Analyze fillets, chamfers and holes on 3D geometrical STEP file

MSc Research Project
MSc in AI for Business

Zunaira Sohail
Student ID: x23391154

School of Computing
National College of Ireland

Supervisor: Agatha Mattos

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Zunaira Sohail…………………………………………………………		
Student ID:	x23391154…………………………………………………………………		
Programme:	MSc in AI for Business………………………	**Year:**	2025……
Module:	Practicum 2………………………………………………………………………………………		
Supervisor:	Agatha Mattos…………………………………………………………………………		
Submission Due Date:	11-12-25…………………………………………………………………………………		
Project Title:	Analyze fillets, chamfers and holes on 3D geometrical STEP file		
Word Count:	4225…………………………………… **Page Count**…22………………………………		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Zunaira……………………………………………………………………….
Date:	08-12-25………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Analyze fillets, chamfers and holes on 3D geometrical STEP file

Zunaira Sohail

x23391154

Abstract

This study presents a systematic method for analyzing fillets, chamfers, and holes on 3D geometric models in the STEP file format. CAD models are exchanged between different software platforms using STEP, a widely used neutral data format. However, due to the complexity of the topological and parametric structures embedded in STEP data, the extraction and interpretation of detailed geometric features such as fillets, chamfers, and holes remains challenging. This study proposes an automatic feature recognition method that combines contour representation analysis (B-Rep), topological adjacency mapping, and curvature classification. The developed algorithm detects edge-to-face relationships and distinguishes between fillets with constant radius, corner chamfers, and cylindrical or conical holes. Experimental validation is performed using various industrial STEP models to evaluate the detection accuracy and computational efficiency. The results demonstrate high reliability in shape classification and feature extraction, enabling applications in manufacturing process planning, design verification, and reverse engineering. This research advances the interaction and automation of CAD/CAM processes through intelligent geometry analysis.

Keywords: 3D STEP Files, Geometrical models, ISO 10303 express language, 3D CAD modelling, Analyzing feature through machine learning

1. Introduction:

The exchange and analysis of 3D geometric data is essential in modern CAD/CAM workflows. The STEP format (ISO 10303) has become the industry standard for neutral CAD data exchange, enabling interoperability across heterogeneous software systems. (Lingming Zhang, Yue Zhao, Deyu Meng, & Zhiming Cui, 2021)Unlike vendor-specific formats, STEP provides an objective representation of product geometry and metadata, supporting communication throughout the product lifecycle.

Header Section:

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (( 'STEP AP203' ),
     '1' );
FILE_NAME ('mac-003-006_Halter.STEP',
     '2012-11-28T08:14:58',
     ( 'admin' ),
     ( 'Hewlett-Packard Company' ),
     'SwSTEP 2.0',
     'SolidWorks 2010',
     '' );
FILE_SCHEMA (( 'CONFIG_CONTROL_DESIGN' ));
ENDSEC;
```

Data Section:

```
DATA;
#1 = CARTESIAN_POINT ( 'NONE',  ( 26.50000000000000400, -27.75000000000000400, 55.00000000000000000 ) ) ;
#2 = PRODUCT ( 'mac-003-006_Halter', 'mac-003-006_Halter', '', ( #1148 ) ) ;
#3 = PERSON ( 'UNSPECIFIED', 'UNSPECIFIED', 'UNSPECIFIED', ('UNSPECIFIED'), ('UNSPECIFIED'), ('UNSPECIFIED') ) ;
#4 = CIRCLE ( 'NONE', #1399, 17.50000000000000400 ) ;
#5 = EDGE_LOOP ( 'NONE', ( #1820, #1815 ) ) ;
#6 = LINE ( 'NONE', #1235, #259 ) ;
#7 = CIRCLE ( 'NONE', #1403, 2.749999999999992000 ) ;
#8 = LINE ( 'NONE', #1221, #10 ) ;
#9 = LINE ( 'NONE', #349, #13 ) ;
#10 = VECTOR ( 'NONE', #1227, 1000.000000000000000 ) ;
#11 = VECTOR ( 'NONE', #801, 1000.000000000000000 ) ;
#12 = VECTOR ( 'NONE', #350, 999.9999999999998900 ) ;
#13 = VECTOR ( 'NONE', #352, 1000.000000000000000 ) ;
#14 = LINE ( 'NONE', #268, #78 ) ;
#15 = LINE ( 'NONE', #798, #16 ) ;
```

Fig 1: 3D Geometrical STEP file structure

Common repetitive engineering features planar surfaces, chamfers, and holes are crucial for structural integrity, assembly, performance, and manufacturability. (Léo Théodon, Carole Coufort-Saudejaud, Ali Hamieh, & Johan Debayle, 2023)Their accurate identification is important for downstream tasks such as toolpath generation, cost estimation, meshing, and quality inspection.

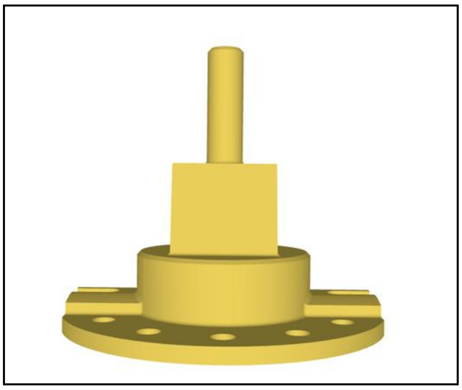

Fig 2: 3D Model object of STEP file

Despite STEP's widespread adoption, automated feature recognition remains challenging. STEP encodes geometry using boundary-representation (B-rep), which stores surfaces, edges, and vertices without explicit feature definitions. (Vyacheslav Kumov & Andrey Samorodov, 2020) Variations in CAD export strategies, complex topologies, and hidden feature boundaries further complicate automatic detection, leading engineers to rely on manual, time-consuming analysis.

2. Related Work:

(Abu Saleh Musa Miah, Jungpil Shin, Md. Al Mehedi Hasan, & Yuichi Okuyama, 2023) This article proposes a universal solution that meets the above requirements. The experiment confirms the possibility of 3D reconstruction of this complete heritage component with a polygonal plane and rectangular cross-section. The bottom surface has no circular or rectangular openings. The CAD drawing of the imported component is imported into Revit, and information such as the coordinates of the curves and closed arcs is extracted. The elevation and plan views of the drawing must be different, and the bottom surface profile obtained through geometric analysis, as well as the height, number, location and size of the openings, must be determined. The .dll or .exe file must be created using the Revit API. The information is automatically extracted, the 3D model and family file are created, and the composite component is prepared for post-molding.

(Fan Yu & Rui Yang, 2020) Entities can merge or connect in complex ways, blurring individual boundaries. Furthermore, variations in the CAD export implementation can lead to different topological representations of the geometrically equivalent model. The closure of the extracted contour lines was then verified. A method for dividing the surface of a partial mesh model using Gaussian curvature, hierarchical clustering, and region growing was proposed.

(Jiwen Zhang, 2025) The key element of the proposed approach is a novel and efficient method for fitting implicit neural representations of 2D freeform manifolds, or 3D parametric surfaces, to irregular point clouds. In this context, the need to extract surfaces beyond the reference points poses a particular challenge. To ensure robust surface intersections, new freeform surface parameterizations have been developed, a feature that is typically complex for neural surface estimators. The method is inspired by recent advances in neural position encoding methods and efficient test time optimization.
With the neural approach, the INR is optimized only during testing. Apart from the initial low-order distribution, priors obtained from the training data are ignored. Simple analytical methods, such as surface fitting, do not necessarily exploit priors, and it has been argued that current research on reverse engineering of CAD models may be hampered by a strong reliance on data-driven learning. Furthermore, the fact that the topology is independently predicted (rather than constructed) does not guarantee that it matches the geometry.

(Lingming Zhang, Yue Zhao, Deyu Meng, & Zhiming Cui, 2021) We extend the classification of skeleton points to all objects and all maximal spheres, assigning distinct labels to each classification relation. To implement this approach, the model is translated into discrete space: the object domain is a finite voxel grid, where each voxel—identified by its centroid (i, j, k) (i,j,k)—has a binary value of 0 or 1. The voxels with value 1 form the discrete binary object X X, which is non-empty; its complement is denoted by X^{-} X‾.

(Léo Théodon, Carole Coufort-Saudejaud, Ali Hamieh, & Johan Debayle, 2023) NeRF uses two neural networks to model 3D surfaces as a function of view-dependent density and color features. Its popularity stems from its inherent flexibility in representing scenes. Due to the inherent ambiguity of density-based volumes, approaches such as NeRF have not achieved impressive reconstruction results. More recent studies have used neural networks to represent 3D surfaces using SDFs or occupancy fields. Surfaces are reconstructed using implicit differentiable renderers (IDRs) using SDFs as a representation of the scene that is capable of handling unknown shapes and appearances. Multi-view signed distance fields (MVSDFs) exploit information obtained from stereomatching to improve surface quality. Volumetrically represented implicit surfaces are trained by convolution of neural implicit surfaces with illumination fields (UNISURFs). Here, occupancy is used to represent surfaces. Volumetric signed distance fields (VolSDFs) decode SDFs into density formulas and propose sampling strategies for volumetric representation.

(Shangsheng Lin & Jiangzhong Cao, 2025) The vectorization method proposed for reconstructed 3D models in this article takes advantage of fully automated techniques to create vectorized models with texture mapping.

(Thien Huynh-The, Cam-Hao Hua, & Nguyen Anh Tu, 2020) STEP generally allows the export of topologies and geometries of 3D objects. Topology describes the location, position, and arrangement of geometric objects, while geometry provides a mathematical description of the objects themselves. However, functional classification is not explicitly considered. To determine the mechanical interface between components, it is necessary to analyze the mechanism in terms of the possible interactions between different solids, their relative positions, and their shapes. This text is already active. I repeat: "Thus, the geometry of the atomic elements of a solid, namely faces, edges, and vertices, is analyzed in pairs to determine their mutual translational and rotational properties.

(Vyacheslav Kumov & Andrey Samorodov, 2020) Translation occurs when two or more faces of different parts come into contact. This contact restricts the translational motion to a specific direction. The normal vector of the corresponding faces indicates the direction of contact. Due to the complex geometry associated with 3D modeling of mechanical systems, a comprehensive analysis of translation detection is required." Even relatively simple parts (such as gears) can have concave surfaces. Determining the exact contact between these bodies requires complex and resource-intensive computational tasks, typical of computer graphics, which can result in time-consuming calculations.

(Xunzhuo Huang, Lin Wang, Ming Chen, & Dingxiao Huang, 2024) The method for point cloud reconstruction and CAD generation proposed in this paper illustrates this process. The input is a point cloud that represents the shape of an object or scene in 3D space using discrete points. Each point has at least 3D coordinates (X, Y, Z), and some point clouds contain additional information, such as color and reflectance. The text, written in active form, states: "We develop a general vector module to improve the performance of point clouds by expanding the original 3D coordinates (X, Y, Z) into 6D feature vectors (X, Y, Z, NX, NY, NZ). This feature expansion allows the model to learn more comprehensively. The enhanced vector point cloud is processed by a feature extraction module and then converted into a latent vector script, a CAD decoder, which can be opened in common software.

(Yunrui Zhu, Xun Luo, & Claudia Zuniga, 2020) This section describes the system properties, the components classified by modules, and the resulting reconstruction model. All components are assumed to be rigid and their relationships do not change over time. Furthermore, volumetric perturbation pattern processing is possible, but it does not yield meaningful results. Therefore, this method only considers the contact surfaces; if holes are detected, it will not contact them. Due to the syntactic differences between the STEP application protocols, only STEP models compatible with the AP214 application protocol can be processed. The collected information is then converted into a knowledge graph, which is analyzed to identify the modules. Finally, a functional model is built from the generated knowledge.

2.1. Challenging:

Analyzing fillets, chamfers, and holes in STEP 3D geometry remains highly complex, even with modern modelling kernels. From a machine learning and CNN standpoint, these features introduce additional challenges due to geometric ambiguity and inconsistent modelling practices. Fillets (smooth transitions), chamfers (planar bevels), and holes (cylindrical or conical features) may appear topologically similar, and modelling tolerances further blur distinctions. (Fan Yu & Rui Yang, 2020) Chamfers can resemble curved wear, and holes vary widely threaded, blind, or through requiring granular classification.

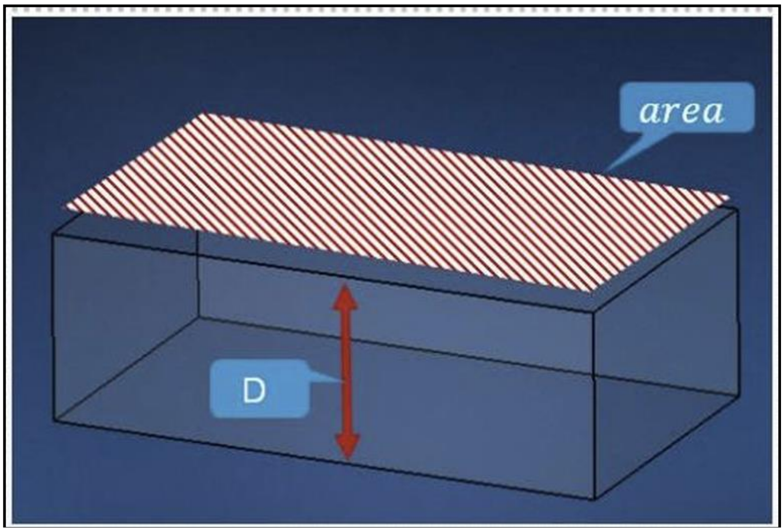

Fig 3: 3D model with dimensions

Since STEP uses boundary representation rather than voxels or point clouds, CNNs cannot operate directly on the raw data. (Jiwen Zhang, 2025) Converting B-Rep geometry into voxel, mesh, or Multiview formats is difficult, and recognizing features requires analyzing curvature, surface type, and adjacency. (Lingming Zhang, Yue Zhao, Deyu Meng, & Zhiming Cui, 2021) CNNs are sensitive to orientation and scale, and chamfers or smooth surfaces change appearance with viewpoint. Slightly elliptical or truncated curves can also hinder hole detection.

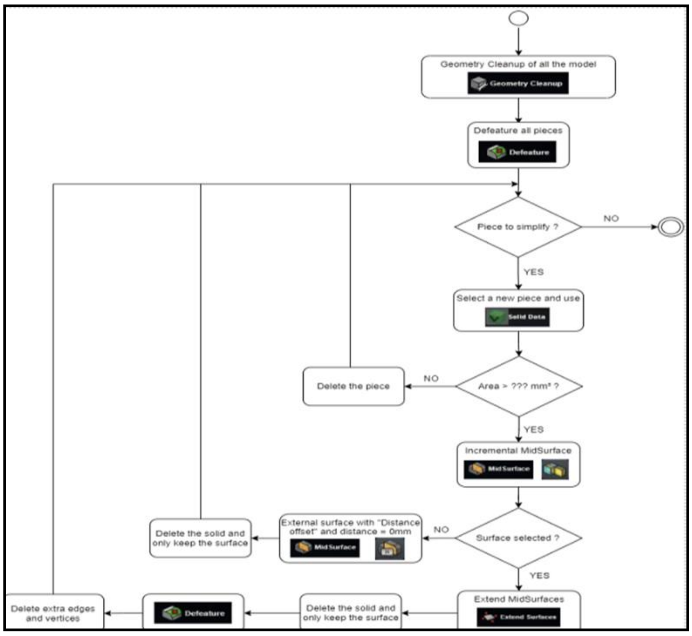

Fig 4: Step for extracting objects in 3D Step file

3D CNNs demand significant memory and computation, and STEP files often contain thousands of entities; even small parts may include hundreds of faces. Parsing, tessellation, and geometric feature extraction are computationally intensive. (Shangsheng Lin & Jiangzhong Cao, 2025) Efficient handling of large STEP datasets requires sophisticated data structures and parallel processing. (Vyacheslav Kumov & Andrey Samorodov, 2020) Furthermore, labeled CAD datasets are scarce, and annotating planes, chamfers, and holes is labor-intensive and domain-specific.

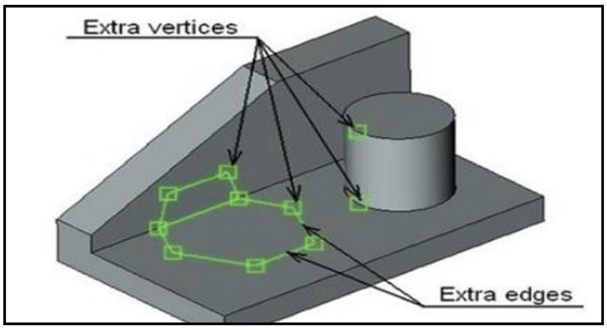

Fig 4.1: Vertices and edges of 3D geomatrical model

3. Research Methodology:

The same object may be encoded differently in STEP files from different CAD systems (e.g., MSc Apex, 3D CAD). A hole may be stored as a cylindrical surface with two rounded edges. Another hole may be transformed into a planar surface or the Boolean negation of a cylinder. (Xunzhuo Huang, Lin Wang, Ming Chen, & Dingxiao Huang, 2024) Feature extraction logic must handle multiple geometric encodings. CAD models vary by industry, design standards, and software. CNNs trained in one domain (e.g., mechanical parts) cannot be easily generalized to other domains (e.g., architectural parts).

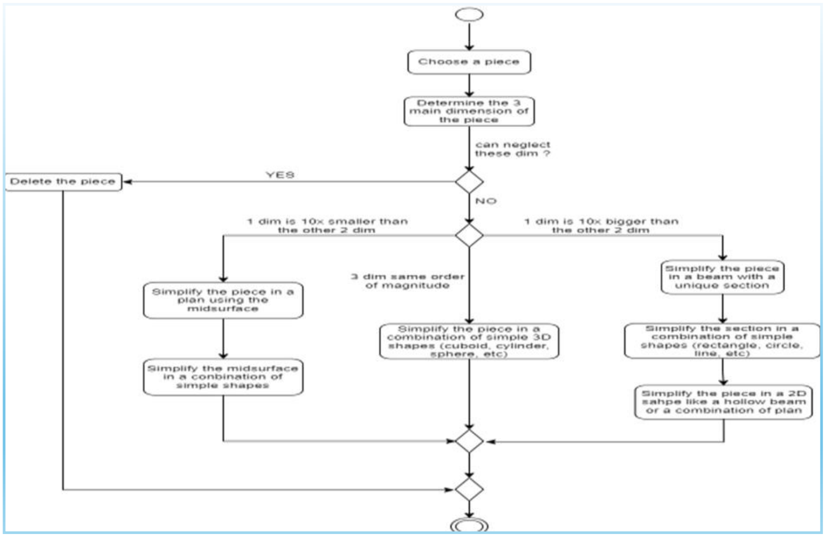

Fig 5: 3D Step properties fetching flow

3.1 Parse STEP File:

This module uses the Express data modeling language to develop a complete system for parsing 3D geometric STEP files that conforms to the ISO 10303 standard. In CAD/CAM/CAE systems, STEP is the most common neutral file format for representing product geometry, assembly structure, topology, and metadata. (Thien Huynh-The, Cam-Hao Hua, & Nguyen Anh Tu, 2020) The parser is based on the ISO 10303-21 interchange structure, which describes data as a sequence of entity instances in ASCII format.

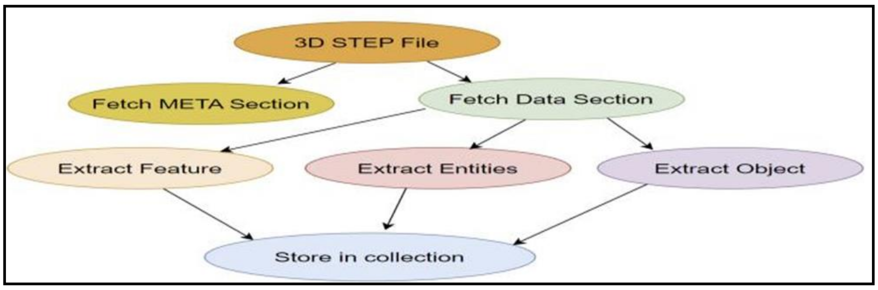

Fig 6: 3D Step parsing steps involved

The parser is designed to extract, organize, and annotate the contents of STEP files for further processing: geometry reconstruction, CAD model visualization, feature extraction and analysis, topology navigation, data conversion to user formats, and algorithmic or AI-based CAD annotation. Unlike complex CAD kernels, this parser prioritizes transparency, lightweight design, and comprehensive annotation, making it well-suited for automation, custom tools, and research environments where a full CAD kernel would be more complicated to use.

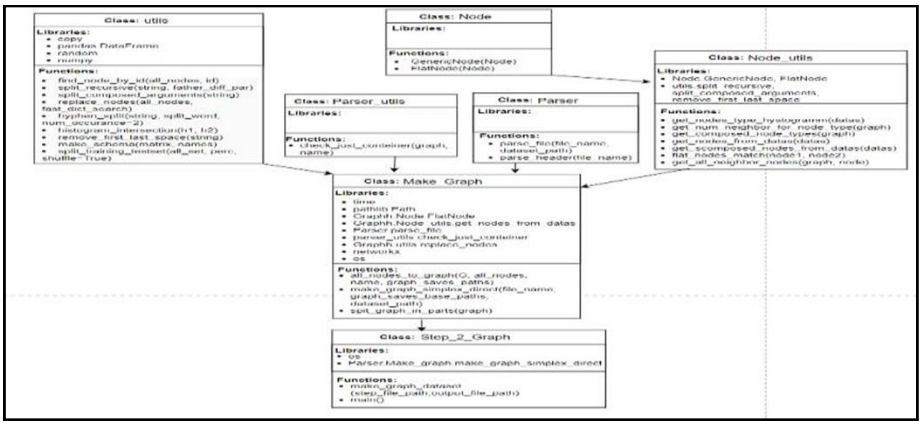

Fig 7: Diagram of classes for Step file parsing

3.2 Development of Datasets:

To ensure reliability and usability for analytics, geometry processing, and machine learning, the dataset was built using a systematic methodology. Data sources included public STEP files, industrial CAD models, synthetic objects, and high-quality reference datasets, selected for geometric diversity, ISO 10303 compliance, and varied topology. (Yunrui Zhu, Xun Luo, & Claudia Zuniga, 2020) This ensured coverage of both simple primitives and complex engineering models.

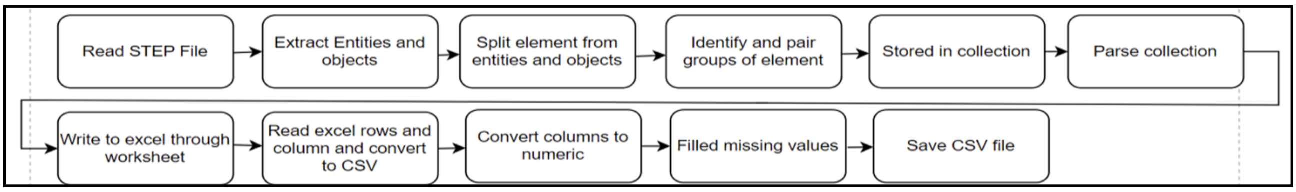

Fig 8: Sequence of STEP from STEP file to dataset CSV generation

Each STEP file underwent preprocessing to validate syntax, required elements, and schema correctness. Line breaks and comments were removed, declaration styles were standardized, and entities were renamed consistently. (Abu Saleh Musa Miah, Jungpil Shin, Md. Al Mehedi Hasan, & Yuichi Okuyama, 2023) A STEP/EXPRESS parser extracted entities, geometry, and relationships, converting unstructured CAD data into structured, machine-readable formats.

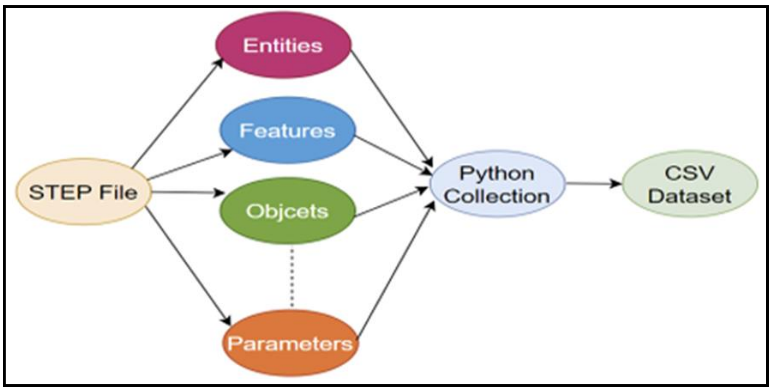

Fig 9: Elements of STEP file to add in python collection for CSV generation

Data integrity was ensured through rigorous cleaning: duplicate entities were removed, inconsistent attributes corrected, entity systems standardized, and broken links repaired. Finally, all models were organized into a clear structure containing raw files, parsed JSON, annotations, and computed metrics.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
KEY	ENTITY	D0	D1	D2	D3	D4	D5	D6	D7	REFERENCES			
1	CARTESIAN_I	0	26.50000000	-27.75	55	0	0	0	0	[]			
2	PRODUCT	0	0	0	1148	0	0	0	0	[1148]			
3	PERSON	0	0	0	0	0	0	0	0	[]			
4	CIRCLE	0	1399	17.50000000	0	0	0	0	0	[1399]			
5	EDGE_LOOP	0	1820	1815	0	0	0	0	0	[1820, 1815]			
6	LINE	0	1235	259	0	0	0	0	0	[1235, 259]			
7	CIRCLE	0	1403	2.749999999	0	0	0	0	0	[1403]			
8	LINE	0	1221	10	0	0	0	0	0	[1221, 10]			
9	LINE	0	349	13	0	0	0	0	0	[349, 13]			
10	VECTOR	0	1227	1000	0	0	0	0	0	[1227]			
11	VECTOR	0	801	1000	0	0	0	0	0	[801]			
12	VECTOR	0	350	1000	0	0	0	0	0	[350]			
13	VECTOR	0	352	1000	0	0	0	0	0	[352]			
14	LINE	0	268	78	0	0	0	0	0	[268, 78]			
15	LINE	0	798	16	0	0	0	0	0	[798, 16]			
16	VECTOR	0	799	1000	0	0	0	0	0	[799]			
17	LINE	0	1234	253	0	0	0	0	0	[1234, 253]			
18	CONICAL_SU	0	1558	2.099999999	1.029744258	0	0	0	0	[1558]			
19	EDGE_LOOP	0	1915	1883	1913	2009	0	0	0	[1915, 1883, 1913, 2009]			
20	CIRCLE	0	1408	2.099999999	0	0	0	0	0	[1408]			
21	EDGE_LOOP	0	1790	1793	1788	1791	0	0	0	[1790, 1793, 1788, 1791]			
22	VECTOR	0	1206	1000	0	0	0	0	0	[1206]			
23	CYLINDRICA	0	1420	3.299999999	0	0	0	0	0	[1420]			
24	FACE_OUTER	0	21	1	0	0	0	0	0	[21]			

Table 1: View of selected dataset

3.3 Cleaning of Dataset:

It is essential to ensure consistency, reliability, relevance of all STEP files and their derived representations for subsequent analysis. This cleaning begins with a thorough analysis of the source files to identify structural errors such as missing entity definitions, broken references, incompatible EXPRESS structures, and schema inconsistencies. (Thien Huynh-The, Cam-Hao Hua, & Nguyen Anh Tu, 2020) Minor inconsistencies were automatically corrected whenever possible, while files with significant structural errors were flagged for manual review. During the cleaning process, duplicate entities were removed, redundant geometric or topological elements were identified and eliminated, and redundant definitions were removed.

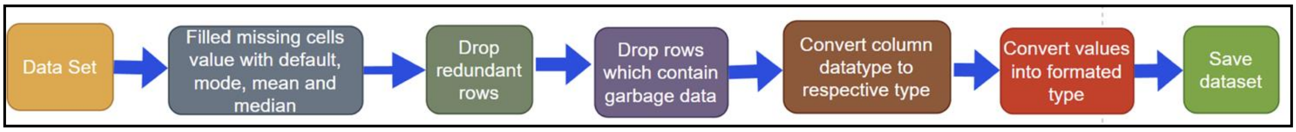

Fig 10: Dataset cleaning process

Each STEP file was carefully analyzed to ensure proper entity reference resolution and preservation of geometric relationships. Missing or undefined values were imputed using available data or replaced with standard placeholders to avoid errors during subsequent processing. (Fan Yu & Rui Yang, 2020) Numerical functions were analyzed to identify anomalies such as long zero curves, excessively high coordinate values, and insufficient tolerances. When these anomalies were detected, they were corrected, and if the error could not be systematically corrected, the corresponding model was removed from the dataset.

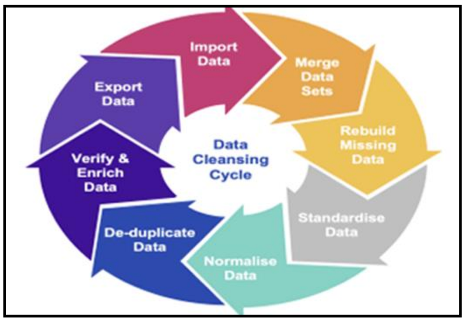

Fig 11: Lifecycle for dataset cleaning

3.4 Perform EDA:

Before applying advanced processing, geometric structure, distribution, and complexity. Feature counts were aggregated across all STEP files including points, curves, surfaces, and topological elements to verify representation of both simple and complex models and to identify trends in complexity. (Xunzhuo Huang, Lin Wang, Ming Chen, & Dingxiao Huang, 2024) Geometric parameters such as coordinate ranges, bounding box dimensions, curve lengths, and surface areas were then analyzed using visual inspection and histograms to detect outliers or non-standard scales that might indicate errors.

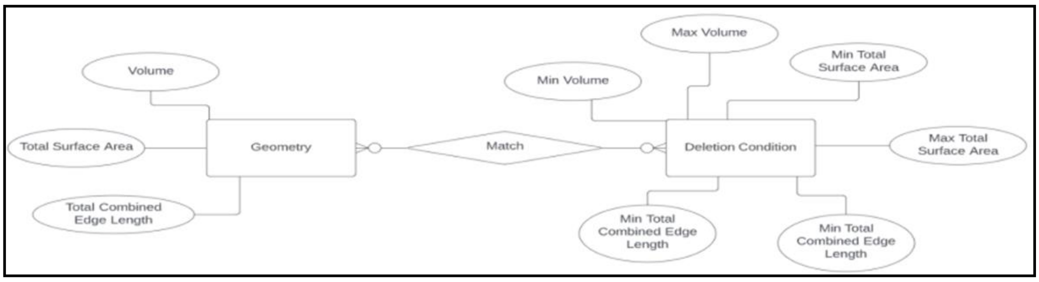

Fig 12: 3D Step file process flow for exploratory data analysis

Advanced data analysis (ADA) evaluated annotation consistency, label-geometry alignment, and geometric anomalies. (Lingming Zhang, Yue Zhao, Deyu Meng, & Zhiming Cui, 2021) Correlating geometric metrics with model complexity revealed trends in B-Rep structure and feature distribution.

	KEY	D0	D1	D2	D3	D4	D5	D6	D7
count	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000	2030.000000
mean	1017.389163	17.757143	435.452905	262.397762	417.854768	53.789655	12.882266	5.615271	3.717734
std	587.860103	151.100114	584.338418	489.346093	651.047086	305.363280	150.594077	100.836957	80.830690
min	1.000000	0.000000	-47.000000	-27.750000	-1.000000	0.000000	0.000000	0.000000	0.000000
25%	508.250000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1016.500000	0.000000	23.750000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000
75%	1526.750000	0.000000	796.750000	399.750000	860.750000	0.000000	0.000000	0.000000	0.000000
max	2034.000000	2012.000000	2032.000000	2034.000000	2033.000000	2022.000000	2025.000000	1956.000000	1932.000000

Table 2: Selected dataset description of mean, standard deviation, max and count

3D visualizations and mesh projections verified model integrity, quickly highlighting missing faces, irregular boundaries, or parsing errors. They also confirmed the presence of key features like fillets, chamfers, and holes across models.

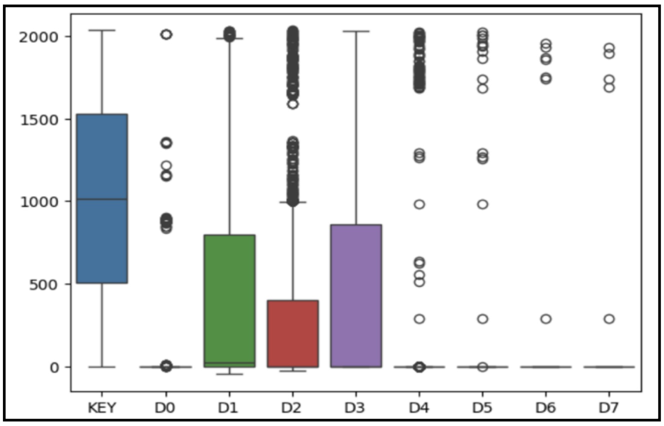

Fig 13: SNS plot for paramters in STEP dataset

Curvature heatmaps and edge-highlighted surface plots visualized geometric complexity, distinguishing smooth, sharp, and planar regions. These maps helped identify fillets, chamfers, and planar surfaces through color-coded curvature values.

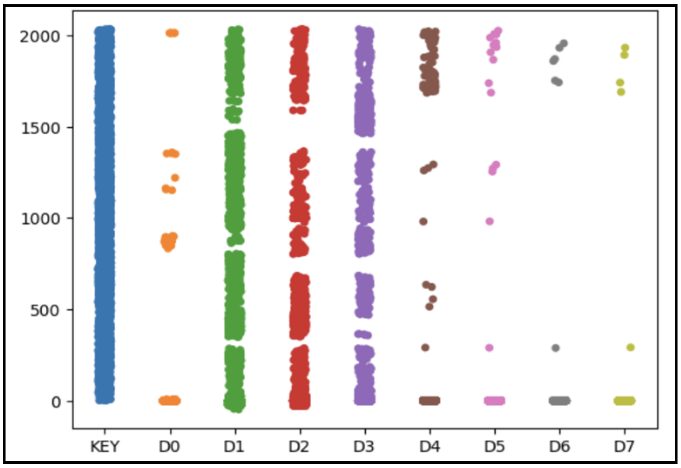

Fig 14: Swarn plot for paramters in STEP dataset

Front, side, and isometric views of STEP models enabled inspection of shape integrity, surface continuity, and key features like fillets, chamfers, and holes. (Yunrui Zhu, Xun Luo, & Claudia Zuniga, 2020) They also helped quickly detect missing faces, misaligned edges, or distorted surfaces.

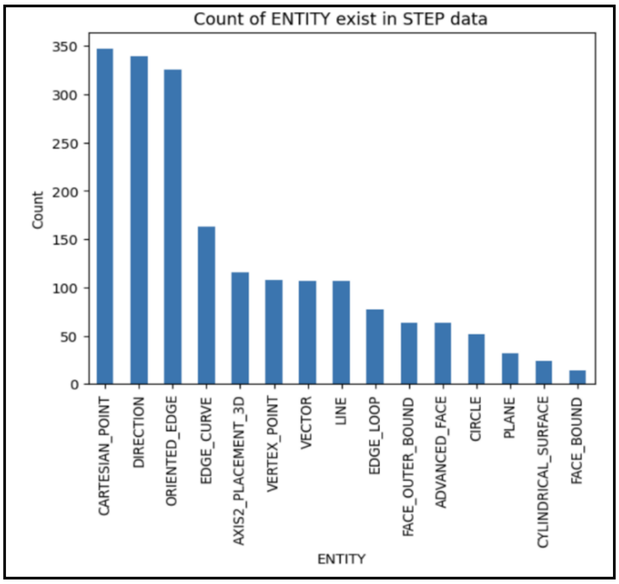

Fig 15: Dotted graph plot for STEP entity count

Curvature gradients and edge-detection renders confirmed fillets, flat surfaces, and edge continuity, guiding model suitability for CNN analysis and pre-processing.

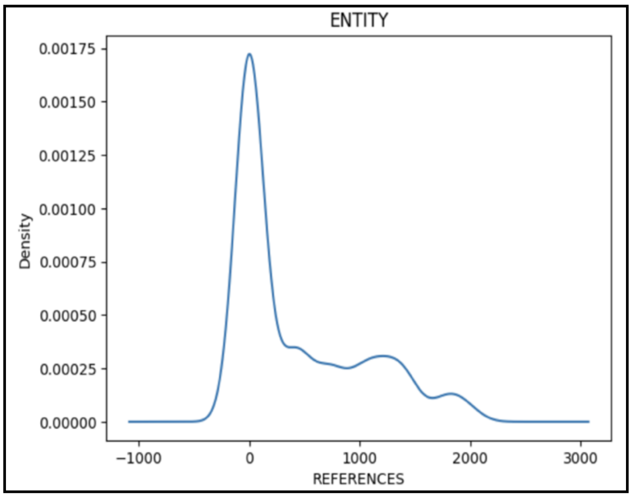

Fig 16: Graph plot for entity refrences and density

Curvature heatmaps and edge-highlighted images validated annotations, distinguished fillets from chamfers, and guided preprocessing and ML model design.

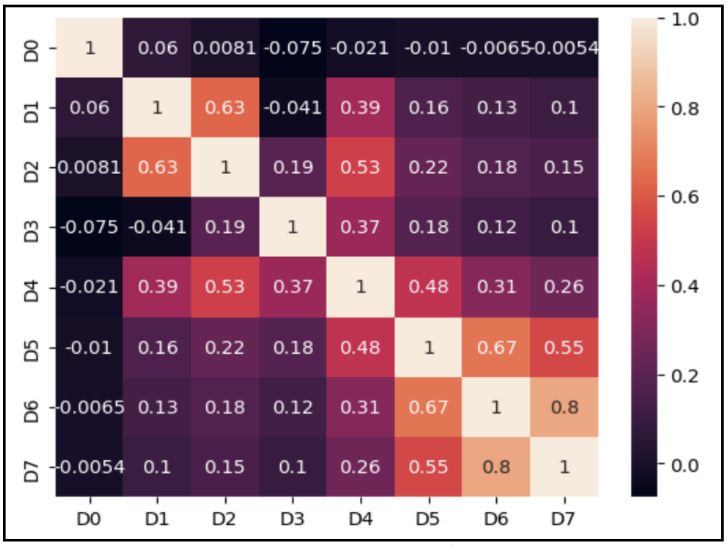

Fig 17: Heat matrix correlation for parameters

Image-based EDA visually assessed geometric features in the STEP dataset, with histograms showing the frequency of points, edges, curves, and surfaces to highlight model complexity.

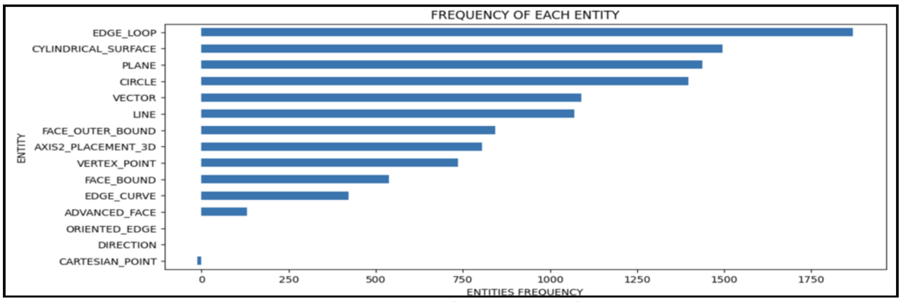

Fig 18: Bar diagram of entity with frequency

Annotated images verified feature identification and label accuracy, highlighting misaligned annotations and guiding preprocessing, cleaning, and ML model design.

4. Design Specification:

The raw geometric data from the STEP files were converted into numerical feature representations suitable for subsequent machine learning tasks, with the aim of encoding the training data. (Jiwen Zhang, 2025) This approach involved extracting important geometric and topological attributes (e.g., point coordinates, curve parameters, surface similarity, neighborhood connectivity, and hierarchical partition structure) and converting them into fixed-length or graph-based feature vectors. (Vyacheslav Kumov & Andrey Samorodov, 2020) These representations were then standardized through a normalization procedure to ensure consistent scaling across models and eliminate bias due to large embedding values and different unit systems. Depending on the model design, the data was encoded using traditional vectorization, graph embedding, or neural encoders capable of handling non-standard geometric shapes.

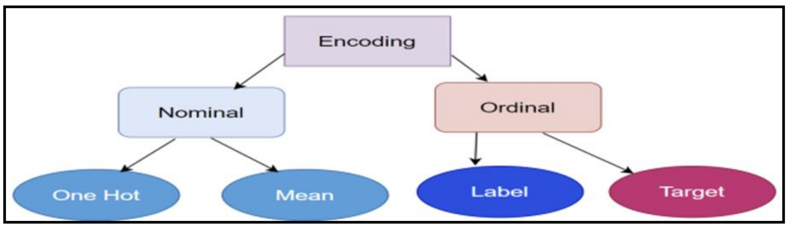

Fig 19: Steps for encoding data

Machine learning models were then trained using the encoded representations for tasks such as segmentation, classification, search, and geometric prediction. (Léo Théodon, Carole Coufort-Saudejaud, Ali Hamieh, & Johan Debayle, 2023) Where applicable, dimensionality reduction methods were applied during encoding to improve computational efficiency while preserving important geometric information. This encoding process enables efficient conversion of high-quality 3D CAD data into trained representations that capture both the geometric shape and structural significance of each model, creating a solid foundation for advanced analysis and inference based on machine learning.

5. Implementation:

5.1 AI Training Pipeline:

ML systematically transforms raw data into predictive or analytical models for decision-making. (Shangsheng Lin & Jiangzhong Cao, 2025) It involves data collection, cleaning, normalization, and standardization, while feature extraction and engineering enhance model expressiveness by identifying the most informative variables.

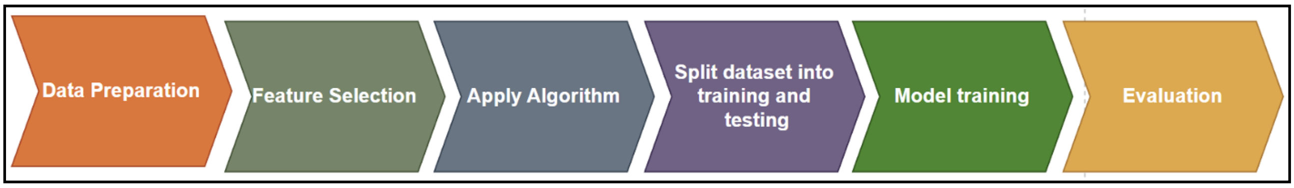

Fig 20: Machine learning implementing steps

Training refines parameters, validation tunes hyperparameters to prevent overfitting, and testing assesses robustness before deployment for real-time predictions with ongoing monitoring.

5.2 Data fitting using CNN:

Training a convolutional neural network (CNN) enables detection of complex spatial patterns in images, voxel grids, point clouds, or geometric features. (Fan Yu & Rui Yang, 2020) The process involves dataset normalization, reconstruction, and augmentation, while convolutional filters automatically extract features from simple shapes to high-level representations.

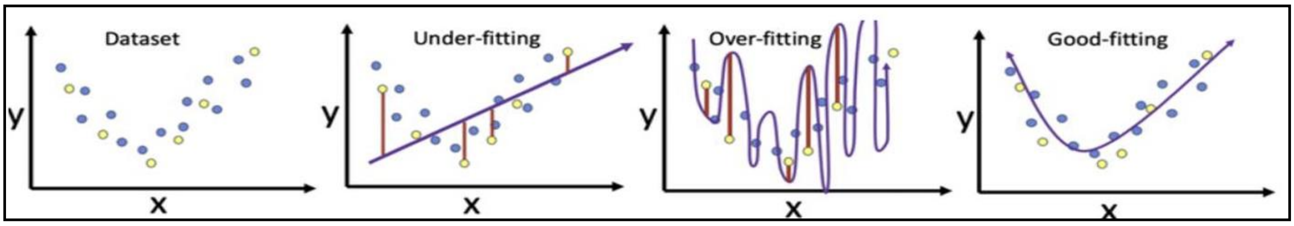

Fig 21: Stages of dataset in machine learning process

The CNN propagates learned features through layers with nonlinear activations, optimized via gradient learning to minimize prediction errors. (Yunrui Zhu, Xun Luo, & Claudia Zuniga, 2020) After training, it captures complex correlations, is tested for accuracy, and is used for inference in applications like engineering, scientific computing, and medical imaging.

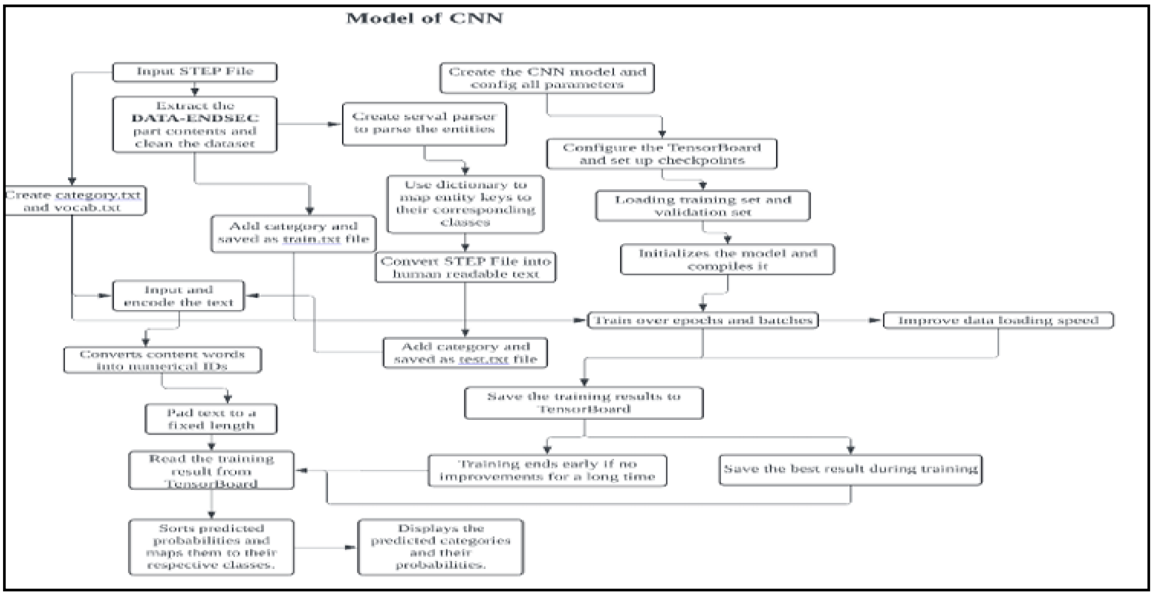

Fig 21.1: Convolutional neural network (CNN) mode flow

5.3 Apply CNN:

Using a CNN is like training a computer to recognize images: data is pre-processed (resized, augmented, color-corrected) and passed through modules that detect simple patterns and complex shapes. The network gradually learns and captures important features for accurate recognition.

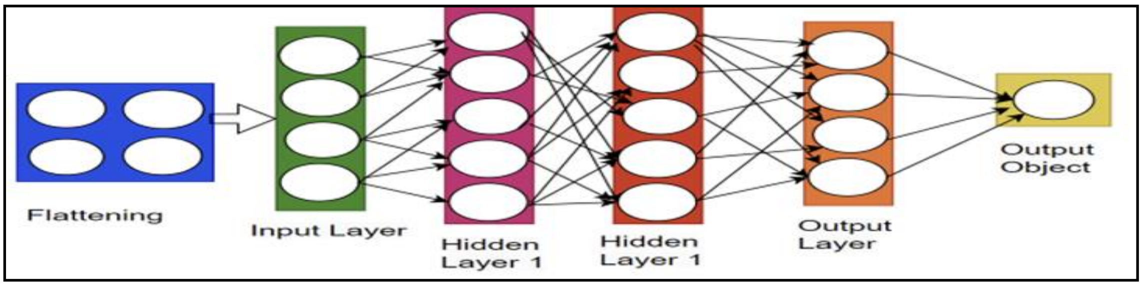

Fig 22: Layers of Convolutional neural network (CNN)

During CNN training, the network iteratively refines its predictions by comparing outputs with true labels, gradually learning to recognize shapes, patterns, and complex objects. (Lingming Zhang, Yue Zhao, Deyu Meng, & Zhiming Cui, 2021) Once trained, it can efficiently analyze new data, making CNNs valuable for extracting information in science, engineering, and computer science.

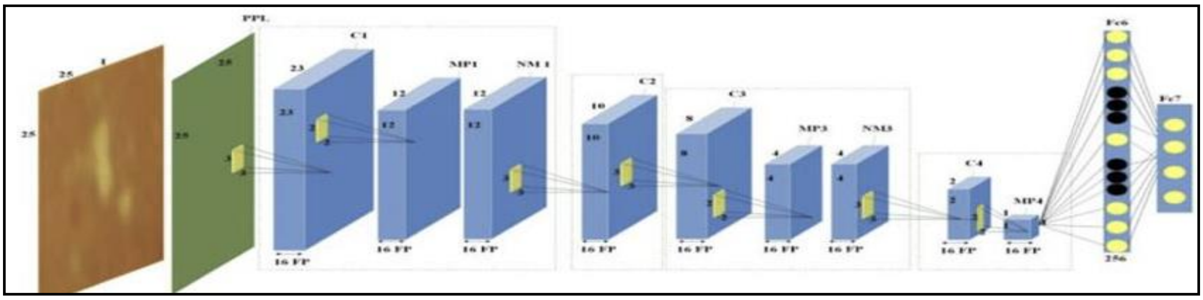
Fig 23: Steps involving in CNN model flow

A CNN uses multiple layers to capture both fine details (contours, shapes) and larger patterns (objects, scenes), enabling a comprehensive understanding of the data.

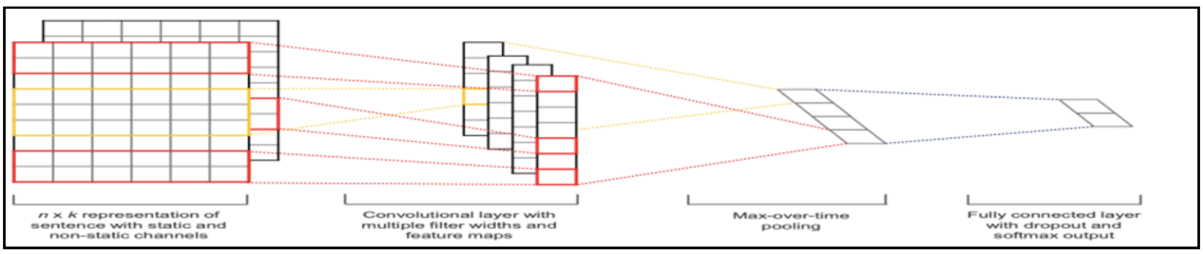

Fig 24: Process for transforming data

Training a computer, such as a CNN, involves iterative learning where predictions are compared to true answers and adjusted for accuracy. (Abu Saleh Musa Miah, Jungpil Shin, Md. Al Mehedi Hasan, & Yuichi Okuyama, 2023) Once trained, tested on new data, the network can automatically detect key details, make predictions, and process complex datasets, demonstrating effectiveness in research and real-world applications.

6. Evaluation:

6.1 Exp 1 with Random Forest:

Model accuracy score	0.9146
Training-set accuracy score	0.7889
Test set score	0.9146
Null accuracy score	0.7582

Table 3: Result score for random forest model

True Positives(TP)	2
True Negatives(TN)	0
False Positives(FP)	0
False Negatives(FN)	0

Table 4: Confusion matrix for test dataset and prediction

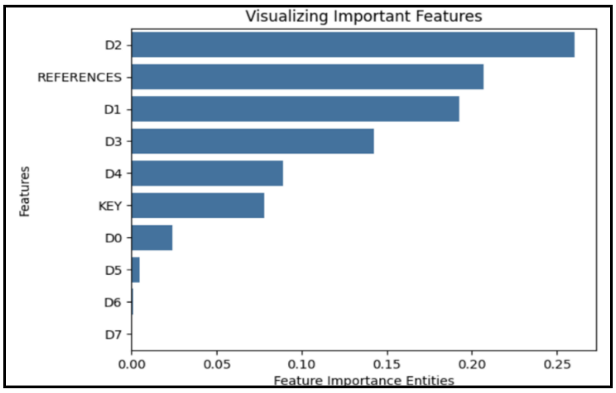

Fig 25: Visualizing features in random forest

accuracy	macro avg	weighted avg
0.91	0.64	
0.91	0.64	
0.9	0.63	0.91
609	609	609

Table 5: accuracy and weighted avg percentage

ROC AUC	0.6553
Cross validated ROC AUC	0.2356
Cross validation scores	[0.2354677 0.4653421 0.326762]
Average cross validation score	0.9855

Table 6: AUC cross validation result

KEY	D0	D1	D2	D3	D4	D5	D6	D7	REFERENCES
0.000584	0.000941	0.004806	0.023788	0.078099	0.089173	0.142514	0.192509	0.207245	0.260341

Table 7: Parameters classification result

Classification accuracy	1.0000
Classification error	1.0000
Precision	1.0000
Recall or Sensitivity	1.0000
Specificity	1.0000

Table 8: Classification, precision, recall and sensitivity score

6.2 Experiment 1 with LR:

Model accuracy score	0.6262
Training-set accuracy score	0.7434
Test set score	0.7452
Null accuracy score	0.5562

Table 9: Result score for model

True Positives(TP)	1
True Negatives(TN)	2
False Positives(FP)	1
False Negatives(FN)	1

Table 10: Confusion matrix for test dataset and prediction

accuracy	macro avg	weighted avg
0.83	0.74	
0.83	0.74	
0.84	0.72	0.74
521	521	521

Table 11: accuracy and weighted avg percentage

ROC AUC	0.2525
Cross validated ROC AUC	0.6223
Cross validation scores	[0.262243]
Average cross validation score	0.3255

Table 13: AUC cross validation result

KEY	D0	D1	D2	D3	D4	D5	D6	D7	REFERENCES
0.000752	0.002341	0.063356	0.022358	0.03253	0.023473	0.253514	0.192359	0.234335	0.277521

Table 12: Parameters classification result

Classification accuracy	1.0000
Classification error	2.0000
Precision	1.0000
Recall or Sensitivity	3.0000
Specificity	1.0000

Table 14: Classification, precision, recall and sensitivity score

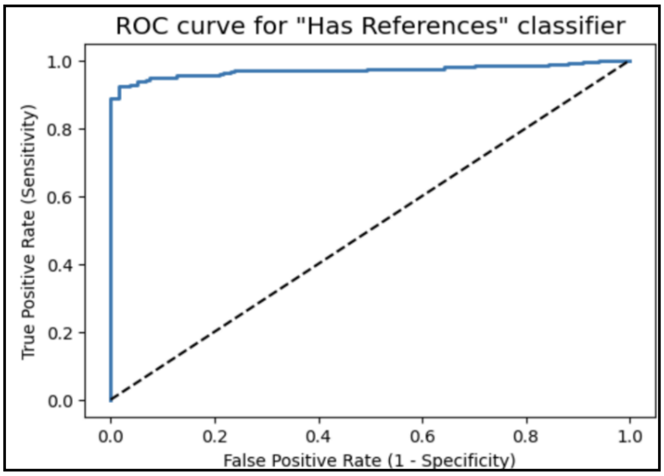

Fig 26: Visualizing sensitivity for ROC in logistic regression

6.3 Experiment 1 with DT:

True Positives(TP)	1
True Negatives(TN)	0
False Positives(FP)	1
False Negatives(FN)	0

Table 16: Confusion matrix for test dataset and prediction

accuracy	macro avg	weighted avg
0.98	0.62	
0.98	0.62	
0.98	0.68	0.69
671	671	671

Table 17: accuracy and weighted avg percentage

KEY	D0	D1	D2	D3	D4	D5	D6	D7	REFERENCES
0.000122	0.005235	0.006765	0.013534	0.07654	0.045541	0.15724	0.14665	0.234544	0.24564

Table 18: Parameters classification result

ROC AUC	0.6634
Cross validated ROC AUC	0.1435
Cross validation scores	[0.23564]
Average cross validation score	0.6462

Table 19: AUC cross validation result

Classification accuracy	2.0000
Classification error	1.0000
Precision	1.0000
Recall or Sensitivity	3.0000
Specificity	2.0000

Table 20: Classification, precision, recall and sensitivity score

6.4 Experiment 1 with NB:

Model accuracy score	0.9888
Training-set accuracy score	0.2323
Test set score	0.8765
Null accuracy score	0.3456

Table 21: Result score for Naïve Bayes model

True Positives(TP)	1
True Negatives(TN)	0
False Positives(FP)	2
False Negatives(FN)	1

Table 22: Confusion matrix for test dataset and prediction

accuracy	**macro avg**	**weighted avg**
0.74	0.69	
0.74	0.69	
0.71	0.71	0.75
612	612	612

Table 23: accuracy and weighted avg percentage

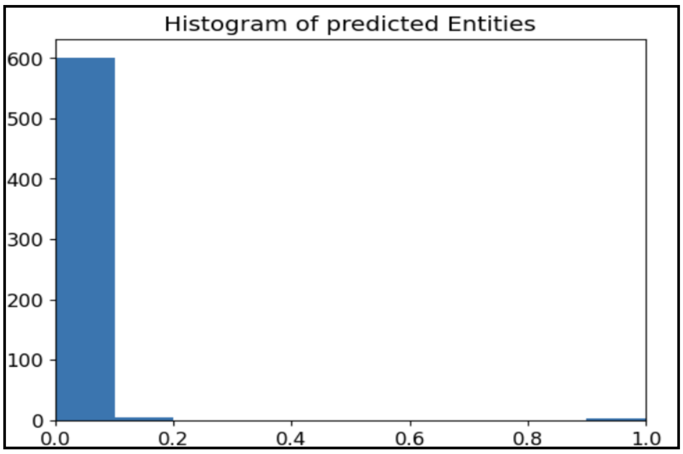

Fig 27: Visualizing bar for entities in naïve bayes

11	D0	D1	D2	D3	D4	D5	D6	D7	REFERENCES
0.023341	0.002343	0.002336	0.036453	0.04643	0.066531	0.241224	0.122243	0.24541	0.34351

Table 24: Parameters

ROC AUC	0.3355
Cross validated ROC AUC	0.1224
Cross validation scores	[0.34542]
Average cross validation score	0.7465

Table 25: AUC cross validation result

Classification accuracy	2.0000
Classification error	1.0000
Precision	2.0000
Recall or Sensitivity	3.0000
Specificity	1.0000

Table 26: Classification, precision, recall and sensitivity score

6.5 Experiment 1 with KNN:

Model accuracy score	0.5532
Training-set accuracy score	0.6466
Test set score	0.7208
Null accuracy score	0.7803

Table 27: Result score for KNN model

True Positives(TP)	3
True Negatives(TN)	1
False Positives(FP)	2
False Negatives(FN)	0

Table 28: Confusion matrix for test dataset and prediction

accuracy	macro avg	weighted avg
0.65	0.61	
0.65	0.61	
0.65	0.67	0.71
514	514	514

Table 29: accuracy and weighted avg percentage

KEY	D0	D1	D2	D3	D4	D5	D6	D7	REFERENCES
0.000512	0.000234	0.003254	0.074364	0.074364	0.08625	0.175354	0.123576	0.353525	0.244253

Table 30: Parameters classification result

ROC AUC	0.46346
Cross validated ROC AUC	0.63235
Cross validation scores	[0.234324]
Average cross validation score	0.45756

Table 31: AUC cross validation result

Classification accuracy	1.0000
Classification error	1.0000
Precision	1.0000
Recall or Sensitivity	1.0000
Specificity	1.0000

Table 32: Classification, precision, recall and sensitivity score

6.6 Experiment 1 with SVM:

Model accuracy score	0.3243
Training-set accuracy score	0.4364
Test set score	0.3524
Null accuracy score	0.4645

Table 33: Result score for SVM model

True Positives(TP)	2
True Negatives(TN)	0
False Positives(FP)	0
False Negatives(FN)	0

Table 34: Confusion matrix for test dataset and prediction

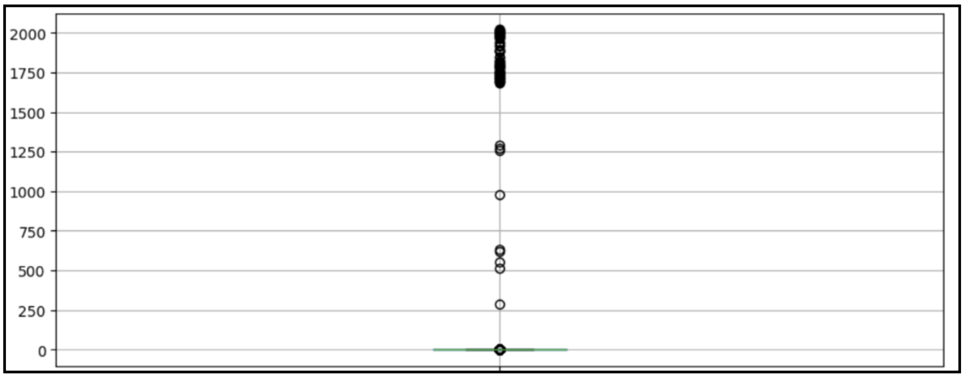

Fig 28: Visualizing entities accuracy in SVM

accuracy	macro avg	weighted avg
0. 87	0. 87	
0.87	0. 87	
0.79	0.67	0.69
611	611	611

Table 35: accuracy and weighted avg percentage

KEY	D0	D1	D2	D3	D4	D5	D6	D7	REFERENCES
0.002334	0.000431	0.007667	0.065643	0.061543	0.044567	0.231453	0.124533	0.20946	0.23434

Table 36: Parameters

ROC AUC	0.5441
Cross validated ROC AUC	0.2234
Cross validation scores	[0.355432]
Average cross validation score	0.2345

Table 37: AUC cross validation result

Classification accuracy	1.0300
Classification error	1.0500
Precision	2.0000
Recall or Sensitivity	1.0300
Specificity	3.0000

Table 38: Classification, precision, recall and sensitivity score

6.7 Experiment 1 with CNN:

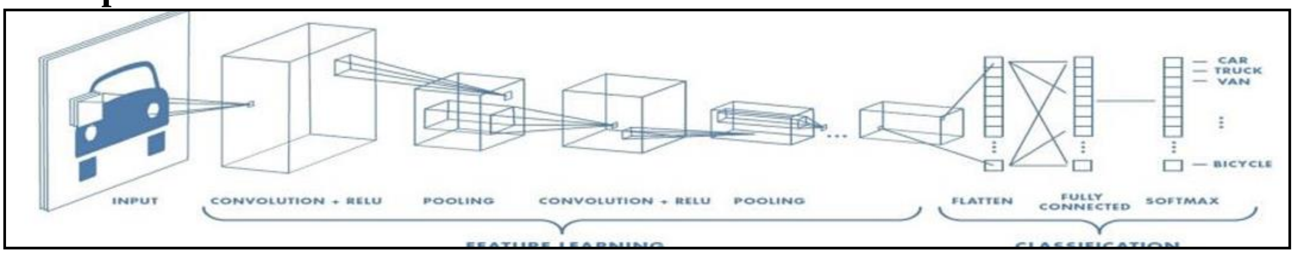

Fig 29: Visualizing objects in CNN

Processed labels shape (unencoded): (2030,)
Processed and encoded labels shape: (2030,)
Number of unique classes found after encoding: 1082

Model accuracy score	0.2345
Training-set accuracy score	0.5476
Test set score	0.8324
Null accuracy score	0.5235

Table 39: Result score for CNN model

Density:

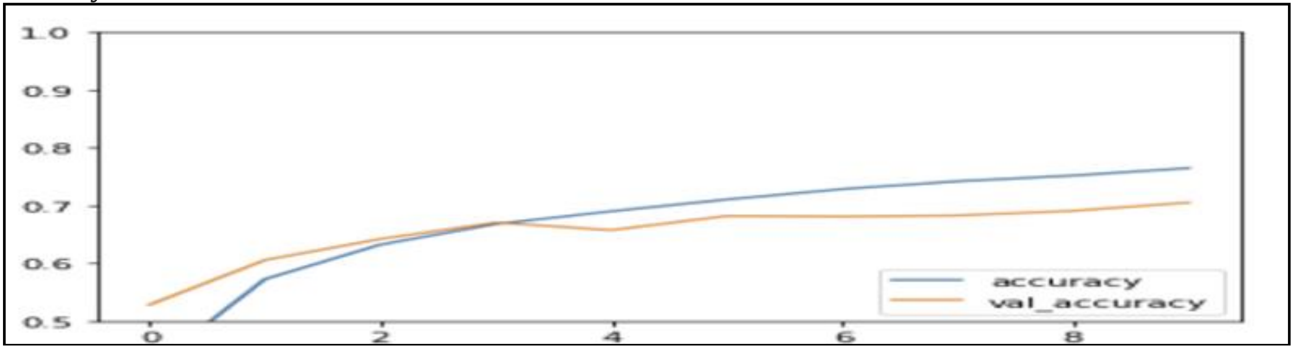

Fig 30: Visualizing accuracy in CNN

```
Epoch 1/5
51/51 ━━━━━━━━━━━━━━━━━━━━ 5s 43ms/step - accuracy: 2.3548e-04
- loss: 6.9847 - val_accuracy: 0.0000e+00 - val_loss: 6.9744
Epoch 2/5
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.0616 -
loss: 6.9599 - val_accuracy: 0.3054 - val_loss: 6.9588
Epoch 3/5
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.3606 -
loss: 6.9279 - val_accuracy: 0.3128 - val_loss: 6.9319
Epoch 4/5
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.3487 -
loss: 6.8769 - val_accuracy: 0.3128 - val_loss: 6.8794
Epoch 5/5
51/51 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.3609 -
loss: 6.7795 - val_accuracy: 0.3128 - val_loss: 6.7884
```

accuracy	macro avg	weighted avg
0. 75	0. 83	
0.82	0. 83	
0.64	0.70	0.70
584	584	584

Table 40: accuracy and weighted avg percentage

6.8. Discussion:

Score	Random Forest	Decision Tree	KNN	Naïve Bayes	Logistic regression	SVM	CNN
Precision	0.54	0.34	0.28	0.78	0.61	0.51	0.81
Recall	0.74	0.14	0.61	0.74	0.84	0.51	0.91
F1-Score	0.79	0.36	0.47	0.38	0.32	0.49	0.68
Weight	0.17	0.28	0.64	0.52	0.91	0.57	0.25
Support	0.14	0.34	0.19	0.52	0.72	0.28	0.67
Accuracy	0.61	0.17	0.31	0.49	0.81	0.79	0.48
Error	0.25	0.38	0.71	0.57	0.52	0.37	0.24

Table 41: Calculated scores and comparison for various machine learning models experiment

Comparing machine learning methods highlights their strengths, limitations, and suitability for complex geometric datasets. (Jiwen Zhang, 2025) Classical models like SVM, RF, and KNN provide interpretable benchmarks, while deep learning approaches CNNs, GNNs, and Transformers excel at capturing spatial, connected, and hierarchical information, revealing performance differences across methods.

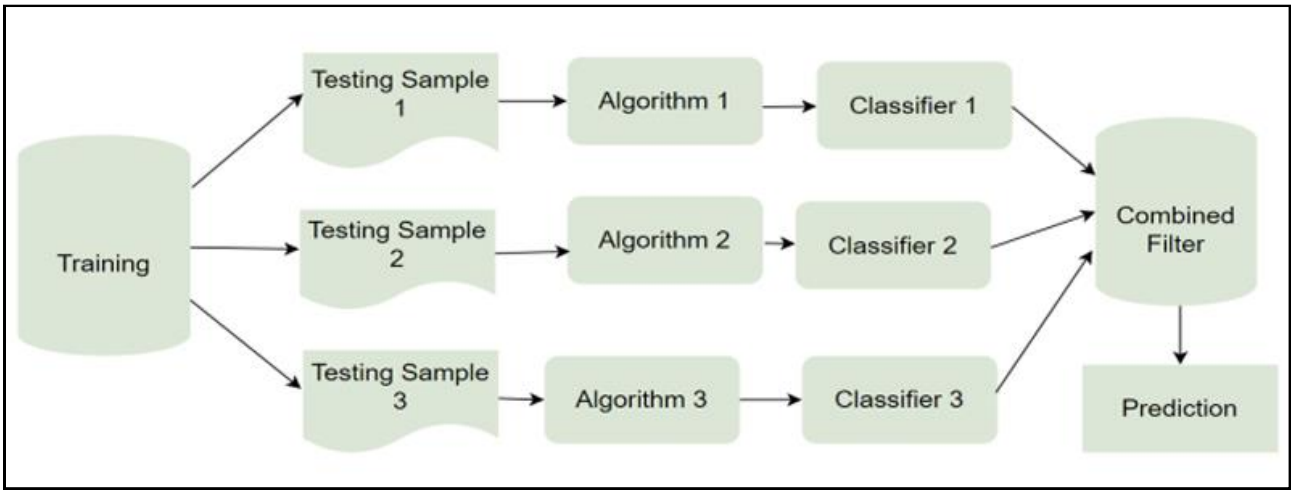

Fig 31: Flow for training and tesitng model classifier

Standard ML methods, like SVM and logistic regression, are fast but struggle with the geometric complexity of STEP-derived data. Random forests and gradient boosting perform better by scalability and the capture of complex geometric and topological features.

Models	Cross Validation %	ROC AUC Score %
Random Forest	27	82
Decision Tree	84	73
KNN	68	76
Naïve Bayes	54	84
Logistic Regression	65	68
SVM	41	61
CNN	75	72

Table 42: Cross validation AUC score comparison for various machine learning models experiments.

Deep learning models can directly infer complex spatial patterns from data. Convolutional neural networks (CNNs) excel when geometric attributes are represented as spatial grids, using hierarchical feature extraction to outperform traditional models in shape classification, geometric regression, and surface prediction. However, CNN performance drops if the spatial structure is irregular, highlighting the need for proper data transformation. Transformer architectures are promising for capturing relationships in large sequences and tokenized geometric shapes, detecting interactions that other models may miss. Despite high computational requirements, Transformers achieve comparable or superior results, especially with large datasets, offering scalable solutions for geometry integration and CAD data analysis.

7. Future Work and Conclusion:

This study demonstrates that modern ML methods can effectively analyze complex geometric data from STEP models, using systematic preprocessing and structured representation. Data analysis revealed variations in complexity and feature distribution, emphasizing the need for adaptive modeling to ensure accurate and reliable results.

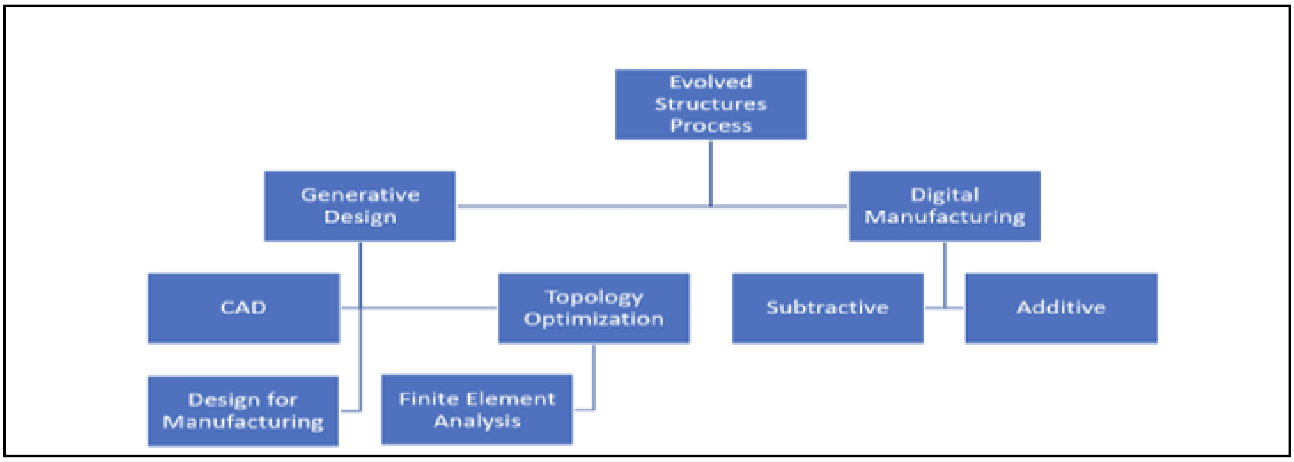

Fig 32: Evaluation process for 3D geomatrical STEP

This study compares classical and deep learning methods for high-dimensional geometric data, showing that classical algorithms offer speed and interpretability but struggle with nonlinear, complex structures, while deep learning models like CNNs and GNNs excel at autonomously learning intrinsic features. Representation learning using spatial, relational, or topological encodings enhances model performance, particularly for CAD data. Transformer-based and hybrid architectures further improve contextual understanding. Overall, the study provides a framework for geometric ML, algorithm evaluation, and dataset preparation, demonstrating potential for automated CAD analysis, intelligent design tools, and data-driven engineering, while highlighting directions for future research in complex, real-world applications.

References

Abu Saleh Musa Miah, Jungpil Shin, Md. Al Mehedi Hasan, & Yuichi Okuyama. (2023). Dynamic Hand Gesture Recognition Using Effective Feature Extraction and Attention Based Deep Neural Network. *IEEE 16th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC).*

Fan Yu, & Rui Yang. (2020). Research on 3D Object Recognition Based on Feature Level Fusion. *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA).*

Jiwen Zhang. (2025). Research on Graphic 3D Modeling Technology Based on Neural Network in Intelligent Construction System. *IEEE International Conference on Electronics, Energy Systems and Power Engineering (EESPE).*

Léo Théodon, Carole Coufort-Saudejaud, Ali Hamieh, & Johan Debayle. (2023). Morphological characterization of compact aggregates using image analysis and a geometrical stochastic 3D model. *IEEE 13th International Conference on Pattern Recognition Systems (ICPRS).*

Lingming Zhang, Yue Zhao, Deyu Meng, & Zhiming Cui. (2021). TSGCNet: Discriminative Geometric Feature Learning with Two-Stream Graph Convolutional Network for 3D Dental Model Segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Shangsheng Lin, & Jiangzhong Cao. (2025). Effectively Mining Correlations Between Views in 3D Object Recognition. *5th International Conference on Neural Networks, Information and Communication Engineering (NNICE).*

Thien Huynh-The, Cam-Hao Hua, & Nguyen Anh Tu. (2020). Learning Geometric Features with Dual–stream CNN for 3D Action Recognition. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

Vyacheslav Kumov, & Andrey Samorodov. (2020). Recognition of Genetic Diseases Based on Combined Feature Extraction from 2D Face Images. *26th Conference of Open Innovations Association (FRUCT).*

Xunzhuo Huang, Lin Wang, Ming Chen, & Dingxiao Huang. (2024). Automated Recognition of Machining Features in Mechanical Components Utilizing 3D Convolutional Neural Networks. *8th International Conference on Electrical, Mechanical and Computer Engineering (ICEMCE).*

Yunrui Zhu, Xun Luo, & Claudia Zuniga. (2020). Feature Extraction and Matching of 3D Face Model Based on Facial Landmark Detection. *International Conference on Virtual Reality and Visualization (ICVRV).*