FUSION OF NON-THERMAL AND THERMAL SATELLITE IMAGES BY BOOSTED SVM CLASSIFIERS FOR CLOUD DETECTION

Nafiseh Ghasemian [a] , Mehdi Akhoondzadeh [b,*]

[a] M.Sc. Student, Remote Sensing Department, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, North Amirabad Ave., Tehran, Iran –n.ghasemian@ut.ac.ir

[b] Assistant Professor, Remote Sensing Department, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, North Amirabad Ave., Tehran, Iran-makhonz@ut.ac.ir

KEY WORDS: boosted SVM, Landsat, cloud detection, majority vote

ABSTRACT:

The goal of ensemble learning methods like Bagging and Boosting is to improve the classification results of some weak classifiers gradually. Usually, Boosting algorithms show better results than Bagging. In this article, we have examined the possibility of fusion of non-thermal and thermal bands of Landsat 8 satellite images for cloud detection by using the boosting method. We used SVM as a base learner and the performance of two kinds of Boosting methods including AdaBoost.M1 and σ Boost was compared on remote sensing images of Landsat 8 satellite. We first extracted the co-occurrence matrix features of non-thermal and thermal bands separately and then used PCA method for feature selection. In the next step AdaBoost.M1 and σ Boost algorithms were applied on non-thermal and thermal bands and finally, the classifiers were fused using majority voting. Also, we showed that by changing the regularization parameter (C) the result of σ Boost algorithm can significantly change and achieve overall accuracy and cloud producer accuracy of 74%, and 0.53 kappa coefficient that shows better results in comparison to AdaBoost.M1.

1. INTRODUCTION

Remote sensing data have a lot of applications for example in the monitoring of the environment, management of disasters, urban planning, agriculture and military issues. In most of these applications, an automatic analysis of data is required. One of the most basic tools for analyzing remote sensing data is pixel level classification (Benediktsson et al., 2007). Numerous algorithms for classification have been proposed, among them, ensemble learning methods proved their efficiency as can significantly improve classification accuracy. One of the ensemble learning algorithms is Boosting. The general Boosting idea is to develop classifier team D incrementally, adding one classifier at the time. There are two implementations of AdaBoost with reweighting and with resampling (Kuncheva, 2004). In this article, the first type was applied. A lot of researches are done on the effectiveness of ensemble learning methods on land cover classification. Colstoun et al. (2003) used boosted decision tree classifiers for land cover classification on multi temporal Landsat 7 ETM+ images and achieved over all accuracy of 82% on ground truth data. Pal (2008) used boosted and bagged SVM for land cover classification. Bagged SVM improved the classification accuracy but boosted one, decreased the performance of support vector machines. In these articles, the efficiency of boosted SVM in meteorological applications has not been studied. Cloud detection methods can be divided into two categories: data mining and physical methods. Support vector machines (SVM), artificial neural networks can be considered as type one. Physical methods use characteristics like albedo, brightness temperature for cloud masking. In early 2000 some researchers showed that data mining methods can significantly improve cloud detection in comparison to physical ones (Han et al., 2006). Physical and histogram based methods use spectral features. Spectral features are simple and useful for cloud detection but in complicated cases, for example in presence of ice clouds and snow, only spectral features are not sufficient and use of textural features should be considered. Textural features use spatial distribution of gray values for cloud classification. SVD, GLCM, GGCM and wavelet packets (WTP) are some examples of this category. Li et al. (2003) applied both spectral and textural features as input to maximum likelihood classification. Modis cloud mask (MOD35) was used as initial classification and the classification was improved in both cloud and non-cloud areas. In this survey the final classification result is dependent a lot to MODIS cloud mask algorithm that requires precise calibration of satellite instruments, in addition MODIS has a large variety of spectral bands but in Landsat 8 satellite images we have limited choice of spectral bands hence the use of some features like brightness temperature differences is not possible. Lima et al. (2009) introduced one new Boosting algorithm "σ Boost" for increasing diversity between SVM classifiers. In this article, the training error of the previous iteration was used for computing the Gaussian width (σ) kernel parameter in next iteration. In this article, the performance of two kinds of boosted SVM, AdaBoost.M1 and σ Boost with two different penalty parameters (c=20 and c=30) have been compared for cloud detection. This article has four sections. In section one, the theory of support vector machine, AdaBoost.M1, σ Boost, methods for combining SVM classifiers are described. In section two methodologies are explained. In section three the

* Corresponding author

experimental results are discussed and finally, in section four, the conclusion is explained.

2. METHODOLOGY

In this article, support vector machine classifier was used as a base learner of boosting algorithm and base classifiers were fused at decision level by majority vote method. At the following an overview of SVM classifier, boosting algorithm and methods for fusing SVM classifiers are discussed.

2.1 Support Vector Machine (SVM)

SVM was developed from the theory of structural risk minimization (Li et al., 2008). In a binary classification problem the decision function of SVM is:

$$f(x) = \langle w, \phi(\mathrm{x}) \rangle + \mathrm{b} \qquad (1)$$

In above equation, $\phi(x)$ is mapping sample x from input space to a high dimensional feature space. $\langle .,. \rangle$ denotes the dot product in feature space. The optimization problem for finding optimal values of w and b can be expressed as follows:

$$\text{minimize: } \mathrm{g(w,}\xi\mathrm{)} = \frac{1}{2}\left\| w^2 \right\| + C\sum_{i=1}^{N} \xi_i \qquad (2)$$

$$\textit{subject to: } \mathrm{y}_i(\langle \mathrm{w}, \phi(\mathrm{x}_i) \rangle + \mathrm{b}) \geq 1 - \xi_i, \quad \xi_i \geq 0 \qquad (3)$$

Where ξ_i is the ith slack variable and C is the regularization parameter. This optimization problem can be written as:

$$\text{minimize: } \mathrm{W}(\alpha) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j) \qquad (4)$$

$$\text{subject to: } \sum_{i=1}^{N} y_i \alpha_i = 0, \forall \mathrm{i}: 0 \leq \alpha_i \leq C \qquad (5)$$

Where α_i is the Lagrange multiplier corresponding to sample x_i and k(.,.) is a kernel function that maps input vectors into a suitable feature space.

$$k(\mathrm{x}_i, \mathrm{x}_j) = \langle \phi(\mathrm{x}_i), \phi(\mathrm{x}_j) \rangle \qquad (6)$$

By using the suitable kernel function (in this article the Gaussian kernel was applied) the samples are mapped nonlinearly into a high dimensional feature space. In this space, an optimal separating hyper-plane is constructed by the support vectors. The generalization performance of SVM is highly affected by SVM parameters, for example, σ and regularization parameter C. They have to be set beforehand.

2.2 AdaBoost.M1

AdaBoost was proposed initially for two classes and then extended for multiple classes. The most straightforward implementation of AdaBoost is AdaBoost.M1. The steps of this algorithm were shown in Fig.1. (Note that in this article we did not use resampling version of AdaBoost):

2.3 σ Boost

In (Li et al., 2008), it was mentioned that the accuracy of Boosting is highly related to the diversity of weak learners that compose the ensemble. This method use changing the Gaussian width RBF-SVM kernel for changing the internal structure of the weak classifier's kernels and creating diversity in the ensemble. The steps of this algorithm for binary classification were shown in Fig.2. It can be easily generalized to multi-class classification.

2.4 Methods for aggregating SVMs

After training, we need to aggregate several independently trained SVMs in an appropriate combination manner (Kim et al., 2003). The combination methods can be divided into two categories linear and non-linear combination methods. Majority vote and least square based weighting are linear methods and double layer hierarchical combining methods are nonlinear. In this method, an upper-layer SVM is used to combine several lower level SVMs.

2.4.1 Majority voting

Majority vote is the simplest method for aggregating SVMs. Let f_k(k=1,2,…,k) be a decision function of kth SVM in the SVM ensemble and C_j (j=1,…,C) denote a label of jth class. Then, the final decision of SVM ensemble $f_{mv}(x)$ for a given test vector x due to majority voting is determined by

$$f_{mv}(\mathrm{x}) = \arg \max_j \mathrm{f}_j \qquad (7)$$

2.4.2 The LSE-based weighting

The LSE-based weighting treats several SVMs in SVM ensemble with different weights. Often, the weights of SVMs are determined based on the accuracy of their classification. Let f_k(k=1,2,…,k) be a decision function of kth SVM in the SVM ensemble. The weight vector w can be obtained by $w_E=A^{-1}y$,where $A=(f_j(x))_{K\times L}$ is a matrix that contains result of each classifier for each training data (L is the number of training samples) and y is label vector of training data. The final decision of SVM ensemble $f_{mv}(x)$ for a given test vector x based on the LSE-based weighting is determined by

$$f_{LSE}(\mathrm{x}) = \mathrm{sign}(\mathrm{w}.[\mathrm{f}_i(\mathrm{x})_{k\times 1}]) \qquad (8)$$

2.4.3 The double layer hierarchical combining

In this method, the result of several SVMs is feed into a super SVM in the upper layer (Kim et al., 2003). Let f_k(k=1,2,…,k) be a decision function of kth SVM in the SVM ensemble and F be the decision function of super SVM in upper layer. The final decision of SVM ensemble $f_{mv}(x)$ for a given test vector x based on the double layer hierarchical combining is determined by

$$f_{SVM}(\mathrm{x}) = F(\mathrm{f}_1(\mathrm{x}), \mathrm{f}_2(\mathrm{x}), ..., \mathrm{f}_k(\mathrm{x})) \qquad (9)$$

Where k is the number of SVMs in the ensemble.

3. IMPLEMENTATION

The domain of case study was shown in section 3.1. Feature selection procedure was explained in section 3.2 and setting parameters of σ Boost algorithm was mentioned in section 3.3.

3.1 Case study

A cloudy Landsat 8 satellite image was selected as the case study. This image shows Alborz Mountains in Iran with latitude and longitude as following (Fig.3). Fig.4 shows work flow of our study.

Sample data of this study was selected by random sampling from the image. 70% of this data was selected randomly as training and 30% put away as validation using holdout method.

3.2 Feature selection

Co-occurrence matrix features were selected for classification. Because SVM classifier is sensitive to the correlation between input features, the Principal Component Analysis (PCA) method was applied to these features to discard dependency between them. These features are listed in Table 1. In Table 1 *p(i,j)* is *(i,j)th* element of normalized gray level co-occurrence matrix, $p_x(i,j)$ Obtained by summing the rows of $p(\mathrm{i}, \mathrm{j})$ matrix:

$p_x(i,j) = \sum_{j=1}^{N} P(i,j)$, $p_y(i,j) = \sum_{i=1}^{N} P(i,j)$, N_g is the number of

Textural features	formula
mean	$\frac{1}{(2n+1)^2} \sum_{i=x-n}^{x+n} \sum_{j=y-n}^{y+n} p(i,j)$
Angular second moment	$\sum_i \sum_j \{p(i,j)\}^2$
contrast	$\sum_{n=0}^{N_g-1} n^2 \{ \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ \lvert i-j \rvert = n}}^{N_g} p(i,j) \}$ (Haralick et al., 1973)
correlation	$\frac{\sum_i \sum_j (ij)\, p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
variance	$\sum_i \sum_j (i-\mu)^2 p(i,j)$
Entropy	$-\sum_i \sum_j p(i,j) \log p(i,j)$
dissimilarity	$\sum_i \sum_j \lvert i-j \rvert p(i,j)$
homogeneity	$\sum_i \sum_j \frac{1}{1-(i-j)^2} p(i,j)$

Table 1. gray level co- occurrence features used for classification

gray levels in quantized image. This features were extracted from bands 2,3,4,5,6,7 and 9 (non-thermal bands) of sensor OLI of Landsat 8 and also for band 10 of TIRS. Also, OLI and TIRS bands were considered as a feature; therefore we have 63 features for non-thermal bands and 9 features for thermal bands.

1. initialize the parameters :
- set the weights $w^1 = [w_1, ..., w_n]$ $w_i = 1$
- initialize the ensemble $D = \varnothing$
- select maximum number of weak classifier L

2. for k=1,…,L

- build a classifier D_k with the whole training set
- calculate the error of classifier at step k by

$\varepsilon_k = \sum_{j=1}^{N} w_j^k l_j^i$ ($l_k^i = 0$ if D_k misclassifies z_j and else $l_k^i = 1$)

- if $\varepsilon_k = 0$ or $\varepsilon_k \geq 0.5$ ignore D_k, reinitialize w_j^k to $\frac{1}{N}$ and continue.
- else calculate

$\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$ where $\varepsilon_k \in (0, 0.5)$

- update the individual weights

$w_j^{k+1} = \frac{w_j^k \beta_k^{(1-l_j^i)}}{\sum_{i=1}^{N} w_i^k \beta_k^{(1-l_k^i)}}$, j=1,...,N

3. return weak classifiers $D = \{D_1, ..., D_L\}$ and $\beta_1, ..., \beta_L$

end

Classification phase

4. calculate the support for class ω_k by

$\mu_i(x) = \sum_{D_k(x)=\omega_i} \ln(\frac{1}{\beta_k})$

5. the class with maximum support is chosen as label for x

Fig.1: AdaBoost.M1 algorithm

3.3 Setting parameters of σ Boost algorithm

σ Boost algorithm has two parameters that must be considered before implementation of the algorithm, σ_{base} , and σ_{scale} , for setting σ_{base} , we searched through the RBF-SVM σ values that give cross validation accuracy between 90 and 95 percent on training data. Between those values, one desired value was selected as σ_{base}. Note that we should weaken SVM classifier by choosing a σ value that gives not very high accuracy. σ_{scale} is a parameter that determines sensitivity to error. High values of the σ_{scale} parameter, cause large change on σ of RBF kernel for next iteration. In this article, the value of this parameter was set to 0.8.

4. RESULTS AND DISCUSSION

Results of the AdaBoost.M1 method were shown in Fig.5. Classification with non-thermal bands converge with four and with thermal band converges with two iterations hence we have six base classifiers. At first iteration of non-thermal classification, we have only two classes and the snow class is not recognized. In the second iteration, there are some pixels as snow and in third the inner pixels of the snowy area has been well detected. The third classification has the maximum vote.

Boosting of SVM with thermal bands has only two iterations and we reach to training error of zero in the third classification. AdaBoost.M1 reaches to training error of zero in thermal bands faster than non-thermal. The votes of thermal bands are lower than non-thermal ones, therefore, the fusion results are more affected by classification results of non-thermal bands. For fusion of these weak classifiers, majority vote result was used. It can be seen in the final classification map that the predicted snow pixels have been a bit exaggerated, especially at the edges.

For training of each weak classifier at each iteration we used RBF kernel of SVM classification. Parameters of RBF kernel, C, and σ, updated in each iteration by grid search method and are different numbers for each base classifier.

Given

$S = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m) \mid x_i \in X, y_i \in \{-1, 1\}\}$ the input training set and T the number of iterations

Initialize $D_1(t) = 1, \forall (x_i, y_i) \in S$

Set $\beta_0 = 0$

For t=1,..., T do

Set $\sigma = \sigma_{base} + \sigma_{scale} \times \sigma_{base} \times \beta_{t-1}$

Train the base classifier providing distribution D_t

Obtain weak hypothesis $h_t : X \to \{-1, 1\}$

Calculate $e_t = \sum_{i=1}^{m} D_t(t) \lvert h_t(x_i - y_i) \rvert$

Calculate $\beta_t = \sqrt{\frac{e_t}{1-e_t}}$ and update the distribution $D_{t+1} = D_t(i) \times \beta_t^{1-\lvert h_t(x_i) - y_i \rvert}$

End for

Output: assign a vote to each weak classifier based on training error (vote of t th classifier is $\log(\frac{1}{\beta_t})$) and use majority vote result to obtain final hypothesis

Fig.2: σ Boost algorithm

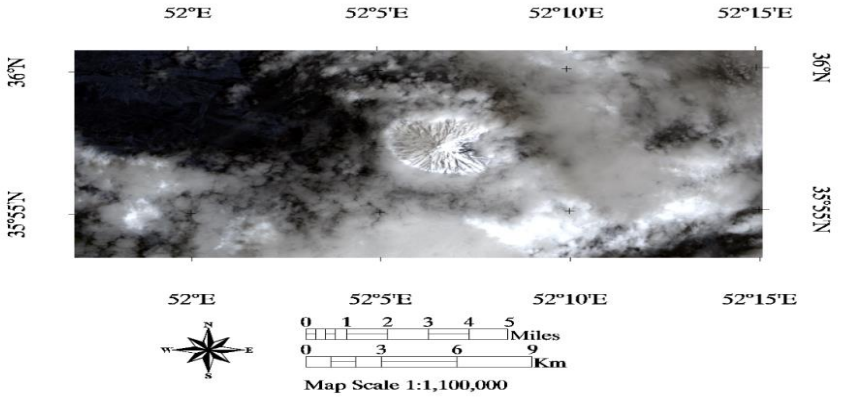

Fig.3: The selected case study

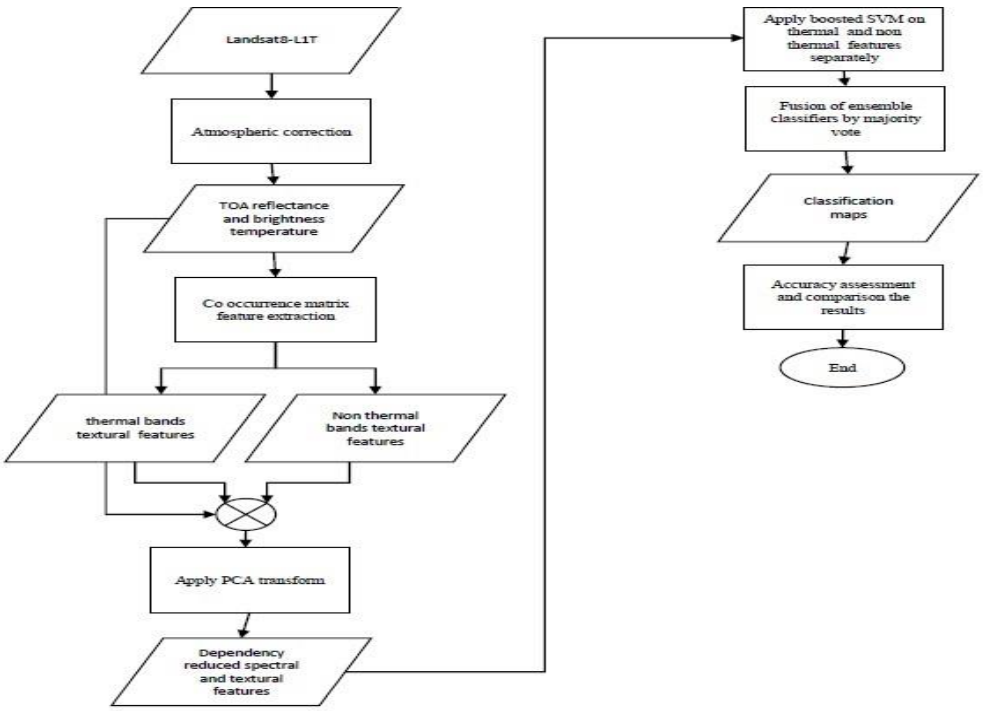

Fig. 4: work flow of procedure

In Fig.6 Boosting results of the σ Boost method are shown. Each weak classifier has a fixed value of regularization parameter (C) (here c=30) and σ was updated based on the training error in the former iteration. The initial value for σ was set by grid search method so that some training error exists. For increasing speed of the algorithm, the maximum number of iterations for both thermal and non-thermal bands was set to five in σ Boost method. There is an important point in vote value of classifiers. As we know in different kinds of Boosting methods the vote values are based on training error and this is not always a good Criterion in rating classifiers. For example, in the final iteration of thermal classification, all of the pixels have been detected as a cloud (blue) but it has a high vote. This may be related to the overfitting of classification model to training samples which are difficult to be classified. By reducing the C parameter from 30 to 20 the fusion results are improved (Fig. 7).

By reducing SVM C parameter, we decreased the cost of misclassification and overfitting classification model to training data. In final classification of the thermal band, the result of classification is the same as case c=30 and all pixels detected as cloud but the vote of the classifier is smaller than the similar iteration in former case with c=30 and this caused better fusion result. The quantitative results of AdaBoost.M1 and σ Boost (c=20 and c=30) were shown in Table 2. As shown in Table 2 σ Boost (c=20) got higher overall accuracy than other cases also it

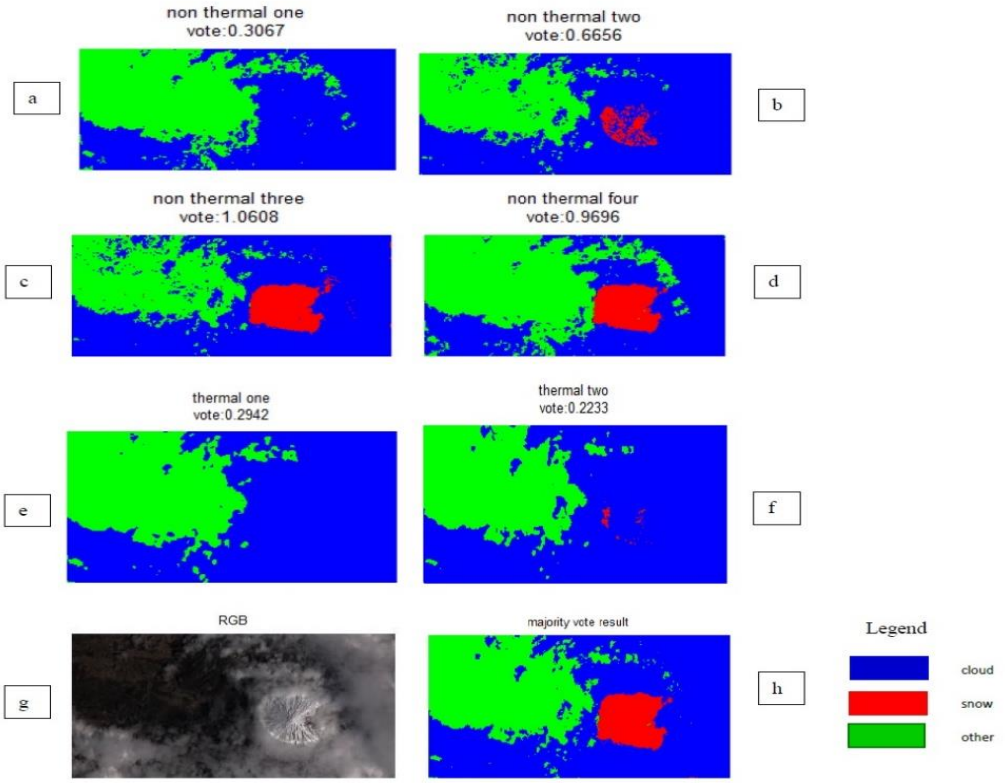

Fig. 5: a-f: classification results of each of weak classifiers .g: RGB image. h: fusion result

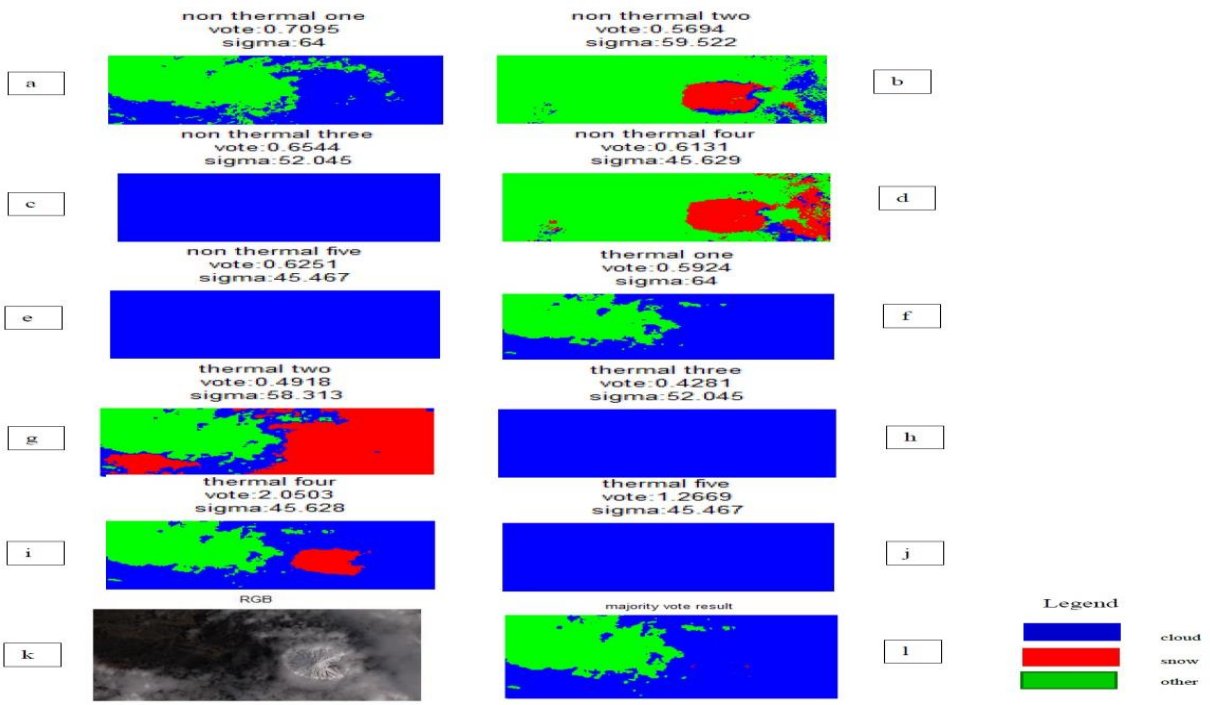

Fig. 6: a-e: classification results of σ Boost on non-thermal bands; f-j: classification results of σ Boost on thermal bands; k:RGB image; l: fusion result (c=30)

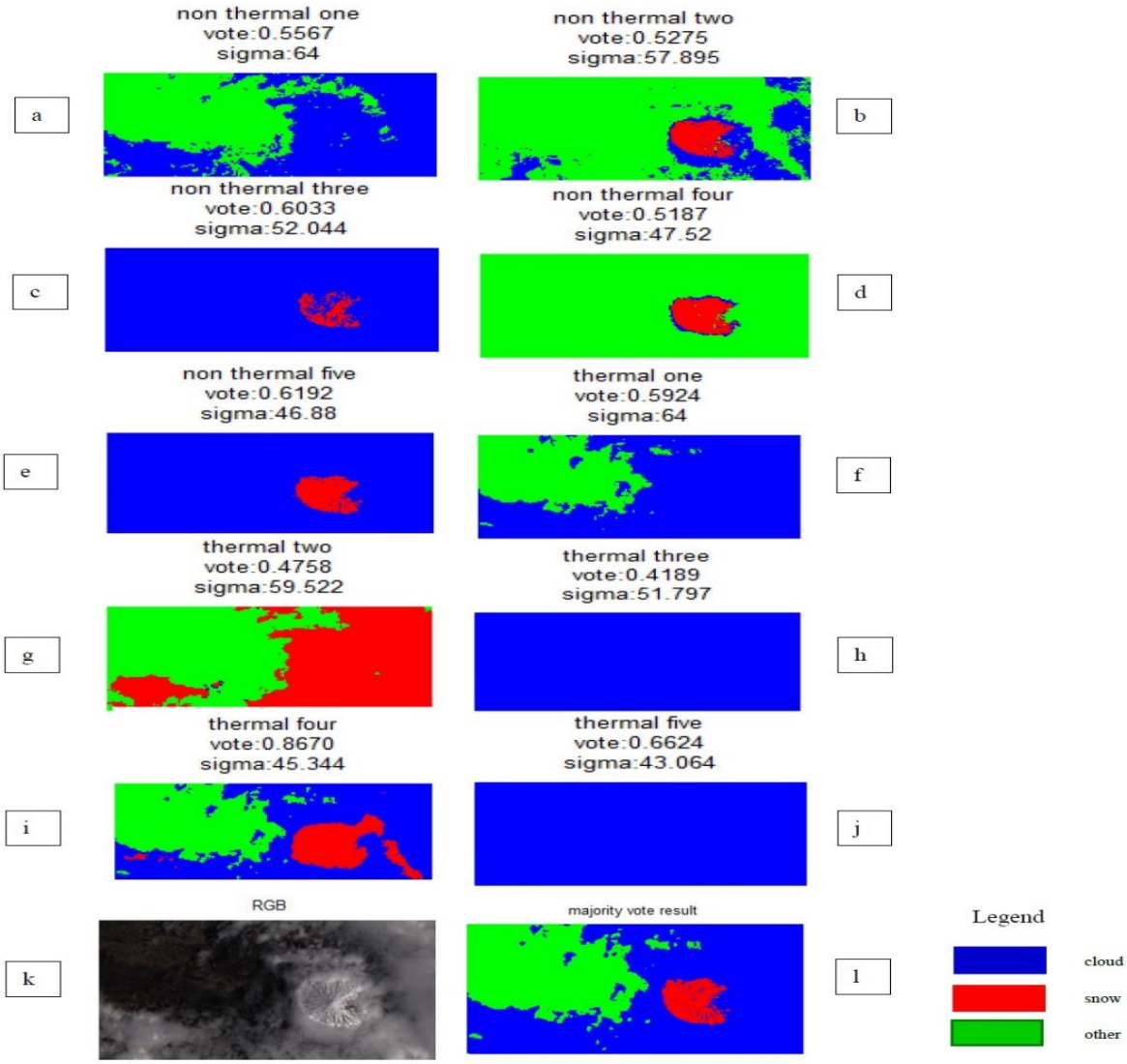

Fig. 7: a-e: classification results of σ boost on non-thermal bands; f-j: classification results of σ

shows higher accuracy in prediction of cloud pixels. This shows that by finding a suitable value for regularization parameter, σ Boost has this potential to achieve better accuracy result than AdaBoost.M1.

5. CONCLUSION

In this article, two kinds of Boosting algorithm including AdaBoost.M1 and σ Boost with SVM classifier as base learner were compared. Also the effect of changing penalty parameter (C) on accuracy results of two σ Boost with same $\sigma_{initial}$ ($\sigma_{initial=64}$) was examined. With reducing C parameter we can reduce misclassification cost and get more reasonable votes for base learners and this can improve fusion result. Results shows, with suitable C parameter, σ Boost can achieve higher accuracy results than AdaBoost.M1. In this work, this was shown that changing SVM penalty parameter in σ Boost algorithm effects on training error and hence σ updated values and finally output votes of boosting algorithm. So this effect is considerable.

The computation time of Adaboost.M1 and σ Boost does not differ a lot. Reasonable speed can be achieved if the number of iterations sets not very high and $\sigma_{initial}$ and σ_{scale} parameters are adjusted carefully. Changing C parameter does not effect on the number of iterations (computation time) of σ Boost. The σ_{scale} parameter controls the number of iterations.

In future works effect of boosting on other classification methods, such as decision tree, maximum likelihood and etc. for fusion of thermal and non-thermal satellite images can be examined.

Boosting method	Overall accuracy	Cloud producer accuracy	kappa
AdaBoost.M1	63%	52%	0.4
σ Boost (c=20)	74%	74%	0.53
σ Boost (c=30)	53%	60%	-0.09

Table 2. overall accuracy, cloud producer accuracy and kappa coefficient of AdaBoost.M1 and σ Boost with c=30 and c=20 values

REFERENCES

Azimi-Sadjadi, M. R. & Zekavat, S. A. Cloud classification using support vector machines. Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International, 2000. IEEE, 669-671.

Benediktsson, J. A., Chanussot, J. & Fauvel, M. 2007. Multiple classifier systems in remote sensing: from basics to recent developments. *Multiple Classifier Systems.* Springer.

De Colstoun, E. C. B., Story, M. H., Thompson, C., Commisso, K., Smith, T. G. & Irons, J. R. 2003. National Park vegetation mapping using multitemporal Landsat 7 data and a decision tree classifier. *Remote Sensing of Environment,* 85**,** 316-327.

Han, B., Kang, L. & Song, H. 2006. A fast cloud detection approach by integration of image segmentation and support vector machine. *Advances in Neural Networks-ISNN 2006.* Springer.

Haralick, R. M., Shanmugam, K. & Dinstein, I. H. 1973. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on***,** 610-621.

Kim, H.-C., Pang, S., JE, H.-M., Ki,m D. & Bang, S. Y. 2003. Constructing support vector machine ensemble. *Pattern recognition,* 36**,** 2757-2767.

Kuncheva, L. I. 2004. *Combining pattern classifiers: methods and algorithms,* John Wiley & Sons.

Li J., Menezel, W. P., Yang, Z., Frey, R. A. & Ackerman, S. A. 2003. High-spatial-resolution surface and cloud-type classification from MODIS multispectral band measurements. *Journal of Applied Meteorology,* 42**,** 204-226.

Li, X., Wang, L. & Sung, E. 2008. AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence,* 21**,** 785-795.

Lima, N. H. C., Neto, A. D. D. & De Melo, J. D. Creating an ensemble of diverse support vector machines using adaboost. Neural Networks, 2009. IJCNN 2009. International Joint Conference on, 2009. IEEE, 1802-1806.

Pal, M. 2008. Ensemble of support vector machines for land cover classification. *International journal of remote sensing,* 29**,** 3043-3049.

Tian B., Shaikh, M. A., Azimi-Sadjadi, M. R., Haar, T. H. V. & Reinke, D. L. 1999. A study of cloud classification with neural networks using spectral and textural features. *Neural Networks, IEEE Transactions on,* 10**,** 138-151.