Springer

Journal of Mechanical Science and Technology 00 (2010) 0000~0000
www.springerlink.com/content/1738-494x
submitted manuscript under review

KSME

An Evaluation Methodology for 3D Deep Neural Networks using Visualization in 3D Data Classification

Hyun-Tae Hwang[1], Soo-Hong Lee[1*], Hyung Gun Chi[3], Nam Kyu Kang[1], Hyeon Bae Kong[2], Jiaqi Lu[1] and Hyungseok Ohk[1]

[1] *School of Mechanical Engineering, Yonsei University, Seoul 03722, Korea*
[2] *Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 08826, Korea*
[3] *School of Mechanical Engineering, Purdue University, IN 47907, USA*

(Manuscript Received 000 0, 2009; Revised 000 0, 2009; Accepted 000 0, 2009) -please leave blank

Abstract

"Making 3D deep neural networks debuggable". In the study, we develop and propose a 3D deep neural network visualization methodology for performance evaluation of 3D deep neural networks. Our research was conducted using a 3D deep neural network model, which shows the best performance. The visualization method of the research is a method of visualizing part of the 3D object by analyzing the naive Bayesian 3D complement instance generation method and the prediction difference of each feature. The method emphasizes the influence of the network in the process of making decisions. The result of visualization through the algorithm of the study shows a clear difference based on the result class and the instance within the class, and the authors can obtain insight that can evaluate and improve the performance of the DNN (Deep Neural Networks) model by the analyzed results. 3D deep neural networks can be made "indirectly debuggable," and after the completion of the visualization method and the analysis of the result, the method can be used as the evaluation method of "general non-debuggable DNN" and as a debugging method.

Keywords: 3D deep neural network; Deep learning; Visualization; Convolutional neural network; CAD model;

1. Introduction

Deep neural networks are known to show the highest accuracy among the many methods for solving the data classification problem. The deep neural network, which has been known to show high performance for classification of 2D information, has recently been applied to other fields and has achieved excellent performance [1-4].

Recently, deep neural networks have been applied to the classification of 3D objects, and many studies showing excellent performance have been conducted. These studies have obtained successful results using various DNN design skills for standard datasets (ModelNet, ShapeNet). [5]

However, although there are good results for standard datasets, they are not sufficient for real industrial applications. DNNs cannot clearly account for the small percentage of uncertainties that cause them to malfunction. Although a DNN is non-debuggable, some researchers are currently working on solutions to the problem (e.g., research on model uncertainty [6]) There is one issue with the method, however, and researchers are unsure of the stability of the actual dataset. To utilize artificial neural networks in real industrial applications, it is necessary to solve the uncertainty (the uncertain reason for results and uncertain expectations of the result in a new dataset) and secure stability because the judgment process inside the artificial neural network is a process that cannot be actually seen or clearly explained.

In the field of 2D image classification using DNNs, there is a precedent study of model visualization. Zintgraf et al. [7] and Simonyan [8] visualize the classification model and features. They visualize the internal classification process of the artificial neural network and confirm the judgment process of the network model by the approach that is used to address the uncertainty. The network model also verifies how certain features of an input image affect classification. However, most previous

† This paper was recommended for publication in revised form by Associate Editor 000 000
* Corresponding author. Tel.: +82-2-2123-2823, Fax.: +82-2-312-2159
E-mail address: shlee@yonsei.ac.kr

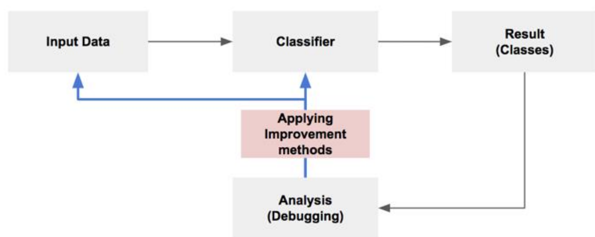


Fig. 1. Typical classifier improvement process

research is restricted to the field of 2D image classification. Zintgraf et al. [7] show results on 3D image classification for MRI scans of the human brain. However, they use a simple linear regression classifier that has limited performance due to the restrictive size of their dataset.

Therefore, in the study, we investigate the case of 3D data and deep neural networks more thoroughly and examine which features affect 3D models with high accuracy. Through the visualization process of the 3D network model, we find a feature that affects 3D model classification judgment. In this way, it is possible to visualize how 3D classification networks judge which features are used in the 3D classification field using artificial neural networks. Therefore, if the features influencing the use of a 3D classification network are known, the uncertainty of the DNN mentioned above can be addressed.

In addition, the authors believe that providing insight by visually representing the context of the DNN internal classification process to determine the uncertainty of the result by human intuition is the best DNN performance evaluation methodology at the current research level. (Fig. 1 shows that our method would be an improvement to the classifying process. Later, in the paper, specific cases will be introduced.)

Thus, in the paper, we focused on (1) deep neural networks applied to the classification of 3D objects and (2) the existing performance evaluation method of deep neural networks, and we developed the 3D DNN visualization method suitable for the 3D design field.

As a result, the study has developed a method for expressing the process of 3D deep neural network operation and provides intuitive insight into DNN performance evaluations. Through the insight, we can gain a deeper understanding of the judgment process of DNNs and gain access to their uncertainties.

The goal of the work is to provide an indirect debugging method and to help improve the 3D DNN model or 3D classification techniques.

2. Related Works

2.1 3D Object Classification

Deep neural network-based methods have been used in the field of 2D image classification as well as 3D image classification. These methods also show high performance in terms of accuracy. There are three major techniques for the notable 3D classification model, including (1) the 2.5D-based classification method [12,16-17], (2) the point cloud-based classification method [9], and (3) the voxel-based classification method [10-11]. According to Maturana et al. [11], (1) cannot completely use the geometric information of the data, and it is difficult to integrate the information according to the viewpoint; (2) has a loss of distinction between unknown space and free space; and (3) has many representations by distinguishing between the unknown space and free space. The method features a simple data structure that makes it easy to store and manipulate the model. The voxel-based classification model has the highest performance among these three classification models. According to Z. Wu et al. [5], a 10 ~ 20% improvement in performance compared to the pre-defined feature-based classification method was observed, and Brock and Andrew et al. [10] recently showed classification accuracy over 95% for standard datasets (ModelNet [5]).

The research by [13-14] is currently being applied to industries such as object picking. Therefore, it is expected that high-performance 3D object classification deep neural networks can be applied in a real industry setting.

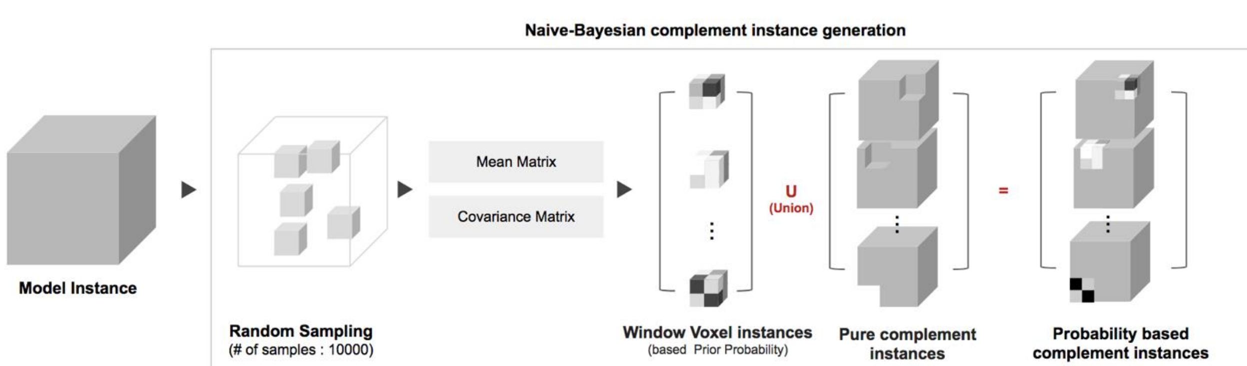


Fig. 2. Naïve Bayesian 3D complement instance generation method

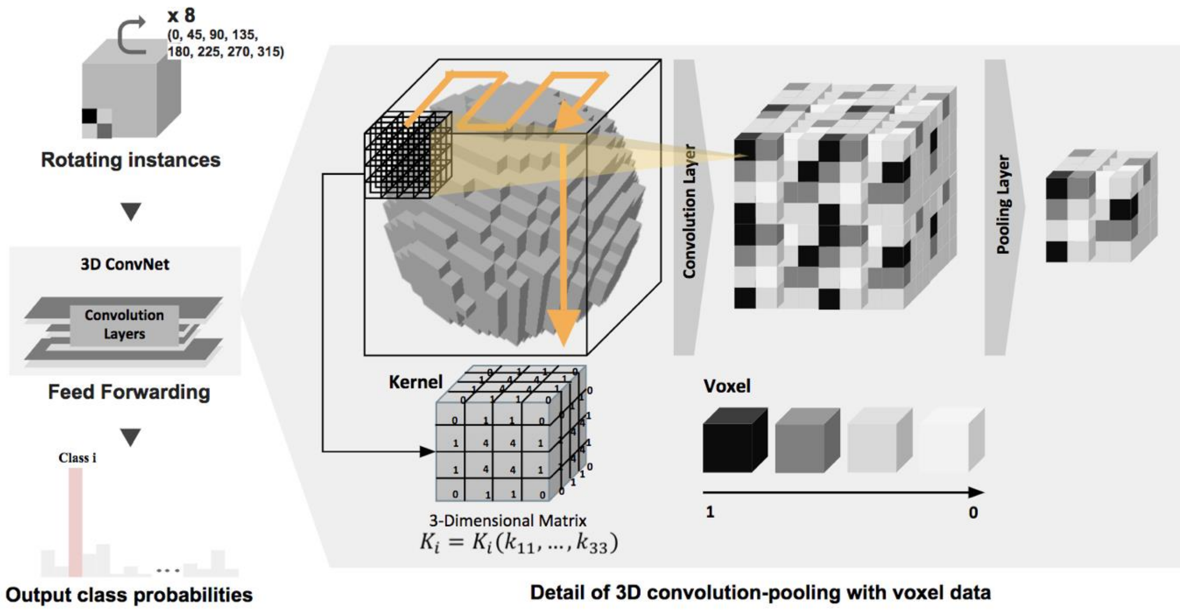


Fig. 3. 3D data classification using a 3D DNN.

2.2 Uncertainty of Deep Neural Networks and the Visualization of Deep Neural Networks

Deep neural networks are becoming more popular, and many attempts have been made to evaluate their performance. The performance evaluation of DNNs is not just about the accuracy that DNNs represent. Attempts have also been made to show the uncertainty of DNN classification results. 3D classification performance of DNN is not fully used in various fields including DNNs. Since they are not fully capable of judging, it is necessary to refer to the classification process before the general use of DNNs. The recent uncertainty of DNNs is being studied. There are studies that have addressed DNNs through quantification of DNN uncertainty [5, 15]. A study of the uncertainty of the deep neural networks through stochastic methodology [5] is one such representative study. A study that evaluates the performance of DNNs through stochastic methodologies is a quantitative study of the ambiguous concept of uncertainty, focusing on the possibility of DNN uncertainty through quantitative probability. However, the method has difficulty explaining the fundamental uncertainty of DNNs as a probability-based method. Therefore, a more intuitive method emerged than this stochastic approach. Deep neural network visualization methods [7-8,18] help to understand the entire network classification process, and the approach is studied in the paper intuitively.

Here, we will visualize the classification process for "Convnet using Voxel-Based Variational Autoencoders" [10], which is one of the most successful 3D classification models. In addition, visualization is an intuitive method of providing insight into the network classification process. Visualization was selected as a way of approaching uncertainty in the network. The model is visualized as a method that improves the limitations of existing visualization methods in a way optimized for 3D data.

3. Methods

The algorithm we designed in the study is a 3D DNN visualization algorithm. We initially want to find "why" the DNN classifies as such a good result. If we could see what had a positive or negative impact on the classification results (through the visualization method), we thought that we could indirectly provide insight into debugging. Therefore, the algorithm was developed to achieve network visualization, as shown in Fig. 2. Specifically, it consists of a naïve Bayesian-based compliment set model generation part and prediction difference analysis. Details are explained in chapters 3.1 and 3.2. The following environment for the above algorithm experiment was created.

Furthermore, for the computational efficiency of the algorithm, we need little specification for computing because the computation process requires complicated calculation. (The covariance matrix and the normal distribution for each number of complement sets must be calculated for each instance.) A computing machine was created with Ubuntu 16.04, an Intel Core i7-7700HQ 2.8 GHz processor, 32 GB 1600 MHz memory, and a GTX 1080ti VGA display.

The dataset used in the experiment was ModelNet 40 [5]. ModelNet 40 is the most commonly used 3D standard dataset. The DL library consists of Theano, a Python package. The

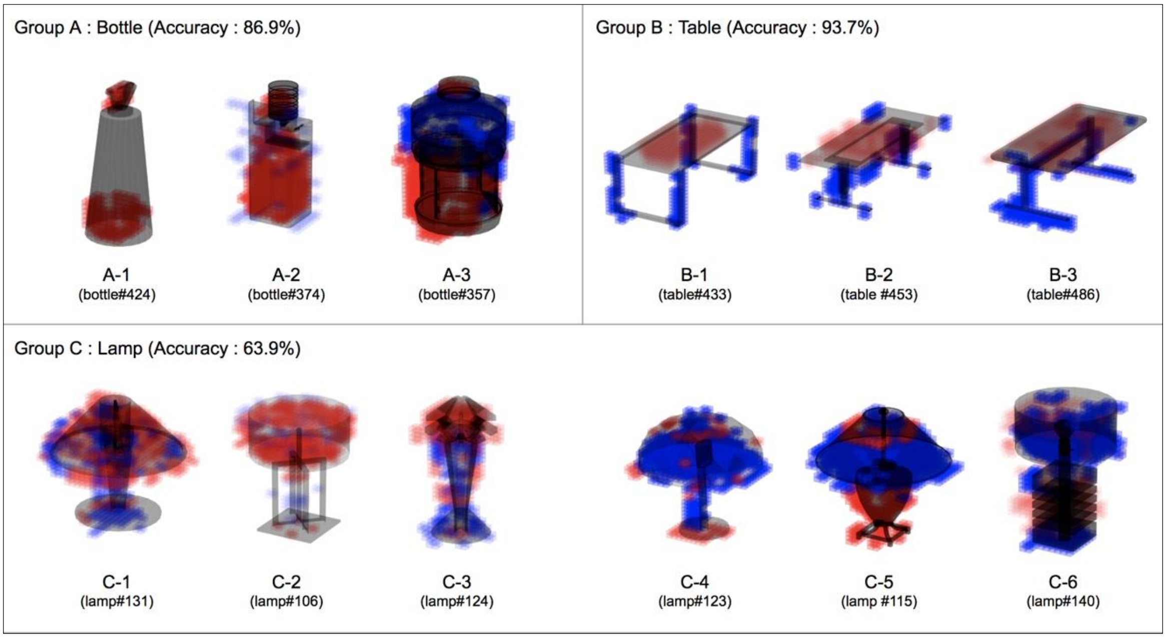


Fig. 4. Visualization of classification results of instances of specific classes (with ID on ShapeNet)

DNN model used Generative-and-Discriminative-Voxel-Modeling. The above model shows the highest classification performance at present.

Our method of representing 3D DNN visualization is based on the method proposed by Zintgraf et al. [7]. The method is based on an instance-based algorithm as well as the method proposed in the previous study because it is a measure of the intuitive network model uncertainty that we want to approach.

3.1 Naïve Bayesian 3D Complement Instance Generation Method

To investigate how much the certain part affects classification, we have to create the new object. The new object is designed so that the certain part does not affect classification. Therefore, the "naïve Bayesian 3D complement instance generation method" is the method that makes the certain parts unaffected. In addition, Fig. 2 illustrates how the individual parts in the input object are replaced by "Naïve Bayesian 3D Complement Instance Generation Method". Fig. 2 expressed the density of individual voxel for gray color, and the mixed colored parts indicate the replaced voxel that is calculated by the method.

Here, N dimensional data input instance $\boldsymbol{X} = (x_1, \dots, x_N)$ is the maximum posterior probability rule,

$$f(X) = argmax(f(\hat{y}|X))$$

and classifier f is the process of obtaining a reliable $X_{\setminus i}$ to input the input $X_{\setminus i}$ excluding a particular feature x_i.

We also used naïve Bayesian approaches [20] such as Robnik-Šikonja's saliency map [19] and Zintgraf's prediction difference map [7]. The first case proceeded with each feature as an independent variable, and the second case concluded that assigning each voxel feature as an independent variable did not reflect the characteristics of the visual data.

We set the grouped feature independent of the group using the patch/window concept and estimated it based on the correlation within the group. In the paper, we checked both the independence of the individual features and the independence of the grouped features. We agree in part with the approach of [7], which is dependent on a local patch and independent of each patch when considering parameters that are solid, hollow, etc.

The main procedure is shown in Fig. 2. First, a grouped voxel is called a 3-dimensional voxel set of $k \times k \times k$ $(k < N)$. We randomly sample 10000 grouped voxels in the model instances and set the multivariate normal distribution based on the prior probability by finding the mean matrix and covariance matrix between each feature in the grouped voxel. The internal feature values of each grouped voxel location are calculated through the computed multivariate normal distribution. We create a naïve Bayesian 3D complement instance by creating a union with the grouped voxel set and the 3D complement instance set from which the voxel of the position is removed. And since the results differ depending on the input angle of the 3D DNN, the "Window Voxel" will vary depending on the rotation angle.

This is the process of creating the input instance X by inserting the most common feature and not affecting the feature.

3.2 Uncertainty of deep neural networks and the visualization of deep neural networks

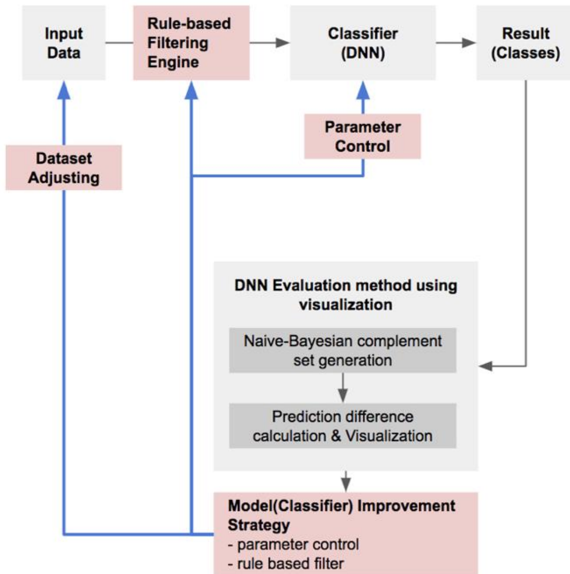


Fig. 5. Performance improvement strategy and optimization process using DNN evaluation techniques based on visualization method

With the output of section 3.1, we can see how much a particular part affects the classification. It can also be analyzed by expressing it as 3D visualization data.

The analysis is performed by *the prediction difference analysis* approach [7] by Zintgraf et al. To determine the effect of the feature x_i of the corresponding instance on the classification of the class, $f(X_{\setminus i})$ was calculated in 3.1, and $f(X_{\setminus i})$ was calculated by using the 3D convolutional neural network (see Fig. 3). Fig. 3 shows the classification process with 3D DNN. (The classifier (classifying function) f is a 3D convolutional neural network in Fig. 3, a certain type of 3D DNN)

The procedure is performed for the entire i $(0 \leq i \leq N)$ to calculate the difference between $f(X)$ and $f(X_{\setminus i})$ for each i and to confirm the classification effect when there is no corresponding part. To calculate the effect of each part as a negative and positive, we used the metric of weight evidence (from negative to positive numbers through the odds ratio difference). By expressing positive and negative in red and blue in the volume model of input, it is possible to observe the impact on the classification.

In addition, since it is an analysis process based on voxel and instance, a clear influence is visually expressed by overlapping with the input model to provide insight.

4. Results and Discussion

The results of the visualization through the proposed algorithm showed a clear difference according to result class and instance class (Section 4.1). The authors also concluded that the analyzed results could provide insight to evaluate or improve the performance of the DNN model (Section 4.2).

4.1 Differences in Visualization Results by Class and Instance

Visualization results are different according to (1) class and (2) class instance (See Fig. 4).

(1) Class level comparison: Classes with a high accuracy of classification results show that there are common features.

As shown in Fig. 4, classes with high classification performance (ex. bottle, table) and classes with low classification performance (ex. lamp) showed significant differences (the presence of common features that have a major impact on the classification).

Examples of visualization results in Bottle and Table classes are clear. In the case of Bottle, if one observes A-1, A-2, A-3, it is possible to see that the cylindrical and upper proboscis at the bottom of the bottle are the main classification features. A cylindrical section that is not identical in the area can also be found to have a more negative effect. (A-3) In the case of tables, the wide, thin top of the table should appear positive when looking at B-1, B-2, and B-3. The leg part is negative, and we assume it is a feature that is common to the desk class.

(2) Instance level comparison (within class): Classes characterized by low accuracy in the classification results showed that major common features are not detected.

An example of a lamp class shows this clearly. In Fig. 4 - group C, 1, 2, and 3 are successfully classified instances, and 4, 5, and 6 are failed ones. From the results, the instances classified as lamps (C-1, C-2, C-3) reveal that the head section had a major impact. However, the instances (C-4, C-5, C-6) that failed to be classified as a lamp had the same head part but failed to detect.

The above results indicate that we can extract insights from visual data on why DNN is so good at classification and why it is not. Then, 4.2 will consider strategies on how the information can improve model performance.

4.2 Discussion: Providing insight into DNN model improvement indirectly

Strategies will be discussed to improve classification performance. The point is that the visualization result from our algorithm can provide indirect insight for model improvement (e.g., determine why certain classes show low classification performance)

The case noted in 4.1(2) is a case where the classification accuracy is low, and the instances have parts that could be common key features, but they cannot be detected. The C-4, C-5, C-6 groups in Fig. 4 clearly show that the lamp head is not detected. The proposed solution is an example strategy for the above case. The classification process can be improved with a rule-based engine using existing detection algorithms [9,12,16-17]. For example, an engine that detects only the head of a lamp can be used. Therefore, we expect to improve on 36.1% of the instances of the lamp class (with no head part detected).

The above example corresponds to the insertion of a rule-based filtering engine in Fig. 5. (Note that there have been many approaches that combine DNN, ML and a rule-based approach. [21-24] We can provide insight for more specific orientation so that we can experiment and debug.

The author thinks that the method can indirectly debug DNNs that cannot explicitly be debugged due to multidimensional multi-parameters, which are unlike those of other algorithms and can help improve the performance strategy. Fig. 5 shows how our methods can be applied in a DNN utilizing process. (Comparing with Fig. 1, we determine how it can be used in a "DNN" utilized classification process.) Through the visualization results, the data scientist intervenes and makes directional improvements based on results rather than random methods such as 1) modification of the model itself, 2) design of a rule-based engine or 3) the process is completed.

5. Conclusions

The authors began the research with the goal of "making 3D deep neural networks (indirectly) debuggable", and after completing the visualization method and analyzing the results, they provided information on the explicit result of "general non-debuggable DNN." The results can be used as an evaluation method of the debugging and indirect debugging methods.

Future work will proceed in the following two directions.

(1) A study on 3D DNN optimization (application) process using a visualization method: The proposed visualization methods and optimization methodology will be applied to a specific case. The present dataset of the autonomous mobile automaker and the dataset of the non-destructive inspection in the factory are under consideration. In addition, the judgment of the improvement direction will be conducted in the long term to judge the machine rather than the human.

(2) A study of DNN uncertainty: The visualization method is merely a visual representation of the classification performance status of the model, and it must fundamentally combine the creation of a robust model by lowering the debugging methodology and uncertainty. The relationship between visual data and uncertainty will be examined further.

Acknowledgment

This work would not have been possible without the support of Luisa Zintgraf.

Nomenclature

DNN : Deep Neural Networks

References

[1] L. Kong, C. Alberti, D. Andor, I. Bogatyy and D. Weiss, Dragnn: A transition-based framework for dynamically connected neural networks, *International Conference on Learning Representations 2017 conference (2017)*.

[2] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev and M. Collins, Globally normalized transition-based neural networks, In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, (2016) 2442–2452.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, *AAAI*, 4 (2017) 12.

[4] J. Salamon and J. P. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification, IEEE Signal Processing Letters 24.3, (2017) 279-283.

[5] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang and J. Xiao,3d shapenets: A deep representation for volumetric shapes, *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015) 1912-1920.

[6] Y. Gal and Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, *international conference on machine learning*, (2016) 1050-1059.

[7] L. M. Zintgraf, T. S. Cohen, T. Adel and M. Welling, Visualizing deep neural network decisions: Prediction difference analysis, International Conference on Learning Representations 2017 conference (2017).

[8] K. Simonyan, A. Vedaldi and A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, International Conference on Learning Representations 2014 conference (2014)

[9] C. R. Qi, H. Su, K. Mo and L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, *Proc. Computer Vision and Pattern Recognition IEEE*, 1(2) (2017) 4.

[10] A. Brock, T. Lim, J. M. Ritchie and N. Weston, Generative and discriminative voxel modeling with convolutional neural networks, Advances in Neural Information Processing Systems 29 (2016).

[11] D. Maturana and S. Sebastian, Voxnet: A 3d convolutional neural network for real-time object recognition, *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE* (2015).

[12] B. Shi, S. Bai, Z. Zhou and X. Bai, Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12), (2015) 2339-2343.

[13] S. K. Kim and M. Likhachev. Planning for grasp selection of partially occluded objects, *Robotics and Automation (ICRA), 2016 IEEE International Conference* (2016).

[14] A. Zeng, K. T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez and J. Xiao, Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge, *Robotics and Automation (ICRA), 2017 IEEE International Conference* (2017) 1386-1383

[15] P. Mcclure, N. Kriegeskorte. Representing inferential uncertainty in deep neural networks through sampling, International Conference on Learning Representations 2017 conference (2017).

[16] H. Su, S. Maji, E. Kalogerakis and E. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, *Proceedings of the IEEE international conference on computer vision* (2015) 945-953.

[17] K. Sfikas, T. Theoharis and I. Pratikakis, Exploiting the

PANORAMA representation for convolutional neural network classification and retrieval. *Eurographics Workshop on 3D Object Retrieval* (2017).

[18]M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, *Computer vision–ECCV 2014* (2014) 818–833.

[19]M. Robnik-Šikonja and I. Kononenko, Explaining classifications for individual instances, *IEEE Transactions on Knowledge and Data Engineering*, 20.5 (2008) 589-600.

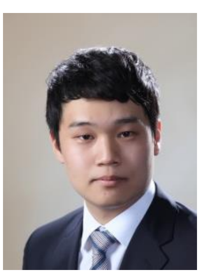
[20]K. P. Murphy, Naive bayes classifiers, *University of British Columbia 18* (2006).

[21]J. V. Román, S. C. Pérez, S. S. Lana and J. C. Cristóbal, Hybrid approach combining machine learning and a rule-based expert system for text categorization. *AAAI* (2011).

[22]H. Li. and P. E. Love Combining rule-based expert systems and artificial neural networks for mark-up estimation. *Construction Management & Economics*, *17*(2), (1999) 169-176.

[23]D. Le Hanh, K. K. Ahn, N. B. Kha, and W. K. Jo, Trajectory control of electro-hydraulic excavator using fuzzy self tuning algorithm with neural network. Journal of Mechanical Science and Technology, 23(1), (2009) 149-160.

[24]R. Jafari-Marandi, M. Khanzadeh, B. K. Smith and L. Bian, Self-Organizing and Error Driven (SOED) artificial neural network for smarter classifications. Journal of Computational Design and Engineering, 4(4), (2017) 282-304.



Hyun-Tae Hwang is currently studying his Ph.D. at Yonsei University in Seoul, Korea. He received his bachelor's degree in mechanical engineering from Yonsei University in 2013. His current research interests include PLM, Collaborative Design, Machine Learning.

Soo-Hong Lee is currently as a full- time professor at the Department of Mechanical Engineering, Yonsei University in Seoul, Korea. He received his bachelor's degree in mechanical engineering from Seoul National University in 1981 and his master's degree in mechanical engineering design from Seoul National University in 1983. He completed his Ph.D. from Stanford University, California, USA, in 1991. His current research interests include Intelligent CAD, Knowledge-based Engineering Design, Concurrent Engineering, Product Design Management, Product Lifecycle Management, Artificial Intelligence in Design, and Design Automation