University of Texas at Tyler
Scholar Works at UT Tyler

Computer Science Faculty Publications and Presentations

School of Technology (Computer Science & Technology)

Summer 2017

Multispectral Image Analysis using Decision Trees

Arun D. Kulkarni
University of Texas at Tyler, akulkarni@uttyler.edu

Anmol Shrestha
University of Texas at Tyler

Follow this and additional works at: https://scholarworks.uttyler.edu/compsci_fac

🎨 Part of the Computer Sciences Commons

Multispectral Image Analysis using Decision Trees

Arun Kulkarni

Department of Computer Science
The University of Texas at Tyler
Tyler, Texas, USA

Anmol Shrestha

Department of Computer Science
The University of Texas at Tyler
Tyler, Texas, USA

Abstract—**Many machine learning algorithms have been used to classify pixels in Landsat imagery. The maximum likelihood classifier is the widely-accepted classifier. Non-parametric methods of classification include neural networks and decision trees. In this research work, we implemented decision trees using the C4.5 algorithm to classify pixels of a scene from Juneau, Alaska area obtained with Landsat 8, Operation Land Imager (OLI). One of the concerns with decision trees is that they are often over fitted with training set data, which yields less accuracy in classifying unknown data. To study the effect of overfitting, we have considered noisy training set data and built decision trees using randomly-selected training samples with variable sample sizes. One of the ways to overcome the overfitting problem is pruning a decision tree. We have generated pruned trees with data sets of various sizes and compared the accuracy obtained with pruned trees to the accuracy obtained with full decision trees. Furthermore, we extracted knowledge regarding classification rules from the pruned tree. To validate the rules, we built a fuzzy inference system (FIS) and reclassified the dataset. In designing the FIS, we used threshold values obtained from extracted rules to define input membership functions and used the extracted rules as the rule-base. The classification results obtained from decision trees and the FIS are evaluated using the overall accuracy obtained from the confusion matrix.**

Keywords—Decision trees; knowledge extraction; fuzzy inference system; Landsat imagery

I. INTRODUCTION

Many pixel-based classification and clustering algorithms have been developed to analyze Landsat images. These include the minimum distance classifier, maximum likelihood classifier (MLC), and non-parametric techniques such as the support vector machine (SVM), decision tree (DT), ensemble of decision trees, multi-layered perceptron model, fuzzy inference system, and fuzzy neural networks. The maximum likelihood classification algorithm is one of the most well-known algorithms. It assumes the normal distribution for reflectance values and calculates the mean vector and covariance matrix for each class using training set data. The classifier uses Bayes' rule to calculate posterior probabilities and assigns a pixel to the class with the highest posterior probability [1]. The SVM algorithm is appealing for Landsat data analysis because of its ability to successfully handle small datasets, often producing higher classification accuracy than traditional methods [2]. Vapnik [3] proposed the SVM algorithm. The use of a kernel for SVMs was suggested by Boser *et al.* [4]. The SVM is a binary classifier that assigns a sample to one of the two linearly separable classes. In the SVM algorithm two hyper-planes are selected so as not only to maximize the distance between the two classes but also not to include any points between them

[5]. The SVM algorithm is extended to nonlinearly separable classes by mapping samples to a higher dimensional feature space. Huang *et al.* [6] have used the SVM algorithm to classify pixels in remotely sensed images. They have shown that for most training cases slightly higher accuracies were achieved when the model was trained with a randomly selected fixed number of samples for each class. Mitra *et al.* [7] have used the SVM algorithm for Landsat image analysis. Moumtrakis *et al.* [8] have provided a review of usage of SVM in remote sensing.

Neural networks are preferred for classification because of their parallel processing capabilities as well as learning and decision-making abilities. Several studies aimed at evaluating the performance of neural networks in comparison with traditional statistical methods to remote sensing applications are available. Benediktsson and Sveinsson [9] have used neural networks for feature extraction and classification for multisource data. Neural networks with learning algorithms such as backpropagation (BP) can learn from training samples and are used Landsat data analysis [10]-[16]. Laprade [17] has used the split-merge clustering algorithm for segmentation in aerial images. Hathaway and Bezdek [18] have used the fuzzy K-means for pixel classification in multispectral images. Pal *et al.* [19] and Kulkarni and McCaslin [20] have used fuzzy neural networks for classification of pixels in Landsat images. Neural networks provide a reasonable alternative to statistical methods for classifying pixels in Landsat images [21].

Decision trees represent another type of classification algorithm that is non-parametric in nature [22]. Pooja *et al.* [23] classified pixels in multispectral images using decision trees. They used C4.5 algorithm to implement the decision tree. However, they did not consider the problem of over-fitting. Hansen *et al.* [24] have suggested classification trees as an alternative to traditional land cover classifiers. Lowe and Kulkarni [25], [26] have used Random Forest and the decision tree for classification of pixels in Landsat data. They have considered two scenes that represent the Mississippi River bottom land and the Yellowstone forest areas. Random Forest algorithm is an ensemble of trees. In designing a decision tree the stopping criterion is that each terminal node represents samples of the same category or the height of the tree exceeds the specified limit. A major problem with a decision tree is that the tree can be built to fit training data perfectly. While the decision tree may accurately classify training data, it could be too trained to the training data that it may not perform well on the real world data. This problem is known as overfitting the training data and something one should be aware of in designing a decision tree. Overfitting of a tree may result in a number of misclassifications. Often the decision tree obtained

from the training samples needs to be pruned to be used as a model for classifying other pixels in the scene. Again the question arises as to what level we should prune the decision tree so that it can be used as a reliable model. In this paper, we implement decision trees using the C4.5 algorithm and classify pixels in a Landsat scene. We also considered the problem of overfitting the decision tree. To overcome the effect of overfitting, we used the method of pruning to improve classification accuracy. Also we consider the effect of sample size on classification accuracy. We used the MATLAB Machine Learning and Computer Vision toolboxes to implement the C4.5 algorithm, and we have chosen Landsat 8 Scene from Juneau, Alaska area. Furthermore, we extracted knowledge from the pruned decision tree regarding classification rules. To validate extracted rules, we built a fuzzy inference system (FIS) using the rules as the rule base and threshold values from the extracted rule set to define input fuzzy membership functions. We classified all samples from the training set data using the FIS and compared predicted and actual categories. Similar approach was proposed by Taylor *et al.* [27] for extracting knowledge from MARSI dataset. The work presented in this paper differs from the earlier work in [23], [26] in two aspects a) we have considered the problem of overfitting and b) we have validated extracted rules using the FIS. The outline of the paper is as: Section II explains the methodology. Section III describes the data set, implementation, and results, and Section IV provides conclusions.

II. METHODOLOGY

A. Decision Tree Classifiers

Decision tree (DT) classifiers are non-parametric classifiers that do not require any *a priori* statistical assumptions regarding distribution of data. The structure of a decision tree consists of a root node, some non-terminal nodes, and a set of terminal nodes. The data is recursively divided down the DT according to the defined classifier framework. A binary tree is a special case of a decision tree. Kulkarni [28] used a binary decision tree to classify pixels in multispectral images, where a subset of features was used at each non-terminal node to classify samples. To select the subset of features at each non-terminal node, the separability of the classes was used as the criterion. One of the most popular algorithms for constructing a decision tree is ID3 algorithm suggested by Quinlan [29]. The ID3 algorithm was developed for discrete attribute values. The basic idea in the ID3 algorithm is to construct a tree top-down from the root node. At the root node every attribute is tested, and the attribute that best classifies data is selected. The ID3 algorithm uses information gain to make a decision as to which attribute is the best. For each attribute, the information gain is calculated by finding the difference in entropy using (1), where D is the observation vector, m is the number of classes, and p_i is the probability that D belongs class i.

$$entropy\left(D\right) = -\sum_{i=1}^{m} p_i \log\left(p_i\right)$$

(1)

The information gain is calculated by subtracting the entropy before the split and after the split using (2), where A is the attribute being processed. In (2) v is the number of distinct values of attribute A, and $\left|D_j\right|/\left|D\right|$ shows the weight value of the jth split. The process is repeated again for the remaining attributes.

$$Gain\left(A\right) = Info\left(D\right) - Info_A\left(D\right)$$

where

$$Info_A\left(D\right) = \sum_{j=1}^{v} \frac{\left|D_j\right|}{\left|D\right|} \times Info\left(D_j\right)$$

(2)

The ID3 Induction tree algorithm has proven to be effective when working with large datasets that have a number of features, where it is inefficient for human experts to process. C4.5 is a supervised learning algorithm that is descendent of the ID3 algorithm. C4.5 allows the usage of both continuous and discrete attributes.

The main problem with decision trees is overfitting. Mitchell [30] has defined overfitting as: Given a hypothesis space H, a hypothesis h in H is said to overfit the training data if there exists some other h' in H, such that h has a smaller error than h' over the training examples, but h' has a smaller error than h over the entire distribution of instances i. e. Hypothesis $h \in H$ overfits training data if there is an alternative hypothesis $h' \in H$ such that

$$error_{train}\left(h\right) < error_{train}\left(h'\right)$$

and

$$error_D\left(h\right) > error_D\left(h'\right)$$

(3)

where D denotes the entire distribution. Overfitting can decrease the accuracy of a decision tree on real world samples significantly. One method for dealing with overfitting in decision trees is pruning. Removing subtrees from a decision tree is known as pruning. Removing redundant subtrees makes the decision less specific yet performs the same as the original tree. The pruning algorithm goes through the entire tree and removes nodes and subtrees that have no negative effect on the classification accuracy, turning a subtree into leaf node with the common label. Once a decision tree is constructed, classification rules can be extracted by traversing from the root node to each leaf node. The split condition at a non-terminal node represents the antecedent part, and the leaf node represents the consequent part. To evaluate the accuracy of extracted rules, we built a fuzzy inference system (FIS) using the rules and reclassified the training set data.

B. Fuzzy Inference System

A fuzzy inference system (FIS) essentially defines a nonlinear mapping of the input feature vector into a scalar output using fuzzy rules. A general model of a fuzzy inference system (FIS) is shown in Fig. 1. The FLS maps crisp inputs

into crisp outputs. The FIS contains four components: fuzzifier, inference engine, rule base, and defuzzifier [20], [21].
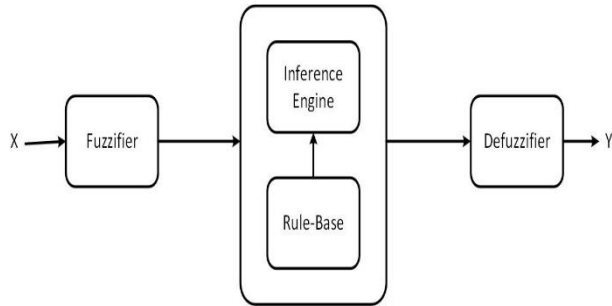

Fig. 1.   Block diagram of a fuzzy inference system (FIS).

The mapping process is described below:

Step 1: The first step is to take inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions.

Step 2: Once the inputs have been fuzzified, we know the degree to which each part of the antecedent has been satisfied for each rule.

Step 3: Apply the implication method. The input for the implication process is a single number given by the antecedent part, and the output is a fuzzy set.

Step 4: Aggregate all outputs. The output of the aggregation process is the combined output fuzzy set.

Step 5: Defuzzify. The input for the defuzzification process is an aggregated output fuzzy set and the output of this is a crisp value.

III.   IMPLEMENTATION AND RESULTS

In this research work, we implemented decision trees and the fuzzy inference system (FIS) using MATLAB scripts from the machine learning and fuzzy logic tool boxes from the MATLAB 15a package and analyzed a Landsat scene.

A. Landsat Scene and Training set

We considered a Landsat scene obtained by the Landsat-8 Operational Land Imager (OLI) on June 13, 2016. The scene is from Juneau, Alaska area with the path and row numbers 58 and 19, respectively. We selected a subset of the original scenes of size 2000 rows by 2000 columns. In order to train the classifiers, we selected four classes: water, vegetation, ice-land, and glaciers [31]. Five training sets each consisting covering areas of the size 100 rows and 100 columns or 10,000 pixels, were selected. The total number of training samples were 50,000 representing five training sets. The classifiers were trained using the samples and reflectance data for bands 2 through 7 [25]. To train each classifier, we selected four classes. The color composite for a raw scene obtained with bands 5, 6, and 7 is shown in Fig. 2. The spectral signatures for classes: water, vegetation, ice-land, and glaciers are shown in Fig. 3. The 3D-scatter plot is shown in Fig. 4. To study the effect of overfitting of a decision tree on classification accuracy, we created two datasets. The first dataset contains 50,000 samples. The second dataset was obtained by adding noise to the first dataset.
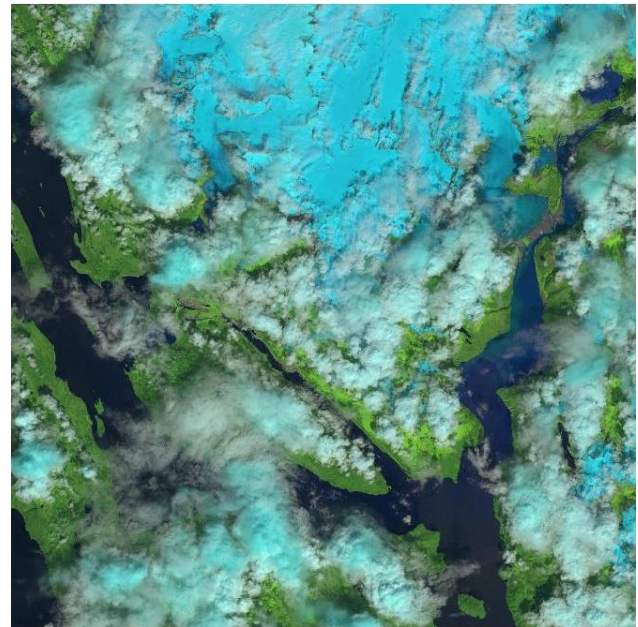
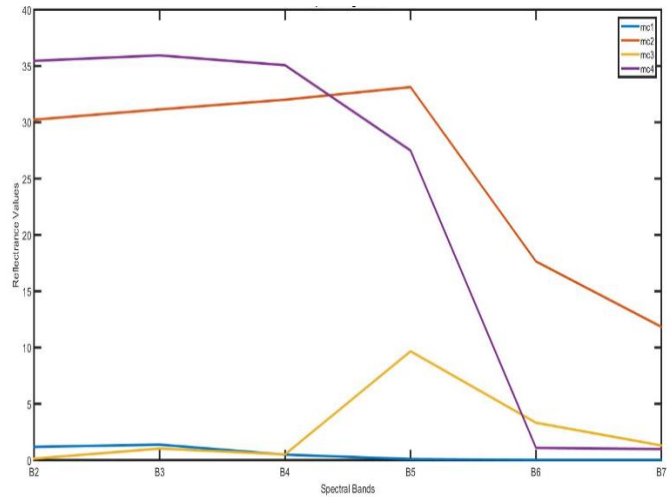

Fig. 2.   Juneau Landsat -8 scenes (Raw Data).

Fig. 3.   Spectral signatures.

B. Decision Tree Implementation

We analyzed both data sets the clean and noisy datasets using full and pruned decision trees. We used the C4.5 algorithm from MATLAB machine learning toolbox. Noisy data samples or outliers often lead to overfitting in decision trees. To evaluate the effect of sample size, we used randomly selected samples. We used from ten percent to ninety percent randomly selected samples for training, and evaluated all samples with each decision tree. Fig. 5 and 6 show the full decision tree and the pruned decision tree, respectively.

The graphs for the overall accuracy of the original and noisy datasets are shown in Fig. 7 and 8, respectively. In Fig. 7 and 8 the x-axis represents the percentage of randomly selected samples that are used for training the classifier, and the y-axis represents the classification accuracy when all samples in the dataset are classified.
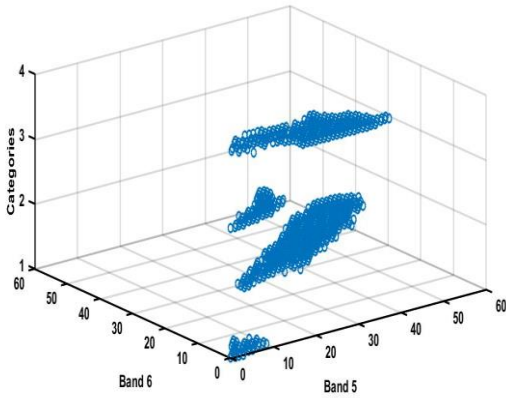
Fig. 4.   Scatter plot.

C. Extracting Classification Rules from Decision Tree

Once we build a decision tree, classification rules can be extracted from the decision tree by traveling down from root node to a leaf node. The arcs of a decision tree represent the antecedent part and the category at the leaf node represents the consequent part of the rule.

We can extract a number of classification rules using the full decision tree. The pruned tree is a generalized version of an over-fitted full tree. The rules represent knowledge extracted from the pruned tree. To validate the extracted rules we built the FIS using the rules as a rule base. To implement the FIS the rules were modified by using the term sets, as the FIS requires rules to be defined using term sets. The modified rule set is used as a rule-base for the FIS.

The classified output obtained with the pruned decision tree is shown in Fig. 9. The colors blue, orange, green, and brown represent categories: water, vegetation, ice-land, and glaciers, respectively.
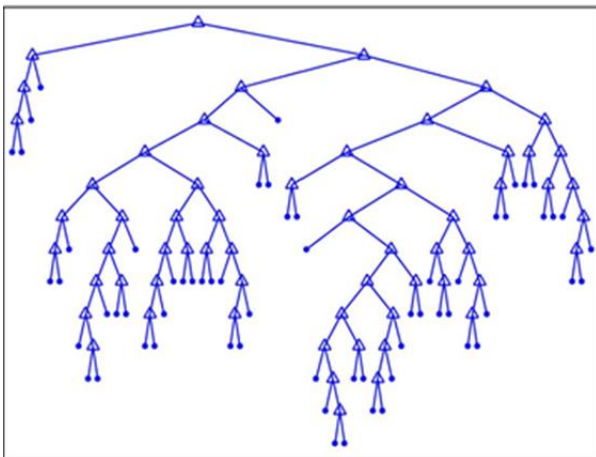
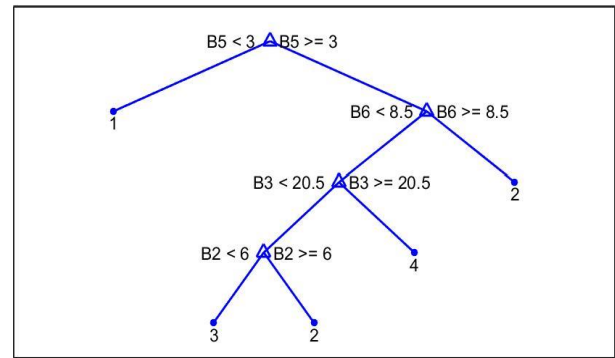

Fig. 5.   Full decision tree.
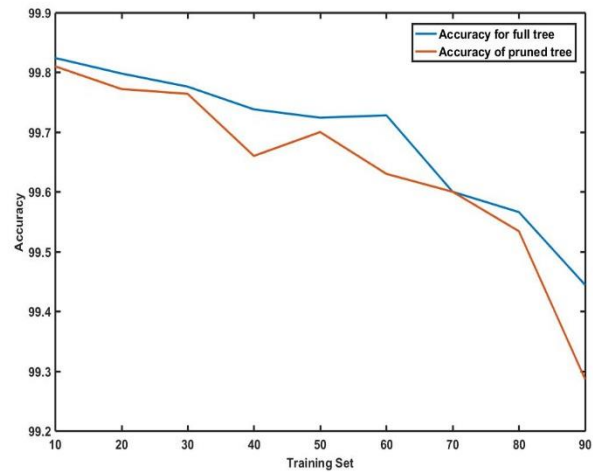

Fig. 6.   Pruned decision tree.

Fig. 7.   Accuracy with the full decision tree and pruned decision tree using clean data.
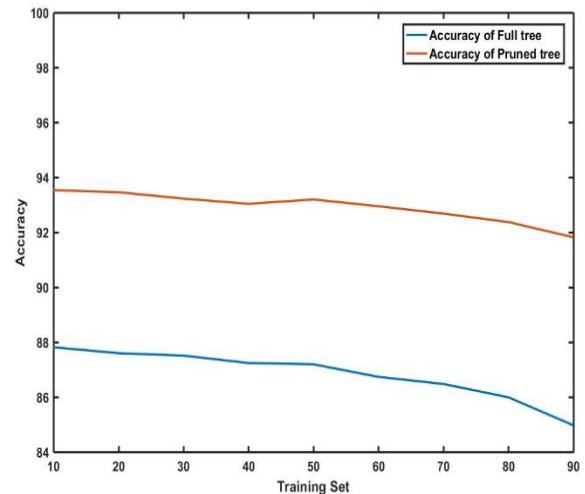

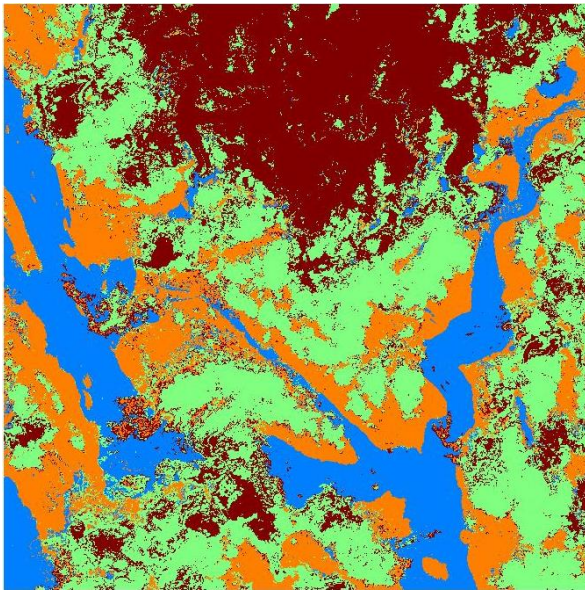Fig. 8.   Accuracy with the full decision tree and pruned decision tree for noisy data.

Fig. 9.   Classified output with the pruned decision tree.

Rule 1: If Band5 < 3 Then Class = Water
Rule 2: If Band5 >= 3 AND Band 6 >= 8.5
         Then Class = Vegetation
Rule 3: If Band5 >= 3 AND Band 6 < 8.5
         AND Band3 > 20.5 Then Class = Glacier
Rule 4: If Band5 >= 3 AND Band 6 < 8.5 AND
         Band3 < 20.5 AND Band2 < 6
         Then Class = Iceland
Rule 5: If Band5 >= 3 AND Band 6 < 8.5
         AND Band3 < 20.5 AND Band2 >= 6
         Then Class = Vegetation

Fig. 10. Extracted rules from pruned decision tree.

Rule 1: If Band5 = Low Then Class = Water
Rule 2: If Band5 = High AND Band6 = High
         Then Class = Vegetation
Rule 3: If Band5 = High AND Band6 = Low
         AND Band3 = High Then Class = Glacier
Rule 4: If Band5 = High AND Band6 = Low
         AND Band3 = Low
         AND Band2 = Low Then Class = Iceland
Rule 5: If Band5 = High AND Band6 = Low
         AND Band3 = Low AND Band2 = High
         Then Class = Vegetation

Fig. 11. Rules for the fuzzy inference system.

D. Fuzzy Inference System

Creating an FIS in MATLAB fuzzy logic toolbox consists of three main steps: 1) defining term sets and fuzzy membership functions for each input; 2) defining term sets and membership functions for the output; and 3) creating the rule-base that implements the inference engine. In the present example we developed the FIS with four inputs and one output. The inputs represent the four features: Band2, Band3, Band5, and Band6 reflectance values. We selected these four features because the extracted rule set contains these features. For each feature we used two term sets with labels Low and High. To define the membership functions, we used threshold values from the rule set in Fig. 10. It can be seen from the rule set shown in Fig. 11 that each variable has used these two term-sets.

The membership functions for Band2, Band3, Band5 and Band6 are shown in Fig. 12(a)-12(d). Fig. 13 shows the output membership functions. Since we have four categories we have chosen four term sets to represent four categories water, vegetation, ice-land, and glaciers. We have created the rule-base by entering using a verbose representation. The fuzzifier converts the input crisp values to input membership values. Depending upon the input fuzzy membership values firing strength for each rule is determined by applying fuzzy operators to antecedent parts. It is possible that for a given input vector more than one rule may get fired. The firing strength of each rule determines the shape of the corresponding output membership function. The reshaped output membership functions are aggregated to form the output fuzzy set, which then is defuzzified to get a crisp output.

The process of generating an output fuzzy set is shown in Fig. 14. Fig. 15 shows the mapping surface. The output of the FIS is then presented to the post-processor block. The post-processor converts the crisp output values to categories. The range of the output membership functions is defined from 0 to 10. In the output fuzzy membership functions values from 0 to 2.5 represent water, values from 2.5 to 5 represent vegetation, values from 5 to 7.5 represent ice-land, and values from 7.5 to 10 represent glaciers.
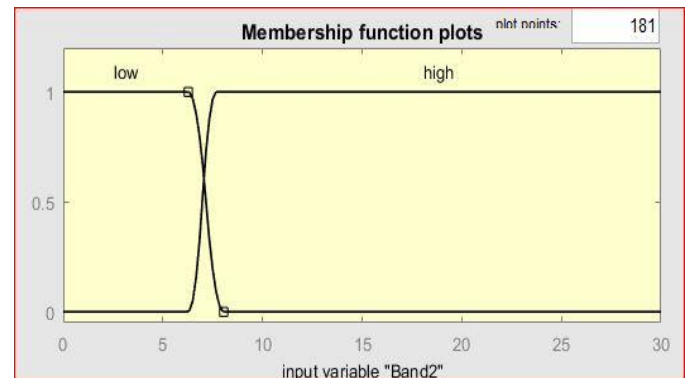

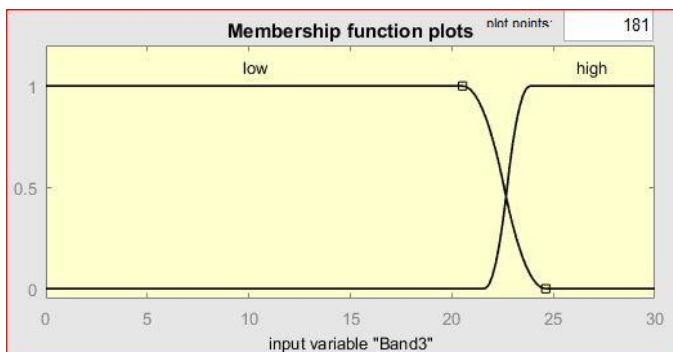Fig. 12. (a) Membership functions for Band2.
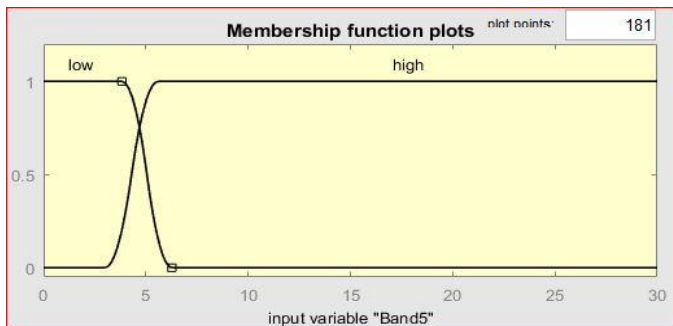
Fig. 12. (b) Membership functions for Band3.

Fig. 13. Output membership functions.

Fig. 12. (c) Membership functions for Band5.

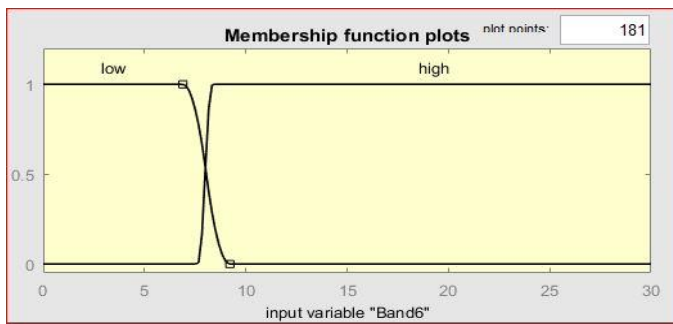Fig. 14. Firing of rules and the process of aggregation.

Fig. 12. (d) Membership functions for Band6.

We classified 50,000 samples from the clean and noisy datasets using the FIS and compared the output of the FIS with actual categories. The confusion matrices for clean and noisy datasets are shown in Tables 1 and 2, respectively. We calculated the overall efficiency using the confusion matrix [32]. Rows in the confusion matrix represent actual categories and columns represent the estimated categories by the FIS. The diagonal values in the confusion matrix show the number of samples that are correctly classified by the FIS. The overall classification accuracy with the FIS was 95.59 percent for the training set data without noise and 80.23 percent for the noisy data set.
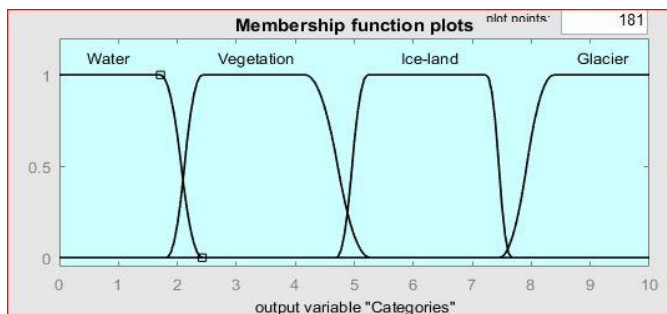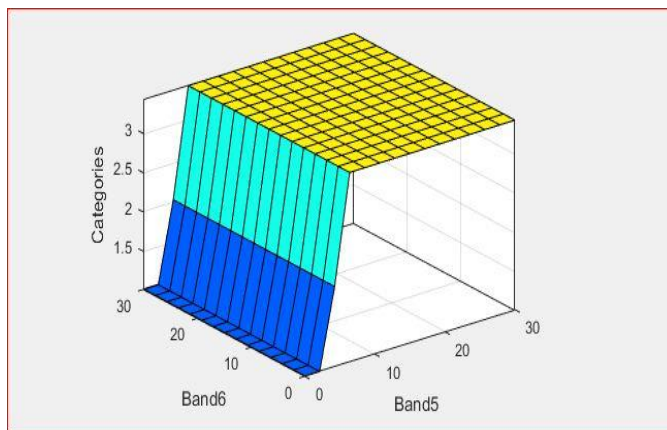

Fig. 15. Mapping surface for the fuzzy inference system.

TABLE. I. CONFUSION MATRIX FOR FIS WITH CLEAN DATA

	Water	Vegetation	Ice-land	Glacier
Water	18663	1301	36	0
Vegetation	12	9924	63	1
Ice-land	182	335	9483	0
Glacier	1	214	59	9726

TABLE. II.   CONFUSION MATRIX FOR FIS WITH NOISY DATA

	Water	Vegetation	Ice-land	Glaciers
Water	16673	3240	84	0
Vegetation	2002	7985	15	1
Ice-land	153	358	7609	1878
Glaciers	30	191	1933	7848

IV.   CONCLUSION

In this research work, we implemented decision trees using the C4.5 algorithm to classify pixels in the Landsat-8 image. Also, we extracted classification rules from the pruned decision tree and evaluated the rules by implementing the FIS. One of the main concerns in decision tree classifiers is that often decision trees are over-fitted, when the training dataset is noisy or contains anomalies in the form of outliers. We generated training set data by selecting training set areas from the scene. To study the effect of overfitting we added noise to the original training set data. We selected five levels for the pruned decision tree because the resulting decision tree represented all categories. Fig. 7 and 8 show the effect of sample size and overfitting. It can be seen from the graphs in Fig. 7 and 8 that the classification accuracy decreases as the percent of randomly selected samples that are used to train the classifier increases. Fig. 8 shows the pruned tree classifier performs better than the full tree classifier for noisy training set data. The pruned tree represents the generalized version of the full decision tree classifier.

The decision trees were implemented using the C4.5 algorithm because the algorithm works with attribute values that are continuous. In the pruned decision tree decisions at each non-terminal node are made using threshold values. To implement an FIS with the extracted rules, we needed to convert the extracted rules that use term sets such as Low and High instead of threshold values. In implementing the FIS, the threshold values from the extracted rule set were used to define fuzzy membership functions for input features. We classified all data samples with the FIS and obtained 95.59 and 80.23 percent overall accuracy for clean and noisy datasets, respectively.

In conclusion, decision trees represent an alternative to conventional algorithm to classify pixels in Landsat images. Furthermore, we can extract knowledge in terms of classification rules from the decision tree. The extracted rules provide a rule-base for the FIS. It is possible to use a combination of rules obtained from multiple sources such as neural networks, decision trees, and an expert's knowledge to implement the rule base in the FIS.

The future work includes 1) developing an algorithm to find the number of levels for pruning the decision tree that yield high accuracy for test data; 2) exploring the possibility of reducing the number of features; and 3) building the FIS with rules that are extracted using multiple methods such as neural networks and decision trees.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification. Wiley New York, 1973.

[2] P. Mantero, G. Moser, and S. B. Serpico, "Partially supervised classification of remote sensing images through SVM-based probability density estimation," IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 3, pp. 559-570, 2005.

[3] V. N. Vapnik and S. Kotz, Estimation of dependences based on empirical data. Springer-Verlag New York, 1982.

[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in Proceedings of the fifth annual workshop on Computational learning theory, 1992, pp. 144-152: ACM.

[5] German Alba, "Remote sensing classification algorithms, algorithm analysis applied to land cover change" Master in Emergency Early Warning and Response Space Applications, CONAE, Argentina, 2014, pp 1-21.

[6] C. Huang, L. Davis, and J. Townshend, "An assessment of support vector machines for land cover classification," International Journal of remote sensing, vol. 23, no. 4, pp. 725-749, 2002.

[7] P. Mitra, B. U. Shankar, and S. K. Pal, "Segmentation of multispectral remote sensing images using active support vector machines," Pattern recognition letters, vol. 25, no. 9, pp. 1067-1074, 2004.

[8] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 66, no. 3, pp. 247-259, 2011.

[9] J. Benediktsson and J. Sveinsson, "Feature extraction for multisource data classification with artificial neural networks," International journal of remote sensing, vol. 18, no. 4, pp. 727-740, 1997.

[10] K. Chen, Y. Tzeng, C. Chen, and W. Kao, "Land-cover classification of multispectral imagery using a dynamic learning neural network," Photogrammetric Engineering and Remote Sensing, vol. 61, no. 4, pp. 403-408, 1995.

[11] G. Foody, "Supervised image classification by MLP and RBF neural networks with and without an exhaustively defined set of classes," International Journal of Remote Sensing, vol. 25, no. 15, pp. 3091-3104, 2004.

[12] W. Y. Huang and R. P. Lippmann, "Neural Net and Traditional Classifiers," in Neural Information Processing Systems, pp. 387–396, 1988.

[13] S. J. Eberlein, G. Yates, and E. Majani, "Hierarchical multi-sensor analysis for robotic exploration," in SPIE 1388, Mobile Robots, vol. 1, pp. 578-586, 1991.

[14] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," Neural computation, vol. 1, no. 3, pp. 372-381, 1989.

[15] S. E. Decatur, "Application of neural networks to terrain classification," in International Joint Conference on Neural Networks, 1989, vol. 1, pp. 283–288.

[16] A. D. Kulkarni and K. Lulla, "Fuzzy neural network models for supervised classification: multispectral image analysis," Geocarto International, vol. 14, no. 4, pp. 42-51, 1999.

[17] R. H. Laprade, "Split-and-merge segmentation of aerial photographs," Computer Vision, graphics, and Image Processing, vol. 44, no. 1, pp. 77-86, 1988.

[18] R. J. Hathaway and J. C. Bezdek, "Recent convergence results for the fuzzy c-means clustering algorithms," Journal of Classification, vol. 5, no. 2, pp. 237-247, 1988.

[19] S. K. Pal, R. K. De, and J. Basak, "Unsupervised feature evaluation: A neuro-fuzzy approach," IEEE Transactions on neural networks, vol. 11, no. 2, pp. 366-376, 2000.

[20] A. Kulkarni and S. McCaslin, "Knowledge discovery from multispectral satellite images," IEEE Geoscience and Remote Sensing Letters, vol. 1, no. 4, pp. 246-250, 2004.

[21] A. D. Kulkarni, "Neural-fuzzy models for multispectral image analysis," Applied Intelligence, vol. 8, no. 2, pp. 173-187, 1998.

[22] M. Ghose, R. Pradhan, and S. S. Ghose, "Decision tree classification of remotely sensed satellite data using spectral separability matrix," International Journal of Advanced Computer Science and Applications, vol. 1, no. 5, 2010.

[23] A. P. Pooja, J. Jayanth, and K. Shivaprakash, "Classification of RS data using decision three approach", International Journal of Computer Applications, vol 23, no. 3, pp 7-11, 2011.

[24] M. Hansen, R. Dubayah, and R. DeFries, "Classification trees: an alternative to traditional land cover classifiers," International journal of remote sensing, vol. 17, no. 5, pp. 1075-1081, 1996.

[25] B. Lowe and A. D. Kulkarni. "Multispectral image analysis using Random Forest", International Journal on Soft Computing (IJSC), vol. 6, no.1, pp. 1-14, 2015.

[26] A. D. Kulkarni, B. Lowe. "Random forest algorithm for land cover classification". International Journal on Recent Innovations Trends in Computing and Communication, vol. 4, no. 3, pp. 58-63, 2016.

[27] C. Taylor, A. Kulkarni, and K. Mokhtar, "Knowledge Extraction from Metacognitive Reading Strategies Data Using Induction Trees," International Journal of Advanced Computer Science & Applications, vol. 1, no. 7, pp. 269-274, 2016.

[28] Kulkarni, A. D. (1983). Categorization of multispectral data using binary tree classifiers. Proceedings of the International Symposium on Machine Processing of Remotely Sensed Data at LARS, Purdue University, pp 609-612.

[29] J. R. Quinlan, "Induction of decision trees," Machine learning, vol. 1, no. 1, pp. 81-106, 1986.

[30] T. Mitchell, Machine Learning, WCB/McGraw-Hill, Boston, MA, pp. 52-80, 1997.

[31] "Land Remote Sensing Image Collections". Remotesensing.usgs.gov. N.p., 2017. Web. 26 Mar. 2017.

[32] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," Remote Sensing of Environment, vol. 37, no. 1, pp. 35–46, 1991.