International Information and Engineering Technology Association
Advancing the World of Information and Engineering

A Comparative Study of Object Classification Methods Using 3D Zernike Moment on 3D Point Clouds

Erdal Özbay[*], Ahmet Çınar

Department of Computer Engineering, Faculty of Engineering, Firat University, Elazig 23119, Turkey

Corresponding Author Email: erdalozbay@firat.edu.tr

ABSTRACT

The point clouds provide responsive geometric representation on many applications. The classification of objects through point clouds is one of the popular subjects of recent years. In this study, we introduce the potential of the 3D Zernike Moment approach for the object classification on the 3D point cloud. Zernike Moment (ZM) has utilized as a feature extractor of point clouds. This paper presents a comparative study of the state-of-the-art classification methods with respect to machine learning algorithms and PointNet which have been developed for classification by Stanford University. Object classification has been applied to a dataset with labeled 3D Zernike Moment features inferences obtained from the 3D point cloud. The performance of the developed method is verified by comparing the experimental results on the Washington RGB-D Object Dataset which consists of forty-five different household objects as point cloud data. Fine Gaussian SVM gives the best results in accuracy (96.0%) according to the results obtained with built-in cross-validation results. The results of the proposed classification of 3D Point Clouds on the 3D Zernike Moment features have significantly higher accuracy. The classification of 3D Zernike Moments on point cloud compared to directly point cloud classification is efficient and effective and lower complexity computation is obtained. It is emphasized that the 3D Zernike Moment features can be optimized for classification. In general, the comparative validation results have been reached a high accuracy in the proposed method. In the future, 3D Zernike Moment feature extractions are emphasized for the usability of classification operations on 3D data.

1. INTRODUCTION

The three-dimensional (3D) point cloud obtained with built-in sensors such as a laser scanner, time-of-flight camera, or stereographic systems provides a reliable and convenient source of information for computer graphics [1]. Point cloud needs to be processed as a basic step in many applications such as segmentation and classification of objects, identification of uncertain areas, and completion of missing parts [2, 3]. Point cloud generation has become easier with the developed 3D scanners being modular and the improvement of positioning systems [4, 5]. Point cloud collection with high accuracy is generally examined in surveying, investigation, and autonomous scene searching and object inventory [6]. The qualified results obtained from these studies are used to create 3D object models for applications such as visualization and simulation.

Many researchers have discussed the object classification with specific segmentation algorithms using data characteristics [7]. A number of new studies have been demonstrated on the voxel structure of the point clouds of scanned objects [8]. A variety of machine learning techniques have been applied to classify 3D point clouds [9]. In addition, supervised machine learning is utilized by presenting pre-labeled examples to obtain useful predictive models that can be applied to new data [10, 11]. Especially in the last few years, neural networks have been the basis of the methods used by advanced computer vision algorithms in many areas such as

classification [12], segmentation [13] and target detection [14]. Generally, the data processed by the neural networks referenced in these applications are two dimensional. However, although the modeling obtained by reconstruction in the 3D graphics area allows capturing a large amount of data, the costs of operating this data in neural networks cannot be reduced [15, 16].

In previous studies, researchers have explored deep learning as well as using state-of-the-art machine learning algorithms to classify over point cloud clusters [17]. PointNet is a preliminary method that can work and process directly on point sets [18]. PointNet's basic working logic is to learn the spatial coding of each point and then to collect all individual point features in a global point cloud signature [17]. In addition, a hierarchical neural network called PointNet ++ is provided, where many points can be processed in a hierarchical manner [17]. However, the cost of time and space between both deep learning methods has increased dramatically compared to introduced Zernike based state-of-the-art methods.

Moments are used in many studies on subjects such as computer vision, image processing, pattern recognition, and multifunctional analysis. In some of these studies, Zernike Moments have been used on orthogonal polynomials [19]. Zernike Moment is independent of the rotational movements on the images of objects and is a commonly used tool in the fields of shape recognition and classification. Similarly, the amplitudes of 3D Zernike Moments use this feature in three-

dimensional image structures [20]. With the help of sensor cameras, which have become widespread in recent years, it has become possible to take three-dimensional images of various scenes. Zernike Moments are calculated in the whole image and can give successful results in object recognition and identification. However, Zernike Moment is not definitely successful in subjects such as an object classification, where image information is more important than image.

For this reason, instead of calculating the moments on the whole image, a new principle calculation of point cloud Zernike Moment around each and every neighbor has been used. The moments of a point set may express the formal characteristics of the distribution of this cluster, such as the center of gravity, variance, skewness, and kurtosis. In this case, voxel data can also be considered as a set of points and the formal properties of its distribution can be evaluated over the moments. As a result, 3D Zernike Moment results of voxel data can be followed by using a parsing result of object classification.

As an application environment, Matlab software has included the Statistics and Machine Learning toolbox, which includes a large number of machine learning algorithms [21]. The Classification Learner Application Toolbox provides quick access to supervised learning methods. The methods in the toolbox are used in many different real-world applications, such as object classification [21].
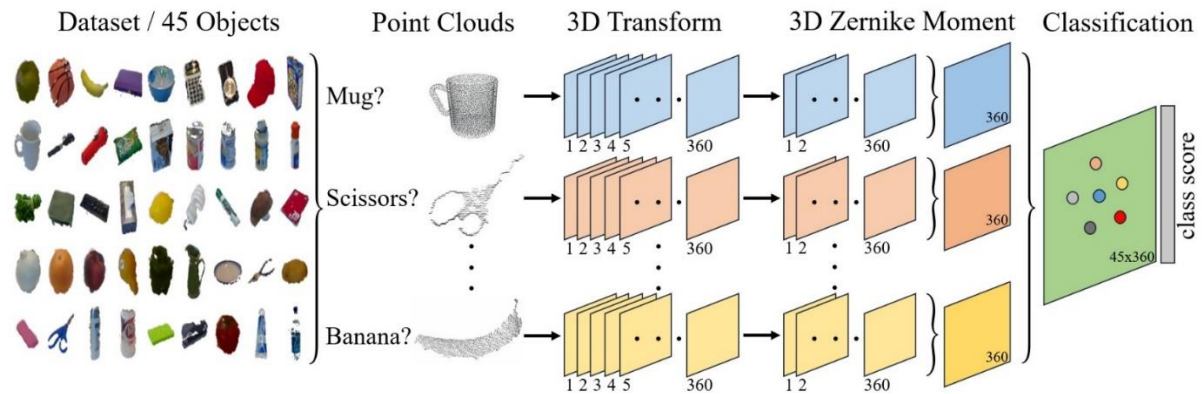

Figure 1. Demonstration of the proposed classification method which represents by feature extraction of 3D Zernike moments on point clouds

In this paper, a comparative study of all machine learning methods in the Classification Learner App for object classification is proposed with 3D Zernike Moments derived from point clouds. For this purpose, Washington Object Dataset with 45 different object datasets has been used for object classification [22]. Classification results have been compared with 22- state-of-the-art classification method and PointNet in terms of computation times and predictive performance. Specifically, it has been constructed over several 3D point clouds of different structures for analysis. Figure 1 shows the flowchart of the proposed method.

Figure 1 shows comparing the classification of a used data set that consists of 45 different small objects using traditional machine learning and deep learning-based classification results. Each point clouds cluster of objects contains three-dimensional numeric information stored in the .ply (Polygon File Format or the Stanford Triangle Format) file in floating point type. Each dataset is augmented by the 3D Transformation process using the original data to calculate the Zernike Moments. In this respect, the data is fixed and is the x, y, z (3D information) stored in the file. 3D Zernike Moment calculation is performed on all data which have 360 features extractions results for objects classification. In this respect, the results of the proposed classification of 3D Point Clouds on the 3D Zernike Moment features have significantly higher accuracy. In particular, the comparative results of the 22 most advanced machine learning methods and the deep learning-based PointNet and PointNet++ 3D point cloud classification methods have been evaluated. It is emphasized that the 3D Zernike Moment feature can be optimized for classification. In general, the comparative validation results have been reached a high accuracy in the proposed method.

Washington RGB-D Object Dataset is used in the article.

The dataset contains the 3D point clouds of views of each object, in .pcd file format readable with the Point Cloud Library (PCL). There is a part of the dataset available containing cropped images of the objects for extracting visual features, and another part of the dataset containing the full 640x480 images from the sensor. As it is known, since point cloud data are numerical coordinate values, they are not affected by environment variables (light, shadow, reflection etc.).

The structure of the paper is as follows: In the second section of the paper, the Materials and Methods section includes 3D transformation, 3D Zernike Moment computation, classification learner apps and PointNet deep learning for object classification. Then, experimental results are analyzed in section three. Finally, the conclusions offer a discussion.

2. MATERIALS AND METHODS

The reconstruction, classification, segmentation, clustering and recognition processes of 3D objects have been important research subjects for the last decade [23]. Studies in this area have benefited from the capabilities of 3D moment invariants as a 3D object shape descriptors [24, 25]. 2D Zernike Moment has been used in various studies such as edge detection [26], image retrieval [27], recognition applications [28], and biometrics [29] in computer graphic fields. The classical 2D Zernike polynomials aim at the orthogonal strengthening of non-orthogonal moments by making 3D. In some studies, several theoretical aspects of the derivation of moments and polynomials of 3D Zernike have been discussed [20].

The reconstruction of a group of 3D Zernike Moment with Point Clouds is a simple and efficient process. In addition, the

3D Zernike Moment is capable of collecting global information about the 3D shapes of objects without needing to specify closed boundaries. It is important to extraction the structure-function relationships of 3D objects through improving shape analysis techniques.

2.1 3D Point cloud transformation

In this study, the full data set of Washington RGB-D Object consisting of 45 independent objects has been examined. There are 45 different .pcd files for each object in this data set. First of all, the data set is converted into a .ply format file containing ASCII numeric values. 3D affine transformation is implemented with x, y, z coordinate values of 3D point clouds. Firstly, the entire set of 3D point clouds is shifted to the central origin for 3D geometric transformation. The 3D affine transformation is then applied to all data. The data set for each object is transformed one degree. In total, 360 different datasets are obtained for the 360-degree transformation of the same object. As a result, a data set containing 360 sets of 3D point clouds is obtained for each data set of 45 objects. Transform operations are applied to each set of 3D point clouds in .ply format. As a result of the transformation, the object particles and their relative distances are maintained [8].

2.2 3D Zernike moment computation

3D Zernike Moment calculation step is applied to 360 different data sets of each object. The entire dataset consists of 45 different objects. Since a 3D Zernike Moment result has been obtained from each data set, a totally 45x360 data cluster has emerged. Three-dimensional Zernike polynomials on point cloud during the calculation of classical Zernike Moment are defined as $Z_{l,m,n}$, orthogonal polynomials [30];

$$Z_{l,m,n}(\Re) = R_{l,m}(r)Y_{m,n}(\theta, \phi), \quad (1)$$

where, $l \in [0, Max]$, $m \in [0, l]$, and $n \in [-m, m]$. The $(l - m)$ must be selected from integers with positive values. The maximum order is defined as a max-term during the calculation operations. $R_{l,m}(r)$ and $Y_{m,n}(\theta, \phi)$ are referred to as radial functions and spherical harmonics with real numerical value. As described in Eq. (2), 3D Zernike can be expanded using the polynomials defined in a unit ball of any function $f(\Re)$;

$$f(\Re) = \sum_{l=0}^{\infty} \sum_{m=0}^{l} \sum_{n=-m}^{m} \Omega_{l,m,n} Z_{l,m,n}(\Re). \quad (2)$$

Ω, which is the coefficient of expansion in Eq. (2), represents 3D Zernike Moment. Then the complex conjugate of polynomials as in Eq. (3) is generated.

$$\Omega_{l,m,n} = \int_0^1 \int_0^{2\pi} \int_0^{\pi} \overline{Z_{l,m,n}(\Re)} f(\Re)(r^2 \sin\theta \, dr d\theta d\phi) \quad (3)$$

The transformation between 3D spherical and cartesian coordinates is formulated with 3D Zernike polynomials as follows;

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \sin\theta \sin\phi \\ r \sin\theta \cos\phi \\ r \cos\phi \end{bmatrix}. \quad (4)$$

3D Zernike polynomials are defined as follows;

$$Z_{l,m,n}(X) = \sum_{v=0}^{k} Q_{k,m,v} |X|^{2v} e_{m,n}(X), \quad (5)$$

where, $k = (l - m)/2$, is an integer value in the interval of $0 \leq v \leq k$. So the coefficient of $Q_{k,m,v}$ is defined as;

$$Q_{k,m,v} = \frac{(-1)^k}{2^{2k}} \sqrt{\frac{2m+4k+3}{3}} \binom{2k}{k} (-1)^v \frac{\binom{k}{v}\binom{2(k+m+v)+1}{2k}}{\binom{k+m+v}{k}} \quad (6)$$

The formulas given above implements the calculation of 3D Zernike Moments with very fast and low complexity using voxels of point clouds. 3D Zernike Moments are expressed as the mathematical calculation of 3D monomial terms over digital point cloud voxels. The 3D Zernike Moment calculation of the point clouds of the original data in the .ply format for each object in the data set is performed in less than one second. In this respect, the 3D Zernike Moment feature extraction step calculated in milliseconds has no additional cost to the classification process [31].

3D Zernike descriptors are generally used to compare to similar structures and the vectors, whereas the independent 3D Zernike Moment is used for feature computation in object classification. In this way, 360 3D Zernike Moment feature results of 45 different objects have labeled to be executed in classification learner application. The definition of a set of suitable features for the high accuracy classification of the 3D point cloud is an issue that directly affects success [32]. In this study, the higher performance is obtained with the classification of 3D Zernike Moment features rather than the direct classification of 3D point clouds.

2.3 The classification learner application

In this paper Classification Learner app in the Statistics and Machine Learning toolbox is used. The state-of-the-art supervised machine learning classification algorithms incorporated into the toolbox through this application provide automated training of the dataset used. Subsequently, the trained classifiers can be exported to the Matlab workspace, where they can be used to compute predictions of new input data using the predictFcn function of the Matlab software. The Matlab software contains 22 popular classifier types and 5 different major classification algorithms in the Classification Learner app toolbox [33]. In following some analyze method has explained:

• *Decision Tree:* The basic principle is based on the division of the input data into groups by means of a clustering algorithm. In this respect, different classes are predicted by selecting all branches starting from root to leaf nodes in a tree structure. The clustering process continues in depth until all the elements of the group have the same class label. There are three different types of decision trees in the application, they called: Fine, Medium, and Coarse.

• *Discriminant analysis:* It is a discriminatory analysis aiming to evaluate the adequacy of classification by characterizing the classes belonging to group members and aiming to find combinations of features that different characteristics between multiple groups. This method is particularly useful for problems with a large number of classes. There are two different types: Linear and Quadratic. The LDA is highly interpretable because it allows reduction dimensionality. It is also possible to model non-linear relationships with QDA. QDA is a regular discriminant analysis technique that is particularly useful for a large number

of features. It is often useful to perform regularized discriminant analysis for many features.

- ***Support vector machine (SVM):*** It is one of the most effective and simple methods used in classification. It works with the logic of separating the boundaries between different groups in a plane for classification by drawing with mathematical functions. The point at which this boundary is to be drawn should be chosen at the furthest distance from members of different groups. SVM determines how to draw this limit. SVM has six different types: Linear, Quadratic, Cubic, Fine Gaussian, Medium Gaussian, and Coarse Gaussian.

- ***K nearest neighbors (KNN):*** is one of the easiest to implement supervised learning algorithms. It is used for solving both classification and regression problems. In this algorithm, data from a sample set with certain classes are used. The distance of the new data to be included in the sample data set according to the existing data is calculated and its nearest neighbor is examined. KNN has six different types: Fine, Medium, Coarse, Cosine, Cubic, and Weighted.

- ***Ensemble classification:*** is a combination of two or more independent classification methods to improve their individual training performance. Five different types are available: Boosted Trees, Bagged Trees, Subspace Discriminant, Subspace KNN, and RUSBoosted Trees.

2.4 Pointnet and PointNet ++ deep learning

PointNet is a pioneering study that can handle the classification of 3D point cloud clusters directly with deep learning [18]. The basic idea of PointNet is to learn a spatial encoding of each point and then aggregate all individual point features to a global point cloud signature [17]. In design, CNN regularly receives a series of data as input and can progressively capture features at increasing scales throughout a high-resolution hierarchy. The neurons have smaller receptor fields at low-resolution levels while larger the receiving fields at the higher-resolution levels. Taking advantage of the abstraction ability of local patterns, it is possible to generalize the unseen cases throughout the hierarchy.

As the second version of the PointNet study, researchers have introduced a hierarchical neural network model called PointNet ++ [17]. The basic approach in PointNet ++ is to divide the set of points into the measure of the distance between the overlapping local regions and the underlying space. Local features that capture fine geometric structures from small neighborhoods are extracted by CNN logic; such larger local features are grouped into units and processed to produce superior features. This process is repeated until the features of the entire 3D point cloud cluster used are obtained.

3. EXPERIMENTAL RESULTS

This section is about the analysis of the comparative results of the classification of 3D point clouds belonging to a set of 45 different objects with all machine learning algorithms using the classification learner app. At the same time, using Python programming language for the same dataset, classification results of 3D point clouds belonging to PointNet and PointNet++ have been compared. In particular, the results obtained by using 22 state-of-the-art classification algorithms have compared in terms of predictive performance and computation times. A computer with an Intel Core i7 processor

with 2.3GHz and 8GB RAM has used to obtain all experimental results.

The predictive performance criterion of the experimental results has determined as the accuracy of the classification algorithms. The accuracy value is calculated as follows:

$$Accuracy = 100 . \frac{TP + TN}{TP + TN + FP + FN} \qquad (7)$$

In Eq. (7), *TP*, *TN*, *FP*, and *FN* express the true positive, true negative, false positive, and false negative numbers respectively. Computation time (*t*) and prediction accuracy (*Acc*) obtained from 3D point clouds by using *Cross-Validation* (with $q = 5$ fold) options of each machine learning algorithm are shown in Table 1. The *Acc* values are obtained using the confusion matrix provided by the built-in validation of the application. In these experiments, two built-in validation options, *Cross-Validation*, and *No-Validation* produce similar *Acc* values. However, the *Cross-Validation* results are slightly more ideal than *No-Validation*. Table 1 depicts the results of cross-validation value of coarse 3D point clouds.

Table 1. Cross-validation results of 3D point clouds

#	Group	Predictive Model	*t* (s)	*Acc* (%)
1		Fine Tree	5.3	55.8
2	A	Medium Tree	3.6	41.8
3		Coarse Tree	**3.1**	26.6
4	B	Linear Discriminant	4.5	36.6
5		Quadratic Discriminant	8.0	62.6
6		Linear SVM	15202.0	56.5
7		Quadratic SVM	30900.0	74.1
8		Cubic SVM	52193.0	71.2
9	C	Fine Gaussian SVM	1893.3	85.1
10		Medium Gaussian SVM	3527.1	78.0
11		Coarse Gaussian SVM	6449.1	67.5
12		Fine KNN	6474.0	80.0
13		Medium KNN	6474.4	83.1
14		Coarse KNN	6484.6	77.7
15	D	Cosine KNN	6712.7	53.2
16		Cubic KNN	6723.9	83.2
17		Weighted KNN	6728.8	80.6
18		Boosted Trees	6891.4	46.7
19		Bagged Trees	7018.2	**85.2**
20		Subspace Discriminant	7079.7	35.1
21	E	Subspace KNN	7144.4	45.0
22		RUSBoosted Trees	7262.2	38.9

In Table 1, the classification has evaluated on the Washington RGB-Dataset using machine learning classification techniques, which are sub-classification techniques divided into five main groups. According to Table 3, the Bagged Trees algorithm of the Ensemble group has given the best result when accuracy is taken as reference. Although it is not very good in terms of time, the method is efficient because accuracy is taken as a reference in offline operations. However, although this method gives the best results on the data used, it is seen that it has much lower accuracy values compared to our proposed 3D Zernike Moment feature extraction classification method. In this respect, the superiority of the proposed method emerges.

The computation time (*t*) and the accuracy (*Acc*) obtained from 360-degree 3D Zernike Moment feature results of the 45 different objects using the *Cross-Validation* (with $q = 5$ fold) options of each machine learning algorithm are shown in Table 2.

Table 2. Cross-validation results of 3D Zernike moment features on 3D point clouds

#	Group	Predictive Model	*t* (s)	*Acc* (%)
1		Fine Tree	8.1	95.4
2	A	Medium Tree	6.6	66.7
3		Coarse Tree	**6.4**	20.0
4	B	Linear Discriminant	8.7	91.2
5		Quadratic Discriminant	8.5	94.7
6		Linear SVM	42.6	95.2
7		Quadratic SVM	87.6	94.8
8		Cubic SVM	112.1	94.4
9	C	Fine Gaussian SVM	47.6	**96.0**
10		Medium Gaussian SVM	72.7	95.4
11		Coarse Gaussian SVM	82.9	94.9
12		Fine KNN	73.9	93.8
13		Medium KNN	73.8	95.3
14		Coarse KNN	74.6	94.7
15	D	Cosine KNN	77.5	8.0
16		Cubic KNN	78.4	95.3
17		Weighted KNN	78.9	93.9
18		Boosted Trees	92.9	96.0
19		Bagged Trees	91.2	93.9
20		Subspace Discriminant	95.2	91.2
21	E	Subspace KNN	98.2	93.8
22		RUSBoosted Trees	105.1	66.7

In Table 2, 3D Zernike Moments of 3D original point cloud data are obtained and classification is given in terms of time cost and accuracy values. Table 2 shows the highest accuracy of 96.0% of the Fine Gaussian SVM classification algorithm which is in the Support Vector Machines group. The highest accuracy value of Table 1 shows that the Bagged Trees algorithm of the Ensemble group yields 85.2%, while the proposed 3D Zernike Moment data classification method achieves a higher accuracy of 96.0%. In this respect, a relative improvement of 12.67% has been achieved. The value of the Bagged Trees algorithm, which produces the best accuracy results in Table 1 in terms of time cost, is 7018.2, while value of the Fine Gaussian SVM method in Table 2 in the proposed 3D Zernike Moment data classification is 47.6. Thus, a significant improvement has been achieved in terms of time cost. From this perspective, it is better to work with 3D Zernike Moment values instead of working with raw 3D point clouds. In addition, our system is highly efficient in terms of both accuracy and time cost as shown in Table 2.

Table 3. Comparison of average results of 3D point cloud and 3D Zernike moment classification accuracy values in five different state-of-the-art categories

#	Methods	3D Point Clouds *Acc* (%) (avg)	3D Zernike Moment *Acc* (%) (avg)
A	Decision Tree	41.4	60.7
B	Discriminant Analysis	49.6	92.9
C	Support Vector Machines	72.0	**95.1**
D	K-Nearest Neighbors	**76.3**	80.1
E	Ensemble	50.1	88.3

Table 3 is the generalized version of Table 1 and Table 2. In this generalization, classification algorithms are divided into five general groups. These groups are Decision Trees, Discriminant Analysis, Support Vector Machines, K-nearest

neighbors, Community classification methods. The first column in Table 3 is the average of the accuracy values obtained with 3D raw point clouds and the highest result is 76.3% of the KNN group. However, the highest column of the average values of the 3D Zernike Moment data and the classification results in the second column in Table 3 is 95.1% of the Support Vector Machines group. According to these values, it is seen that there is an improvement of 24.63%.

The results of PointNet and its hierarchical learning architecture PointNet++ are compared with the Washington Object dataset in terms of accuracy and computation time. For this purpose, 3D point clouds in .ply format belonging to 45 different objects have been first converted to .off (object file format) format. Then, the feature learning architecture has been implemented with CNN over Python software language. The evaluation of the network on 3D point clouds classification is given in Table 4.

Table 4. Classification results of 3D point clouds

#	Methods	*t* (s)	*Acc* (%)
1	PointNet (for 3D point cloud) [16]	29488.5	89.3
2	PointNet++ (for 3D point cloud) [15]	41282.9	90.8
3	Bagged Trees (for 3D point cloud data)	7018.2	85.2
4	Fine Gaussian SVM (for 3D Zernike Moment)	**47.6**	**96.0**

When Table 4 is analyzed, the results of PointNet and PointNet++ methods developed at Stanford University for direct classification through point clouds have given higher accuracy compared to the Bagged Trees machine learning algorithm which obtained the best accuracy value in Table 2, but they have worse accuracy values than Fine Gaussian SVM classification algorithm obtained with 3D Zernike Moments. In contrast to the Fine Gaussian SVM accuracy of 96%, the accuracy of PointNet and PointNet++ algorithms is 89.3% and 90.8%. When all these results are considered, it is seen that the data obtained with 3D Zernike Moments in terms of classification of point clouds reached higher accuracy values.

PointNet unified network takes data as input and output. These data are about the classes label and point segment. PointNet uses max pooling. This framework work on Python programming language. The network takes input points as features, finally an output is classification scores for *k* classes. As activation functions, RELU (Rectified Linear Unit) is used. In the Classification application, the predictive accuracy of the training model has implemented by the cross-validation method. In this method, data is selected by dividing *q* into discrete sets. Only one set is used to validate the model, while the other *q* - 1 is used for training. This process is repeated *q* times and the confusion matrix is obtained as the result of the arithmetic mean of the iteration results. Here, *q* = 5 is set by default.

The innovative aspect of this paper is the classification of point clouds with 3D Zernike Moment feature extractions, which outperforms 22 state-of-the-art machine learning and the trend over recent years of deep learning-based PointNet algorithms. Point clouds are a structure that contains a lot of data. Therefore, it has an inefficient process in terms of time cost when used directly in the classification process. Likewise, PointNet, which is used directly in the deep learning-based classification process of point clouds, has not provided an improvement in time cost. In this respect, the superiority of the proposed method in Table 2 is shown in seconds, especially when compared to time cost with Table 1 and Table 4.

Moreover, the proposed method obtained experimental results with higher accuracy compared to both machine learning and deep learning-based methods. The most innovative aspect of the proposed method is the use of 3D Zernike Moments derived from the point cloud, rather than using all points in the classification of point cloud data. Thus, the classification process is isolated from unnecessary point cloud crowds and the process is shortened in terms of time cost.

4. CONCLUSIONS

In this paper, traditional machine learning and deep learning-based classification methods are compared to a dataset consisting of 45 different small objects. In this respect, the results of the proposed classification of 3D Point Clouds on the 3D Zernike Moment features have significantly higher accuracy. In particular, the comparative results of the 22 most advanced machine learning methods and the deep learning-based PointNet and PointNet++ 3D point cloud classification methods have been evaluated. It is emphasized that the 3D Zernike Moment feature can be optimized for classification. In general, the comparative validation results have been reached a high accuracy in the proposed method. The computation time of PointNet and PointNet++ has taken a long time in contrast to the machine learning algorithms such as Cubic SVM. In particular, the highest accuracy value has been achieved in Support Vector Machine and discriminant analysis methods. The highest accuracy of 96.0% has been obtained with Fine Gaussian SVM, a Support Vector Machine algorithm. Higher accuracy has been obtained in the state-of-the-art machine learning algorithms except Medium Tree, Coarse Tree, Cosine KNN, and RUSBoosted Trees.

The classification algorithms used in future studies are planned to be implemented on a GPU-based parallel system. Thus, real-time studies will be emphasized with faster and more accurate results.

REFERENCES

[1] Liu, H., He, G., Yu, H., Zhuang, Y., Wang, W. (2016). Fast 3D scene segmentation and classification with sequential 2D laser scanning data in urban environments. In 2016 35th Chinese Control Conference (CCC), pp. 7131-7136. https://doi.org/10.1109/ChiCC.2016.7554484

[2] Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A. (2011). On the segmentation of 3D LIDAR point clouds. In 2011 IEEE International Conference on Robotics and Automation, pp. 2798-2805. https://doi.org/10.1109/ICRA.2011.5979818

[3] Balado, J., Díaz-Vilariño, L., Arias, P., González-Jorge, H. (2018). Automatic classification of urban ground elements from mobile laser scanning data. Automation in Construction, 86: 226-239. https://doi.org/10.1016/j.autcon.2017.09.004

[4] Ozbay, E., Cinar, A. (2013). 3D reconstruction technique with kinect and point cloud computing. Global Journal on Technology, 3: 1748-1754.

[5] Zeybek, M., Şanlıoğlu, İ. (2019). Point cloud filtering on UAV based point cloud. Measurement, 133: 99-111. https://doi.org/10.1016/j.measurement.2018.10.013

[6] Lalonde, J.F., Vandapel, N., Hebert, M. (2007). Data structures for efficient dynamic processing in 3-D. The international Journal of Robotics Research, 26(8): 777-796. https://doi.org/10.1177/0278364907079265

[7] Vu, H., Nguyen, H.T., Chu, P.M., Zhang, W., Cho, S., Park, Y.W., Cho, K. (2017). Adaptive ground segmentation method for real-time mobile robot control. International Journal of Advanced Robotic Systems, 14(6): 1729881417748135. https://doi.org/10.1177/1729881417748135

[8] Özbay, E., Çinar, A. (2019). A voxelize structured refinement method for registration of point clouds from Kinect sensors. Engineering Science and Technology, an International Journal, 22(2): 555-568. https://doi.org/10.1016/j.jestch.2018.09.012

[9] Hu, H., Munoz, D., Bagnell, J.A., Hebert, M. (2013). Efficient 3-D scene analysis from streaming data. In 2013 IEEE International Conference on Robotics and Automation, pp. 2297-2304. https://doi.org/10.1109/ICRA.2013.6630888

[10] Güner, A., Alçin, Ö.F., Şengür, A. (2019). Automatic digital modulation classification using extreme learning machine with local binary pattern histogram features. Measurement, 145: 214-225. https://doi.org/10.1016/j.measurement.2019.05.061

[11] Plaza-Leiva, V., Gomez-Ruiz, J., Mandow, A., García-Cerezo, A. (2017). Voxel-based neighborhood for spatial shape pattern classification of lidar point clouds with supervised learning. Sensors, 17(3): 594. https://doi.org/10.3390/s17030594

[12] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[13] Ghiasi, G., Fowlkes, C.C. (2016). Laplacian pyramid reconstruction and refinement for semantic segmentation. In European Conference on Computer Vision, pp. 519-534. https://doi.org/10.1007/978-3-319-46487-9_32

[14] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems, 91-99. https://doi.org/10.1109/TPAMI.2016.2577031

[15] Huang, Q., Wang, H., Koltun, V. (2015). Single-view reconstruction via joint analysis of image and shape collections. ACM Transactions on Graphics (TOG), 34(4): 87. https://doi.org/10.1145/2766890

[16] Yao, X., Guo, J., Hu, J., Cao, Q. (2019). Using deep learning in semantic classification for point cloud data. IEEE Access, 7: 37121-37130. https://doi.org/10.1109/ACCESS.2019.2905546

[17] Qi, C.R., Yi, L., Su, H., Guibas, L.J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems, 5099-5108.

[18] Qi, C.R., Su, H., Mo, K., Guibas, L.J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652-660. https://doi.org/10.1109/CVPR.2017.16

[19] Teague, M.R. (1980). Image analysis via the general theory of moments. JOSA, 70(8): 920-930. https://doi.org/10.1364/JOSA.70.000920

[20] Canterakis, N. (1999). 3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition. In In 11th Scandinavian Conf. on Image Analysis.

[21] Miškuf, M., Michalik, P., Zolotová, I. (2017). Data mining in cloud usage data with Matlab's statistics and machine learning toolbox. In 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 000377-000382. IEEE. https://doi.org/10.1109/SAMI.2017.7880337

[22] https://rgbd-dataset.cs.washington.edu/dataset/rgbd-dataset_pcd_ascii/, Dataset: accessed on 11.15, 2019.

[23] Ozbay, E., Cinar, A., Guler, Z. (2018). A hybrid method for skeleton extraction on Kinect sensor data: Combination of L1-Median and Laplacian shrinking algorithms. Measurement, 125: 535-544. https://doi.org/10.1016/j.measurement.2019.107220

[24] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D. (2003). A search engine for 3D models. ACM Transactions on Graphics (TOG), 22(1): 83-105. https://doi.org/10.1145/588272.588279

[25] Hu, W., Liu, H., Hu, C., Wang, S., Chen, D., Mo, J., Liang, Q. (2013). Vision-based force measurement using pseudo-Zernike moment invariants. Measurement, 46(10): 4293-4305. https://doi.org/10.1016/j.measurement.2013.08.022

[26] Qu, Y.D., Cui, C.S., Chen, S.B., Li, J.Q. (2005). A fast subpixel edge detection method using Sobel-Zernike moments operator. Image and Vision Computing, 23(1): 11-17. https://doi.org/10.1016/j.imavis.2004.07.003

[27] Kim, W.C., Song, J.Y., Kim, S.W., Park, S. (2008). Image retrieval model based on weighted visual features determined by relevance feedback. Information Sciences, 178(22): 4301-4313. https://doi.org/10.1016/j.ins.2008.06.025

[28] Broumandnia, A., Shanbehzadeh, J. (2007). Fast Zernike wavelet moments for Farsi character recognition. Image and Vision Computing, 25(5): 717-726. https://doi.org/10.1016/j.imavis.2006.05.014

[29] Kim, H.J., Kim, W.Y. (2008). Eye detection in facial images using Zernike moments with SVM. ETRI Journal, 30(2): 335-337. https://doi.org/10.4218/etrij.08.0207.0150

[30] Grandison, S., Roberts, C., Morris, R.J. (2009). The application of 3D Zernike moments for the description of "model-free" molecular structure, functional motion, and structural reliability. Journal of Computational Biology, 16(3): 487-500. https://doi.org/10.1089/cmb.2008.0083

[31] Hosny, K.M., Hafez, M.A. (2012). An algorithm for fast computation of 3D Zernike moments for volumetric images. Mathematical Problems in Engineering. https://dx.doi.org/10.1155/2012/353406

[32] Behley, J., Steinhage, V., Cremers, A.B. (2012). Performance of histogram descriptors for the classification of 3D laser range data in urban environments. In 2012 IEEE International Conference on Robotics and Automation, pp. 4391-4398. https://doi.org/10.1109/ICRA.2012.6225003

[33] Pomares, A., Martínez, J.L., Mandow, A., Martínez, M.A., Morán, M., Morales, J. (2018). Ground extraction from 3D lidar point clouds with the classification learner app. In 2018 26th Mediterranean Conference on Control and Automation (MED), pp. 1-9. https://doi.org/10.1109/MED.2018.8442569