Journal of Computational Design and Engineering, 2023, **10**, 992–1009

DOI: 10.1093/jcde/qwad027
Advance access publication date: 4 April 2023
Research Article

Enhancing three-dimensional convolutional neural network-based geometric feature recognition for adaptive additive manufacturing: a signed distance field data approach

Arthur Hilbig*, Lucas Vogt, Stefan Holtzhausen and Kristin Paetzold

Chair of Virtual Product Development, Technische Universität Dresden, George-Bähr-Straße 3c, Dresden, Saxony, 01069, Germany
*Corresponding author. E-mail: arthur.hilbig@tu-dresden.de

Abstract

In the context of additive manufacturing, the adjustment of process data to individual geometric features offers the potential to further increase manufacturing speed and quality, while being widely underestimated in recent research. Unfortunately, the current non-uniform data handling in the CAD-CAM-Link results in a downstream data loss, that prevents the availability of geometric knowledge from being present at any time to apply the more advanced approaches of adaptive slicing and tool path generation. Automatic detection of various geometric entities would be beneficial for classifying partial surfaces and volumetric ranges to gain customized informational insights of geometric parameterization. In this work, an enhanced approach of geometric deep learning for the analysis of voxelized engineering parts will be presented to align the inference representations to modeling paradigms for complex design models like architected materials. Although the baseline voxel representation offers distinct advantages in detection accuracy, it comes with an adversely large memory footprint. The geometry discretization leads to high resolutions needed to capture various detail levels that prevent the analysis of fine-grained objects. To achieve efficient usage of three-dimensional (3D) deep learning techniques, we propose a 3D-convolutional neural network-based feature recognition approach using signed distance field data to limit the needed resolution. These implicit geometric data leverage the advantages of volumetric convolution while alleviating their disadvantages through the use of the continuous signed distance function. When analyzing computer-aided design data for geometric primitive features, a common application task in surface reconstruction of reverse engineering the proposed methodology, achieves a detection accuracy that is in line with the accuracy values achieved by comparable algorithms. This enables the recognition of fine-grained surface instances. The unambiguous shape information extracted could be used in subsequent adaptive slicing algorithms to achieve individual geometry-based hatch generation.

Keywords: additive manufacturing, feature recognition, deep learning, signed-distance field, voxel data, adaptive slicing

1. Introduction

From an industrial perspective, the advancing technology of additive manufacturing (AM) has revolutionized manufacturing procedures (Gao *et al.*, 2015). Because of the condensed manufacturing on a single machine, the simplification of design considerations for AM has led to apparent freedom in the design phase of computer-aided design (CAD). To deal with downsides like anisotropic material properties, shape deviations, or residual stresses, novel, and unresolved problems have occurred along the manufacturing planning workflow (Livesu *et al.*, 2017). Especially with the application of computer-aided manufacturing tools (CAM) in the product development pipeline, common procedures of AM (Fig. 1) are following various requirements that pursue different producibility objectives. Every substep in CAM for AM, which consists of tessellation, part orientation, support structure, slicing and machine tool path generation (TPG), is bound to the physical boundary conditions of the actually chosen AM technology.

The goal of having a ready-to-produce digital object at hand is driven by different main approaches: first, modifying the geometric design by topology optimization (van de Ven *et al.*, 2018); second, compensation of manufacturing process restrictions by part orientation and support structures (Langelaar, 2018); and third, optimizing the actual manufacturing data and machine instructions by slicing and TPG (Dolenc & Mäkelä, 1994; Jiang & Ma, 2020; Mohan Pandey *et al.*, 2003). Only the latter two promise design freedom and form fidelity by maxing out the limits of the AM manufacturing technology constraints.

1.1. Adaptive slicing for AM of architected materials

Since specific design characterizations are often missing, the generation of tool paths and slicing heights of a part model in two-dimensional (2D) layers is a multi-criteria consideration. The optimal parametrization is frequently determined by experienced expert knowledge or if no know-how is available through a trial-

Received: September 13, 2022. **Revised:** March 8, 2023. **Accepted:** March 9, 2023

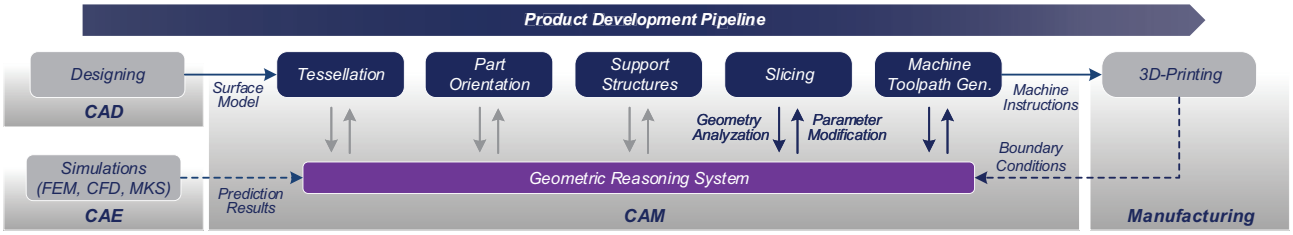

Figure 1: Extended product development pipeline in AM illustrating the integration of a geometric reasoning system in the CAD-CAM-Link.

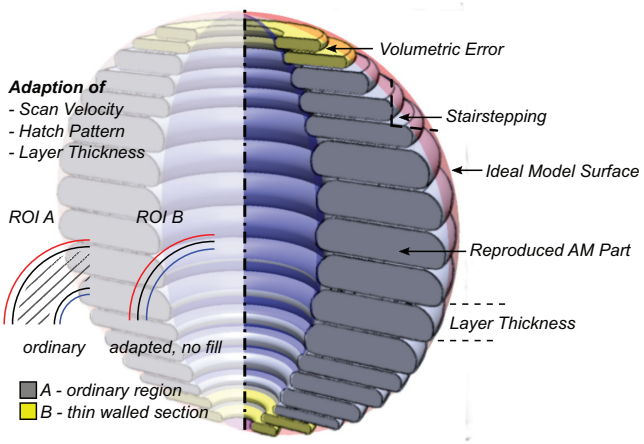

Figure 2: Exemplary adaptions of slicing and TPG of a design model in connection to machine instruction modifications for AM.

and-error process. This workflow is feasible for the mass production of similar parts, but the development of manufacturing-as-a-services by contract manufacturers demands customized components in small batch sizes. The approach of adaptive slicing (Mani *et al.*, 1999) offers opportunities to adjust to varying shape characteristics on a per-layer basis to reduce the stair-stepping effects (Fig. 2). By determining different machine instructions for each region of interest (ROI), diverse algorithms infer the optimized individual slicing heights for a given structure and adjust the layer parameters accordingly. The majority of the current algorithms rely on some geometry metric that either incorporates surface slope e.g., cusp angle/downskin angle (Cormier *et al.*, 2000), curvature radii (Hope *et al.*, 1997), or volumetric differences (Alexa *et al.*, 2017), which are bound to a global tolerated threshold. Especially region-based adaptive slicing and TPG are of great importance when printing architected materials like micro-structures, meta-materials, and scaffolds (see Fig. 3). Applying these architected materials is advantageous when creating lightweight structures by adapting stress and strain properties or optimizing heat transfer rates by a higher surface-to-volume ratio. Also in medical applications such as tissue-engineered bone scaffolds, architected materials facilitate tissue growth (Lehder *et al.*, 2021) by using triply periodic minimal surfaces (TPMSs) such as Gyroid (Fig. 3c) or Split P (Fig. 3d).

Given the geometric complexity of the architected materials, the design process has substantially changed. Since the additional geometric constraints provide and ensure technical feasibility, the often considered low-restricted manufacturing process in AM must also fulfill multiple supplementary requirements, which restricts the design freedom (Tamburrino *et al.*, 2018). This manifests in the inability to design architected materials with the common geometric modeling tools in engineering design. The boundary representation (B-Rep)-oriented modeling in CAD limits the generation of complex geometries because the geometric manipulation is inefficiently handled (Liu *et al.*, 2021; Tamburrino *et al.*, 2018). Also, B-Rep-based modeling kernels are numerically fragile dealing with model intersections of Boolean operations and imposed subsurface constraints (Hoffmann, 2001; Nguyen *et al.*, 2021). As a consequence, hybrid modeling techniques are on the rise using volumetric representations like voxel fields to design solid models (Gao *et al.*, 2015, Section 3).

Manufacturing structures with relatively high geometric complexity in a region-based manner relies on explicit geometry knowledge of the processed region and exact machining calibration. For example, in Korn *et al.* (2018) an adapted scan strategy and slicing for lattice structures in laser beam melting is presented. Since geometric details are near the scan resolution, the needed geometric specification has to be passed through all downstream processes of the design phase, to adapt the scan strategy to specific strut diameters of the lattice structure. But if the level of detail is not known beforehand, or the topology information is just not available, meaningful ROIs need to be derived automatically to adapt to thin-walled sections or acute angle corners. Common segmentation techniques like curvature-based regression are not applicable, because an unambiguous estimation of the threshold of continuous limits is difficult regarding the complexity of architected materials. In addition, multi-criteria decisions like material thickness, manufacturing resolution, and tolerances must be considered, so higher order recognition processes are needed at any point of the development pipeline to adapt slicing and TPG.

1.2. Advantages of geometric adaptivity

A causal purpose of all geometric adaptivity approaches remains the discovery of an optimal trade-off between fabrication time, structural stability, and surface quality with simultaneous shape fidelity of a producible part, called boundary conditions. Only in connection with a sufficient design knowledge of the geometric structures and their topology at any processing stage, a fully customized and optimal machine instruction parametrization can be implemented. To substitute manual expert analysis steps, a fully automated geometric reasoning system would be beneficial (Yang *et al.*, 2017), which analyzes printing parts in regard to geometric features and categorical classifications in regard to recognized ROIs. The optimal parameter settings specific to the given part model could be established by a structure-to-property linkage (Qi *et al.*, 2019; Wang *et al.*, 2020), which combines predictions results of simulations of different CAE tools, manufacturing procedures with their boundary conditions, and design rules by a unique ROI.

This study focuses on the geometry analyzing and recognition subsystem and presents an approach based on convolutional neural networks (CNN; LeCun *et al.*, 1989) used in deep learning (DL). The goal is to incorporate intrinsic properties of the geometric fea-

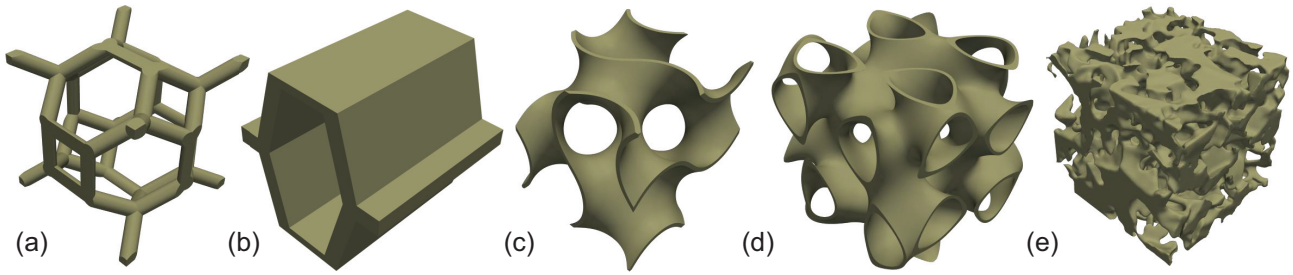

Figure 3: Architectured-materials types: (a) fluorite unit cell, (b) honey comb unit cell, (c) TPMS Gyroid, (d) TPMS Split P, and (e) Fraktale noise.

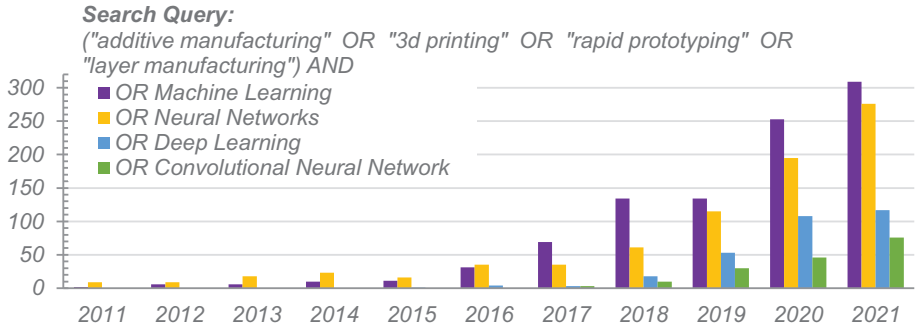

Figure 4: Number of articles of Titel, Keyword, Abstract search in Scopus (https://www.scopus.com) grouped by year of publication with varying search query terms for AM and ML.

tures, which compose the component they slice. These recognition results should reflect a superior understanding of the given geometric entities of a part model, with possibilities to distinguish the functional surfaces from negligible surface or volume areas. An adaptive slicing and TPG leveraging this understanding could not only optimize for global and overall manufacturing instructions but could also prioritize partial surfaces and volumes, which have an operational priority (Chen *et al.*, 2021) and recommend manufacturing feature designs (Yao *et al.*, 2017).

2. Related Work in Neural Geometry Analysis

The adaptation of machine learning (ML) techniques has attracted the interest of the AM research field for some time (Fig. 4), but the main application area was focused on parameter estimation and prediction of manufacturing properties (Qin *et al.*, 2022; Sarkon *et al.*, 2022). Due to the progressive recognition rates of general applicable artificial neural networks (ANN) and DL (Dong *et al.*, 2021) the field of AM is increasingly turning to the use of learned representations of geometric features. The finding of how the geometric data source needs to be processed is often not a straightforward selection of the DL network to choose (Guo *et al.*, 2020). Unstructured point clouds (Qi *et al.*, 2017) or free-form surfaces (Sharma *et al.*, 2020) require different network architectures than raster graphics or voxel fields in the volumetric domain. Because of the beneficial regular-grid-like structure that can be used directly, this study focuses on processing voxel fields by CNNs. The learned convolutional operation allows the extraction of abstract feature maps that can represent any semantic class (Fig. 5). The advantage is a simultaneous observation of global contextual characteristics of geometry and its local neighborhood relations (Garcia-Garcia *et al.*, 2017) at the same time. In addition, when applying CNNs, the geometric sampling densities of the analyzed geometry need less consideration in data pre-processing than for unstructured point cloud networks (Lang *et al.*, 2019; Takashima & Kanai, 2021; Wang *et al.*, 2022).

High-level feature representations can be generated by a multi-stage approach of repeated convolution and pooling layers, whereby the activation maps count represents the unfolded feature characteristics. To reduce the computational complexity and add transformation invariance to the trained network, pooling layers downsample the activation maps. The subsequent inverse-convolutional operation enlarges the compressed abstract feature maps to the input size, which allows for semantic segmentation tasks. Another approach to generate transformation invariance is the atrous convolution, also called diluted convolution. It addresses the issue of spatial downsampling of pooling layers by introducing a spacing between the processed values of the convolutional kernel at different rates (Chen *et al.*, 2017).

Initially developed for image data by Ronneberger *et al.* (2015), the processing manner of CNN semantic segmentation enables a pixel- or voxel-wise evaluation for concurrent classification, localization, and regression tasks. By transferring CNNs to a 3D approach (3D-CNN), "3D U-Net" (Çiçek *et al.*, 2016), and "V-Net" (Milletari *et al.*, 2016) are using voxels as volumetric input data. To gain more discriminative results with constant computational costs and the reduced number of trainable parameters, in Kamnitsas *et al.* (2017) and Dolz *et al.* (2017) dual pathway networks are proposed. By processing multiple input scales, a probability increase to recognize non-prominent structures is obtained.

CNN-based approaches have been applied in the field of AM to a diverse application set (Sarkon *et al.*, 2022): Several approaches address parameter estimations, forecasting (Silbernagel *et al.*, 2019), and build time estimation (Oh *et al.*, 2021). Thermal deviation tolerancing is conducted to compensate them in the design phase (Chowdhury & Anand, 2016; Zhu *et al.*, 2018, 2020). Also, CNN image segmentation is used to optimize parameter settings in man-

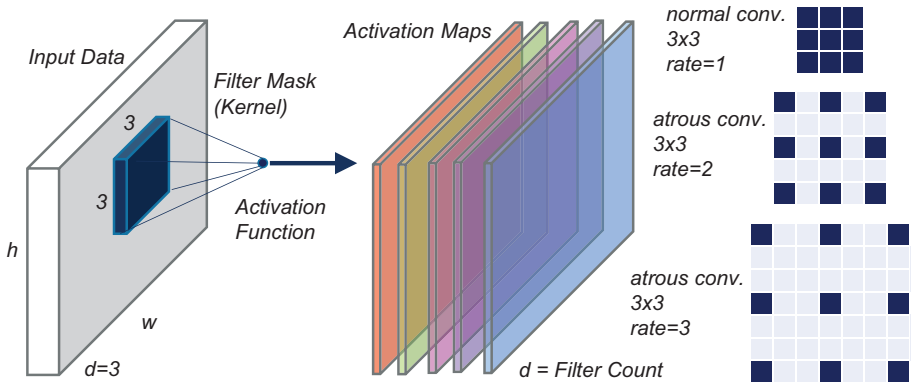

Figure 5: CNN layer of 3-channel image data with kernel size 3 × 3 and atrous convolution with varying rates. Larger atrous convolution rates enlarge the layer field of view.

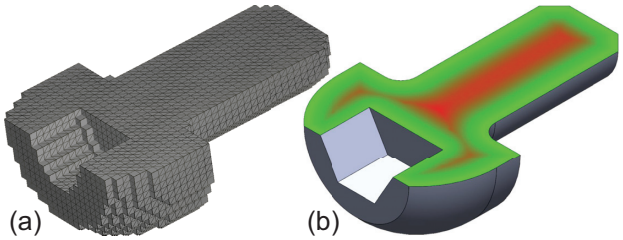

Figure 6: Cross-sectional (a) binary and (b) SDF representation of a bold. Signed distance values interpolated between nearest isosurface values and farthest values. Outer positive values of SDF are clipped.

ufacturing processes (Li *et al.*, 2021), to detect material defects on layer by layer basis (Wong *et al.*, 2020) or to perform stress distribution predictions by learning FEA simulations (Khadilkar *et al.*, 2019). Besides AM, rapid progress is made in computer-aided engineering by using CNNs in similar applications areas. Either machining feature recognition for CNC-manufacturing (Shi *et al.*, 2021; Zhang *et al.*, 2018), object classification (Dekhtiar *et al.*, 2018) or automated obtaining and generating of numerous modeling options in generative design (Oh *et al.*, 2019; Yoo *et al.*, 2021) are currently investigated and further developed.

3. A Signed-Distance Field Approach

With respect to the advanced recognition capabilities of DL and the changed design perspective with hybrid modeling, it is therefore proposed to base the recognition subsystem directly on a voxel representation and apply 3D-CNN architectures. Nevertheless, using the rasterized geometry characterization in the form of voxels, a common trade-off of all mentioned algorithms is the spatial aliasing effect. By the use of binary values (empty = 0, solid = 1), discrete density values (e.g., grayscales and floating-point values), or color descriptors, a low sampling resolution leads to indistinguishable and vanishing fine-grained details (Fig. 6a). The problem with this naive approach is its dependency on discretization accuracy, which leads to coarse surface representations. Recently these discretization inaccuracies in volumetric modeling for AM are be bypassed by using an implicit-based geometry representation, called signed distance field (SDF). SDFs are referred to as the most anti-aliased representation of geometric objects in volumetric grid processing (Jones *et al.*, 2006) and allow efficient constructive solid geometry operations. SDFs achieve subvoxel precision and thereby very smooth surfaces and volumes, through value interpolation (Bán & Valasek, 2020), storing the nearest surface distance in each voxel cell (Fig. 6b).

An SDF is calculated by evaluating the continuous signed distance function for all points p of the SDF. This function $dist(p)$ is defined as follows:

$$dist(p) = sign(p) \cdot min(|p - q|) \tag{1}$$

with

$$sign(p) = \begin{cases} -1 & \text{if } p \in \Omega \\ 1 & \text{else} \end{cases} \tag{2}$$

where q is a point on the surface of the body and Ω is the set of all points within the body. Consequently, the minimal distance to the nearest surface is assigned to every point in the considered volume. Points within a body are identified by their negative signed distance values, in contrast to the positive signed distance values of the points outside of a body. The exact volume boundary is therefore represented as the isosurface by a zero-crossing of the signed values. Especially in combination with level-set methods and octree tree data structures, SDFs allow for very efficient computing algorithms (Frisken *et al.*, 2000; Museth, 2013), where values only need to be calculated in the narrowband of the isosurface.

Implicit modeling by SDFs provides a toolset of high-level functions that facilitate the robust creation of complex structures for AM under varying material properties and global constraints (Biswas *et al.*, 2004). In Letov & Zhao (2022), a geometric modeling framework is proposed for irregular lattice structure. Also, implicit-based modeling is applied in contour representation in process planning of AM and topology optimization of cellular composites (Li *et al.*, 2016), porous microstructures (Sivapuram *et al.*, (2016), octet-truss lattices (Kambampati *et al.*, 2021; Nguyen *et al.*, 2022), or the design of structural optimized Voronoi struts (Cheng *et al.*, 2022). In the context of geometric analysis with DL, SDFs are mainly used as output representations of CNNs instead of using them as input values in supervised learning (Hou *et al.*, 2022; Liu *et al.*, 2022b; Park *et al.*, 2019). For example, in Fang *et al.* (2021) and Xue *et al.* (2019) modified "U-Net" architectures are used to infer 2D-SDF representations of a given organ from computed tomography. The ability of SDFs to generate smooth continuous surfaces is crucial for generating realistic representations of organic structures. A similar use case is the mesh reconstruction from points of 3D geometries from 2D images, where a 3D-SDF is generated through a single input image (Wang *et al.*, 2020; Xu *et al.*, 2019). Thus, the corresponding depth information for a 2D im-

age can be generated without additional depth sensor data (Lin *et al.*, 2020). Zhang *et al.* (2021) apply a learned SDF for multi-view-reconstruction. In Matveev *et al.* (2022), a distance-to-feature mapping is inferred on 3D point or image-based patches to estimate sharp geometric feature lines of components. Hou *et al.* (2022) proposes the inference on an SDF for classifying brain biomarkers (Yang *et al.*, 2021). In conclusion, this study is to the best of the author's knowledge the first one to utilize 3D-CNNs for analyzing geometries by input SDF values to infer higher level output properties of the analyzed design model for adaptive slicing and TPG in AM.

With this SDF approach, the DL geometric reasoning system could be grounded directly on the data representation that is increasingly popular to model architected materials, addressing the need for data-driven design strategies in AM (Zhang & Moon, 2021). In particular, the adaption and validation of the design and process parameters can be executed on the same SDF field model. The aforementioned structure-to-property would be explicitly defined when for example regular-grid simulation techniques like the lattice Boltzmann method are used. Therefore, the key objective of this study can be summarized as follows:

(i) The fundamental investigation of how SDF field values can be incorporated into the recognition subsystem by DL. Aiming to demonstrate whether beneficial recognition accuracies can be observed compared to the binary case for AM applications.
(ii) An investigation of different 3D-CNN network architectures and object scales, specifying the range of applications in AM, to align the field resolution to the downstream manufacturing resolution.
(iii) The definition of post-processing steps in part models to improve the noisy output results of the artificial network processing.

The actual investigation process is constrained by the condition of needing a learning setup that can be precisely validated. Since no commonly accepted segmentation dataset for architectural materials is available at the current stage of development, the proposed SDF-3D-CNN-based approach is oriented to a general reverse engineering (RE) task of surface reconstruction as an exemplary application. Thus, the chosen alternative learning task is defined as semantic segmentation of geometric primitives to prove whether the SDF approach can identify geometric features and various properties of local volumes with increased accuracy compared to the binary voxel approach.

4. Method and Implementation Details

Following the current state of CNN development shown in Section 2, three different baseline network architectures were implemented to investigate and further develop these essential approaches. All architectures are based on learning convolutional filters in 3D space, using 3 spatial dimensions (width, height, and depth) and the SDF distance or the binary value (solid = 1, void = −1) as the voxel element attribute.

The processed input SDF field section will be further referred to as the observation tile. A certain resolution of the observation tile is characterized by the edge length (EL) in voxels of the bounding volume. Common resolutions applied in 3D-CNN-architectures are typically 32 and 64 EL for full-voxel-based methods and up to 256 EL for resource-efficient architectures (Riegler *et al.*, 2016; Wang *et al.*, 2017). The promise of SDF values is the reduction of the required observation tile resolution, to receive only an EL of 16 or 32 voxels. These tile sizes allow for simultaneous processing of a 4096 or 32 768 voxels neighborhood, respectively, while limiting the needed memory consumption. If the tile resolution is sufficient for the recognition of fine-grained geometric primitives compared to the standard data approaches needs to be evaluated afterwards.

Based on the ideas of "3D U-Net" (Çiçek *et al.*, 2016), the first architecture (U-like), shown in Fig. 7a, represents the most straightforward approach to implement a neural segmentation of voxel fields. The input data are initially handled by two consecutive 64-filter convolutional layers, normalized through batch normalization, and processed by a leaky ReLU activation. After the first feature extraction, the data are compressed through max-pooling to the half tile size. At this stage, the feature extraction is repeated through two consecutive 128-filter convolutional layers to encode the input features into a distinct latent space. In the decoding section of U-like, the inverse convolutional operation enlarges the 8^3 feature maps back to a size of 32^3. By skip connections from the extraction stage, the symmetric encoder and decoder are combined by concatenation to regain spatial attributes, lost due to the pooling operation. After the following convolutional layers, the softmax layer transfers the results to a voxel-wise multi-class classification of the geometric primitives, which represents the semantic segmentation of the input features. The final inference can be derived by picking the geometric primitive with the highest probability for each individual voxel. Compared to the standard implementation, the number of trainable parameters is decreased, which allows for a much more efficient learning phase.

Being oriented to combine coarse and fine-grained information, while creating a larger receptive field by multi-scale processing, the second implemented architecture M-like is inspired by Kamnitsas *et al.* (2017) and Dolz *et al.* (2017). The major characteristic is the dual-path architecture (Fig. 7b) including different observation scales. A second path handles with a doubled EL an expanded area around the observation tile of the first path. By having a lower resolution, in which only every second voxel per tile axis is resolved, the computational cost of the eight times larger volume is limited. This implementation induces volume equality and simplifies the concatenation of the two paths. The residual and not mentioned parts of M-like are identical to U-like. In contrast to Kamnitsas *et al.* (2017) and Dolz *et al.* (2017) the need for a subsequent interpolation is prevented because the resulting output resolution corresponds to the observation tile resolution.

The third evaluated architecture (Fig. 8) is based on Wang & Lu (2018), using the atrous convolutional operation (Fig. 5). The features are encoded by a twofold spatial dense extraction (SDE) stage, consisting of stacked atrous residual blocks (ARB). The ARBs layers represent the major processing step for the multi-scale extraction of high-level semantics, reducing the computing complexity and generating specific activation maps. Then the relevant features are selected and converted into the segmentation of the given volume through the so-called attention feature aggregation block (AFA). Further details of the inner structure of the architecture can be found in Wang & Lu (2018).

Most components, which could be resolved in the engineering and AM context, require more than 16^3 voxels to analyze the entire geometry. Therefore a sliding window algorithm (Fig. 9) is used to stepwise scan the whole geometry with the trained CNNs. This enables the voxel resolution of the component model to be set to a specific size so that an observation tile always processes the same dimensions and allows the analysis of arbitrary large components while keeping the graphic processing unit loads

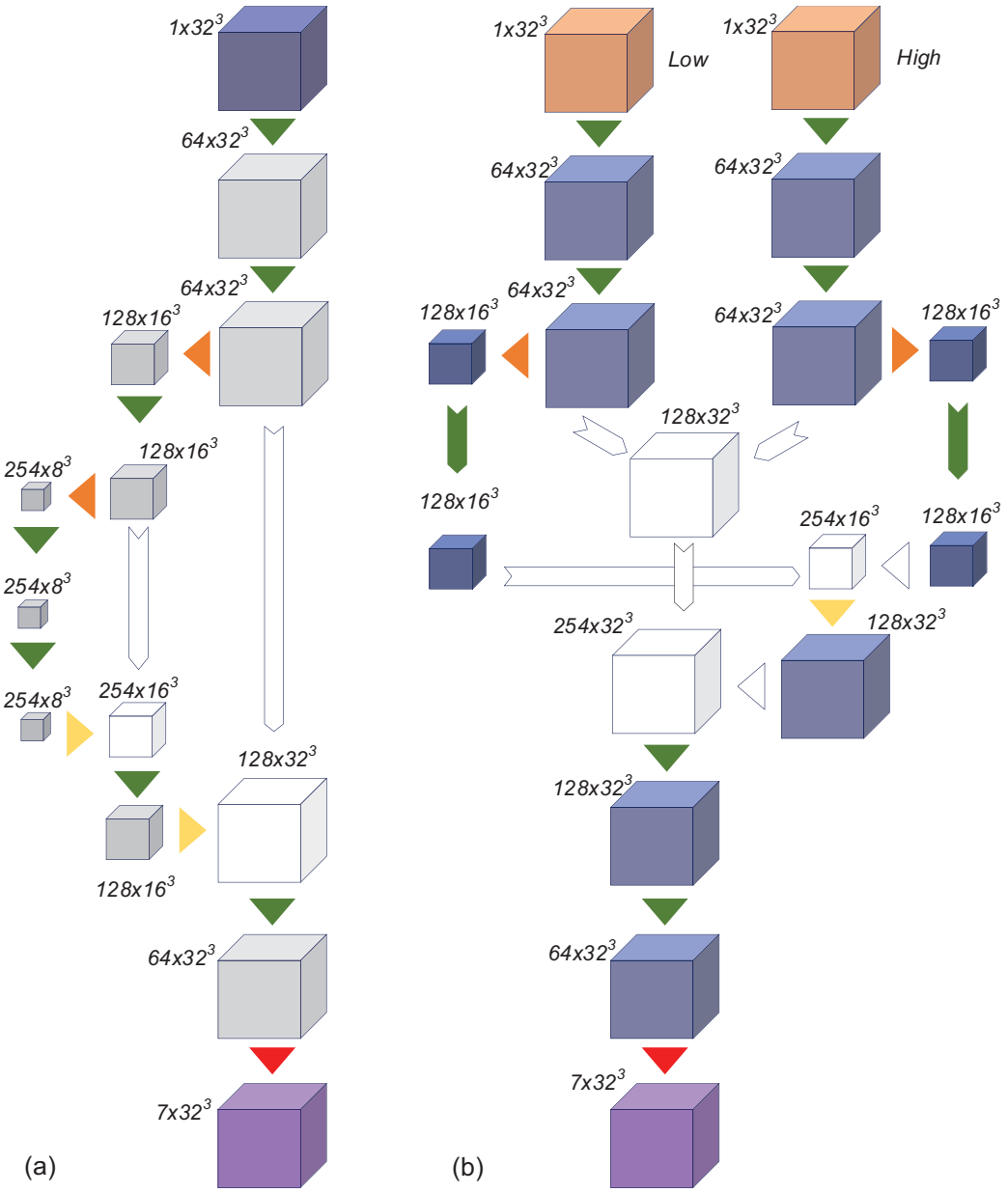

Figure 7: An overview over the (a) U-like and (b) M-like architectures: The input observation tile of size 32 shown in blue ■ cyan is passed through layers of convolution + batch normalization + leaky ReLU depicted in green ■, max-pooling in orange ■, skip connection in white □, and a softmax layer in red ■ to infer the segmented output tile in violet ■. All feature maps are annotated with their respective dimension.

small enough to still be used on average personal computers. Only the adjacent geometric structure and not the overall volume of the analyzed component should be necessary for this task.

A hierarchical structure of an octree (Fig. 9a) is used to pick out only windows that intersect with the SDF's isosurface. These individual windows are only processed in the narrowband of the isosurface and specify the center of the observation tile by the window corners (Fig. 9b). Because the padding mechanism in CNNs is leading to border effects (Alsallakh *et al.*, 2020), these are to be bypassed by an overlapping sliding windows approach by a factor of $2*\mu$ (Fig. 9c). Whereby, μ specifies the number of border voxels affected by incorrect detections and τ the resulting voxel size of the octree. After the learning phase, μ needs to be studied in more detail to obtain an optimal estimate.

5. Experiments for Primitive Type Detection

5.1. Dataset preparation and learning approach

Overall, seven different types of geometric primitives will be investigated in this study and included in the training dataset. These ruled primitive types are plane, sphere, cylinder, conus, and torus (Fig. 10), which are the basis of reconstructing design models of captured 3D-scans in RE. A comprehensive overview of the RE processing pipeline and applied methods can be found in Buonamici *et al.* (2018), while Berger *et al.* (2017) and Kaiser *et al.* (2019) provide a deeper understanding of surface reconstruction and primitive detection. In the field of DL, several primitive detection methods have been proposed (Li *et al.*, 2019; Sharma *et al.*, 2020; Yan *et al.*, 2021), which provide an extensive baseline for comparison.

The fundamental data type used in AM is mostly a triangular mesh representation. Out of triangular mesh, the SDF representa-

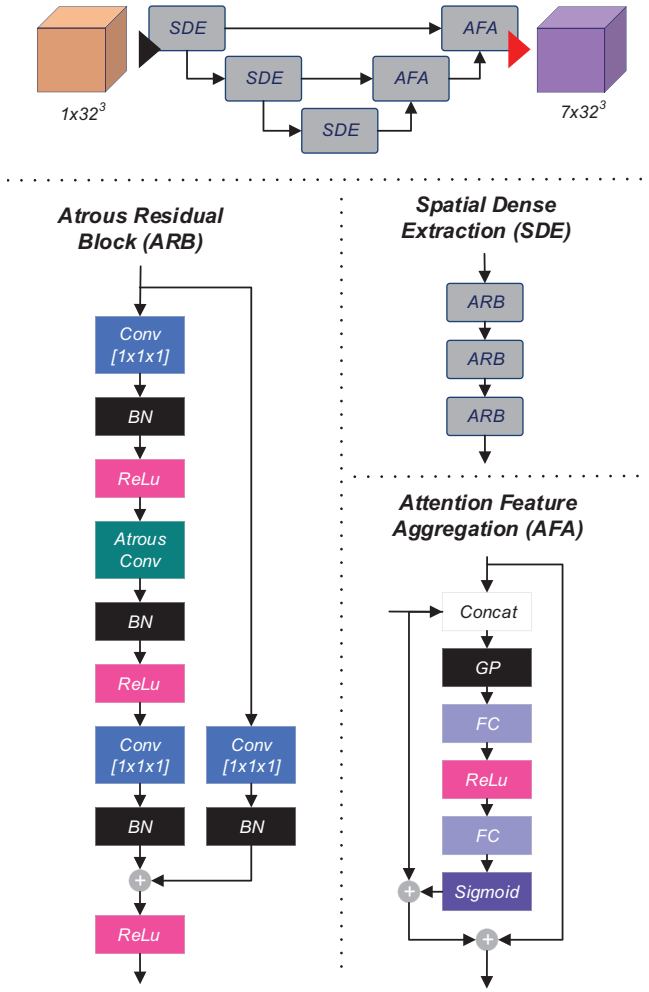

Figure 8: Schema of the A-like network (Wang & Lu, 2018) with the main building blocks (ARB, SDE, and AFA), which consist of different layers of, convolution is shown in blue ■, batch normalization in black ■, ReLU in pink ■, sigmoid in blue ■, atrous convolution in teal ■, and concatenation in white □.

tion is directly calculated by Museth (2013), allowing for compact and fast processing of sparse volumes. Next, the SDF representation is converted into a normalized form, usable for all implemented network structures, and adapted to the observation tile size. All values of an observation tile are scaled by dividing with the voxel size itself to ensure that SDF representation is independent of the chosen voxel resolution. Additionally, the SDF values above 1 or below −1 are clipped to 1 or −1, respectively, to simplify the representation (Figure 12). This standardizes the unimportant areas of all examples without altering the key information near the object's surface since all values between −1 and 1 remain unchanged.

Because the meshes can be of different origins, and their triangulation parameters and geometric structure differ significantly, it requires a learning database that is modifiable at will. Since there is no accessible data in the SDF or flexible mesh format available, we collect existing B-Rep models from the ABC-Dataset (Koch *et al.*, 2019) for a synthetic data approach. Synthetic data mean, that training data are generated in a virtual environment that allows for automatic annotation, vast modification possibilities, and exact input to ground-truth alignment. Each individually selected B-Rep model must contain all primitive types and models with additional geometric classes, such as freeform surfaces, are eliminated. To guarantee a diverse geometry set, all models with the same file size except one are removed, resulting in a database of 12k models.

The per voxel SDF value and primitive class are directly derived from the annotated triangular mesh of the B-Rep. Boundary voxel values refer therefore unambiguous to one subsurface instance. Labels are then transferred to the one-hot representation to create the ground truth data. Only the nearest interior voxel to the isosurface are labeled, the remaining interior and exterior voxels belong to respective void class (Fig. 11).

Even if the B-Rep dataset enables a fast generation of various examples, it is not able to catch all possible transformations which could occur. To compensate for this downside, data augmentations as well as transfer learning are used. Transfer learning allows to adapt and fine-tune of a pre-trained ANN to a new task or domain while using and adjusting the features learned from the original domain (Goodfellow *et al.*, 2018; Peng *et al.*, 2015; Wang & Deng, 2018). Even if the learning setup does not correspond exactly to the transfer learning term definition, it will be used to indicate that the trained weights of the basic stage serve as the initial weights of the subsequent transfer learning stage. Thus, primary geometric features of the primitives will be learned in a basic training stage and more advanced features such as the neighborhood relations will be handled in the transfer learning stage to adapt to the real-world domain. During the basic training stage, large amounts of easily generated synthetic data can be used to teach the baseline architectures the fundamental characteristics of geometric primitives. In the data augmentation, random transformations like rotations and mirroring are applied to the training samples.

A common drawback of all geometric discretization approaches, like the SDF approach, is the recognizable instance scale representing some feature class. Similar to a receptive field, the instance scale in the input tile evokes a stimulus in the input layers, which is sampled at a specific sampling rate. For example, a circle section corresponds to a line, if the value change is smaller than the sampling rate in the binary case. Therefore, no proper recognition result will be observed, and the undersampling leads to non-detectable features for too large sampling rates. For this reason, a consequence of the chosen observation tile resolution is the limitation of the detectable curvature, which could be overcome by the floating-point surface distances of an SDF. While in the field of medical imaging and computer vision this issue has been considered (Ibtehaz & Rahman, 2020), it still needs to be discovered beforehand how many measuring points are needed to represent a specific scale of a geometry class in SDF input data. For this reason, all B-Rep parts in the dataset are sampled at different sampling rates, ranging from 80 to 240 voxels in the most extensive space direction per model with an increment of 20 voxels.

In Table 1, the distribution of the geometric primitives over the used training, validation, and test datasets over the basic and transfer learning stage are shown. Most of the basic training dataset consists of samples only showing one geometric primitive at different angles and transformations. A smaller number of training examples consist of two merged primitives. These samples are necessary for a valid detection performance in transition areas between multiple primitives.

As different training and validation sets are applied to the basic and the transfer learning stage, both training stages share the same test set of the transfer stage in order to monitor the general learning success in the real-world B-Rep domain. In addition, the

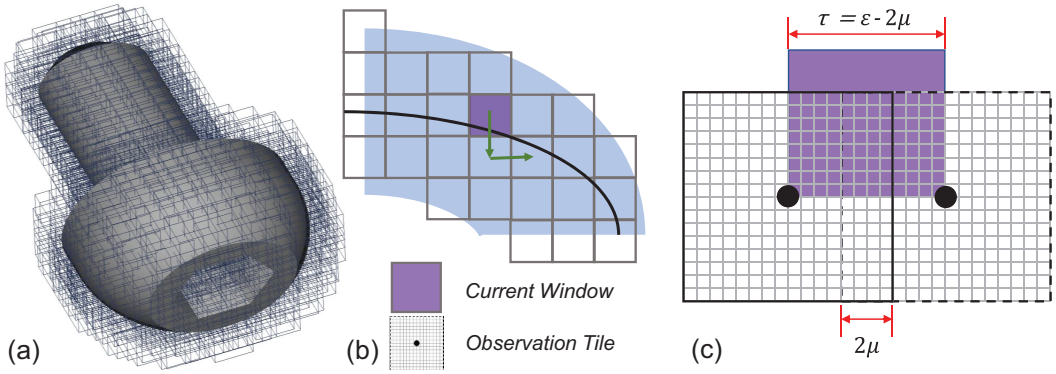

Figure 9: Sliding window approach for exhaustive part model segmentation. (a) Bounding boxes of hierarchical octree structure, (b) narrowband and isosurface of a subregion, and (c) sliding window overlap with processing parameters.

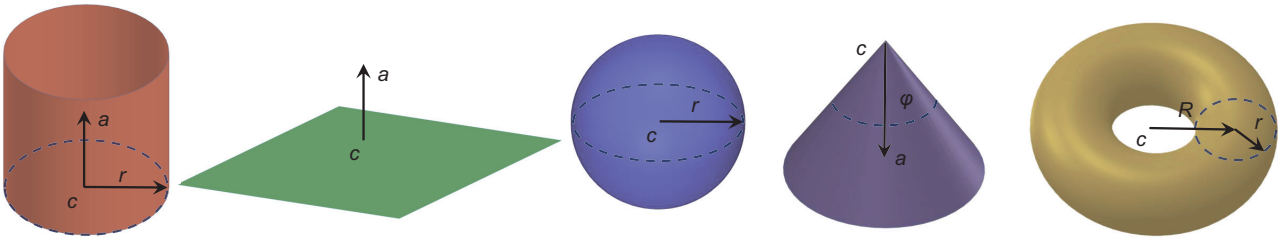

Figure 10: Primitive types and color coding in the learning dataset: Cylinder, plane, sphere, conus, and torus. Here, c is the center, r is the radius, a is the unit vector, R is the center radius, and φ is the half angle.

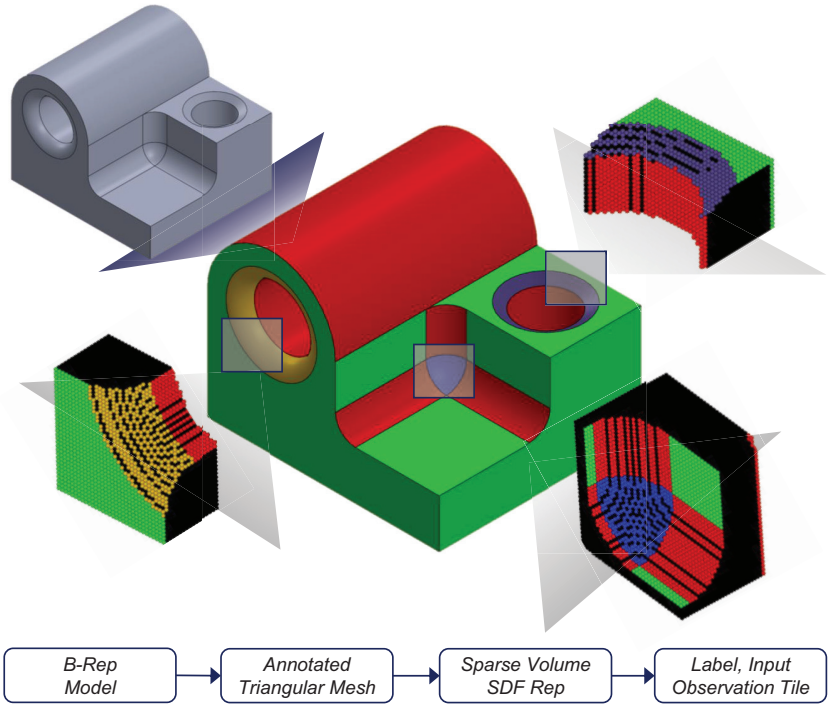

Figure 11: Synthesis process of the training dataset: From B-Rep part models to various observation tiles of different primitive composites. The interior void class is labeled in black.

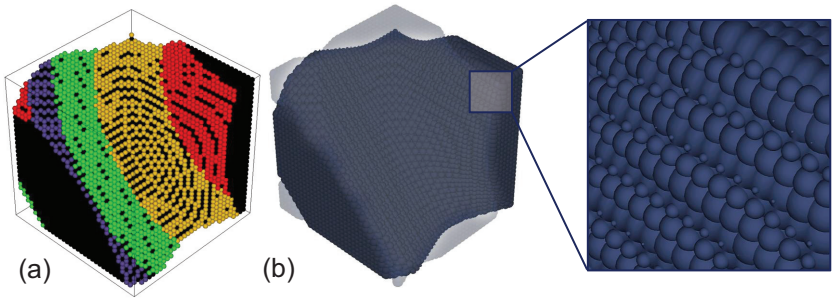

Figure 12: Observation tile preparation. (a) Colorized label tile by primitive type. (b) CNN input values depicted by spheres, whose radii represent the normalized and trimmed SDF values.

Table 1: Per voxel class distribution and overall tile instances percentage in the 32EL learning datasets. Class distribution of the basic datasets is balanced by undersampling.

	Basic (%)		Transfer (%)		
Type	**Train**	**Val**	**Train**	**Val**	**Test**
Plane	18.7	10.9	29.6	29.5	23.4
Cylinder	19.5	22.4	32.5	32.8	33.3
Conus	22.3	22.7	10.2	9.9	9.5
Sphere	18.7	20.6	13.8	13.7	13.3
Torus	20.8	23.2	13.9	14.1	14.3
Tiles (n)	66 981	11 300	34 732	3437	2963

basic stage is supplemented by the validation set of the transfer stage to be considered in the hyperparameter optimization.

5.2. Evaluation metrics

In the learning phase, various evaluation metrics are applied to measure and compare the recognition rates of the different network architectures regarding the similarity to the ground truth. True positives (TPs) and false positives (FPs) denote the number of predicted positives in the incorrect/correct case, while true negatives (TNs) and false negatives (FNs) account for the predicted negatives in the incorrect/correct case.

The intersection over union (IoU), also called Jaccard index, is utilized as an accuracy metric that is well suited because of its insensitive to variations in the frequency of occurrence for different labels (Taha & Hanbury, 2015).

$$mIoU = \frac{|S \bigcap G|}{|S \bigcup G|} = \frac{TP}{TP + FP + FN} \tag{3}$$

The similarity coefficient dice score (DS) is used to measure the overlapping regions of the predicted segmentation S and the ground truth segmentation G, and to apply a more outlier robust metric by the weighted average of precision and recall.

$$DS = \frac{2|S \bigcap G|}{|S| + |G|} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

Additionally, to investigate the exactness contour of the segmentation, the average Hausdorff distance (AHD, Aydin *et al.*, 2021; Rucklidge, 1996) in voxel size is used to measure the similarity of two point sets X and Y. By comparing the minimal Euclidean distance $d(x,y)$ of each boundary point, a sensitive outliers metric is introduced.

$$AHD(X, Y) = \left(\frac{1}{X} \sum_{x \in X} \min_{y \in Y} d(x, y) + \frac{1}{Y} \sum_{y \in Y} \min_{x \in X} d(x, y) \right) / 2 \tag{5}$$

The AHD will be further presented in voxel units with 0.0 as the optimum where the prediction corresponds to the ground truth.

5.3. Network training

To speed up the training process and to stabilize the gradient descent of the cross-entropy loss function of ADAM, the supervised learning was executed in batch training mode with a batch size of 4. For each architecture, an individual learning rate is determined by grid search optimization (*U-like* = 0.1, *M-like* = 0.1, and *A-like* = 0.0001 with default moment estimates of ADAM), with a per epoch learning rate decay of $e^{0.03}$ and a sparse categorical cross-entropy loss.

Considering the basic learning stage curve for U-like and A-like (Fig. 13), the SDF approach shows the overall ability to generalize a suitable solution for all different networks architectures. The training recognition accuracy reaches a mIoU of 90.1% for the *A-like*$_{SDF32EL}$ network after 150 epochs, with the simpler *U-like*$_{SDF32EL}$ just behind with a mIoU of 87.2% after 250 epochs. The basic validation mIoUs rank between 83.5% and 86.2% on the validation set. While A-like converges to a proper maximum, the apparent fluctuating stochastic behavior does not affect the overall recognition ability.

When tested on the transfer dataset, the capability of the trained basic networks exhibits a domain gap with a lower mIoU of 38.7% to 43.8%. It must be noted, that all networks in the basic stage are inaccurate in adapting to the diverse geometry characteristics of multi-class adjacent areas. Through the transfer learning stage, the test set overall accuracy is increased by roughly 40.0%, showing a mIoU of 82.5% to 88.4% for *A-like*$_{SDF32EL}$ and *U-like*$_{SDF32EL}$.

By comparing the resulting confusion matrices of *A-like*$_{SDF32EL}$ of the basic stage (Table 2) to the transfer learning stage (Table 3), the recognition rates of all distinguishable classes become apparent. This allows for a general assessment of the inter-class dependencies. These can be explainable by the geometric class itself. For example, since conical instances cannot be distinguished from a cylinder when their apex angle is small enough, most FN and FP can be observed at this class distinction. In addition, the class distinction between all cylindrical and conical types to the planar class displays the discretization error and the instance scaling problem, which denotes the fuzzy decision-making due to the resolution limits. Also, the conical primitive type reports the lowest TP of 39.0% compared to other primitive type's TP scores at both stages.

Compared to the basic stage, a TP increase of 54.5% to 88.0% of the spherical class can be accounted. While fundamental class boundaries emerge at multi-class adjacent areas, characteristic inter-class dependencies can be observed after the con-

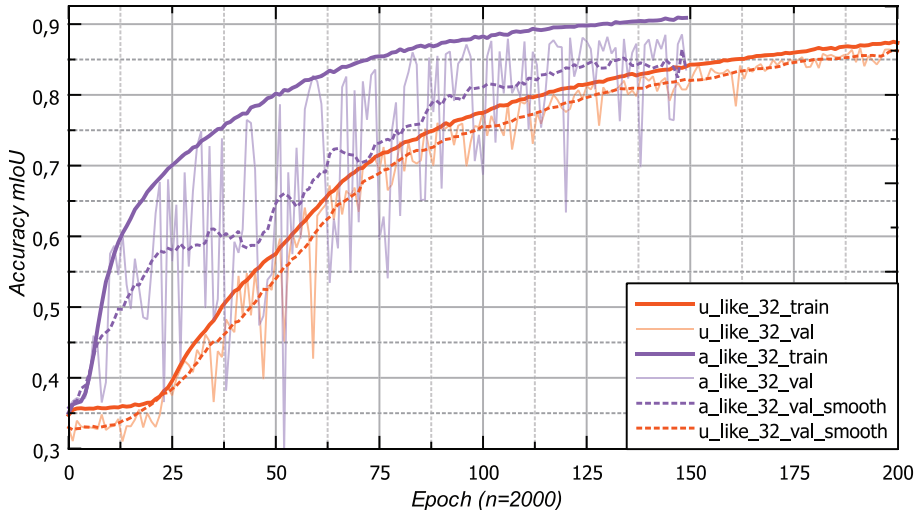

Figure 13: Resulting accuracy mIoU learning curves of U-like and A-like with 32EL in the basic stage. Validation accuracies are smoothed with moving average of 20 points.

Table 2: Per-class confusion matrix of $A\text{-}like_{SDF32EL}$ predictions on the test set after basic learning.

		Predicted label						
		Interior	Exterior	Plane	Cylinder	Conus	Sphere	Torus
True label	Interior	0.99	0.00	0.00	0.00	0.00	0.00	0.00
	Exterior	0.00	1.00	0.00	0.00	0.00	0.00	0.00
	Plane	0.00	0.04	0.57	0.28	0.05	0.04	0.01
	Cylinder	0.00	0.04	0.21	0.54	0.03	0.15	0.03
	Conus	0.01	0.04	0.35	0.17	0.39	0.00	0.04
	Sphere	0.00	0.03	0.08	0.33	0.00	0.57	0.00
	Torus	0.00	0.04	0.15	0.28	0.01	0.07	0.45

Table 3: Per-class confusion matrix of $A\text{-}like_{SDF32EL}$ predictions on the test set after transfer learning.

		Predicted label						
		Interior	Exterior	Plane	Cylinder	Conus	Sphere	Torus
True label	Interior	1.00	0.00	0.00	0.00	0.00	0.00	0.00
	Exterior	0.00	1.00	0.00	0.00	0.00	0.00	0.00
	Plane	0.001	0.01	0.96	0.01	0.00	0.00	0.00
	Cylinder	0.01	0.01	0.02	0.88	0.02	0.01	0.06
	Conus	0.01	0.01	0.03	0.24	0.70	0.00	0.01
	Sphere	0.01	0.00	0.00	0.01	0.00	0.96	0.02
	Torus	0.01	0.01	0.00	0.07	0.00	0.02	0.89

ducted transfer learning. By considering the segmentation results in Fig. 14, it can be noted that FP and FN occur especially at the subsurface boundaries, and can thus be directly related to the simplicity of the basic validation set. This complexity mismatch is also the key difference in the data domains of the transfer and basic datasets, which could be compensated by further labor-intensive manipulation of the synthetic data. However, due to the increased accuracy on the test set after transfer learning, the actual goal of predicting a sufficient variety of geometric transformations was already achieved.

It can be stated in the binary case that no comparable segmentation results to the SDF approach could be achieved in the learning phase (Fig. 15). For all tile resolutions of *A-like* and *U-like* segmentation results around 62.0% to 69.0% mIoU are reached in the binary case. Also, the binary solution is much more outlier-prone with AHD of 4.01 compared to 1.5 for the SDF case of $U\text{-}like_{32EL}$.

5.4. Post-processing of predictions

To further improve the detection performance, several post-processing methods are introduced to remove uncertainties in the derived distributions on the one hand and to remove outliers of homogeneous surface regions on the other hand.

The first post-processing step is inspired by the work of Martinez *et al.* (2019), who analyzed the class distributions for each voxel over multiple inference runs with slightly modified CNNs to infer the certainty of the achieved segmentation. Therefore, in the progress of the post-processing pipeline (Fig. 16), uncer-

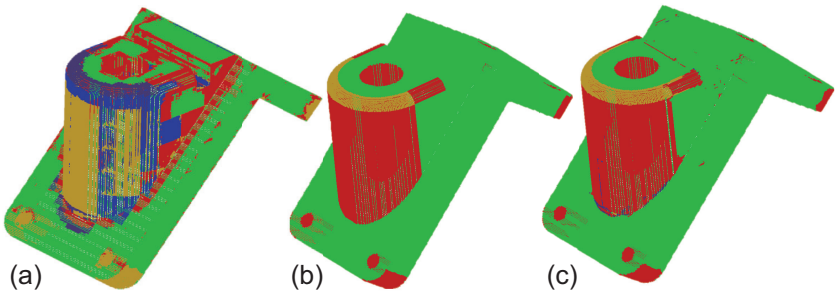

Figure 14: Comparison of segmentation results in multi-class adjacent areas of a part model with *U-like*$_{32EL}$ of (a) basic learning stage, (b) ground truth, and (c) transfer learning stage: Inaccurate region boundaries at class transitions in the basic stage evolve to unambiguous delimitation in the transfer stage.

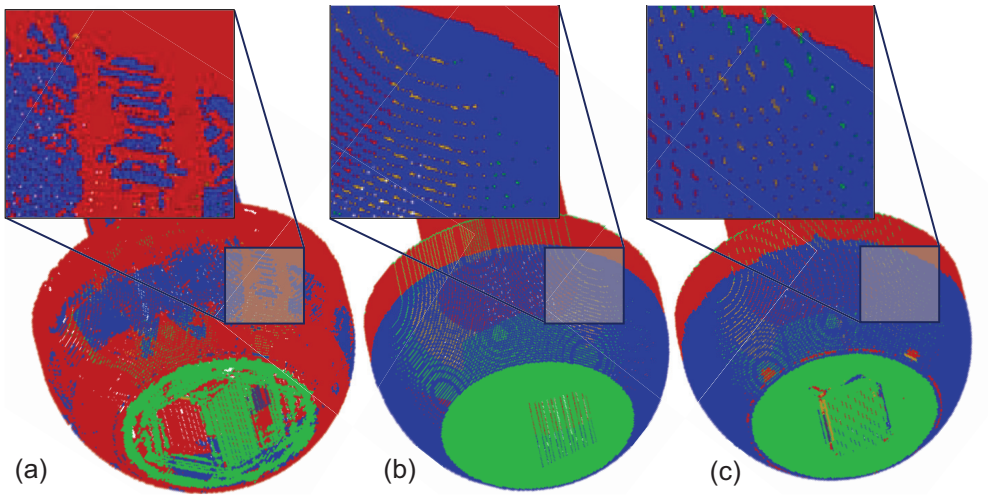

Figure 15: Comparison of semantic segmentation results of (a) binary ($mIoU = 0.3$), (b) ground truth, and (c) SDF ($mIoU = 0.8$) case of *A-like*$_{32EL}$ after the transfer stage. While the SDF approach forms distinct and fine-grained features, misclassifications and incorrect region boundaries can be detected for the binary approach.

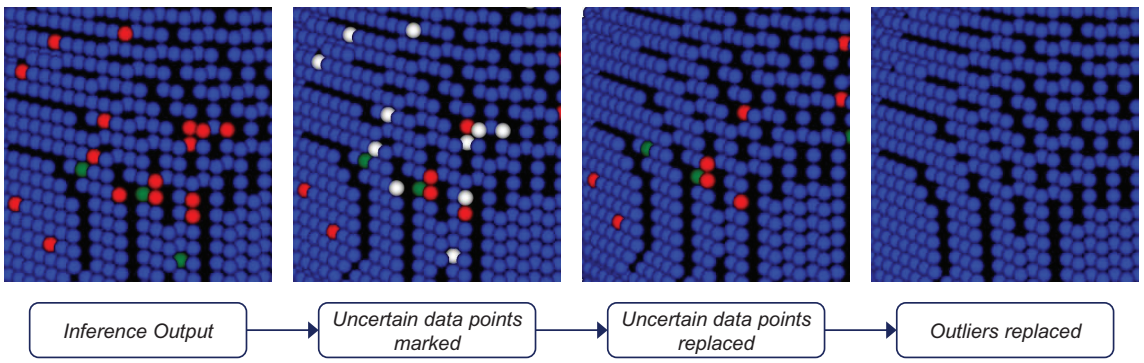

Figure 16: Post-processing steps to eliminate noisy segmentations by uncertainty processing and handling of prediction probabilities.

tain voxels are replaced with the most common label from the immediate neighborhood of this voxel. To achieve this, the probability distribution for the possible labels is analyzed on a per-voxel basis. For example, for *U-like*$_{64EL}$, a class probability in the semantic map of less than 80% implies a 95% chance of being misidentified, from which it can be concluded that these voxels can be considered as uncertain. In the next step, the neighborhood of a marked voxel is combined by the arithmetic mean of all values from the final softmax layer, and the most often occurring class in this area is assigned to the uncertain voxels. The underlying assumption for this step is that a voxel most likely belongs to the same geometric primitive as its immediate neighbors.

After the deployment of the first post-processing approach usually a few single outliers or small patches of incorrect classified voxels remain existent. Therefore, the second post-processing method replaces single outliers and small groups of labels within an otherwise consistent area of labels in the immediate neighborhood. This approach builds on the fact that continuous models of real engineering parts are analyzed, which do not consist of

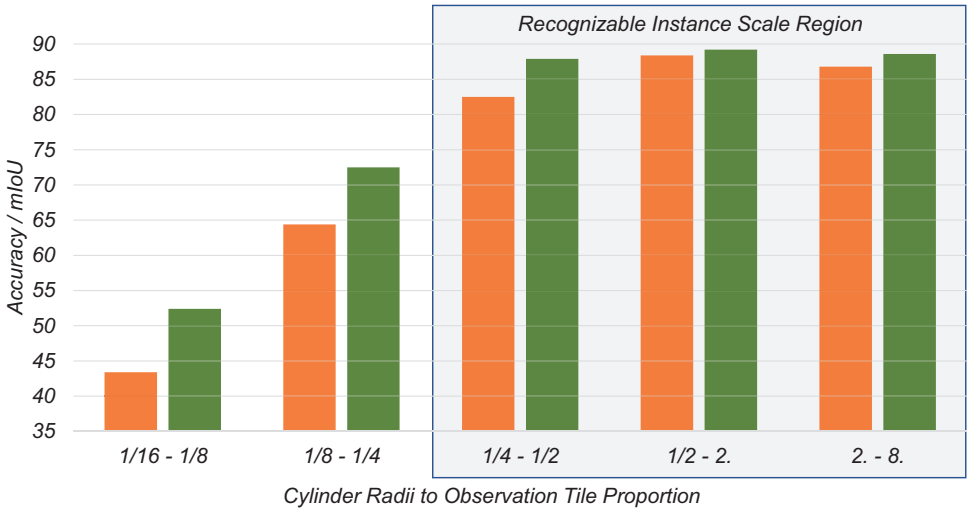

Figure 17: Evaluation of the detection mIoU accuracy over the curvature radii r in the ranges of $r = \{r \in \mathbb{R} : [\frac{1}{16}..8]\}$ of the SDF32EL networks of $U\text{-}like_{32EL}$ and $A\text{-}like_{32EL}$. Instance scales of a cylinder primitive under $\frac{1}{4}EL$ are not identifiable.

Table 4: Accuracy comparison of observation tile resolutions on the transfer test dataset and the trainable parameters count (TParams) of each network architecture.

Network	Repr.	*mIoU*(%)			TParams
U-like	Binary	62.3	64.4	69.2	12 774 167
	SDF	82.5	88.4	89.2	
M-like	Binary	38.9	37.4	41.0	2128 519
	SDF	43.4	43.8	51.2	
A-like	Binary	68.0	67.5	68.1	572 679
	SDF	88.0	86.1	84.5	
		16	**32 EL**	**64**	

point-like surfaces. Therefore, a single label or a very small patch of labels in an otherwise uniform classified surface must be inadequately evaluated. If a minimum patch count of three voxels is not reached, it is replaced by the most common label in the direct neighborhood of these voxels.

Another possible post-processing step to be mentioned is the use of conditional random fields (Krähenbühl & Koltun, 2011), like in Chen *et al.* (2018) and Kamnitsas *et al.* (2017), to potentially incorporate smoothness terms and increase local class consistency, creating more contiguous segmentation regions.

5.5. Parameter evaluation

Unfortunately, the proposed CNN is subject to limitations, which are not only a consequence of the chosen SDF-based approach but a consequence of the geometry discretization and the successive evaluation of the overall geometry of a part model.

The aforementioned sliding window approach (Section 4) with the pending parameter μ was therefore evaluated for the analysis of entire engineering part models. To obtain the optimal border overlap, increments of μ from 0 to 7 of each baseline network were inferenced as a constituent part of the respective test run. Compared to the non-overlapping sliding window case, an overlap of $\mu_{U-Like-32-EL} = 4$ and $\mu_{A-Like-32EL} = 1$ was determined sufficient, having an improved ΔmIOU of 1.8% and 2.2% and ΔDS 2.8% and 3.2%, respectively, when segmenting entire part models of a test set. Thus, the downsampling in the max-pooling layers of U-like intensifies misclassifications at the observation tile borders compared to the atrous-convolution in A-like.

To estimate the discretization configuration in the actual application, precise limits of recognizable instance scales need to be evaluated. These can transform between the units of the normalized voxels and real length units. For this reason, a resolution investigation was conducted using the maximum recognizable curvature of a cylindrical primitive. The comparison of different ranges of cylinder radii r demonstrates the different cylindrical scales (Fig. 17). For radii proportions between 1/16 and 1/8 of the tile dimensioning the mIoU were close to 50% and thus not separable from other classes. This should lead for example to an SDF voxel size of 0.4 mm for a minimal assumed r of 2.5 mm in a part model. Nevertheless, this resolution configuration needs to be adapted to the individual geometric features sizes of the architected materials type.

6. Results and Discussion

To further categorize the benefits of SDF processing, the applied EL of 16 voxels of the observation tile will be compared to the commonly used EL of 32 and 64 voxels. To determine whether general differences can be observed for the length gradations and if the reduced length was chosen sufficiently (Table 4), each baseline model was retrained with the respective resolution. The individual performances are summarized in Table 5, from which different findings can be derived. Supplemented by the learning results for the binary case, the main differences can be put into perspective as to whether the SDF approach is a superior prediction performance (Table 6).

In comparison with the binary case, it can be stated that less eligible detection rates can be achieved. Although naturally, the reliability of a TP detection increases with the sampling rate, the continuous distance values of the SDF field attenuate the need for more supporting points for an unambiguous inference. With the additional consideration of the memory consumption of the graphical processing unit, an 32EL can be a balanced compromise between available computational, detection probability, and training resources for universal usability.

Table 5: Comparison of overall mean accuracy results of 32EL of A-like and U-like after basic and transfer stage with per primitive mIoU on the test transfer dataset. The exterior and interior voxel classes are included in the overall accuracies. **Bold** text highlights the highest achieved accuracy per column.

Network	Repr.	Basic			Transfer							
		mDS	*mIoU*	*mAHD*	*mDS*	*mIoU*	*mAHD*	PL_{mIoU}	CY_{mIoU}	CO_{mIoU}	SP_{mIoU}	TO_{mIoU}
U-like	Binary	45.6	38.9	6.22	75.1	64.4	4.01	73.8	40.8	27.8	65.5	47.8
	SDF	**52.4**	**43.8**	5.53	**92.3**	**88.4**	**1.55**	**93.2**	**72.6**	**71.5**	**96.1**	**88.0**
A-like	Binary	49.0	41.6	5.96	77,4	67.5	3.65	77.4	37.7	49.4	67.2	49.1
	SDF	50.8	43.4	**5.51**	91.8	86.1	1.89	92.3	64.7	68.6	93.2	81.5

Table 6: Example models for resulting performance of *A-like*$_{SDF32EL}$.

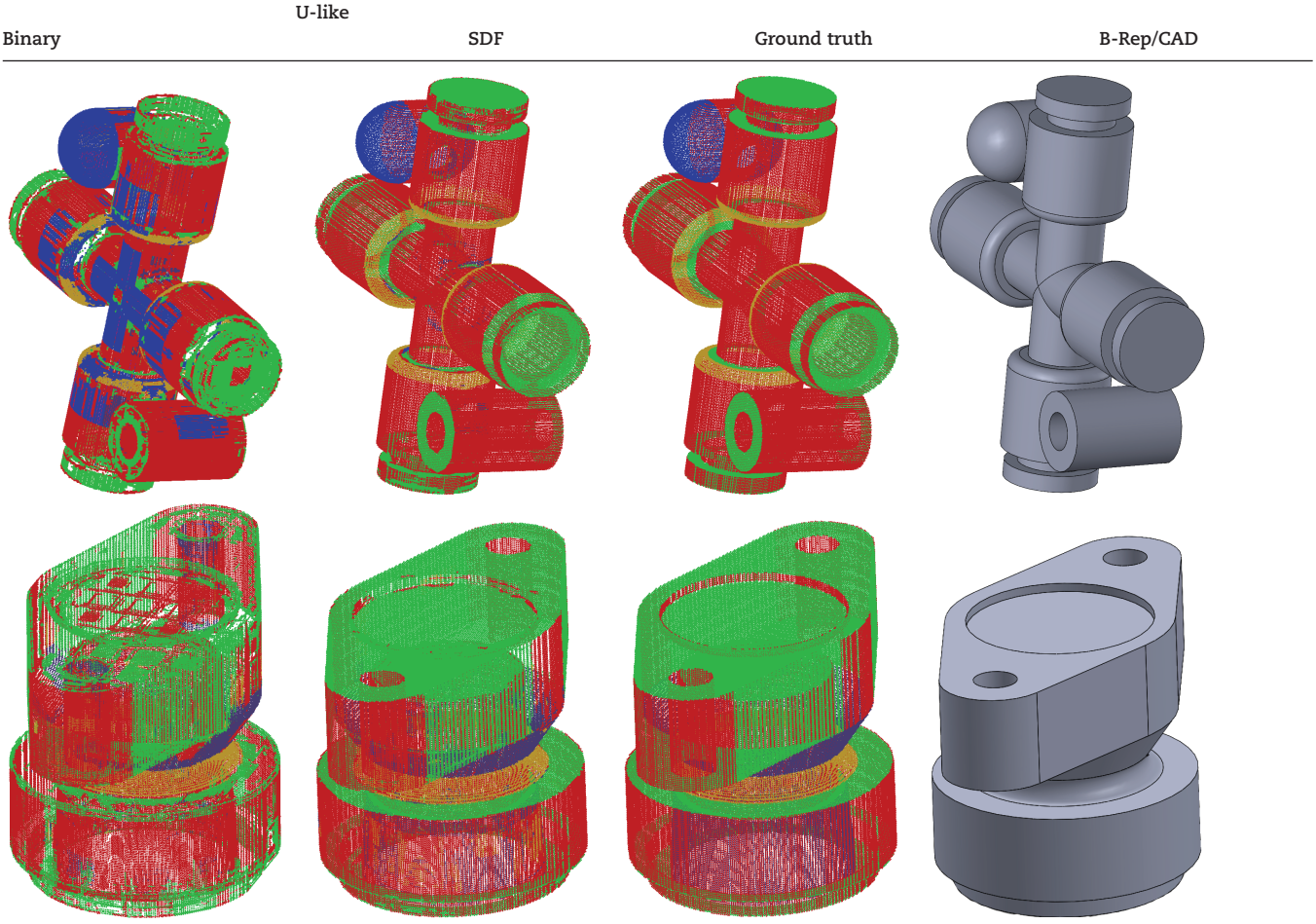

Table 7: Segmentation scores of point-wise primitive detection approaches in DL using the ABC dataset. Segmentation IoU extracted from Yan *et al.* (2021).

Network model	mIoU
Our study: U-like	88.4%
Our study: A-like	86.1%
SPFN (Li *et al.*, 2019)	80.1%
ParseNet (Sharma *et al.*, 2020)	88.6%
HPNet (Yan *et al.*, 2021)	94.2%

Unfortunately, the approach of M-like using a larger receptive field by a dual-path architecture does not alleviate the detection performances. On both data types, M-like convergence only to a maximum mIoU of 51.2% across all ELs, while no expressive segmentation can be derived. A defined rationale for why a reasonable approximation proves difficult cannot be given. As a result, M-like is no longer considered in the further discussion.

6.1. Baseline architecture comparison

The common U-like structure reports the highest segmentation results of all networks in most metrics. Particularly noticeable is that both networks are resistant to outliers with mAHD scores of 1.55 and 1.89 on the SDF case. Because this study is focused on the general practicability and possible improvements through the application of SDF values as input to CNNs, it is not intended to conclude or recommend which model is to be preferred. However, it can be stated that by using the SDF approach, both U-like and A-like recognition performances were improved, which makes the results potentially suitable for any 3D-CNN architecture.

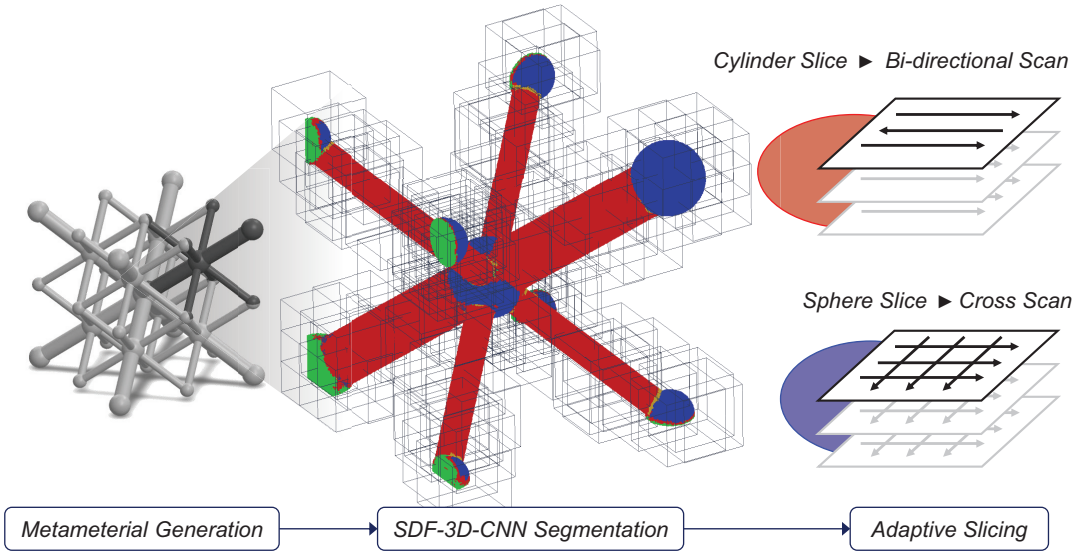

Figure 18: *A-like*$_{SDF32EL}$ segmentation with observation tile bounding boxes of a lightweight hierarchical lattice structure. Only the BCC cell mesh serves as a basis for the subsequent adaptive slicing and TGP.

6.2. Comparison to other approaches

The quantitative improvement over a binary approach is considered significant on the transfer set with an ΔmIoU between 20% for U-like and 25% for A-like, confirming the assumption that the discretization problems of the voxel field can be partially avoided. As a result of the single-precision floating-point format and its seven decimal digits of precision, a broader codomain as an additional input feature for the networks was available. In the future, it could still be clarified how the codomain of the observation tile could be normalized in such a way, that saturation phenomena of the activation functions could be bypassed. The intrinsic advantage of the SDF over the binary representation was often observed with regard to fine-grained structures when distinctive features have simply disappeared due to discretization in the binary case.

Through the domain adaption in the transfer learning stage, the needed variety of geometry transformations and their underlying features enhanced the resulting segmentation performance on the B-Rep test set. Compared to a single-stage learning process, the mIoU, mDS, and mAHD are improved by a Δ5.4%, Δ4.2%, and Δ0.6 for *U-like*$_{32EL}$, respectively. This is probably due to the consistent feature distribution of primitive transformations, which otherwise would not be present, and would prevent a transformation invariance. Thus, the transfer learning stage should be considered for each AM application of a geometric reasoning system to achieve a significant improvement of the recognition rate, without an adaptation of the network architecture.

In summary, A-like and U-like networks can generalize to the defined task of segmenting geometric primitives with the SDF approach, even for smaller observation volumes. A conclusive and general opinion regarding the network structure to be selected will not be given here because the distinct effects of dataset characteristics can affect the detection performance in a wide range. However, the use of the comparatively simple U-Like architecture represents a valid starting point for any further research and geometric reasoning in adaptive slicing. Whereas, the fluctuation of A-like can be taken as an indication that this network structure has the greatest capacity for more complex features.

Compared to other models that address the detection of geometric primitives (Table 7), it can be observed that similar detection results are achieved like Yan *et al.* (2021) or Sharma *et al.* (2020) with A-like and U-like. An exhaustive comparison performed on the same test dataset is out of the scope of this study. Nevertheless, their network architectures are much more sophisticated and constructed to the specific use case. The SDF framework presented here is feasible for any needed feature recognition in the AM process pipeline and is therefore much more universal in its application with less labor-intensive pre- and post-processing steps.

6.3. Use-case study in adaptive slicing

After proving the advantages of the SDF representation as input values in 3D-CNN networks, a use-case study shows the integration of the trained A-like network into an exemplary CAM process in AM (Fig. 18). This is to demonstrate a hypothetical and straightforward use case so that an otherwise static slicing and TPG respond to different geometry domains.

Following the generation of the geometry segmentation, the inferenced primitive labels are transferred to the inner volume by their nearest surface labels. The individual slice parametrization could be generated by distinguishing if a slice intersects with a predetermined primitive type to adapt the slicing and TPG strategy of the underlying AM process. With full integration into the extended product development pipeline, there would certainly be a need for a meaningful definition of the ROIs and further research about the adaption of slicing strategies. As the presented label approach is already volumetric, other spatial expansions to annotate such as 3D regions of specific thickness and multi-critical downskin angles or other material quality aspects are conceivable, which in turn require retraining of the 3D-CNN models by the redefined dataset.

7. Conclusions

In the context of AM, the adaptive parameterization of slicing data and TPG to individual geometric features offers the potential to further increase manufacturing speed and quality by an automated geometric reasoning system.

Therefore, this study presented a 3D-CNN-based feature recognition of geometric primitives by leveraging SDF data, which is becoming the common data representation for architected materials, providing a basic example of a structure-to-property linkage. The SDF representation approach limits the needed resolution and enhances the recognition accuracy compared to a binary voxel representation. Thus problematic anti-aliasing effects of Euclidean data types can be mitigated. By evaluating different 3D-CNN network structures, the feature recognition possibilities were considered in detail, showing the overall segmentation capabilities on various tessellated B-Rep models.

Furthermore, the octree-based sliding window procedure enables the observation of any volumetric expansion of part models with the required voxel resolution.

In the future, learning more specific geometric features of architected materials with thin-walled sections needs to be investigated. Especially regarding the examination of implicit volumetric feature characteristics, the currently used surface labeling needs to be modified.

Acknowledgments

The article processing charges were funded by the joint publication funds of the TU Dresden, including Carl Gustav Carus Faculty of Medicine, and the SLUB Dresden as well as the Open Access Publication Funding of the DFG.

Conflict of interest statement

None declared.

References

Alexa, M., Hildebrand, K., & Lefebvre, S. (2017). Optimal discrete slicing. *ACM Transactions on Graphics*, **36**(1), 1–16. https://doi.org/10.1145/2999536.

Alsallakh, B., Kokhlikyan, N., Miglani, V., Yuan, J., & Reblitz-Richardson, O. (2020). Mind the pad – cnns can develop blind spots.

Aydin, O. U., Taha, A. A., Hilbert, A., Khalil, A. A., Galinovic, I., Fiebach, J. B., Frey, D., & Madai, V. I. (2021). On the usage of average hausdorff distance for segmentation performance assessment: hidden error when used for ranking. *European radiology experimental*, **5**(1), 4. https://doi.org/10.1186/s41747-020-00200-2.

Bán, R., & Valasek, G. (2020). First order signed distance fields. in *Eurographics - Short Papers*.

Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Guennebaud, G., Levine, J. A., Sharf, A., & Silva, C. T. (2017). A survey of surface reconstruction from point clouds. *Computer Graphics Forum*, **36**(1), 301–329. https://doi.org/10.1111/cgf.12802.

Biswas, A., Shapiro, V., & Tsukanov, I. (2004). Heterogeneous material modeling with distance fields. *Computer Aided Geometric Design*, **21**(3), 215–242. https://doi.org/10.1016/j.cagd.2003.08.002.

Buonamici, F., Carfagni, M., Furferi, R., Governi, L., Lapini, A., & Volpe, Y. (2018). Reverse engineering modeling methods and tools: a survey. *Computer-Aided Design and Applications*, **15**(3), 443–464. https://doi.org/10.1080/16864360.2017.1397894.

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, **40**(4), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184.

Chen, Q., Xu, J., & Zhang, S. (2021). Cylindricity and flatness optimization for mechanical parts in additive manufacturing based on tolerance adaptive slicing. *The International Journal of Advanced Manufacturing Technology*, **115**(11-12), 3839–3857. https://doi.org/10.1007/s00170-021-07271-4.

Cheng, H., Liu, B., Liu, M., & Cao, W. (2022). Design of three-dimensional voronoi strut midsoles driven by plantar pressure distribution. *Journal of Computational Design and Engineering*, **9**(4), 1410–1429. https://doi.org/10.1093/jcde/qwac060.

Chowdhury, S., & Anand, S. (2016). Artificial neural network based geometric compensation for thermal deformation in additive manufacturing processes. *Volume 3: Joint MSEC-NAMRC Symposia*, American Society of Mechanical Engineers.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3d u-net: Learning dense volumetric segmentation from sparse annotation. *Medical image computing and computer-assisted intervention - MICCAI 2016*, vol. 9901 of Lecture Notes in Computer Science, pp. 424–432, eds Ourselin, S., Joskowicz, L., Sabuncu, M. R., Unal, G., Wells, W., Springer.

Cormier, D., Unnanon, K., & Sanii, E. (2000). Specifying non–uniform cusp heights as a potential aid for adaptive slicing. *Rapid Prototyping Journal*, **6**(3), 204–212. https://doi.org/10.1108/13552540010337074.

Dekhtiar, J., Durupt, A., Bricogne, M., Eynard, B., Rowson, H., & Kiritsis, D. (2018). Deep learning for big data applications in cad and plm – research review, opportunities and case study. *Computers in Industry*, **100**, 227–243. https://doi.org/10.1016/j.compind.2018.04.005.

Dolenc, A., & Mäkelä, I. (1994). Slicing procedures for layered manufacturing techniques. *Computer-Aided Design*, **26**(2), 119–126. https://doi.org/10.1016/0010-4485(94)90032-9.

Dolz, J., Ayed, I. B., Yuan, J., & Desrosiers, C. (2017). Isointense infant brain segmentation with a hyper-dense connected convolutional neural network. 616–20.

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, **40**, 100379. https://doi.org/10.1016/j.cosrev.2021.100379.

Fang, H., Yang, X., Kin, T., & Igarashi, T. (2021). Brain surface reconstruction from mri images based on segmentation networks applying signed distance maps. *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 1164–1168, IEEE, Piscataway, NJ.

Frisken, S. F., Perry, R. N., Rockwood, A. P., & Jones, T. R. (2000). Adaptively sampled distance fields. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, 249–254, ACM Press, New York.

Gao, W., Zhang, Y., Ramanujan, D., Ramani, K., Chen, Y., Williams, C. B., Wang, C. C., Shin, Y. C., Zhang, S., & Zavattieri, P. D. (2015). The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design*, **69**, 65–89. https://doi.org/10.1016/j.cad.2015.04.001.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A review on deep learning techniques applied to semantic segmentation.

Goodfellow, I., Bengio, Y., & Courville, A. (2018). Deep Learning: Das umfassende Handbuch : Grundlagen, aktuelle Verfahren und Algorithmen, *neue Forschungsansätze*. mitp, Frechen, 1st edn.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2021). Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, **43**(12), 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434.

Hoffmann, C. M. (2001). Robustness in geometric computations*. *Journal of Computing and Information Science in Engineering*, **1**(2), 143–155. https://doi.org/10.1115/1.1375815.

Hope, R. L., Roth, R. N., & Jacobs, P. A. (1997). Adaptive slicing with sloping layer surfaces. *Rapid Prototyping Journal*, **3**(3), 89–98. https://doi.org/10.1108/13552549710185662.

Hou, W., Zong, C., Wang, P., Xin, S., Chen, S., Liu, G., Tu, C., & Wang, W. (2022). Sdf-rvd: Restricted voronoi diagram on signed distance field. *Computer-Aided Design*, **144**, 103166. https://doi.org/10.1016/j.cad.2021.103166.

Ibtehaz, N., & Rahman, M. S. (2020). Multiresunet : Rethinking the u-net architecture for multimodal biomedical image segmentation. *Neural networks : the official journal of the International Neural Network Society*, **121**, 74–87. https://doi.org/10.1016/j.neunet.2019.08.025.

Jiang, J., & Ma, Y. (2020). Path planning strategies to optimize accuracy, quality, build time and material use in additive manufacturing: A review. *Micromachines*, **11**(7), https://doi.org/10.3390/mi11070633.

Jones, M. W., Baerentzen, J. A., & Sramek, M. (2006). 3d distance fields: a survey of techniques and applications. *IEEE transactions on visualization and computer graphics*, **12**(4), 581–599. https://doi.org/10.1109/TVCG.2006.56.

Kaiser, A., Ybanez Zepeda, J. A., & Boubekeur, T. (2019). A survey of simple geometric primitives detection methods for captured 3d data. *Computer Graphics Forum*, **38**(1), 167–196. https://doi.org/10.1111/cgf.13451.

Kambampati, S., Jauregui, C., Museth, K., & Kim, H. A. (2021). Geometry design using function representation on a sparse hierarchical data structure. *Computer-Aided Design*, **133**, 102989. https://doi.org/10.1016/j.cad.2020.102989.

Kamnitsas, K., Ledig, C., Newcombe, V. F. J., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D., & Glocker, B. (2017). Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, **36**, 61–78. https://doi.org/10.1016/j.media.2016.10.004.

Khadilkar, A., Wang, J., & Rai, R. (2019). Deep learning–based stress prediction for bottom-up sla 3d printing process. *The International Journal of Advanced Manufacturing Technology*, **102**(5-8), 2555–2569. https://doi.org/10.1007/s00170-019-03363-4.

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., & Panozzo, D. (2019). Abc: A big cad model dataset for geometric deep learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE.

Korn, H., Koch, P., Kordaß, R., Holtzhausen, S., Schöne, C., Müller, B., & Stelzer, R. (2018). Adapted scan strategy and slicing tool for improvement of strut precision in lattice structures. *Advancing Precision in Additive Manufacturing*, vol. Raleigh, NC: ASPE, 2018, pp. S.50–54, Berkeley.

Krähenbühl, P., & Koltun, V. (2012). Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, https://doi.org/10.48550/arXiv.1210.5644.

Lang, I., Manor, A., & Avidan, S. (2019). Samplenet: Differentiable point cloud sampling. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7575–7585.

Langelaar, M. (2018). Combined optimization of part topology, support structure layout and build orientation for additive manufacturing. *Structural and Multidisciplinary Optimization*, **57**(5), 1985–2004. https://doi.org/10.1007/s00158-017-1877-z.

LeCun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., Henderson, D., Howard, R. E., & Hubbard, W. (1989). Handwritten digit recognition: Applications of neural net chips and automatic learning. *IEEE Communication*, 41–46.

Lehder, E. F., Ashcroft, I. A., Wildman, R. D., Ruiz-Cantu, L. A., & Maskery, I. (2021). A multiscale optimisation method for bone growth scaffolds based on triply periodic minimal surfaces. *Biomechanics and modeling in mechanobiology*, **20**(6), 2085–2096. https://doi.org/10.1007/s10237-021-01496-8.

Letov, N., & Zhao, Y. F. (2022). A geometric modelling framework to support the design of heterogeneous lattice structures with non-linearly varying geometry. *Journal of Computational Design and Engineering*, **9**(5), 1565–1584. https://doi.org/10.1093/jcde/qwac076.

Li, H., Luo, Z., Zhang, N., Gao, L., & Brown, T. (2016). Integrated design of cellular composites using a level-set topology optimization method. *Computer Methods in Applied Mechanics and Engineering*, **309**, 453–475. https://doi.org/10.1016/j.cma.2016.06.012.

Li, L., Sung, M., Dubrovina, A., Yi, L., & Guibas, L. (2019). Supervised fitting of geometric primitives to 3d point clouds. 2647–2655. https://doi.org/10.1109/CVPR.2019.00276.

Li, L., McGuan, R., Isaac, R., Kavehpour, P., & Candler, R. (2021). Improving precision of material extrusion 3d printing by in-situ monitoring & predicting 3d geometric deviation using conditional adversarial networks. *Additive Manufacturing*, **38**, 101695. https://doi.org/10.1016/j.addma.2020.101695.

Lin, C.-H., Wang, C., & Lucey, S. (2020). Sdf-srn: Learning signed distance 3d object reconstruction from static images. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Advances in neural information processing systems, ed. ,H. Larochelle, Curran Associates Inc.

Liu, P., Zhang, K., Tateo, D., Jauhri, S., Peters, J., & Chalvatzaki, G. (2022). Regularized deep signed distance fields for reactive motion generation. *IROS 2022 Kyèto - IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE.

Liu, Y., Zheng, G., Letov, N., & Zhao, Y. F. (2021). A survey of modeling and optimization methods for multi-scale heterogeneous lattice structures. *Journal of Mechanical Design*, **143**(4), https://doi.org/10.1115/1.4047917.

Livesu, M., Ellero, S., Martínez, J., Lefebvre, S., & Attene, M. (2017). From 3d models to 3d prints: an overview of the processing pipeline. *Computer Graphics Forum*, **36**(2), 537–564. https://doi.org/10.1111/cgf.13147.

Mani, K., Kulkarni, P., & Dutta, D. (1999). Region-based adaptive slicing. *Computer-Aided Design*, **31**(5), 317–333. https://doi.org/10.1016/S0010-4485(99)00033-0.

Martinez, C., Potter, K. M., Smith, M. D., Donahue, E. A., Collins, L., Korbin, J. P., & Roberts, S. A. (2019). Segmentation certainty through uncertainty: Uncertainty-refined binary volumetric segmentation under multifactor domain shift. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 484–486, IEEE.

Matveev, A., Rakhimov, R., Artemov, A., Bobrovskikh, G., Egiazarian, V., Bogomolov, E., Panozzo, D., Zorin, D., & Burnaev, E. (2022). Def: Deep estimation of sharp geometric features in 3d shapes. ACM *Transactions on Graphics*, **41**(4), 1–22. https://doi.org/10.1145/3528223.3530140.

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. *Fourth International Conference on 3D Vision (3DV)*, 565–571.

Mohan Pandey, P., Venkata Reddy, N., & Dhande, S. G. (2003). Slicing procedures in layered manufacturing: a review. *Rapid Prototyping Journal*, **9**(5), 274–288. https://doi.org/10.1108/13552540310502185.

Museth, K. (2013). Vdb: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics*, **32**(3), 1–22. https://doi.org/10.1145/2487228.2487235.

Nguyen, C. H. P., Kim, Y., Do, Q. T., & Choi, Y. (2021). Implicit-based computer-aided design for additively manufactured functionally graded cellular structures. *Journal of Computational Design and Engineering*, **8**(3), 813–823. https://doi.org/10.1093/jcde/qwab016.

Nguyen, P. C. H., Kim, Y., & Choi, Y. (2022). Lightweight design with metallic additively manufactured cellular structures. *Journal of Computational Design and Engineering*, **9**(1), 155–167. https://doi.org/10.1093/jcde/qwab078.

Oh, S., Jung, Y., Kim, S., Lee, I., & Kang, N. (2019). Deep generative design: Integration of topology optimization and generative models. *Journal of Mechanical Design*, **141**(11), 436. https://doi.org/10.1115/1.4044229.

Oh, Y., Sharp, M., Sprock, T., & Kwon, S. (2021). Neural network-based build time estimation for additive manufacturing: a estimation comparison. *Journal of Computational Design and Engineering*, **8**(5), 1243–1256. https://doi.org/10.1093/jcde/qwab044.

Park, J. J., Florence, P., Straub, J., Newcombe, R., & Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, IEEE.

Peng, X., Sun, B., Ali, K., & Saenko, K. (2015). Learning deep object detectors from 3d models. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1278–1286, IEEE.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS*.

Qi, X., Chen, G., Li, Y., Cheng, X., & Li, C. (2019). Applying neural-network-based machine learning to additive manufacturing: Current applications, challenges, and future perspectives. *Engineering*, **5**(4), 721–729. https://doi.org/10.1016/j.eng.2019.04.012.

Qin, J., Hu, F., Liu, Y., Witherell, P., Wang, C. C., Rosen, D. W., Simpson, T. W., Lu, Y., & Tang, Q. (2022). Research and application of machine learning for additive manufacturing. *Additive Manufacturing*, **52**, 102691. https://doi.org/10.1016/j.addma.2022.102691.

Riegler, G., Ulusoy, A. O., & Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention - MICCAI 2015*, vol. 9351 of Lecture Notes in Computer Science, pp. 234–241, eds N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi, Springer.

Rucklidge, W. (1996). *Efficient visual recognition using the Hausdorff distance*, Springer Berlin Heidelberg.

Sarkon, G. K., Safaei, B., Kenevisi, M. S., Arman, S., & Zeeshan, Q. (2022). State-of-the-art review of machine learning applications in additive manufacturing; from design to manufacturing and property control. *Archives of Computational Methods in Engineering*, https://doi.org/10.1007/s11831-022-09786-9.

Sharma, G., Liu, D., Maji, S., Kalogerakis, E., Chaudhuri, S., & Měch, R. (2020). *Parsenet: A parametric surface fitting network for 3d point clouds*. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm, eds. Computer vision – eccv 2020, pp. 261–276. Springer International Publishing. https://doi.org/10.1007/978-3-030-58571-6_16.

Shi, P., Qi, Q., Qin, Y., Scott, P. J., & Jiang, X. (2021). Intersecting machining feature localization and recognition via single shot multibox detector. *IEEE Transactions on Industrial Informatics*, **17**(5), 3292–3302. https://doi.org/10.1109/TII.2020.3030620.

Silbernagel, C., Aremu, A., & Ashcroft, I. (2019). Using machine learning to aid in the parameter optimisation process for metal-based additive manufacturing. *Rapid Prototyping Journal*, **26**(4), 625–637. https://doi.org/10.1108/RPJ-08-2019-0213.

Sivapuram, R., Dunning, P. D., & Kim, H. A. (2016). Simultaneous material and structural optimization by multiscale topology optimization. *Structural and Multidisciplinary Optimization*, **54**(5), 1267–1281. https://doi.org/10.1007/s00158-016-1519-x.

Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, **15**, 29. https://doi.org/10.1186/s12880-015-0068-x.

Takashima, H., & Kanai, S. (2021). Recognition of free-form features for finite element meshing using deep learning. *Computer-Aided Design and Applications*, **19**(4), 677–693. https://doi.org/10.14733/cadaps.2022.677-693.

Tamburrino, F., Graziosi, S., & Bordegoni, M. (2018). The design process of additively manufactured mesoscale lattice structures: A review. *Journal of Computing and Information Science in Engineering*, **18**(4), https://doi.org/10.1115/1.4040131.

van de Ven, E., Maas, R., Ayas, C., Langelaar, M., & van Keulen, F. (2018). Continuous front propagation-based overhang control for topology optimization with additive manufacturing. *Structural and Multidisciplinary Optimization*, **57**(5), 2075–2091. https://doi.org/10.1007/s00158-017-1880-4.

Wang, C., Tan, X. P., Tor, S. B., & Lim, C. S. (2020a). Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing*, **36**, 101538. https://doi.org/10.1016/j.addma.2020.101538.

Wang, Z., Vandersteen, C., Demarcy, T., Gnansia, D., Raffaelli, C., Guevara, N., & Delingette, H. (2020b). A deep learning based fast signed distance map generation. *MIDL 2020 - Medical Imaging with Deep Learning*.

Wang, M., & Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, **312**, 135–153. https://doi.org/10.1016/j.neucom.2018.05.083.

Wang, M., Chen, Q., & Fu, Z. (2022). Lsnet: Learned sampling network for 3d object detection from point clouds. *Remote Sensing*, **14**(7), 1539. https://doi.org/10.3390/rs14071539.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., & Tong, X. (2017). O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics*, **36**(4), 1–11. https://doi.org/10.1145/3072959.3073608.

Wang, Z., & Lu, F. (????). Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes.

Wong, V. W. H., Ferguson, M., Law, K. H., Lee, Y.-T. T., & Witherell, P. (2020). Automatic volumetric segmentation of additive manufacturing defects with 3d u-net. *AAAI Spring Symposium*.

Xu, Q., Wang, W., Ceylan, D., Mech, R., & Neumann, U. (2019). Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *32nd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Advances in neural information processing systems, eds ,H., Larochelle, H. Wallach, Curran Associates Inc.

Xue, Y., Tang, H., Qiao, Z., Gong, G., Yin, Y., Qian, Z., Huang, C., Fan, W., & Huang, X. (2020). Shape-aware organ segmentation by predicting signed distance maps. *The Thirty-Fourth AAAI Conference on Artificial Intelligence, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence*, **34**, 12565–12572. Palo Alto, California USA.

Yan, S., Yang, Z., Ma, C., Huang, H., Vouga, E., & Huang, Q. (2021). Hpnet: Deep primitive segmentation using hybrid representations. *2021 IEEE/CVF International Conference on Computer Vision Workshops*, 2733–2742, IEEE.

Yang, J., Chen, Y., Huang, W., & Li, Y. (2017). Survey on artificial intelligence for additive manufacturing. *2017 23rd International Conference on Automation and Computing (ICAC)*, 1–6, IEEE.

Yang, Z., Wu, J., Thompson, P. M., & Wang, Y. (2021). Deep learning on sdf for classifying brain biomarkers. *Annual International Con-*

ference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference, **2021**, 1051–1054. https://doi.org/10.1109/EMBC46164.2021.9630850.

Yao, X., Moon, S. K., & Bi, G. (2017). A hybrid machine learning approach for additive manufacturing design feature recommendation. *Rapid Prototyping Journal*, **23**(6), 983–997. https://doi.org/10.1108/RPJ-03-2016-0041.

Yoo, S., Lee, S., Kim, S., Hwang, K. H., Park, J. H., & Kang, N. (2021). Integrating deep learning into cad/cae system: generative design and evaluation of 3d conceptual wheel. *Structural and Multidisciplinary Optimization*, 1–23. https://doi.org/10.1007/s00158-021-02953-9.

Zhang, J., Yao, Y., & Quan, L. (2021). Learning signed distance field for multi-view surface reconstruction. *2021 IEEE/CVF International Conference on Computer Vision*, 6505–6514, ed ,E. Mortensen, IEEE.

Zhang, Y., & Moon, S. K. (2021). Data-driven design strategy in fused filament fabrication: status and opportunities. *Journal of Computational Design and Engineering*, **8**(2), 489–509. https://doi.org/10.1093/jcde/qwaa094.

Zhang, Z., Jaiswal, P., & Rai, R. (2018). Featurenet: Machining feature recognition based on 3d convolution neural network. *Computer-Aided Design*, **101**, 12–22. https://doi.org/10.1016/j.cad.2018.03.006.

Zhu, Z., Anwer, N., Huang, Q., & Mathieu, L. (2018). Machine learning in tolerancing for additive manufacturing. *CIRP Annals*, **67**(1), 157–160. https://doi.org/10.1016/j.cirp.2018.04.119.

Zhu, Z., Ferreira, K., Anwer, N., Mathieu, L., Guo, K., Qiao, L., & Keine, A. (2020). Convolutional neural network for geometric deviation prediction in additive manufacturing. *Procedia CIRP*, **91**, 534–539. https://doi.org/10.1016/j.procir.2020.03.108.