Proceedings of the ASME 2020 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference

IDETC/CIE 2020

August 16–19, 2020, St. Louis, MO, USA

IDETC2020-22487

ORTHOGONAL DISTANCE FIELDS REPRESENTATION FOR MACHINE-LEARNING BASED MANUFACTURABILITY ANALYSIS

Aditya Balu, Sambit Ghadai, Soumik Sarkar, and Adarsh Krishnamurthy*
Department of Mechanical Engineering
Iowa State University
Ames, Iowa 50011

ABSTRACT

Computer-aided Design for Manufacturing (DFM) systems play an essential role in reducing the time taken for product development by providing manufacturability feedback to the designer before the manufacturing phase. Traditionally, DFM rules are hand-crafted and used to accelerate the engineering product design process by integrating manufacturability analysis during design. Recently, the feasibility of using a machine learning-based DFM tool in intelligently applying the DFM rules have been studied. These tools use a voxelized representation of the design and then use a 3D-Convolutional Neural Network (3D-CNN), to provide manufacturability feedback. Although these frameworks work effectively, there are some limitations to the voxelized representation of the design. In this paper, we introduce a new representation of the computer-aided design (CAD) model using orthogonal distance fields (ODF). We provide a GPU-accelerated algorithm to convert standard boundary representation (B-rep) CAD models into ODF representation. Using the ODF representation, we build a machine learning framework, similar to earlier approaches, to create a machine learning-based DFM system to provide manufacturability feedback. As proof of concept, we apply this framework to assess the manufacturability of drilled holes. The framework has an accuracy of more than 84% correctly classifying the manufacturable and non-manufacturable models using the new representation.

1 INTRODUCTION

The philosophy of Design for Manufacturing (DFM) is to provide manufacturability feedback to the designer before the manufacturing phase [1, 2, 3, 4]. Traditionally, designers have to go through several iterations of design and manufacturing reviews before final production. This decentralized approach increases product development time. DFM reduces the design lifecycle time by performing manufacturing checks during the design process. DFM rules are usually hand-crafted based on the manufacturing process, material, tool geometry, coolant use, and several other manufacturing process considerations [5, 6]. Although DFM rules accelerate the engineering product design process, such a practice relies on designer experience and training to discern and identify the rules while creating a complex component that involves several features [7]. Computer-aided DFM systems are useful in making this process easier for designers.

Computer-aided DFM analysis [3] can help in integrating the manufacturing experience into CAD systems. However, the main limitation of using hand-crafted rules for manufacturability analysis still exists. The proposition of making the next-generation CAD systems more robust and less dependent on hand-crafted rules has motivated several studies on creating automated design tools with cognitive capabilities for better human-machine interaction [4, 8]. They have built-in "*intelligence*" for analyzing the manufacturability in a practical and structured manner. Recently, machine learning-based DFM tools have shown great promise to provide such cognitive capabilities [9, 10].

Deep learning (DL) based prediction models are particularly suitable for building a cognitive CAD system due to their hierarchical feature learning capability without explicit hand-crafting. The hierarchical architecture of deep learning can be used to learn progressively complex features by capturing *features of features* [11]. Therefore, DL can be used to learn

*Corresponding author: adarsh@iastate.edu.

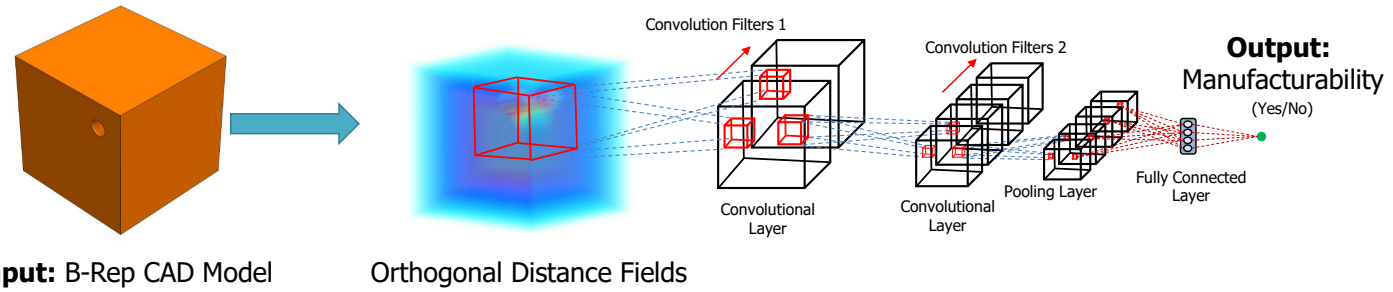

FIGURE 1. FRAMEWORK FOR DEEP-LEARNING BASED DESIGN FOR MANUFACTURABILITY. THE CAD MODEL IS CONVERTED INTO ORTHOGONAL DISTANCE FIELD REPRESENTATION AND IS INPUT TO THE 3D CONVOLUTIONAL NEURAL NETWORK (3D-CNN) FOR MANUFACTURABILITY CLASSIFICATION.

DFM rules from several examples of manufacturable and non-manufacturable components without explicit hand-crafting of the rules/features (Figure 1). Recently, deep learning has been gaining popularity within the engineering community, such as in prognostics [12, 13], engineering design and analysis [14, 15, 16, 17, 18], and robotic path planning [19]. In the context of manufacturing, several recent works explore the capability to learn features that are manufacturable [10]. A more detailed review of machine learning approaches in manufacturing can be found in [20, 21, 22, 23]. In particular, the DFM framework explored by [9] is very relevant to this work. To the best knowledge of the authors, [9] is the only work on DFM using 3D-convolutional neural networks (3D-CNNs). The proposed DFM prediction model can be integrated with CAD systems, providing interactive feedback about the manufacturability of the design. Finally, their framework can also identify the source of non-manufacturability in the design, which can then be suitably modified by the designer to make the overall design manufacturable. However, f the DL algorithms to identify *non-manufacturable* features, t CAD model of the component needs to be suitably represented

The CAD representation needs to be compatible with the h erarchical learning capability of DL algorithms to interpret t results suitably. Traditional mechanical CAD systems represe the geometry using boundary representation (B-rep). In a I rep, the geometry of a solid model is represented using only i boundary entities (for example, faces in 3D). However, there no information that corresponds directly to the volume of m terial contained within a solid model. Hence, it is challengir to build a DL framework that can learn spatial attributes fro a B-rep. Volume representations, on the other hand, are muc more suitable for DL algorithms, especially 3D-CNNs. Previous studies used a voxelized representation of the CAD model. In a voxelized representation, the entire model can be represented using a long string of binary digits that can be used as input for training the machine learning network.

However, the voxelized data is still limited in its capability to represent the solid model. The voxelized representation of the CAD model contains only digital information (*in-outs*) of whether the corresponding voxel is inside or outside the model. It does not have information about the proximity of a voxel to the boundary. In addition, as shown in Figure 2, the voxelized representation does not contain any information about the distance between features. However, the proximity of other features or the boundary might be important in determining the manufacturability of a part. Therefore, we introduce a new orthogonal distance field (ODF) representation for CAD models for use in machine learning applications in the context of manufacturability analysis.

In this paper, we introduce a new representation called orthogonal distance fields to represent the volumetric features of a CAD model. We then use it as input to a deep learning design for manufacturing (DLDFM) framework. We compare the ac-

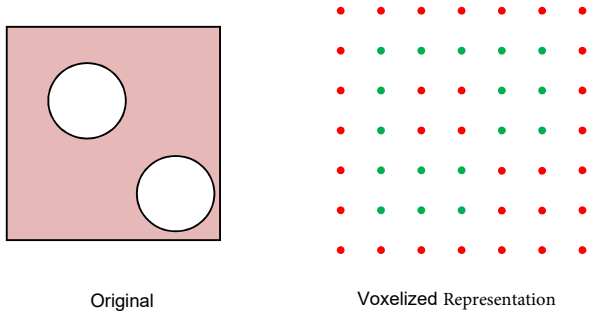

FIGURE 2. LIMITATIONS OF BINARY VOXELIZED APPROACH. THE CAD MODEL ON THE LEFT HAS 2 HOLES AND ITS VOXELIZED REPRESENTATION IS SHOWN ON THE RIGHT. THERE IS NO INFORMATION OF PROXIMITY OF THE HOLES TO THE BOUNDARY.

curacy of the ODF representation in identif
and non-manufacturable drilled holes with
approaches. The main contributions of this p

1. A GPU-accelerated method to convert E
orthogonal distance field representation
2. A deep-learning(3D-CNN) based DFM
lyze the manufacturability of drilled ho
try using the ODF representation.
3. Performance comparisons of the DFM
ing the ODF representation with voxeliz
CAD geometries.

2 ORTHOGONAL DISTANCE FIELDS

We represent the CAD model using a
distance fields to overcome the limitation of
tations. A survey on usage of numerous app
fields in computer vision, path planning, a
ization applications is presented in [24]. In
learning, distance fields provide additional information about the proximity of a voxel to the closest boundary, and information about the curvature and thickness of the geometry is also present in the distance fields [24].

Given a point $q \in \mathcal{S}$, a solid model that is bounded by $d\mathcal{S}$ in $\mathscr{R}^3$, the distance field D can be mathematically represented using the equation:

$$D = \min\left(dist(q,p)\right) | q \in \mathcal{S}, p \in d\mathcal{S}. \tag{1}$$

Even though there are efficient methods to convert the B-rep to distance fields [24], computing the accurate minimum distance values for all $q \in \mathcal{S}$ is computationally expensive. This problem is exacerbated in a machine learning framework, where thousands of models need to be converted to the distance field representation for training the DL network. Fortunately, the machine learning framework can still learn from approximate but consistent distance information. Hence, to accelerate the conversion from B-reps to distance representations for training, we develop a new approximate distance field representation called orthogonal distance fields. We compute the distance from each point in the solid model to the closest boundary in the six orthogonal directions (+x, -x, +y, -y, +z, -z), namely ODF. The ODF can be mathematically represented using the equation:

$$D_{ODF} = \min\left(dist(q,p)\right) | q \in \mathcal{S}, p \in d\mathcal{S}^q_{ODF}, \tag{2}$$

where $d\mathcal{S}^q_{ODF}$ is a subset of $d\mathcal{S}$. $d\mathcal{S}^q_{ODF}$ is the union of sets $d\mathcal{S}^q_x$, $d\mathcal{S}^q_y$, and $d\mathcal{S}^q_z$, given as:

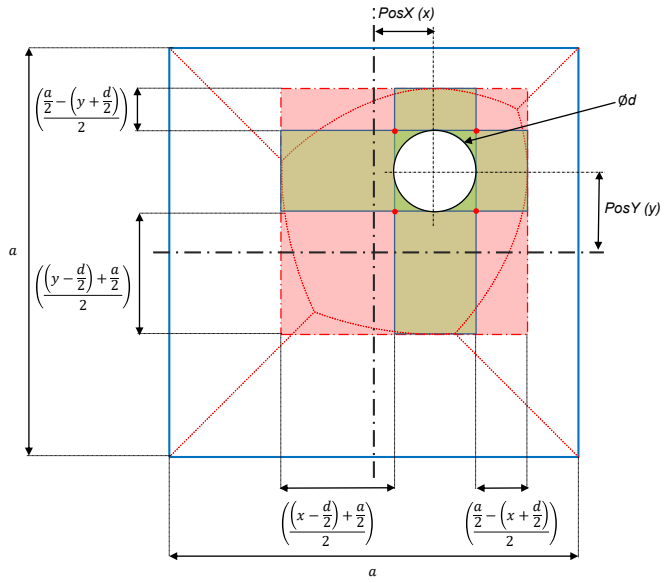

FIGURE 3. ORTHOGONAL DISTANCE FIELD FOR A SQUARE WITH A HOLLOW CIRCLE. THE FOUR RECTANGLES IN PINK COLOR REPRESENTS THE REGION WITH SIGNIFICANT APPROXIMATION.

$$d\mathcal{S}^q_x = \{p \,|\, \forall p \in d\mathcal{S}, p_y = q_y, p_z = q_z\}, \tag{3}$$
$$d\mathcal{S}^q_y = \{p \,|\, \forall p \in d\mathcal{S}, p_x = q_x, p_z = q_y\}, \tag{4}$$
$$d\mathcal{S}^q_z = \{p \,|\, \forall p \in d\mathcal{S}, p_x = q_x, p_y = q_y\}. \tag{5}$$

This computation is significantly less expensive and can be easily accelerated using the GPU.

The orthogonal distance fields are the same as the exact distance fields for simple geometries, such as a cuboid. However, the distances deviate significantly when the object includes any non-convex edges. For example, consider a square with a hollow circle inside, as shown in Figure 3. In this case, there are a few regions (marked in red) where the closest boundary to the point is the circle, but the orthogonal distance fields compute the minimum distance from the edge of the square. The area of this region is

$$A_I = \frac{(a-d)^2}{4}. \tag{6}$$

In addition, there are regions (marked in green) where the closest boundary is correctly identified, but the distance computed using orthogonal distance fields is different than the correct shortest distance. The maximum deviation between the orthogonal distance and the shortest distance occurs at the corners of the red

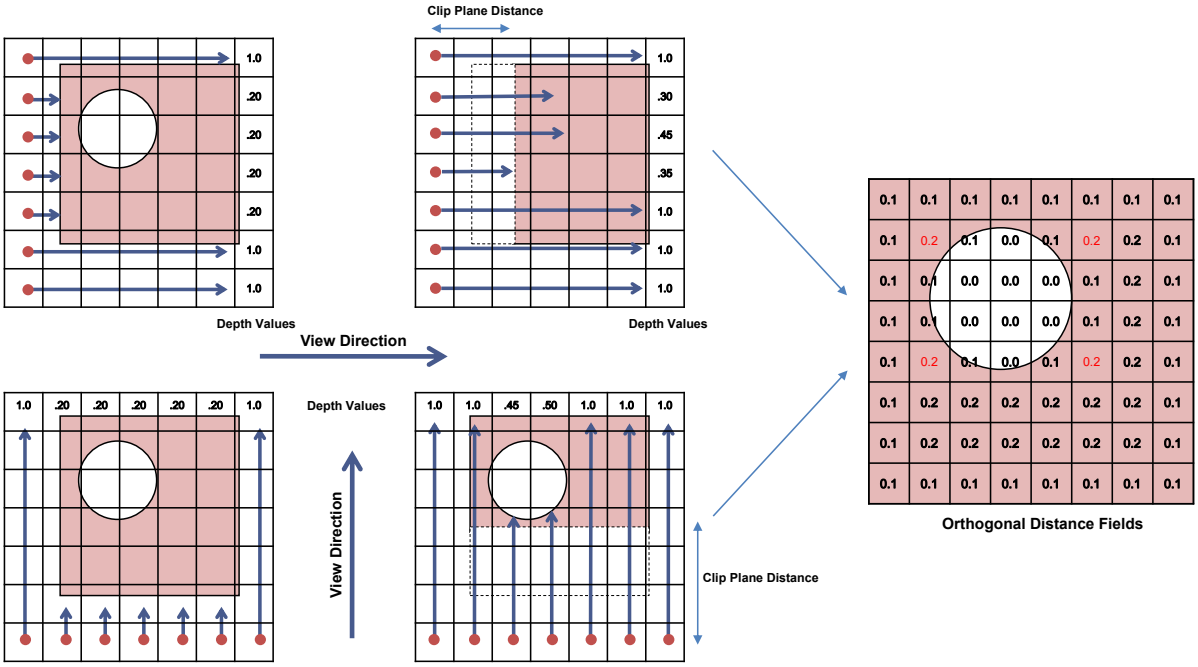

FIGURE 4. PERFORMING ORTHOGONAL DISTANCE FIELD COMPUTATION IN 2D USING GPU RENDERING. A CLIPPED CAD MODEL IS RENDERED SLICE-BY-SLICE AND THE DEPTH OF THE CLOSEST PIXEL FRAGMENT IS STORED. FINALLY, THE MINIMUM DEPTH VALUE FROM ALL THE FOUR ORTHOGONAL DIRECTIONS IN 2D (ONLY 2 SHOWN HERE) AT EACH PIXEL IS COMPUTED AS THE ORTHOGONAL DISTANCE FIELD.

rectangles (marked with a red dot). The area of this green region is

$$A_{II} = ad - \left(\frac{\pi}{4} + 1\right) d^2 \tag{7}$$

There is a C^0 discontinuity in the orthogonal distance fields at the intersection of the two (green and red) regions. This discontinuity can be smoothened in a discrete voxel mesh using Gaussian filters. In addition, they can be convolved with linear filters to obtain the correct distance fields [25]. A layer of the 3D-CNN has the capability to learn the weights and filters to perform this operation. Hence, an appropriately tuned 3D-CNN using orthogonal distance fields should have a similar performance to a different 3D-CNN that uses the correct distance fields.

2.1 GPU-Accelerated Algorithm to Compute ODF

We have developed methods for accelerated computation of volume representation of CAD models using graphics processing units (GPUs). These GPU-based methods are more than 10x faster than existing state-of-the-art CPU-based methods and can create a volumetric representation of a CAD model with more than 250,000,000 voxels. This high-resolution representation would have sufficient resolution to capture small features in CAD models. We construct a fine grid in the region occupied by the object using its axis-aligned bounding box. The B-rep model is then decomposed into its component surfaces, which is tessellated into triangles with a very fine resolution that is less than one-tenth of the resolution of the volumetric grid.

We use an approach similar to the voxelization approach as mentioned in Ghadai et al. [9]. The CAD model is rendered using orthographic projection to an off-screen framebuffer slice-by-slice by clipping it while rendering. The depth values for each pixel of this clipped model, stored in the depth buffer, is then used to get the orthogonal distance value for each pixel in the slice. After the clipped model has been rendered, the normalized depth values are read from the depth buffer (Figure 4).

The normalized depth values read from the depth buffer corresponds to the closest boundary along the specified orthogonal direction in which the model is rendered. However, this depth value is measured from the near plane of the projection. Hence, the depth values need to be adjusted with the distance of the model clipping plane from the near plane to get the distance to the closest orthogonal boundary. After rendering all the slices, the depth to the closest orthogonal boundary is stored in each voxel for this particular direction. Similarly, depth information for all the six directions are obtained. The orthogonal distance field is then obtained by taking the minimum of the six distance

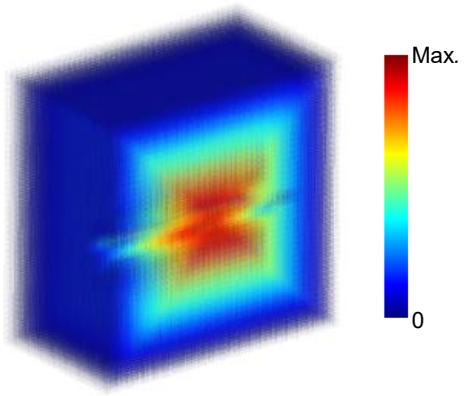

FIGURE 5. VOLUME RENDERING OF THE ORTHOGONAL DISTANCE FIELD OF A 3D CAD MODEL. THE COLOR REPRESENTS THE DISTANCE TO THE BOUNDARY. MOVING FURTHER INSIDE THE OBJECT REPRESENTS A LARGER DISTANCE TO THE BOUNDARY, AND THE COLOR CHANGES FROM BLUE TO RED.

values. The orthogonal distance fields for a simple cube with a hole rendered using volume rendering is shown in Figure 5.

The time taken to compute the distance fields is the sum of the time taken to evaluate each of the 6 directional distance fields. As an example, the total time taken to compute the orthogonal distance fields for a cubical block with hole is 0.177 seconds. These timings are obtained by running our orthogonal distance fields computation algorithm on an Intel Xeon CPU with 2.4 GHz processor, 64 GB RAM, and an NVIDIA Quadro K2200 GPU.

3 DESIGN FOR MANUFACTURING RULES

DFM guidelines were developed by design and manufacturing researchers to make design process compliant manufacturing [5]. In addition, researchers used these DFM guidelines to develop manufacturability analysis systems (MAS), which take into account the rules provided by DFM guidelines to analyze the manufacturability of a part [26]. Many MAS frameworks require the parameters of the part as additional user input, to analyze its manufacturability. These MAS frameworks then use the knowledge base of different DFM rules to provide manufacturability feedback. There are very few interactive parametric 3D solid modeling tools that provide manufacturability feedback to the designer and enable changes to the design for improving manufacturability [27]. A few studies on machine learning-based manufacturability feedback have also been performed [9, 10]. In this paper, we extend the work in [9] to demonstrate the performance of ODF representation. DFM rules for drilling have been developed based on the parameters of cylindrical geometry, as well as the geometry of the stock. Important geometric parameters are the diameter of the hole, the depth of hole, and the position of hole. Additional details of the implementation of DFM rules can be found in [9]. These rules are used to generate the ground truth manufacturability data for the training set, which is then used to learn manufacturable and non-manufacturable features by the DFM framework. However, it should be noted that for a DFM framework, one need not explicitly mention the rules. Instead, for industrial applications, one can train the DL model using industry-relevant historical data available in the organization, which need not be strictly rule-based. Historical data can also be based on experience during previous attempts to manufacture a part. Based on the DFM rules for drilling mentioned in [9], we generated 9531 CAD models in total for the training and validation set. Out of these, 75% of models were used for training the 3D-CNN, and remaining 25% of the models were used for validation or fine-tuning the hyper-parameters of the 3D-CNN. The dataset consists of CAD models of a primitive shape (cube) with a cylindrical hole of different depths and diameters present at various locations along one of the 6 faces of the cube as shown in Figure 6. The holes are perpendicular to the face of the cube, and both blind and through holes are considered. A detailed description of the training process is provided in Section 4. The trained DLDFM network is then tested using test-set CAD models generated separately consisting of 674 geometries.

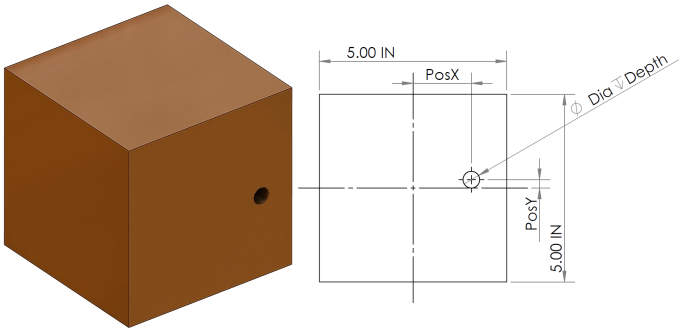

FIGURE 6. A SAMPLE BLOCK WITH A DRILLED HOLE WITH ITS DIMENSIONS IN THE ORTHOGRAPHIC VIEW.

4 3D-CNNs FOR DFM

Convolutional neural networks (CNNs) [28] are natural candidates to learn salient features in a hierarchical manner from volumetric representations of CAD models. Recently, 3D-Convolutional Neural Networks (3D-CNN) have been extensively used for several 3D object recognition [29, 30], human action recognition [31], imaging studies [32, 33, 34] and several other problems. Traditionally, the 2D-Convolutional Neural Networks have been used for object classification and image segmentation. The key distinction between 3D-CNN and its 2D counterpart is that the convolutional filters used are 3D.

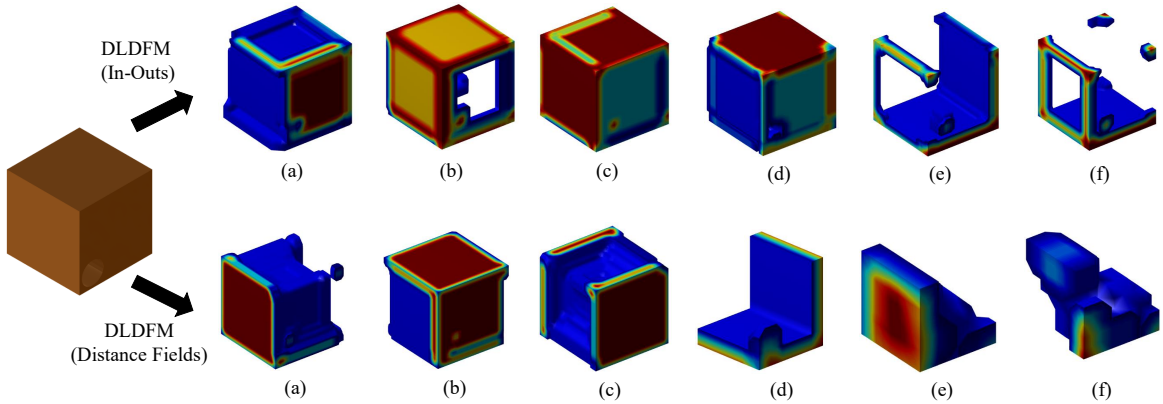

FIGURE 7. FEATURE MAP VISUALIZATION WITH RESPECT TO THE SECOND (a, b, v) AND THE THIRD (d, e, f) LAYERS OF THE 3D-CNN TRAINED USING *IN-OUTS* INFORMATION (TOP ROW) AND *ORTHOGONAL DISTANCE FIELDS* (BOTTOM ROW).

3D-CNNs consist of three different kinds of layers: convolutional, pooling, and fully-connected layers (Figure 1). Each convolutional layer in a 3D-CNN consists of different filters, which convolve with the input to generate a set of activation maps. The next convolutional layer takes the output activation map of the previous layer and performs convolutions with a new set of filters to get a new activation map. Thus, given an input volumetric representation, the filters are convolved in a hierarchical manner, which enables the 3D-CNN to recognize complex features. The first convolutional layer usually contains primitive information about the edges, corners etc of the model. The next layer can combine this primitive information to obtain more complex information about the features of the model. The pooling layer of the 3D-CNN performs a sub-sampling operation. Sub-sampling the volumetric representation can allow the 3D-CNN to focus on larger features, while ignoring smaller features that might not affect the manufacturability of the part. The fully-connected layer combines activations of all feature-of-features from the last convolutional or pooling layer to generate final decision.

Following Ghadai et al. [9], the manufacturability of a part is framed as a binary classification problem. The input to the 3D-CNN is a CAD model represented in a volumetric grid of size $64 \times 64 \times 64$. The data in this volumetric grid can either be the *Binary Inside-outside* or the *Orthogonal Distance Field* representation. The input volumetric data is first padded with zeros before convolution is performed. Zero padding is necessary in this case to ensure that the information about the boundary of the CAD model is not lost while performing convolution operation.

There are several hyper-parameters of the 3D-CNN to tune in order to ensure optimal learning. These include: number of layers, number of filters and filter size in each convolutional layer, and stride of the pooling layer. Specific approaches used to optimize the 3D-CNN are similar to those mentioned in [9].

5 RESULTS AND DISCUSSION

The generated B-rep CAD geometries are converted to volumetric representation using voxelization (as explained in [9]) and orthogonal distance field representation as explained in the Section 2. The grid size of $64 \times 64 \times 64$ is used for the volumetric representation. We train the 3D-CNN network using the voxelized representation and the orthogonal distance field representation of the CAD geometry.

5.1 Tuning of the Hyper-parameters

The hyper-parameters for the 3D-CNN are fine-tuned to have the least validation loss. The architecture of the 3D-CNN used is shown in Table 1. A batch size of 60 is selected while training the network. The training was performed using Keras [35] in a Python environment. The DLDFM network was trained in a workstation with a CPU RAM of 128GB, and a NVIDIA Titan X GPU with 12GB GPU RAM.

5.2 Visualization of the Features

The ability of the 3D-CNN to learn the features and hence, predict the manufacturability of a part can be understood by visualizing the feature maps. Figure 7 shows the output obtained from the second layer and third layer of the 3D-CNN. It can be seen that the 3D-CNN is able to recognize a few primitive information about the geometry; for example, the edges, the face, the hole in the cube, etc. Further, it can be observed that the next layer contains geometries of higher complexity.

5.3 Test Results

We first trained the 3D-CNN using the voxelized CAD geometries (*in-outs* information only). After successful training,

TABLE 1. HYPER-PARAMETER SELECTION FOR DIFFERENT 3D-CNNs.

DLDFM Network	Network Architecture
DLDFM with *in-outs* information	8 Convolution filters of size 8 Max. Pooling with subsampling size 2 8 Convolution filters of size 4 Max. Pooling with subsampling size 2 8 Convolution filters of size 2
DLDFM with *Orthogonal Distance Fields* information	8 Convolution filters of size 8 Max. Pooling with subsampling size 2 8 Convolution filters of size 6 Max. Pooling with subsampling size 2 8 Convolution filters of size 4 8 Convolution filters of size 2 Max. Pooling with subsampling size 2

TABLE 2. QUANTITATIVE PERFORMANCE ASSESSMENT OF THE 3D-CNN ON TEST DATASET.

Test Dataset	**Model Description**	**True Positive**	**True Negative**	**False Positive**	**False Negative**	**Accuracy**
674 models 408 Manufacturable 266 Non-Manufacturable	In-outs	391	90	17	176	0.7136
	In-outs + Surface Normals	334	201	74	65	0.7938
	Distance Fields	370	200	38	66	**0.8456**

the 3D-CNN was tested on the test dataset to understand its performance. The *in-outs* based 3D-CNN network performs well with an accuracy of 71% on 674 models of the representative test set (Table 2). Among the different examples, the scenario where the hole is close to the boundary seems to be very tricky, and the *in-outs* based 3D-CNN failed to predict them correctly. Note that this is one of the drawbacks of the *in-outs* representation of the CAD geometry. The information about the proximity of the voxels to the boundary is not available.

Hence, to overcome the drawback of *in-outs* based 3D-CNN, we had previously proposed [9] the use of surface normals to identify the boundary and provide better performance. This representation improves the performance of the 3D-CNN to an accuracy of 79%. Now, we use the proposed ODF representation of the geometries to train the 3D-CNN. The ODF representation increases the performance of the network to 84%, which is a significant improvement in results compared to *in-outs* and *in-outs along with surface normals* as proposed previously in [9]. Further, we also analyze the scenarios where the hole is close to the boundary. Using the ODF representation, the number of false-negative predictions are drastically reduced.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have developed an orthogonal distance field (ODF) representation in the context of deep-learning-based DFM framework. The 3D convolutional neural network (3D-CNN) used for this purpose has the capability to learn features directly from a volumetric representation of computer-aided design (CAD) geometries. Traditionally, voxelized models, without any additional shape information, are used. The 3D-CNN network developed using ODF achieves better accuracy in correctly classifying manufacturability than using only voxelized CAD geometry.

Future work could involve using the DFM framework for generative design and design modeling frameworks, and scaling the ML framework for other manufacturing processes using scale-variant features. Using the feedback provided by the 3D-CNN, an interactive decision-support system for DFM can be integrated with current CAD systems, which can provide real-time manufacturability analysis while a component is being designed. Such an AI-assisted manufacturability decision support framework would ultimately reduce design time, leading to significant cost-savings.

ACKNOWLEDGEMENTS

This work was partly supported by the National Science Foundation under grant number CMMI-1644441. We gratefully acknowledge the support of NVIDIA Corporation with the donation of a TITAN Xp GPU used for this research.

REFERENCES

[1] McMains, S., 2006. "Design for manufacturing feedback at interactive rates". In Proceedings of the Tenth ACM Symposium on Solid and Physical Modeling, p. 239.

[2] Zeng, Y., and Horváth, I., 2012. Fundamentals of next generation CAD/E systems.

[3] Gupta, S. K., Regli, W. C., Das, D., and Nau, D. S., 1997. "Automated manufacturability analysis: a survey". *Research in Engineering Design,* ***9***(3), pp. 168–190.

[4] Goel, A. K., Vattam, S., Wiltgen, B., and Helms, M., 2012. "Cognitive, collaborative, conceptual and creative - four characteristics of the next generation of knowledge-based CAD systems: a study in biologically inspired design". *Computer-Aided Design,* ***44***(10), pp. 879–900.

[5] Boothroyd, G., Dewhurst, P., and Knight, W., 2002. *Product Design for Manufacture and Assembly*. M. Dekker.

[6] Bralla, J. G., 1999. *Design for manufacturability handbook*. McGraw-Hill,.

[7] Yuan, J. T. J., Kong, K. Y., Parveen, H., Zhixiang, H., Rajasekaran, G., Behera, J. K., Sanaei, R., Otto, K. N., and HölttäYOttok, K., 2014. "An overview of design cognition between experts and novices". In Proceedings of International Conference on Advanced Design Research and Education (ICADRE14), pp. 156–160.

[8] Yin, Y. H., Nee, A. Y., Ong, S., Zhu, J. Y., Gu, P. H., and Chen, L. J., 2015. "Automating design with intelligent human–machine integration". *CIRP Annals,* ***64***(2), pp. 655–677.

[9] Ghadai, S., Balu, A., Sarkar, S., and Krishnamurthy, A., 2018. "Learning localized features in 3d cad models for manufacturability analysis of drilled holes". *Computer Aided Geometric Design,* ***62***, pp. 263–275.

[10] Zhang, Z., Jaiswal, P., and Rai, R., 2018. "Featurenet: machining feature recognition based on 3d convolution neural network". *Computer-Aided Design,* ***101***, pp. 12–22.

[11] LeCun, Y., Bengio, Y., and Hinton, G., 2015. "Deep learning". *Nature,* ***521***(7553), pp. 436–444.

[12] Liang, Y., Wu, D., Liu, G., Li, Y., Gao, C., Ma, Z. J., and Wu, W., 2016. "Big data-enabled multiscale serviceability analysis for aging bridges". *Digital Communications and Networks,* ***2***(3), pp. 97–107.

[13] Gangopadhyay, T., Locurto, A., Michael, J. B., and Sarkar, S., 2020. "Deep learning algorithms for detecting combustion instabilities". In *Dynamics and Control of Energy Systems*. Springer, pp. 283–300.

[14] Lore, K. G., Stoecklein, D., Davies, M., Ganapathysubramanian, B., and Sarkar, S., 2015. "Hierarchical feature extraction for efficient design of microfluidic flow patterns". In Proceedings of The 1st International Workshop on Feature Extraction: Modern Questions and Challenges, NIPS, pp. 213–225.

[15] Guo, X., Li, W., and Iorio, F., 2016. "Convolutional neural networks for steady flow approximation". In Proceedings of the ACM Knowledge, Discovery, and Data Mining Conference (KDD 2016), pp. 481–490.

[16] Lee, X. Y., Balu, A., Stoecklein, D., Ganapathysubramanian, B., and Sarkar, S., 2019. "A case study of deep reinforcement learning for engineering design: Application to microfluidic devices for flow sculpting". *Journal of Mechanical Design,* ***141***(11).

[17] Balu, A., Nallagonda, S., Xu, F., Krishnamurthy, A., Hsu, M.-C., and Sarkar, S., 2019. "A deep learning framework for design and analysis of surgical bioprosthetic heart valves". *Scientific Reports,* ***9***(1), pp. 1–12.

[18] Singh, R., Shah, V., Pokuri, B., Sarkar, S., Ganapathysubramanian, B., and Hegde, C., 2018. "Physics-aware deep generative models for creating synthetic microstructures". *arXiv preprint arXiv:1811.09669*.

[19] Lu, Y., Yi, S., Liu, Y., and Ji, Y., 2016. "A novel path planning method for biomimetic robot based on deep learning". *Assembly Automation,* ***36***(2), pp. 186–191.

[20] Bajic, B., Cosic, I., Lazarevic, M., Sremcev, N., and Rikalovic, A., 2018. "Machine learning techniques for smart manufacturing: Applications and challenges in industry 4.0". *Department of Industrial Engineering and Management Novi Sad, Serbia*, p. 29.

[21] Wang, J., Ma, Y., Zhang, L., Gao, R. X., and Wu, D., 2018. "Deep learning for smart manufacturing: Methods and applications". *Journal of Manufacturing Systems,* ***48***, pp. 144–156.

[22] Wuest, T., Weimer, D., Irgens, C., and Thoben, K.-D., 2016. "Machine learning in manufacturing: advantages, challenges, and applications". *Production & Manufacturing Research,* ***4***(1), pp. 23–45.

[23] Razvi, S. S., Feng, S., Narayanan, A., Lee, Y.-T. T., and Witherell, P., 2019. "A review of machine learning applications in additive manufacturing". In ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection.

[24] Jones, M. W., Baerentzen, J. A., and Sramek, M., 2006. "3d distance fields: A survey of techniques and applications". *IEEE Transactions on visualization and Computer Graphics,* ***12***(4), pp. 581–599.

[25] Sanchez, M., Fryazinov, O., Fayolle, P.-A., and Pasko, A., 2015. "Convolution filtering of continuous signed distance

fields for polygonal meshes". *Computer Graphics Forum,* ***34***(6), pp. 277–288.

[26] Shukor, S. A., and Axinte, D., 2009. "Manufacturability analysis system: Issues and future trends". *International Journal of Production Research,* ***47***(5), pp. 1369–1390.

[27] Shugrina, M., Shamir, A., and Matusik, W., 2015. "Fab forms: customizable objects for fabrication with validity and geometry caching". *ACM Transactions on Graphics,* ***34***(4), p. 100.

[28] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012. "Imagenet classification with deep convolutional neural networks". In Advances in Neural Information Processing Systems, pp. 1097–1105.

[29] Maturana, D., and Scherer, S., 2015. "VoxNet: A 3D convolutional neural network for real-time object recognition". In International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 922–928.

[30] Maturana, D., and Scherer, S., 2015. "3D convolutional neural networks for landing zone detection from LiDAR". In IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 3471–3478.

[31] Ji, S., Xu, W., Yang, M., and Yu, K., 2012. "3D convolutional neural networks for human action recognition". *IEEE transactions on pattern analysis and machine intelligence,* ***35***(1), pp. 221–231.

[32] Kamnitsas, K., Chen, L., Ledig, C., Rueckert, D., and Glocker, B., 2015. "Multi-scale 3D convolutional neural networks for lesion segmentation in brain mri". *Ischemic stroke lesion segmentation,* ***13***, p. 46.

[33] Kleesiek, J., Urban, G., Hubert, A., Schwarz, D., Maier-Hein, K., Bendszus, M., and Biller, A., 2016. "Deep MRI brain extraction: A 3D convolutional neural network for skull stripping". *NeuroImage,* ***129***, pp. 460–469.

[34] Payan, A., and Montana, G., 2015. "Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks". *arXiv preprint arXiv:1502.02506*.

[35] Chollet, F., 2015. Keras. `https://github.com/fchollet/keras`.