Feature-Based Reverse Engineering in Circular Design

Nithin Gengamuthu (603115), Sabin Cristanious Samraj (603097), Sai Akash Kumar Somasundaram Vijaya Kumar (603096), Davis Linston Stephen Raj (603058)

Clausthal University of Technology

Abstract

Reverse engineering mechanical parts for CAD reconstruction often depends on manual feature identification, which is time-consuming and error-prone. To improve automation, this project explores two deep learning approaches: A hybrid 3D Convolutional Neural Network (CNN) that combines classification (to detect features like holes) and regression (to estimate parameters such as diameter and center), and a CNN adapted for regression that directly predicts dimensional parameters without feature classification. Parametric CAD models were generated using the CADQuery library, voxelized, and annotated with class labels and dimensions. Both models were developed in PyTorch. Preliminary results show that the hybrid CNN offers better interpretability by identifying both feature types and their dimensions [2,6], while the CNN adapted for regression simplifies the pipeline but may lack semantic clarity. This evaluation highlights the potential of hybrid models to improve automation and accuracy in CAD reconstruction workflows aligned with circular design goals.

1. Introduction

In a circular economy, reverse engineering supports sustainability by enabling the reuse and redesign of components through digital reconstruction. This is especially useful for legacy parts or unmodeled components, where editable CAD models are necessary for lifecycle extension and closed-loop manufacturing. A core challenge lies in converting 3D scanned data into accurate parametric CAD models. Traditional methods focus on geometry but lack semantic understanding—like recognizing holes or estimating their parameters—which is critical for CAD/CAM workflows [2,5].

Recent studies have explored deep learning as a solution. While 3D Convolutional Neural Networks (CNNs) are effective for classification of features, they struggle with estimating continuous parameters. Hybrid approaches that combine classification with regression offer a promising alternative for feature recognition and parameter extraction [4,6].

1.1 Research Question

How does a hybrid CNN model combining classification and regression tasks influence the cost function during hole feature recognition and parameter integration in CAD reconstruction, compared to a CNN model adapted solely for regression?

This study proposes such a hybrid CNN to detect holes and predict their diameter and position from voxelized 3D data. We evaluate its effectiveness against a CNN adapted for Regression, focusing on model accuracy, convergence behavior, and its potential to support automated reconstruction workflows in sustainable product design.

1.2 Areas of Relevance and Contribution diagram

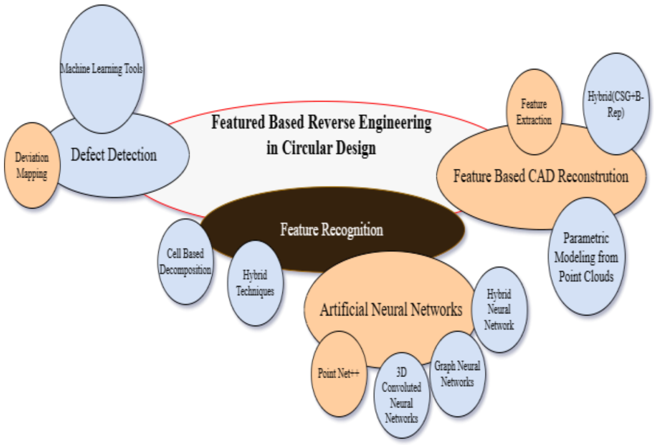

Figure 1: ARC diagram for Feature-Based Reverse Engineering in Circular Design.

At the center lies Feature Recognition, the core element that enables the translation of raw 3D scan data into structured, editable CAD features. The ARC (Areas of Relevance and Contribution) is organized into three primary domains:

Defect Detection:

Focuses on identifying imperfections in scanned data before reconstruction including key components such as:

Deviation Mapping: Compares scanned geometries against ideal CAD models to highlight surface discrepancies.

Machine Learning Tools: Automate the detection and classification of defects to improve inspection speed and accuracy.

Cell-Based Decomposition: Divides complex parts into smaller sections for localized defect analysis.

Hybrid Techniques: Combine rule-based and learning-based methods for more reliable defect detection.

Artificial Neural Networks (ANNs):

Neural networks serve as the computational backbone of intelligent feature recognition in reverse engineering.

PointNet / PointNet++: Directly process point cloud data while preserving geometric structure.

3D Convolutional Neural Networks (3D CNNs): Analyze voxelized data to learn 3D spatial patterns relevant to feature detection.

Graph Neural Networks (GNNs): Leverage mesh connectivity to understand shape topology and boundary conditions.

Hybrid Neural Networks: Combine CNNs with transformers or recurrent layers to improve generalization on complex data.

Feature-Based CAD Reconstruction:

This domain deals with converting detected features into fully editable CAD models:

Feature Extraction: Identifies geometric entities such as holes, slots, fillets, and chamfers.

Hybrid (CSG-to-B-Rep): Transforms Constructive Solid Geometry (CSG) into Boundary Representation (B-Rep) for CAD integration.

Parametric Modelling from Point Clouds: Automates the conversion of 3D scans into parametric models suitable for CAD tools.

The ARC diagram illustrates a data-driven, AI-augmented pipeline, where feature recognition acts as the central bridge between physical scans and CAD-ready digital twins—advancing automation, precision, and sustainability in reverse engineering workflows.

The Digital Twin technology plays a crucial role in this project by enabling a seamless connection between physical components and their digital counterparts. By reconstructing accurate, editable CAD models from 3D scans through feature-based recognition, the system creates a digital twin that reflects real-world geometry. This twin supports automated redesign, simulation, and quality control, making the reverse engineering process more efficient, data-driven, and aligned with circular design and smart manufacturing goals.

1.3 Question-Method Matrix

Table 1: QMM table

Methods / Questions	Method A	Method B	Method C
RQ1	Prototyping and Quantitative comparative study experiment with metrics	Comparative Study With Manual Methods	Reusability Simulation in Design Reconfigur-ation

The Question–Method Matrix (QMM) in Table 1 outlines the systematic manner in which the main objectives of the project are investigated. The research question is supported by a number of methodological strategies - experimental and analytical to ensure an exhaustive examination of the problem area.

Research Question 1, which represents the core technical focus of the project, compares the performance of a hybrid CNN model (with both classification and regression heads) and CNN adapted for regression on Hole feature recognition and parameter estimation.

Method A involves building prototypes of both hybrid and CNN adapted for regression models and comparing their accuracy, loss functions, using metrics such as MSE, accuracy, and training time.

Method B extends the comparison by including manual feature recognition as a baseline, evaluating time efficiency and error rates.

Method C tests the usability of CNN-extracted features by simulating their integration into CAD systems like Siemens NX, verifying whether the features fit design intent and constraints.

This matrix ensures that the research question is addressed from both theoretical and applied perspectives, supporting the development of a robust and generalizable system for reverse engineering using machine learning and CAD automation.

1.4 State of the Art in Research

Reverse engineering has advanced from manual feature modelling to AI-driven automation using 3D data formats such as meshes, point clouds, and voxel grids. Recent developments focus on deep learning—particularly 3D Convolutional Neural Networks (3D CNNs)—for extracting and processing volumetric medical data such as CT, MRI, and PET scans.

Some of the Advanced 3D CNN Architectures are:

- 3D U-Net: Extends the 2D U-Net into 3D, excelling in medical segmentation.
- VoxResNet: Incorporates residual connections in 3D volumes, improving depth without gradient vanishing.
- 3D DenseNet: Encourages feature reuse through dense connections, reducing parameter count.
- Hybrid Models: Combine 2D + 3D CNNs or integrate RNNs and attention modules to capture context and temporal dynamics.
- Transformer-3D CNN Hybrids (emerging): Combine CNNs for local features with transformers for global context modelling.

Our research draws on these foundations, adapting a hybrid 3D CNN that performs both classification and regression. This enables the model not only to identify feature types but also predict parametric values such as diameter and position. Compared to traditional pipelines, which separate detection from dimensional inference, this dual-task model integrates both tasks for improved efficiency.

2. Material/Data and Methods

1. Data Generation Using CADQuery – For Hybrid CNN and CNN Adapted for Regression:

Parametric 3D models of sheet metal components were generated using the CADQuery library. Key geometric features - specifically circular holes—were programmatically varied in terms of size and location. Each model was exported in .STEP format for further processing. Ground truth parameters such as hole diameter and center position (X, Y, Z) were saved in .csv format for annotation and training.

For the experiment, a dataset of STEP files was created, where each file represents a sheet metal part with and without holes.

Hole generation followed specific constraints:

- **Sheet metal dimensions:** 100 mm × 100 mm × 10 mm
- **Hole count:** 0 and 1
- **Hole positions:** At least 10 mm away from edges and center
- **Hole diameters:** From 1 mm to 25 mm

The hole parameters were stored in structured lists to serve as labels for classification and regression tasks across training, validation, and testing phases. This same dataset was used for both the hybrid CNN model and the CNN adapted for regression.

All experiments were implemented using the PyTorch deep learning framework and accelerated using Graphics Processing Units (GPUs). The configuration of experimental conditions is detailed in Table 2 and Table 3. Also, See Appendix for system details.

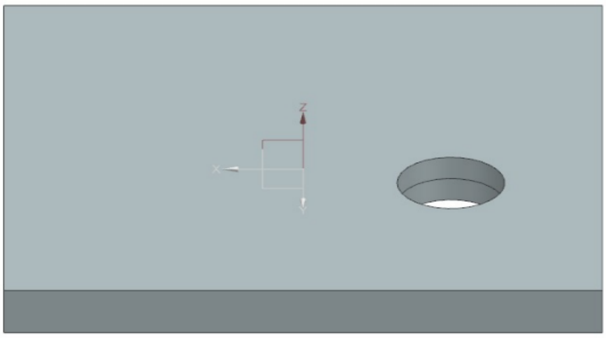

Figure 2: Generated 3D CAD Model using CADQuery

2. Preprocessing

In the preprocessing stage, the exported .STEP models were imported into Open3D, where they were first converted into point clouds. Each point cloud was then normalized—centered and uniformly scaled to fit within a unit cube—ensuring consistent spatial representation regardless of the original model size.

After normalization, the point clouds were voxelized into binary 3D grids with a fixed resolution of 200×200×20, producing a standardized format compatible with 3D Convolutional Neural Networks (CNNs). These voxel grids serve as structured volumetric data, where each voxel (volume pixel) represents the presence (0) or absence (1) of hole in space.

Since 3D CNNs cannot directly process the boundary representation (B-Rep) formats used in STEP files, voxelization becomes a critical step. It converts complex geometries into uniform, fixed-size arrays—analogous to 3D images—that enable CNNs to learn spatial features such as holes and perform regression or classification tasks effectively.

Each voxelized model was stored as a .npy file, a binary file format from the NumPy library. These files store the 3D voxel matrix as a multi-dimensional array, encoding spatial occupancy information using 1s (material) and 0s (empty space). This .npy format was used as the direct input to both the hybrid CNN and the regression-only CNN models.

3. Annotation and labelling

They were performed to assign each voxel grid a classification label based on feature type: for instance, 0 for holes, continuous feature parameters such as diameter, positional coordinates (x, y, z), and orientation were associated with each sample as regression targets. All annotation data were maintained in a master labels.csv file, ensuring traceability between voxel grids and their respective ground-truth values.

4. Dataset Structuring

The dataset was divided into Training (80%), Validation (20%) subsets, with balanced representation of feature types.

5. CNN Model Architecture

Hybrid CNN Model for Classification and Regression -

A Hybrid 3D CNN architecture will be trained with a Voxelized Dataset (.npy files) extracted from a generated primary STEP files dataset to learn spatial patterns that can distinguish hole-containing parts from solid ones. The model will output class probabilities indicating the presence or absence of a hole. This classification stage forms the first component of a hybrid model. If a hole is detected, a regression head is then used to estimate its geometric parameters, such as diameter and center position using the feature vectors passed from classification head.

This two-stage hybrid pipeline combines the strengths of CNNs in spatial feature recognition with regression-based parameter estimation, supporting accurate and interpretable feature extraction for CAD reconstruction workflows.

CNN Model adapted for Regression -

Convolutional Neural Networks (CNNs) are widely used for classification tasks, where they learn to assign input units to predefined categories.

In regression tasks, CNNs predict continuous values—such as dimensions or coordinates—for example, from input images. While CNNs for classification are well-established, CNN-based regression remains less explored, especially in 3D domains [3].

The adapted CNN for regression predicts continuous values instead of class labels, i.e., Hole (0) and No Hole (1). Convolutional layers are used to extract spatial features, followed by fully connected layers through which real numbers—such as diameters or positions—are output. Unlike in classification, Sigmoid Activation Function [3] is used in the final layer, along with loss functions like Mean Squared Error (MSE), enabling the network to learn precise numerical relationships and make suitable predictions for hole diameters and positions.

Although MSE is used as the primary training loss due to its differentiable nature and stronger penalty on large errors. Mean Absolute Error (MAE) will also be used as an evaluation metric to provide a more interpretable measure of prediction accuracy.

Data loading involves reading voxel .npy files and corresponding hole parameters from .csv. Each part normalizes the parameters, and returns voxel data.

6. Training Procedure

During training, the hybrid CNN model was optimized using a dual-loss strategy to handle both classification and regression tasks simultaneously. Cross-Entropy Loss was applied to the classification head to learn discrete feature types, while Mean Squared Error (MSE) or Mean Absolute Error (MAE) was used for the regression head to predict continuous geometric parameters.

The CNN model adapted for regression was trained using Mean Squared Error (MSE) and Mean Absolute Error (MAE) as there is no classification head in this adapted architecture [3].

Table 2: Hyperparametric Configuration

HYPERPARAMETER	VALUE
Optimizer	Adam
Data sets	1000
Epochs	200
Learning Rate	0.001
Batch size	8

7. Performance Evaluation

The evaluation of performance of the hybrid CNN model and CNN adapted for regression in this project focuses on accuracy, loss, training to classify geometric features - specifically holes - and to predict their parametric details, such as diameter and center position. Given that hole features are essential in reverse engineering for part mating and reusability, accurate detection and dimensional reconstruction are critical.

For the classification task, the model distinguishes between hole and no-hole feature.

The primary metric used is classification accuracy, defined as:

Accuracy = Correctly Predicted Feature Types/Total Predictions (1)

The loss during classification task in Hybrid CNN architecture is measured by using Cross entropy loss function:

$$LCE = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i) \quad (2)$$

Where:

- LCE is the cross-entropy loss
- yi is the ground truth label
- y^i is the predicted probability for class iii (output of the Softmax layer)
- C is the number of classes

High accuracy indicates the model's reliability in identifying whether a given voxel grid corresponds to a hole or another feature, which directly supports automated CAD reconstruction.

For the regression task, the model estimates hole-specific parameters such as diameter and center coordinates (x, y, z).

These values are crucial for precise CAD reconstruction and tolerance-based fit analysis. The quality of these predictions is measured using:

Mean Absolute Error (MAE):

$$MAE = (1/n) \times \Sigma\, |y_i - \hat{y}_i| \quad (3)$$

Mean Squared Error (MSE):

$$MSE = (1/n) \times \Sigma\, (y_i - \hat{y}_i)^2 \quad (4)$$

Where:

- n = total number of data points
- yi = actual value (ground truth) for the i^th sample
- ŷi = predicted value for the i^th sample

These metrics evaluate how close the predicted diameter and position of each hole are to the ground truth values obtained from the CAD-generated dataset.

Training Analysis:

Training vs validation loss curves

Epoch-wise accuracy tracking

8. Tools Used

During the dataset generation stage, CADQuery was used to programmatically create parametric CAD models of sheet metal components containing holes of varying positions and diameters. These models were exported as STEP files.

In the preprocessing phase, Open3D was employed to convert these STEP files into point clouds. The point clouds were then normalized to fit within a unit cube and voxelized into fixed-resolution binary 3D grids (200×200×20), enabling compatibility with convolutional neural networks.

Each voxelized model was saved as a .npy file using NumPy, which also handled data labeling and structuring into classification and regression targets.

For scripting and coordination across these stages, Python served as the primary language. The CNN models—both the hybrid and adapted for regression architectures—were developed task learning.

3. Results and Findings

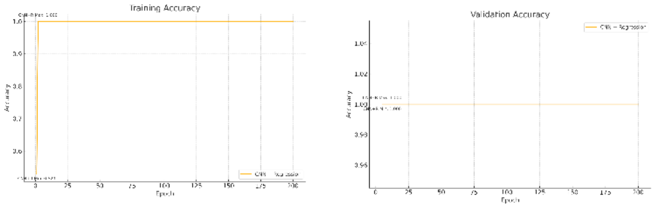

Figure 3: Accuracy during Training and Validation for Classification

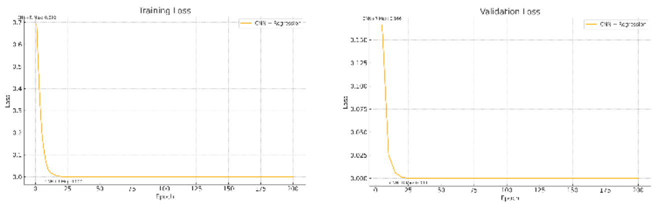

Figure 4: Training and Validation Loss for Hybrid CNN

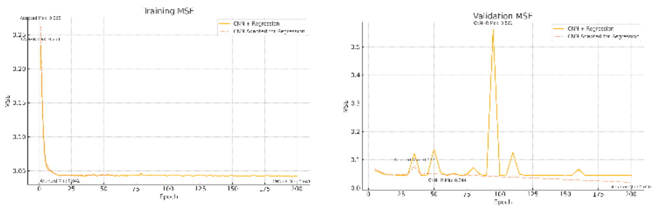

Figure 5: MSE Training and Validation Loss during Regression

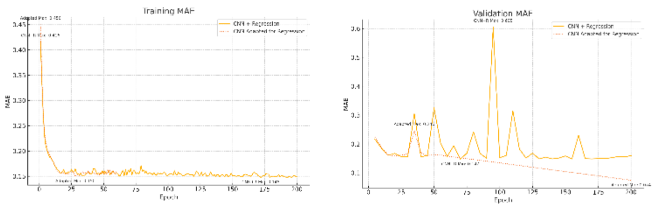

Figure 6: MAE Training and Validation Loss during Regression

Table 3: Cost Function & Training time comparison

Evaluation Metric	Hybrid CNN (Classification + Regression)	CNN Adapted for Regression
MAE (Regression)	0.148	0.150
MSE (Regression)	0.041	0.042
Training Time	120 seconds	190 seconds

4. Discussion and Conclusions

The hybrid CNN model combining classification and regression task consistently outperformed the CNN model adapted for regression in terms of lower training and validation MAE and MSE. Its dual-head structure, combining classification and regression, allowed for richer feature learning and better generalization. It also showed more stable convergence over 200 epochs, making it a more robust choice despite its added complexity. The adapted model, while simpler, showed earlier saturation and limited adaptability across unseen validation points.

Limitations

The adapted model lacked full validation coverage, requiring interpolation. Its structure does not support classification-based accuracy metrics. The hybrid model, while more accurate, incurs higher computational cost due to its larger architecture and dual-head processing.

Additionally, both models were trained without data augmentation and may be affected by voxel resolution limits or loss imbalance between classification and regression objectives.

Future Scope

Future development can focus on extending the current model to handle multi-hole detection, where multiple holes of varying sizes and locations are present in a single component. Additionally, the model can be improved to classify different hole types, such as through-holes, blind holes, and counterbores, using a multi-head architecture. Another important direction is the detection of threaded holes, which requires higher resolution or hybrid representations to capture fine geometric features. A simple binary classifier can be added to distinguish threaded from unthreaded holes. These enhancements would make the system more robust and suitable for real-world applications in automated CAD/CAM workflows and design validation.

Real-World Applicability

Example: Reverse Engineering of Brake Components

Hole features—such as their shape, size, and position—are critical in engineering applications. They directly affect assembly alignment, structural integrity, and functional performance. Accurate recognition of these features provides valuable insight into the original design intent, as well as the manufacturing and assembly processes involved.

A practical example of this is the reverse engineering of brake system components, particularly mounting brackets and plates. These parts often include precision-drilled holes that serve as load-bearing connections and alignment interfaces. Due to their tight tolerances and functional importance, manual interpretation can be time-consuming and error-prone.

By applying the developed hybrid CNN model, the recognition and parameterization of such hole features can be automated from 3D scan data.

This allows engineers to quickly reconstruct editable, parametric CAD models that not only replicate geometry but also infer manufacturing processes and assembly logic—essential for reuse, redesign, or quality control in circular design workflows.

5. Appendix:

Experimental Environment Configuration

To ensure reproducibility and transparency, the table below outlines the hardware and software configuration used throughout this project. All preprocessing, model training, and evaluation were carried out using the specified environment. These details are especially relevant when assessing training time, GPU utilization, and memory limitations, which may influence model scalability and performance.

Table 4: Experimental setup configuration

ENVIRONMENTAL PARAMETER	VALUE
Operating System	Windows 10
Deep Learning (CNN) Framework	Pytorch
Programming Language	Python 3.12
CPU	Intel(R) Core (TM) i7-10870H CPU @ 2.20GHz
GPU	NVIDIA GeForce GTX 1650Ti
RAM	16 GB

References

[1] Shaheen, Fatma; Verma, Brijesh; Asafuddoula, Md. **Impact of Automatic Feature Extraction in Deep Learning Architecture.** In: *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2016, pp. 1–6.
DOI: 10.1109/DICTA.2016.7797025

[2] Lee, Hyunoh; Lee, Jinwon; Kim, Hyungki; Mun, Duhwan. **Dataset and Method for Deep Learning-Based Reconstruction of 3D CAD Models Containing Machining Features for Mechanical Parts.** *Journal of Computational Design and Engineering*, Volume 9, Issue 1, February 2022, pp. 114–127.
DOI: 10.1093/jcde/qwab089

[3] Cira, C.-I.; Díaz-Álvarez, A.; Serradilla, F.; Manso-Callejo, M.-Á. **Convolutional Neural Networks Adapted for Regression Tasks: Predicting the Orientation of Straight Arrows on Marked Road Pavement Using Deep Learning and Rectified Orthophotography.** *Electronics*, 2023, 12(3980), pp. 1–15.
DOI

[4] Zhang, D.; He, F.; Tu, Z.; Zou, L.; Chen, Y. **FeatureNet: Machining Feature Recognition Based on 3D Convolutional Neural Network.** *Computer-Aided Design*, Vol. 101, pp. 12–22, Apr 2018.
DOI: 10.1016/j.cad.2018.03.006

[5] Balu, A.; Ghadai, S.; Krishnamurthy, A.; Sarkar, S. **Learning and Visualizing Localized Geometric Features Using 3D-CNN: An Application to Manufacturability Analysis of Drilled Holes.** *Computer-Aided Geometric Design*, Vol. 62, pp. 263–275, Jun 2018.
DOI: 10.1016/j.cagd.2018.03.006

[6] Lee, J.; Lee, H.; Mun, D. **3D Convolutional Neural Network for Machining Feature Recognition with Gradient-Based Visual Explanations from 3D CAD Models.** *Scientific Reports*, 12, Art. 14864, 2022.
DOI: 10.1038/s41598-022-19212-6

[7] Muraleedharan, L. P.; Muthuganapathy, R. **A Learning-Based Approach to Feature Recognition of Engineering Shapes.** *arXiv preprint*, 2021.
DOI: 10.48550/arXiv.2112.07962