Intelligent Threshold Prediction in Hybrid Mesh Segmentation using Machine Learning Classifiers

Vaibhav J Hase[1],Yogesh J Bhalerao[2], Saurabh Verma[3],
V D Wakchaure[4], G JVikhe Patil[4]

[1]Research Scholar, SPPU, Department of Mechanical Engineering, AVCOE, Sangamner(India).
[2] Professor in Mechanical Engineering, MIT Academy of Engineering, Alandi, Pune (India).
[3]Center for Computational Technologies, Pune (India).
[4] SPPU, Department of Mechanical Engineering, AVCOE, Sangamner (India).

ABSTRACT

The optimal threshold is the decisive factor in CAD mesh segmentation. It is difficult for a layman to set the optimal threshold for hybrid mesh segmentation. In this research work, a generalized technique is developed to predict optimal threshold (Area Deviation Factor) for CAD mesh model which makes hybrid mesh segmentation automatic by using a nonparametric, supervised learning classifier, i.e., K Nearest Neighbor (KNN). The proposed approach classify a CAD mesh model based on mesh attributes and predict the threshold. We demonstrate and validate the algorithm's ability to predict threshold using extensive testing on test models taken from various benchmarks, and it is found to be robust and consistent. We use percentage coverage and the number of primitives as a measure to test the efficacy of the algorithm. The experimentation shows that the optimal value of the threshold for segmentation results in better coverage. The KNN classifier predicts the threshold correctly with coverage of more than 95%. The novelty of the proposed method lies in threshold prediction based on mesh quality. The predicted threshold can be linked to a downstream application like automatic feature recognition from CAD mesh model.

Keywords:CAD mesh model, Feature recognition,hybrid mesh segmentation,k-Nearest Neighbors.

1.Introduction

Standard Triangulated Language (STL) is a widely used CAD data exchange format supported by almost all different commercial CAD/CAM software including Photoshop. STL finds applications in additive manufacturing, computer graphics, tool path generation, reverse engineering, CAE applications, inspection, sheet metal forming, 3D printing,and other engineering applications [1].

CAD mesh models are obtained from CAD systems. When the CAD model is discretized into a mesh model and stored in STL format, the geometrical and topological information needed for segmentation is lost. Feature recognition from CAD mesh modelis vital in commercial CAD/CAM software [2]. This extracted feature information can be utilized for downstream applications like tool path generation, computer-aided process planning, FEA, reverse engineering, and Mesh generation [3],etc.

Feature recognition is a tool to recreate the feature in the target system after data exchange [4]. Majority of the research work has been carried out in extracting volumetric and free-form features in last two decades. However, most FR tool works on B-rep models. Innovative methods of 3D design and manufacturing are mesh based [5]. A need exists to develop FR from CAD mesh model. The STL format is supported globally by all major CAD/CAM systems which make STL a means of platform-independent CAD data exchange [10]. If we recognize features from STL model, it will be a unique data translator utility [6].

Mesh segmentation is the most favored approach for extracting surface features [7], and in the past few years, it has been studied comprehensively. The objective of mesh segmentation is partitioning the input CAD mesh modelinto meaningful and distinct regions [8]. Most of the mesh segmentation methods partition models based on mesh attributes like Curvature, Geodesic distance, Convexity, Dihedral angle,etc. Different mesh segmentation criteria and techniques have been comprehensively summarized by [8–13]. However, the results of segmentation greatly depend on the choice of mesh attributes.

For mechanical engineering applications, segmentation aims to partition CAD mesh model into regions. Each region can be fitted to a distinct, mathematically analyzable form [14]. We have proposed and implemented hybrid mesh segmentation, to partition the mesh model and to extract complex features like blends.

All commercial CAD/CAM systems triangulate the CAD model by the surface. Triangles which lie on the same surface have the same quality. We use the triangle property “Facet Area” to segment the model. A significant step in hybrid mesh segmentation is to set the appropriate Area Deviation Factor (threshold) at the beginning. Identifying the correct threshold value for mesh segmentation algorithms mostly involves a manual trial and error-based approach which is laborious [15]. Also, choosing an inadequate threshold can lead to either over-segmentation or under segmentation. This makes threshold setting too complicated for a layman. Inadequate threshold leads to over-segmentation (multiple small patches) or under segmentation. Oversegmentation needs a post-processing merging step, and it increases processing time [16] whereas under segmentation leads to deficient results [17]. However, for a layman, setting the appropriate threshold is too complicated. Manual prediction is laborious and error-prone. Therefore, an intelligent threshold prediction is of great importance.

As stated above, Area Deviation Factor (ADF) is the decisive factor in segmentation quality. For a test case as shown in Fig. 1, we measure the sensitivity of the ADF varying from 0.30 to 0.75 on segmentation quality. It would be difficult for a layman to set the appropriate threshold.

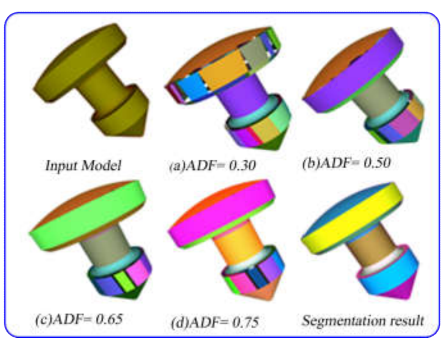


Fig. 1. Sensitivity of the ADF. (a) input model (b) ADF = 0.30 (c) ADF = 0.50 (d) ADF = 0.65 (e) ADF = 0.75

CAD mesh is sparse, nonuniform and nonstreamlined triangulation. We have proposed and implemented hybrid mesh segmentation successfully to partition CAD mesh modelusing "Facet Area." The objective of this research work is to develop a generalized technique to predict threshold correctly for CAD mesh modelwhich makes a hybrid mesh segmentation process automatic by using the k-Nearest Neighbors (KNN) classifier.

The KNN is one of the most simple and highly efficient, nonparametric, supervised learning classifiers in pattern recognition. The KNN classifier needs training data and a predefined value of K based on distance computation. It finds applications in machine learning, pattern recognition, data mining, text categorization, object recognition, etc. [18].

1.1 Contributions

We address two main research questions:

- What is the best way to predict ADF automatically in hybrid mesh segmentation for CAD mesh model using Machine Learning (ML)?
- Which distance metric will be least affected by noise when used with the KNN classifier for a special dataset which contains attributes of CAD mesh model?

The following are our significant contributions:

- Successfully applied KNN classifier for predicting threshold using KNN, based on mesh quality.
- Intelligent threshold prediction makes hybrid mesh segmentation automatic which results in automatic feature recognition from a CAD mesh model.

1.2 Outline

The outline of the paper is as follows. Section 2 discusses prior work related to KNN classification approaches; Section 3 provides a brief background on KNN and the computation of each distance metric; Section 4 gives details of the framework of the proposed empirical study. The experimental results and implementation details are presented in Section 5 and Section 6; Discussion is provided in Section 7; Section 8 depicts the application of intelligent threshold prediction;Finally, the conclusions and future work are proposed in Section 9.

2.Literature Review

In the last few decades, a flurry of research activity has been carried out on KNN classifier. In this section, we review the-state-of-the-art of KNN classifier.We limit our review of those approaches in which KNN is used for prediction.

In the medical field, KNN finds applications in the prediction of various diseases. Amutha et al. [19] presented a modified KNN based algorithm for the prediction of "Tread mill"test in cardiology using a clinical dataset with *Hamming*distance metric. Mordvanyuk et al. [20] presented KNN based prediction of glucose level with a *Euclidean*distance metric. Aydın etal. [21] used KNN to detect breast cancer on Bow-Tie antenna dataset with the *Euclidean* distance metric with an accuracy of 90%. Kim et al.[22]confirmed the applicability of the KNN method for predicting the influent characteristics of wastewater treatment plant. Zhongguo et al. [23] proposed a method for choosing classification algorithms and predicted the optimum parameter for a specified algorithm based on knowledge learning from historical data sets. Prasath et al.[24] had evaluated KNN classifier on 28 datasets from the UCI machine learning repository and identified top 10 distance metrics among 54 distance metrics. Interested readers can consult a detailed and exhaustive literature survey[18,24–30]on KNN.

Bulut and Amasyali [31] presented a supervised method to find local optimum K and evaluated it on 36 UCI data sets and achieved a better accuracy. However, their method has more time complexity. Zhongguo et al. [32] used a data set's intrinsic characteristics to find the optimal value of K for the KNN. Prasath et al. [24] had evaluated the KNN classifier on 28 datasets from the UCI machine learning repository and identified top 10 distance metrics among 54 distance metrics. TABLE 1 provides a comparison between previous studies for different distance metric.

Table 1. Comparison between previous studies.

Sr. No.	Researchers	Technique	No. of Distances	Distances	Findings	Benchmark Dataset
1	Prasath et al. [24]	KNN	54	28	Identified top 10 distance metric	UCI
2	Nurul Ezzati et al. [33]	MIC	5	1	Minkowski	BCI Competition IV
3	Chomboon et al. [34]	KNN	11	8	Euclidean, Manhattan, Chebychev,etc	Synthetic datasets
4	Mulak and Talhar [35]	KNN	3	1	Manhattan Accuracy:97.8%	KDD
5	Hu et al. [36]	KNN	4	37	Chi-square	UCI
6	Todeschini et al.[37]	KNN	18	8	Manhattan, Euclidean etc.	UCI
7	Lopes and Ribeiro [38]	1-NN	5	15	Euclidean Manhattan	UCI
8	Hassanat et al.[39]	KNN	3	28	Hassanat	UCI
9	Lindi [40]	KNN	3	2	Hassanat	AT &T face database
10	Alkasassbeh et al. [41]	KNN	54	28	Hassanat	UCI

Hassanat et al. [39] proved that the rule of thumb, i.e., K as the square root of the size of the training dataset, is not an excellent choice for KNN. Also, a large value of K does not help to increase the accuracy of KNN. They suggest using the odd value of K to increase speed. Alkasassbeh et al.[41] attempt to enhance the performance of KNN using Hassanat distance metric. The reported result illustrates that *Hassanat* distance metric is invariant to data scale and noise and is superior to *Euclidean* and *Manhattan* distances. Lindi [40] investigated the performance of KNN classifier for face recognition system proposed for the NAO robot using three distance metrics. Nurul Ezzati et al.[33] argue that the performance of KNN depends on K-value and distance metrics by performing Motor Imagery Classification based on Electroencephalogram Signal using various distance metrics. The reported result indicates that *Minkowski* distance was best with 70.08% accuracy

Variants of KNN. Van et al. [42]propose weighted KNN for indoor VLC positioning. Singh et al. [43] evaluated the impact of text and numeric data type on classifier performance. They evaluated KNN, Naïve Bayes, and Random Forest. They concluded that KNN is best for a quick prediction. Yu et al.[44] proposed a hybrid k-Nearest Neighbor classifier to tackle the special dataset (having noisy attributes) from KEEL dataset repository. Tang et al.[45] have proposed multiple points weighted KNN classifier and depicted better performance over

Support Vector Machines and Bayesian classifier for remote sensing. Nour and Qasem [46]presented the weighted KNN and fuzzy KNN to classify medical datasets. The reported result illustrates that fuzzy-KNN has the best accuracy. Nair et al.[47]propose a hybrid classification model to enhance the performance of the standard KNN (k = 1) classification using stacking approach. They hybridized three classifiers: 1NN, rotation forest, and simple logistic regression and tested on 13 datasets from the UCI repository.

2.1 Literature Findings

To the best of our knowledge, no work has been carried out in the past to predict threshold by KNN for CAD mesh model. In previous works, researchers tended to conquer the 3D model classification problem by exploiting machine learning techniques. However, all these techniques constrained to 3D CAD models [48]. Most of the algorithm used datasets drawn from the UCI machine learning repository [49]. UCI data sets are not suitable for threshold prediction in the hybrid segmentation of CAD mesh models.Till date, there has been no accepted benchmark for 3D CAD models [48]. A need exist to build 3D CAD model database as the training datasets for the research proposed.

The CAD mesh model has multi dimensions and unknown distribution data along with noise. To the best of our knowledge, no previous work addressed in threshold prediction based on mesh quality. The goal of this research work is to predict the threshold based on mesh quality.

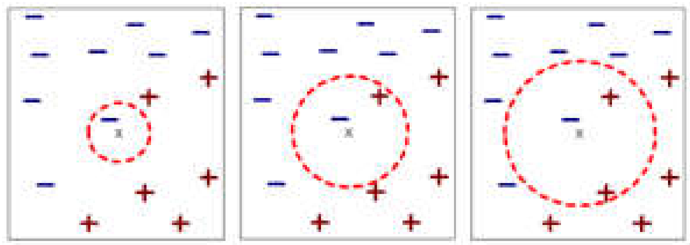
3. Background

This sectionprovides a brief backgroundof hybrid mesh segmentation along with the KNN classifier and distance metrics that will be employed throughout this paper.

3.1 K-Nearest Neighbors (KNN)

KNN algorithm is a classical and well-established method in machine learning [50]. KNN is Lazy learning, nonparametric pattern classification algorithm. It is best suitable for the classification of a data set in which no prior knowledge of the distribution of the dataset is available. The KNN predicts the threshold for unlabelled test case using the threshold of closest neighbors in the training dataset. The closeness depends on the distance metric used.

A KNN algorithm predicts threshold of unlabelled test case “X" base on majority voting. The input for KNN is Feature vector (training dataset) which must be assigned label. In our case, the threshold is assigned to each training dataset. When an unlabelled test case (of which threshold is to be predicted) is given, the KNN compares it with the training dataset.Fig. 2 shows the 1, 2 and 3- nearest neighbors of data point which is placed at the center of the circle. In Fig. 2(a), nearest neighbor (NN) of the unlabelled test case is negative, so the negative class label is assigned to the unlabelled test case. If there is a tie between the two classes, then the random class is chosen for the unlabelled test case. Fig. 2(c), three nearest neighbors are present, one is negative, and the other two is positive. So in this case, majority voting is used to assign a class label to the unlabelled test case.



(a) 1-NN (b) 2-NN (c) 3-NN

Fig. 2. KNN of a record “X” with varying NN [51]

3.2 Distance Metric

The distance metric is vital to finding the similarity and dissimilarity between data points.The performance of KNN greatly depends on the choice of distance metric [52]. In the past decades, significant efforts have been made to find the best distance metric which can handle invariance in data scale and noise.

Hassanat presented a new dimensionality invariant distance metric which is invariant to data scale, noise, and outliers [53] .Given two objects “X” (Feature Training Vector) and “Y" (FeatureTesting Vector). Distance is calculated from these two vectors variable values. The higher the difference, the more different are the two objects. In this paper, following similarity measures are used in KNN. TABLE 2 shows a list of 11 similarity measures.

Table 2. List of 11 similarity measures

ID	Name	Acronym	Distances
1	Manhattan	*MD*	$MD(X_i,Y_i) = \sum_{i=1}^{n} \lvert (X_i - Y_i) \rvert$
2	Euclidian Distance	*ED*	$ED\ (X,Y) = \sqrt{\sum_{i=1}^{n} \lvert (X_i - Y_i) \rvert^2}$
3	Hassanat Distance	*HasD*	$HasD(X,Y) = 1 - \frac{1+min(X_i,Y_i)}{1+max(X_i,Y_i)},\ min(X_i,Y_i) \geq 0$
4	Wave- Hedges distance	*DWH*	$DWH(X,Y) = \sum_{i=1}^{n} \left(1 - \frac{min(X_i,Y_i)}{max(X_i,Y_i)}\right)$
5	Lorentzian distance	*LD*	$LD\ (X,Y) = \sum_{i=1}^{n} ln\ (1 + (X_i - Y_i) \rvert)$
6	Canberra distance	*CanD*	$CanD\ (X,Y) = \sum_{i=1}^{n} \left(\frac{\lvert (X_i - Y_i) \rvert}{\lvert X_i \rvert + \lvert Y_i \rvert}\right)$
7	Squared Chi-Squared	*SCSD*	$SCSD(X,Y) = \sum_{i=1}^{n} \left(\frac{(X_i - Y_i)^2}{\lvert X_i + Y_i \rvert}\right)$
8	Clark distance	*ClaD*	$ClaD\ (X,Y) = \sqrt{\sum_{i=1}^{n} \left(\frac{\lvert X_i - Y_i \rvert}{X_i + Y_i}\right)^2}$
9	Divergence distance	*DivD*	$DivD\ (X,Y) = 2 \sum_{i=1}^{n} \left(\frac{(X_i - Y_i)^2}{(X_i + Y_i)^2}\right)$
10	Average distance	*AD*	$AD\ (X,Y) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (X_i - Y_i)^2}$

11	Whittaker's index of association Distance	*WAID*	WIAD(X,Y)= $\frac{1}{2}\sum_{i=1}^{n} \left| \frac{X_i}{\sum_{i=1}^{n} X_i} - \frac{Y_i}{\sum_{i=1}^{n} Y_i} \right|$

4. Methodology

In this section, we present an overview of the framework of threshold prediction using a KNN algorithm with various distance metrics and describe the functionality of each component. The framework is shown in Fig. 3

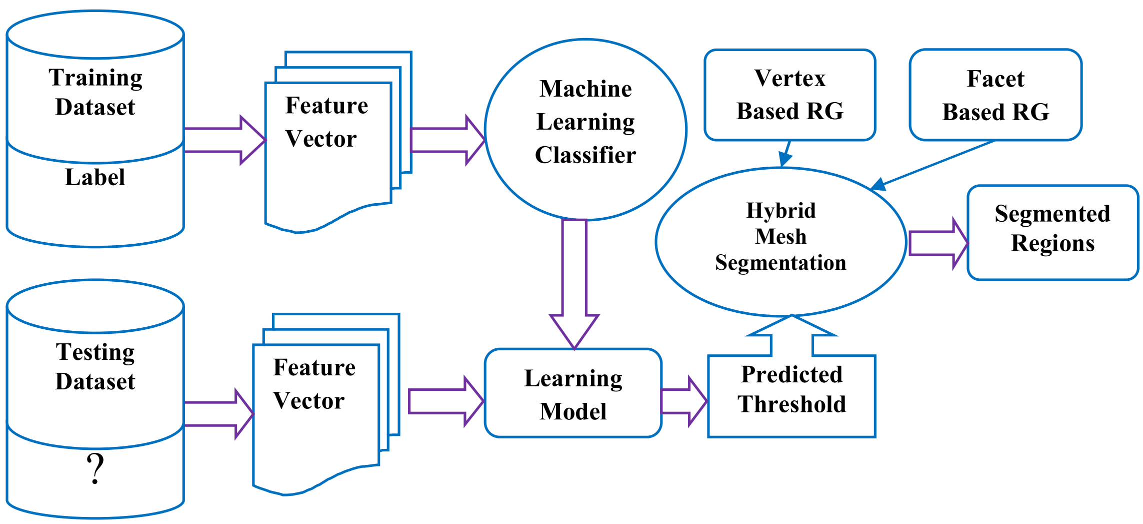


Fig. 3. The framework for threshold prediction

- Step 1: We have created a new dataset of CAD mesh model with different distribution. Fig.4 shows The Dataset (threshold) distribution of entire Dataset.

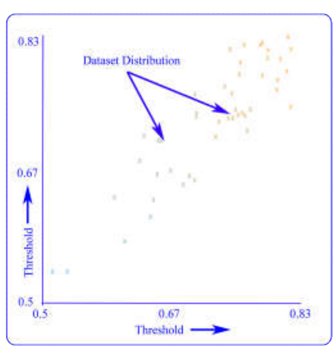


Fig. 4. The dataset (threshold) distribution of the entire dataset

For each CAD model, compute per face quality according to "Facet Area."The computed *Standard deviation* (StdDev), *Average* (Avg) and *Median* (Med) used as a "Feature Vector." The "Feature Vector" provides input to the machine learning algorithm. One of the most important attributes of a"Feature Vector" is to assign a class label (threshold) to it. Choosing the correct threshold value for a test case is a difficult task. There is no universal way of finding a correct threshold. A trial and error approach used a trial and error approach [15] to identify a threshold value for each CAD model. A labelis set for a training dataset of CAD mesh model, based on the accuracy of feature extracted.TABLE 3 and TABLE 4 depict mesh attribute description of the training and testing dataset respectively.

Table 3: Description of the training dataset used

Attribute Name	Type	Distinct	Unique	Statistic			
				Min	Max	Mean	StdDev
StdDev	Num	47	47(100%)	0	13796.494	418.54	2000.26
Avg	Num	47	47(100%)	0	2111.854	66.161	306.695
Med	Num	46	47(100%)	0	574.12	19.59	84.404
Class	Num	19	12(26%)	0.5	0.83	0.728	0.078

Table 4: Description of the testing dataset used

Attribute Name	Type	Distinct	Unique	Statistic			
				Min	Max	Mean	StdDev
StdDev	Num	20	20(100%)	0	959.58	116.236	220.249
Avg	Num	20	20(100%)	0	422.254	35.615	92.339
Med	Num	19	18(90%)	0	278.192	17.793	61.471
Class	Num	8	5(25%)	0.6	0.822	0.757	0.058

- Step 2: Use the dataset created in step 1 for data classification by applying the KNN algorithm with various distance metrics to compute the k-nearest data points for making a classification.
- Step 3: For a query model of which threshold to predict is classified by KNN, based on mesh attributes of a training data. KNN assigns class labels of nearest neighbors to predict the class label of the unknown test case.
- Step 4: Analyze the results and conclude about the performance of classification and prediction using various distance metrics.

4.1. Hybrid mesh segmentation

The goal of hybrid mesh segmentation is to partition the CAD mesh model into basic primitives like a plane, sphere, cylinder, and cone. It is difficult to segment CAD mesh modelby using facet based region growing or vertex based region growing alone. Vertex-based region growing technique is used to detect curved surface whereas Facet-based growing technique is used to detect curved features and planes. None of these techniques on their own gives a robust solution to recognize feature from CAD mesh model.

Hybrid mesh segmentation uses region growing algorithms to clusters facets into groups. The approach is hybrid as we use the “Facet Area” property to group facets together, using a combination of vertex-based and facet-based region growing algorithms [54]. A promising approach that has become evident is a hybrid (facet and vertex based) one wherein the advantages of the above approaches are combined A detailed description of hybrid mesh segmentation is beyond the scope of this paper.

A major step in hybrid mesh segmentation is to set the appropriate Area Deviation Factor (threshold) at the beginning. Inadequate threshold leads to over-segmentation or undersegmentation. We predict this thresholdusinga KNN classifier.

5. Experimental setup

The dataset is split into two sets (30:70). The first set (30% of the dataset) is used for testing. The second set (70% of the dataset)is used for training [41]. The value of K is set to one. The overall experimental framework is as detailed in Section 4.We used 11 classifiers to classify the test samples and predict the class.

In the training phase, thresholds are set for each model D by trial and error approach [23]. In the testing phase, the labelis assigned to d of D which has rank one based on a distance measure.To make the algorithm generalize to unseen model, none of a single query model from training dataset used in testing dataset [55].

5.1. Performance evaluation measures

In this research work, we use the accuracy as a performance evaluating measure for KNN predictor.

$$\text{Accuracy} = \frac{\text{Number of correct predictions for unseen data}}{\text{Total number of test samples}} \quad (1)$$

We use percentage coverage as a performance evaluating measure for hybrid mesh segmentation.It is a ratio of a number of features recognized to the number of features present in a CAD mesh model.

5.2. Experimental results and discussion

5.2.1 Evaluate best Distance Measure

To evaluate quantitatively, the best distance metric, the experiment has been carried out on 20 models of the testing dataset. For every model, the threshold is predicted using all 11 predefined distance with K=1 and 3. The accuracy of each distance on 20 models of a testing dataset has been evaluated. The accuracy of 11 distances on a testing dataset is summarized in Fig. 5

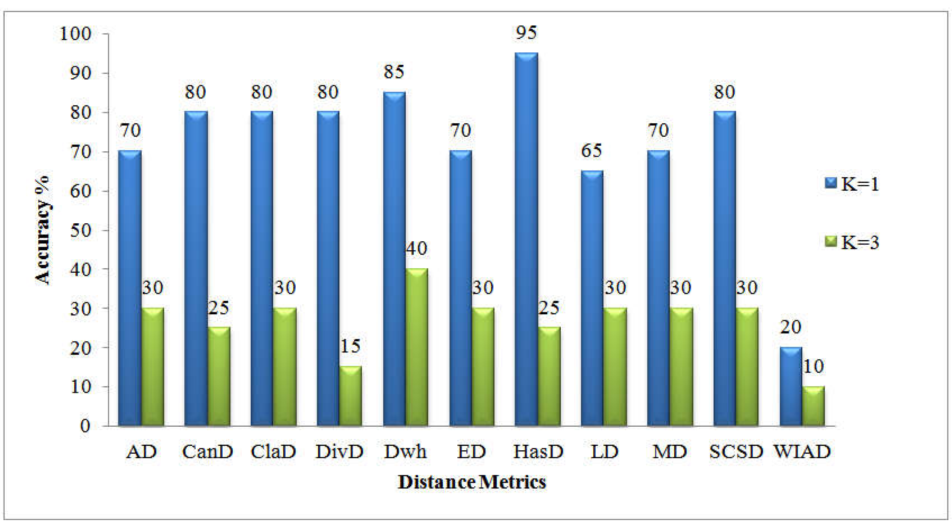


Fig. 5. Accuracy (in %) for testing data set as a function of K value

The following are the observations of the experimentation:

- *HasD* outperforms all other tested distances followed by *Dwh*. *WAID* recorded the lowest accuracies.
- In Squared L_2 distance measure family, *ClaD, SCSD,* and *DivD* achieve similar performance with accuracy of 80%.
- *CanD* (L_1 distance measure family) is better than *MD*.
- Among L_p distance measure family, *ED* and *MD* have similar performance (70% accuracy).

5.2.2 The optimal value for K

The performance of KNN significantly depends on the value of K. As noted by Singh et al. [49], even the value of K is not advisable which results in a tie. The odd values of K result in greater accuracy over even values. To authenticate this, experimentation has been carried out on datasets with the odd values k = 1, 3, 5 and 7. We analyze the performance of KNN by measuring the error rate for each odd values k = 1, 3, 5 and 7. The K value which gives the least misclassification error is selected to predict. We get the least misclassification error when k = 1.

5.2.3 Predicting Threshold

To predict threshold for a query model q, First, we measure per face quality according to "Facet Area,"i.e.,*Standard deviation*(StdDev), *Average* (Avg), *Median* (Med). Measure *HasD* distance vector q_i between q and every model q_i in the training dataset. The distances are then ranked in ascending order of their response of distance D. With K=1, assign the class to q of q_i which has rank 1 (minimum Distance), which is a predicted threshold for q.

6. Implementation and Testing (Results)

The proposed algorithm has been implemented and tested using VC++ running on a computer with Intel Core i3 processor, 8GB RAM, 64-bit windows 8.1 operating system. The developed system can accept any STL file generated by CAD system like Solidworks™, Autodesk™ Inventor™ 2018 and Onshape™.A plugin has been developedfor Autodesk™ Inventor™ 2018.The user interface (UI) MeshFR is as shown in Fig.6.

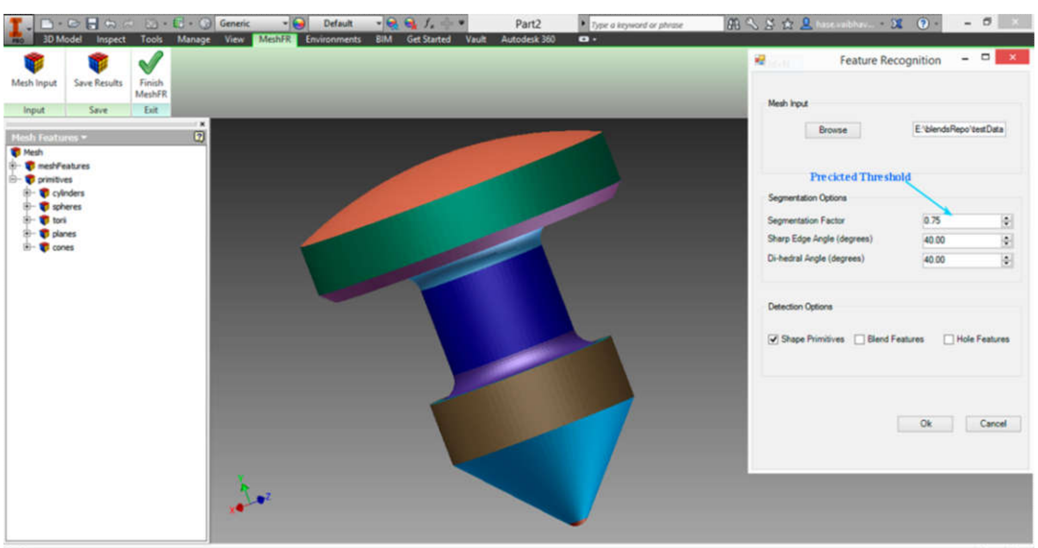


Fig. 6. MeshFR Add-in along with automatic threshold prediction

6.1. Case study

Most of the existing algorithms are error prone to the threshold. The threshold depends on mesh quality and mesh density. Hence each model requiresa different threshold. The proposed approach automatically and intelligently predicts a threshold for each model based on facets quality

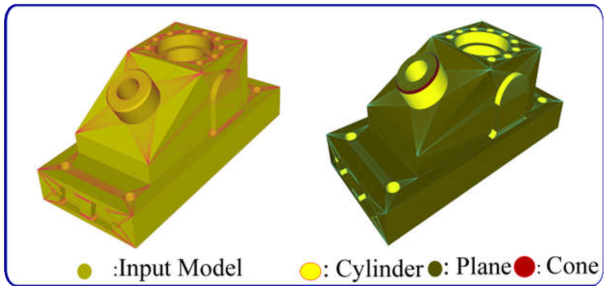


Fig. 7. "Anchor" model

Fig. 7illustrates a typical "Anchor" model taken from NIST repository [56]. The part is imported in AutodeskTMInventorTM 2018 and exported as a CAD mesh model. The model has 7100 facets and 3542 vertices. The test case is used to verify the efficacy of our algorithm.

The algorithm starts with an input CAD mesh model. For "Anchor"model, compute per face quality according to the area. The computed Standard deviation(StdDev), Average (Avg) and Median (Med) are used as an input "Feature Vector." The input Mesh attributes are as shown in TABLE 5. KNN predict threshold based on input "Feature Vector."

Table 5. The input Mesh attributes for "Anchor" model

StdDev	Avg	Med
394.879028	59.690456	17.341423

KNN predict threshold based on input attributes. To evaluate the best distance metric quantitatively, the experiment has been carried out on "Anchor" model. For a model, we predict the threshold using all 11 predefined distances.

The comparison of prediction of 11 distances on testing "Anchor" model is summarized in Fig. 8. To evaluate best distance measure, experimentation has been carried out with varying values of K from 1 to 7. It can be seen that K = 1, *HasD* gives excellent results with an accuracy of 95%.We get a minimum error in prediction when k = 1. Out of 11 distance metrics,10 achieved similar performance for K =1. WAID recorded the worst prediction. The experimentation shows that best results are obtained for K = 1 with threshold = 0.75. The results show that the KNN classifier predicts the threshold correctly.

We use percentage coverage, the number of primitives as an indicator of the successful segmentation algorithm. The numbers of actual features in "Anchor" model are 81, and extracted features are 81. Total coverage is 100.

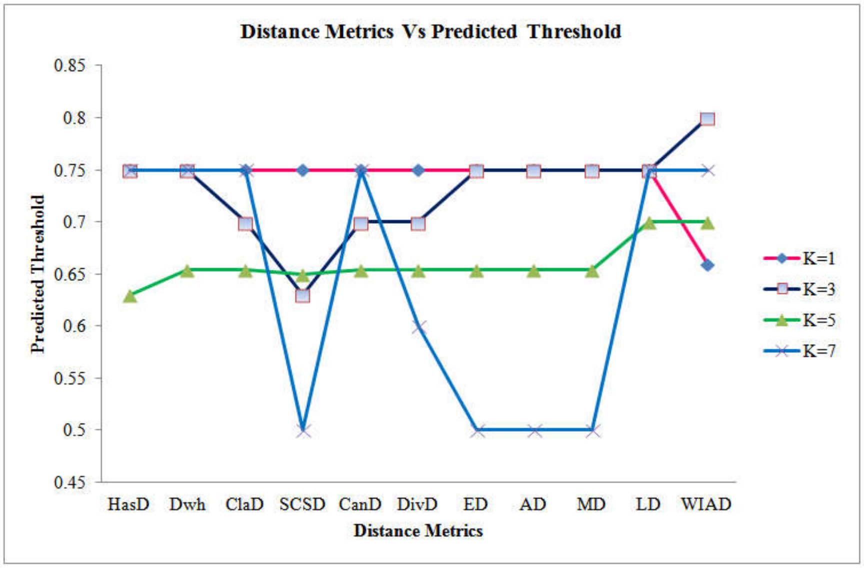


Fig. 8. illustrates a comparison of prediction of 11 distance metrics with K = 1,3,5,7 for "Anchor" model

7. Discussions

In this paper, we have used mesh attributes to predict threshold which rarely explored by the research community. As noted by Muraleedharan et al. [23], it is a difficult task to identify a threshold value for getting the expected results. Most of the time, a trial and error based approach is used to identify a correct threshold. However, in this work, an attempt has been made to set threshold intelligently. The predicted threshold can be used as an input for automatic feature recognition from CAD mesh model.

7.1 Limitations

Although the proposed method is useful in threshold prediction for CAD mesh model, it has some limitations. Firstly, the accuracy of prediction depends on the size of data set used for training KNN. Currently, the dataset used is limited. Therefore, in future work, we increase the size of the dataset with diverse classes.

8. Application of threshold prediction

Fig. 9 shows a few test cases extracted from various benchmarks demonstrating the algorithms ability to predict threshold which subsequently uses for segmenting CAD mesh model.

The efficacy of the algorithm has been tested on realistic CAD mesh model taken from NIST repository and AIM Shape Repository [57]. We use percentage coverage and the number of primitives as a measure to test the efficacy of the algorithm. TABLE 6shows the benchmark for convergence.

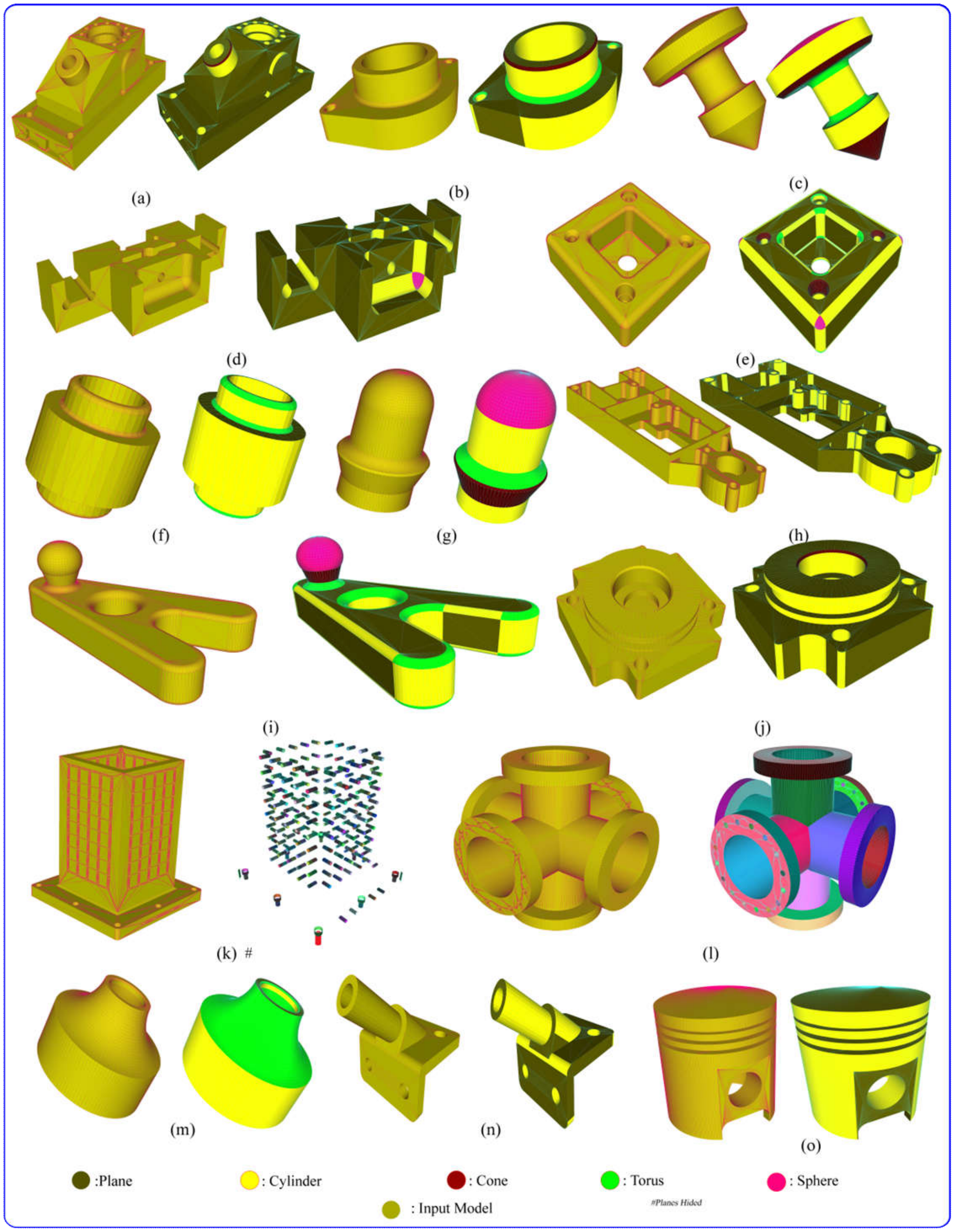


Fig. 9. Application of threshold prediction in automatic FR from CAD mesh model

The results show that the KNN classifier predicts the threshold correctly. We use percentage coverage, the number of primitives as an indicator of the successful segmentation algorithm. The numbers of actual primitives in "Anchor" model are 81, and extracted futures are 81. Total coverage is 100.

Table 6. Benchmark for convergence

Fig	a.Anchor	b.Assy	c.TestCase1	d.Bracket	e.TestCase2	f.Stator	g.Test Case3	h.Test Case4	i.TestCase5	j.C_rear	k.Tooling Block	l.Cross	m.Impeller	n.Camy	o.Piston
F	7100	8022	6116	1732	12068	2592	5244	5148	9786	2674	38932	17104	11644	944	16792
V	3542	4007	3060	862	6034	1296	2624	2566	4893	1339	19092	8480	5822	4640	8394
S	1.79	2.03	1.55	0.4414	2.23	0.65	1.34	1.4	2.58	0.6836	9.84	2.86	2.87	0.2422	4.05
A_{df}	0.75	0.8	0.75	0.8	0.75	0.75	0.8	0.7	0.7	0,80	0.7	0.8	0.7	0.76	0.8261
NP	104	19	11	50	69	12	7	146	32	45	630	62	9	24	50
T	0.954	0.658	0.569	0.262	1.078	0.323	0.567	0.671	1.966	0.419	4.257	1.545	1.164	0.141	1.716
C	99.87	100	100	99.7691	100	100	100	97.32	100	100	99.5839	100	100	99.153	99.9881

F: Number of Facets V: Number of Vertex S:STL Size(in MB) A_{df}:Predicted Area deviation factor

NP: Number of primitives T:Overall Timing(in seconds) C:% Coverage

9. Conclusions

In this paper, an elegant method has been proposed and implemented for predicting ADF (Threshold). Proposed method guides a layman to set the appropriate threshold intelligently.

With the help KNN algorithm, we have intelligently predicted threshold which makes hybrid mesh segmentation automatic. The KNN method has been evaluated for 11 distance measures for predicting threshold in hybrid mesh segmentation for CAD mesh model based on the performance measure. It is observed that *HasD* outperforms in all other tested distances followed by *Dwh* which can predict threshold for the course and dense CAD meshes. *WAID* recorded the lowest accuracies. By experimentation, we attempt to calculate optimal K value for CAD mesh modelin hybrid mesh segmentation. We have found that *HasD* distance performed the best when applied to the CAD domain dataset.

The proposed algorithm has been tested with CAD models taken from NIST repository [67] and exported to STL using Solidworks™, Autodesk™ Inventor™ 2018 and Onshape™.With varying facets and vertices found to be consistent in predicting ADF (threshold) automatically. The developed system has been extensively tested for Feature recognition applications.

Future work involves improving the accuracy of threshold predicting using fuzzy-KNN or Genetic Algorithm (GA) or deep neural networks. We can improve robustness by increasing the size of the dataset.

Acknowledgments

This work has been supported by *Center for Computational Technologies (CCTech), Pune, India*. The authors are also grateful to Dr.P. J. Pawar for their constructive, thoughtful suggestions that helped to improve the manuscript.

References

[1] J. Zhang and Y. Li, *Region segmentation and shape characterisation for tessellated CAD models, International Journal of Computer Integrated Manufacturing, 29(8)*, 2016, 907–915.

[2] D. Xiao, H. Lin, C. Xian, and S. Gao,*CAD mesh model segmentation by clustering, Computers & Graphics, 35(3)*, 2011, 685–691.

[3] X. Cui, S. Gao, and G. Zhou, *An Efficient Algorithm for Recognizing and Suppressing Blend Features,*

Computer-Aided Design and Applications, 1(1–4), 2004, 421–428.

[4] M.Fu, S. Ong, W.Lu, I. B. Lee, and A. Y. Nee, *An approach to identify design and manufacturing features from a data exchanged part model, Computer-Aided Design, 35(11)*, 2003, 979–993.

[5] J. Corney, C. Hayes, V. Sundararajan, and P. Wright, *The CAD/CAM Interface: A 25-Year Retrospective, Journal of Computing and Information Science in Engineering, 5(3)*, 2005, 188–197.

[6] V. B. Sunil and S. S. Pande, *Automatic recognition of features from freeform surface CAD models, Computer-Aided Design, 40(4)*, 2008, 502–517.

[7] J. Wang and Z. Yu, *Surface feature based mesh segmentation, Computers & Graphics, 35(3)*, 2011, 661–667.

[8] D.-M. Yan, W. Wang, Y. Liu, and Z. Yang, *Variational mesh segmentation via quadric surface fitting, Computer-Aided Design, 44(11)*, 2012, 1072–1082.

[9] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal, *Mesh Segmentation - A Comparative Study, in IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06),* 2006, 7–7.

[10] A. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis, *3D Mesh Segmentation Methodologies for CAD applications, Computer-Aided Design and Applications, 4(6)*, 2007, 827–841.

[11] A. Shamir, *A survey on Mesh Segmentation Techniques, Computer Graphics Forum, 27(6)*, 2008, 1539–1556.

[12] P. Theologou, I. Pratikakis, and T. Theoharis, *A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation, Computer Vision and Image Understanding, 135*, 2015, 49–82.

[13] T. Le and Y. Duan, *A primitive-based 3D segmentation algorithm for mechanical CAD models, Computer Aided Geometric Design, 52–53*, 2017, 231–246.

[14] S. Xú, N. Anwer, C. Mehdi-Souzani, R. Harik, and L. Qiao, *STEP-NC based reverse engineering of in-process model of NC simulation, The International Journal of Advanced Manufacturing Technology, 86(9–12)*, 2016, 3267–3288.

[15] L. P. Muraleedharan, S. S. Kannan, A. Karve, and R. Muthuganapathy, *Random cutting plane approach for identifying volumetric features in a CAD mesh model, Computers & Graphics, 70*, 2018, 51–61.

[16] B. Yi, Z. Liu, J. Tan, F. Cheng, G. Duan, and L. Liu, *Shape recognition of CAD models via iterative slippage analysis, Computer-Aided Design, 55*, 2014, 13–25.

[17] J. S. Chen and H. Feng, *Automatic prismatic feature segmentation of scanning-derived meshes utilising mean curvature histograms, Virtual and Physical Prototyping, 9(1)*, 2013, 45–61.

[18] N. Bhatia and Vandana, *Survey of Nearest Neighbor Techniques, International Journal of Computer Science and Information Security, 8(2)*, 2010, 302–305.

[19] A. Amutha, R. Padmajavalli, and D. Prabhakar, *A Novel Approach for the Prediction of Treadmill Test in Cardiology using Data Mining Algorithms implemented as a Mobile Application, Indian Heart Journal, 70(4),* 2018,511-518.

[20] N. Mordvanyuk, F. Torrent-Fontbona, and B. López*, Prediction of glucose level conditions from sequentialdata, Proceedings of the 20th International Conference of the Catalan Association for Artificial Intelligence, Deltebre, Terres de L'Ebre, Spain,300,*2017,227.

[21] E. A. Aydın and M. Kaya Keleş, *Breast cancer detection using K-nearest neighbors data mining method obtained from the bow-tie antenna dataset, International Journal of RF and Microwave Computer-Aided Engineering, 27(6)*,2017, e21098.

[22] M. Kim, Y. Kim, H. Kim, W. Piao, and C. Kim, *Evaluation of the k-nearest neighbor method for forecasting the influent characteristics of wastewater treatment plant, Frontiers of Environmental Science & Engineering, 10(2)*, 2016, 299–310.

[23] Y. Zhongguo, L. Hongqi, S. Ali, and A.Yile, *Choosing Classification Algorithms and Its Optimum Parameters based on Data Set Characteristics, Journal of Computers, 28(5)*, 2017, 26–38.

[24] V. B. S. Prasath, H. A. A. Alfeilat, O. Lasassmeh, and A. B. A. Hassanat, *Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier - A Review*, 2017, 1–50.

[25] V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection, ACM Computing Surveys, 41(3)*, 2009, 1–58.

[26] H. Brighton and C. Mellish, *Advances in Instance Selection for Instance-Based Learning Algorithms, Data Mining and Knowledge Discovery, 6(2)*, 2002, 153–172.

[27] M. Sujatha, S. Prabhakar, and G. Devi, *A Survey of Classification Techniques in Data Mining, International Journal of Innovations in Engineering and Technology, 2(4)*, 2013,86–92.

[28] M. RezaAbbasifard, B. Ghahremani, and H. Naderi, *A Survey on Nearest Neighbor Search Methods, International Journal of Computer Applications, 95(25)*, 2014, 39–52.

[29] L. Jiang, Z. Cai, D. Wang, and S. Jiang, *Survey of Improving K-Nearest-Neighbor for Classification, in Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 1,* 2007, 679–683.

[30] A. Kataria and M. D. Singh, *A Review of Data Classification Using K-Nearest Neighbour Algorithm, International Journal of Emerging Technology and Advanced Engineering, 3(6)*, 354–360, 2013.

[31] F. Bulut and M. F. Amasyali, *Locally adaptive k parameter selection for nearest neighbor classifier: one nearest cluster, Pattern Analysis and Applications, 20(2)*, 2017, 415–425.

[32] Y. Zhongguo, L. Hongqi, Z. Liping, L. Qiang, and S. Ali, *A case based method to predict optimal k value for k-NN algorithm, Journal of Intelligent & Fuzzy Systems, 33(1)*, 2017, 55–65.

[33] N. E. Md Isa, A. Amir, M. Z. Ilyas, and M. S. Razalli, *The Performance Analysis of K-Nearest Neighbors (K-NN) Algorithm for Motor Imagery Classification Based on EEG Signal, MATEC Web of Conferences, 140,*2017, 01024.

[34] K. Chomboon, P. Chujai, P. Teerarassammee, K. Kerdprasop, and N. Kerdprasop, *An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm, in The Proceedings of the 2nd International Conference on Industrial Application Engineering 2015,*2015, 280–285.

[35] P. Mulak and N. Talhar, *Analysis of Distance Measures Using K-Nearest Neighbor Algorithm on KDD Dataset, International Journal of Science and Research, 4(7)*, 2015, 2101–2104.

[36] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai, *The distance function effect on k-nearest neighbor classification for medical datasets, SpringerPlus, 5(1)*, 2016, 1304.

[37] R. Todeschini, D. Ballabio, V. Consonni, and F. Grisoni, *A new concept of higher-order similarity and the role of distance/similarity measures in local classification methods,Chemometrics and Intelligent Laboratory Systems, 157*, 2016, 50–57.

[38] N. Lopes and B. Ribeiro, *On the Impact of Distance Metrics in Instance-Based Learning Algorithms* (Eds. Cham: Springer International Publishing, 2015).

[39] A. Hassanat and Ahmad Ali Alhasanat, *Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach, International Journal of Computer Science and Information Security, 12(8)*, 2014, 33–39.

[40] G. A. Lindi, *Development of face recognition system for use on the NAO robot,* Stavanger University, Norway, 2016.

[41] M. Alkasassbeh, G. A. Altarawneh, and A. B. Hassanat, *on Enhancing the Performance of Nearest Neighbour,Canadian Journal of Pure and Applied Sciences, 9(1)*, 2015.

[42] M. T. Van, N. Van Tuan, T. T. Son, H. Le-Minh, and A. Burton, *Weighted k-nearest neighbour model for indoor VLC positioning, The Institution of Engineering and Technology, 11(6)*, 2017, 864–871.

[43] A. Singh, M. N. Halgamuge, and R. Lakshmiganthan, *Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms, International Journal of Advanced Computer Science and Applications, 8(12)*, 2017, 1–10.

[44] Z. Yu, H. Chen, J. Liuxs, J. You, H. Leung, and G. Han, *Hybrid k-Nearest Neighbor Classifier, IEEE Transactions on Cybernetics, 46(6)*, 2016, 1263–1275.

[45] Y. Tang, L. Jing, P. M. Atkinson, and H. Li, *A multiple-point spatially weighted k -NN classifier for remote sensing, International Journal of Remote Sensing, 37(18)*, 2016, 4441–4459.

[46] M. Qasem and M. Nour, *Improving Accuracy for Classifying Selected Medical Datasets with Weighted Nearest Neighbors and Fuzzy Nearest Neighbors Algorithms, in 2015 International Conference on Cloud Computing*, 2015, 1–9.

[47] P. Nair, N. Khatri, and I. Kashyap, *A novel technique: ensemble hybrid 1NN model using stacking approach, International Journal of Information Technology, Springer,* 2018.

[48] F. Qin, L. Li, S. Gao, X. Yang, and X. Chen, *A deep learning approach to the classification of 3D CAD models, Journal of Zhejiang University SCIENCE C, 15(2)*, 2014, 91–106.

[49] A. Asuncion and D. Newman, *UCI Machine Learning Repository*, 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[50] T. Cover and P. Hart, *Nearest neighbor pattern classification, IEEE Transactions on Information Theory, 13(1)*, 1967, 21–27.

[51] K. Tan, Steinbach, Karpatne, *Introduction to Data Mining* (Pearson Education India, 2007)

[52] Q. Weinberger and K.Saul, *Distance metric learning for large margin nearest neighbor classification, Journal of Machine Learning Research, 10 (2)*, 2009, 207–244.

[53] A. B. Hassanat, *Dimensionality Invariant Similarity Measure, Journal of American Science,10(8)*, 2014, 221–226.

[54] V. Hase, S. Verma, Y. Bhalerao, V. Wakchaure, and G. Vikhe Patil, *Manuscripts submitted for publication*.Unpublished work.

[55] H. Laga and M. Nakajima, *Supervised Learning of Similarity Measures for Content-Based 3D Model Retrieval, in Large-Scale Knowledge Resources. Construction and Application,* 2008, 210–225.

[56] National Design Repository, Drexel University. [Online]. Available: http://edge.cs.drexel.edu/repository/.

[57] AIM@SHAPE. [Online]. Available: http://visionair.ge.imati.cnr.it/ontologies/shapes/.