The Random Forest Approach for Land Cover Mapping.

Panyanat Aonpong, Teerasit Kasetkasem
Department of Electrical Engineering
Kasetsart University
Bangkok, Thailand
Emails: panyanat.o@gmail.com
fengtsk@ku.ac.th

Itsuo Kumazawa
Imaging Science and Engineering Laboratory
Tokyo Institute of Technology
R2-59, 4266 Nagatsuta-cho, Midori-ku
Yokohama, Japan
Email: kumazawa@isl.titech.ac.jp

Preesan Rakwatin
GISTDA
The Government Complex
Bangkok, Thailand
Email: preesan@gistda.or.th

Thitiporn Chanwimaluang
NECTEC
Pathumthani, Thailand
Email: thitiporn.chanwimaluang@nectec.or.th

Abstract – **In this paper, we proposed a new random forest algorithm designed specifically for the land cover mapping problem. Three approaches are investigated, namely, pixel-based, neighbor-looking and combination of both. In the pixel-based approach, we use the fact that all decision trees are different whereas, in the neighbor-looing, the decisions from neighboring pixels are used when the decisions from the Random forest is not clear. Our results on simulated and actual data set showed that our new RF approaches outperformed the traditional one.**

Keyword-Land cover mapping; Random Forest; Neighbor-looking; Pixel-based method; Image Processing; classification

I. INTRODUCTION

The random forest algorithm [1] developed by combining many different decision trees together to form a “forest”. From literatures [1] it has shown to have high classification accuracy with less complexity than other algorithms. The random forest (RF) algorithm can be applied to various types of applications. such as data classification, image processing and so on.

Land cover mapping is useful for resource allocation, conjecture growth of every area. and etc. Remote sensing can help in the analysis of large area. It gives us a good overall of the area even more. By this point, we were able to prepare the population to increase in the future. Managing Green areas, allocation residence zone is not too dense. and etc. These methods can be applied with RF easily.

As a result, this paper will focus on how to further improve the performance of RF in the land cover mapping problems. Here, we concentrate on two main approaches, namely, pixel-based and neighborhood-based approaches. In the pixel-based approach, a new voting scheme is proposed when the differences in term of classification accuracy of each decision tree are taken into account. In the neighborhood-based approach, we employ the inheriting properties of land cover maps, i.e., neighboring pixels are more likely to belong to the same land cover classes than another. Here, the voting scheme at a pixel will take the decisions from its surrounding pixels when the decision trees in a RF have ambiguous decisions. Lastly, we combine both pixel-based and neighborhood-based methods together to achieve even higher performance.

II. BACKGROUND

Hereafter mentioned the algorithm is based on and related to the Random Forest Algorithm.

A. Decision Tree [2]

Decision tree is a kind of hierarchy data structure. It looks like a real tree upside down. Decision tree consists of many nodes, each node is responsible for the test. Branches of Trees Represents the possible features of the selected test and leaves which is at the bottom of the tree represents results of prediction. The node at the top of the tree is called the root node.

In general, the data used to train Binary Decision tree contains a mixed group. To make a Binary Decision tree, it has many algorithms to generate a tree. However, one of the most popular algorithm is ID3 algorithm [3].

In ID3 algorithm, we need to take information gain value of each attribute, this value can be calculated by the following equation. For simplicity, let us focus on two-classes problem where p and n denote positive and negative classes at a given node where the sample must be separated into group using one observed attribute. Prior to the division, the randomness can be measured through the entropy, i.e.,

$$I(p,n) = -\frac{N_p}{N_p+N_n} log_2 \frac{N_p}{N_p+N_n} - \frac{N_n}{N_p+N_n} log_2 \frac{N_n}{N_p+N_n}, \quad (1)$$

when N_p represents the number of data in the class *p* and N_n represents the number of records in the class *n* at the node of interest. Here, $I(p,n)$ is the entropy between Classes *p* and *n*.

Now, by assuming that we can observe *v* attributes from the data with a value $\{A_1, A_2, ... ,A_v\}$, the problem is which of these *v* attributes should be considered from separation of data into different groups. The ID3 algorithm considered the information gain before and after division. If the randomness remains the same, we say that there is no information gain (i.,e., Classes *p* and *n* have roughly the same proportion in all separated groups and before the division) . However, if the randomness disappears after grouping (i.e., all member of one group belongs to Class *p* while another belongs to Class *n*), the information gain is maximum. To measure the information gain, we need to first measure the randomness remained after division of samples into groups using the Attribute A_i. Again, we used the averaged entropy as the measurement of randomness, and it is given as

$$E(A_i) = \sum_{k=1}^{G_i} \frac{N_k}{N} I(N_{p_k}, N_{n_k}) \quad , \quad (2)$$

where G_i is the groups of data that can be separated using the Attribute A_i, N_k is the number of sample in Group *k*, and N_{p_k} and N_{n_k} are the number of sample in Group *k* that belongs to positive and negative classes. If the tree is binary, G_i is fixed to be equal to two. The information gain of Attribute A_i is given by

$$gain(A_i) = I(p,n) - E(A_i), \quad (3)$$

The attribute with highest information gain will be used first to divide samples into different groups based on the attribute value. The samples in different groups will be divided further using the same process until there is no more attribute to consider (i.e., v in this example).

III. Random Forest

In this section, we will define the random forest used in our study and discuss the limitation. The improvement technique to the current random forest will be discussed in the next section.

The random forest [1] is a classification algorithm based on decision tree method. The random forest almost always has higher accuracy than the decision tree [1] since the random forest algorithm incorporate many weak decision trees. Here, the final decision is the combined decision from all the incorporated decision trees in which the majority vote is often employed.

The power of random forest comes from the bagging process used in building each individual decision trees. Here, randomly selected set of samples from a training set are used to train each decision trees. Clearly, the randomly selected samples for each tree should not be identical, otherwise all decision trees will be the same and the performance gain over a single decision tree cannot be obtained. Since each randomly selected sample sets are slightly different, each decision trees are constructed differently. Hence, they will respond differently to the observed data where some tree may respond well in some samples and poorly in other samples. When decision trees make a decision on a given sample, all decision trees that respond well will classify sample into a correct class whereas all decision trees that respond poorly will classify the sample into different classes. When decisions are aggregated, the corrected one becomes dominant. As a result, the traditional RF will use the majority rule to combine decisions from all decision trees.

Because the random forest is composed of many decision trees, the suitable number of the trees should be considered. If forest has fewer trees, the corrected decision may not be dominant and classification performance will be poor. However, if there are too many trees, it will take long time to train. To determine the optimum size of the forest, the out-of-bag technique is employed. The out-of-bag technique keeps track of the estimation error of the current forest. The out-of-bag data separated from bootstrap training is often set to be about one-third of the total number of samples. The out-of-bag samples are used to estimate the performance gain of adding more tree into the forest. If the performance gain is small, the algorithm stops adding more trees into the RF.

In the process of training, the training samples from an input image are randomly chosen, and store these samples in a bootstrap memory. The bootstrap is divided into two parts. The first part is bagging, used to train and expand the size of the forest. Here. bagging contains about 2/3 of the size of the bootstrap. Second is out-of-bag data is used in out-of-bag method. These out-of-bag data is about one-third of the bootstrap. This part is used for the estimation error of Random forest and also help in determining the appropriate size of the forest as well.

The training process of random forest is iterative where each iteration a new decision tree is added into the forest. after creating a decision tree, the next step is to determine the overall performance of the random Forest. The out-of-bag samples are submitted into the forest where the final decisions of each samples are made. Here the majority rule is employed. The overall accuracy is estimated from

$$acc \quad = \frac{\sum_{i=0}^{N_O} I_i}{N_0}, \quad (4)$$

when the size of out-of-bag data equal N_0 pixels, and I_i is defined as .

$$I_i = \begin{cases} 1 \ , \ if \ rf_i = oob_i \\ 0 \ , \quad otherwise \end{cases} . \quad (5)$$

Here, rf_i represents result of RF in Pixel *i* and oob_i represents ground data at Pixel *i*.

If accuracy increases from the random forest in the previous iteration, new decision tree is added and the performance evaluation is repeated. However, the accuracy decreases or remain roughly the same, the training phase is terminated.

The random forest described above assumes that the decisions on each pixel can be independently from each other.

However, in remote sensing image, the neighboring pixels are likely to belong to the same land cover classes than others. Furthermore, the major vote approach used in random forest in combining decisions from all decision trees is based on the assumption that all decision trees have similar performance but, in fact, some trees are more accurate than another. In the next, section, we will incorporate these two ideas into a new random forest algorithm for land cover mapping.

IV. Proposed Algorithm

In this section, we propose three approaches to improve the accuracy of RFs. The first approach employs the decision of the RF from neighboring pixels while the second approach attempts to improve the performance of the pixel-based operation by considering the performance of each individual tree. The last approach is to combine both methods together.

A. Neighbor-Looking method

Our idea here can be simply explained as follows. If a pixel is classified into a class, say *A*, with slight majority, say 51% *A* and 49% *B*, there is a less certainty about the decision when comparing to a more solid voting scores, say 99% A and 1% B. As a result, the random forest should take the voting score of the neighboring pixels into consideration. For example, if the random forest classified a pixel, *s*, into A with 51% A and 49%B when all neighboring pixels of *s* are classified into B with voting score of 1%A and 99%B. In this case, there is a high probability that the random forest makes incorrect decision at *s* since neighboring pixels are more likely to belong to the same land cover classes. Hence, the final decision should be changed to B.

To achieve this goal, the proposed random forest uses a new vote scheme by combining the decision makes from all decision trees from the pixel of interest and its surrounding pixels through a weighing scheme. The proposed weighting scheme uses the weight function that depends on distances between pixels and the equation used to compute the weight is given by

$$w(d_{s,p}) = e^{-\frac{d_{s,p}}{\alpha}}, \tag{6}$$

where $d_{s,p}$ is distance between pixels *s* and *p*, and α is a parameter to be adjusted. Hence, a new voting score is given by

$$vs_c^n(s) = \frac{\sum_{p \in G_s'} w(d_{s,p}) vs_C(p)}{(|G_s|+1)T}, \tag{7}$$

where G_s is a set of neighboring pixels of *s*, $G_s' = G_s \cup \{s\}$, T is the number of trees, $|G|$ is the number of element of *G*, and $vs_c(p)$ is the voting score of Class *c* at a pixel *p* used in equation 8 defined as

$$vs_c(p) = \frac{T_c(p)}{T}., \tag{8}$$

Here, $T_c(p)$ is the number of trees voting for Class *c* at a pixel *p*. Our algorithm varies the values of α from small to large; values to slowly incorporate decisions from surrounding pixels. At the first iteration, α is set to zero and only a voting score in a pixels of interest is considered. The algorithm will find pixels that have a difference of a few votes and tend to ambiguous. If the decision is not ambiguous, the random forest stop the iterative process and make the final decision. However, if the voting score is not clear, α is increased, more neighboring pixels are added, and a new voting score is computed. This process is repeated until the pixels that are confusing to runs out or close to the minimum amount. The final decision rule is given as

$$rf(s) = \arg\max_c vs_c(s), \tag{9}$$

B. Pixel-Base method

In this pixel-based method, we explore the fact that not all decision trees inside a random forest have the same performance. Some trees may respond relatively well to a certain land cover classes where poorly to other classes. As a result, the majority voting scheme is suboptimum.

To incorporate this case, we first assume that all decision trees make their decisions independently from another. Let $P_i(c,k)$ be the probability that the *i*-th tree detects land cover Class *c* when the land cover class *k* is actually present. These probabilities can be estimated using the out-of-bag samples. Hence, the probability of Class *c* is actually present in a pixel given the decisions from all trees is equal to

$$\begin{aligned} P_c &= Pr(c|d_1, \ldots, d_T) \\ &= \frac{\prod_{i=1}^{T} Pr(d_i|C)Pr(c)}{Pr(d_1,\ldots,d_T)}, \end{aligned} \tag{10}$$

since $Pr(d_1, \ldots, d_T)$ is independent of a choice of land cover classes, it can be treated as a constant. Furthermore, if we do not have any prior knowledge on presence and absence of each and land cover class, we have assume $Pr(c) = \frac{1}{C}$. As a result, we have

$$P_c = K \prod_{i=1}^{T} Pr(d_i|c), \tag{11}$$

The optimum goal of our new random forest is to choose the most likely land cover classes based on decisions from all the decision tree, i.e.,

$$c^{best} = \arg\max_c P_c, \tag{12}$$

These criteria are also known as the maximum *a posteriori* (MAP). The above criteria can also be written as

$$c^{best} = \arg\min_c E_c, \tag{13}$$

where

$$\begin{aligned} E_c &= -\log P_c \\ &= -\sum_{i=1}^{T} Pi(di,c) + K' \end{aligned} \tag{14}$$

where K' is a constant.

C. Combination between Pixel-based & Neighbor-looking methods

To make the Random Forest more accurate, we also propose the methodology to combine both the pixel-based and neighbor-looking methods. It using total energy of the forest of Pixel-base method and use the energy of surrounding pixel when the results energy of each feature are so close.

Figure 2 shows the structure and functionality of this combined random forest that incorporates both methods. In blue blocks show common operating of the Random Forest, orange blocks parts used in the optimization of Random Forest and green blocks show results of structure of the learning algorithm and bringing this structure for classification. For the overall process can be summarized as follows.

1.) Bootstrap creating, from all of the data collected the image with both samples and ground truths at the same position as a bootstrap. Create a bagging and out-of-bag data by randomly separated divided into these two groups with two-third in the bagging and one-third in out-of-bag.

2.) Create Decision tree, random samples from bagging is used to train decision trees using equation (1), (2) and (3) in an operation to create a best Decision Tree from available information. (Each step in creating a Decision Tree will always be the best split based on highest information gain).

3.) Out-of-bag method, use every tree in the forest for test with sample information taken in out-of-bag data to validate the results accuracy of Forest with response in out-of-bag data. Accuracy can be calculated by equation (4) and (5). If the accuracy rate has changed, do step 3) and 4) again, if the accuracy rate has remained constant, repeat the test by step 3) and 4) to ensure that the accuracy levels are stabilized, thus ending duplication and proceed to the next step.

4.) Pixel-Base method, try to use each tree with out-of-bag sample then test correction with out-of-bag response and collect statistical, use equation (10) and (11) for calculate energy of each tree in different output of its.

5.) Random Forest prediction, Random forest from Step 1), 2), 3) and 4) were used to analyze images taken value from each pixel to test with every Decision Tree in the forest then vote with energy of each tree. We can calculate total energy by using equation (12) and (13). Feature which has lower total energy will become the result of that pixel like equation (14). Do this until the entire pixel in the image.

6.) Neighbour-looking, check the energy of each pixel of the image by using equation (6). If the pixels are similar energy, we will start a new decision based on the values of surrounding pixels calculated with the weight using equation (7) and (8). If energy of each feature also close, make a new calculate with the reduction of the α down until energy of each feature are clearly different or different value are almost constant change. If the end is not yet a consensus, hold the first value before proceeding Neighbor-looking method. Finally, both final energy value will be compare together, feature which has lower energy will become result of this pixel.

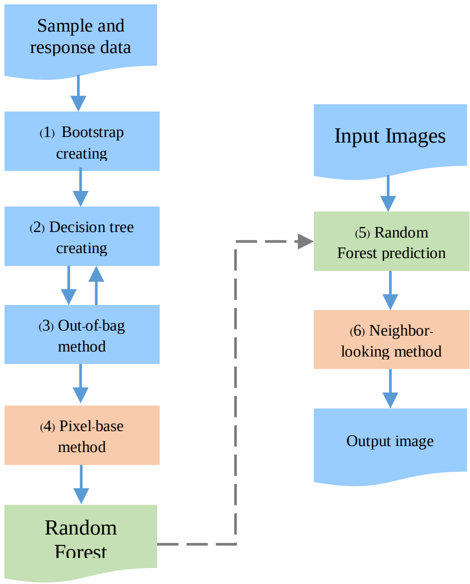

Figure 1. The diagram of the proposed system

7.) Forest voting, voting by using every result from every tree in the forest and calculate final result by using equation (9).

V. RESULT AND DISCUSSION

Example 1

In this example, we evaluate the proposed methods on a binary-class images (Figure 2) with different signal-to-noise ratios (SNR) ranging from 0 dB to 20 dBs. For a given SNR, we repeat the same experiments 50 times where new noises are randomly generated. We used both percentage of correctly classified pixels (PC) and kappa coefficients [4] to evaluate the classification performance. Kappa coefficients measures the agreement between classified map and response image by removing the correctness by chance and it is defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1-p_o}{1-p_e}, \qquad (15)$$

where p_o is probability of matched pixels when comparing with both ground data and classified map, and p_e is probability of correctly classified by chance only. If every pixel is correct close to ideal image, we expect that κ value become close to 1.

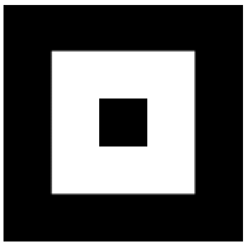

Figure 2. Response image.

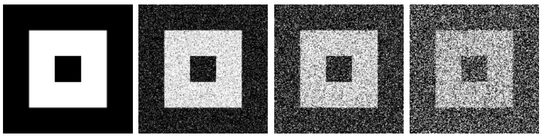

Figure 3. Example of images use in this research
(Response image, SNR = 10, 5, 0 respectively).

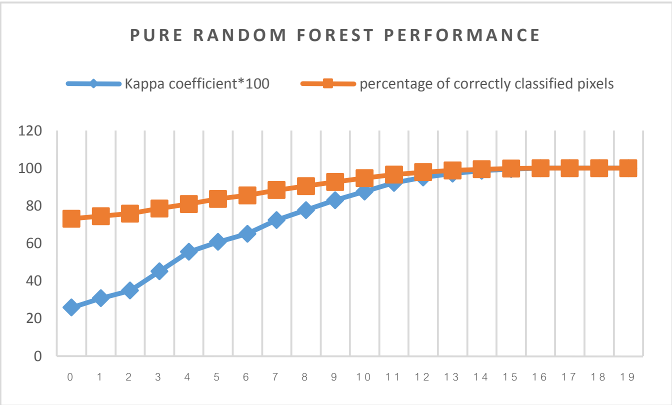

Figure 4. performance evaluation on a traditional RF with different SNR

Figure 3 shows the performance of the traditional random forest algorithm [1] for different SNRs. As expected, both PC and Kappa coefficient increases as SNR increase. However, for low SNR, the Kappa coefficient is very low due to the fact that many pixels are randomly assigned to one of these two classes.

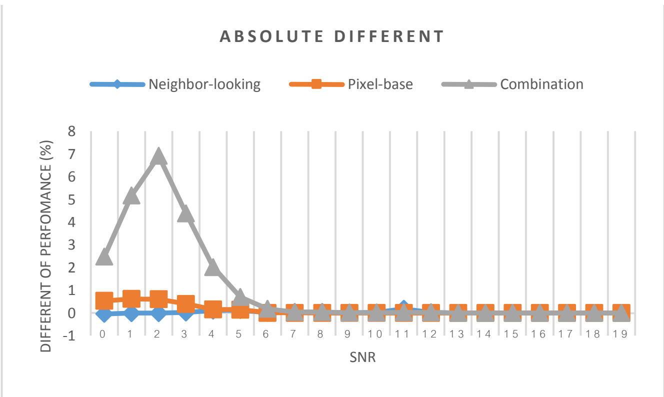

Figure 5. Enhanced Random Forest performance increasing from original Random Forest in absolute different term with different SNR

Figure 4 and 5 show the performance gains of the proposed random forests using neighbor-looking, pixel-based and combination of both to the traditional approach. We found that when SNR is low, the performances of the pixel-based approach and combined methods are much higher than the traditional random forest. However, the neighbor–looking approach has lowest gain since the neighbor-looking does not work well when the lower SNR because this method uses the voting resukts from surrounding pixels that are also erroneous. The pixel-based and combining approaches treat each decision trees differently, and, hence, the less accurate decision trees have a little influent. As a result, the performances increase significantly through the traditional approach. However, when SNR is high all algorithms are very accurate. Hence, there are little or no perform gain at all.

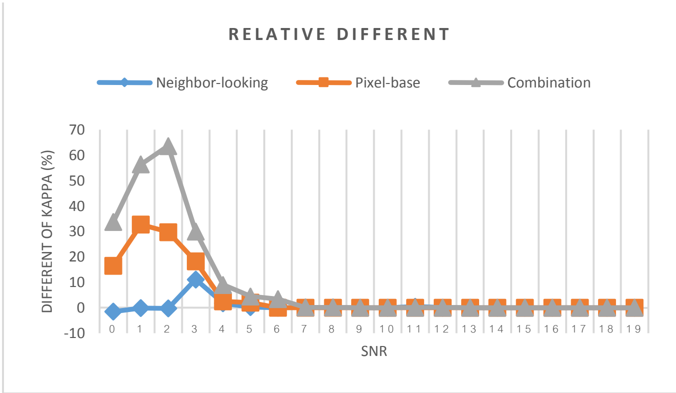

Figure 6. Enhanced Random Forest performance increasing from original Random Forest in relative different term with different SNR

Example 2

In this example, we apply our proposed methods to real remote sensing images. Day-night band images (DNB) [5] was taken by the Suomi National Polar-orbiting Partnership (NPP) by using Visible infrared imaging Radiometer suit (VIIRs). DNB is a visual image with low resolution. Ideal for exploring the area as a whole, which does not require too much detail. These pictures need less memory to store in the same area. It has 750-meters resolution. Since light cannot penetrate clouds very well, it makes the classification of image difficult. However, we can reduce these problems by providing multiple images at the same scene at different times, which is used in combination. Here, we use 8-day composite Day-Night images of Bangkok area (multi-temporal).

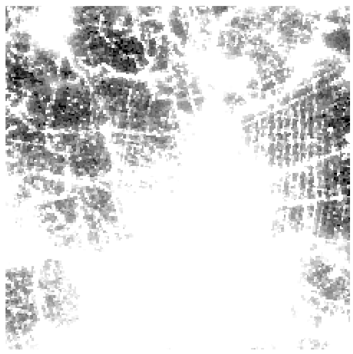

Figure 7. Example of Day-night band image

The ground truth image from Land Development Department at Thailand at 30-meters resolution as training set (Figure 8) where white and black colors indicate urban and non-urban classes.

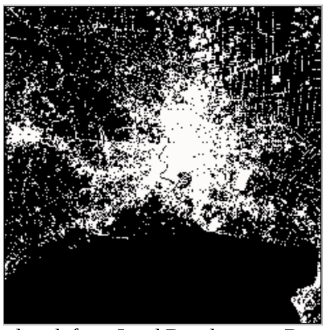

Figure 8. Ground truth from Land Development Department (LDD).

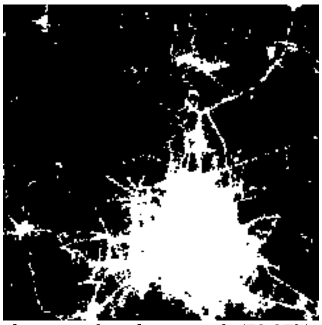

Figure 9. One of pure random forest result (79.87%, Kappa acc. 0.5016)

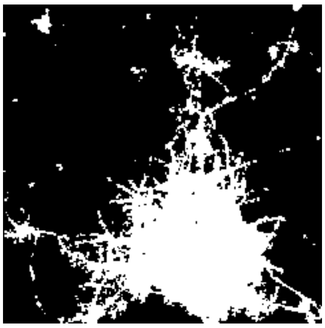

Figure 10. One of enhanced random forest by neighbor-looking method's result (80.88%, Kappa acc. 0.5135)

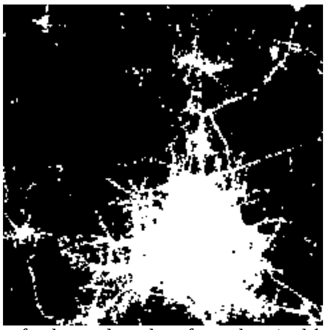

Figure 11. One of enhanced random forest by pixel-base method's result (80.88%, Kappa acc. 0.5135)

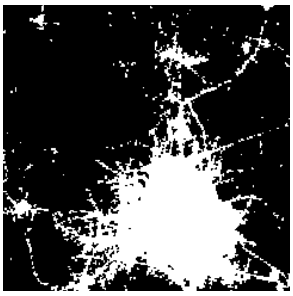

Figure 12. One of enhanced random forest by combination method's result (81.52%, Kappa acc. 0.5139)

TABLE I. Average accuracy and kappa coefficient (100 times examination)

Method	Traditional random forest	Neighbor-looking	Pixel-base	combination
Average performance	78.5376% (0.4501)[a]	78.5552% (0.5002)[a]	81.45% (0.5158)[a]	81.51% (0.5159)[a]

[a] kappa coefficient

Figures 9-12 display the resulting classified map from traditional RF, RF with neighbor-looking, RF with pixel-based, and RF with combination of neighbor-looking and pixels-based approach. The performance evaluations given in Table I. Here, we observe the same trend as in the previous experiment where the combination of pixel-based and neighbor-looking approaches yields the highest performance.

VI. CONCLUSION

In this paper, we proposed three approaches, namely, pixel-based, neighbor-looking, and combination of both methods, to enhance the performance of the random forest for the land cover mapping. In the pixel-based approach, a new voting scheme was proposed where the performances of decision trees were considered. In the neighbor-looking, our RF takes decisions from neighboring pixels into consideration when the decision trees provide ambiguous decisions. The combination approach integrates a new voting scheme with the neighborhood information. From our experiments, all of the proposed methods outperformed the traditional RF where the combination of both pixel-based and neighbor-looking performs the best.

REFERENCES

[1] L.Breiman, *"Random Forest,"* January 2001

[2] Mitchell, *"Decision Trees Learning," Machine Learning,* pp.51-77,

[3] J.R.Quinlan, *"Induction of Decision Trees," Machine Learning1,* pp.81-106, *1986*

[4] Arie Ben-David, *"Comparison of classification accuracy using cohen's weighted kappa. Expert Systems with Applications,"* 34(2):825-832, 2008.

[5] R. Jaturapitpornchai, T.Kasetkasem, P.Rakwatin, I.Kumazawa, T.Chanwimaluang, *"A Level-based Method for Urban Mapping using NPP-VIIRS Nighttime Light Data*", The International Conference of Information and Communication Technology for Embedded Systems.

[6] J.R.Quinlan, M.Hayes, Richards(Eds), *"Decision Trees and multi-valued attributes," Machine intelligence 11*, pp.305-318, 1998(b)

[7] Suhong Yang *Notice of RetractionA research on teaching evaluation by students based on voting paradox model method*," Education Technology and Computer (ICETC), 2010 2nd International Conference on, On *page(s): V1-430 - V1-432 Volume: 1, 22-24 June 2010*