ONE-AGAINST-ALL REMOTE SENSING IMAGE CLASSIFICATION USING SUPPORT VECTOR MACHINE

Onuwa Okwuashi, PhD
Department of Geoinformatics & Surveying, University of Uyo, Nigeria
Dubem Isaac Ikediashi, PhD
Department of Building, University of Uyo, Nigeria

Abstract

This research presents a new method of extending a binary support vector machine algorithm to a multi-class remote sensing image task using a one-against-all technique. A Landsat image is used for the experiment. The land use classes of interest are: developed, undeveloped and water. The spectral bands are extracted in ArcGIS while MATLAB programming software is used for the modelling. The selection of support vector machine kernel functions and parameters are based on the k-fold cross-validation. The initial classification result yields four land use classes: developed, undeveloped, water and unclassified; while the final classification result is resolved to three land use classes: developed, undeveloped and water. For the final result, the Kappa statistic was obtained for 20 iterations; the highest Kappa statistic is obtained at the 9th iteration while the least Kappa statistic is obtained at the 20th iteration. The initial result yields a Kappa value of 0.7787 which indicates a substantial agreement with the ground truth data; while the final image yields a Kappa value of 0.8671 which indicates an almost perfect agreement with the ground truth data.

Keywords: Remote sensing, One-against-all, Image classification, Satellite image, Support vector machine

Introduction

Remote Sensing (RS) image classification tasks are typically multi-class problems, implemented with multi-class classifiers. Recently binary classifiers have been applied to RS problems notably by either One-against-All (1AA) or One-Against-One (1A1) technique. A major disadvantage of these techniques is the unclassification of some pixels. Several methods of extending binary classifiers to multi-class tasks have been proposed in past literatures (Mountrakis et al., 2011). This work therefore presents a robust method of modelling a 1AA classification based on Support Vector Machine (SVM) (Cortes and Vapnik, 1995).

SVM is intrinsically a contemporary binary machine learning algorithm that has been applied extensively in several disciplines. However, applications of binary classification are very limited especially in RS land cover classification where most of the classification problems involve more than two classes (Melgani and Bruzzone, 2004).

In this experiment, a 1AA technique will be used to extend the binary SVM classifier to a multi-class task. This 1AA technique is called a "winner-takes-all" classification. For an *N* class classification, *N* binary classifiers are created. Each classifier is trained to discriminate one class from the remaining *N-1* classes (Melgani and Bruzzone, 2004). The algorithms used for classifying RS images can either be supervised or unsupervised. Notable conventional unsupervised techniques include kmeans and fuzzy c-means while the supervised include K nearest neighbour, maximum likelihood classifier and Gaussian mixture model. SVM implemented in this work is a supervised technique.

Support vector machine

For an SVM linear case, a binary problem that belongs to classes -1 and +1 respectively can be classified using a linear hyperplane. To separate these two sets of objects, a few training samples must be chosen. Assuming that the training set has n-training samples, that is, $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, where $x_i \in \Re^N$ is an N dimensional vector that belongs to one of classes $y_i \in \{-1, +1\}$. The stated binary classification problem can be separated using a linear decision function (Vapnik, 2000),

$$f(x) = w \cdot x + b \tag{1}$$

where $w \in \Re^N$ is a vector that determines the orientation of the desired hyperplane required for the separation, and $b \in \Re$ is called the "bias." The optimal hyperplane needed to separate the two objects is,

$$y_i (w \cdot x + b) \geq 1 \tag{2}$$

The solution to this problem can be found by solving the following constrained optimization problem (or primal problem) (Vapnik, 2000), minimise,

$$\frac{1}{2} w \cdot w + C \sum_{i=1}^{n} \xi_i \tag{3}$$

subject to: $y_i (w \cdot x + b) \geq 1 - \xi_i$, $\xi_i > 0$, and for $\forall i = 1, ..., n$; where C, $0 < C < \infty$, is called the penalty value; while ξ_i are the slack variables. The optimisation problem or dual form derived by solving equation 3 can be expressed as, maximise:

$$\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) \tag{4}$$

subject to: $\sum_{i=1}^{n}\alpha_i y_i = 0$, and, $0 \le \alpha_i \le C$, for $i = 1,...,n$ (Vapnik, 2000). The resulting decision function for the linear case can be given as,

$$f(x) = sign\left(\sum_{i=1}^{n} y_i \alpha_i^0 (x_i \cdot x) + b^0\right) \tag{5}$$

where x_i are the training samples; y_i are the target labels of the training samples (such that, $y_i \in \{-1,+1\}$); α_i^0 are the Lagrangian multipliers; b^0 is known as the "bias;" while x denotes the test set (Vapnik, 2000).

For the nonlinear case, the optimisation problem can be written as, maximise:

$$\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \tag{6}$$

subject to: $\sum_{i=1}^{n}\alpha_i y_i = 0$, and, $0 \le \alpha_i \le C$, for $i = 1,...,n$. While the resulting decision function can be given as,

$$f(x) = sign\left(\sum_{i=1}^{n} y_i \alpha_i^0 (x_i, x) + b^0\right). \tag{7}$$

Given two arbitrary support vectors $x_A \in$ class *A* and $x_B \in$ class *B*, the bias can be evaluated as,

$$b^0 = -\frac{1}{2}\sum_{i=1}^{n} y_i \alpha_i^0 \big(K(x_A, x_i) + K(x_B, x_i)\big) \tag{8}$$

The kernel $K(x_i, x_j)$ can be any of the following common kernel functions: the linear kernel $x \cdot x_i$, polynomial kernel $(x \cdot x_i + 1)^d$ and Radial Basis Function (RBF) kernel $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\gamma^2}\right)$ (Gunn, 1998).

A multi-class problem

The formulation of a 1AA technique is such that, a data point will be classified under a certain class if and only if that class accepts it and all the other classes reject it. Using a simulated multi-spectral satellite remote sensing image (where: 1=water, 2=undeveloped, and 3=developed) (Table 1), with a matrix size of 5 x 5, that consists of three simulated spectral bands (Tables 2-4); we intend to classify the three classes (water, undeveloped and developed) contained in the image. All the three spectral bands in Tables 2-4 contain hypothetical digital numbers.

Table 1. Actual image

1	1	1	1	2
1	1	1	2	2
3	3	1	2	2
3	3	2	2	2
3	3	3	2	2

Table 2. Band 1

1	0	4	2	26
8	10	9	27	20
42	40	7	26	24
47	43	22	29	30
46	45	50	23	25

Table 3. Band 2

78	73	72	74	103
75	70	80	104	101
180	190	76	106	108
186	182	100	109	107
188	184	183	105	110

Table 4. Band 3

30	36	34	37	66
33	38	31	67	63
90	93	39	68	62
97	96	60	65	61
92	98	99	66	64

Table 5. Training data

Water	Band 1 (1,2) = 0	Band 2 (1,2) = 73	Band 3 (1,2) = 36
Water	Band 1 (2,1) = 8	Band 2 (2,1) = 75	Band 3 (2,1) = 33
Undeveloped	Band 1 (1,5) = 26	Band 2 (1,5) = 103	Band 3 (1,5) = 66
Undeveloped	Band 1 (2,4) = 27	Band 2 (2,4) = 104	Band 3 (2,4) = 67
Developed	Band 1 (3,1) = 42	Band 2 (3,1) = 180	Band 3 (3,1) = 90
Developed	Band 1 (4,2) = 43	Band 2 (4,2) = 182	Band 3 (4,2) = 96

To classify the image given in Table 1, a training set was randomly selected. The training data (six pixels) consist of elements from the three classes (Table 5). For modelling convenience let the remaining nineteen pixels that were not used for training the classifier (Table 6) represent the test set. Conventionally the size of the test set is usually smaller than that of the training set in machine learning. But for the purpose of illustration, let the remaining nineteen pixels serve as the test set. As mentioned in the preceding section, the 1AA technique is primarily known as a “winner-takes-all” classification. *N* binary SVM classifiers must be constructed for an *N* class classification; *N* being the number of classes involved. Each classifier is trained to discriminate one class from the remaining $N-1$ classes.

A label is assigned to a class if and only if that class accepts it, and every other class rejects it. If this is not the case the pixel is left unclassified. Three classifiers were required for the classification (Table 7). From Table 7, Classifier 1: Water (+1) versus All (-1), Classifier 2: Developed (+1) versus All (-1), and Classifier 3: Undeveloped (+1) versus All (-1). The modelling was implemented in MATLAB using the polynomial kernel of degree, *d=2*. The penalty value was, *C=100*. The training and test results were presented in Table 7.

Table 6 . Test data

Band 1 (1,1) = 1	Band 2 (1,1) = 78	Band 3 (1,1) = 30
Band 1 (4,1) = 47	Band 2 (4,1) = 186	Band 3 (4,1) = 97
Band 1 (5,1) = 46	Band 2 (5,1) = 188	Band 3 (5,1) = 92
Band 1 (2,2) = 10	Band 2 (2,2) = 70	Band 3 (2,2) = 38
Band 1 (3,2) = 40	Band 2 (3,2) = 190	Band 3 (3,2) = 93
Band 1 (5,2) = 45	Band 2 (5,2) = 184	Band 3 (5,2) = 98
Band 1 (1,3) = 4	Band 2 (1,3) = 72	Band 3 (1,3) = 34
Band 1 (2,3) = 9	Band 2 (2,3) = 80	Band 3 (2,3) = 31
Band 1 (3,3) = 7	Band 2 (3,3) = 76	Band 3 (3,3) = 39
Band 1 (4,3) = 22	Band 2 (4,3) = 100	Band 3 (4,3) = 60
Band 1 (5,3) = 50	Band 2 (5,3) = 183	Band 3 (5,3) = 99
Band 1 (1,4) = 2	Band 2 (1,4) = 74	Band 3 (1,4) = 37
Band 1 (3,4) = 26	Band 2 (3,4) = 106	Band 3 (3,4) = 68
Band 1 (4,4) = 29	Band 2 (4,4) = 109	Band 3 (4,4) = 65
Band 1 (5,4) = 23	Band 2 (5,4) = 105	Band 3 (5,4) = 66
Band 1 (2,5) = 20	Band 2 (2,5) = 101	Band 3 (2,5) = 63
Band 1 (3,5) = 24	Band 2 (3,5) = 108	Band 3 (3,5) = 62
Band 1 (4,5) = 30	Band 2 (4,5) = 107	Band 3 (4,5) = 61
Band 1 (5,5) = 25	Band 2 (5,5) = 110	Band 3 (5,5) = 64

Table 7. Training and test results (training result: b^0 & α^0; test result: $f(x)$)

CLASSIFIER 1		CLASSIFIER 2		CLASSIFIER 3	
Water (+1) versus All (-1)		Developed (+1) versus All (-1)		Undeveloped (+1) versus All (-1)	
$(b^0 = 2.3074)$		$(b^0 = -2..8422e-14)$		$(b^0 = 0.0039)$	
α^0	$f(x)$	α^0	$f(x)$	α^0	$f(x)$
0	1.0554	0	-2.0015	0.3256*1.0e-6	-1.6331
0.2322*1.0e-7	-7.2163	0.0675*1.0e-5	2.8611	0	-0.9254
0.2322*1.0e-7	-7.0489	0	3.8327	0.0868*1.0e-6	-2.1812
0	1.0018	0.1842*1.0e-5	-1.1838	0.1606*1.0e-6	-0.3274
0	-7.0825	0	0.6780	0	-2.8952
0	-7.0862	0.1101*1.0e-5	1.5080	0.0783*1.0e-6	-0.6962
	1.0908		-1.8352		-0.9159
	0.9062		-0.5683		-1.3956
	0.8577		-1.8773		-0.7039
	-0.6363		-1.2766		0.3258
	-7.2028		3.6342		0.1299
	-1.0878		-1.0000		1.1064
	-1.1863		-1.4273		1.0190
	-1.2553		0.0327		0.6636
	-1.0338		-1.9492		0.6763
	-0.7379		-2.3010		0.4479
	-1.0377		-0.9341		0.0902
	-1.0569		0.9427		0.4285
	-1.1998		-0.9530		0.2224

From Table 7, the training results were furnished by b^0 and α^0, while the test result was furnished by $f(x)$. Scores with $f(x) > 0$ were assigned 1, while scores with $f(x) < 0$ were assigned 0. The results from the three binary classifiers were presented in Tables 8, 9 and 10. The following MATLAB pseudo codes were applied to the outcome of the three classifiers to derive the result of the final image (Table 11):

```
WATER = (WATER_ALL==1) & (DEV_ALL==0);
DEVELOPED = (DEV_ALL==1) & (UNDEV_ALL==0);
UNDEVELOPED = (UNDEV_ALL=1) & (WATER_ALL==0);
RESULT = WATER + 3* DEVELOPED + 2* UNDEVELOPED
```

From Table 11, 16 pixels were correctly classified while 3 pixels were left unclassified (the zeros in Table 11 were the unclassified pixels). The resulting classification solution using the 1AA SVM model (Table 11) was compared with the original digital image given in Table 1; the classification accuracy was 16 ÷ 19 = 84.21%. The confusion matrix given in Table 12 was derived based on a cell-by-cell comparison of the actual (Table 1) and predicted image (Table 11). The classification overall accuracy (computed from Table 12) = sum of diagonal elements ÷ sum of all elements in the matrix. Therefore, overall accuracy = 21 ÷ 25 = 84.00%.

Table 8. Result for Water versus All

1	1	1	1	0
1	1	1	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Table 9. Result for Developed versus All

0	0	0	0	0
0	0	0	0	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0

Table 10. Result for Undeveloped versus All

0	0	0	0	1
0	0	0	1	1
0	0	0	1	1
0	0	1	1	1
0	0	1	1	1

Table 11. Predicted image

1	1	1	1	2
1	1	1	2	2
3	3	1	2	2
3	3	2	0	0
3	3	0	2	2

Table 12. Confusion matrix for the classification

	Reference data			
	Water	Undeveloped	Developed	Unclassified
Predicted data				
Water	7	0	0	0
Undeveloped	0	8	0	0
Developed	0	0	6	0
Unclassified	0	2	1	0

Resolving the classes of pixels resulting from 1AA classification

The result of the 1AA classification is regarded as a tentative classification result. For example, a three class problem with land uses such as developed, undeveloped and water will yield four classes after the 1AA classification. In order words, the fourth class represents the unclassified pixels. The final classes of all the pixels (based on designated land use classes) are obtained by employing first the following equation,

$$p^{t+1}(\alpha)_{(w_k)} = Q\left(1+\exp\left\{A*\left[h^t(\alpha)_{(w_k)}*\left(h^t(\alpha)_{(w_k)}*(\gamma^t)^{-1}\right)^{-1}\right]+B\right\}\right)^{-1} \tag{9}$$

where: $p^{t+1}(\alpha)_{(w_k)} =$ the probability that pixel α belongs to class w_k of the t-th iteration

$h^t(\alpha)_{(w_k)} =$ the Moore neighbourhood function of pixel α that belongs to class w_k of the t-th iteration

γ^t = is a uniform random variable within the range 0 and 1 of the t-th iteration

$Q =$ is a coefficient

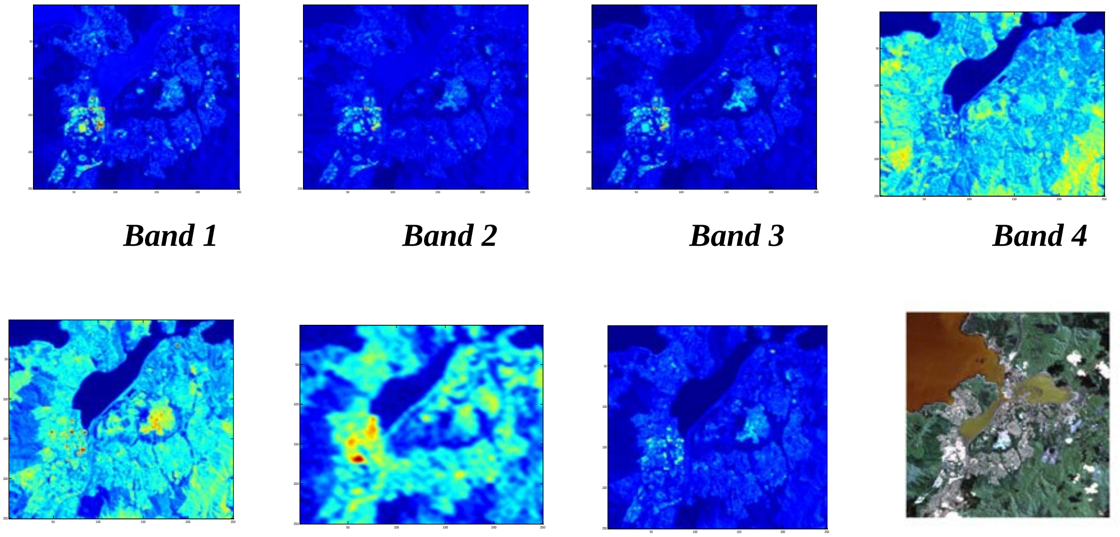
$w_k = w_1, w_2, w_3, ..., w_m$ classes

$A \,\&\, B =$ constants

By employing equation 9, each pixel will yield k number of probability values, where k represents land use classes. Equation 10 represents transition rules that decide the eventual class of a pixel. The probability in equation 9 is calculated iteratively.

$$\begin{array}{llllllll}
If & \left(p^{t+1}(\alpha)_{(w_1)} > p^{t+1}(\alpha)_{(w_2)}\right) & \& & . \; . \; . & \& & \left(p^{t+1}(\alpha)_{(w_1)} > p^{t+1}(\alpha)_{(w_m)}\right) & then & class\_1 \\
If & \left(p^{t+1}(\alpha)_{(w_2)} > p^{t+1}(\alpha)_{(w_1)}\right) & \& & . \; . \; . & \& & \left(p^{t+1}(\alpha)_{(w_2)} > p^{t+1}(\alpha)_{(w_m)}\right) & then & class\_2 \\
. & . & . & & . & . & . & . \\
. & . & . & & . & . & . & . \\
. & . & . & & . & . & . & . \\
If & p^{t+1}(\alpha)_{(w_m)} > p^{t+1}(\alpha)_{(w_1)} & \& & . \; . \; . & \& & \left(p^{t+1}(\alpha)_{(w_m)} > p^{t+1}(\alpha)_{(w_{m-1})}\right) & then & class\_m
\end{array} \tag{10}$$

The research case study is based on a multispectral Landsat 7 ETM satellite image of Porirua, New Zealand, acquired in 2006 (Figure 1). The Landsat image consists of seven spectral bands. The original satellite data were first reviewd in ArcGIS software and all seven bands were extracted using the layer properties tool and visualised in MATLAB (Figure1). Before importing the data into MATLAB, they were first converted from raster to ASCII data using the ArcGIS conversion tool. MATLAB cannot read raster files; hence the data must be in ASCII format for onward processing in MATLAB. In MATLAB the final study area was extracted from the original satellite image. All the seven bands were used for the classification experiment.



Band 1 ***Band 2*** ***Band 3*** ***Band 4***

Band 5 ***Band 6*** ***Band 7*** ***Satellite image***

Figure 1. Landsat image and its extracted bands 1-7

The three classes to be classified are water, developed and undeveloped cells. The stratified random sampling was used to select the training data. The experiment was implemented with a polynomial kernel of degree d = 3 and penalty value C = 100 (Figure 2). The polynomial, Radial Basis Function and the linear kernels were experiment. The polynomial kernel furnished the best accuracy. The selection of d and C was done using the k-fold cross-validation process where k = 10 (Bhardwaj et al., 2005). At each experiment, nine datasets (k-1datasets) were put together to train the SVM while the remaining one dataset was held to test the accuracy of the experiment. The accuracy estimation was based on Kappa statistic expressed from equations 11-13. The experiment was repeated in ten folds until all the 10 datasets were used for both training and testing (Figure 2).

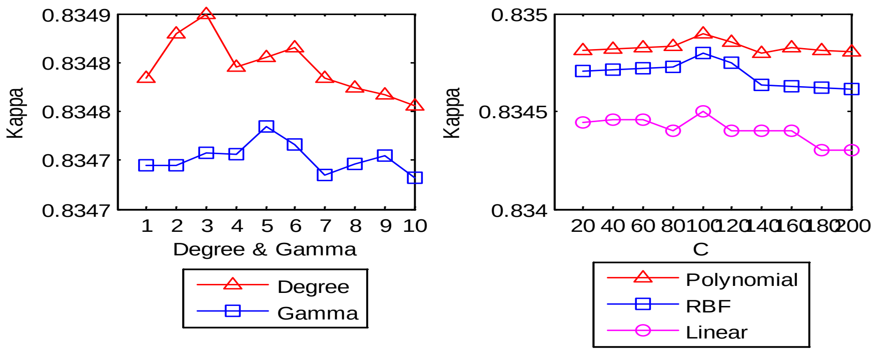


Figure 2. Cross-validation results for: (a) Degree and Gamma (b) *C*

Results

In the experiment, each land use class was individually classified against the rest classes; that is three classifiers were designed. The tentative result presented in Figure 3 was obtained by combining all the three classifiers. Figure 3 was visualised in ArcGIS, by converting the ASCII data from

MATLAB to raster data in ArcGIS using the conversion tool.

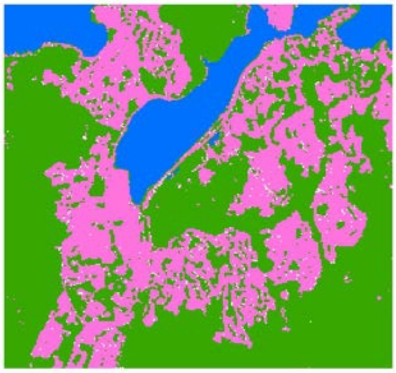
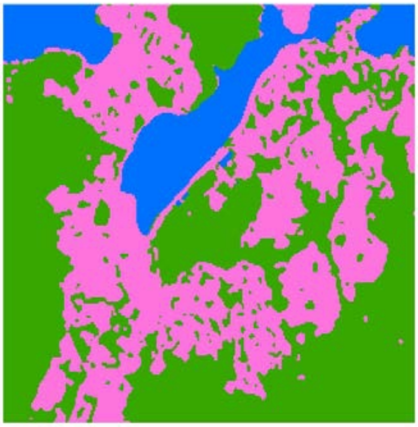


Figure 3. Initial result

Figure 3 contains four land use classes: water, developed and undeveloped. Since the classes of the pixels in Figure 3 are tentative, equations 9 and 10 were employed to obtain the final result in Figure 4. Since the calculated probabilities were determined stochastically (equation 9), each level of iteration was run 20 times to compute the mean Kappa statistic at the 95% confidence level. The final class of each pixel in Figure 3, was resolved to three classes using equations 9 and 10.

Figure 4. Final result



The Kappa statistic was used to evaluate the accuracy of the classification. The Kappa statistic can be expressed mathematically as,

$$K = \frac{P_o - P_c}{1 - P_c} \qquad (11)$$

where,

$$P_o = \sum_{i=1}^{m} P_{ii} = \frac{1}{N}\sum_{i=1}^{m} n_{ii} \qquad (12)$$

and,

$$P_c = \sum_{i=1}^{m} P_{i+}P_{+i} = \frac{1}{N^2}\sum_{1=1}^{m} n_{i+}n_{+i} \;, \qquad (13)$$

(Ma and Redmond, 1995; Lo and Yeung, 2007),
where,

P_o = proportion agreement observed

P_c = proportion agreement expected by chance

n_{ii} = the total number of correctly classified points by class along the diagonal of the error matrix

N = the total number of points checked (sampled)

P_{ii} = the proportion of correctly classified sample points by class at the diagonal of the error matrix (i.e. n_{ii} / N)

P_{i+} = the marginal distribution of the sample data (n_{i+} / N where n_{i+} is the row sum by class)

P_{+i} = the marginal distribution of the reference data(n_{+i} / N where n_{+i} is the column sum of class)

m = the total number of classes

The Kappa estimate of the classification was obtained by a cell-by-cell comparison between the classified and the ground truth data. The mean Kappa estimates for twenty iterations were presented in Figure 5.

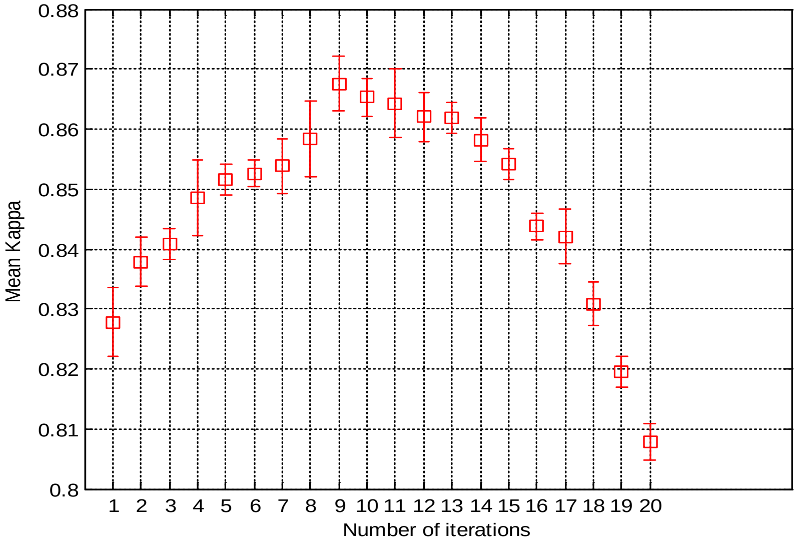


Table 13. Confusion matrix for the initial result

	Ground truth data			
	Developed	Undeveloped	Water	Unclassified
Predicted data				
Developed	12043	698	784	0
Undeveloped	3034	37301	637	0
Water	647	772	5950	0
Unclassified	349	272	13	0

Using the interpretation in Table 15 (Landis & Koch 1977), this result however indicates a substantial agreement between the ground truth and the predicted data (Table 15). The confusion matrix for the final result (Table 14) indicates substantial improvement in the amount of incorrectly classified pixels, by comparing the confusion matrix of the initial result with that of the final result.

Table 14. Confusion matrix for the final result

	Ground truth data			
	Developed	Undeveloped	Water	Unclassified
Predicted data				
Developed	13674	576	660	0
Undeveloped	1214	38206	598	0
Water	589	698	6285	0
Unclassified	0	0	0	0

The off-diagonal elements were the incorrectly classified pixels. The final image yielded a Kappa value of 0.8671. Using the interpretation in Table 15 (Landis & Koch 1977), this result indicates an almost perfect agreement between the ground truth and the predicted data. This work showed the possibility of obtaining accurate classification result using binary classifiers, done simply by enhancing the known classical algorithm.

Table 15. Interpretation of Kappa statistic

KAPPA	INTERPRETATION
< 0	No agreement
0.0 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 1.00	Almost perfect agreement

Conclusion

This work elucidated how a binary classifier can be extended to multi-class task by applying a new technique that resolved the classes of unclassified pixels resulting from the 1AA classification basically by employing pixel neighbourhood and stochasticity. This new technique improved the calculated Kappa statistic from 0.7787 to 0.8671.

References:

Bhardwaj, N., Langlois, R., Zhao, G. and Lu, H. Kernel-based machine learning protocol for Predicting DNA- inding proteins. *Nucleic Acids Research*, 33:6486–6493, 2005.

Cortes, C. and Vapnik, V. Support vector networks. *Machine learning,* 20:273-297, 1995.

Gunn, S. *Support vector machines for classification and regression.* Technical Report, ISIS. Southampton, England: Department of Electronics and Computer Science, University of Southampton, 1998.

Landis, J. R. and Koch, G. G. The measurement of observer agreement for categorical data. *Biometrics,* 33:159–174, 1977.

Lo, C. P. And Yeung, A. K. W. (Eds.). *Concepts and techniques of geographic information systems (2nd ed.).* Upper Saddle River, New Jersey: Pearson Prentice Hall, 2007.

Ma, Z. And Redmond, R. L. Tau coefficients for accuracy assessment of classification of remote sensing data. *Photogrammetric Engineering and Remote Sensing*, 61:435-439, 1995.

Melgani, F. and Bruzzone, L. Classification of Hyperspectral Remote Sensing Images with Support Vector Machines. *IEEE Transactions on Geoscience and Remote Sensing,* 42:1778-1790, 2004.

Mountrakis, G., Im, J. and Ogole, C. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing,* 66:247–259, 2011.

Vapnik, V. N. *The nature of statistical learning theory*. New York: Springer-Verlag, 2000.