Solid Freeform Fabrication 2018: Proceedings of the 29th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

USING AUTOENCODED VOXEL PATTERNS TO PREDICT PART MASS, REQUIRED SUPPORT MATERIAL, AND BUILD TIME

C. Murphy*, N. Meisel†, T. W. Simpson*, and C. McComb†

*Department of Mechanical and Nuclear Engineering
†School of Engineering Design, Technology, and Professional Programs
The Pennsylvania State University, University Park, PA 16802

Abstract

Additive Manufacturing (AM) allows designers to create intricate geometries that were once too complex or expensive to achieve through traditional manufacturing processes. Currently, designing parts using features specific to AM, commonly referred to as Design for Additive Manufacturing (DfAM), is restricted to experts in the field. As a result novices in industry may overlook potentially transformational design potential enabled by AM. This project aims to automate DfAM through deep learning making it accessible to a broader audience, and enabling designers of all skill levels to leverage unique AM geometries when creating new designs. To execute such an approach, a database of files was acquired from industry-sponsored AM challenges focused on lightweight design. These files were converted to a voxelized format, which provides more robust information for machine learning applications. Next, an autoencoder was constructed to a low-dimensional representation of the part designs. Finally, that autoencoder was used to construct a deep neural network capable of predicting various DfAM attributes. This work demonstrates a novel foray towards a more extensive DfAM support system that supports designers at all experience levels.

1. Introduction and Motivation

Many new manufacturing technologies embrace a digital thread, which generates an abundance of rich design data, providing a potential resource for training novice designers and incumbent workers trying to keep their skills current. However, best practices for utilizing this digital thread to provide feedback to designers have neither been developed nor critically assessed. This essay proposes a deep learning approach to extract knowledge from digital repositories to predict DfAM attributes associated for part designs. These predictions can in turn be used to support engineers as they develop intuition for novel designs with advanced manufacturing technology.

For this current work, we focus on AM as a representative digital manufacturing technology that is evolving rapidly. In 2015, it was estimated that 35% of engineering job postings required additive manufacturing skills [1]. Moreover, the number of job postings mentioning AM and 3D printing grew by 1834% between 2010 and 2014 [1]. This increased competition for engineers with AM skills is expected to impact companies of all sizes [2]. Because of these rapid increases in demand combined with rapidly evolving AM technology [3], there is an urgent need for targeted AM training and support for novice designers in these evolving companies [4,5]. This makes AM a great test case to investigate the proposed machine learning feedback approach.

2. Background

2.1. Advances in Design for Additive Manufacturing Feedback Tools

Lightweight redesign of structures and parts is a common objective in aerospace and other application areas (other objectives, such as part reduction, are also increasingly common). To encourage designers to better leverage the geometric opportunities offered by AM in the pursuit of lightweight structures, current approaches in the literature primarily focus on the use of robust topology optimization to mathematically determine the ideal structure, subject to certain performance metrics (e.g., strength-to-weight ratio) and manufacturability constraints (e.g., support material usage, minimum feature size). Current research efforts to implement topology optimization design algorithms within the AM context are extensive. These algorithms can take a variety of forms, including optimal distribution of material based on density [6] or optimal sizing of an initial ground structure [7], with more advanced algorithms even capable of accounting for multiple material phases [8]. While powerful, topology optimization methods may have difficulty converging to a manufacturable design solution without extensive prescriptive constraints [9]. As discussed by Marc Saunders (Director of Global Solutions Centres at Renishaw), "It is a mistake to think that designs that have been optimized for load bearing can simply be printed at the touch of the button" [10]. This qualifier is due to the often-overlooked manufacturability restrictions of AM, which largely go unaccounted for in current commercially available topology optimization algorithms. Sometimes termed "restrictive DfAM" [11], manufacturability constraints for AM include a system's minimum manufacturable feature size and hole size [12], geometric accuracy and repeatability [13], anisotropic material properties due to part orientation [14,15], and concerns over support material usage and removal [16]. While such constraints do appear in current topology optimization research (e.g., self-supporting angle considerations used in [17,18] and minimum feature size considerations used in [8]), they further increase the complexity of implementing such design algorithms in practice. These constraints, in turn, raise the difficulty of novices incorporating robust topology optimization approaches suitable for AM into their design process.

In response to these challenges with topology optimization, an emerging branch of research has focused on the development and codification of design rules and heuristics to support DfAM [19]. The establishment of such "rules of thumb" is driven by the need to fundamentally shift how engineers think when designing parts because of the stark differences between AM and traditional manufacturing [20]. Typically, such design rules are focused on improving part manufacturability with AM; some examples of these principles include hollowing out parts [21], minimizing support features [22], and rounding interior corners [23]. Recently, such principles have been delivered in simplified worksheets have been shown to reduce the number of failed prints when used by novices [24]. However, the inherent challenge with heuristic design rules for DfAM is that it is challenging to create rules that are comprehensive and universally applicable; the ever-evolving and expanding types of print processes, materials, and capabilities for AM has resulted in the proposal of extensive set of rules for each different AM type (see, for example, characteristic rulesets for material extrusion [25], material jetting [26], and powder bed fusion [12,27]). While researchers have proposed modular approaches to streamlining heuristic rulesets for DfAM [28], such formatting is currently not applied uniformly across the field of DfAM. As digital thread and digital twin paradigms become more common, machine learning may offer a way to generate insights by capitalizing on digital assets.

2.2. Machine Learning to Predict AM Quality

Rudimentary neural networks have been used to support DfAM in several ways, including estimation of build time [29], prediction of bead geometry for weld-based rapid prototyping [30], and compensation for thermal deformation [31]. In most cases, these and other AM-related neural networks primarily serve to approximate time-consuming calculations that directly connect the "as-designed" structure to the "as-manufactured" structure. However, we are not aware of any attempt to utilize these algorithms to provide design-centric feedback to assist novice designers. More generally, research on data-driven methodologies for engineering design is on the rise [32] and has included text-based design descriptions [33,34], patent text [35–37], and even online reviews [38,39]. However, the rise of online communities for solid modeling and engineering design (such as GrabCAD and Thingiverse) has made entirely new sources of data available for data mining and machine learning approaches. Other work has utilized these 3D databases to extract design principles based on human analysis of a subset of designs [40] and to train algorithms to predict functionality of a product based on form [41]; however, from what our research has led us, *no published work has attempted to automatically derive features of designs that are correlated with metrics of performance or manufacturability*. This may be because GrabCAD and similar databases permit contributions from non-experts [42], making low quality solutions more prevalent in the corpus of available data. Therefore, existing approaches either avoid inferring quality or utilize human judgment to select good designs. Traditionally, the presence of these poor solutions in the dataset has been viewed as detrimental; however, our work hypothesizes that if poor solutions can be identified, then they might be leveraged alongside well-engineered solutions to provide more balanced feedback for novice designers.

A key tenet of the current work is that the data from the online design repositories can be combined with machine learning in order to recognize the features that comprise both well-designed and poorly-engineered designs and structures for AM. Neural networks [43] are a common machine learning algorithm that have been particularly successful in two-dimensional image recognition tasks [43–45]. This success has led researchers to apply similar methodology to 3D recognition tasks [46], facilitated by recent advances in computing that enable such tasks to be performed at scale. Seminal 3D classification datasets and efforts include ObjectNet3D [47], ShapeNet [48], VoxNet [49], and PointNet [50]. Most of these approaches focus on recognizing or creating objects with a given form and category (e.g., [51,52]), but there has been little work that seeks to derive the deeper relationship between desired functionality (e.g., performance and manufacturability) and requisite form (e.g. voxelized geometry), which is the focus of our work.

The current work makes use of autoencoders to build machine representations of voxelized part geometries. An autoencoder is a neural network that is specifically designed to compress an input so that it is represented with a small number of variables, and then reconstruct it with the highest degree of accuracy possible [53]. Once trained, the autoencoder is a useful artifact in and of itself. The *encoder* (the portion of the network that compresses a sample) can be used for dimensionality reduction, the *decoder* (the portion of the network that reconstructs a sample based on the compressed representation) can be used to generate synthetic data, and the full *autoencoder* can be used to denoise samples. The current work specifically uses variational autoencoders that compress samples into a latent space so that the samples are approximately normally distributed [54], ensuring that the latent variables contain dense information. This is made possible by training the neural network with a two-part objective function containing a standard loss term as well as a measure of how normally-distributed the training samples are in the latent space (usually Kullback–Leibler divergence [55]). Variational autoencoders have been applied to a wide variety of data, including speech waveforms [56], human faces [57], geometric primitives [58], and used to model user surprise and curiosity [59].

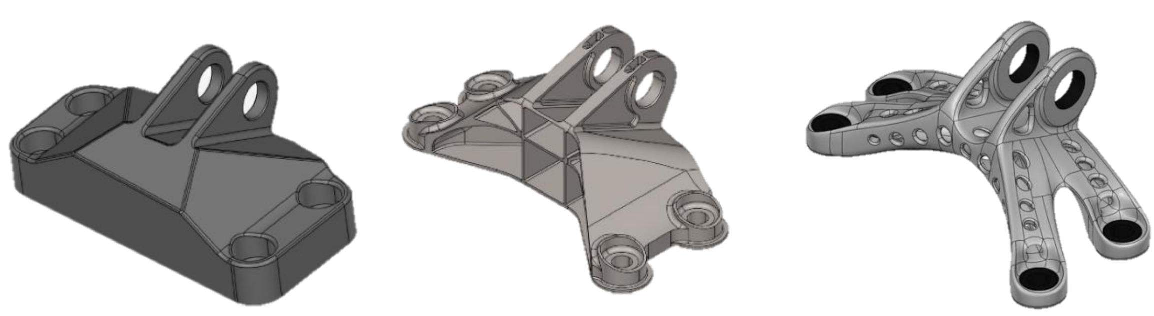
3. Methodology

The work in this paper consisted of three stages, each of which is detailed in this section. The first stage involved preparing data to be used for training and testing the deep neural network. Here, we used the solutions submitted to the General Electric (GE) GrabCAD design competition [60]. The next stage entailed training a variational autoencoder to learn a low-dimensional representation of the high-dimensional part designs. The final stage trained a deep neural network to predict initially basic, yet relevant AM metrics (part mass, mass of support material, and build time), based on the low-dimensional representation achieved with the variational autoencoder. All neural networks were trained using Keras [61] and Theano [62]. The software developed to accomplish this training is available in the Python programming language under an MIT License.[1]

The approach used here mirrors other work that used variational autoencoders to relate fluid response spectra to voxelized geometry for both design and analysis applications [58]. In that work, two autoencoders were trained: one for geometry and a second for the fluid response spectra. The primary difference is that the application entertained in the current work predicts a scalar value, necessitating only a single, geometry-based autoencoder to be trained.

3.1. Data Preparation

In 2013, GE hosted a design competition through the open source website GrabCAD [60], tasking competitors with using DfAM techniques to redesign a jet engine bracket with minimal weight while still satisfying the original loading conditions. Figure 1 shows the original geometry supplied to participants alongside the winning geometry and another finalist geometry that makes use of AM's geometric complexity capabilities. These submissions were uploaded in a variety of file formats. This challenge is one of the most well-known and well-populated case studies in crowdsourced DfAM, making it an ideal starting point for training our neural networks.



(a) (b) (c)

Figure 1. GE Jet Engine Bracket Challenge (a) original design, and (b) winning design, and (c) a design demonstrating significant geometric complexity.

[1] https://github.com/THREDgroup/AMnet/releases/tag/v0.1.0-beta

After initial data filtering to remove designs that were incomplete or nonfunctional due to file format incompatibilities, approximately 300 designs were chosen for analysis in this work. The quality of the designs was not controlled for during this preprocessing step, as a varied set of training data is desirable for machine learning applications.

The filtered files were downloaded and converted to stereolithography (STL) files, a standard and platform-agnostic file type used in AM. The STL files were next converted to a voxelized format using a MATLAB function [63] to ensure that each part could be represented with a vector of fixed size, a necessary requirement for many machine learning algorithms. Being an open source competition, the designs varied in shape, size, and orientation. For this project, the parts were converted at a resolution of 50^3 voxels with the edge length of a single voxel at 4mm, chosen to balance fidelity of the voxelized format against the speed at which machine learning algorithms could be trained.

For each part, several metrics were computed, including the mass of the part itself, the mass of required support material, and the estimated build time. These estimates were made using the MakerBot Desktop software (version 3.9.1.1143) [64], assuming an extruder temperature of 205 °C, 105 mm/s travel speed, 23 mm/s z-axis speed, 0.2 mm layer height, critical overhang angle of 68 degrees, 10% infill, and MakerBot PLA as the material. Parts were oriented to minimize build time.

Data augmentation approaches were used to increase the size of the available dataset [65]. Specifically, we used a rotational approach to increase the robustness of the trained network. This was accomplished by applying 24 unique rotations to each part, increasing the number of available samples in the dataset from 296 training samples to 7104 training samples. An example of the result of this data augmentation approach is provided in Figure 2, displaying 24 unique rotations for a single voxelized part geometry. For these rotated parts, build time and required support material were not re-assessed. This was intended to train the neural network to predict build time and required support material for any part assuming that the part would be oriented for minimum build time.

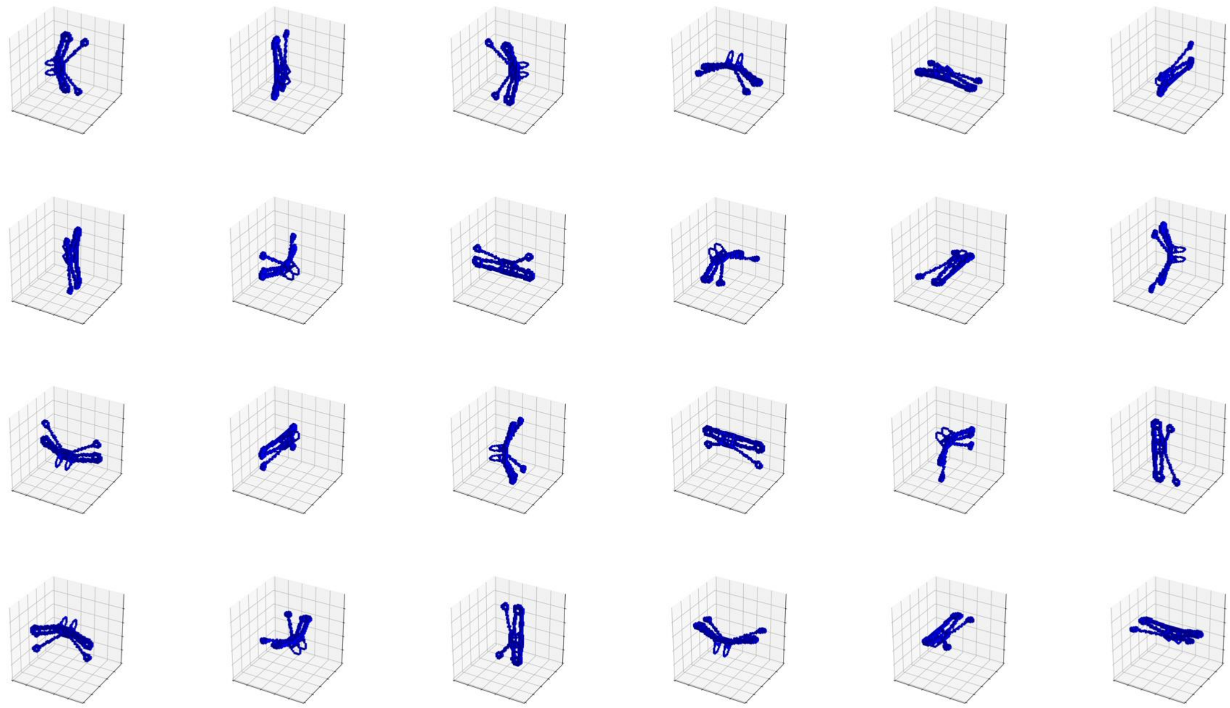


Figure 2. An example of the data augmentation approach, demonstrating 24 unique rotations of a single voxelized part.

3.2. Autoencoder Design and Training

The first neural network trained in this work was a variational autoencoder for voxelized geometry. An autoencoder is a specialized neural network that compresses and reconstructs data, and in doing so implicitly learns a compact way to represent that data. A variational autoencoder accomplished this by compressing the input into a *latent space*, a vector space in which the training data are normally distributed. The structure of the variational autoencoder used in this work is shown in Figure 3. This autoencoder was designed to compress the voxelized input (125000 total binary variables, each representing a voxel) into a *N*-dimensional latent space, with *N* approximately three orders of magnitude smaller than the inputs. The effect of the value of *N* is assessed later in this paper.

The autoencoder was trained using the root mean square propagation algorithm (also known as RMSprop) [66], a standard algorithm for this type of application. The loss function, the function minimized while training a neural network, used here was composed of two terms. The first was a binary cross-entropy term, a standard measure of the difference between two vectors that are expected to contain binary values. The second term used was an estimate of the Kullback-Leibler divergence [55] of the samples in the latent space (the innermost layer of the network). This second term is essential for ensuring that the neural network learns a compact representation in the latent space. The neural network was trained for 20 epochs at which point the testing loss had stopped decreasing. Training was performed on 75% of the dataset (5328 samples), and validation testing was performed on the remaining 25% of the dataset (1776 samples).

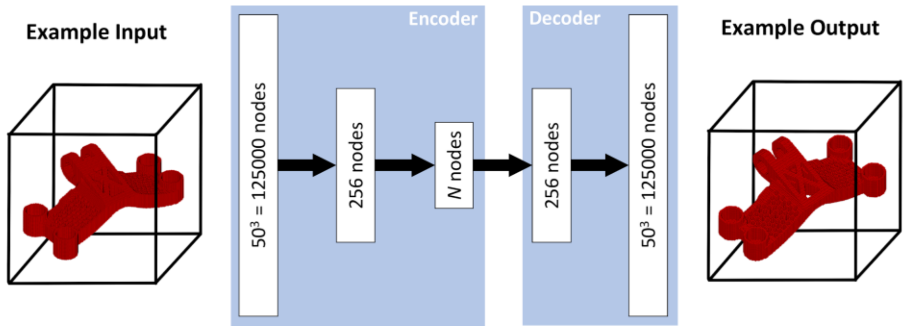


Figure 3. Structure of deep network for autoencoder.

3.3. Prediction Network Design and Training

Following the creation of the autoencoder, three additional neural networks were created to accomplish prediction - that is, the assessment of part mass, required support material, or build time based on the voxelized part geometry as an input. These prediction networks used the weights learned in the encoder portion of the variational autoencoder as a starting point. In essence, this recycles the compressed representation learned by the variational autoencoder, giving each neural network an initial representation to work from and therefore decreasing overall training time. It should also be noted that the same set of weights was utilized for all three prediction networks. The structure of these prediction networks is shown in Figure 4.

The RMSprop algorithm [66] was used to train each network. The loss function used was mean squared error since the desired output for each network was a real-valued number. These neural networks were trained for 10 epochs, at which point testing loss had stopped decreasing. Training was performed on 75% of the dataset (5328 samples), and validation testing was performed on the remaining 25% of the dataset (1776 samples). It should be noted that the split of the data into training and testing sets matched the split used for the autoencoder. This ensured that the prediction networks truly had no experience with any of the samples in the testing dataset.

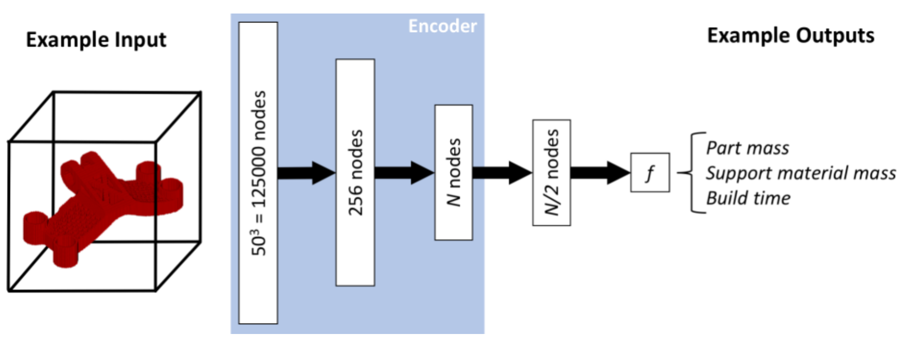


Figure 4. Structure of deep network for prediction of DfAM attributes. Note that the encoder portion of the autoencoder is repurposed in this network to allow mapping into the low-dimensional representation.

4. Results

Figure 5 shows the results of training the autoencoder and all three of the prediction networks using a latent space with varying dimensionality from 2 dimensions to 128 dimensions. The fraction of variance explained by the neural network (specifically, the coefficient of determination) as computed on the validation testing dataset is shown in the y-axis. The accuracy of the autoencoder continues to increase steadily up to 32 dimensions and achieves its highest accuracy with 128 dimensions. In contrast, the accuracy of the three prediction networks stops increasing fairly early, around 4 or 8 latent space dimensions. Based on this assessment, $N=8$ was chosen as the preferred value for the current application.

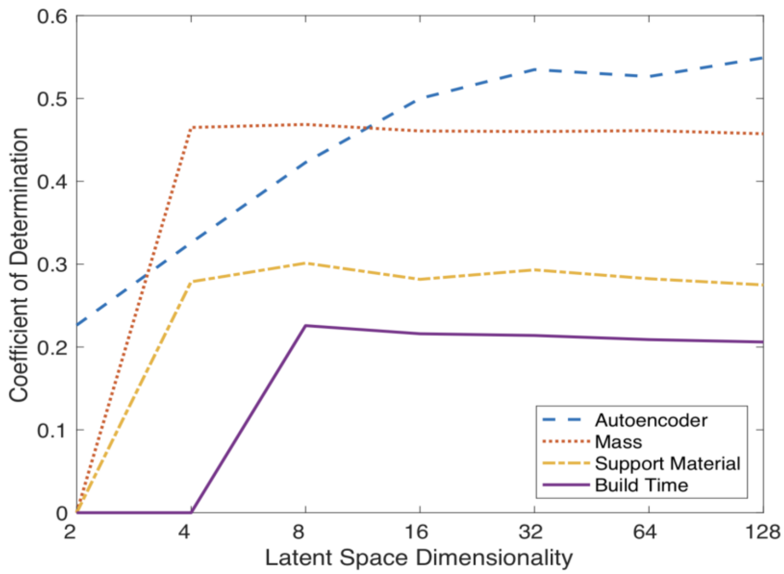


Figure 5. Coefficient of determination (computed on testing set) for autoencoder and prediction networks as a function of latent space dimensionality.

Several paired examples of original voxelized part designs and reconstructed voxelized part designs (after being fed through the autoencoder with 8 latent dimensions) are provided in Figure 6. On average, this autoencoder accounted for 42.3% of the variance in the validation testing dataset. Although the reconstructions are only approximations of the original inputs, they still capture the broad geometric representation of the part. Specifically, the geometry associated with the bolt holes and loading holes is typically reconstructed well. This is expected since these geometric components occur with a great degree of similarity across many different part designs submitted to the competition. The reconstruction is less accurate for the unique geometries that occur in between these loading points. The autoencoder also appears to only approximately capture lattice type structures. For instance, the design provided in Figure 6a utilized a lattice structure that was only approximately captured during voxelization, and it appears to have been almost entirely disregarded by the autoencoder upon reconstruction. A similar issue regarding reconstruction is apparent in Figures 6e and 6f. The accuracy and characteristics of reconstructions are important because they are representative of which features the network has learned well and which it has learned poorly. It is likely that shortcomings in the geometry autoencoder will limit the accuracy of networks designed to predict mass, required support material, and build time.

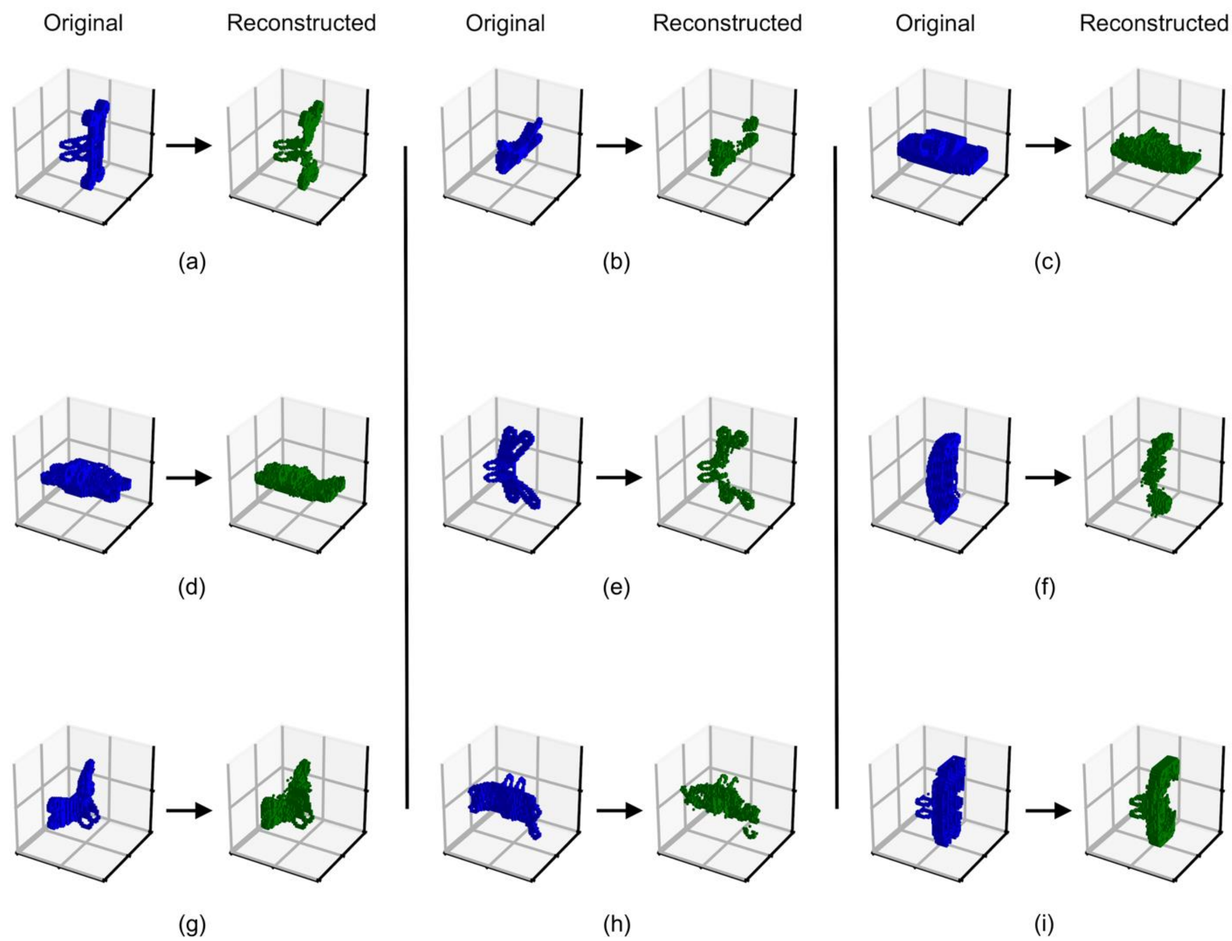
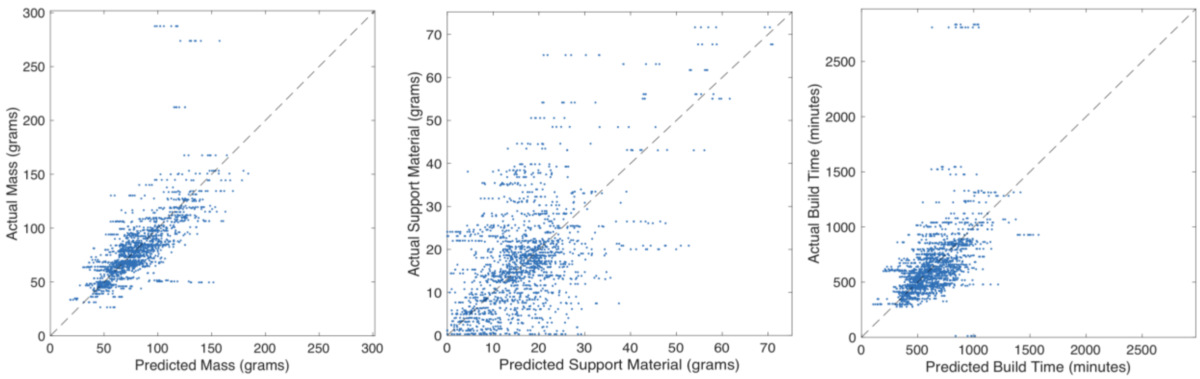


Figure 6. Autoencoder examples showing original and reconstructed voxelized geometries.

The three prediction networks that made use of the autoencoder explained 46.8% of the variance in part mass, 30.1% of the support material mass, and 22.5% of the predicted build time. These values are relatively low and are likely limited by the accuracy of the encoder, but give promise of the feasibility of this approach. Figure 7 shows plots of the predicted value for these metrics, derived from the three neural networks, against the actual value, derived from the MakerBot Desktop software (version 3.9.1.1143) [64]. Every data point on these plots represents a single training sample, and the dashed line represents the ideal prediction. The results of the mass prediction network might have been skewed slightly due to the existence of outliers. While most solutions are lighter than 175 g, some are as high as 300 g, the mass of these heavy parts was significantly underpredicted. A similar effect is apparent in Figure 7c. Most solutions print in less than 1500 minutes, but several take over 2500 minutes. This might have skewed the network during training, as the build time for these parts was significantly underpredicted.



(a) (b) (c)

Figure 7. Prediction versus actual value for (a) part mass, (b) support material mass, and (c) build time.

A likely underlying cause for the limited predictive accuracy is the limited accuracy of the autoencoder. Particularly true for parts with lattice-like regions (see Figure 6), the autoencoder did not accurately reconstruct the voxelized geometry, which was disappointing. Since the autoencoder disregards some of the more detailed portions of the part design, it is likely that these portions were not accounted for in any of the prediction networks. The functionality of the autoencoder in this regard might be improved by utilizing convolutional layers [67]. Convolutional methods enable neural networks to recognize and represent localized patterns, which would in turn make it more likely that lattice structure would be recognized and encoded in this application. Convolutional neural networks have been trained with high accuracy in a variety of image recognition and other tasks.

Another factor that might have contributed to the limited accuracy of the prediction networks (particularly with respect to support material and build time) is the way in which the initial dataset was augmented. Both build time and support material are inherently orientation dependent; however, these characteristics were not re-evaluated after the parts were rotated in the augmentation procedure, with the original intention that the neural networks would learn to predict the minimum build time and associated support material required for the part regardless of the orientation in which it was presented. It is possible that the prediction accuracy would increase if every part was associated with support material mass and build time computed for the specific orientation of the part that was autoencoded. This and additional future work are discussed next.

5. Conclusions and Future Work

This work represents a novel foray into the automated computation of DfAM attributes using deep neural networks, with an initial focus on predicting part mass, mass of required support material, and build time based on voxelized part geometry. First, part designs were downloaded from the GE GrabCAD design challenge and voxelized, and this dataset was augmented through a rotation technique. Next, a specialized neural network known as an autoencoder was trained to compress and reconstruct part designs in order to learn a compact representation of the voxelized geometries. Finally, three predictive neural networks were trained (each recycling the encoding portion of the autoencoder) to predict part mass, support material, and build time based on a voxelized input. Limited accuracy was achieved for these three networks, with coefficients of determination of 46.8% for part mass, 30.1% for support material mass, and 22.5% for build time.

Despite the limited accuracy of these three predictive networks, this work demonstrates the basic feasibility of leveraging deep learning methods to predict attributes in parts designed for AM.

Two incremental improvements to the existing deep learning network may increase training accuracy. First, modifying the networks trained here to utilize convolutional layers could improve the accuracy of the autoencoder (and therefore the prediction network as well) by enabling the networks to recognize and represent localized voxel patterns. This could account for the detailed geometric regions that are not recognized by the current autoencoder. Second, refining the data augmentation procedure could also lead to improved outcomes. The size of the dataset might be augmented further by using a randomized set of rotations that are not constrained to 90-degree turns. This would likely necessitate the rotation of STL files prior to voxelization. Separately assessing the required support material and build time for every rotation of the part could also increase predictive accuracy by providing more robust and complete training data.

Predicting attributes such as those investigated in the current work may be useful to designers, but the underlying equations that lead to the predictions are complex and not human-interpretable. More detailed design recommendations (such as where to add or subtract material) will be of limited use unless accompanied by some form of justification or explanation. This will be especially true for components that are flight or safety critical. Therefore, if this method is to be extended for more detailed DfAM recommendations, those recommendations must be paired with design-related and human-understandable rationales. Emerging principles and algorithms in the field of eXplainable Artificial Intelligence (XAI) may be of use towards this end.

Acknowledgements

This material is based upon work supported by the Penn State College of Engineering Research Initiative Program and the Rodney A. Erickson Discovery Grant Program. The authors gratefully acknowledge the support of the NVIDIA Corporation through the donation of the Quadro P5000 GPU used in this work. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Wanted Analytics, 2015, *Demand for 3D Printing Skills Soars*.

[2] Wohlers, T., and Caffrey, T., 2014, *Wohlers Report 2015: 3D Printing and Additive Manufacturing State of the Industry Annual Worldwide Progress Report*.

[3] Hornick, J., 2017, "3D Printing Patent Landscape," 3Dprint.com.

[4] Vazquez, E., Passaretti, M., and Valenzuela, P., 2016, "3D Opportunity for the Talent Gap," Deloitte Insights.

[5] Simpson, T. W., Williams, C. B., and Hripko, M., 2017, "Preparing Industry for Additive Manufacturing and Its Applications: Summary & Recommendations from a National Science Foundation Workshop," Addit. Manuf., **13**, pp. 166–178.

[6] Zegard, T., and Paulino, G. H., 2016, "Bridging Topology Optimization and Additive Manufacturing," Struct. Multidiscip. Optim., **53**(1), pp. 175–192.

[7] Chu, J., Engelbrecht, S., Graf, G., and Rosen, D. W., 2010, "A Comparison of Synthesis Methods for Cellular Structures with Application to Additive Manufacturing," Rapid Prototyp. J., **16**(4), pp. 275–283.

[8] Gaynor, A. T., Meisel, N. A., Williams, C. B., and Guest, J. K., 2014, "Multiple-Material Topology Optimization of Compliant Mechanisms Created Via PolyJet Three-Dimensional Printing," J. Manuf. Sci. Eng., **136**(6), p. 061015.
[9] Lu, J., and Chen, Y., 2012, "Manufacturable Mechanical Part Design with Constrained Topology Optimization," Proc. Inst. Mech. Eng. Part B J. Eng. Manuf., **226**(10), pp. 1727–1735.
[10] Saunders, M., 2017, "Is Topological Optimisation Really Optimal?"
[11] Laverne, F., Segonds, F., Anwer, N., and Le Coq, M., 2015, "Assembly Based Methods to Support Product Innovation in Design for Additive Manufacturing: An Exploratory Case Study," J. Mech. Des., **137**(12), pp. 121701–121701.
[12] Kranz, J., Herzog, D., and Emmelmann, C., 2015, "Design Guidelines for Laser Additive Manufacturing of Lightweight Structures in TiAl6V4," J. Laser Appl., **27**(S1), p. S14001.
[13] Fahad, M., and Hopkinson, N., 2012, "A New Benchmarking Part for Evaluating the Accuracy and Repeatability of Additive Manufacturing (AM) Processes," 2nd Int. Conf. Mech. Prod. Automob. Eng., pp. 234–238.
[14] Carroll, B. E., Palmer, T. A., and Beese, A. M., 2015, "Anisotropic Tensile Behavior of Ti-6Al-4V Components Fabricated with Directed Energy Deposition Additive Manufacturing," Acta Mater., **87**, pp. 309–320.
[15] Ahn, S.-H., Montero, M., Odell, D., Roundy, S., and Wright, P. K., 2002, "Anisotropic Material Properties of Fused Deposition Modeling ABS," Rapid Prototyp. J., **8**(4), pp. 248–257.
[16] Hu, K., Jin, S., and Wang, C. C. L., 2015, "Support Slimming for Single Material Based Additive Manufacturing," CAD Comput. Aided Des., **65**, pp. 1–10.
[17] Gaynor, A. T., and Guest, J. K., 2016, "Topology Optimization Considering Overhang Constraints: Eliminating Sacrificial Support Material in Additive Manufacturing through Design," Struct. Multidiscip. Optim., **54**(5), pp. 1157–1172.
[18] Mirzendehdel, A. M., and Suresh, K., 2016, "Support Structure Constrained Topology Optimization for Additive Manufacturing," CAD Comput. Aided Des., **81**, pp. 1–13.
[19] Rosen, D. W., 2014, "Research Supporting Principles for Design for Additive Manufacturing," Virtual Phys. Prototyp., **9**(4), pp. 225–232.
[20] Seepersad, C. C., 2014, "Challenges and Opportunities in Design for Additive Manufacturing," 3D Print. Addit. Manuf., **1**(1), pp. 10–13.
[21] Thompson, M. K., Moroni, G., Vaneker, T., Fadel, G., Campbell, R. I., Gibson, I., Bernard, A., Schulz, J., Graf, P., Ahuja, B., and Martina, F., 2016, "Design for Additive Manufacturing: Trends, Opportunities, Considerations, and Constraints," CIRP Ann., **65**(2), pp. 737–760.
[22] Gorguluarslan, R. M., Park, S.-I., Rosen, D. W., and Choi, S.-K., 2015, "A Multilevel Upscaling Method for Material Characterization of Additively Manufactured Part Under Uncertainties," J. Mech. Des., **137**(11), p. 111408.
[23] Adam, G. A. O., and Zimmer, D., 2014, "Design for Additive Manufacturing—Element Transitions and Aggregated Structures," CIRP J. Manuf. Sci. Technol., **7**(1), pp. 20–28.
[24] Booth, J. W., Alperovich, J., Chawla, P., Ma, J., Reid, T., and Ramani, K., 2017, "The Design for Additive Manufacturing Worksheet," J. Mech. Des., **139**(October), pp. 1–9.
[25] Stratasys Direct Manufacturing, 2015, "Fused Deposition Modeling (FDM) Design Guidelines," pp. 1–4.
[26] Meisel, N., and Williams, C., 2015, "An Investigation of Key Design for Additive

Manufacturing Constraints in Multimaterial Three-Dimensional Printing," J. Mech. Des., **137**(11), pp. 1–9.
[27] Seepersad, C. C., Govett, T., Kim, K., Lundin, M., and Pinero, D., 2012, "A Designer's Guide for Dimensioning and Tolerancing SLS Parts," *23rd Annual International Solid Freeform Fabrication Symposium*, Austin, TX, pp. 921–931.
[28] Jee, H., and Witherell, P., 2017, "A Method for Modularity in Design Rules for Additive Manufacturing," Rapid Prototyp. J., **23**(6), pp. 1107–1118.
[29] Di Angelo, L., and Di Stefano, P., 2011, "A Neural Network-Based Build Time Estimator for Layer Manufactured Objects," Int. J. Adv. Manuf. Technol., **57**(1–4), pp. 215–224.
[30] Xiong, J., Zhang, G., Hu, J., and Wu, L., 2014, "Bead Geometry Prediction for Robotic GMAW-Based Rapid Manufacturing through a Neural Network and a Second-Order Regression Analysis," J. Intell. Manuf., **25**(1), pp. 157–163.
[31] Chowdhury, S., and Anand, S., 2017, "Artificial Neural Network Based Geometric Compensation for Thermal Deformation in Additive Manufacturing Processes," pp. 1–10.
[32] Parraguez, P., and Maier, A. M., 2017, "Data-Driven Engineering Design Research: Opportunities Using Open Data," 21st Int. Conf. Eng. Des. (ICED 17), **7**(August), pp. 41–50.
[33] Cheong, H., Li, W., Cheung, A., Nogueira, A., and Iorio, F., 2017, "Automated Extraction of Function Knowledge From Text," J. Mech. Des., **139**(11), p. 111407.
[34] Shi, F., Chen, L., Han, J., and Childs, P., 2017, "A Data-Driven Text Mining and Semantic Network Analysis for Design Information Retrieval," J. Mech. Des., **139**(11), p. 111402.
[35] Fu, K., Cagan, J., Kotovsky, K., and Wood, K., 2013, "Discovering Structure in Design Databases Through Functional and Surface Based Mapping," J. Mech. Des., **135**(3), p. 031006.
[36] Fu, K., Chan, J., Schunn, C., Cagan, J., and Kotovsky, K., 2013, "Expert Representation of Design Repository Space: A Comparison to and Validation of Algorithmic Output," Des. Stud., **34**(6), pp. 729–762.
[37] Song, B., and Luo, J., 2017, "Mining Patent Precedents for Data-Driven Design: The Case of Spherical Rolling Robots," J. Mech. Des., **139**(11), p. 111420.
[38] Lim, S., and Tucker, C. S., 2017, "Mitigating Online Product Rating Biases Through the Discovery of Optimistic, Pessimistic, and Realistic Reviewers," J. Mech. Des., **139**(11), p. 111409.
[39] Jiang, H., Kwong, C. K., and Yung, K. L., 2017, "Predicting Future Importance of Product Features Based on Online Customer Reviews," J. Mech. Des., **139**(11), p. 111413.
[40] Perez, K. B., Anderson, D. S., and Wood, K. L., 2015, "Crowdsourced Design Principles for Leveraging the Capabilities of Additive Manufacturing," Int. Conf. Eng. Des., (July), pp. 1–10.
[41] Dering, M., and Tucker, C., 2017, "A Convolutional Neural Network Model for Predicting a Product's Function, Given Its Form," J. Mech. Des., **139**(November), pp. 1–14.
[42] Petrick, I. J., and Simpson, T. W., 2013, "Point of View: 3D Printing Disrupts Manufacturing: How Economies of One Create New Rules of Competition," Res. Manag., **56**(6), pp. 12–16.
[43] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012, "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst., pp. 1–9.
[44] Ren, S., He, K., Girshick, R., and Sun, J., 2015, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks."

[45] Zeiler, M. D., and Fergus, R., 2014, “Visualizing and Understanding Convolutional Networks,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), **8689 LNCS**(PART 1), pp. 818–833.

[46] Ioannidou, A., Chatzilari, E., Nikolopoulos, S., and Kompatsiaris, I., 2017, “Deep Learning Advances in Computer Vision with 3D Data,” ACM Comput. Surv., **50**(2), pp. 1–38.

[47] Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., and Savarese, S., 2016, “Objectnet3D: A Large Scale Database for 3D Object Recognition,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), **9912 LNCS**, pp. 160–176.

[48] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F., 2015, “ShapeNet: An Information-Rich 3D Model Repository.”

[49] Maturana, D., and Scherer, S., 2015, “VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition,” *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 922–928.

[50] Garcia-Garcia, A., Gomez-Donoso, F., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M., and Azorin-Lopez, J., 2016, “PointNet: A 3D Convolutional Neural Network for Real-Time Object Class Recognition,” *2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp. 1578–1584.

[51] Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L., 2017, “Representation Learning and Adversarial Generation of 3D Point Clouds,” pp. 1–20.

[52] Wu, J., Zhang, C., Xue, T., Freeman, W. T., and Tenenbaum, J. B., 2016, “Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling,” (Nips).

[53] Hinton, G. E., 2006, “Reducing the Dimensionality of Data with Neural Networks,” Science, **313**(5786), pp. 504–507.

[54] Kingma, D. P., and Welling, M., 2013, “Auto-Encoding Variational Bayes,” (Ml), pp. 1–14.

[55] Kullback, S., and Leibler, R. A., 1951, “On Information and Sufficiency,” Ann. Math. Stat., **22**(1), pp. 79–86.

[56] Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y., 2015, “A Recurrent Latent Variable Model for Sequential Data.”

[57] Rezende, D. J., Mohamed, S., and Wierstra, D., 2014, “Stochastic Backpropagation and Approximate Inference in Deep Generative Models.”

[58] McComb, C., 2018, “Towards the Rapid Design of Engineered Systems Through Deep Neural Networks,” *Design Computing and Cognition ’18*.

[59] Grace, K., Maher, M. Lou, Wilson, D., and Najjar, N., 2017, “Personalised Specific Curiosity for Computational Design Systems,” *Design Computing and Cognition ’16*, Springer International Publishing, Cham, pp. 593–610.

[60] 2013, “GE Jet Engine Bracket Challenge” [Online]. Available: https://grabcad.com/challenges/ge-jet-engine-bracket-challenge/entries.

[61] Chollet, F., 2015, “Keras.”

[62] Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., Bengio, Y., Bergeron, A., Bergstra, J., Bisson, V., Bleecher Snyder, J., Bouchard, N., Boulanger-Lewandowski, N., Bouthillier, X., de Brébisson, A., Breuleux, O., Carrier, P.-L., Cho, K., Chorowski, J., Christiano, P.,

Cooijmans, T., Côté, M.-A., Côté, M., Courville, A., Dauphin, Y. N., Delalleau, O., Demouth, J., Desjardins, G., Dieleman, S., Dinh, L., Ducoffe, M., Dumoulin, V., Ebrahimi Kahou, S., Erhan, D., Fan, Z., Firat, O., Germain, M., Glorot, X., Goodfellow, I., Graham, M., Gulcehre, C., Hamel, P., Harlouchet, I., Heng, J.-P., Hidasi, B., Honari, S., Jain, A., Jean, S., Jia, K., Korobov, M., Kulkarni, V., Lamb, A., Lamblin, P., Larsen, E., Laurent, C., Lee, S., Lefrancois, S., Lemieux, S., Léonard, N., Lin, Z., Livezey, J. A., Lorenz, C., Lowin, J., Ma, Q., Manzagol, P.-A., Mastropietro, O., McGibbon, R. T., Memisevic, R., van Merriënboer, B., Michalski, V., Mirza, M., Orlandi, A., Pal, C., Pascanu, R., Pezeshki, M., Raffel, C., Renshaw, D., Rocklin, M., Romero, A., Roth, M., Sadowski, P., Salvatier, J., Savard, F., Schlüter, J., Schulman, J., Schwartz, G., Serban, I. V., Serdyuk, D., Shabanian, S., Simon, É., Spieckermann, S., Subramanyam, S. R., Sygnowski, J., Tanguay, J., van Tulder, G., Turian, J., Urban, S., Vincent, P., Visin, F., de Vries, H., Warde-Farley, D., Webb, D. J., Willson, M., Xu, K., Xue, L., Yao, L., Zhang, S., and Zhang, Y., 2016, "Theano: A Python Framework for Fast Computation of Mathematical Expressions," arXiv e-prints, **abs/1605.0**.

[63] A, A., "Mesh Voxelisation," Mathworks File Exch. Softw. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation.

[64] Industries, M., "MakerBot Desktop Software."

[65] Wang, J., and Perez, L., 2017, "The Effectiveness of Data Augmentation in Image Classification Using Deep Learning," Convolutional Neural Networks Vis. Recognit.

[66] Tieleman, T., and Hinton, G., 2012, "Lecture 6.5-Rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude," **2**, pp. 26–31.

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