A hybrid learning-driven computer vision framework for reverse engineering via enhanced 3D shape reconstruction

Mahmoud Salem[1[0000-0003-2459-0271]], Ahmed Elkaseer[1,2[0000-0002-2500-3617]], Andrew Rees[3] and Steffen G. Scholz[1,2,3[0000-0002-7901-0919]]

[1] Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology, 76344 Eggenstein-Leopoldshafen, Germany

[2] Karlsruhe Nano Micro Facility, Karlsruhe Institute of Technology, 76344 Eggenstein-Leopoldshafen, Germany

[3] Department of Mechanical Engineering, Swansea University, Swansea SA1 8EN, UK

mahmoud.salem@kit.edu

Abstract. Reverse engineering (RE) has played a key role in producing low demands parts, especially with the recent advent of robust additive manufacturing (AM) techniques. The synergetic interaction of both cutting-edge RE and AM techniques significantly enhance part re-producing and minimize the product development cycle time, even if there is no blueprint for the desired product. Recently, computer vision algorithms have enhanced the RE process and strengthen its capabilities to reconstruct challenging shapes. Nevertheless, the large body of the reported literature is restricted to estimate the 3D shape of the scanned part from a single/multiple 2D/3D image based on predefined classes using supervised learning. The ability to reconstruct intricate geometrical features of real mechanical parts and complex shapes has not been fully realized yet. In this context, this paper reports on a hybrid learning technique-based conceptual computer vision framework to enhance RE process for reproducing of low demand products. The hybrid learning proposed herein is a supervised and unsupervised learning technique using a dual deep learning models to enrich the computer vision technique with the ability to reconstruct 3D complex features using a single 3D depth image.

Keywords: Computer Vision, Machine Learning, Deep Learning, Hybrid learning, 3D Reconstruction, Production, Additive Manufacturing (AM), Reverse Engineering (RE).

1 Introduction

Additive manufacturing (AM, also known as 3D printing) plays an essential role in modern manufacturing because it enables printing components in a Make-To-Order approach, for low-demand products and replaceable spare parts [1]. Although, regular mass-production systems offer cost-effective solution for high demand products, this is not the case for producing low-demand products and spare parts [1-2]. 3D Printing of spare parts could be an achievable task if the blueprint of the part to be manufactured is already exists, but it will be very costly and time-consuming to reproduce the blueprint if it is no longer available [2]. Reverse Engineering (RE) helps avoid redrawing/remodeling of existing parts, with no blueprint available, to facilitate their 3D printing. RE is described by acquiring and recognizing the features of the existing part to reconstruct its shape in a computer added design (CAD) format, e.g. standard triangle language (STL) format, in order to be produced [2].

In RE, the reconstruction process of the 3D model is carried out via characterizing the original product, i.e. old spare part, using a camera or a 3D scanner. Current RE methods are developed based on 3D scanning and photogrammetry [3]. The photogrammetry workflow starts by capturing various images of an existing part. Then these images are fed into the photogrammetry algorithm to estimate the 3D shape and features of the product and to remove unwanted features such as background and overlap with another part. Finally, the 3D shape is to be converted to the STL/CAD format and loaded onto the 3D printer. Figure 1 shows the workflow of the RE process.

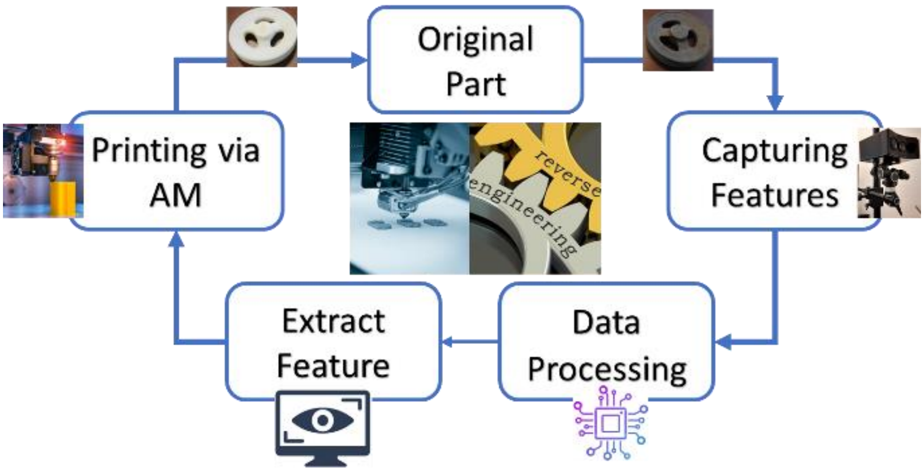

Fig. 1. Workflow of reverse engineering of additively manufactured parts (modified from [3]).

Most of recent photogrammetry methods are computer vision algorithms and developed to convert a large number of 2-dimensional red-green-blue (2D RGB) images into a 3-dimensional image including the depth feature and is called “RGB-D image”, “depth image”, or “3D image”. This depth image is used to generate a 3D model. The disadvantages of using the photogrammetry method are as follows; capturing a large number of 2D images is time-consuming and needs a high processing power afterwards to reconstruct the 3D image/STL file of the part. In addition, the method omits the ability to estimate missing information such as uncaptured features or damaged/broken area in the scanned part. The aforementioned limitations preclude the full potential of RE to produce accurate replaceable spare parts and low demand products.

Machine learning (ML) and deep learning (DL) can be utilized to impower computer vision algorithms to enable the reconstruction of the 3D part features using a limited number of 2D/3D images. ML and DL are defined as a group of computer algorithms that can autonomously expand their classification/estimation/decision making capabilities utilizing a number of learning techniques [5-7]. In ML/DL, the algorithms build a learning model using a sample of data, termed "training data" or "ground truth data", which is the previous information of what the output of the model should be for a given input. The outcome of the learning process enables the algorithm to make predictions or take a specific action without being explicitly pre-programmed to perform [5-7]. ML/DL algorithms are used in a wide variety of applications such as search engine recommendations, natural language processing, speech recognition, smart robotics/environment interaction, and computer vision.

In ML/DL, the learning techniques could be supervised, unsupervised, semi-supervised or reinforcement. The main difference among these learning techniques is the level of availability of ground truth data. In particular, in supervised learning, a labeled dataset is utilized to train a function and enrich its ability to map inputs to outputs of the processed dataset. Face recognition is considered a supervised learning problem, in which the algorithm learns how to autonomously distinguish and correlate human face images with associated "labels", people names in this case. Dissimilar to the previous learning technique, in unsupervised learning, there is no labeled outputs and thus its goal is to perform clustering of similar features out of a group of data. A search engine recommendation exemplifies un supervised learning task, where the algorithm clusters relevant words/sentence based on understanding their features. The semi-supervised learning bridges both former techniques by utilizing a blend of labeled and unlabeled datasets. This enables the algorithm to extract information from the labeled data to take proper decisions such as classification of the unlabeled dataset [5-7]. Finally, the reinforcement learning which is a technique of rewarding true activities and punishing false activities. This technique assigns true values "1" to the required actions and false values "0" to unwanted actions in order to train the algorithm "agent" how to take right decisions. Smart robotics/environment interactive systems and autonomous cars are good examples for the implementation of the reinforcement learning technique. Figure 2 shows the different between the four learning techniques.

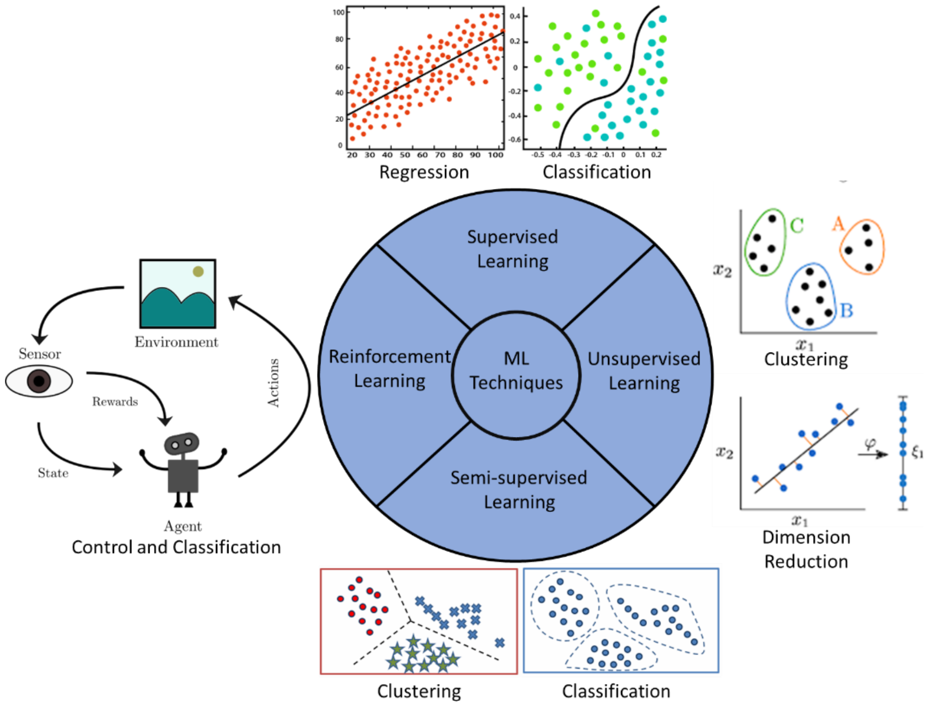

Fig. 2. The different between the four machine learning algorithms

In the literature, the utilization of ML/DL in computer vision for 3D reconstruction was restricted to supervised learning. In this case, the algorithm recognizes the part and correlate it to one class of the predefined dataset, which enables to retrieve the full part features from the saved features database and complete the missing features of the part. However, and in case the part doesn't exist in the database, the algorithm will not be able to estimate the missing features [7-10]. Further elaboration of the 3D reconstruction using supervised learning will be given in Section 2.

In this paper, the authors introduce a new conceptual computer vision framework based on hybrid leaning approach using supervised and unsupervised learning techniques. The proposed leaning approach is designed based on dual deep learning algorithms to enhance the 3D reconstruction method. Following this introduction, the remainder of this paper is organized as follows; in Section 2, the state-of-the-art is reviewed. In Section 3, the proposed approach entailing computer vision reconstruction method is explained. In Section 4, the hardware specs and the overall system architecture are presented. In Section 5, the implications and expected outcome are discussed. In Section 6, the conclusion is drawn and future work is suggested.

2 State-of-the-Art of 3D shape reconstruction

As formerly stated, recent computer vision algorithms [7-8], have shown high potential to reconstruct a 3D object from a small number of 2D/3D image, only if the object is part of the saved database. It is worth emphasizing that this methodology of 3D reconstruction inspired as an extension of the face recovery algorithms [9], but again with limited competence to estimate missing features and full features of a complex shape.

Recently, there are a number of studies [10-12] attempted to reconstruct complex shapes by acquiring multiple images based on supervised approach. Nevertheless, the limitation in these studies was the low resolution of the reconstructed part. In [13-14], authors introduced a high-resolution 3D shape via Octree representation. Octree is a tree-base data structure which enables structure nodes to formalize 3D graphical data. Also, in order to increase the quality of the produced 3D image, the authors in [15] designed a pseudo-renderer to predict depth 3D shapes. The pseudo-renderer performs decisions to filled the gaps between similar pixel segments and set gaps between different pixel segments to reproduce the full 3D features.

In order to avoid error and enhance quality of the estimated part, the missing features of the 3D shape can be completed by applying plane fitting algorithm on the captured 3D images/points cloud. The algorithm can complete the missing area in the 3D shape as shown in Figure 3. Plane fitting is an algorithm to extract the plane feature from points cloud. It can be implemented via a number of methods such as least squares fitting (LSF) and principal component analysis (PCA). In LSF, the algorithm performs iterative steps to find the best fitted plane with the least-squares constraint of the distances from the scanned points to the plane. In the PCA, the algorithm calculates the eigenvector of point cloud as a perpendicular vector for finding out the parameters of the best plane.

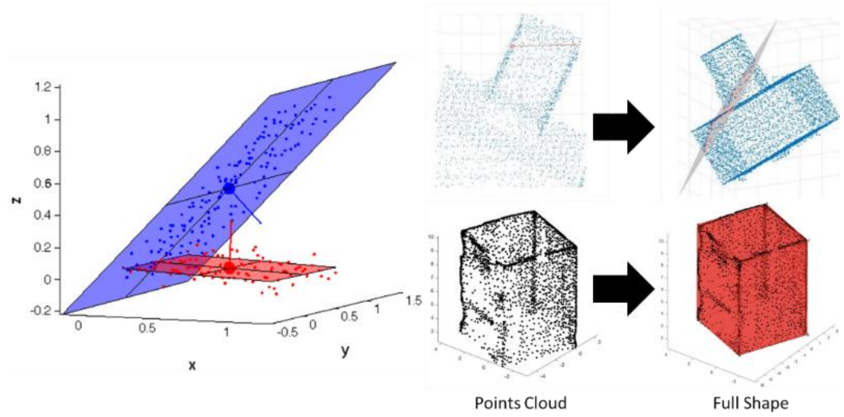

Fig. 3. Convert points cloud to full shape using plane fitting algorithm (modified from [18-19])

In [20-21], the authors utilized a shape symmetry technique to estimate the unknown features in the image using template feature algorithm based on supervised learning. Again, the captured image is compared with some of template which contains some different classes. If the captured image matches one of templates, the algorithm collects the full features from the database and reconstructs the missing features of the part. For example, morphable 3D models are exploited for face reconstruction, as shown in Figure 4(a). This face reconstruction method was edited to reconstruct simple objects in [21], as shown in Figure 4(b).

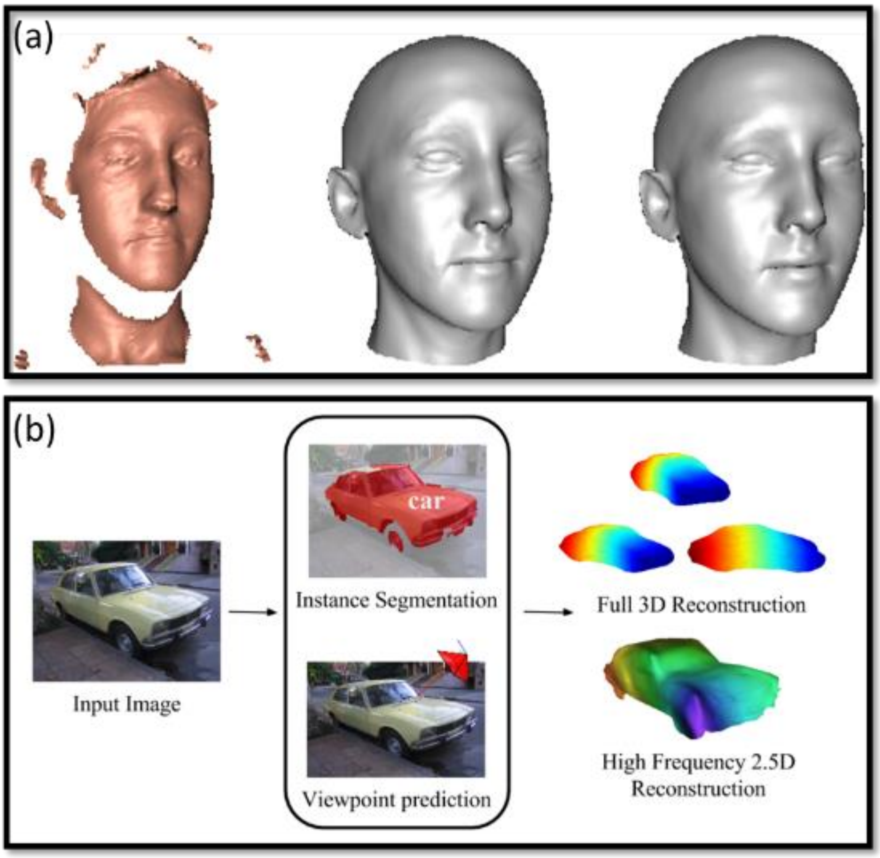

Fig. 3. The estimation of missing information, (a) 3D face reconstruction while, (b) 3D object reconstruction (modified from [20-21])

With the rapid improvement of depth cameras, the RGB-D images are utilized to reconstruct the 3D object shape. Recent studies [24] estimated the 3D shape via a deep learning algorithm from several 3D images such as ShapeNets DL model. The ShapeNets [25] introduces a 3D reconstruction of shape from a single 3D image. The process is performed as follows; the ShapeNet model recognizes the 3D shape from a single depth image based on 10 object classes classifier (see Figure 5(a)) and then completes the missing features based on the saved featured in the database, as shown in Figure 5 (b-c).

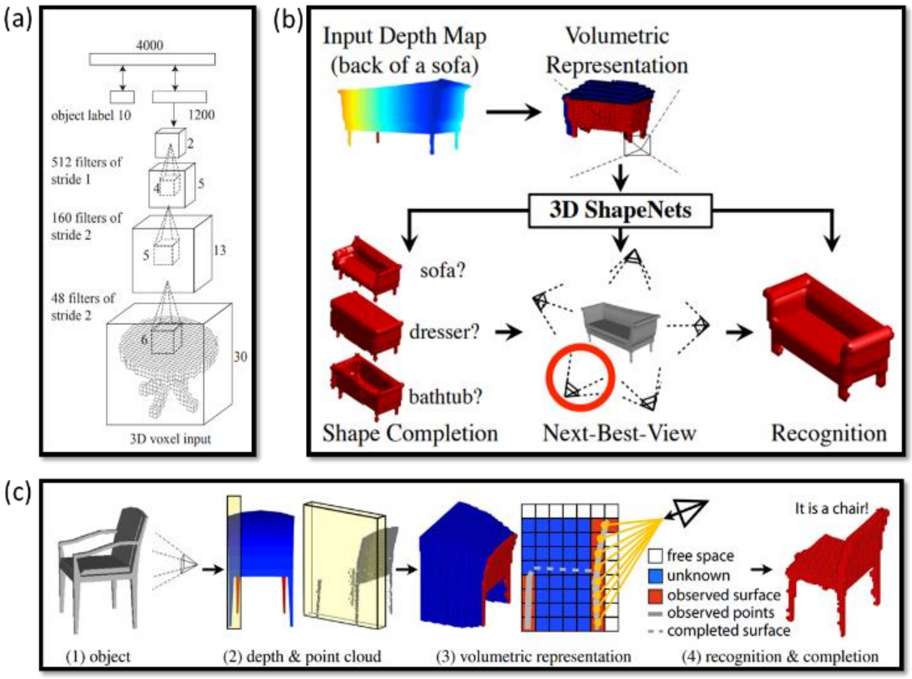

Fig. 5. 3D shape reconstruction using 3D ShapeNets, (a) shows the architecture of the classifier which is used in recognition stage, (b) represents the algorithm architecture, (b) shows the workflow of the algorithm from the step of the object capturing till the completion of the missing features and generates the full 3D image (modified from [25]).

Looking at the reviewed literature, to the best of the authors knowledge all of the existing algorithms are found limited to classifying the objects on the basis of objects stored in a database by means of supervised learning. In case of capturing part that is not stored in the classes of the database, the algorithm will fail to estimate part missing features. This is even more problematic given that most of the available datasets are for toys, human bodies and there is no comprehensive dataset for mechanical and engineering part. Also, there is no a clear path for how existing method can be utilized to reconstruct complex shapes.

3 Proposed Algorithm

In this paper, the authors introduce a new computer vision framework based on a hybrid, supervised and unsupervised, learning techniques to estimate full 3D shape structure from a single image. The proposed design will be built using the C++ programming language, the open computer vision library (OpenCV) and the PyTorch machine learning framework [26-28]. The algorithm starts by capturing a 3D image for the existing spare part from a single view. Then the algorithm preprocesses the captured 3D image for edge detection and background removing. Then, the algorithm will try to classify the shape in a supervised way. If this fails because the object/shape of the image belongs to an unknown class, the algorithm will estimate the shape in a fully unsupervised way using a defined group of geometric shapes (see Section 3.2). The proposed approach workflow is shown in Figure 6. In the following sub-sections, the three phases of design and development of the proposed algorithm (collecting data and preprocessing - model design and training - platform selection and model deployment) are explained in detail.

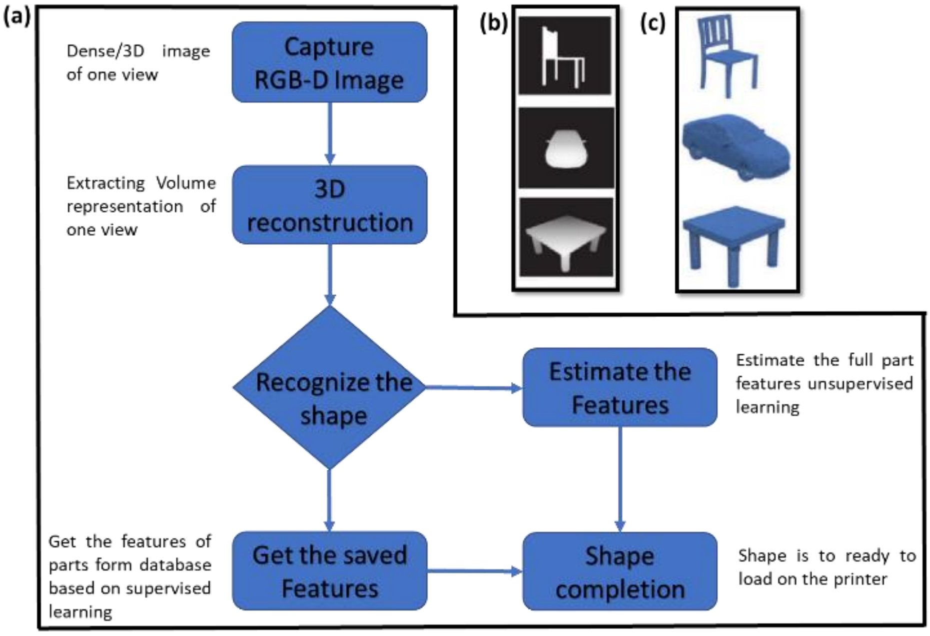

Fig. 6. The workflow of the proposed approach

3.1 Data Collecting and Preprocessing

In this phase the data will be collected and preprocessed. The data will be collected from four dataset sources. The first source is the "Object Scans" dataset [29] which has more than ten thousand RGB-D images for the real objects such as mugs, toys, construction cars. Also, it has more than four hundred reconstructed CAD model which allows for training of 3D classifier model and 3D reconstruction model. The second dataset is a large-scale annotated mechanical component benchmark which is called "MBC dataset" and has 58,696 mechanical components in 3D shape in 68 classes [30]. The third data source is a big CAD model dataset which is called "ABC dataset" and has a one million CAD models for geometric learning purpose [31]. Every model of the ABC dataset is a group of obviously parametrized arcs, curvatures and surfaces which offer the ground truth for variance amounts such as sematic segmentation, geometric feature recognition, and object feature reconstruction. Finally, the fourth source is in house dataset build in our 3D printing labs for printed parts.

All of the four dataset groups will be preprocessed to create RGB-D images from different orientations of the objects, front, back, top, bottom, right and left side along with its corresponding full feature 3D image. Then a data augmentation technique will be implemented on the dataset in order to increase the amount of data by adding modified copies of the already existing data or newly created data form the existing dataset [32]. This procedure increases the accuracy of the deep leaning model by avoiding overfitting of the model [32], which takes place when the model function is strictly aligned to a limited set of data points [32].

The purpose of compiling the four different dataset groups is to increase the variety and number of features and classes of the spare parts and real objects (see Section 3.2). This variety make the proposed DL model is able to deal with missing features in complex parts.

3.2 Model Design and Training

The proposed hybrid learning technique will be trained by using labeled and unlabeled data to be able to estimate the missing features for given input. The potential of using hybrid learning in 3D reconstruction can be summarized in the following points;

1. In supervised learning, the algorithm is trained using classes of training examples and can estimate parts precisely if they belong to these training classes.
2. In unsupervised learning, all examples are unlabeled, and the algorithm will estimate the part features without restriction to specific classes. However, estimation quality will be low in comparison with the supervised learning case.
3. In the hybrid learning, both of labeled and unlabeled dataset of parts is utilized to optimize the parameters of the model. Where the dual DL models get the advantage of supervised and unsupervised leaning and minimize their individual disadvantages.

The proposed design of the DL model is inspired from a Pixel2Mesh++ model and GAN-BERT model [33-34]. Pixel2Mesh++ is DL model based on supervised learning technique for 3D reconstruction and GAN-BERT is DL model based semi-supervised learning model utilized for text classification, as optical character recognition (ORC) problem. In the proposed DL model of supervised learning, the algorithm reconstructs the shapes which is listed in the database (has labels) but if the shape is recognized as unknown shape, the shape will be reconstructed using the unsupervised leaning model. The unsupervised model is trained via labeled and unlabeled dataset with additional noise and geometric data. The utilization of this dataset is as follows; a gaussian noise data will be generated to produce fake cases by captivating the input of a 100-dimensional gaussian feature. This noise data is added to the geometric shapes vectors to generate fake noisy geometric data. The generator of the DL model receives this data and reconstructs the 3D shape (see Figure 7).

The discriminator is built based on a multilayer perceptron with a last layer of SoftMax activation function. The discriminator will receive the fake noisy geometric data from the generator (G) and the real object data generated by graph convolutional networks (GCN). Then, the SoftMax's layer has output dimension K+1-dimension vector of logits, where k is the number of defined classes in the defined database and 1 is the unknow class. The discriminator classifies the input data into real, those belong to class of labeled dataset of the supervised learning, or nonreal object, which are not part of the database. In case of prediction the input as real instance, the algorithm predicts which classes this object belongs to and then reconstructs its features based on the saved features in the database. In case of nonreal object, the discriminator will provide the shape which is generated according to the training using the noise geometric data, as unsupervised leaning.

In particular, the mathematical formulation of the proposed model is as follows [34]; Where D and G as the discriminator and generator, and p_D and p_G as the real data distribution and the generated samples, correspondingly. The model will be trained to classify object in K classes and the generator will work to estimate the class K+1. The objective of D is defined as follow;
$p_m(y(hat) = y|x, y = k + 1)$ is the probability provided by the model m that a generic case x is linked with the fake class and p_m (ˆy = y|x, y ∈ (1, ..., k)) that x is considered real object, thus related to one of the target classes.

$$Loss_D = Loss_{D\ (supervised)} + Loss_{D\ (unsupervised)}$$
$$Loss_{D\ (supervised)} = \mathbb{E}s_{D\ x,y\sim pd} Log[p_m(y(hat) = y|x, y \in (1,..,K))]$$
$$Loss_{D\ (supervised)} = \mathbb{E}s_{D\ x,y\sim pd} Log\big[1 - p_m\big(y(hat) = y\big|x, y \in (k+1)\big)\big] - \mathbb{E}s_{D\ x,y\sim G} Log[p_m(y(hat) = y|x, y \in (k+1))]$$

$Loss_{D\ (supervised)}$ measures the error in estimate the false class to a real example among the defined k categories. $Loss_{D\ (unsupervised)}$ measures the error in incorrectly recognizing a real (unlabeled) example as fake and not recognizing a fake example. At the same time, G is expected to generate examples that are similar to the ones sampled from the real distribution pd. As suggested in [33], G has to generate data approximating the statistics of real data as much as possible. In other words, the average example generated in a batch by G should be similar to the real prototypical one. Formally, let's f(x) denote the activation on an intermediate layer of D. The feature matching loss of G is then defined as: $Loss_{G(Feature\ matchine)} = \| \mathbb{E}x{\sim}pd\ F(x) - \mathbb{E}x{\sim}G\ F(x) \|_2^2$.

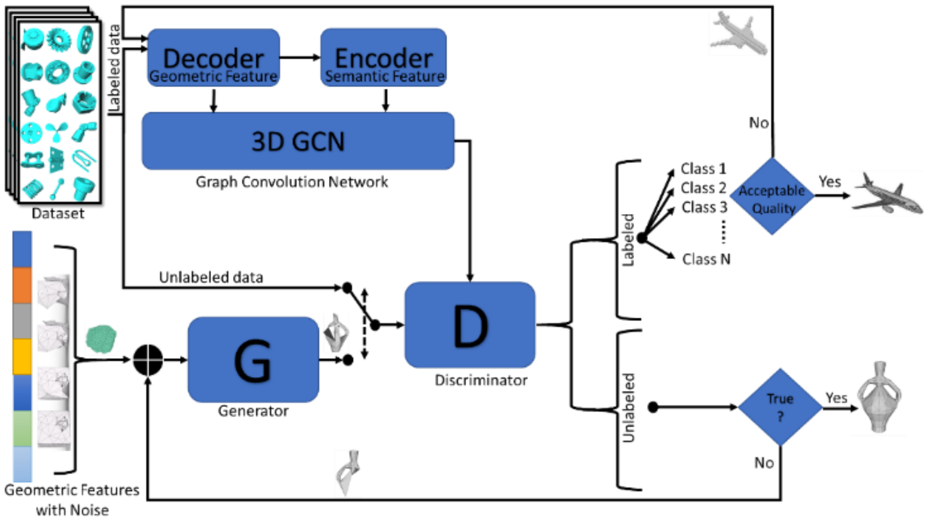

Fig. 7. The architecture of the proposed hybrid learning model.

In the training phase, the algorithm optimized three losses values, which are discriminator loss, generator loss, and the GCN. The discriminator loss is the summation of the losses of hybrid algorithms: first loss measures the error in assigning the wrong class to a real example among the original k categories, while the second loss measures the error in incorrectly recognizing of the unrecognized object. The generator loss is also the result of summation from 2 other losses (features matching loss and feature generation loss). Feature matching loss ensures that the generator produces 3D image which is similar to which input to the discriminator via the real images. The feature generation loss measures the error induced by fake geometric which identified by the discriminator. The GCN loss is the error of labeled wrong class in supervised learning.

3.3 Platform Selection and Model Deployment

For the proposed algorithm, a D455 Intel depth camera is selected for capturing the 3D image of the desired part [35]. The camera specs are as follows; its depth resolution is up to 1280 x 720 and its depth accuracy is less than 2 %. The camera frame rate per second is up to 90. The camera interfaces with a Jetson X2 embedded kit. This kit is loaded by the algorithm software [36]. The Jetson X2 is a small Embedded Linux controller with a graphical processing unit (GPU) to enable running software in a parallel computing way. The parallel processing features make the kit a good platform to train and deploy the deep neural network models. Also, it has a several software packages to enable installing machine/deep learning frameworks such as Scikit-learn, PyTorch, TensorFlow.

The Jetson X2 kit is utilized in training phase and in deployment phase as well because the model architecture is loaded on the kit and the DL/ML libraries as well. The dataset is loaded on the kit. After that, the training phase is started and it expected to stop after 1000 epoch and the validation phase will start to make sure there is no over-fitting. After that, the training will be stopped and the model is extracted in PyTorch format, i.e. pth extension, see Figure 8.

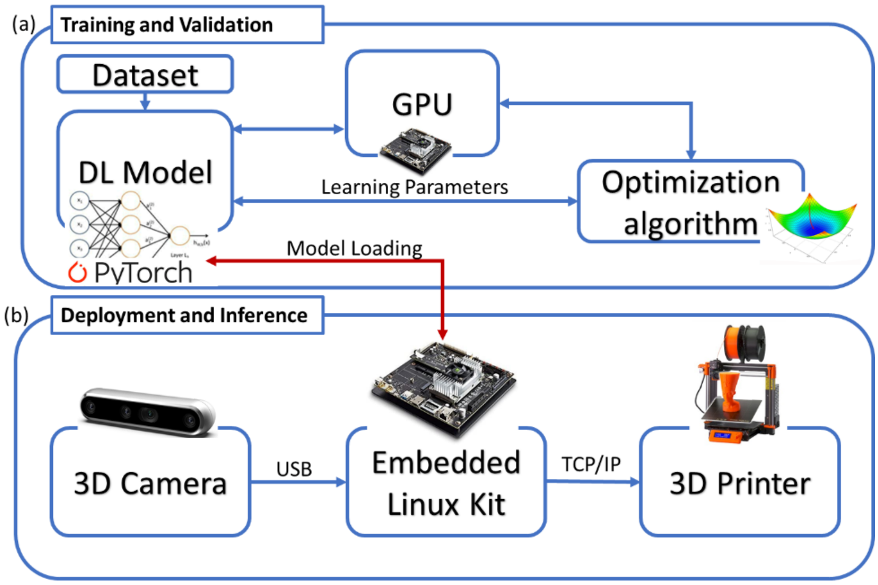

Fig. 8. The block diagram of the hardware setting in: (a) the training and validation phase, (b) deployment and inference phase.

4 Implications and Expected Outcome

This research presents a new methodology of utilizing computer vision approach along RE in AM process. As apparent from this research, RE processes are recently plagued by a number of limitations such as the need for a large number of images and huge processing power and long time. These limitations can be addressed by the proposed approach which can perform the 3D reconstruction stage of RE by using only one depth image and estimate the hole part features.

The utilization of using hybrid learning techniques in the proposed computer vision framework provides dual different DL algorithms which has the ability to deal with known and unknow objects and estimate the full 3D features from a single depth image for complex shape. The utilization of the proposed framework enables high potential for utilized computer vision for successful implementation of RE. In terms of manufacturing, through this approach, industries would be able to gain the benefits that computer vision offers in terms of enhanced RE implementation in short time with low processing power which has a high impact on the performance of AM process.

5 Conclusions

In this paper, the authors proposed a conceptual computer vision framework to extract the full 3D feature of the acquired object in the RE process. The proposed algorithm is able to learn deep representation of 3D shapes from a single RGB-D image to reconstruct its 3D shape in a short time with low processing power. This approach is designed based on hybrid learning techniques. The implementation of the proposed approach is intended to strengthen the manufacturing competence by enhancing the quality of AM printed part and enable performing successful RE in the spare parts which has missing features or damage area. The proposed approach covers a wide range of engineering parts with variety of features in training phase in order to be able to estimate missing features in complex shapes. For the future work, authors are plan to implement this framework and experimentally validate the results.

Acknowledgement

This work was carried out with the support of the Karlsruhe Nano Micro Facility (KNMFi, www.knmf.kit.edu) a Helmholtz Research Infrastructure at Karlsruhe Institute of Technology (KIT, www.kit.edu) and under the Helmholtz Research Programme MSE (Materials Systems Engineering) at KIT.

References

1. Li and et al.: The Influence of Additive Manufacturing on the Configuration of Make-to-Order Spare Parts Supply Chain under Heterogeneous Demand. Int. J. Prod., 57, 3622–3641, (2019)
2. Mecheter and el at.: Additive Manufacturing Technology for Spare Parts Application: A Systematic Review on Supply Chain Management. Appl. Sci. 12, 4160, (2022).
3. Katsuhiro and et al.: A Sustainable Reverse Engineering Process. Procedia CIRP, 98, 517-522. (2021).
4. Charles et al.: In-Process Digital Monitoring of Additive Manufacturing: Proposed Machine Learning Approach and Potential Implications on Sustainability, KES SDM, (2020).
5. Blanz V. and Vetter T.: Face Recognition based on Fitting a 3D Morphable Model, IEEE Trans. Pattern Anal. Mach. Intell 25(9), 1063–1074, (2003).
6. Kar and et al.: Category-specific object reconstruction from a single image, CVPR, (2015).
7. P. Dou, S. K. Shah, and I. A. Kakadiaris.: End-to-end 3D face reconstruction with deep neural networks, CVPR, (2017).
8. Gwak and et al. “Weakly supervised 3D Reconstruction with Adversarial Constraint,” 3DV (14), (2017).

9. Tulsiani and et al., “Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency,” CVPR, (2017).
10. Yan and et al.: Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision,” NIPS, (2016).
11. Murthy and et al., “Reconstructing Vehicles from a Single Image: Shape Priors for mRoad Scene Understanding,” ICRA, (2017).
12. A. Johnston and et al.: Scaling CNNs for High Resolution Volumetric Reconstruction from a Single Image, ICCV Workshops, (2017).
13. Lin, C.-H., Kong, C., Lucey, S.: Learning Efficient Point Cloud Generation for Dense 3D Object Reconstruction,” AAAI, (2018).
14. Wu and et al.: MarrNet: 3D Shape Reconstruction via 2.5D Sketches, NIPS, (2017).
15. Monszpart and et al.: RAPter: Rebuilding Man-made Scenes with Regular Arrangements of Planes,” ACM Trans. on Graphics 34(4) 1–12, (2015).
16. Mitra, N. J., Guibas, L. J., Pauly, M.: Partial and Approximate Symmetry Detection for 3D Geometry, SIGGRAPH, (2006).
17. Pauly and et al.: Discovering structural regularity in 3D geometry,” ACM Transactions on Graphics 27(3), 1, (2008).
18. Richard an et al.: Plane Segmentation from Point Clouds. Pollack Periodica. (13), 159-171, (2018).
19. Usama Mehmood, Cuboid fit (RANSAC) (https://github.com/usama-mehmood3/Cuboid-Fitting-RanSAC), GitHub. Retrieved May 10, 2022.
20. Amberg, B., Knothe, R., Vetter, T.: Expression invariant 3D face recognition with a Morphable Model, IEEE FG (8), 1-6, (2008).
21. Kar A., Tulsiani S., Carreira J., Malik J.: Category-specific object reconstruction from a single image, CVPR, (2015).
22. Hartley, R., Zisserman A.: Multiple View Geometry in Computer Vision. Cambridge University Press, (2004).
23. Newcombe, R. A., Lovegrove, S. J., Davision, A. J.: DTAM: Dense Tracking and Mapping in Real-time, ICCV, (2011).
24. Blanz. V., Vetter, T.: Face Recognition based on Fitting a 3DMorphable Model, IEEE Trans. Pattern Anal. Mach. Intell 25(9), 1063–1074, (2003).
25. Wu and et al.: 3D ShapeNets: A Deep Representation for Volumetric Shapes, CVPR, (2015).
26. C++ programming language, https://www.cplusplus.com/, last access 9 May 2022.
27. OpenCV, https://opencv.org/, last access 9 May 2022.
28. PyTorch, https://pytorch.org/, last access 9 May 2022.
29. Sungjoon and et al.: A Large Dataset of Object Scans, CVPR, (2016).
30. Kim and et al.: A Large-scale Annotated Mechanical Components Benchmark for Classification and Retrieval Tasks with Deep Neural Networks, ECCV, (2020).
31. Koch and et al.: ABC: A Big CAD Model Dataset for Geometric Deep Learning, CVPR, (2019).

32. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature (521), 436–444 (2015).
33. Nanyang and et al.: Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images, CVPR, (2018).
34. Danilo and et al.: GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples, ACL, (2020).
35. Intel 3D Camera, https://www.intelrealsense.com/depth-camera-d435/, last access 9 May 2022.
36. Jetson Embedded Kit, https://developer.nvidia.com/embedded/jetson-developer-kits, last access 9 May 2022.