An Approach to Improve Nearest Neighbor Classifier Performance

R. Raja Kumar[1*], P. Viswanath[2] and C.Shobha Bindu[3]

[1]Dept of CSE, RGMCET, Nandyal, India,rajsri1229@yahoo.co.in
[2]Dept of CSE, IIITS-Chittoor, India, viswanath.p@iiits.in
[3]Dept of CSE, JNTUA, Anantapuramu, India, shobabindhu@gmail.com

Abstract.

This paper extends our proposed method [1]. In this paper, comparative analysis of our method is presented and the results of the proposed work[1] are enhanced further. The method is based on retaining examples which are near to the decision boundary, while eliminating the rest. The prototype set thus selected is a subset of original training set. The method is empirically compared against several relevant methods using several standard data sets. The results obtained shows that the proposed method has the advantages over the existing techniques discussed in this paper. The proposed technique has the ability to build the reduced set (Prototype set), in a faster way. It requires only one scan of the dataset/training set/database over which the Prototype Set is building. Finally these advantages are achieving without sacrificing accuracy which is the core component in measuring the ability of classification process.

Keywords: Other Class Nearest Neighbors, Pattern Recognition, Prototypes, nearest neighbors

1. Introduction

Nearest Neighbor classification (NNC) is a classification technique in Pattern recognition[2]. This is simple method and easy to understand. Given test pattern, it finds thenearest neighbor of the given test pattern and the class label of test patterns is inferred to thetest pattern[3]. A simple extension of NNC is to consider the k nearest neighbors and the class label is given based on majority voting decision.

Besides its popularity, NNC suffers severe drawbacks also. It stores the training set, as a result the space complexity is O(m), where m is the number of training patterns. Next, it has to find the similarity (or distance) with each and every training pattern, therefore the time complexity is also O(m). Note, for every test pattern the time taken is O(m). Next, outliers or noisy training patterns can affect its performance. Indexing can be used to reduce the time required to find the nearest neighbor. R-Tree or KDTree can be used for this purpose. The training set patterns information is stored in the multidimensional tree using these trees[4]. But these trees have the drawback that the design time is increased and does not address the outliers or noise problem.

Prototype selection method is one such method where smaller set of patterns(Prototypes) are selected. These Prototypes are the patterns present in the training set or new patterns derived from the training set.

There were papers on how to reduce the training set. The early method in thiscontext proposed by Hart called Condensed Nearest Neighbor(CNN)[5].

This methodstarts from empty set, a random pattern is selected from the training set. Using this set,the training set is classified. All the misclassified patterns are put into the new set. Thiswill be repeated till there are no patterns added to the set. The new set now called ascondensed set. The condensed set is used to classify unlabeled patterns. The condensedset is subset of the original training set and it achieves 100% classification accuracy overthe set from which it has drawn. CNN has two drawbacks 1) keeping unnecessary samplesand 2)occasional storing of internal patterns instead of boundary patterns[6]. Ivantomek in 1976, proposed two modifications to CNN which overcome the above limitations[6]. Swonger found an iterative condensation algorithm(ICA)[7]which allowsadditions and deletions from the condensed set. Gates found reduced nearest neighborrule(RNN)[8] in 1972. This method is opposite to CNN approach. Here the trainingset is considered

as condensed set because by default a training set is a condensed set over itself. Instead of growing the condensed set as in CNN, now from this set unnecessarypatterns are removed without degrading classification accuracy. Proximity graphsare used by Sanchez, Pla, Ferri for prototype selection[9]. Devi and Murthy have usedstochastic techniques for prototype selection on optical character recognition data[10].Kuncheva and Jain used soft computing techniques like tabu search, simulated annealingand genetic algorithms[11,12].

The size of the condensed sets obtained by above methods is considerably smallwhen compare to the training set. Hence the nearest neighbor classification by using thecondensed set takes less computational time without sacrificing classification accuracysignificantly. The minimalist of the condensed set is a key point. A condensed set whichis having minimum number of patterns is called the optimal set which cannot be furtherreduced. Deriving such a set is complex task which depends on factors like number offeatures in the training set and range of the features. If the variance of the values ofthe features of a class, then less number of patterns are required to represent that classinformation.

In 2002 devi and murthy [13] proposed a prototype selection method called Modifiedcondensed nearest neighbor(MCNN). In this algorithm, prototype set is built in anincremental manner.

The rest of this paper is organized as follows. In section 2, working of nearest neighbor method is discussed. In section 3,algorithms are presented. In section 4 Numerical examples is included. In Section 5 Data sets are discussed. In section 6, Experimental results are presented. In section 7, an analysis of method is discussed. Section 8 presents the conclusion

2.Working of Nearest Neighbor Classification

NNC is a simple classifier used in the fields like Pattern Recognition, Machine Learning and Data Mining etc. We try to understand the bird's view of the NNC using Figure 1.

In Figure 1 two classes are present. The positive class patterns are represented by using '+' and negative class patterns are represented by '-'. The two classes are separated by the Piecewise Linear Decision Boundary. The true decisionboundary is notknown by definition, so the next best alternate is Piecewise Linear Decision Boundary[14]. Let a query or test pattern '*' has to be classified. NNC by rule, finds the nearest neighbor among all the patterns and the query pattern is classified accordingly. The nearest Pattern for the query pattern '*' is '-'. Hence the query pattern '*' belongs to negative class example by NNC rule.

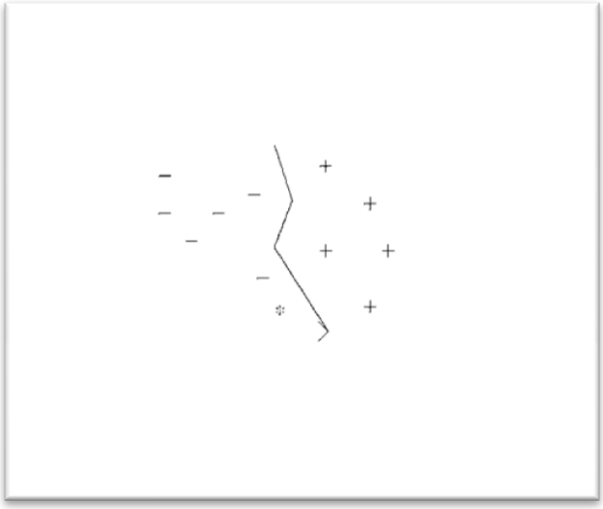

Figure 1: Piecewise Linear Decision Boundary

Let us understand NNC using a sample data set presented in Table 1. A and B are the two attributes and the class label is in the last column. Class Label set is {Yes, No}.

Table 1: Data Set

No.	A	B	Class
Y_1	1	1	Yes
Y_2	1	2	Yes
Y_3	2	1	Yes
Y_4	2	2	Yes
Y_5	5	1	No
Y_6	5	2	No
Y_7	6	1	No
Y_8	6	2	No

Let a query pattern Q= (2.5, 2.5) has to be classified. Then NNC finds the distancesfrom this query pattern 'Q' to all the patterns in the dataset 1. We used Euclideandistance for finding Nearest Neighbor. The distances from the pattern 'Q' to the patternsin the dataset shown in table 1 are shown in the table 2. The Nearest Neighbor for 'Q'is pattern *Y*4. The class label of *Y*4 is Yes. Hence 'Q' is inferred the class label Yes and it is said to be 'Q' belongs to class 'Yes'. The rule used here is known as NNC or 1-NNCsince only one nearest neighbor is taking into picture to decide class label.

If more than one nearest neighbor is consider inferring the class label, then it isk-NNC. k refers to

the no of neighbors to be considered. For example, the pattern 'Q' is assigned the class label by majority voting. It is shown in the table 3.

Table 2: Distance (Euclidean) between 'Q' and patterns of Table 1

No.	Q
Y1	2.12
Y2	1.75
Y3	1.75
Y4	0.70
Y5	2.91
Y6	2.54
Y7	3.80
Y8	3.53

Table 3: Class Label Inferred to 'Q' using k-NNC

k	Yes	No	Class label
1	1	0	Yes
2	2	0	Yes
3	3	0	Yes
4	4	0	Yes
5	4	1	Yes
6	4	2	Yes
7	4	3	Yes
8	4	4	Tie

The value of the 'k' should be chosen carefully, other wise this may lead to a 'tie'situation as shown in the Table 3. Ties can be resolved by random voting. If k=1, thenk-NNC is equal to 1-NNC and If k=m where m is the number of patterns in the dataset,then every pattern in the dataset has to be considered.

3 Our Methodology

The proposed method keeps the patterns that are close to decision boundary. Obviouslythese patterns play very important role in classifying a query pattern. From Figure 1 itcan be observed that the two pattern from '-' class and two patterns from '+' class areactually deciding the class label of query pattern '*'. These patterns are the nearest patternsto the decision boundary. The other patterns are ignored from the training set. So,the patterns close to decision boundary can be retained and remaining can be discarded.This is the central theme of our method. We named this method as Other Class NearestNeighbor algorithm (OCNN) discussed in the Section 4.

4 Algorithms and Notations

4.1 Algorithm: CNN

The steps followed in CNN algorithm are given in detail in[15]. This iterative processis carried out till there are no misclassified patterns or samples. The condensed set hastwo properties[15].

4.2 Algorithm: OCNN

Let D be the given training set. A tuple in D is represented as (X_i,L_i)=1,2 ... n ,where$X_i$represents an *d*-dimensional feature vector and $L_i$represents the corresponding classlabel. $S_{Xi}$is the set of nearest neighbors from other classes with respect to pattern *Xi*. $Y_i\varepsilon$ {W_1, W2,W_n}where c is the number of distinct classes present in D.

OCNN (X_i) is the method to find the nearest neighbors in the other classes for the pattern $X_i$where i = 1, 2 . . . n and n is the training set size. The algorithm is outlined in Pseudo-code 1

Algorithm1 Other Class Nearest Neighbors(OCNN)

Begin
1 Initialize set S = { }
2. for (i=1;i<=n;i++)
3. {
4. FindS_{Xi}=OCNN(X_i)
5. S= S U S_{Xi}
6.}
7. S is the final prototype set.
End

Pseudo-code1:OCNN

The sub procedure OCNN (X_i) is outlined below in Pseudo-code2.

Algorithm 2: Sub procedure OCNN(X_i)
Begin
1. for all $X_j \varepsilon$ D
2. Find distances d(X_i,X_j) for all $i \neq j \cap Y_i \neq Y_j$
3. Arrange the distances in sorting order d(X_i,X_j) $\neq$ d(X_i,X_k)
4. Pick the first distances where k=1,2...n
End

Pseudo-code2:sub procedureOCNN(X_i)

The final prototype set obtained by the above algorithm is the reduced set. It is asubset of the training set. But it does not ensure the consistency

property. By using thismethod the size of the dataset reduces.

This procedure is applied on the example shown in the Figure 1. The result obtainedis shown in Figure 2. From the Figure 2, a subset S which consists of four patterns twofrom + class and two from - class are chosen by the algorithm. The ignored patterns areclose to boundary. The rule followed is k-NN. If k increases then the size of the subset Salso increases.

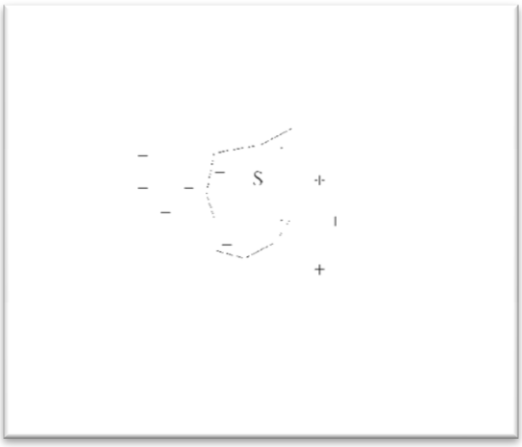

Figure 2: Patterns Selected by OCNN

Numerical Example: Let us consider dataset in table 1. Let $Y_i$refers to the pattern i in the dataset *i.e.*, *Y2*means (1, 2) in the dataset. The distances between the patterns arepresent in the table 4.

1) We understand the working of algorithm 1 and find the reduced set how it forms. The S_{Yi}=OCNN (Y_i) where i = 1, 2...8 obtained as follows. We considered k=2.

S_{Y1}=OCNN(Y_1)={Y_5,Y_6} S_{Y2}=OCNN(Y_2)={Y_5,Y_6}
S_{Y3}=OCNN(Y_3)={Y_5,Y_6} S_{Y4}=OCNN(Y_4)={Y_5,Y_6}
S_{Y5}=OCNN(Y_5)={Y_4,Y_3} S_{Y6}=OCNN(Y_6)={Y_4,Y_3}
S_{Y7}=OCNN (Y_7)={*Y*4,*Y*3} S_{Y8}=OCNN(Y_8)={*Y*4,*Y*3}

The reduced set S is computed by the union of all S_{Yi}For this example, S=S_{Y1}US*Y*2US$_{Y3}$US$_{Y4}$US$_{Y5}$US$_{Y6}$US$_{Y7}$US$_{Y8}$. The final set S={*Y*5,*Y*6,*Y*4,*Y*3}presented in table 5.

Table 5: Results of OCNN on table 1

No.	A	B	Class
Y3	2	1	Yes
Y5	5	1	No
Y4	2	2	Yes
Y6	5	2	No

2) The CNN finds the condensed set S={*Y*1,*Y*5}, which is the condensed prototype set on the data set on Table [15].

4.3 MCNN Method:

In this method, a typical pattern is selected from each class and it is put in a set. Thetraining set patterns are classified by using these typical patterns. From the wronglyclassified patterns, a representative pattern is computed per class wise. The training setis classified again with this set of prototypes. This process is an iterative processand repeated till there are no misclassification samples present in the training set. Nowthe final typical pattern set is the desired prototype set. It is a subset of original training set and consistent over the training set. In a
Class of patterns the pattern which is nearto centroid is selected as typical pattern.

5 Data Sets

In this section the data sets information is presented. We tested our method on 8 benchmark data sets. The data sets are taken from Murphy and Aha[16] except Optical CharacterRecognition (OCR) data set. It has 6666 training patterns and 3333 test patterns.It has 192 features and 10 class labels correspond from digit '0' to digit '9'. We havetaken 67% of data as train set and rest as training set. The detailed information aboutdata sets is presented in table 6. The datasets are divided into training and test sets. Thetraining sets are used to build the prototype sets and test sets are used for evaluating theperformance of the methods. Nearly $\frac{2}{3}$of the dataset is considered as training and $\frac{1}{3}$ofthe dataset is considered as test set.

Table 6: Data Sets Description

Data set	#of training Patterns	#of Test Patterns	# of features	# of classes
Wine	120	57	13	3
OCR	6670	3333	192	10
Thyroid	144	71	5	3
Liver	230	155	6	2
Iris	99	51	4	3
Balance	416	209	4	3
PIMA	512	256	8	2
Opt.digit.rec	3823	1797	64	10

6 Experimental Results

We tested our method with the eight standard data sets. The accuracies on these datasetsare presented in table 7. In this table, the accuracies obtained using OCNN, MCNN,CNN and accuracy obtained by using all the patterns in the training set is shown. We gotsatisfactory results. From these results, it is evident enough to say that our methodis competing with MCNN and CNN methods. On Iris, Liver, Pima data sets OCNNis showing better performance than all the other methods. On OCR, Iris, Liver, Pima,opt.digit.recognition data sets OCNN is performing better than MCNN method. The bestmethod is indicated with bold number. The italic style is used where our method performingbetter than MCNN. The graphical results on OCR data have shown in Figure 3and on IRIS data is shown in figure 4. The results over Balance, Liver, Opt.digit.rec andPimaare shown in the form of graphs in Figure 6, Figure 5, Figure 8 and Figure 7respectively.

Table 8 :Accuracy after applying Boot Strap Method

Dataset	OCNN	OCNN with Bootstrapping
OCR	90.15	92.8
Wine	94.74	95.01
Thyroid	92.1	95.99
Iris	98.33	98.89
Liver	73.03	74.14
Balance	64.04	67.82
PIMA	76.78	77.89
Optical.digit.rec	96.55	98.32

Table 7: Accuracy Obtained over data sets

	OCNN		MCNN		CNN		All Prototypes	
Data Set	No of Prototypes	Accuracy %	No of Prototypes	Accuracy %	No of Prototypes	Accuracy %	No of Prototypes	Accuracy %
OCR	1650	90.15	1527	88	1580	87.04	6670	92
Wine	45	94.74	14	94.74	27	94.74	120	94.74
Thyroid	38	92.1	18	95.77	16	91.55	144	94.37
Iris	22	98.33	10	92.16	16	96.08	99	94.12
Liver	154	73.03	136	72.17	150	64.35	230	63.48
Pima	329	76.78	250	67.97	266	65.23	512	69.14
Balance	214	64.04	157	74.64	143	71.77	416	77.03
Opt.digit.rec	622	96.55	325	94.32	305	96.38	3823	98

Further to improvise the results, we used Hama Moto Bootstrapping technique[17]to enhance the performance of OCNN. The Accuracy obtained over the datasets afterapplying bootstrapping method is shown in table 8. The accuracy is considerablyimproved and it is outperforming the CNN and MCNN methods.

7 Analysis

OCNN algorithm has the following properties and these properties make OCNN agood algorithm.

Property 1:OCNN algorithm converges in finite time and takes finite no of steps.

Proof: The algorithm builds the prototype set based on the Nearest Neighbors fromother class. Let there are n_1 patterns present in class C_1 and n_2 patterns are present inclassC_2. Let n=n_1+n_2 be the total patterns in the training set. In step 2 of the algorithm, aset $S_X$is computed which includes patterns from the opposite class. Nearest Neighborsare computed based on k-NN rule. There two extreme cases. When k=1, at least onepattern is selected. When k=n2, all the patterns from the opposite class is included.Since the no of patterns is finite, it computes$|S_X|$=n sets. Step 3 eliminates the duplicatepatterns and also computes final reduced prototype sets.

Property 2:OCNN algorithm is order independent.
Proof: The algorithm repeatedly computes the steps 2-3 till all the patterns are overfrom the training set. For each pattern, the nearest neighbors do not change until thereis no change in the distance measure and for the finite training set. Even if the patternsare selected in any particular order or in random order the final set S is same.

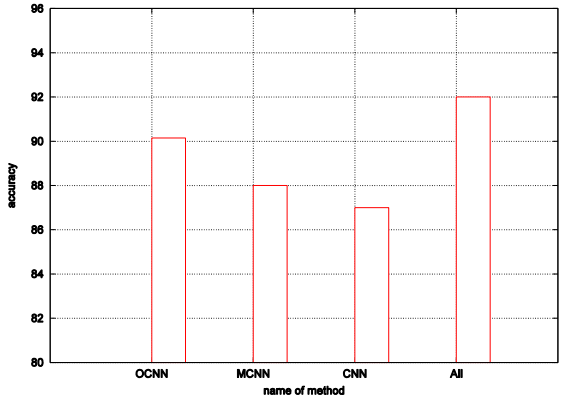

Figure 3: Accuracy on OCR

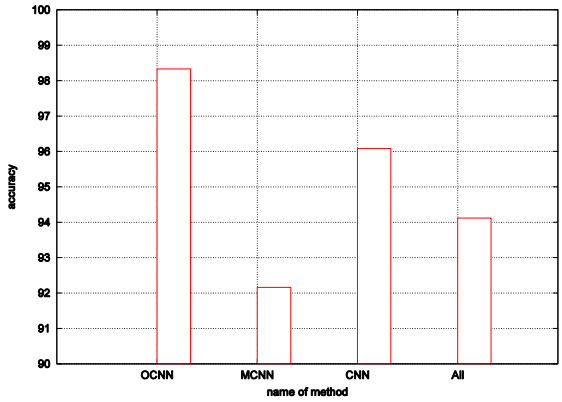

Figure 4: Accuracy on IRIS

7.1 comparisons with the CNN algorithm

– The CNN algorithm is order dependent[13]. It produces condensed set based on theorder of patterns selected. If the order of patterns in the training set changes thencondensed set also changes. But OCNN is order independent.
– The CNN algorithm requires multiple scans of the dataset (steps 3-6). This maylook good for small size datasets but when the dataset size is large, it takes substantialamount of time. OCNN requires only one scan of the dataset. From table9 and Figure 9, it is clear that approximately OCNN is taking $\frac{1}{2}$ of the CNN algorithm.

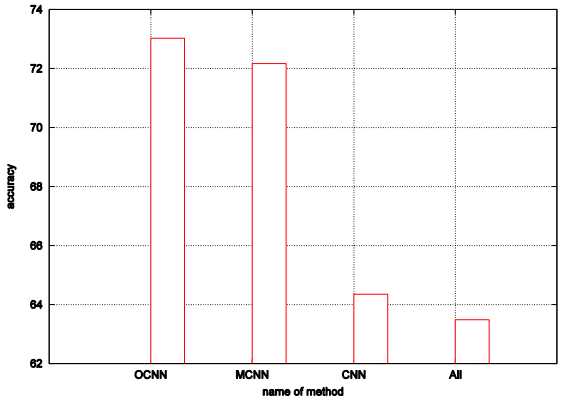

Figure 5: Accuracy on LIVER

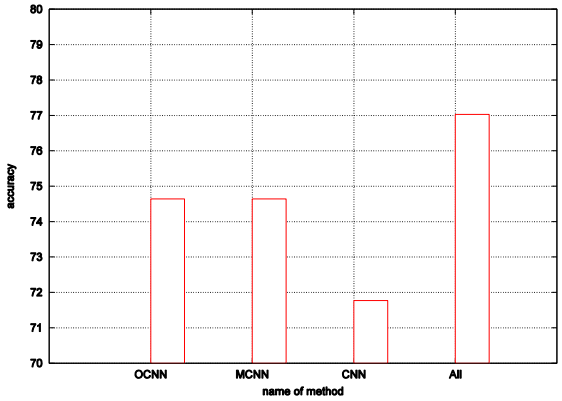

Figure 6:Accuracy on Balance

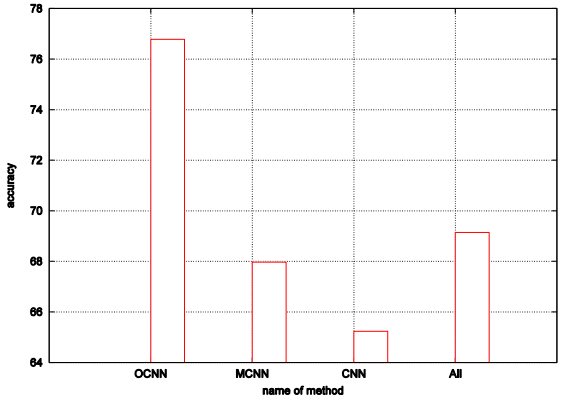

Figure 7: Accuracy over Pima

7.2 comparison with MCNN algorithm

– Both MCNN and CNN are order independent.
– The MCNN algorithm also requires multiple scans of the dataset. This is good forsmall size datasets but

for the large datasets, it requires more time. OCNN requiresonly one scan of the dataset. It computes the reduced sets within less time thanMCNN for the same training set. From table 9 and Figure 9, it is clear that approximatelyOCNN is taking$\frac{1}{3}$ of the time taken by MCNN method.

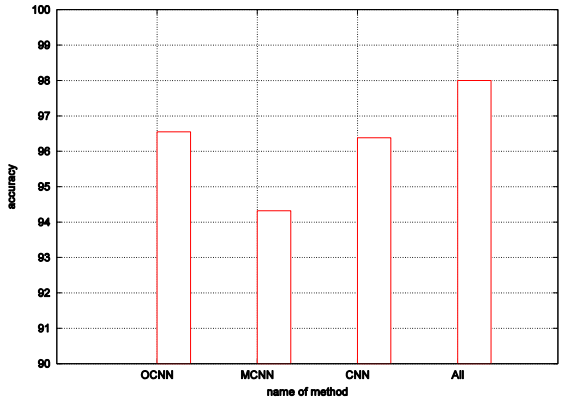

Figure 8: Accuracy on Opt.Digit.Rec

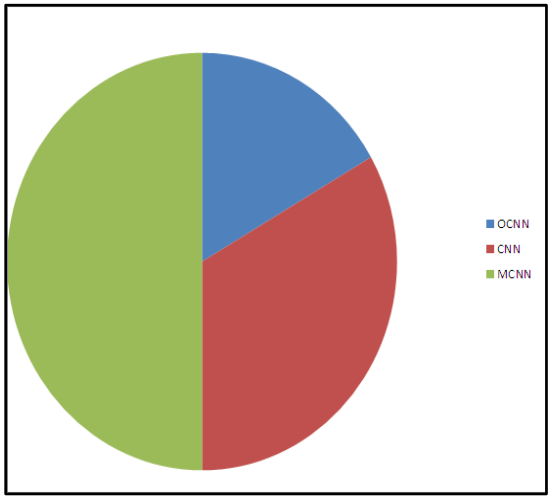

Figure 9: Time required to built Prototype Set

8. Conclusion and Future Enhancement

We described a new technique for building the prototype set for the nearest neighborclassification. This method is based on preserving the patterns near to decision boundarysurface and excludes the remaining patterns. The experiments are tested on eight standarddatasets and results obtained are good. The accuracy obtained can be comparedwith CNN and MCNN algorithms.

Further enhancement of this work is, a deletion operation might be incorporated such that bad prototypes (which are not much useful in classification process) can bedeleted. The Prototype set can be chosen such that it should have consistency property.Fuzzy set approach can be used to deal with inconsistencies and to improve theaccuracy.

9. References

1. Raja Kumar, R., Viswanath, P., Shoba Bindu, C. (2014). A New Prototype Selection Method for Nearest Neighbor Classification, Proc. of Int. Conf. on Advances in Communication, Network, and Computing, CNC, 2014 ACEEE, pp 1-8.

2. Babu, V. S., Viswanath, P. (2007, December). Weighted k-nearest leader classifier for largedata sets. In International Conference on Pattern Recognition and Machine Intelligence (pp.17-24). Springer Berlin Heidelberg.

3. Cover, T., Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on informationtheory, 13(1), 21-27.

4. Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching.Communications of the ACM, 18(9), 509-517.

5. P.E. Hart.(1967). The condensed nearest neighbor rule, IEEE Trans. Inform. Theory IT-14(3),515-516.

6. Tomek, I. (1976). Two modifications of CNN. IEEE Trans. Systems, Man and Cybernetics, 6,769-772

7. Swonger, C. W. (1972). Sample set condensation for a condensed nearest neighbor decisionrule for pattern recognition,511-519.

8. G.W. Gates(1972). The reduced nearest neighbour rule. IEEE Trans. Inform. Theory IT-18(3),431-433.

9. Sanchez J. S. Pla, F. and Ferri, F. J. (1997). Prototype selection for the nearest neighbour rulethrough proximity graphs. Pattern Recognition Letters, 18(6), 507-513.

10. Devi, V. S., Murty, M. N. (2000). Handwritten digit recognition using soft computing tools.In Soft Computing for Image Processing (pp. 507-524). Physica-Verlag HD.

11. Kuncheva, L. I. (1995). Editing for the k-nearest neighbors rule by a genetic algorithm.Pattern Recognition Letters, 16(8), 809-814.

12. Kuncheva, L. I., and Jain, L. C. (1999). Nearest neighbor classifier: simultaneous editing andfeature selection. Pattern recognition letters, 20(11), 1149-1156.

13. Devi, V. S., and Murty, M. N. (2002). An incremental prototype set building technique. Pattern Recognition, 35(2), 505-513.

14. Duda, R. O., Hart, P. E., and Stork, D. G. (2001). Pattern classification. 2nd. Edition. NewYork.

15. Kumar, R. Raja, P. Viswanath, and C. Shobha Bindu. An Approach to Reduce the ComputationalBurden of Nearest Neighbor Classifier, Procedia Computer Science, 2016

16. Murphy, P., Aha, D. W. (1995). UCI repository of machine learning databases–a machine readablerepository.

17. Hamamoto, Y., Uchimura, S., and Tomita, S. (1997). A bootstrap technique for nearest neighborclassifier design. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(1),73-79.