PeerJ Computer Science

# An autonomous framework for interpretation of 3D objects geometric data using 2D images for application in additive manufacturing

Mohammad reza Rezaei[1], Mahmoud Houshmand[1] and Omid Fatahi Valilai[2]

[1] Department of Industrial Engineering, Sharif University of Technology, Tehran, Tehran, Iran
[2] Department of Mathematics & Logistics, Jacobs University Bremen, Bremen, Bremen, Germany

## ABSTRACT

Additive manufacturing, artificial intelligence and cloud manufacturing are three pillars of the emerging digitized industrial revolution, considered in industry 4.0. The literature shows that in industry 4.0, intelligent cloud based additive manufacturing plays a crucial role. Considering this, few studies have accomplished an integration of the intelligent additive manufacturing and the service oriented manufacturing paradigms. This is due to the lack of prerequisite frameworks to enable this integration. These frameworks should create an autonomous platform for cloud based service composition for additive manufacturing based on customer demands. One of the most important requirements of customer processing in autonomous manufacturing platforms is the interpretation of the product shape; as a result, accurate and automated shape interpretation plays an important role in this integration. Unfortunately despite this fact, accurate shape interpretation has not been a subject of research studies in the additive manufacturing, except limited studies aiming machine level production process. This paper has proposed a framework to interpret shapes, or their informative two dimensional pictures, automatically by decomposing them into simpler shapes which can be categorized easily based on provided training data. To do this, two algorithms which apply a Recurrent Neural Network and a two dimensional Convolutional Neural Network as decomposition and recognition tools respectively are proposed. These two algorithms are integrated and case studies are designed to demonstrate the capabilities of the proposed platform. The results suggest that considering the complex objects which can be decomposed with planes perpendicular to one axis of Cartesian coordination system and parallel withother two, the decomposition algorithm can even give results using an informative 2D image of the object.

**Subjects** Autonomous Systems, Computer Aided Design, Emerging Technologies

**Keywords** Recurrent Neural Networks, Long-Short Term Memory cells, Convolutional Neural Networks, Feature recognition for additive manufacturing, Shape interpretation, Artificial intelligence in additive manufacturing, Additive manufacturing, Cloud based manufacturing

Submitted 16 February 2021
Accepted 17 June 2021
Published 10 August 2021

Corresponding author
Omid Fatahi Valilai,
o.fatahivalilai@jacobs-university.de

Academic editor
Chan Hwang See

Additional Information and Declarations can be found on page 28

DOI 10.7717/peerj-cs.629

## INTRODUCTION

Today, the global market for products forces companies to compete for keeping their market position (*Buckley, 2009*). In this context, the ability to respond to the altering demands of

**How to cite this article** Rezaei M, Houshmand M, Fatahi Valilai O. 2021. An autonomous framework for interpretation of 3D objects geometric data using 2D images for application in additive manufacturing. *PeerJ Comput. Sci.* **7**:e629 http://doi.org/10.7717/peerj-cs.629

the customers with customization in a relatively short time is a necessity (*Gunasekaran et al., 2019*). Consequently methods to create agility in companies are of much concern in this era and every progress which can improve the responsiveness of the companies facing high amount of altering demands and customizing the product features, is eagerly considered.

Beside the intense competition to attract customers, globalization has created a basis for better cooperation and collaboration between companies that can help them in their competition; the introduction of the internet of things (IoT) and cloud manufacturing, which has enabled companies to create pools of services and manufacturing resources to achieve cheaper manufacturing processes are two events that are going to reshape companies (*Xu, 2012*). On the other hand the large amount of data and complex decision making conditions, force companies to expand their reliance upon the artificial intelligence based tools (*Kiron & Schrage, 2019*; *Davenport & Ronanki, 2018*). Artificial intelligence provides companies with tools that ease the analysis of data and information in different phases of the production process and in some cases, it can replace the costly and error prone human agent. In fact, the rapid growth of knowledge in the field of cognitive science has improved the artificial intelligence tools to become more accurate mimicry of the human agent. As a result, they become more capable of replacing human agent in different domains. These paradigms shape the pillars of the new emerging industrial revolution, coined as industry 4.0 (*Schwab, 2016*; *Vaidya, Ambad & Bhosle, 2018*).

Another important progress in technology which is also at heart of the industry 4.0 is the additive manufacturing, or as known as its most popular technology, 3D printing (*Schwab, 2016*; *Vaidya, Ambad & Bhosle, 2018*). Additive manufacturing, as its old name "rapid prototyping" suggests, has been initially used as a technique for creating prototypes of products for further analysis during the product development process (*Campbell, Bourell & Gibson, 2012*). In fact, producing objects in a relatively short time and with high flexibility, was the first intention to use these methods, and after the introduction of less expensive methods, and increase in the demand side pressures on the companies, became their prominent advantage and justified their application (*Mehrpouya et al., 2019*; *Thomas, 2013*).

The combination of the power of cloud manufacturing and additive manufacturing methods can create a flexible, competitive and lower cost production ecosystem (*Baumann & Roller, 2017*; *Hartman, 2020*; *Wang et al., 2019*; *Valilai & Houshmand, 2015*). However, an important concern has to be addressed: although additive manufacturing methods have a great flexibility in producing objects without being restricted to geometric specifications, the selection of the proper additive manufacturing technology and proper additive manufacturing parameters are essential for the product pricing and decision making upon required techniques (*Wang, Blache & Xu, 2017b*). As a result it has to be decided autonomously that which product order should be produced by each of the available services in the cloud (*Goodarzi et al., 2020*). This is very essential when the additive manufacturing is considered to be accomplished by an XaaS approach in the cloud manufacturing demands (*Valilai & Houshmand, 2015*), where it has to be decided that whether each product be manufactured by each method and even the production parameters in each method have to be determined (*Baumann & Roller, 2017*).

The main objective of this paper is to propose an autonomous intelligent framework to interpret the geometry of the shape to create a basis for better integration of the design and manufacturing phases of the production process, reducing the constraints created on the design phase on behalf of manufacturing considerations (*Wang et al., 2020*; *Wang, Blache & Xu, 2017a*). In fact, the flexible structure proposed in this paper can create a basis for an autonomous recognition and expansion of the defined features based on the requirements of the system. In this framework, determining the best production technique to create the desired product can be accomplished autonomously by comparing the attributes of the features with capabilities of the available techniques.

An analysis of the studies about applications of machine learning in additive manufacturing shows that while there are studies which applied machine vision methods in solving issues of design and manufacturing processes in additive manufacturing (*Wang et al., 2020*; *Scime & Beuth, 2018*) and also there are studies which try to improve the integrity between these two processes by decomposing the part to smaller components (namely features) to make decisions upon them (*Zhao & Guo, 2020*; *Ranjan, Samant & Anand, 2017*), there is a lack of a flexible and autonomous structure, based on machine vision, to interpret the shape in terms of features for further use. In fact, studies like *Zhao & Guo (2020)* and *Ranjan, Samant & Anand (2017)* which try to interpret the geometry of the shape based on their features, omit the flexibility and learnability created by artificial intelligence based methods which makes the application of proposed methods problematic in applying them in cloud based platforms where nonprofessional customers want their products to be produced using simple 3D designs or even 2D pictures. On the other hand, studies like *Wang et al. (2020)* and *Scime & Beuth (2018)*, which try to apply artificial intelligence based methods, overlook the need for autonomous intelligent methods in shape interpretation to make decisions upon production process, especially in cloud based platforms. The proposed framework in this paper is intended to be a basis to fill this gap.

The proposed framework is a multiplex of algorithms which decompose the shape into simpler parts and recognize, or categorize each part using two kind of neural networks separately. To illustrate the problem in better words the next section is dedicated to a review of some related previous works. In 'Proposed framework' the autonomous intelligent framework is explained in terms of its two main phases which are shape decomposition and recognition; then their connection is built up, such that they create one autonomous intelligent structure. Case examples shows the applicability of the framework applying it to two cases. In 'Conclusion and Discussion' the overall conclusion of the paper is explained and the future research potentials related to this paper are suggested.

## LITERATURE REVIEW

As stated earlier, as this paper aims to propose an autonomous framework for geometry interpretation, the dominant contexts of the Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) using Long-Short Term Memory cells (LSTM) are applied as mainstream artificial intelligence models. This work aims to use the capabilities of these models which can be found in literature studies like *Ajit, Acharya & Samanta (2020)* and *Yu et al. (2019)*.

## Related previous works

An overall review of some related works in different approaches related to this study is presented in Table 1.

More comprehensive reviews on the two main interests of this paper, including cloud based additive manufacturing systems and the application of artificial intelligence in additive manufacturing can also be found in *Guo & Qiu (2018)* and *Goh, Sing & Yeong (2020)* respectively.

## Gap analysis and research approach

Studies related to this paper are of the following categories:

- Studies with focus on shape, object or feature recognition or categorization. These studies are either of two groups:
  - A group of these studies are trying to improve methods in artificial intelligence and machine vision in a general approach (*Lake et al., 2017*; *Miyagi & Aono, 2017*; *Yu et al., 2019*; *Yin et al., 2020*; *Ajit, Acharya & Samanta, 2020*). Such studies provide a good insight but their suggested methods should be modified to become applicable in an autonomous intelligent manufacturing framework. In fact some of their considerations, should be changed to be adjusted for applications in manufacturing.
  - The other group of these studies include studies which are either adjusted for application in manufacturing or their view is inline with manufacturing concerns (*Biegelbauer, Vincze & Wohlkinger, 2008*; *Wohlkinger et al., 2012*; *Maidin, Campbell & Pei, 2012*; *Yao, Moon & Bi, 2017*; *Wang, Blache & Xu, 2017a*; *Shi et al., 2018*; *Pham et al., 2018*). The main problem with these studies is that they either omit artificial intelligence methods or apply methods which can not show a satisfying level of flexibility and learnability.
- Studies with focus on methods to estimate parameters of additive manufacturing or estimate production quality in additive manufacturing (*Munguía, Ciurana & Riba, 2009*; *Moroni, Syam & Petrò, 2014*; *Piili et al., 2015*; *Shi et al., 2017*; *Zohdi, 2018*; *Zhao et al., 2018*; *Chan, Lu & Wang, 2018*; *Zhao & Guo, 2020*; *Williams et al., 2019*). The viewpoint in these methods is restricted to calculate or improve special parameters or values, and the general interpretation of the geometry of the shape from manufacturing point of view is either omitted, which restricts the application domain of them and of their related works to special conditions and/or shop floor level decisions, or done using non-intelligent or partially intelligent methods. As a result, the latter group face the same issues as the second group of the first category.
- Studies with focus on the selection of manufacturing method or service composition in a cloud (*Conner et al., 2014*; *Mai et al., 2016*; *Simeone, Caggiano & Zeng, 2020*). These studies consider decisions in higher levels than the second category but they are either very general architectures of the cloud which omit some issues in the integration of the design and manufacturing or try to solve them using not or partially flexible solutions.

The studied literature in Table 1 shows that the most common application of artificial intelligence in the additive manufacturing deals with parameter fixation or output

**Table 1** **Dominant research studies in the literature.**

| Research study | Research focus | Contribution |
|---|---|---|
| *Biegelbauer, Vincze & Wohlkinger (2008)* | Object recognition | A model for pose detection and object recognition based on the relative position of previously defined basic shapes |
| *Wohlkinger et al. (2012)* | Object recognition | A model to recognize 3D objects in CAD models based on 3D shape descriptors |
| *Lake et al., (2017)* | An overview of general approaches in artificial intelligence | An overview of the main trends in artificial intelligence methods and their shortcomings |
| *Miyagi & Aono (2017)* | Shape recognition | Applying a hybrid of CNN and LSTM to recognize 3D shapes represented as voxel grids. |
| *Yin et al. (2020)* | Shape recognition | Applying a modified version of CNN architecture for 3D shape recognition |
| *Munguía, Ciurana & Riba (2009)* | Build time estimation | A neural network for time estimation in additive manufacturing |
| *Maidin, Campbell & Pei (2012)* | Feature categorization | A design feature database to be used in product design in different disciplines |
| *Conner et al. (2014)* | Manufacturing method selection | A reference system for making decisions about the manufacturing method for a part considering customization, complexity and production volume |
| *Moroni, Syam & Petrò (2014)* | Production accuracy estimation | A simulation based model to estimate the accuracy of the manufactured part relative to the designed one |
| *Piili et al. (2015)* | Cost estimation of laser additive manufacturing | A model for the cost estimation of laser additive manufacturing of stainless steel |
| *Mai et al. (2016)* | Service composition in cloud manufacturing | A high level architecture for a cloud manufacturing system based on additive manufacturing for customized parts |
| *Yao, Moon & Bi (2017)* | Feature recommendation in design | A method providing feasible conceptual design solutions for inexperienced designers by recommending appropriate additive manufacturing design features |
| *Shi et al. (2017)* | Manufacturability analysis in a cloud based system | A three dimensional model to assess the manufacturability of a design in a cloud based additive manufacturing system |
| *Wang, Blache & Xu (2017a)*, *Wang, Blache & Xu (2017b)* | Design for additive manufacturing in a cloud based system | Creation of a knowledge based framework to help users to apply design for additive manufacturing |
| *Rudolph & Emmelmann (2017)* | Automated order processing in cloud based additive manufacturing | Representing a cloud based platform to process customer orders automatically |
| *Zohdi (2018)* | Accurate tool path generation | A genetic algorithm based method for the tool path generation for a rapid free-form printing |
| *Pham et al. (2018)* | Shape categorization | A shape interpretation model to use CNN to categorize shapes due to their legitimecy of production |
| *Zhao et al. (2018)* | Nonplanar slicing and path generation | An algorithm to slice non planar surfaces based on flattening them into planar |
| *Shi et al. (2018)* | Feature based manufacturability analysis | A model for manufacturability analysis based on heat kernel signature |
| *Chan, Lu & Wan (2018)* | Cost estimation in cybermanufacturing | A framework for manufacturing cost estimation using LASSO regression |
| *Zhao & Guo (2020)* | Adaptive slicing based on type of features | An adaptive slicing method based on the type of the feature |
| *Williams et al. (2019)* | Quantitative manufacturing metrics estimation | A 3D Convolutional Neural Network to estimate manufacturing metrics from design |
| *Simeone, Caggiano & Zeng (2020)* | Service composition | A modular platform to improve the resource efficiency in a cloud of additive manufacturing services |

prediction, while geometrical interpretation as an initial step to connect customer view of demanded part with an automated production system has been neglected. In fact in some cases this issue has been overcame by explicitly defined shape models or other non flexible methods which can become confusing. The literature of design for additive manufacturing paradigm (*Vaneker et al., 2020*), emphasizes on some limitations on the design phase; these limitations create constraints on behalf of product customization and design by customers view (*Wang, Blache & Xu, 2017a*) while it is the duty of the production system to flexibly welcome the needs of the customers and to configure itself to meet them.

The main approach in this paper is to propose a framework to flexibly decompose an input shape, or even a 2D picture from the shape, taken from an informative viewpoint, into simpler shapes and to categorize them based on their similarity to previously categorized shapes. Although applying different neural networks is a common categorization method in different fields of study, including additive manufacturing, in most cases a neural network is responsible for the categorization from a scratch (*Yin et al., 2020*; *Williams et al., 2019*). The results of these models can get very accurate but as they are blind to the common, simpler building blocks of the shape, facing the complex shapes created by these building blocks usually gets problematic (*Lake et al., 2017*). The two fold framework proposed in this paper can overcome this problem by an initial decomposition before categorization or recognition.

This paper proposes a framework to interpret geometrical shapes (concerning additive manufacturing related ones) to analyze their production process automatically. As investigated in the literature, the automation of this step has not been of much concern. In fact this process is commonly accomplished by human agents and the human agent decides the production method or process; the dependence upon human agents makes the process time and money consuming and creates a discontinuity in the flow of data and information in an intelligent cloud based manufacturing ecosystem; but by using a flexible autonomous intelligent framework these costs can be reduced and a fully automated platform can be created.

Moreover, as the outputs of the proposed framework are general shape interpretations, they can be used as an input for parameter fixation models or intelligent systems to flexibly determine the best parameters to achieve better outputs; in doing so, this framework can ease the highly appealing shape categorization based quality improvement of the additive manufacturing parts. Finally, it should be said that the cognitive models such as one developed in *Hummel & Biederman (1992)* or gestalt perceptual apparatus (*Wagemans, 2015*) that are the main inspirations for the core idea of this paper, have to be of much greater considerations in engineering fields; because they create great insights into the human cognition, which can bring about many advantages, most importantly agility and flexibility.

## PROPOSED FRAMEWORK

As explained before, regarding the industry 4.0 revolution which demands the autonomous intelligence in production and collaboration in the manufacturing process to its full extent,

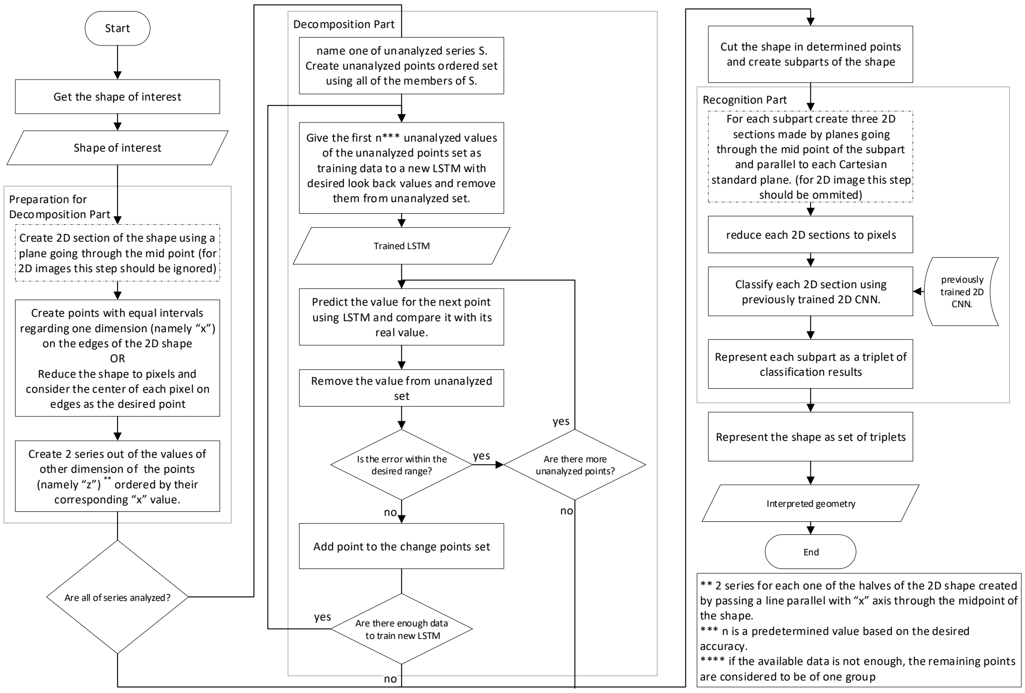
Start
Get the shape of interest
Shape of interest
Preparation for Decomposition Part
Create 2D section of the shape using a plane going through the mid point (for 2D images this step should be ignored)
Create points with equal intervals regarding one dimension (namely "x") on the edges of the 2D shape OR Reduce the shape to pixels and consider the center of each pixel on edges as the desired point
Create 2 series out of the values of other dimension of the points (namely "z") ** ordered by their corresponding "x" value.
Are all of series analyzed?
Decomposition Part
name one of unanalyzed series S. Create unanalyzed points ordered set using all of the members of S.
Give the first n*** unanalyzed values of the unanalyzed points set as training data to a new LSTM with desired look back values and remove them from unanalyzed set.
Trained LSTM
Predict the value for the next point using LSTM and compare it with its real value.
Remove the value from unanalyzed set
Is the error within the desired range?
yes
Are there more unanalyzed points?
yes
no
no
Add point to the change points set
Are there enough data to train new LSTM
yes
no
Cut the shape in determined points and create subparts of the shape
Recognition Part
For each subpart create three 2D sections made by planes going through the mid point of the subpart and parallel to each Cartesian standard plane. (for 2D image this step should be ommited)
reduce each 2D sections to pixels
Classify each 2D section using previously trained 2D CNN.
previously trained 2D CNN.
Represent each subpart as a triplet of classification results
Represent the shape as set of triplets
Interpreted geometry
End
** 2 series for each one of the halves of the 2D shape created by passing a line parallel with "x" axis through the midpoint of the shape.
*** n is a predetermined value based on the desired accuracy.
**** if the available data is not enough, the remaining points are considered to be of one group

**Figure 1** **Flowchart showing the main structure of the proposed framework.**

Full-size DOI: 10.7717/peerjcs.629/fig-1

having a framework that provides the system with the required flexibility and automation is a necessity. To create such a system, there has to be a flexible method to recognize the shape and to assign a production process based on this recognition. So one of the most important steps to have this system is to have an autonomous intelligent shape interpretation system.

The interpretation of the shape is one of the most crucial steps in the production process, which takes part between conceptual design and manufacturing process; To create an autonomous intelligent shape interpretation structure, a two step framework is proposed that decomposes the shape into shapes which can be more easily recognized in the first step, and in the second step, each of the resulted shapes are recognized.The proposed framework uses simple information about the points, edges and plains of the object and as a result, can do analysis using the STL file format input or simple B-Rep model of shape or more complex CAD representation methods. The main structure of the proposed framework has been depicted in Figs. 1 and 2. The two flowcharts show the same structure and their only difference is the data preparation phase for neural networks where two different approaches are proposed. The details of the framework, with respect to the second approach (Fig. 2) is discussed further in next parts of this section.

## Shape decomposition

The introduction of the design for manufacturing concept made an important contribution in the creation of the links between manufacturing processes and the design procedure (*Wang, Blache & Xu, 2017a*). This paradigm is advantageous in the cases where human

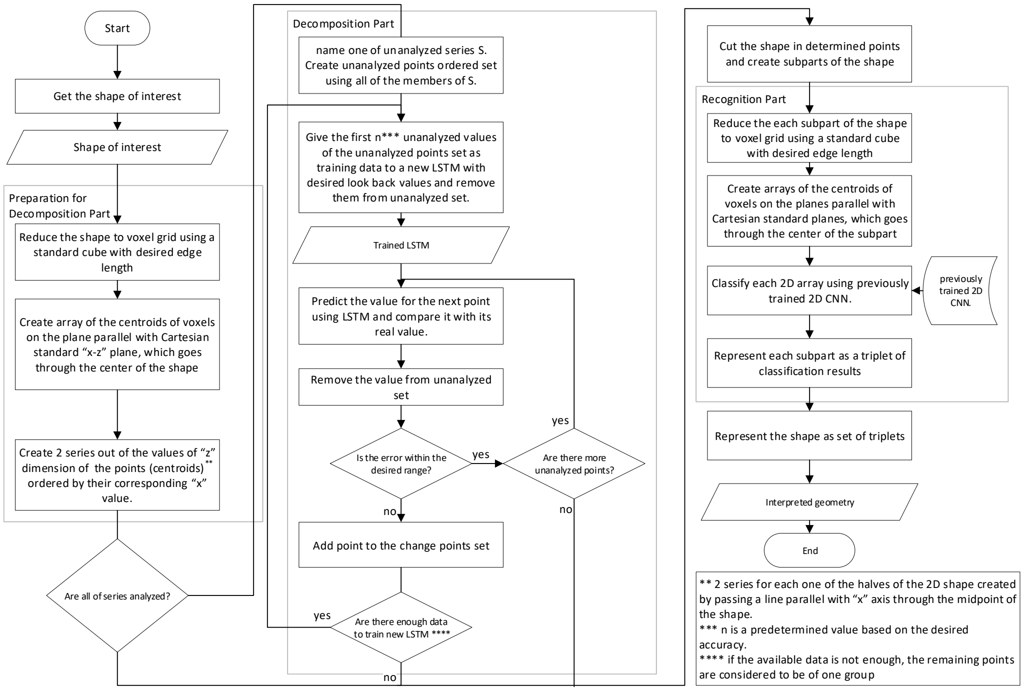
Start
Get the shape of interest
Shape of interest
Preparation for Decomposition Part
Reduce the shape to voxel grid using a standard cube with desired edge length
Create array of the centroids of voxels on the plane parallel with Cartesian standard "x-z" plane, which goes through the center of the shape
Create 2 series out of the values of "z" dimension of the points (centroids)** ordered by their corresponding "x" value.
Are all of series analyzed?
Decomposition Part
name one of unanalyzed series S. Create unanalyzed points ordered set using all of the members of S.
Give the first n*** unanalyzed values of the unanalyzed points set as training data to a new LSTM with desired look back values and remove them from unanalyzed set.
Trained LSTM
Predict the value for the next point using LSTM and compare it with its real value.
Remove the value from unanalyzed set
Is the error within the desired range?
yes
Are there more unanalyzed points?
yes
no
no
Add point to the change points set
yes
Are there enough data to train new LSTM ****
no
Cut the shape in determined points and create subparts of the shape
Recognition Part
Reduce the each subpart of the shape to voxel grid using a standard cube with desired edge length
Create arrays of the centroids of voxels on the planes parallel with Cartesian standard planes, which goes through the center of the subpart
Classify each 2D array using previously trained 2D CNN.
previously trained 2D CNN.
Represent each subpart as a triplet of classification results
Represent the shape as set of triplets
Interpreted geometry
End
** 2 series for each one of the halves of the 2D shape created by passing a line parallel with "x" axis through the midpoint of the shape.
*** n is a predetermined value based on the desired accuracy.
**** if the available data is not enough, the remaining points are considered to be of one group

**Figure 2** **Flowchart showing the main structure of the proposed framework with another data preparation approach.**

Full-size DOI: 10.7717/peerjcs.629/fig-2

agents play the key role, assuring that in the designed part the essential manufacturing factors has been taken into account; but considering the automatic manufacturing, specially from unstructured design data which is common in inverse engineering, it cannot be of any use. In fact there we need a method to interpret the shape, and there, the shape perception models of human cognition can be technically accurate source of inspiration.

The subject of the shape interpretation has been of interest before the introduction of the design for manufacturing concept and a great deal of literature in the process planning field is dedicated to the interpretation of the shape from a manufacturing point of view. In fact, there and in most other cases, the interpretation of the shape means the description of the shape in terms of a variety of some known shapes the attributes of which are known by the system and considering manufacturing point of view, can be produced by rather simple production processes (*Ji & Marefat, 1997*). In the manufacturing point of view, these are called geometrical features (*Ji & Marefat, 1997*). The basic idea of some of feature based shape interpretations can be found in perception models to describe human object recognition too. For example the constructive solid geometry (CSG) model (*Ghali, 2008*), is to some degree similar to the powerful perception model introduced by (*Hummel & Biederman, 1992*).

The classical feature based interpretation had some shortcomings which leaded to its abundance:

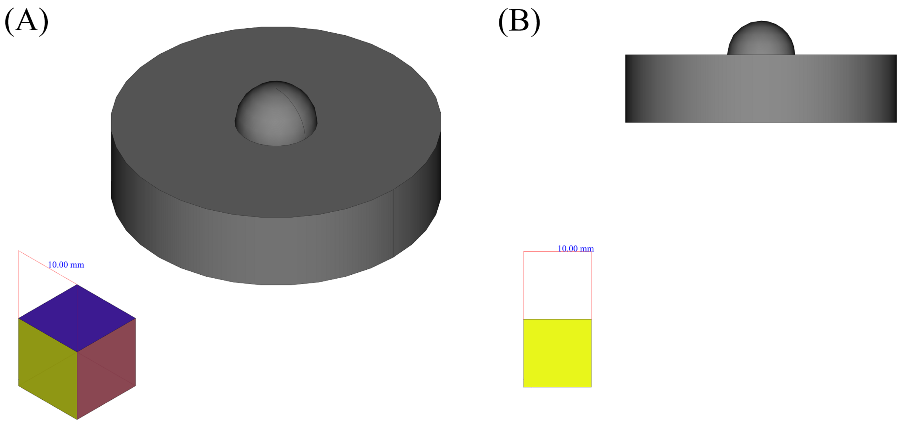

**Figure 3** **An example shape.** (A) Isometric view. (B) Side view.
Full-size DOI: 10.7717/peerjcs.629/fig-3

- Due to the classical artificial intelligent systems which were based on predetermined rules, there had to be rules to identify each feature, the creation of which was not an easy accomplishment.
- The complete definition of the features, required by those systems, made the recognition process restricted and the expansion of the system a difficult process.
- As they were completely based on classical manufacturing methods, their application in additive manufacturing methods become very limited.

Although being problematic in some cases, geometrical features could be a good guide to interpret other shapes and to create knowledge, to analyze the complex shapes and produce them using known methods and considerations. On the other hand, facing the shape as a whole limits the power to use previous knowledge and in most cases, this knowledge would become vain. The method that can be used here, to keep the main concept of geometrical features, while making it more flexible and expandable is to decompose the shape into simpler shapes based on the conformity of the places of their outer edges points. As an example for this method, suppose the shape depicted in Fig. 3. It is obvious that the overall shape of the object is consisted of a cylinder and one half of a sphere. To decompose the shape, the only information used here is the placement of the vertices, faces and edges, *i.e.*, the information that is dealt with in ordinary 3D computer aided drawings, can be gathered through 3D scanning methods and is also available in STL file formats.

The main idea behind the method is to convert the type of one dimension from positional to temporal, and to consider corresponding position as a time series. For a better clarification, now suppose an imaginary scanner, scans the upper part of the shape, in the Cartesian $x-z$ standard plane, and that plane passes through the center of the shape. If the scanner, scans one point per second and saves its relative position regarding the $z$ axis, the placements of the scanned points will create a function of time as depicted in Fig. 4. Now, the critical points would be the points circled red, as they show the sudden changes in the behavior of the points or in better words, they show facing another connected shape; In these words, a simple shape, or geometrical feature in its new definition, would be a

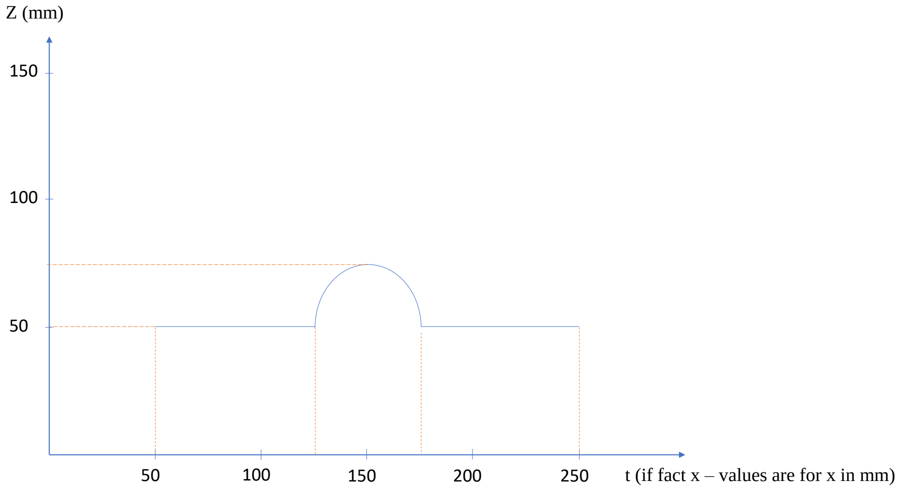

**Figure 4** **The $x-z$ plane section of the shape of Fig. 3.**

Full-size DOI: 10.7717/peerjcs.629/fig-4

group of points, the behavior of which regarding a certain start point and direction can be determined by a pattern. This definition of the simple shape is in accordance with human perceptual apparatus, and to some degree in line with general gestalt rules (*Wagemans, 2015*), (especially continuity rule) and symmetry based definition of shape in *Pizlo (2014)*.

In fact Fig. 4 is a 2D section of the overall lower face of the shape, on the Cartesian $x-z$ plane (suppose that the midpoint of the shape is the reference point of the system), but considering the $x$ axis as a time axis, provides us with a powerful, learner and flexible tool to recognize changes: RNN using Long-Short Term Memory cells (LSTM).

A simple change detection algorithm for changes in behavior using LSTM for a data set $DT$ consisting of a series of data sorted according to time (or any other similar basis, *i.e.*, the position regarding one axis) is proposed in Algorithm 1.

Considering the objects created from simple shapes put together in one direction (namely the direction of coordinate $z$) and can be separated using cutting planes perpendicular to axis $z$ and parallel with $x-y$ Cartesian standard plane, the proposed algorithm would be as described in Algorithm 2.

By applying the SLDA algorithm into two rotations of the shape depicted in Fig. 3, and considering only Changes-1 set, the points circled red in Fig. 5 are resulted.

As is obvious, the resulted changes are not entirely similar: considering the two circled points in one row, as on the picture on the left side, shows an extra change. These errors are expected as the method is completely based on an statistical examination, checking the zero hypothesis of continuity of the previous pattern against the existence of a new one and as the neural networks have their own randomness in behavior, by fixing an architecture for our neural network and a coefficient of error in second step of LCDA algorithm (named $ME$ there) an exchange of type 0 and type 1 errors in detection is decided. The results

**Algorithm 1** LSTM based Change Detection Algorithm (namely "LCDA")

```
1:  Get as an input $DT = \{z_1, z_2, ..., z_n\}$ which are the values of the series , the sudden changes in which is of interest (or in other words values of the edge points on the $z$ axis in the before mentioned example).
2:  Get as an input $DX = \{x_1, x_2, ..., x_n\}$ as the corresponding independent values for $DT$ elements for each of which, i.e. $x_i$, the value of $z_i$ is of interest (or in other words corresponding values of the edge points on the $x$ axis in the before mentioned example, according to which the imaginary point scanner of the example moves).
3:  Get as input parameters $MTR$ as minimum size of data, $W$ as the look-back or window parameter of LSTM and $ME$ as minimum error coefficient.
4:  Get as input an LSTM architecture $architecturelstm$.
5:  Define set $Changes = \{\}$.
6:  Define variable $BGN$ to represent index of the first training data for LSTM for each training iteration. In better words, $z_{BGN}$ is the first data given to LSTM each time it is trained. Assign the initial value as $BGN = 1$.
7:  Define variable $TRSL$ to represent index of the last training data for LSTM for each training iteration. In better words, $z_{TRSL}$ is the last data given to LSTM each time it is trained. Assign the initial value as $TRSL = MTR$.
8:  Define variable $ILD$ to represent placeholder for the data that needs to be checked. Assign the initial value as $ILD = TRSL$.
9:  while $|DT| - BGN > MTR$ and $ILD < |DT|$ do:
10:     Create training data set and name it $TR$ such that $TR = \{z_{BGN}, ...., z_{BGN+TRSL}\}$
11:     Check whether the set $Chx = \{x_i \in x_{BGN} - 1, ...., x_{BGN+TRSL}$ such that: $x_{i+1} \neq x_i + 1\}$ is empty; if not add its members to $Changes$ set. Set $TRSL$ equal to the largest value in $Chx$ and go to step 17[1].
12:     Create check data set and name it $CD$ such that $CD = \{z_{ILD-W-1}, ..., z_{ILD-1}\}$. The predicted value of $z_{ILD}$ using LSTM in each iteration is of interest.
13:     Create an LSTM and name it $lstm$ with predefined architecture, $architecturelstm$ and train it using $TR$.
14:     Use $lstm$ to predict the values of $TR$. Find the mean and standard deviation of absolute value of difference between predicted and real values of $TR$. Name the results $MER$ and $SDTR$ respectively.
15:     Use $lstm$ to predict the value $z_{ILD}$ applying the $CD$ as input. Name the result $zp_{ILD}$.
16:     if $|zp_{ILD} - z_{ILD}| > MER + (ME \times SDTR)$ then
17:         Update $BGN$ such that $BGN = TRSL$.
18:         Append $ILD$ to the $Changes$ set.
19:         Set $TRSL = TRSL + MTR$.
20:         Set $ILD = TRSL$.
21:     else
22:         Set $TRSL = TRSL + 1$.
23:         Set $ILD = TRSL$.
24: Give the $Changes$ set, which is the set of indexes showing the placement of the changes, as the output.
```

**Algorithm 2** Simple LSTM based Decomposition Algorithm (namely "SLDA")

```
Get as an input, the shape of interest and name it $S$.
Get as an input parameter $ml$ which determines the size of the largest edge of the shape to be processed by the algorithm (which is determined relative to the computation power of the used computer).
Create a voxelgrid of $S$ using an standard cube as unit (the edge length of which determines the precision in contrasting different types of objects and bounded by the computing power of the computer). Name the resulted voxelgrid, $VD$ which by definition would be: $VD = [vd]_{(}cl \times cl \times cl)$ where $vd$ would be 0 or 1 and $cl$ is a value larger than or equal to the mathematical ceiling of the $ml$ value divided by the edge length of the unit cube (for complete information on voxelization see Kaufman, Cohen & Yagel (1993)).
Create matrix $XZ = [vd_{(}x, \lfloor cl/2 \rfloor, z)]$.
Define set $Nx = \{x' | \exists z' \text{ such that } v_{x'z'} \neq 0\}$. Make $Nx$ an ordered set by sorting it values.
Find minimum and maximum values of $z$ regarding each $x$ in the indexes of nonzero elements of $XZ$ and create ordered sets $minxz$ and $maxxz$ out of these values sorted according to their $x$ values respectively.[2]
Create $minxzi$ and $maxxzi$ ordered sets by reversing the order in $minxz$ and $maxxz$ respectively.
Apply LCDA algorithm to $minxz$ and $minxzi$ separately; use $Nx$ as the $DX$ input for the algorithm and append the resulted set of indexes together to create set $Changes-1$ (change the indexes in $minxzi$ as to show the respective index of value in $minxz$). Do the same for $maxxz$ and $maxxzi$ and create the set $Changes-2$[3].
Set $TChanges = Changes-1 \cup Changes-2$.
Give $TChanges$, which shows the placement of $x$ dimensions of the changes from a simple shape to another in the voxelgrid $VD$, as an output
Create $Parts$ set from cutting the shape $S$ using planes parallel with $x-y$ plane which go through points of $S$ corresponding to detected changes in $TChanges$.[4]
```

shown before are from assigning value 3 to $ME$ and applying an LSTM neural network with an architecture shown in Fig. 6.

## Recognition algorithm

The decomposition process creates some simpler shapes to be recognized. The next step is to categorize these simpler shapes regarding their similarity with familiar cases. To do so, CNNs are the best candidate to use, as they can do the recognition process with a surprisingly high accuracy (Tong, 2018). Although there are 3D CNNs which could be used to carry out the recognition process in 3D shapes, the proposed framework uses a 2D CNN, taking sections passing through the center of the shape and parallel with each of the Cartesian coordinate system planes $(x-y, y-z, x-z)$ as inputs. This method is applied because:

[1] Obviously i=1 and i = n are not of interest.

[2] This step is crucial to create functions out of points in one direction and an accurate definition of a series based on them.

[3] Although one check of changes can be enough, but as the changes may show off on the either side of the shape both *minxz* and *maxxz* are considered. Also as the change detection algorithm is prone to errors resulted from the batch of initial training data, it is suggested to perform algorithms on both orderings of the sets.

[4] In fact, two cutting planes going through the top and bottom of the corresponding voxels should be used and the shape between them should be omitted; this is due to to the use of voxelgrid.

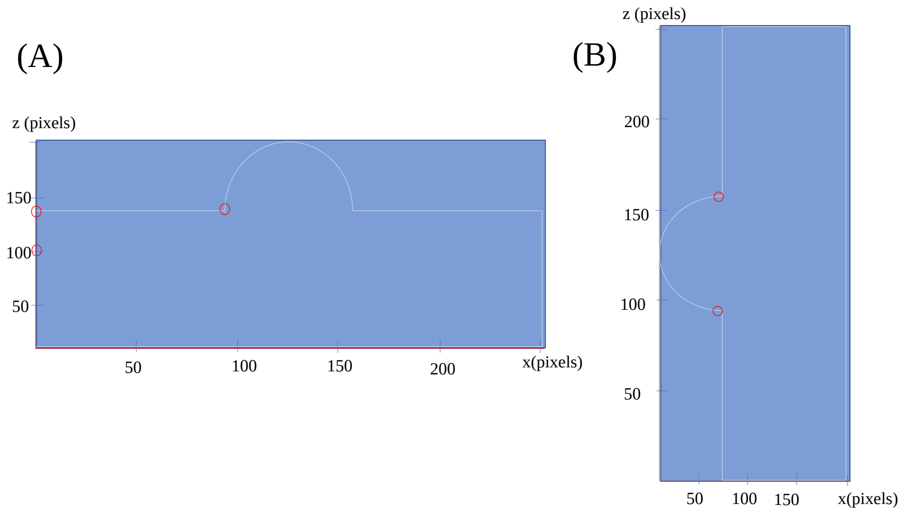

**Figure 5 Changes found for shape in Fig. 3 in two orientation** (A) Hemisphere component facing up. (B) Hemisphere component facing left.

Full-size DOI: 10.7717/peerjcs.629/fig-5

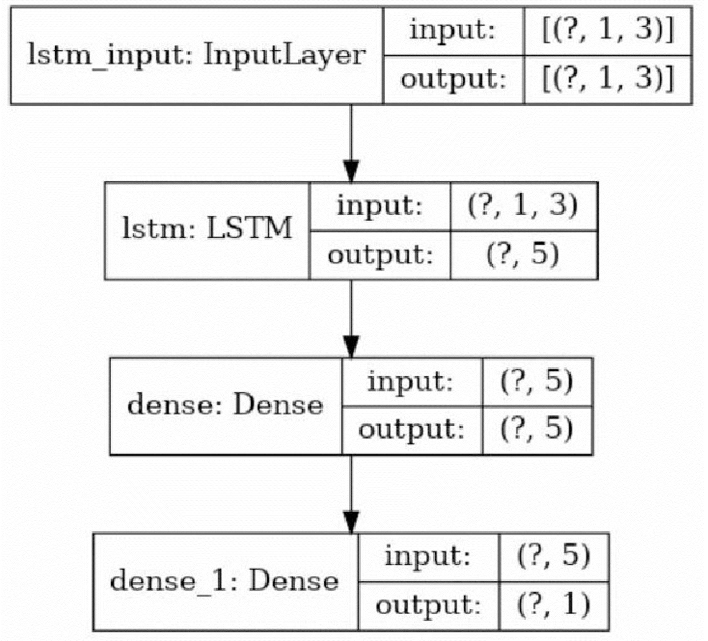

**Figure 6 The architecture of the applied LSTM neural network. The question marks in the cells are due to the consideration of the overall data size as the first element of the array (the picture created using Keras© library in python 3. 8.5©).**

Full-size DOI: 10.7717/peerjcs.629/fig-6

- 3D CNNs are hard to train and are not as common as 2D CNNs, as a result finding good architectures for them is difficult.
- Creating 3D matrices as inputs for 3D CNNs is memory consuming

- In most cases, as the shapes usually have some kind of symmetry, these 2D sections are enough to interpret the shape
- If the input dimensions regarding three Cartesian coordinates have similar upper bound, and are interpreted by matrices of the same sizes, one 2D CNN would be enough.

So a 2D CNN with an architecture depicted symbolically by Fig. 7 is trained and used for the recognition process. Using this neural network, the type of the object is determined by a triplet code, representing the type of each section.

## The main algorithm

Combining the mentioned steps, the overall framework would be Algorithm 3.

**Algorithm 3** Main Algorithm

```
Get as an input, the shape of interest and name it $S$.
Create ordered sets $Parts = \{\}$ and $Types = \{\}$.
Get as an input parameter $ml$ which determines the size of the largest edge of the shape to be processed by the algorithm (which is determined relative to the computation power of the used computer).
Apply the SLDA algorithm and save the resulted shapes in the $Parts$ set.
**for** each $p$ in $Parts$ **do**
    Rescale $p$ such that its largest edge be of length less than or equal to $ml$ .
    Rotate $p$ such that its face with the largest area has a normal parallel to the axis $z$.
    Create a voxelgrid of $p$ using an standard cube as unit (the edge length of which determines the precision in contrasting different types of objects and bounded by the computing power of the computer). Name the resulted voxelgrid, $VD$ which by definition would be: $VD = [vd]_{(}cl \times cl \times cl)$ where $vd$ would be 0 or 1 and $cl$ is a value larger than or equal to the mathematical ceiling of the $m$ value divided by the edge length of the unit cube.
    Create matrices $XY = [vd_{(}x, y, \lfloor cl/2 \rfloor)]$ , $XZ = [vd_{(}x, \lfloor cl/2 \rfloor, z)]$ and $YZ = [vd_{(}\lfloor cl/2 \rfloor, y, z)]$ where $x, y, z \in [1, cl]$.
    Create an ordered triplet $[XY, XZ, YZ]$ using fore mentioned matrices.
    Use the previously trained 2D CNN to determine the type regarding each matrix in $[XY, XZ, YZ]$ and create $[t(XY), t(XZ), t(YZ)]$ where $t(a)$ is the type of $a$ according to the neural network.
    Append the triplet $[t(XY), t(XZ), t(YZ)]$ to the $Types$ set.
Give $Types$ set which shows the type of parts in terms of triplets, as output.
```

This framework provides the compositionality and learnability at the same time; while the neural networks are used to find out the type of the building blocks of the shapes, which can be trained to recognize new basic shapes, the framework decomposes the main shape into simple shapes to be interpretable, without trying to recognize the type of the overall shape.

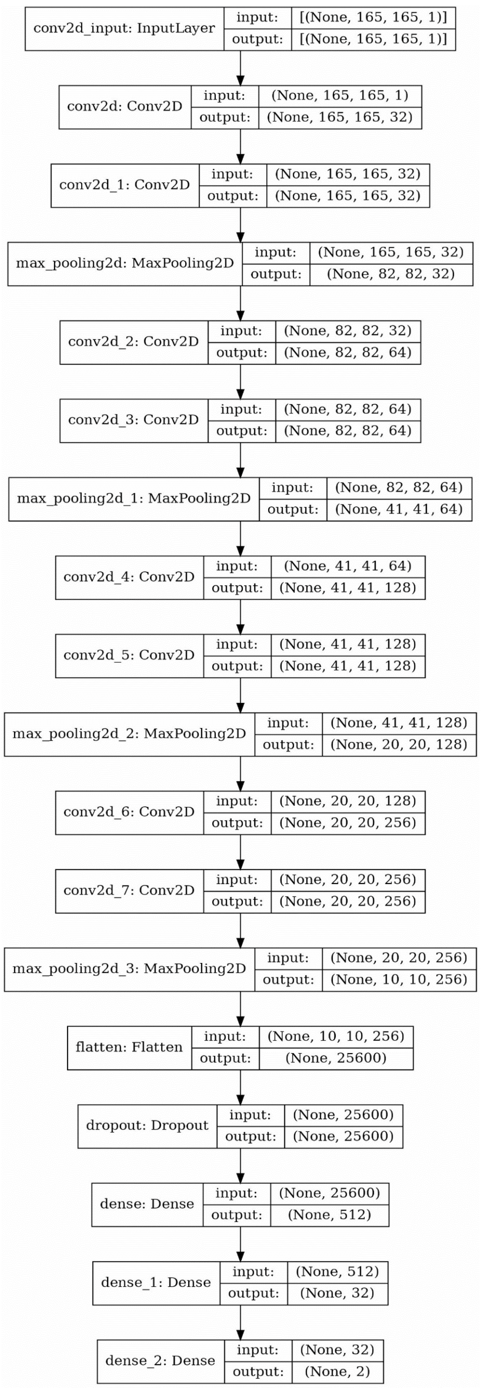
conv2d_input: InputLayer | input: | [(None, 165, 165, 1)] | output: | [(None, 165, 165, 1)]
conv2d: Conv2D | input: | (None, 165, 165, 1) | output: | (None, 165, 165, 32)
conv2d_1: Conv2D | input: | (None, 165, 165, 32) | output: | (None, 165, 165, 32)
max_pooling2d: MaxPooling2D | input: | (None, 165, 165, 32) | output: | (None, 82, 82, 32)
conv2d_2: Conv2D | input: | (None, 82, 82, 32) | output: | (None, 82, 82, 64)
conv2d_3: Conv2D | input: | (None, 82, 82, 64) | output: | (None, 82, 82, 64)
max_pooling2d_1: MaxPooling2D | input: | (None, 82, 82, 64) | output: | (None, 41, 41, 64)
conv2d_4: Conv2D | input: | (None, 41, 41, 64) | output: | (None, 41, 41, 128)
conv2d_5: Conv2D | input: | (None, 41, 41, 128) | output: | (None, 41, 41, 128)
max_pooling2d_2: MaxPooling2D | input: | (None, 41, 41, 128) | output: | (None, 20, 20, 128)
conv2d_6: Conv2D | input: | (None, 20, 20, 128) | output: | (None, 20, 20, 256)
conv2d_7: Conv2D | input: | (None, 20, 20, 256) | output: | (None, 20, 20, 256)
max_pooling2d_3: MaxPooling2D | input: | (None, 20, 20, 256) | output: | (None, 10, 10, 256)
flatten: Flatten | input: | (None, 10, 10, 256) | output: | (None, 25600)
dropout: Dropout | input: | (None, 25600) | output: | (None, 25600)
dense: Dense | input: | (None, 25600) | output: | (None, 512)
dense_1: Dense | input: | (None, 512) | output: | (None, 32)
dense_2: Dense | input: | (None, 32) | output: | (None, 2)

**Figure 7** **The structure of applied 2D CNN. As mentioned before the question marks in the cells are due to the consideration of the overall data size as the first element of the array (the picture created using Keras© library in python 3.8.5©).**

Full-size DOI: 10.7717/peerjcs.629/fig-7

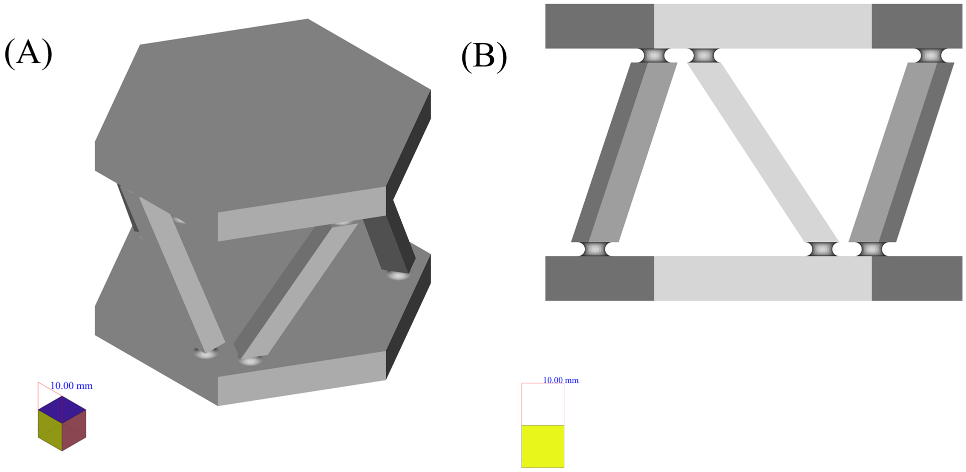

**Figure 8** **Six force sensor based on Stewart platform.** (A) Isometric view. (B) Side view.
Full-size DOI: 10.7717/peerjcs.629/fig-8

## CASE EXAMPLES

In this section to illustrate that how the proposed model works, it has been applied to two cases.

### General considerations

- In both cases CNN with architecture depicted in Fig. 7 has been applied. Activation functions in all layers except the last one were rectified linear unit and in the last layer it was set to Softmax. Optimization method was adaptive subgradient method (*Duchi, Hazan & Singer, 2011*).
- To prepare data to be used as an input for the CNN, the shape has been scaled, to make the length of the largest edge equal with 80 units. The cube with an edge length of 0.5 units is used to create voxelgrid of the shape. Then the shape has been put in a 185 × 185 × 185 array of voxels
- To train the CNN in both cases train data consisting of 4,000 randomly created circles or ellipses as curved or partially curved shapes and 4,000 randomly created rectangles, pentagons or hexagons as linear shapes has been used. The shapes were represented as 185 × 185 arrays. Uniformly distributed random integers have been generated and based on their value each shape for each group was created.
- For the first case, LSTM with architecture depicted Fig. 6 has been applied. Optimization method was RMSprop (*Tieleman & Hinton, 2012*). To initialize LSTM in each iteration, 40 points were used and the look-back value was set to 1.

### Case 1

The first case, that has been chosen to be the input of the proposed framework, is the six force sensor based on Stewart platform depicted in Fig. 8. There were two reasons to choose this shape as a case:

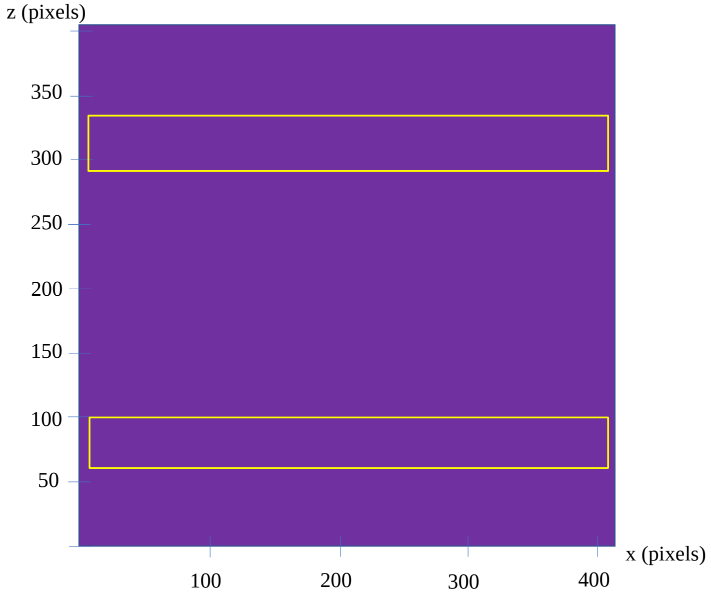

**Figure 9** **The first midsection of the part.**

Full-size DOI: 10.7717/peerjcs.629/fig-9

- The shape is a symmetrical combination of different types of simpler shapes and is able to show the power of the framework in dealing with similar shapes.
- The shape is analyzed as a real case for the application of the manufacturing framework presented in *Zhao & Guo (2020)* where it has been decomposed using structured data available in high level CAD model and based on the curvature of its components, decision upon its manufacturing process has been made. Here it has been shown that how the shape analysis step can be done using less structured data and in a completely autonomous manner.

In the first step, to decompose the shape the framework can be applied in two ways:

1. Applying LCDA algorithm to the 2D mid section of the part as in SLDA algorithm. In this respect the SLDA is improved to act in a divide and conquer manner such that by detection of each change, a perpendicular slicing is done and the algorithm is applied to each of resulted parts.
2. Applying SLDA algorithm but instead of the mid section in step 4, the 2D image of the shape as a whole, on a plane, as depicted in the right hand side of Fig. 8 be used.

Here, both approaches are applied to the example, applying the first approach, at first. To apply the model, the shape has been scaled uniformly, to make the size of the largest edge of the shape, equal with 100 units. Then, the standard cube with the edge length of 0.25 units has been used to create the voxelgrid of the shape. By creating a midsection of the voxelized part as a whole, the 2D image depicted in Fig. 9 is resulted. Applying the

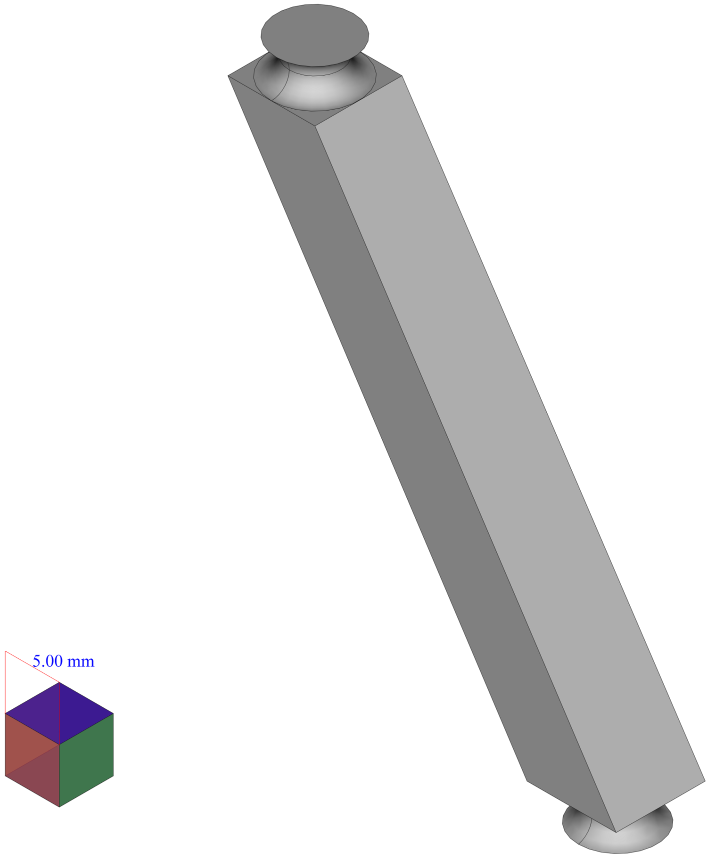

**Figure 10** One of six remaining components.

Full-size DOI: 10.7717/peerjcs.629/fig-10

LCDA, the step 11 of the algorithm, guarantees the separation of the two prismatic ends in a consequent manner.

As these two ends can not be decomposed further, six components with the shape depicted in Fig. 10 remain to be decomposed. Although the algorithm faces them separately (as it is blind to such similarities), only one of them is considered here (the others are faced in the same manner).

The analysis of the depicted components using only one side of the 2D section results in detected changes, which are circled blue in Fig. 11.

As mentioned before, to apply the second approach to analyze this example, the 2D image of the object as shown on the right hand side of the Fig. 8 can be used as an input to SLDA algorithm.

Here a 2D picture of the shape with resolution of $3,000 \times 1,500$ pixels has been used. Applying the second approach, and only applying the algorithm for one side of the picture, the points which are circled red in the Fig. 12 are resulted.

In both approaches, more change points than the real ones are marked; as mentioned before, due to randomness in the behavior of neural networks, in these models a decision upon acceptance interval for a statistical examination should be done considering the

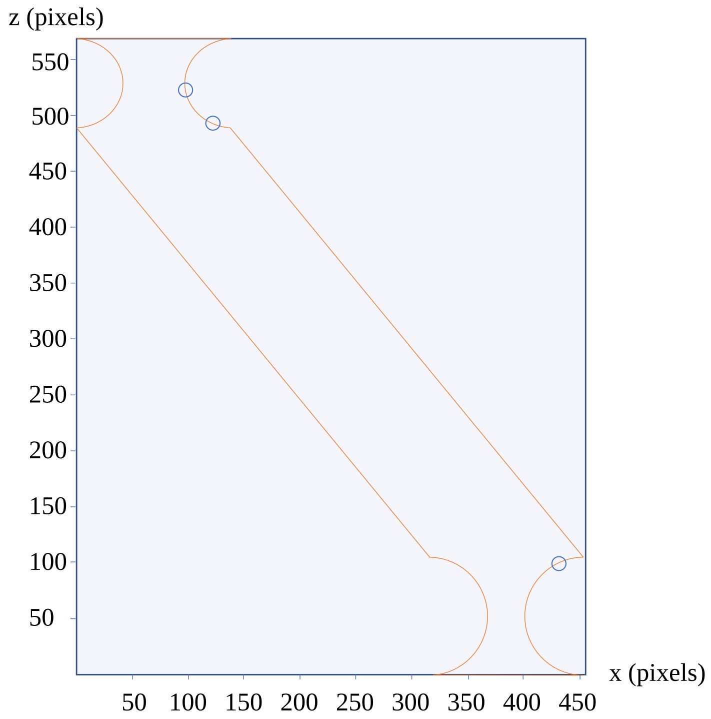

**Figure 11** **The results of the analysis of the component depicted in Fig. 10.**
Full-size DOI: 10.7717/peerjcs.629/fig-11

errors type 0 and 1; as the larger errors in recognizing excessive points as change points, is much less important than error of not recognizing the real change points, the acceptance interval is configured for less error of not recognizing which results in excessive points, marked as change points. Another reason for the excessive change points is disruptions created by representation of the continuous lines by discrete pixels.

The next step would be the recognition of the shapes based on the presupposed standard. As mentioned before, the applied method is to create three 2D sections of the parts using planes parallel with each of the Cartesian standard planes which pass through the center of the shapes. In the first step, a preparation process as described in section 'General considerations', has to be conducted which results in matrices with standard sizes that can be used as an input to a 2D CNN with a fixed size of input layer. The resulted voxelgrid which can be interpreted as a 3D matrix can be sliced into three 2D matrices by fixing a number for each of its coordinate indexes; and as if these numbers be the middle of the maximum and minimum values of that index, the concerned 2D sections would be resulted. The resulted 2D sections for each shape is depicted underneath it in the Fig. 13 (The resulted matrices are depicted as 2D shapes in which the yellow pixels represent value 1 and other pixels represent value 0). Having the matrix representation for the three sections of concern, their type would be determined using a previously trained 2D CNN; Based on the type of the data and the required precision and the determination power for different axes, different neural networks may be used, however a single neural network trained with different part types is enough to determine the types of the sections. Using

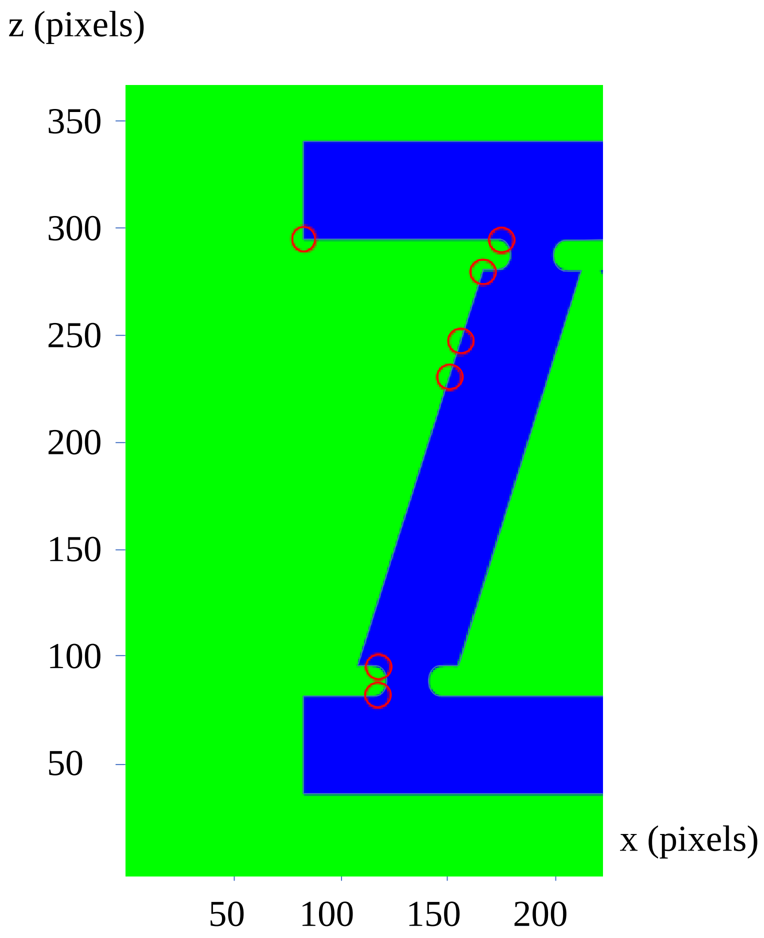

**Figure 12** **The results of the analysis of the 2D image of the shape.**
Full-size DOI: 10.7717/peerjcs.629/fig-12

a neural network trained by two different types of 2D matrices, a type representing multi linear (partially or completely) sections (namely type "01") and another representing circular ones (namely type "00"), the framework determines the type of each section as depicted in the last row of Fig. 13.

### Case 2

Another example that shows the applicability of the framework based on convex decomposition proposed in Appendix 1 is depicted in Fig. 14. This shape consists of 3 boxes and 3 cylinders connected to them. The reasons for choosing the shape in Fig. 14 are:

- It is a combination of many components and there are many cases in the real world where combinations of cylinders or cones and boxes, create the main outline of the shape.
- It can depict the power of the framework being within the limitations of the framework. In general terms, shapes which become problematic using the convex decomposition proposed in Appendix 1 include:

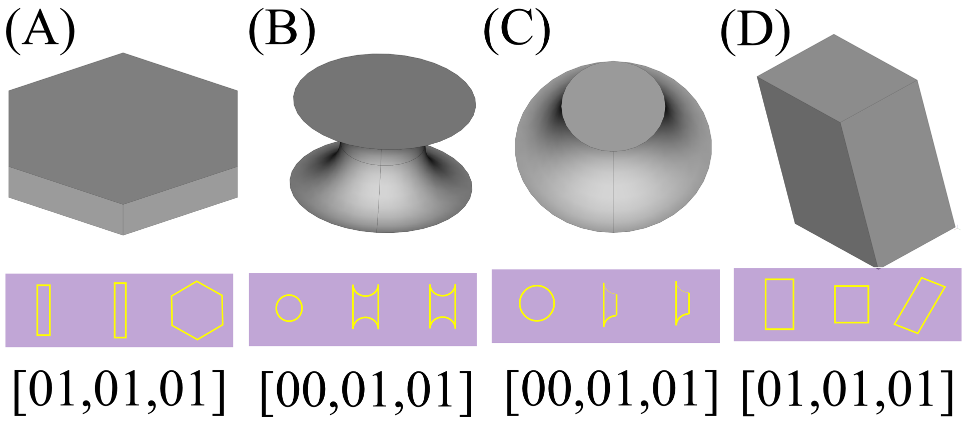

**Figure 13** **The 2D sections of the components.** Pictures (A) to (D) depict one of each type of resulted components and their sections and the results of the recognition phase.
Full-size DOI: 10.7717/peerjcs.629/fig-13

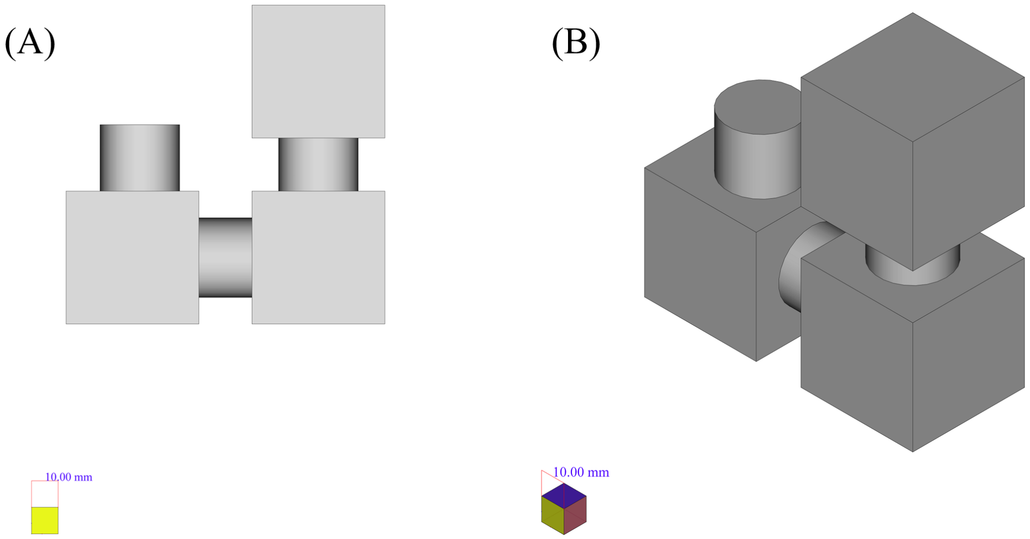

**Figure 14** **The initial shape of the example.** (A) Side view. (B) Isometric view.
Full-size DOI: 10.7717/peerjcs.629/fig-14

- Shapes which include sphere or torus or a part of them. The problem here happens due to the tessellated representation of these shapes consists of large number of facets and as the analysis in the convex decomposition algorithm is based on facets, this multitude of facets, facing many directions can cause unwanted cuts.
- Shapes consisting of the intersection of two (or more) shapes with curved cross sections. Here The problem happens due to the inseparability of these intersections using facets.

The first step is to decompose this shape into its convex components. To do this the shape is tessellated to small planar triangles covering the shape. The resulted triangles would be accounted as one with each other with respect to their coplanarity, and the resulted

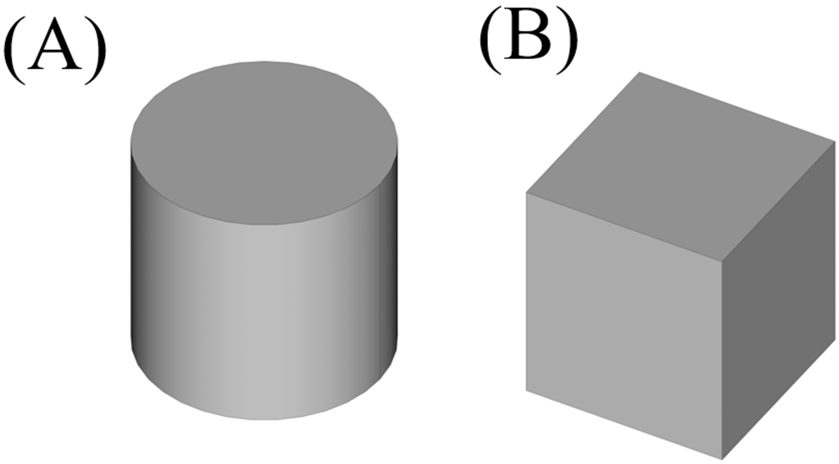

**Figure 15** **The resulted convex shapes from the convex decomposition of the initial shape.** (A) Cylinder which connects the cubes (B) Cube.

Full-size DOI: 10.7717/peerjcs.629/fig-15

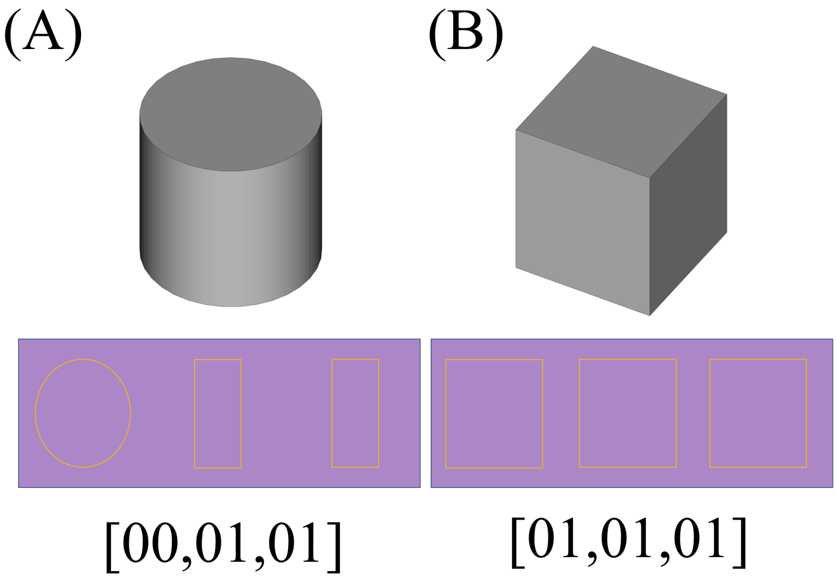

**Figure 16** **2D sections of each of the convex parts of the original shape.** (A) 2D sections and the results of the recognition phase for each of the cylindrical components. (B) Results for each of the cubic components.

Full-size DOI: 10.7717/peerjcs.629/fig-16

planes would be used to decompose the shape into its convex parts respectively. Applying this method, the model decomposed the shape in Fig. 14 into separate shapes. As resulting shapes are similar, only two types of the resulted components are depicted in Fig. 15.

The resulted 2D sections for each shape is depicted underneath it in the Fig. 16 (like the previous example, the resulted matrices are depicted as 2D shapes in which the yellow pixels represent value 1 and other pixels represent value 0). By applying the recognition CNN applied in the previous case, the results would be as depicted in the last row of Fig. 16.

As mentioned before, the two fold framework proposed by this paper can be a preliminary step in micro and macro process planning of additive manufacturing products. The results of the categorizations, can be considered as features used by different viewpoints of process planning without need to define them explicitly; in fact, categories considered in the examples were intended to be simple cases, for the illustrative purpose of these examples. 2D CNN's are one of the most powerful tools in the categorization of images (*Tong, 2018*) and by enriching their training data and some changes in their architecture, they are capable of categorizing a vast variety of images.

## CONCLUSION AND DISCUSSION

The cases analyzed in section "Case examples" are examples showing the power of the proposed framework in different regards including:

- Considering the input structure: the framework can perform the analysis using simple design information. In fact, the information provided in a 2D image of a symmetrical shape, taken from a viewpoint that shows the changes in the behavior of the outer edges of the shape is enough.
- Considering the decomposition phase: the decomposition is done, just using data provided in the shape itself and without need for additional training data. The geometric positions of the points on the shape is enough to decompose the shape.
- Considering the recognition phase: the recognition or categorization is accomplished using 2D CNN trained by data which represent simple attributes. The overall structure of the shape is represented by a combination of these simple attributes. This makes the preparation of the training data easier while representing a method to interpret more complex attributes as composition of simpler ones.
- Considering the application phase: each of the recognized curved or linear components can be manufactured applying special parameter level considerations based on their type (*Zhao & Guo, 2020*). From a higher level manufacturing technique viewpoint, some methods have better outputs facing curved surfaces (*Zhang et al., 2020*) and in a cloud platform where many techniques are available, such a framework can help to decide upon the technique autonomously.

In general terms, as it has been shown in section "Case examples" a divide and conquer viewpoint in the interpretation of complex shapes, can provide us with a powerful tool to interpret a vast variety of shapes. Regarding the recognition phase, although the power of convolutional neural networks is dominant, the simple categorization of the sections as to be completely circular versus partially or completely multi linear, is concerned and the outputs were in accordance with the expectations.

The framework proposed in this paper, creates a basement for an autonomous intelligent additive manufacturing system which can get a design in STL (or any other 3D design) file format, and interpret it as a union of some simpler shapes, regarding the desired geometrical attributes of which, the decision upon the production process can be made. In fact, the resulted categories would provide the system with information about the geometrical structure of the product that is of much concern in the process planning for the product.

One potential approach to expand this work is based on the learning capability: The recognition algorithm of this structure can be applied by using only two initial categories and some examples for each category; by the passage of time the data categorized by the neural network can be labeled based on the accuracy of categorization. The inaccurately categorized data can be braked into some groups by a clustering based method and when the data volume in each of the groups raises to a lower bound, these group of data be added to the training data of the neural network as a new category. Applying these steps, the proposed framework can learn new categories during its application.

To expand the shape decomposition part, it should be reminded that the proposed algorithm deals with simple shapes which meet the explained conditions. To improve the algorithm to be applicable to all types of shapes, two extensions are needed to be made:

- Improve the algorithm to be more sensitive to the changes in 2D sections images. The algorithm in the proposed form in Section 3, considers the changes to be only in the direction $z$ of the Cartesian coordinate system (or in better words, by an accurate definition of the coordinate system, they would be so). However, in cases such as the propeller depicted in Fig. 17 there is no such a definition of the coordinate system that meets this condition.
- Apply the algorithm for the change detection not only to a single 2D section of the shape. Obviously, to deal with shapes with changes in different directions, most of the times, one 2D section is not enough. As an example consider the propeller of Fig. 17. Some of sections in this shape, which have been created by planes perpendicular to the $z$ axis and parallel with $x-y$ plane are as depicted in Fig. 18. These pictures show that these simple cuttings would not be enough to decompose the shape and detected changes in each section should be connected in a special manner to create accurate cutting planes to simplify the shape.

As an example of the desired output of the algorithm, suppose that we want to cut off one of the blades of the propeller. To illustrate the desired cut, consider Fig. 19. In Fig. 19 analysis in each of the perpendicular sections would result in the blue points (blue line segments show a section of the perpendicular section plane) and the red lines create the desired cut in 3D space (obviously with an error corresponding to the space between the lines as the behavior between two planes considered to be linear).

The expansion of the decomposition algorithm to solve such cases is the other related research potential. As the last proposed related research potential, it is recommended that the categorization outputs become inputs to a model, algorithm or another intelligent structure, to determine the production method based on the attributes of each category.

## APPENDIX A1. CONVEX DECOMPOSITION AS A SIMPLER METHOD FOR DECOMPOSITION

Regarding the bounded randomness of the behavior of the neural networks (or other change detection methods), and considering the primitive based structure of the most

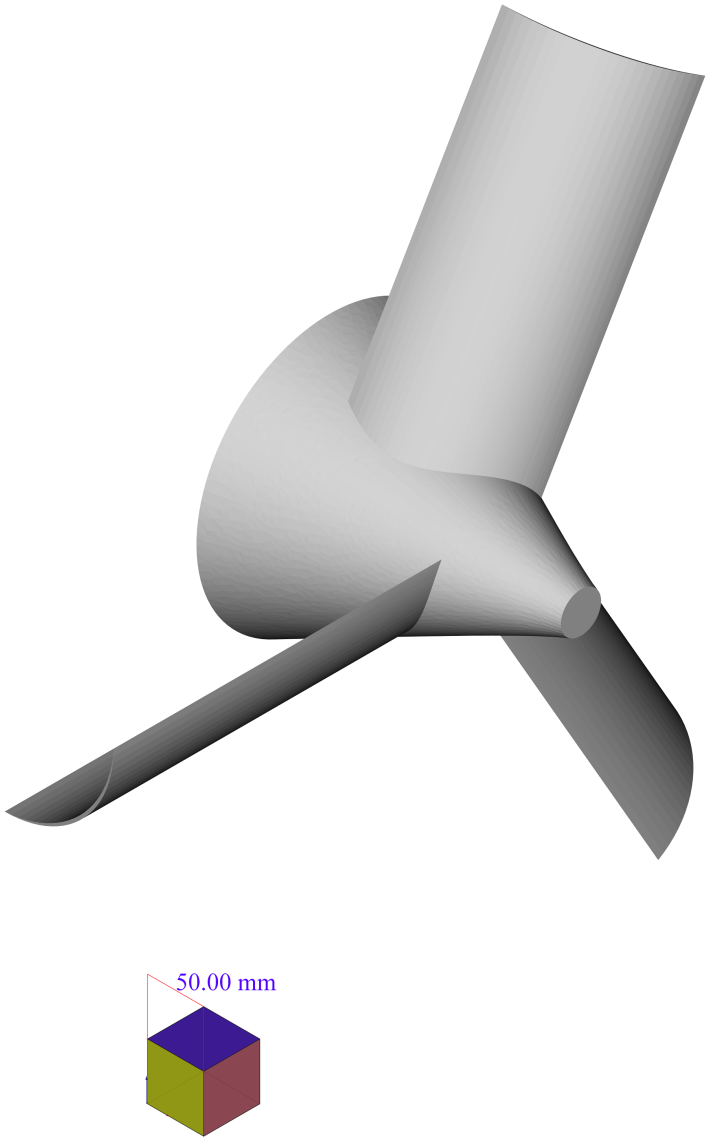

**Figure 17** A simple propeller.

Full-size DOI: 10.7717/peerjcs.629/fig-17

products, in this appendix the convex parts of the shape are considered as the building blocks to be recognized. This point of view, creates a good intermediary level between fixed, predetermined basic features and the consideration of the shape as a whole because:

- The convex parts of the shape can be recognized using the information provided by STL (or other B-Rep) representations of the shape, using simple and general rules and assessments and there are many exact and heuristic methods to decompose the shape into its convex parts.
- Many convex parts in the real world are somehow modified versions of the basic CSG convex features. In fact, in many cases, the shape consists of some convex simple shapes

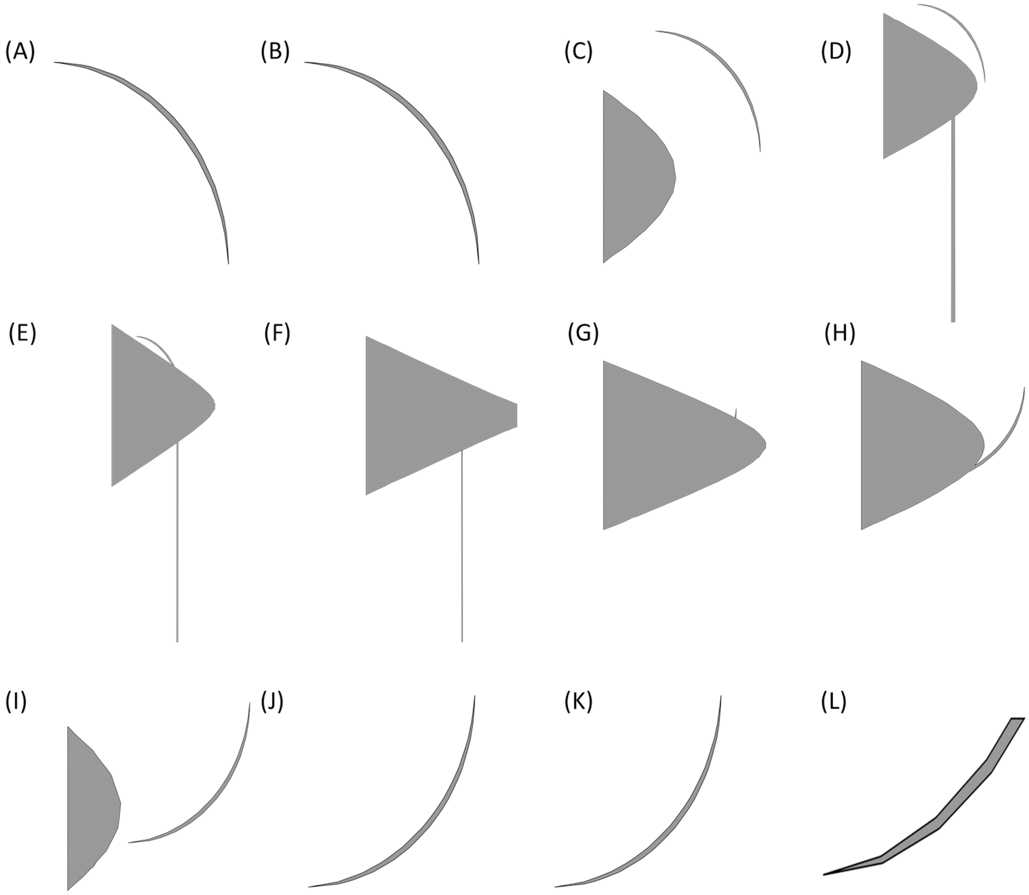

**Figure 18 2D sections of the propeller.** Pictures (A) to (L) show sections created by planes cutting the propeller in different heights.

Full-size DOI: 10.7717/peerjcs.629/fig-18

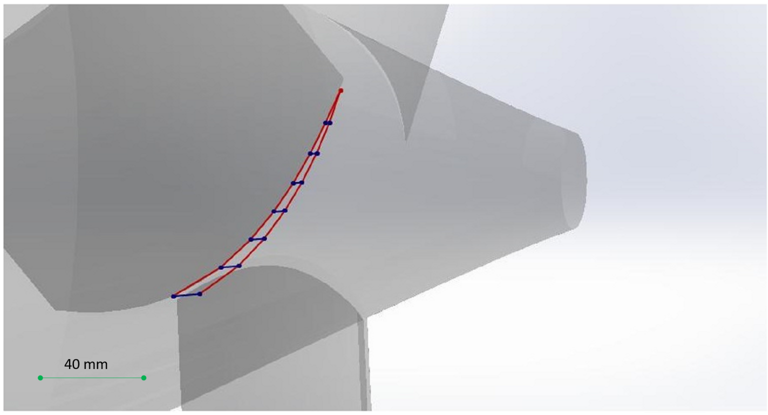

**Figure 19 The semi-curved shape of the desired cut.**

Full-size DOI: 10.7717/peerjcs.629/fig-19

that are similar to CSG convex features to a degree; but this similarity does not mean being limited to these features. The known building blocks in the proposed system can be expanded automatically by adding them to the dataset of the framework.
- From production point of view, the most of convex parts can be created by additive manufacturing using an accurate rotation without need for external base creation (complete sphere is an important exception).

As a result, at the first step, the shape is decomposed into its convex parts using the following algorithms. The “CDA” algorithm can substitute the data preparation and decomposition sections in Figs. 1 and 2:

**Algorithm 4** Slicer Algorithm (namely "SA")

```
Get as an input, the shape of interest and name it $S$.
Tessellate the faces of the input shape (reduce the faces into flat triangular facets of a predetermined maximum edge length). Name the resulted shape $TS$ and the set of the resulted triangles $STS$.
Find the set of maximal subsets of $STS$ considering the coplanarity of the triangular facets. Name the resulted set $CSTS$.
For each element of the $CSTS$ find the covering super plane using the normal vector and one point of the plane (which are the normal and one point of one of its members respectively). Name the set of resulting planes $SP$.
for each plane $pl \in SP$ : do
    Create set $SLP = \{\}$.
    (Optional) If the plane $pl$ slices $S$ into at least two distinct shapes and doesn’t cut through any edges of $TS$ except in the vertices of the $TS$, add the resulted shapes into $SLP$; give $SLP$ as the output and finish the algorithm.
    If the plane $pl$ slices $S$ into at least two distinct shapes, add the resulted shapes into $SLP$; give $SLP$ as the result and finish the algorithm.
```

**Algorithm 5** Convex Decomposition Algorithm (namely "CDA")

```
Get the input shape and name it $O$. Create ordered sets $Convexshapes = \{\}$and $Concaveshapes = \{O\}$.
while the cardinality of the $Concaveshapes$ set is not zero: do
    Apply the SA algorithm on the first member of the $Concaveshapes$, which we name $CV$ and name the result set $RS$.
    if $RS$ is not empty then
        Append the members of $RS$ to the end of the $Concaveshapes$ set.
    else
        Move $CV$ from $Concaveshapes$ set to $Convexshapes$ set.
Give $Convexshapes$ set as the output.
```

## ACKNOWLEDGEMENTS

The authors appreciate the support of the Python©language team and also the following Third party Python libraries: Keras© , Matplotlib©, Trimesh© , FreeCAD© , Numpy© and Scipy© .

## ADDITIONAL INFORMATION AND DECLARATIONS

### Funding

The authors received no funding for this work.

### Competing Interests

The authors declare there are no competing interests.

### Author Contributions

- Mohammad reza Rezaei conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Mahmoud Houshmand analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.
- Omid Fatahi Valilai conceived and designed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.

### Data Availability

The following information was supplied regarding data availability:

The developed software package is available at figshare: Fatahi Valilai, Omid (2021): 3D geometry intrepration. figshare. Software. https://doi.org/10.6084/m9.figshare.14016038.v1.

The case studies are available at figshare: Fatahi Valilai, Omid (2021): 3D object samples. figshare. Figure. https://doi.org/10.6084/m9.figshare.14016122.v1.

## REFERENCES

**Ajit A, Acharya K, Samanta A. 2020.** A review of convolutional neural networks. In: *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*. Vellore, India: IEEE, 1–5.

**Baumann F, Roller D. 2017.** Additive manufacturing, cloud-based 3D printing and associated services–overview. *Journal of Manufacturing and Materials Processing* **1**(**2**):15 DOI 10.3390/jmmp1020015.

**Biegelbauer G, Vincze M, Wohlkinger W. 2008.** Model-based 3D object detection. *Machine Vision and Applications* **21**(**4**):497–516 DOI 10.1007/s00138-008-0178-3.

**Buckley PJ. 2009.** The impact of the global factory on economic development. *Journal of World Business* **44**(**2**):131–143 DOI 10.1016/j.jwb.2008.05.003.

**Campbell I, Bourell D, Gibson I. 2012.** Additive manufacturing: rapid prototyping comes of age. *Rapid Prototyping Journal* **18(4)**:255–258 DOI 10.1108/13552541211231563.

**Chan SL, Lu Y, Wang Y. 2018.** Data-driven cost estimation for additive manufacturing in cybermanufacturing. *Journal of Manufacturing Systems* **46**:115–126 DOI 10.1016/j.jmsy.2017.12.001.

**Conner BP, Manogharan GP, Martof AN, Rodomsky LM, Rodomsky CM, Jordan DC, Limperos JW. 2014a.** Making sense of 3-D printing: creating a map of additive manufacturing products and services. *Additive Manufacturing* **1**:64–76 DOI 10.1016/j.addma.2014.08.005.

**Davenport TH, Ronanki R. 2018.** Artificial intelligence for the real world. *Harvard Business Review* **96(1)**:108–116.

**Duchi J, Hazan E, Singer Y. 2011.** Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* **12(null)**:2121–2159 DOI 10.5555/1953048.2021068.

**Ghali S. 2008.** Constructive solid geometry. In: *Introduction to geometric computing*. London: Springer London, 277–283.

**Goh GD, Sing SL, Yeong WY. 2020.** A review on machine learning in 3D printing: applications, potential, and challenges. *Artificial Intelligence Review* 1–32 DOI 10.1007/s10462-020-09876-9.

**Goodarzi EV, Houshmand M, Valilai OF, Ghezavati V, Bamdad S. 2020.** Manufacturing cloud service composition based on the non-cooperative and cooperative game theory. In: *2020 IEEE international conference on industrial engineering and engineering management (IEEM)*. Singapore: IEEE, 1122–1125 DOI 10.1109/IEEM45057.2020.9309921.

**Gunasekaran A, Yusuf YY, Adeleye EO, Papadopoulos T, Kovvuri D, Geyi DG. 2019.** Agile manufacturing: an evolutionary review of practices. *International Journal of Production Research* **57(15-16)**:5154–5174 DOI 10.1080/00207543.2018.1530478.

**Guo L, Qiu J. 2018.** Combination of cloud manufacturing and 3D printing: research progress and prospect. *The International Journal of Advanced Manufacturing Technology* **96(5–8)**:1929–1942 DOI 10.1007/s00170-018-1717-3.

**Hartman S. 2020.** Maximize additive manufacturing with the cloud-based AM network. *Available at https://blogs.sw.siemens.com/additive/cloud-based-additive-manufacturing/* (accessed on 21 May 2021).

**Hummel JE, Biederman I. 1992.** Dynamic binding in a neural network for shape recognition. *Psychological Review* **99(3)**:480 DOI 10.1037/0033-295X.99.3.480.

**Ji Q, Marefat MM. 1997.** Machine interpretation of CAD data for manufacturing applications. *ACM Computing Surveys* **29(3)**:264–311 DOI 10.1145/262009.262012.

**Kaufman A, Cohen D, Yagel R. 1993.** Volume graphics. *Computer* **26(7)**:51–64 DOI 10.1109/MC.1993.274942.

**Kiron D, Schrage M. 2019.** Strategy for and with AI. *MIT Sloan Management Review* **60(4)**:30–35.

**Lake BM, Ullman TD, Tenenbaum JB, Gershman SJ. 2017.** Building machines that learn and think like people. *Behavioral and Brain Sciences* **40**:e253 DOI 10.1017/S0140525X16001837.

**Mai J, Zhang L, Tao F, Ren L. 2016.** Customized production based on distributed 3D printing services in cloud manufacturing. *The International Journal of Advanced Manufacturing Technology* **84(1-4)**:71–83 DOI 10.1007/s00170-015-7871-y.

**Maidin S, Campbell R, Pei E. 2012.** Development of a design feature database to support design for additive manufacturing. *Assembly Automation* **32**:235–244 DOI 10.1108/01445151211244375.

**Mehrpouya M, Dehghanghadikolaei A, Fotovvati B, Vosooghnia A, Emamian SS, Gisario A. 2019.** The potential of additive manufacturing in the smart factory industrial 4.0: a review. *Applied Sciences* **9(18)**:3865 DOI 10.3390/app9183865.

**Miyagi R, Aono M. 2017.** Sliced voxel representations with LSTM and CNN for 3D shape recognition. In: *2017 Asia-pacific signal and information processing association annual summit and conference (APSIPA ASC)*. Kuala Lumpur: IEEE, 320–323 DOI 10.1109/APSIPA.2017.8282044.

**Moroni G, Syam WP, Petrò S. 2014.** Towards early estimation of part accuracy in additive manufacturing. *Procedia CIRP* **21**:300–305 DOI 10.1016/j.procir.2014.03.194.

**Munguía J, Ciurana J, Riba C. 2009.** Neural-network-based model for build-time estimation in selective laser sintering. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **223(8)**:995–1003 DOI 10.1243/09544054JEM1324.

**Pham G, Lee S-H, Kwon O-H, Kwon K-R. 2018.** Anti-3D weapon model detection for safe 3D printing based on convolutional neural networks and D2 shape distribution. *Symmetry* **10(4)**:90 DOI 10.3390/sym10040090.

**Piili H, Happonen A, Väistö T, Venkataramanan V, Partanen J, Salminen A. 2015.** Cost estimation of laser additive manufacturing of stainless steel. *Physics Procedia* **78**:388–396 DOI 10.1016/j.phpro.2015.11.053.

**Pizlo Z. 2014.** Making a machine that sees like us. Oxford: Oxford University Press.

**Ranjan R, Samant R, Anand S. 2017.** Integration of design for manufacturing methods with topology optimization in additive manufacturing. *Journal of Manufacturing Science and Engineering* **139(6)**:061007 DOI 10.1115/1.4035216.

**Rudolph J-P, Emmelmann C. 2017.** A cloud-based platform for automated order processing in additive manufacturing. *Procedia CIRP* **63**:412–417 DOI 10.1016/j.procir.2017.03.087.

**Schwab K. 2016.** *The fourth industrial revolution*. New York: Crown Business.

**Scime L, Beuth J. 2018.** A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing* **24**:273–286 DOI 10.1016/j.addma.2018.09.034.

**Shi C, Zhang L, Mai J, Zhao Z. 2017.** 3D printing process selection model based on triangular intuitionistic fuzzy numbers in cloud manufacturing. *International*

*Journal of Modeling, Simulation, and Scientific Computing* **08**(**02**):1750028 DOI 10.1142/S1793962317500283.

**Shi Y, Zhang Y, Baek S, Backer WD, Harik R. 2018.** Manufacturability analysis for additive manufacturing using a novel feature recognition technique. *Computer-Aided Design and Applications* **15**(**6**):941–952 DOI 10.1080/16864360.2018.1462574.

**Simeone A, Caggiano A, Zeng Y. 2020.** Smart cloud manufacturing platform for resource efficiency improvement of additive manufacturing services. *Procedia CIRP* **88**:387–392 DOI 10.1016/j.procir.2020.05.067.

**Thomas D. 2013.** Economics of the us additive manufacturing industry. *NIST Special Publication* **1163**.

**Tieleman T, Hinton G. 2012.** Lecture 6.5—RmsProp: divide the gradient by a running average of its recent magnitude. **4**(**2**):26–31 COURSERA: neural Networks for Machine Learning.

**Tong F. 2018.** Foundations of Vision. In: Wixted JT, ed. *Stevens' handbook of experimental psychology and cognitive neuroscience.* Hoboken: John Wiley & Sons, Inc., 1–61 DOI 10.1002/9781119170174.epcn201.

**Vaidya S, Ambad P, Bhosle S. 2018.** Industry 4.0 A Glimpse. *Procedia Manufacturing* **20**:233–238 DOI 10.1016/j.promfg.2018.02.034.

**Valilai OF, Houshmand M. 2015.** Depicting additive manufacturing from a global perspective; using Cloud manufacturing paradigm for integration and collaboration. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **229**(**12**):2216–2237 DOI 10.1177/0954405414546706.

**Vaneker T, Bernard A, Moroni G, Gibson I, Zhang Y. 2020.** Design for additive manufacturing: framework and methodology. *CIRP Annals* **69**(**2**):578–599 DOI 10.1016/j.cirp.2020.05.006.

**Wagemans J. 2015.** *Oxford handbook of perceptual organization.* Oxford: Oxford University Press.

**Wang Y, Blache R, Xu X. 2017a.** Design for additive manufacturing in the cloud platform. In: *Volume 3: manufacturing equipment and systems*. Los Angeles, California, USA: American Society of Mechanical Engineers, V003T04A029 DOI 10.1115/MSEC2017-2708.

**Wang Y, Blache R, Xu X. 2017b.** Selection of additive manufacturing processes. *Rapid Prototyping Journal* **23**(**2**):434–447 DOI 10.1108/RPJ-09-2015-0123.

**Wang C, Tan X, Tor S, Lim C. 2020.** Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing* **36**:101538 DOI 10.1016/j.addma.2020.101538.

**Wang Y, Zheng P, Xu X, Yang H, Zou J. 2019.** Production planning for cloud-based additive manufacturingA computer vision-based approach. *Robotics and Computer-Integrated Manufacturing* **58**:145–157 DOI 10.1016/j.rcim.2019.03.003.

**Williams G, Meisel NA, Simpson TW, McComb C. 2019.** Design repository effectiveness for 3D convolutional neural networks: application to additive manufacturing. *Journal of Mechanical Design* **141**(**11**):111701 DOI 10.1115/1.4044199.

**Wohlkinger W, Aldoma A, Rusu RB, Vincze M. 2012.** 3DNet: large-scale object class recognition from CAD models. In: *2012 IEEE International Conference on Robotics and Automation*. 5384–5391 DOI 10.1109/ICRA.2012.6225116.

**Xu X. 2012.** From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing* **28**(**1**):75–86 DOI 10.1016/j.rcim.2011.07.002.

**Yao X, Moon SK, Bi G. 2017.** A hybrid machine learning approach for additive manufacturing design feature recommendation. *Rapid Prototyping Journal* **23**:983–997 DOI 10.1108/rpj-03-2016-0041.

**Yin J, Huang N, Tang J, Fang M. 2020.** Recognition of 3D Shapes Based on 3V-DepthPano CNN. *Mathematical Problems in Engineering* **2020**:1–11 DOI 10.1155/2020/7584576.

**Yu Y, Si X, Hu C, Zhang J. 2019.** A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation* **31**(**7**):1235–1270 DOI 10.1162/neco_a_01199.

**Zhang H, Liu D, Huang T, Hu Q, Lammer H. 2020.** 3D printing method of spatial curved surface by continuous natural fiber reinforced composite. *IOP Conference Series: Materials Science and Engineering* **782**:022059 DOI 10.1088/1757-899X/782/2/022059.

**Zhao D, Guo W. 2019.** Mixed-layer adaptive slicing for robotic Additive Manufacturing (AM) based on decomposing and regrouping. *Journal of Intelligent Manufacturing* **31**:985–1002 DOI 10.1007/s10845-019-01490-z.

**Zhao G, Ma G, Feng J, Xiao W. 2018.** Nonplanar slicing and path generation methods for robotic additive manufacturing. *The International Journal of Advanced Manufacturing Technology* **96**(**9–12**):3149–3159 DOI 10.1007/s00170-018-1772-9.

**Zohdi T. 2018.** Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning. *Computer Methods in Applied Mechanics and Engineering* **331**:343–362 DOI 10.1016/j.cma.2017.11.030.