Generalization of Pointnet Framework with Synergy of Random Forest Classifier

Sridevi S[1], , Paayas P[2], Balasubramaniam E[3], Balachandran S[4], , Kanimozhi T[5], Shanmugakumar M[6], Jae Sung Choi[7]

[1]*Associate Professor, CSE Amrita School of Computing, Chennai*
[2]*Computer Science and Engg. Amrita School of Computing, Chennai*
[3]*Professor, Mech. VelTech Rangarajan Dr.Sagunthala, Chennai*
[4]*Central Library, Amrita School of Computing, Chennai*
[5]*Associate Professor, ECE, VelTech Rangarajan Dr.Sagunthala, Chennai*
[6]*Founder and CEO, MATIC,* IIITDM, Kancheepuram
[7]*Department of Computer Science Engineering, Sun Moon University,* Asan-si, Korea
Corresponding author email: balasriya@gmail.com

Abstract—In the real world, data is perceived in three dimen- sions (3D). Automatically analysing 3D visual features is essential in several real-time applications notably autonomous robots, autonomous vehicles, and augmented reality. It is apparent that 3D data are represented in a variety of ways, such as a polygonal mesh, volumetric pixel grid, point cloud, etc., in contrast to 2D images, which are represented as pixel arrays. 90% of current developments in computer vision and machine learning solely pertain to two-dimensional pictures. Point clouds are a popular type of 3D point cloud data (PCD) information captured by visual sensors such as LIDAR and RGB-Depth camera. One of crucial challenges in 3D object classification task while handling the 3D PCD information is the lack of connectivity in 3D space representation. We present a hybrid model in this paper that uses PointNet deep learning model in first phase to automatically extract features as well as reduce the dimension of features and to train a variety of machine learning models in second phase to classify such extracted features. By contrasting several machine learning models according to evaluation metrics, we are able to identify the optimal machine learning model. The presented hybrid model provided higher accuracy than existing method. Moreover, the proposed hybrid model is capable of more accurately classifying the PCD object even with the lesser dense points and computationally efficient in making quicker decisions which suits the essential requirements of real-time autonomous navigation applications.

Index Terms—Hybrid models, PointNet, SVM, RF, GBM, KNN

I. Introduction

A deep net architecture called PointNet was the first amongst handling point cloud data directly for performing diverse tasks including scene semantic processing, object categorization, and component segmentation. It was the first architecture to use point clouds as an input directly for 3D identification tasks when it was introduced in 2017[1]. It is just a 3D image of a scene or an object that is often gathered via a LiDAR (Light Detection and Ranging) sensor. These sensors produce brief light pulses and then time how long it takes for them to return to the sensor. A 3D model of the item or scene, similar to the one above, can be made using this data. LiDAR sensors are becoming more and more common, we may find them in some smartphones, drones, mapping aircraft, and autonomous cars. Common convolutional architectures need extremely regular input data

R. Murugan et al. (eds.), *Proceedings of the International Conference on Signal Processing and Computer Vision (SIPCOV 2023)*, Advances in Engineering Research 239,
https://doi.org/10.2991/978-94-6463-529-4_4

formats, such those of image grids or 3D voxels, for weight sharing and other kernel optimizations. Because they are not in a standard format, most researchers normally transform point clouds or models to common 3D voxel grids or collections of photographs before feeding them to a deep net architecture. However, this data representation change makes the resulting data needlessly large and introduces quantization artefacts that can mask the data inherent invariances.

PointNet [1] is a unified architecture that takes point clouds which is basically an unstructured data directly as input and generates either class labels for the full input or per point segment/part labels for each point in the input. Our network's basic architecture is remarkably simple because each point is treated identically and independently in the early stages. Each point is represented in the basic setup by only three coordinates (x, y, z). Normals and other local or global properties can be

used to compute additional dimensions.

One significant disadvantage of PointNet is that it does not take into account the local structure of the point cloud, which can be critical for specific jobs. To solve this, the authors suggested PointNet++ [12], a PointNet version that processes the point cloud in a hierarchical manner while taking into account local structures. PointNet is an all-around effective and adaptable deep learning architecture that is well suited for analyzing point clouds and other unstructured 3D data. It is a popular option for many 3D deep learning projects dueto its simplicity and ability to work directly with raw point clouds.

However, let's prioritise enhancing the model's accuracy. In [15], this research uses CNN and XGBoost-based hybrid models in categorization of prenatal ultrasound scan planes pictures in detection of congenital heart abnormalities and achieved a test accuracy of 98.65% on a custom dataset. This demonstrates that hybrid model tactics outperform traditional deep learning and machine learning methodologies. In [16], a neural network architecture integrating gated recurrent Unit (GRU) and support vector machine (SVM) is used to iden- tify intrusions in network traffic data by substituting SVM with softmax in the final output layer of a GRU model. The proposed model achieved 81.54% training accuracy and 84.15% testing accuracy, whereas the latter achieved 63.07% training accuracy and 70.75% testing accuracy. In [18], this work provides a hybrid CNN-KNN model for facial expression recogntion (FER) on the Raspberry Pi 4, in which CNN is used for feature extraction and the KNN is used for expression recognition.The proposed methodology achieves an accuracy of 75.26% (a minor improvement of 0.06%) when compared to the state-of-the-art Ensemble of 8 CNN model.

Therefore, in this work, we have discovered a method for fusing ML models with the Pointnet architecture. Deep learn- ing revolution in machine learning began in 2006. An up to date quick training approach for Bayesian network with plenty of hidden units is presented by Hinton et al., in [2]. Deep belief networks are the name given to these belief systems. Following their research, Krizhevsky et al. [3] created AlexNet for the Imagenet image recognition competition. Convolutional neural network AlexNet has three fully connected layers, five convolutional layers, and ReLu for non-linearity. LeCun et al., introduce convolutional neural networks [4]. The most potent algorithms up until the advent of deep learning were support vector machines (SVM) [6] and ensemble techniques like gradient boosting machines (GBM) [7], random forest (RF) [8] and K-Nearest-Neighbours (KNN) [9]. However, the feature extraction stage was what determined their success. Professional experience was necessitated to extract high-level characteristics from sets of data during the feature extraction procedure [5]. In conclusion, a hybrid method that combines deep learning architectures and machine learning algorithms can increase a model's performance. This method enables the model to extract complicated aspects from the data and gener- ate reliable conclusions that are easy to understand, leading to increased accuracy and resilience. Instead of employing only pure deep learning models, our purpose in this research is to construct hybrid models which is fusing of deep learning and machine learning models (PointNet+SVM, PointNet+RF, PointNet+GBM, and PointNet+KNN) and evaluate them to see whether we can improve results of the PointNet model. In this light, the following is a list of our contributions:

- Extracting features from the PointNet model in general.
- Using a transfer learning approach to combine featureextractors with conventional machine learning algorithms.
- Putting forth a broad foundation for combining.

II. METHODOLOGY

Two components make up the suggested strategy, and two phases serve as an illustration of the general architecture of the suggested hybrid model depicted in Fig. 1. Preparing the data-set and processing it are carried out in the first phase. The features are then taken out of the PointNet model and trained with a machine learning algorithm in the second phase, which is utilised for classification. To assess the performance accuracy of our proposed hybrid model, we used a well-known benchmark data set from a pool of data sets launched by the Princeton ModelNet project in great detail.According to the authors [11], this project aims to offer academics a thorough, organised collection of 3D CAD models for things that were found on the web. Over 150000 models were classified into 662 categories by Amazon Mechanical Turk workers.The authors contributed a data-set of 3D object models called Mod- elNet10, which is freely available for the research community in the computer vision and machine learning domains.

ModelNet10 data-set comprises 399 models in total, with 10 distinct object categories, each of which has 40–60 models. For simplicity of use, the models are supplied in the OBJ file format and are already aligned to a standard coordinate system. In our research, ModelNet10 is employed as a benchmark data-set for assessing the performance of several 3D object recognition and classification techniques.

TABLE I
DATASET DESCRIPTION

Category	Training set	Testing set
Desk	200	86
Table	392	100
Nightstand	200	86
Bed	515	100

Toilet	344	100
Dresser	200	86
Bathtub	106	50
Sofa	680	100
Monitor	465	100
Chair	889	100

A. Dataset preparation and preprocessing

We made use of the publicly accessible ModelNet10 data- set. Ten categories make up the data-set used for the proposed research. Objects like a desk, table, nightstand, bed, bathtub, dresser, bathtub, sofa, monitor, and chairs are among those ten mesh. This is frequently used in 3D scanning and modelling applications to capture an object's or environment's accurate shape and geometry. ModelNet10 3D object data-set is pre-processed to get converted from mesh object to point cloud representation with varied dense of points with 512, 1024 and 2048 points as depicted in Fig.2 (a to c) by uniform down sampling

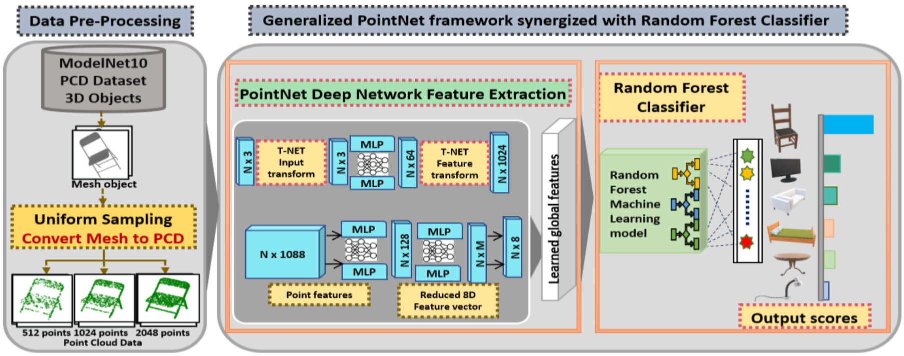

Fig. 1. Architecture of Generalized PointNet model synergized with Random Forest Classifier

(a) (b) (c)

Fig. 2. PCD data of chair object uniformly sampled with (a) 512 points (b)1024 points and (c) 2048 points

categories. From this data-set, 908 (20%) shapes were split for the testing set, and 3,991 (80%) shapes were used for the training set. Table.I clearly depicts the dataset breakup details. The practise of picking points from a mesh in a systematic and consistent manner is referred to as uniform sampling of mesh data. This method yields a point cloud, which is a collection of points that depicts the mesh's surface. The point cloud's points are evenly spaced and of consistent density, resulting in a detailed and accurate representation of the

A. Feature extraction, dimensionality reduction & Classifica-tion section

PointNet is a deep learning network created to extract usable characteristics from unprocessed point cloud data by utilising its intrinsic geometric structure [1]. PointNet's architecture is made up of three primary components namely a symmetric function, transformation layer and a shared Multi-Layer Per- ceptron (MLP). The symmetric function is applied individually to each point in the point cloud and transfers the point to a high-dimensional feature space. After that, the MLP [13] is applied to the feature vectors to extract higher-level features for classification or segmentation [1].

In order to align the raw point cloud data in a consistent and useful way, the model first applies a collection of transforma- tion networks to the data. These transformation networks can extract features like edges, corners, and other significant prop- erties since they are taught to learn the underlying geometric structure of the data. Typically, the transformation network is made up of a number of layers of neural network functions, including fully connected, convolutional, and pooling layers. Together, these layers help the primary PointNet network pro- vide a more meaningful representation of the input point cloud data. The input point cloud data is often subjected to a number of linear transformations in the first layer of the transformation network, including rotation and scaling. In order to provide the main PointNet network with point cloud data that is constant and predictable, this is done. The transformation network's subsequent layers, like the convolutional and pooling layers, usually carry out more complicated processes.

Overall, the PointNet deep learning model's transformation network is a crucial part that is in charge of pre-processing the incoming point cloud data in order to make it more suited for processing by the main PointNet network. After the data has been aligned, the model uses a series of feature extraction networks to extract meaningful features from the data. These networks can extract properties like shape, size, and texture and are trained to recognize critical features like edges, corners, and other crucial qualities. The model then classifies the data based on the collected features using a collection of classification networks. These networks are taught to identify data patterns and to categorize the data into several groups, such as objects, surfaces, or forms. In a nutshell, the PointNet model extracts characteristics from input point clouds by passing them through a succession of layers, including a shared multi-layer perceptron (MLP) [13] and a max-pooling layer. Each point in the point cloud is mined for local features using the shared MLP, and these local features are then combined into a global feature representation using the max-pooling layer. The classification layer uses this global feature representation to estimate the class or category of the point cloud after passing it through. Depending on how the PointNet model is implemented, the particular steps of the feature extraction process may vary, but the overall goal is to extract meaningful characteristics from the input point clouds that can be utilised to get accurate predicions. The main contribution of this proposed research towards successful dimensionality reduction achievement is that we chose to affix 8 nodes in the final layer of MLP prior to the softmax classification layer of the primary PointNet architecture based on numerous trial and error experimental iterations.

Generically, a hybrid strategy that combines deep learning architectures and machine learning algorithms can enhance a model's performance. This is due to the fact that machine learning methods such as support vector machines (SVMs) [6], random forests[8], gradient boosting (GBM) [7] and k-nearest neighbours (KNN) [9] can offer a reliable and understandable decision-making process, whereas deep learning models such as convolutional neural networks (CNNs) [14] and PointNet model [1] are able to extract complex features from data. Ultimately, a hybrid strategy can enhance the performance of a model. With the help of this method, the model may gather intricate details from the data and generate reliable, compre- hensible conclusions, increasing its accuracy and robustness.

III. RESULT AND ANALYSIS

The proposed research experimental results were acquired by synergizing the PointNet model with machine learning techniques in the Google Collaboratory platform. The model is executed using the 25 GB RAM-capable Tesla P100 Graphical Processing Unit. The Sklearn Python package makes it easier to utilize machine learning models and be used to assess the model using various metrics.

TABLE II
TEST PERFORMANCE OF THE PROPOSED HYBRID MODEL FOR MODELNET10 PCD DATASET SAMPLED WITH 512 POINTS

Accuracy of PointNet model with 512 dense PCD = 63%

Model	Accuracy	Sensitivity	Precision	Recall	F1
PointNet+SVM	73.6	70.4	71.5	70.4	69.9
PointNet+RF	74.7	73.3	72.9	73.3	72.7
PointNet+GB	63.7	59.7	61.7	59.7	59.3
PointNet+KNN	73.9	73.1	72.9	73.1	71.4

TABLE III
TEST PERFORMANCE OF THE PROPOSED HYBRID MODEL FOR MODELNET10 PCD DATASET SAMPLED WITH 1024 POINTS

Accuracy of PointNet model with 1024 dense PCD = 62%

Model	Accuracy	Sensitivity	Precision	Recall	F1
PointNet+SVM	67.3	64.4	64.6	64.4	63.8
PointNet+RF	72.1	72.5	69.6	72.5	69.7
PointNet+GB	62.6	63.9	60.4	63.9	62.1
PointNet+KNN	71.7	71.3	69.6	71.3	68.5

To examine the PointNet deep learning model's performance and the hybrid technique put forth in this research, we used the ModelNet10 data-set in the proposed work. We changed the sample points for the performance analysis experiment by 512, 1024, and 2048 points to observe the model's behaviour in relation to the various evaluation metrics. The model is executed by choosing the sparse categorical cross entropy as a loss function and is minimized by the Adam Optimizer. Additionally, at a learning rate of 0.001, the model is trained across the data-sets for 20 epochs. The system is observed to require about 2 minutes, 36 seconds, on average, for training 512 points, 5 minutes 3 seconds for 1024 points and 6 minutes 44 seconds for 2048 sample points.

After 20 epochs of training on the ModelNet10 data-set, we found that the accuracy of the PointNet model for 512 points is 63%. From Table 2, we can deduce that the hybrid model's values for each evaluation metric are greater than those of the PointNet model in each category. When compared to other hybrid models, the hybrid model generalized with the combination of random forest classifier has superior accuracy, sensitivity, precision, recall, and F1 score respectively. In a similar way, the accuracy attained using the PointNet model for 1024 points is 62%, and for 2048 points is 62.4%. Tables 3 and 4, provides information that allows us to conclude that the hybrid model generalized with the combination of

TABLE IV
TEST PERFORMANCE OF THE PROPOSED HYBRID MODEL FOR MODELNET10 PCD DATASET SAMPLED WITH 2048 POINTS

Accuracy of PointNet model with 2048 dense PCD = 62.4%

Model	Accuracy	Sensitivity	Precision	Recall	F1
PointNet+SVM	71.4	69.4	67.9	69.4	67.4
PointNet+RF	73.9	71.8	72	71.8	71.2
PointNet+GB	66.3	65	63.8	65	63.1
PointNet+KNN	73.6	71.5	71.9	71.5	70.9

random forest classifier is marginally more accurate than other machine learning models in terms of assessment measures. Random forest classifier takes 0.0165 seconds when fused with the PointNet model for 512 points, 0.0203 seconds for 1024 points, and 0.0096 seconds for 2048 points. As a result, we can conclude that the proposed hybrid strategy to generalize PointNet architecture while synergizing with the random forest machine learning classifier effectively outperforms all other combinations while classifying all three versions of point density ModelNet10 PCD objects with 512, 1024 and 2048 points.

IV. CONCLUSION

In conclusion, even when we modify the sample points as described in this study, with our proposed hybrid model we get reliable results, and the computational time required is shorter when dealing with less dense point cloud data. As a result, analysing point cloud data has proven to be beneficial when a hybrid technique combining PointNet and machine learning models is used for real time applications like autonomous navigation and many more. By combining the powerful feature extraction and dimensionality reduction powers of modified PointNet with the dependable classification abilities of a random forest classifier and other machine learning approaches, the hybrid technique outperforms either model alone. In terms of accuracy, sensitivity, precision, recall, and F1 score, our experiments on the ModelNet10 data-set revealed that this hybrid technique beats each model alone. It's vital to remember that this strategy needs significant improvement in terms of classification accuracy, further increasing learning efficiency while reducing model complexity and the accompanying processing cost. Future studies should investigate how to improve this strategy's performance even further, for example, by combining a few additional machine learning approaches in addition to those that were mentioned in this research. However, we believe that the hybrid technique of PointNet and Random Forest is a viable way for processing point cloud data, and we invite other scholars and practitioners to investigate this approach further.

REFERENCES

[1] Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas in "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation", arXiv:1612.00593 December 2^{nd} 2016.

[2] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. Neural Computation, 18(7):1527– 1554, 2006.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[4] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. ´ Gradient-based learning applied to document recognition. Proceedingsof the IEEE, 86(11):2278–2324, 1998.

[5] A. Ç ayir, I. Yenidoğan and H. Dağ, "Feature Extraction Based on Deep Learning for Some Traditional Machine Learning Methods," 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosnia and Herzegovina, 2018, pp. 494-497, doi: 10.1109/UBMK.2018.8566383.

[6] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machinelearning, 20(3):273–297, 1995.
[7] Jerome H Friedman. Greedy function approximation: a gradient boostingmachine. Annals of statistics, pages 1189–1232, 2001.
[8] Leo Breiman. Random forests. Machine learning, 45(1):5–32, 2001.
[9] Guo, Gongde & Wang, Hui & Bell, David & Bi, Yaxin. (2004). KNN Model-Based Approach in Classification.
[10] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts- Escolano, M. Cazorla and J. Azorin-Lopez, "PointNet: A 3D Convo- lutional Neural Network for real-time object class recognition," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 2016, pp. 1578-1584, doi: 10.1109/IJCNN.2016.7727386
[11] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920.
[12] Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas in Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–7 December 2017.
[13] Gardner, Matt W., and S. R. Dorling. "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmosphericsciences." Atmospheric environment 32.14-15 (1998): 2627-2636.
[14] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understand- ing of a convolutional neural network." 2017 international conferenceon engineering and technology (ICET). Ieee, 2017.
[15] Satish, S., Sridevi, S. (2023). CNN and XGBoost Based Hybrid Model in Classification of Fetal Ultrasound Scan Planes Images in Detection of Congenital Heart Defects. In: Joby, P.P., Balas, V.E., Palanisamy, R. (eds) IoT Based Control Networks and Intelligent Systems. Lec- ture Notes in Networks and Systems, vol 528. Springer, Singapore.https://doi.org/10.1007/978-981-19-5845-8 4
[16] A. F. M. Agarap, 'A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data', in Proceedings of the 2018 10th International Conference on Machine Learning and Computing, 2018, pp. 26–30.
[17] A. D. Naik and M. Supriya, 'Classification of Indian Classical Dance 3D Point Cloud Data Using Geometric Deep Learning', in Computational Vision and Bio-Inspired Computing, 2021, pp. 81–93.
[18] M. N. Ab Wahab, A. Nazir, A. T. Zhen Ren, M. H. Mohd Noor, M. F. Akbar and A. S. A. Mohamed, "Efficientnet-Lite and Hybrid CNN- KNN Implementation for Facial Expression Recognition on Raspberry Pi," in IEEE Access, vol. 9, pp. 134065-134080, 2021, doi: 10.1109/AC-CESS.2021.3113337.