Resolving the Celestial Classification using Fine k-NN Classifier

Sangeeta Yadav , Dr. Amandeep Kaur
Computer Science and Technology
Central University of Punjab
Bathinda, Punjab, India
shongmusic@gmail.com

Neeraj Singh Bhauryal
Centre for Applicable Mathematics
Tata Institute of Fundamental Research
Bangalore, India
neerajbhauryal@yahoo.com

Abstract **--With the rapid growth in space technology, space exploration is on the high demand. With each such type of mission, data is accumulating in heaps. Be it manned or unmanned mission, its credibility is defined by the quality of research which can be conducted on the data collected in such missions through remote or on the capsule experiments. Thus there is huge demand of soft techniques, which can make the space or celestial data as useful as possible. One of the major issues is dearth of automated technique for image classification of celestial bodies. Though many image classification techniques exist, but none of them is totally attributed to celestial bodies. An artificial neural network based classifier is proposed to classify celestial object from its image. Texture features are extracted from 90 images of size of 225*225 of different planets. Different classifiers were applied on this training data. Accuracy of different classifiers is compared to find out the best classifier for space data classification. Different validation schemes are applied and the results are compared to figure out the best validation scheme.**

Keywords-k-NN, Image histogram, Fine KNN, Planetary image

I. INTRODUCTION

It is an ever remaining interest for scientist and folks to know our universe more and more. And this thrust enables them to reach more and more unreached celestial objects. It can be Mars, Venus etc. With each space expedition, the data is rising in heaps. The size of data is so big that it will take months to classify it manually. Being in a digital era, this manual classification must be replaced by some automated image classification technique to reduce the manual work of space scientist. Hence there is a need to develop an image classification technique which fits best for celestial data. This paper will describe an image classification technique for celestial data.

Planets (celestial objects) can be differentiated by their color, shape of internal objects in them, texture and several other features. Given a large set of images of planets, it becomes difficult and time consuming for a human being to classify these images of planets[5]. The paper has proposed an artificial neural network based classifier to classify planet images in an accurate and efficient way.

II. LITERATURE REVIEW

First time the neural network was formulated and analyzed by Hodges and Fix[1]. They gave a technique using which an unknown class can be given the label carried by its most closest k neighbors. It was further used by Johns to give an instance of Bayes Rule. There are many techniques available for image classification which can be categorized broadly into three categories (1). Supervised classification (2). Unsupervised classification (3). Mixture of both previously mentioned techniques. Some techniques are described in section III.F. Fine K-NN is a widely known technique for sparsely dense data[13]. Sebestyen and Nilsson[2] have explained the implicit meaning of KNN and prescribed it for problems of pattern recognition. It is a non parametric method of classification[14].Its produces a class membership derived from voting by the nearest neighbors.

Fine KNN have high and an unique capability to reduce the size of features. It can be used to select and extract the features together[8].For very large datasets, fine KNN can be made efficient by introducing minor improvements[9].Many other authors[3] have focused on the minor modifications[6] so that the fine KNN can give a better estimate of a probability density function.

Classification is generally done on the shape and texture features[10].Feature selection is done by taking following things into consideration[17] 1.ease of use. 2 time taken for the training 3. least overfitting[15]. Sometimes the contrast of the image is also considered[9][11].Common drawbacks of these filtering is that it requires a high manual intervention[12].There is a need of a reliable technique which can classify with least human intelligence and intervention.

III. PROPOSED TECHNIQUE

In this section, a brief explanation of some terms is formally explained to aid the understanding of further experimental section

A. Image histogram

It is a representative measure of intensity distribution of an image data. It plots the frequency of each intensity value in an image. It is rotational invariant and by just seeing it once, a viewer can roughly estimate the entire intensity distribution.

978-1-5090-3669-1/16/$31.00 ©2016 IEEE

B. *Texture Feature*

It is a set of measures calculated to quantify the observed texture of an image. It is the information of location of different colour intensities in an image[7]. It plays an important role in differentiating between planet images.

C. *Fine KNN*

It is commonly used for its ease of interpretation and low calculation time. Training dataset is a feature space of multiple dimensions and has a classified label attached to it. In training stage the dataset is labeled as per the class. In classification period, k is a user defined test constant. It defines the number of neighboring observation sets which will be considered to vote for the query point. K can be selected by using different validation schemes. Curve of validation error and K value attains a minimum value; this minimum point is the best choice for K. This value of K gives best results. Each time a test feature vector is in input stream, select a K for it. Label with majority of K neighbors will be assigned to the test image. Generally the vote is considered on the basis of Euclidean distance.

D. *k-nearest neighbour*

It is a non-parametric classifier. It can be applied only to k closest training examples (in the feature space). Its output is based on class membership which means that the object will be assigned to the most occurring class in its neighborhood. K=1 is a singleton class case and is assigned to its own class. It has a drawback of sensitivity to local data.

E. *Support Vector Machines*

It is based on a hyperplane construction method in a multidimensional space. The hyperplane is selected which is at a large distance from the training data point of any class to reduce the generalization error. To ease the separation, original feature space is generally mapped to higher dimensions. A kernel function is selected such that the dot products of variables in original space can be computed easily.[3]

Hyperplane is a set of points which gives a constant dot product with a vector in same space. These vectors can be generated by linear combinations with feature vector parameters of an image. Following relation can generate this mapping:

$$\sum_j l_j * k(x_j, x) = constant.$$

Where l_j is the image parameter, k is kernel function and x_j is feature vector.

F. *Classifiers used in the classification:*

Other than fine KNN various other classifiers are also used for the classification as given below:

1. Fine KNN
2. Complex Tree
3. Medium Tree
4. Simple Tree
5. Linear Discriminant
6. Quadratic Discriminant
7. Linear SVM
8. Quadratic SVM
9. Cubic SVM
10. Fine Gaussian SVM
11. Medium Gaussian SVM
12. Coarse Gaussian SVM
13. Medium KNN
14. Coarse KNN
15. Cosine KNN
16. Cubic KNN
17. Weighted KNN
18. Boosted Trees
19. Bagged Trees
20. Subspace Discriminant
21. Subspace KNN
22. RUSBoosted Trees

IV. METHODOLOGY

A. *Image database collection:*

Though there are many databases available for space expedition data but there is no online database for planets exclusively. Different images of all the planets are downloaded from the internet. Images with extreme brightness and haziness are not included.

B. *Pre-processing of Images*

Images of different sizes are taken. It provides randomness to the training set and thus gives more generic results. Few images of the database are given below:

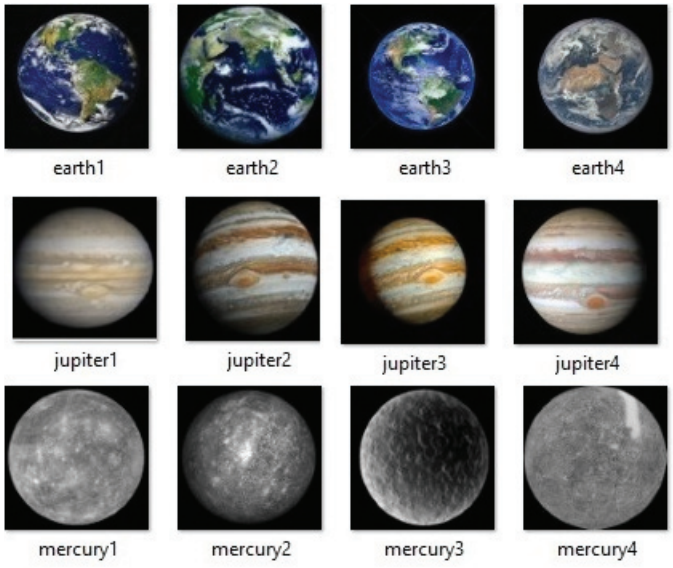

Fig 1.Example of training dataset of planetary images

C. *Separation of RGB components of the images:*

For further feature extraction operations, Store R, G, B components of the image in separate matrices.

D. *Histogram calculation of images:*

Calculate the frequency of each intensity value ranging from [1….256] for each channel. Concatenate all histogram vectors to produce a single one dimension feature. Plot of this histogram is shown in the figure 1 where x axis represents the intensity values and their frequency of the occurrence of these intensity values is represented on y axis.

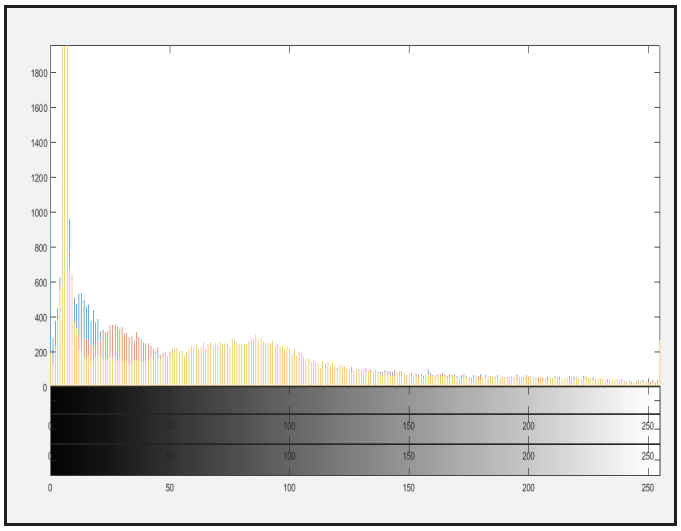

Fig 2. Histogram of RGB channel of planetary images

E. *Labeling the responses:*

Create an array in which all the feature vectors are stored row by row and the last column is used to label the data. For example earth will be written against the feature vector for earth and similarly for Venus, Mercury etc.

F. *Select the validation scheme:*

Validation scheme defines the ratio in which the data will be divided among training and testing sets.

Data can be validated using following schemes

1. Cross –Validation: Data will be divided in k folds. At a time one fold will be treated as testing data and rest will be training data .Similarly each fold will be tested once against other folds and accuracy will be average of accuracy of all these folds.

2. Holdout Validation: In this a % of data will held out for testing against rest of data. It is more applicable to large data sets.

3. No validation: In this whole dataset is tested again itself .Hence it gives imaginary results.

In the paper, results for cross validation and holdout validation for many classifiers are given in table1 and table2 respectively.

G. *Train and testing*

Train the classifier using the cross validation and hold out validation scheme. Test the images and calculate Precision, Recall and accuracy of the classifier.

V. EXPERIMENTAL RESULTS

Various Classifiers are used to classify the planetary data and the accuracy of classification is given in table 1 and table 2.

Where Accuracy =True Positive Rate /False Positive Rate

Given that True Positive Rate is the proportion of positives that are correctly identified and false positive rate is the proportion of negatives that are correctly identified.

VI. CONCLUSION

From the results for holdout validation scheme given in table 2, it is concluded that on choosing larger number of disjoint sets of the training data, trees gives best accuracy. With increase in the % of held out in holdout validation scheme SVM and KNN proves to be best classifiers. Hence we can say that SVM and KNN are best to classify planetary data using holdout validation scheme.

And for cross validation scheme as per the results given in table 1, Fine KNN gives consistent results on all number of folds. Though bagged trees have higher accuracy but it is only in some specific case. So overall fine KNN gives consistent results and better accuracy for planetary classification

VII. REFERENCES

[1] E. Fix and J. L. Hodges, Jr., “Discriminatory analysis, nonparametric discrimination,” USAF School of Aviation Medicine, Randolph Field, Tex., Project 21-49-004, Rept. 4, Contract AF41(128)-31, February 1951.

[2] Nils Nilsson, “Learning Machines”. New York: McGraw-Hill, 1965, pp. 120-121.

[3] Altman, N. S. (1992). "An introduction to kernel and nearest- neighbor nonparametric regression". The American Statistician. 46(3): 175-185.

[4] William H.; Teukolsky, Saul A.; Vetterling, William T.;Flannery, B. P. (2007). "Section 16.5. Support Vector Machines".Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: Cambridge University Press.

[5] T. F. Stepinski, W. Ding, and R. Vilalta, “Detecting Impact Craters in Planetary Images Using Machine Learning,” *ntelligent Data Anal. Real-Life Appl. Theory Pract. IGI Glob.*, pp. 1–12, 2011.

[6] Y. Xu, Q. Zhu, Z. Fan, M. Qiu, Y. Chen, and H. Liu, “Coarse to fine K nearest neighbor classifier,” *Pattern Recognit. Lett.*, vol. 34, no. 9, pp. 980–986, 2013.

[7] M. Yang, K. Kpalma, and J. Ronsin, *A survey of shape feature extraction techniques*, vol. 2008, no. November. 2008.

[8] Tang, S., Chen, H., Lv, K., & Zhang, Y. (2015). Large Visual Words for Large Scale Image Classification 1 *. *International Conference on Image Processing (ICIP)*, 1–5.

[9] L. Bandeira, W. Ding, and T. F. Stepinski, "Automatic Detection of Sub-km Craters Using Shape and Texture Information," in Proceedings of the 41st Lunar and Planetary Science Conference, Mar. 2010.

[10] E. R. Urbach and T. F. Stepinski, "Automatic detection of sub-km craters in high resolution planetary images," Planetary and Space Science, vol. 57, no. 7, 2009.

[11] W. Ding, T. F. Stepinski, Y. Mu, L. Bandeira, R. Ricardo, Y. Wu, Z. Lu, T. Cao, and X. Wu, "Sub-kilometer crater discovery with boosting and transfer learning," ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 4, Jul. 2011.

[12] Cohen, J. P., Lo, H. Z., Lu, T., & Ding, W. (2016). Crater Detection via Convolutional Neural Networks. Retrieved from http://arxiv.org/abs/1601.00978

[13] Wang, Y., Yang, G., & Guo, L. (2015). A novel sparse boosting method for crater detection in the high resolution planetary image. *Advances in Space Research*, *56*(5), 982–991. https://doi.org/10.1016/j.asr.2015.05.014

[14] Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". *The American Statistician*. **46** (3): 175–185. doi:10.1080/00031305.1992.10475879.

[15] Bermingham, M. L., Pong-Wong, R., Spiliopoulou, a, Hayward, C., Rudan, I., Campbell, H, Haley, C. S. (2015). Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Scientific Reports*, *5*, 10312. https://doi.org/10.1038/srep10312

[16] Tahir, M. A., Bouridane, A., & Kurugollu, F. (2007). Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier. *Pattern Recognition Letters*, *28*(4), 438–446. https://doi.org/10.1016/j.patrec.2006.08.016

[17] Duval, B., Hao, J.-K., & Hernandez Hernandez, J. C. (2009). A memetic algorithm for gene selection and molecular classification of cancer. *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation - GECCO '09*, 201. https://doi.org/10.1145/1569901.1569930

[18] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

Table 1

S. No.	Type of Classifier/Cross Validation scheme	10 folds	20 folds	30 folds	40 folds
1	Complex Tree	71.1	82.2	85.6	83.3
2	Medium Tree	71.1	82.2	85.6	83.3
3	Simple Tree	67.8	76.7	83.3	81.1
4	Linear Discriminant	75.6	76.7	76.7	76.7
5	Quadratic Discriminant	81.1	78.9	77.8	77.8
6	Linear SVM	76.7	76.7	80	76.7
7	Quadratic SVM	78.9	82.2	83.3	82.2
8	Cubic SVM	82.2	84.4	86.7	85.6
9	Fine Gaussian SVM	84.4	85.6	85.6	85.6
10	Medium Gaussian SVM	85.6	87.8	88.9	87.8
11	Coarse Gaussian SVM	73.3	73.3	73.3	72.2
12	Fine KNN	84.4	86.7	86.7	86.7
13	Medium KNN	71.1	73.3	72.2	72.2
14	Coarse KNN	33.3	22.2	33.3	11.1
15	Cosine KNN	64.4	68.9	67.8	63.3
16	Cubic KNN	63.3	68.9	65.6	67.8
17	Weighted KNN	82.2	85.6	85.6	85.6
18	Boosted Trees	33.3	22.2	33.3	11.1
19	Bagged Trees	87.8	91.1	92.2	91.1
20	Subspace Discriminant	72.2	68.9	76.7	74.4
21	Subspace KNN	81.1	85.6	83.3	83.3
22	RUSBoosted Trees	33.3	44.4	33.3	15.6

Table 2

S. No.	Type of Classifier/% of held out	Holdout Validation with given % held out				
		10%	20%	30%	40%	50%
1	Complex Tree	100	72.2	70.4	86.1	81
2	Medium Tree	100	72.2	70.4	86.1	81
3	Simple Tree	100	72.2	70.4	80.6	77.8
4	Linear Discriminant	88.9	66.7	55.6	80.6	73
5	Quadratic Discriminant	88.9	72.2	70.4	72.2	76.2
6	Linear SVM	88.9	77.8	74.1	83.3	74.6
7	Quadratic SVM	88.9	88.9	81.5	88.9	77.8
8	Cubic SVM	88.9	88.9	77.8	91.7	85.7
9	Fine Gaussian SVM	77.8	83.3	77.8	86.1	85.7
10	Medium Gaussian SVM	88.9	83.3	81.5	88.9	79.4
11	Coarse Gaussian SVM	77.8	72.2	59.3	83.3	66.7
12	Fine KNN	100	83.3	74.1	88.9	87.3
13	Medium KNN	88.9	55.6	63	63.9	54
14	Coarse KNN	33.3	33.3	33.3	33.3	31.7
15	Cosine KNN	66.7	66.7	59.3	69.4	55.6
16	Cubic KNN	77.8	38.9	55.6	55.6	60
17	Weighted KNN	88.9	77.8	74.1	91.7	68.9
18	Boosted Trees	33.3	33.3	33.3	33.3	31.7
19	Bagged Trees	100	83.3	74.1	88.9	85.7
20	Subspace Discriminant	88.9	66.7	74.1	69.4	71.4
21	Subspace KNN	100	66.7	77.8	75	82.5
22	RUSBoosted Trees	33.3	33.3	33.3	33.3	44.4