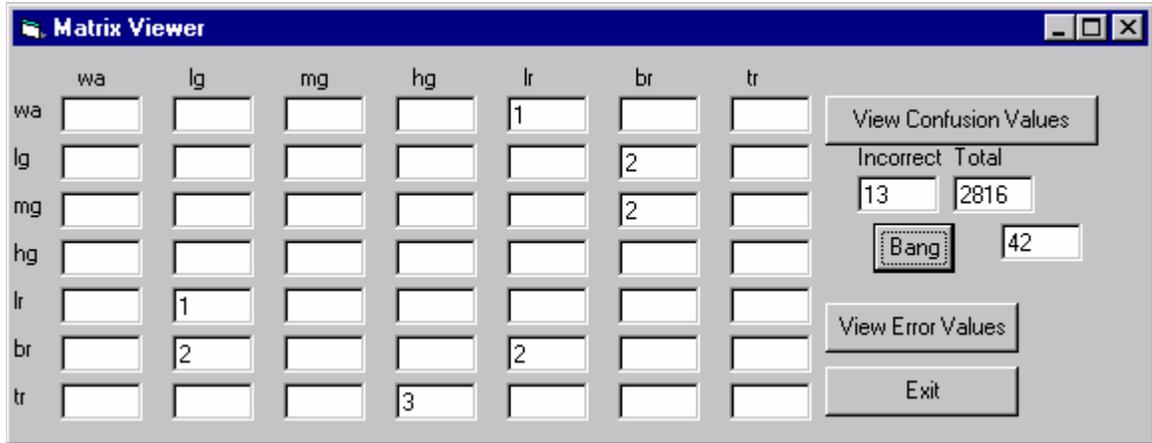
0600

	wa	lg	mg	hg	lr	br	tr	
wa			1					<input type="button" value="View Confusion Values"/> Incorrect Total 36 2816 <input type="button" value="Bang"/> 107 <input type="button" value="View Error Values"/> <input type="button" value="Exit"/>
lg					9	5		
mg				2				
hg							2	
lr		6				1		
br		8			1			
tr				1				

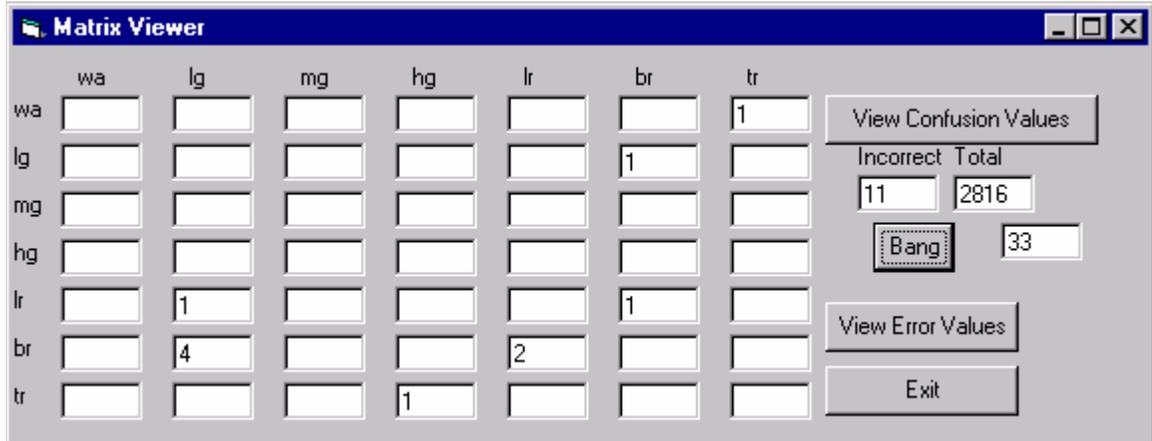
0650



0700



0750



0800

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa				1			
lg					2	1	
mg				1		1	
hg			2				
lr		5				3	
br		5					
tr				1			

View Confusion Values

Incorrect Total
22 2816

Bang 60

View Error Values

Exit

0850

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa				1			
lg					6	1	
mg							
hg							
lr		3				3	
br		4			6		1
tr				3			

View Confusion Values

Incorrect Total
28 2816

Bang 81

View Error Values

Exit

0900

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa							1
lg					5	2	
mg				1			
hg			1				
lr		3				3	
br		3	1		5		
tr						1	

View Confusion Values

Incorrect Total
26 2816

Bang 69

View Error Values

Exit

0960

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa						1	
lg					1	1	
mg							
hg							
lr		1				1	
br		3	1		9		
tr							

View Confusion Values

Incorrect Total
18 2816

Bang 46

View Error Values

Exit

0980

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa							1
lg						2	
mg							
hg							
lr		1				3	
br		1			2		
tr						1	

View Confusion Values

Incorrect Total
11 2816

Bang 31

View Error Values

Exit

0990

Matrix Viewer

	wa	lg	mg	hg	lr	br	tr
wa							2
lg			1		2	1	
mg							
hg							
lr		2				2	
br		6	1	1	6		1
tr				4			

View Confusion Values

Incorrect Total
29 2816

Bang 86

View Error Values

Exit

APPENDIX G

Network Built	Topology	r^2
3332	10-2-7	.9722
3333	10-3-7	.9746
3334	10-4-7	.9795
3335	10-5-7	.9819
3325	11-11-7	.9831
3335	10-5-7	.9819
3345	7-13-7	.9806
3355	8-1-7	.9438
3225	5-1-7	.9532
3325	11-11-7	.9831
3425	13-6-7	.9889
2425	13-8-7	.9889
3425	13-6-7	.9889
4425	13-3-7	.9575