The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-025-16118-1

ORIGINAL ARTICLE

AutoSplit: a two-stage AI architecture for enhanced classification of manufacturing processes with a focus on the identification of additive manufacturing components

Mehdi Nazarian[1] · Rafael Neves[1] · Léon Klick[2] · Fabian Schöfer[3] · Robert Lau[1] · Arthur Seibel[3] · Felix Weigand[1]

Received: 9 February 2025 / Accepted: 10 July 2025 / Published online: 29 July 2025

Abstract

In the product development phase of mechanical assemblies, engineers encounter an increasing variety of potential manufacturing routes for metal parts. Despite the advantages of additive manufacturing (AM), conventional methods often dominate due to a lack of interdisciplinary knowledge required for additive or hybrid manufacturing approaches. To streamline the development of hybrid manufactured components, this paper presents a novel two-stage methodology for automating part classification in manufacturing processes. A two-stage classification approach was selected to filter standard parts (e.g., screws, nuts, bolts), enabling a pre-filtering step that improves classification performance and reduces overfitting by minimizing the number of ST-components with similar features. The first stage employs convolutional neural networks (CNNs) for image-based classification and multi-layer perceptrons (MLPs) for feature-based classification, achieving 88.84% ±0.6 (SD) accuracy in differentiating standard from non-standard parts. The second stage utilizes a random forest classifier to categorize non-standard parts into three manufacturing processes (AM, machining, and sheet metal), achieving 82.0% ±1.1 (SD) accuracy, with particularly strong performance in machining identification ($F1$-score: 0.85 ±0.03 (SD)). The approach is trained on a comprehensive dataset of 20,000 CAD files sourced from GrabCAD, Fusion360, and TraceParts, evenly distributed across four categories. System performance was evaluated using fivefold cross-validation, demonstrating robust generalization across diverse part geometries and materials. This methodology provides guidance for selecting appropriate manufacturing routes for both redesigns and new designs.

Keywords Industry 4.0 · Manufacturing process classification · Convolutional neural network · Feedforward neural network · Additive manufacturing · Random forest classification

Abbreviations

AM	Additive manufacturing
CAD	Computer-aided design
STEP	Standard for the exchange of product data
CNN	Convolutional neural network
FNN	Feedforward neural network
PLM	Product lifecycle management
MLP	Multi-layer perceptron
ML	Machine learning
DL	Deep learning
PBF-LB/M	Powder bed fusion-laser beam/metal
RF	Random forest
SMOTE	Synthetic minority over-sampling technique
OCCT	OpenCasCade Technology
ST-Parts	Standard parts
3D	Three dimensional
ISO	Isometric (projection)
B-rep	Boundary representation
CV	Cross-validation
RGB	Red green blue (color model)
ROC	Receiver operating characteristic
SVM	Support vector machine
UV-Net	UV mapping neural network

✉ Mehdi Nazarian
mehdi.nazarian@iapt.fraunhofer.de

[1] Fraunhofer Institute for Additive Production Technology IAPT, Am Schleusengraben 14, 21029 Hamburg, Germany

[2] Autoflug GmbH, Industriestraße 10, 25462 Rellingen, Germany

[3] Leuphana University Lüneburg, Universitätsallee 1, 21335 Lüneburg, Germany

GPU	Graphics processing unit
CPU	Central processing unit
SD	Standard deviation
AUC	Area under the curve (typically ROC curve)
ResNet	Residual network
AFR	Automatic feature recognition
CV-Mean	Cross-validation mean
CV-Std	Cross-validation standard deviation
avg.	Average
D (e.g., 256)	Dense layer with 256 neurons

1 Introduction

Efficient manufacturing processes are essential for modern industry, including Industry 4.0. One promising approach to enhance efficiency is the integration of various manufacturing methods, commonly referred to as hybrid manufacturing [1, 2].With the rise of AM, numerous strategies for combining AM with conventional manufacturing techniques have been developed, revealing significant potential [3]. However, despite these advantages, widespread adoption in industrial sectors remains limited due to a lack of interdisciplinary knowledge during the product development phases. Designers often spend considerable time searching for relevant information and frequently rely on existing designs for new product development [4]. Therefore, accurately categorizing and optimizing different manufacturing processes is essential for effective product development, production planning, and maximizing efficiency [5, 6]. As the complexity and variety of mechanical parts continue to grow, along with the volume of computer-aided design (CAD) data, the need for accurate and efficient methods to categorize manufacturing processes becomes increasingly important [6, 7].

Over the last decades, the research field of automatic feature recognition (AFR) has introduced several new approaches to identify parts for conventional manufacturing. The early developed rule-based approaches rely on predefined rules based on extensive knowledge of researchers, facing a major drawback in flexibility [8, 9]. More recently, advanced learning-based approaches were used to improve the accuracy for the identification of varying machining features in comparison to rule-based approaches [10–12]. Despite the increased flexibility through learning-based approaches, the research predominately focuses on machining features. The wide variety of new design possibilities with hybrid manufacturing approaches, especially including AM, is not considered yet. Domain-specific datasets are used which do not represent the major differences of typical geometric features in AM, machining, or sheet metal manufacturing.

The challenge lies in developing a classification system that can accurately differentiate between various manufacturing processes while also accounting for standard (ST) parts commonly available in the market. ST-Parts (e.g., screws, nuts, bolts) often share similar features with custom-manufactured components, such as those produced through AM or machining processes. This similarity can lead to misclassifications in traditional single-stage classification systems [13]. This study aims to cover the combination of AM, sheet metal, machining, and ST-Parts in a potential hybrid manufacturing approach.

To address the challenge of increased variation of geometric features, a two-stage architecture for metal manufacturing process classification is proposed. Developed as part of the AutoSplit project, this approach leverages data mining, data analysis, and machine learning techniques to efficiently uncover complex patterns from large datasets of individual parts. The two stages of the proposed architecture are visualized in Fig. 1 and can be summarized as follows:

Sage 1 focuses on reducing misclassifications and improving system accuracy by filtering out ST-Parts. Stage 2 then categorizes non-ST-Parts based on their specific manufacturing processes.

The classifier is defined as a composition of two functions with:

$$f(x) = f_2\left(f_1(x)\right) \tag{1}$$

- f_1: first classifier (standard vs. non-standard)
- f_2: second classifier (AM, machining, sheet metal)

Overall: $f = \mathbb{R}^n \rightarrow \{\text{AM, machining, sheet metal}\}$.
where $\mathbb{R}^n$ represents the feature space.

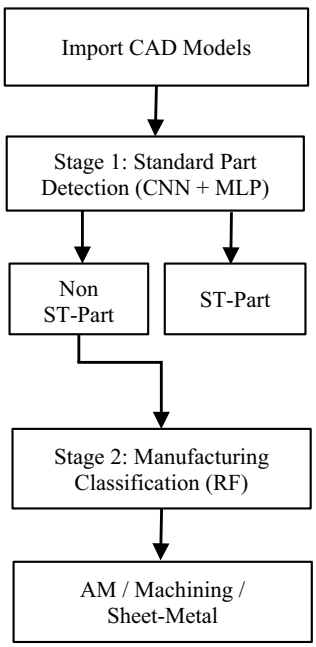


Fig. 1 Overview of the AutoSplit classification framework

2 Literature review

The classification of 3D CAD models has become increasingly important in various industries, including manufacturing, architecture, and product design. While extensive research has been conducted on 3D graphical models and images, studies specifically addressing the classification of 3D CAD models are relatively scarce. This section explores the evolution of relevant techniques and their application to the 3D CAD model classification.

2.1 Traditional methods for 3D model classification

Early approaches to the 3D model classification relied heavily on geometric feature extraction and traditional machine learning algorithms. Ip et al. [14] proposed a method using shape distribution signatures and neural networks for the 3D CAD model classification. This work laid the foundation for feature-based approaches in the field. Subsequently, Li and Godil [15] introduced a technique employing both global and local features, demonstrating improved classification accuracy.

2.2 Neural networks for 3D model classification

The advent of deep learning has revolutionized various classification tasks, including those involving 3D models. Neural networks, particularly CNNs and FNNs, have shown remarkable performance in image and data classification tasks [16, 17].

2.2.1 Convolutional neural networks (CNNs)

CNNs have become the de facto standard for image classification tasks due to their ability to automatically learn hierarchical features from raw pixel data. The architecture of a CNN typically comprises a series of convolutional layers, pooling layers, and fully connected layers [18, 19]. CNNs are based on the idea that spatial relationships in image data can be captured through hierarchical feature extraction, making them particularly effective for tasks such as image recognition and classification. The use of weight sharing in convolutional layers contributes to the efficiency of CNNs, as the same filters are applied across the entire image, reducing the number of parameters and improving model generalization [20, 21].

In the context of 3D model classification, Su et al. [22] introduced a multi-view CNN approach, where 2D renderings of 3D models from multiple viewpoints were used as input to a CNN. This method effectively bridged the gap between 2D image classification techniques and 3D model classification tasks.

2.2.2 Feedforward neural network (FNN) and multi-layer perceptron (MLP)

FNNs, particularly MLPs, have been widely used for various classification and regression tasks [23]. The architecture of an FNN typically includes an input layer, one or more hidden layers, and an output layer, with each neuron in a layer fully connected to the neurons in the subsequent layer [24]. Information flows unidirectionally from the input to the output without any feedback loops [25]. MLPs have shown remarkable performance in processing numeric features, learning to map input data to the corresponding output through supervised learning [26].

2.2.3 Random forest for 3D model classification

While neural networks have become prominent in recent advancements, traditional machine learning methods, particularly ensemble techniques like random forests, continue to demonstrate effectiveness in 3D model classification tasks. Random forests are well-suited to high-dimensional data, offering strong performance and valuable feature importance rankings [27]. Notably, Tombari et al. [28] employed random forests with 3D local surface features for object recognition in point cloud data, achieving accurate classification results. Similarly, Zhang et al. [29] conducted a survey on a 3D point cloud classification, highlighting how traditional methods such as random forests have remained essential benchmarks despite the rise of deep learning approaches.

In the context of a CAD model classification, Ip and Regli [30] applied random forests to classify 3D CAD models based on machining features, showcasing the algorithm's applicability in the manufacturing domain. The consistent success of random forests across these applications underscores their robustness, resistance to overfitting, and utility in handling complex, high-dimensional feature spaces.

2.2.4 Hybrid model: combining CNN and MLP

Recent advancements in deep learning have led to the development of hybrid models that combine the strengths of different neural network architectures. Hybrid models that integrate CNNs and MLPs have shown particular promise in tasks that involve both image and numeric data.

In such hybrid models, CNNs are typically used to automatically extract spatial features from image data, while MLPs process additional numeric features or further refine the CNN-extracted features. This combination allows the model to

leverage both visual and non-visual data, potentially improving classification performance [31].

2.3 Current challenges and limitations in 3D CAD model classification

Despite these advancements, the classification of 3D CAD models presents unique challenges. Unlike 3D mesh models or point clouds, CAD models often contain precise geometric information and hierarchical structures that are not easily captured by standard CNN or MLP architectures. Qi et al. [32] proposed PointNet, a neural network architecture designed to process point cloud data directly. While this approach shows promise for certain types of 3D data, it may not fully capture the structured nature of CAD models. To address these challenges, Jayaraman et al. [33] introduced UV-Net, a neural network specifically designed to operate on boundary representation (B-rep) data, which is prevalent in CAD applications and encodes both geometric and topological features essential for accurate CAD modeling. Nevertheless, UV-Net is not without limitations; it relies on a fixed sampling resolution for representing geometry, a constraint that can impede the capture of finer model details without incurring substantial computational costs. While recent models such as PointNet + + or BRepNet more directly operate on native 3D structures, our image-based approach enables easier integration with CAD rendering workflows and incurs lower computational overhead in our deployment context—making it more practical for early-stage design pipelines [32, 34].

3 AutoSplit framework: methodology and system architecture

The proposed classification method supports early product development by analyzing CAD components to suggest suitable manufacturing routes. It applies to both new designs and redesigns, helping identify hybrid manufacturing strategies. In this study, AM refers specifically to laser-based powder bed fusion (PBF-LB/M). The two-stage pipeline first classifies parts as standard or non-standard. If non-standard, a random forest model uses numerical features to assign the part to AM, machining, or sheet metal. Figure 2 shows an example where a non-standard part is classified as machining.

4 Data processing

As outlined, this study employs a two-stage architecture, each stage serving distinct classification objectives. Consequently, two datasets are required, each prepared individually based on the following process.

4.1 Data acquisition

Data acquisition is the essential first stage in building a machine learning model for the 3D CAD classification, involving the collection and preparation of relevant data for training and evaluation [35]. A diverse dataset of 3D CAD models in STEP format was collected, encompassing ST-Parts and components categorized into four groups: additive manufacturing, machining, sheet metal, and ST-Parts. Three primary online sources were utilized to gather this collection of STEP models:

- Fusion360 gallery: Approximately 17,000 parts were sourced from this dataset [36]. This large dataset provides a wide variety of parts, with a notable prevalence of machining-suitable components.
- GrabCAD library: Approximately 2300 industrial parts were obtained from this public CAD library [37]. Keywords related to AM and sheet metal parts were prioritized. This targeted approach effectively balanced the dataset, particularly for underrepresented manufacturing processes.
- TraceParts: Around 700 parts were sourced from this library [38]. These were primarily utilized in the first stage to collect underrepresented ST-Parts. This library serves as a key resource for obtaining commonly used components.

The aggregation of these sources resulted in a dataset comprising about 20,000 CAD STEP models, encompassing a diverse range of manufacturing processes and part geometries.

Figure 3 illustrates the distribution of the data within the dataset.

To ensure data quality, algorithmic preprocessing was employed to eliminate duplicates and exclude STEP files containing assemblies, as multi-part models could introduce ambiguity and negatively impact the classification accuracy. Furthermore, models with names containing terms such as "covid" "chess," or "phone" were systematically filtered out to avoid the overrepresentation of 3D-printable COVID-19 face shields, chess pieces, and phone cases, which could introduce biases and compromise the generalizability of the model.

4.2 Data labeling

Three mechanical process engineering experts with > 5 years' experience, each with over 5 years of industrial experience in manufacturing processes, independently classified CAD models into four classes, with final labels determined by majority vote to minimize bias. The classification criteria, developed through literature review and

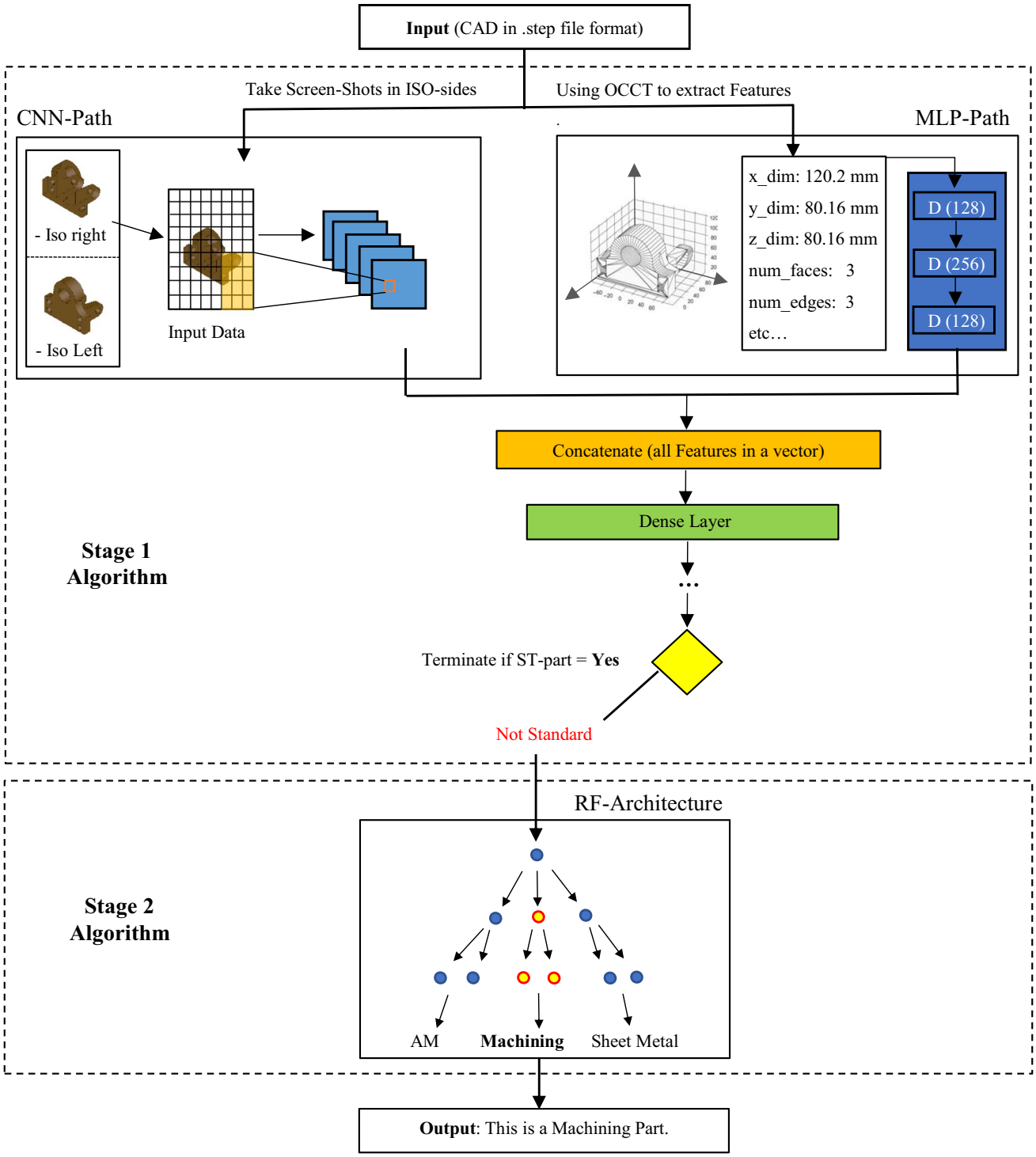


Fig. 2 Classification pipeline illustrated on a real example part (predicted: machining)

expert input, included component size, solid-to-cavity ratio, surface features, geometric complexity, machining accessibility, and material thickness, as shown in Figs. 4, 5, and 6.

4.3 Feature extraction

To make raw STEP data suitable for machine learning applications, significant preprocessing and feature extraction were

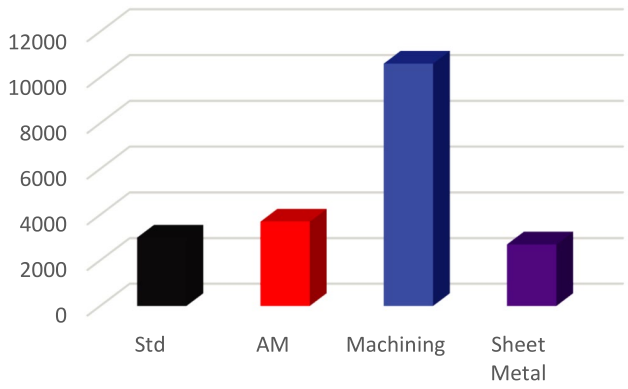


Fig. 3 Data distribution in the dataset

performed to ensure both consistency and informativeness across the dataset. This was achieved using the open-source CAD software kernel Open Cascade Technology (OCCT) [39]. Using the corresponding Python library, we employed the system as an algorithmic feature extractor.

The developed feature extraction algorithm captures 50 quantitative features for each CAD model, including the following:

- Topological features: number of faces and edges, types of edges and surfaces (e.g., planes, cylinders, lines, B-splines).
- Geometrical features: volume, surface area, bounding box dimensions, aspect ratios, center of mass.
- Complexity measures: number of unique face normal vectors, thickness variations, composite complexity scores.

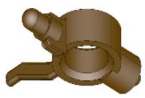
Additive Manufacturing			
Component size: Suitable for a typical AM build space with L-PBF process (approx. 250 x 250 x 300 mm³)		**Solid-to-cavity ratio** [1]: Conducive values	
Surfaces: Complexly shaped surfaces, orientation in different spatial directions		**Geometry**: Bionic, complex, thin-walled geometries, internal structures, undercuts	
			
			

Fig. 4 AM—component definition for classification

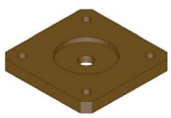
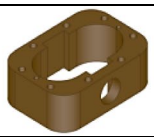
Machining			
Component size: Suitable for machining in mid-sized build space of machining centers (max. 500 x 500 x 300 mm³)		**Solid-to-cavity ratio:** high values	
Surfaces: Flat, orthogonal surfaces, easily accessible machining surfaces, no undercuts		**Geometry:** Block geometries, rotational symmetry, large edge radii, many holes/connections	
			
			

Fig. 5 Machining—component definition for classification

<table>
<tr><th colspan="4">Sheet Metal</th></tr>
<tr><td colspan="2">Component size: Few limitations (3000 x 1500 mm²)</td><td colspan="2">Solid-to-cavity ratio: values vary, can be very low, but high values are also possible</td></tr>
<tr><td colspan="2">Surfaces: Flat surfaces, defined angles between each other</td><td colspan="2">Geometry: Flat geometries, material thickness max. 10mm</td></tr>
<tr><td colspan="2">Accessibility: Accessibility for laser cutting and consideration of bending radius</td><td colspan="2">Welded constructions/Connecting individual elements: Subsequent welding of several components possible</td></tr>
<tr><td>  </td><td>  </td><td>  </td><td>  </td></tr>
<tr><td>  </td><td>  </td><td>  </td><td>  </td></tr>
</table>

Fig. 6 Sheet metal—component definition for classification

- Normalized ratios: relative prevalence of different geometric elements.
- Logarithmic transformations: applied to volume, surface area, face, and edge counts.

Feature selection strategies were tailored to each stage's classification requirements. For stage 1, 31 of the most discriminative features were selected based on the importance of the ranking analysis to distinguish standard from non-standard components. Stage 2 employed a comprehensive set of more than 50 features to capture the increased geometric complexity required for manufacturing process differentiation. The expanded feature set was enabled by the elimination of ST-Parts, allowing the algorithm to leverage more detailed geometric characteristics without standardized component variations. The complete list of selected features is provided in Appendix 1, and the top 10 features were derived via Gini importance and reflect key geometric and topological design factors (Appendix 2).

To quantify feature relevance, we used the Gini-based feature importance from the random forest classifier [27]. The importance of a feature x_i is defined as the total reduction in Gini impurity across all decision nodes that split on x_i:

$$\textbf{Importancce}(x_i) = \sum_{t \in T(x_i)} \Delta G(t) \tag{2}$$

where $T(x_i)$ is the set of tree nodes using x_i and $\Delta G(t)$ is the decrease in Gini impurity caused by the split at node t. A larger $\Delta G(t)$ indicates a better separation of classes, and thus a more relevant feature. This importance ranking was used to select the most relevant features for classification in stage 1.

These extracted features are utilized in both stages of the classification pipeline. Additionally, in the first stage, 2D renderings of CAD models are processed via a CNN to extract visual features. To accomplish this, screenshots are taken from two distinct isometric views of each CAD model (ISO-right and ISO-left), enhancing the model's ability to capture and analyze relevant geometric properties from visual data (Figure 7).

The ISO-right and ISO-left views provide complementary geometric perspectives that, combined with the MLP's numerical features, ensure comprehensive part representation while maintaining computational efficiency [22]. These views are then rasterized into pixel-based images, from which RGB values are extracted and represented in a matrix or vector format. The CAD renderings of the AM, machining, and sheet metal models used in "Sect. 4.3" were consistently generated at a resolution of 600×600 pixels. RGB images were used to enhance the edge and surface detail. They were rescaled as needed for consistency. Compared to grayscale, the brown–black contrast yields higher luminance based on the sRGB formula, increasing local gradients. This can boost early CNN activations and improve feature extraction in structured CAD images [40, 41].

4.4 Data imbalance

To address the effects of class imbalance in the two-stage AutoSplit classification pipeline, we applied a combination

of structural partitioning and statistical resampling, tailored to the respective learning objectives of each stage.

In stage 1, which was designed to differentiate between standard and non-standard components, we implemented a random undersampling strategy. From the complete dataset of 20,000 CAD-classified components, a total of 5000 parts were selected: all 3000 available ST-Parts, combined with a balanced subset of 2000 non-ST-Parts drawn proportionally from the remaining manufacturing categories. This procedure reduced the dominance of the majority class (ST-Parts) without introducing synthetic data, resulting in a more balanced binary classification problem (Fig. 8) [42].

Data augmentation was applied exclusively through geometric transformations (resizing, rescaling, and rotation) of the original CAD objects, while synthetic data (images) generation was deliberately avoided due to engineering design standards and manufacturing constraints. To preserve data integrity and maintain authenticity of engineering design patterns, the analysis was restricted to the available dataset without artificial sample generation. This approach ensures that learned features correspond to genuine manufacturing geometries while introducing controlled variations that reflect realistic manufacturing and assembly scenarios.

In stage 2, the remaining 17,000 non-standard components were assigned to one of three manufacturing process categories: additive manufacturing (class 1, $n = 3700$), machining (class 2, $n = 10{,}600$), and sheet metal (class 3, $n = 2700$), as shown in Fig. 9. Despite the exclusion of ST-Parts, the class distribution remained substantially imbalanced, with machining (class 2) still dominating the dataset.

To address this imbalance, a multi-faceted rebalancing strategy was implemented. First, 30% of machining samples were randomly iteratively downsampled to reduce majority-class dominance. Subsequently, the Synthetic Minority

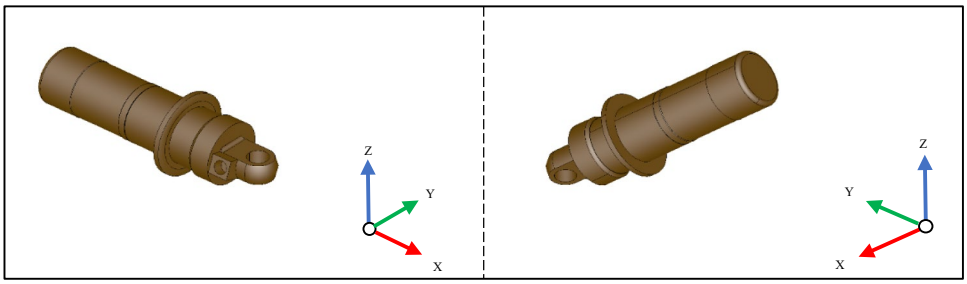


Fig. 7 Two isometric views of CAD-model

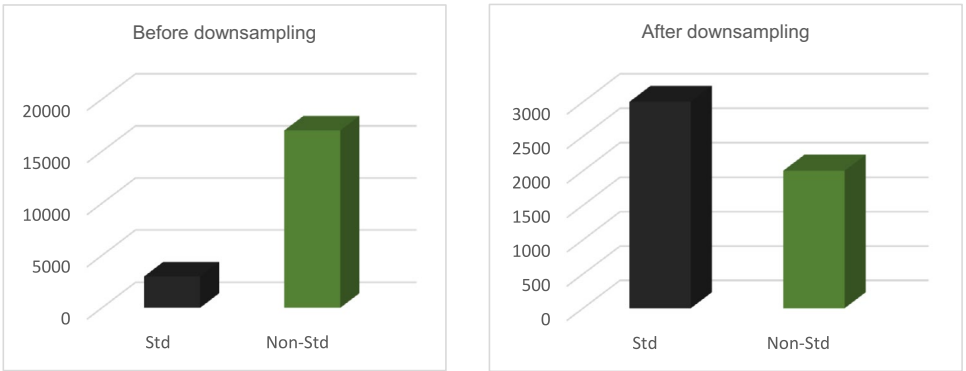


Fig. 8 Stage 1—utilized dataset

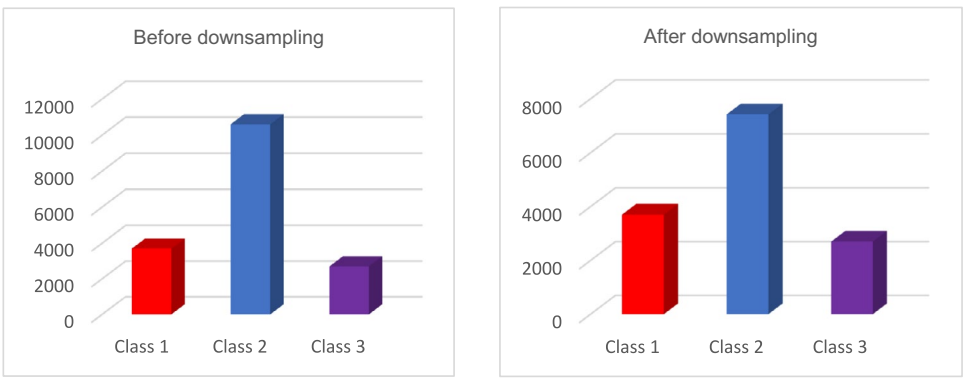


Fig. 9 Stage 2—utilized dataset

Oversampling Technique (SMOTE) was applied to generate synthetic instances for minority classes through linear interpolation in the feature space [43, 44]. Stage 2 used a conservative SMOTE strategy to rebalance the class distribution (AM: 2952, machining: 7420, sheet metal: 1864 → 3517, 7420, 2050), improving minority recall while minimizing overfitting. Detailed results are provided in Appendix 3.

Additionally, class-weighted learning with weights {0: 1.2, 1: 1.0, 2: 1.2} was used to compensate for residual imbalance. These weights were manually tuned to balance recall across underrepresented classes based on iterative experimentation.

This combined approach of downsampling, synthetic oversampling, and weighted learning demonstrated enhanced classification robustness and significantly improved recall performance for underrepresented manufacturing process categories [45, 46].

5 Model design and implementation

In this section, the model architecture will be examined in depth and analyzed in detail, focusing on its core components and technical specifications.

5.1 Stage 1 – Hybrid CNN-MLP architecture for standard part detection

The first stage implements a hybrid architecture that combines a CNN for image processing with a MLP for numerical feature analysis. This design leverages both the visual characteristics of CAD models and their inherent geometric properties (see Fig. 7).

The CNN branch processes two-dimensional projections of CAD models (ISO-right and ISO-left views), utilizing a ResNet50V2 backbone pretrained on ImageNet [47].

The selection of deep residual network architecture stems from its advanced residual learning framework, which addresses the vanishing gradient problem through optimized information flow patterns [48]. The architecture's pre-activation residual units significantly enhance feature extraction capabilities, particularly crucial for identifying complex geometric patterns in manufacturing components [47]. ResNet50V2's batch normalization methodology substantially reduces internal covariate shift, thereby minimizing overfitting tendencies. Furthermore, the architecture's transfer learning capabilities, leveraging pretrained weights from the extensive ImageNet dataset, provide a robust foundation for feature extraction despite the relatively constrained size of manufacturing-specific training data (Fig. 10) [49].

This branch extracts spatial features through a series of convolutional and pooling operations, culminating in a 2048-dimensional feature vector via global average pooling. Subsequent dense layers reduce this representation to 256 dimensions, capturing essential visual characteristics of the components. This dimensionality reduction improves generalization by constraining model complexity and harmonizes the feature scale prior to fusion with the MLP branch. This architectural choice aligns with established practices in transfer learning and multimodal design [50–52].

Simultaneously, the MLP branch processes the 31 numerical features derived from the CAD models,

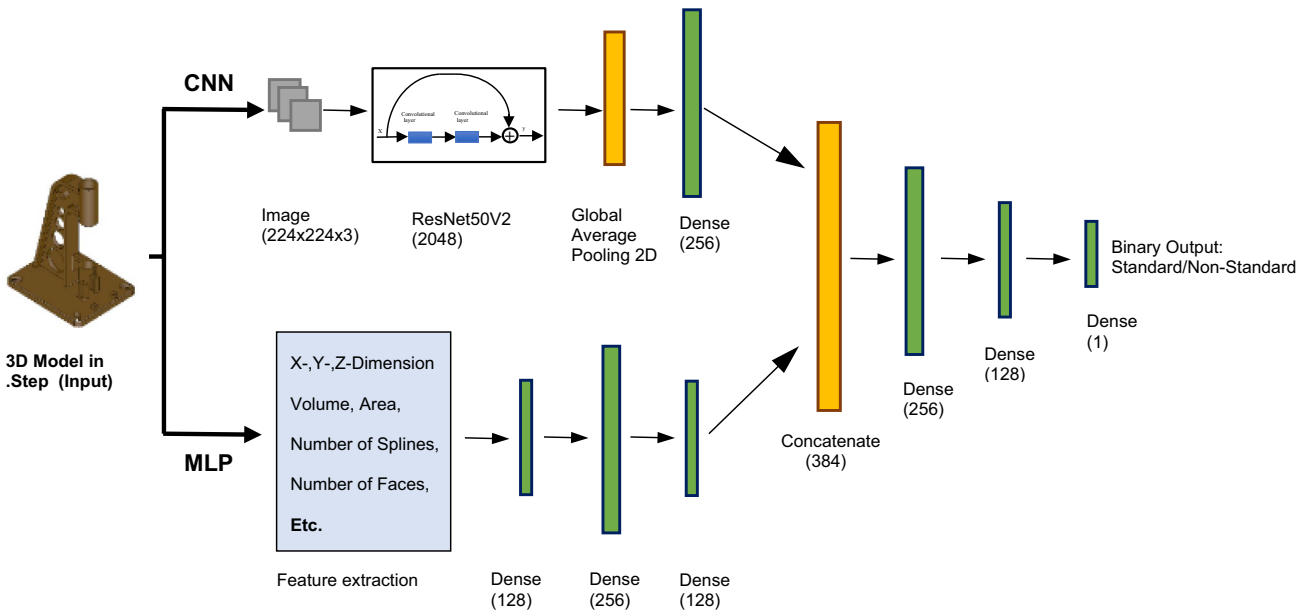


Fig. 10 Stage 1—architecture (hybrid model)

including geometric parameters (volume, surface area) and topological characteristics (edge counts, face types). This branch employs three dense layers with dimensions (128 → 256 → 128), effectively capturing the relationships between various geometric and topological features.

The outputs from both branches undergo concatenation, forming a comprehensive 384-dimensional feature vector that combines visual and numerical characteristics. This concatenated representation passes through additional dense layers before final binary classification via sigmoid activation.

Upon detection of a non-standard component, the system advances the specimen to the secondary analysis phase, where feature extraction and classification are performed utilizing a random forest algorithm.

5.1.1 Bottleneck architecture optimization (stage 1)

During optimizing of a MLP subnetwork of the hybrid model, we evaluated three bottleneck configurations based on the expansion–contraction paradigm, which advocates for an initial increase followed by a reduction in dimensionality to enhance information flow and regularization. The tested configurations are detailed in Table 1.

The bottleneck design provides multiple architectural benefits that contribute to both learning efficiency and generalization performance. By first expanding and then compressing the feature space, it facilitates richer internal representations while simultaneously reducing redundancy and noise, in line with principles from information theory [56]. The symmetric structure of the bottleneck promotes stable gradient flow during backpropagation, which is particularly important for deep networks [57]. Moreover, the enforced dimensional constraint acts as a form of implicit regularization, encouraging the network to focus on salient patterns and preventing overfitting by limiting the parameter search space [24].

The large architecture achieved the highest validation metrics across most categories (e.g., AUC = 0.948, accuracy = 86.8%), as shown in Fig. 11. However, the medium configuration yielded nearly equivalent predictive performance (AUC = 0.941, $F1 = 0.887$), while requiring 25% fewer parameters and reducing training time by approximately 20% (see Table 1). Notably, the validation AUC curve for the medium model displayed higher stability across epochs, indicating better generalization. In contrast, the small configuration exhibited a slightly lower performance (AUC = 0.940, $F1 = 0.873$), and its convergence was more sensitive to learning rate fluctuations.

Based on these findings, the medium configuration [128, 256, 128] was selected as the optimal bottleneck structure. It offers an effective balance between model capacity and

Table 1 Bottleneck architecture optimization

Configuration	Layer dimensions	Parameter count	Theoretical motivation
Small	[64, 128, 64]	~660,000	Model parsimony and minimal overfitting [53]
Medium	[128, 256, 128]	~730,000	Balanced complexity-efficiency trade-off [54]
Large	[256, 512, 256]	~970,000	Maximal expressive power and representational capacity [55]

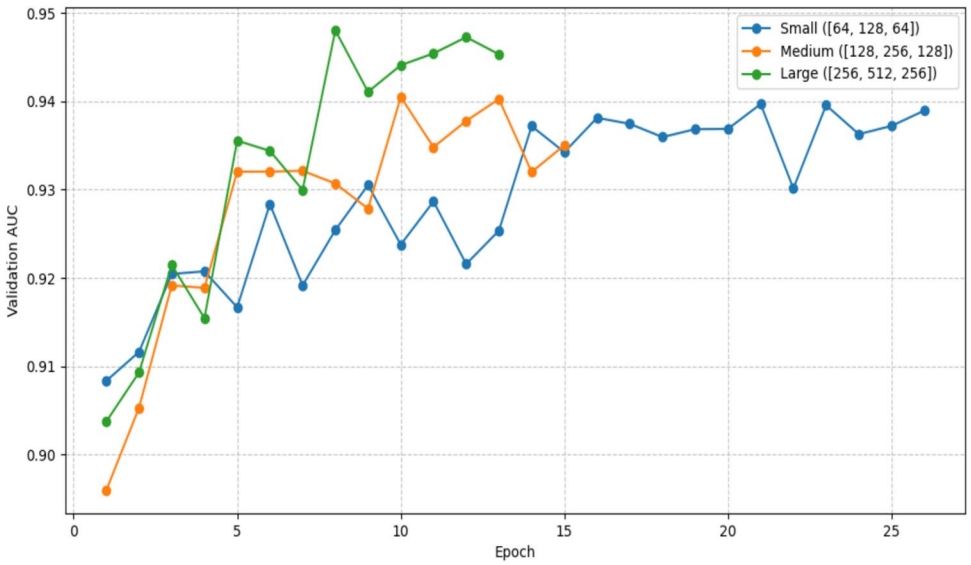


Fig. 11 Validation AUC during training

computational efficiency, supporting stable training dynamics and strong generalization without unnecessary architectural overhead. The medium model demonstrates a significantly lower resource consumption compared to the large configuration while achieving nearly an identical performance (AUC difference of only 0.01). This makes it a cost-effective choice for deployment, requiring substantially less computational resources, memory, and training time. The large configuration will only be considered for upgrade when absolutely necessary, as the minimal performance gain does not justify the increased resource requirements for most practical applications.

5.2 Stage 2 – Random forest classification for manufacturing classification

The second stage implements a random forest-based classification approach to categorize non-ST-Parts into three distinct manufacturing processes: AM, machining, and sheet metal fabrication (see Fig. 12).

Random forests, as introduced by Breiman [27], leverage ensemble learning principles to handle non-linear relationships in high-dimensional feature spaces while providing robust classification performance.

To mitigate potential overfitting challenges identified by Zhang et al. [58], multiple optimization strategies are employed. The model undergoes hyperparameter optimization through RandomizedSearchCV [59], focusing on critical parameters such as maximum tree depth, minimum samples per leaf, and ensemble size. Furthermore, to address the inherent class imbalance in manufacturing data, SMOTE is implemented, generating synthetic examples of minority classes through interpolation between existing instances [45]. The classification architecture culminates in a majority voting mechanism, where individual trees contribute to the final process determination.

This comprehensive approach enables a reliable process classification while maintaining generalization capabilities across diverse component geometries.

5.3 Training methodology

The training process follows a sequential approach, with each stage trained independently, as mentioned previously.

Table 2 and Table 3 summarize the key training configurations.

The hybrid CNN–MLP model was trained using the Adam optimizer (initial learning rate: 1e-3, exponential decay) for 100 epochs with a batch size of 32. For stabilization, batch normalization, dropout, mixed precision, and early stopping applied to validation AUC were employed [51, 60]. The image data were scaled to 600×600 RGB and augmented through data augmentation. The numerical features were normalized using z-transformation (StandardScaler). The dataset was split into training and validation sets with an 80/20 ratio (stratified).

In the second classification step, a random forest ensemble with three variants (different estimator counts and max depth ranges) was employed. Class imbalance was addressed using SMOTE, and hyperparameters were optimized via RandomizedSearchCV [59].

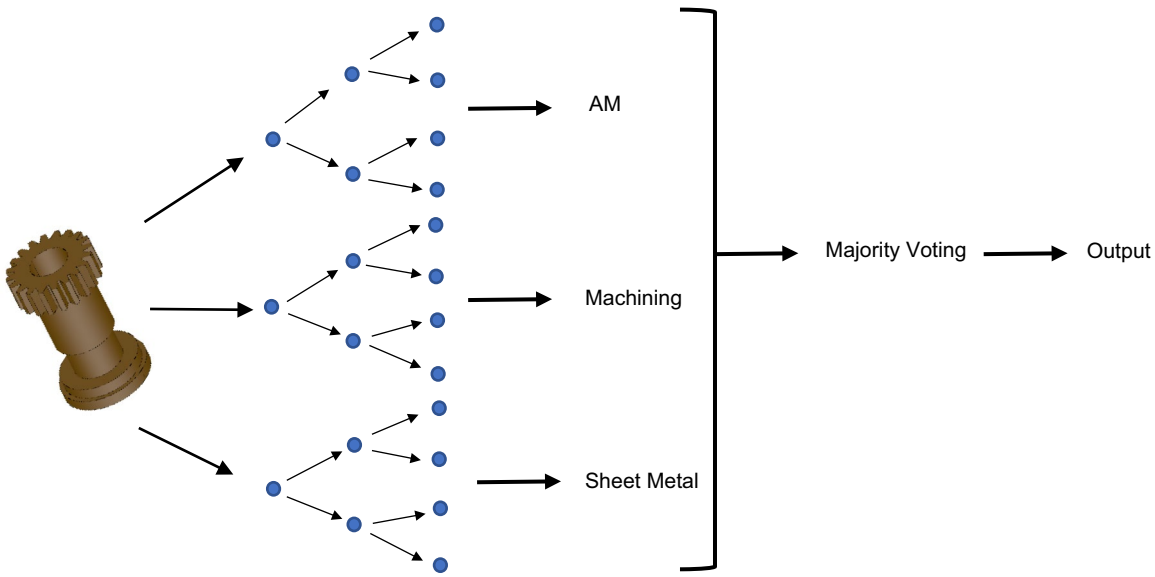


Fig. 12 Illustration of the random forest algorithm for AutoSplit's second stage

Table 2 Training configuration stage 1

Component	Setting
Optimizer (stage 1)	Adam (with ExponentialDecay)
Learning rate (initial)	1e-3
Batch size	32
Epochs	100
Input size (image data)	600×600 RGB
Loss function	Binary cross-entropy
Augmentation	Flip, brightness, contrast, color, rotate
Early stopping	On validation AUC
Normalization (MLP)	*Z*-score (StandardScaler)
Dataset split	80% training/20% validation (stratified)

Table 3 Training configuration stage 1

Component	Setting
Classifier (stage 2)	Random forest ensemble (SMOTE, RandomizedSearchCV)
Estimators range	95–135 trees
Max depth range	11–15
Class balancing	Final weighted classes {0:1.2, 1:1.0, 2:1.2}

5.4 Evaluation methodology

The performance assessment of the AutoSplit framework employs various complementary evaluation metrics, each offering unique insights into the model's classification capabilities. To ensure methodological rigor and fair comparison, all baseline models and architectural variants were trained and evaluated on identical datasets with consistent train-test splits and cross-validation folds. Due to the inherent class imbalance in manufacturing data, metrics beyond simple accuracy are utilized.

The primary evaluation metrics include the following:

$$\mathbf{Accuracy} = (\mathbf{TP} + \mathbf{TN})/(\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}) \tag{3}$$

$$\mathbf{Precision} = \mathbf{TP}/(\mathbf{TP} + \mathbf{FP}) \tag{4}$$

$$\mathbf{Recall} = \mathbf{TP}/(\mathbf{TP} + \mathbf{FN}) \tag{5}$$

$$\boldsymbol{F}1\mathbf{score} = 2 \times (\mathbf{precision} \times \mathbf{recall})/(\mathbf{precision} + \mathbf{recall}) \tag{6}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively [61].

Additionally, confusion matrices are employed for a detailed error analysis, revealing specific misclassification patterns between manufacturing processes. This visualization is particularly valuable for identifying systematic errors in the classification process and understanding the model's behavior across various manufacturing categories.

5.4.1 Model training and optimization

To ensure reliable performance in each stage, both the hybrid model in stage 1 and the random forest classifier in stage 2 were trained using the described dataset. Cross-validation was used to promote generalization to new data, minimizing overfitting and providing robust performance estimates [62]. Hyperparameters were also optimized for both stages to enhance classification accuracy, which is crucial for maintaining high performance and consistency across different CAD model geometries and manufacturing processes.

6 Results and discussion

This section presents a comprehensive analysis of AutoSplit's performance, examining both stages of the classification framework and their effectiveness in manufacturing process determination.

6.1 Performance overview of stage 1

The hybrid neural network demonstrated robust performance in distinguishing between standard and non-standard components. The model achieved a validation accuracy of 88.84% with an AUC score of 0.94, indicating excellent discriminative capability. Notably, the precision-recall trade-off shows balanced performance with a precision of 89.14% and recall of 0.91 (Table 4).

These results confirm the model's ability to generalize across diverse geometric configurations of ST-Parts. The high recall is particularly valuable in early design stages, where false negatives could lead to misclassification of critical components.

Detailed analysis of the confusion matrix reveals:

Class-wise metrics demonstrates a particularly strong performance in a standard part identification in Table 5:

The weighted average *F*1-score of 0.89 indicates a robust overall performance, crucial for the system's first-stage filtering capability.

Table 4 Binary classification prediction

Art of prediction	Predicted standard	Predicted non-standard
Actual standard	**342**	56
Actual non-standard	44	**454**

Table 5 Binary classification performance

Metric	Value
Accuracy	89%
Precision	89%
Recall	91%
F1-score	0.89

Table 6 Manufacturing process classification prediction

Art of prediction	Predicted AM	Predicted machining	Predicted sheet metal
Actual AM	**477**	150	21
Actual machining	144	**1155**	67
Actual sheet metal	17	48	**356**

Table 7 Manufacturing process classification performance

Manufacturing process	Precision	Recall	*F*1-score
AM	75%	73%	0.74
Machining	85%	85%	0.85
Sheet metal	80%	85%	0.82

6.2 Performance overview of stage 2

The subsequent random forest classifier achieved an overall accuracy of 82% in categorizing non-standard components into specific manufacturing processes. The confusion matrix demonstrates the model's classification behavior (Table 6).

Process-specific performance metrics reveal varying effectiveness across manufacturing categories in Table 7. Machining processes show notably higher classification accuracy ($F1 = 0.85$), potentially attributable to more distinctive geometric characteristics. The slightly lower performance in AM ($F1 = 0.75$) and sheet metal ($F1 = 0.82$) classification is due to dataset imbalance, with more samples in the machining category.

6.3 Comparative baseline performance with other architectures

To identify a suitable baseline for later comparison with the proposed AutoSplit framework, we conducted a comprehensive evaluation of both traditional and deep learning–based classification models (Table 8). All models in this comparison were trained on the same dataset, using an identical set of geometric and topological features, and identical training parameters to ensure a consistent and fair evaluation.

Table 8 Accuracy, precision, recall, *F*1-score for each model

Model	Accuracy	Precision	Recall	*F*1-score
Random forest (4-classes)	79.99%	79.94%	79.99%	0.7961
Gradient boosting	79.59%	79.48%	79.59%	0.792
SVM (RBF)	76.66%	77.46%	76.66%	0.7572
Neural network (MLP)	75.70%	75.68%	75.70%	0.7574
SVM (linear)	74.46%	74.74%	74.46%	0.735
Logistic regression	74.13%	74.10%	74.13%	0.7330
K-nearest neighbors	74.07%	73.82%	74.07%	0.7360
Custom CNN + MLP (1-stage)	72.50%	74.00%	70.00%	0.7433
ResNet50V2 + MLP (1-stage)	74.40%	76.13%	70.24%	0.7606
Decision tree	71.36%	71.72%	71.36%	0.715
AdaBoost	70.24%	69.95%	70.24%	0.7010
Naive Bayes	60.54%	59.64%	60.54%	0.5390

Since AutoSplit uses a two-stage pipeline, we first identify the best one-stage model for fair comparison. A detailed class-wise comparison with AutoSplit follows in "Sect. 6.4."

As shown in Fig. 13 and Table 8 the traditional ensemble-based methods—particularly random forest and gradient boosting—outperform other models across all evaluation metrics. Deep learning models using the CNN-based feature extraction, such as ResNet50V2 and the custom CNN-MLP hybrid, achieved only marginal improvements and fell significantly short of the performance of traditional architectures.

Toward the right end of the plot, a notable decline in performance can be observed, particularly for Naïve Bayes and AdaBoost. This underscores the robustness gap between simple classifiers and more expressive ensemble models, making random forest a strong candidate for class-wise benchmarking in subsequent comparisons.

A more detailed analysis of cross-validation results for the second stage, including performance variation across multiple random seeds, is provided in Appendix 4.

6.3.1 Comparative analysis AutoSplit with random forest algorithm

To validate the AutoSplit two-stage architecture, we compared it with the best-performing single-stage random forest classifier. This baseline directly classifies components into four categories: additive manufacturing (AM), machining, sheet metal, and ST-Parts using identical hyperparameters from AutoSplit's stage 2 for fair comparison. This alignment minimizes variability caused by model tuning and isolates architectural differences as the primary factor influencing performance outcomes.

The comparison reveals class-wise results as shown in Tables 9, 10, and 11:

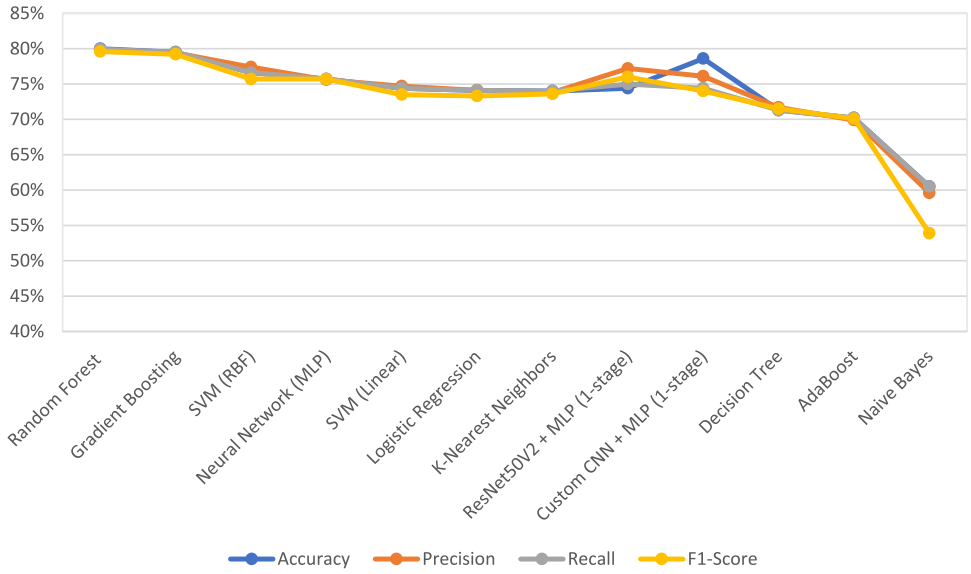


Fig. 13 Empirical comparison phase

Table 9 Classification performance 4-classes architecture

Manufacturing process	Precision	Recall	*F*1-score	Stages
Standard parts	85%	73%	0.78	One stage
AM	75%	60%	0.67	
Machining	80%	90%	0.84	
Sheet metal	82%	75%	0.79	

Table 10 Classification performance for the two-stage architecture

Manufacturing process	Precision	Recall	*F*1-score	Stages
Standard parts	89%	91%	0.87	Stage 1
AM	75%	73%	0.74	Stage 2
Machining	85%	85%	0.85	
Sheet metal	80%	85%	0.82	

Table 11 Performance difference (two-stage vs. single-stage)

Manufacturing process	Precision	Recall	*F*1-score
Standard parts	+4%	+18%	+9%
AM	0%	+13%	+7%
Machining	+5%	−5%	+1%
Sheet metal	−2%	+10%	+3%

For the sake of visual clarity, only mean values from cross-validation are reported. While this facilitates a more straightforward comparison of a model performance, it comes at the cost of a reduced statistical interpretability, as measures of variance—such as standard deviation or confidence intervals—are omitted. This omission limits the expressiveness of the results by concealing potential variability across folds [63].

6.3.2 Overall performance improvement

The two-stage AutoSplit approach demonstrates substantial performance improvements over the single-stage model. These gains are particularly evident in the recall and *F*1-score metrics, which are critical in our application context as they reflect the model's ability to correctly detect relevant class instances.

In the context of manufacturing classification, recall is particularly important, as it indicates the model's ability to correctly identify all relevant components within each manufacturing category. This is a key factor in avoiding costly misclassifications, such as assigning a component to the wrong manufacturing process, which could lead to inefficiencies, waste, or even product failure:

Noteworthy improvements include the following:

- Standard part recall (+18%)
- Standard part identification (+9%)
- AM component recall (+13%)
- AM component identification (+7%)
- Sheet metal component recall (+10%)

These results indicate that the two-stage approach not only increases overall accuracy but also provides a more balanced and reliable performance across diverse manufacturing categories. Although, there is a slight decrease in machining recall (−5%) and sheet metal precision (−2%); these drawbacks are outweighed by improvements in the overall F1-score and recall, especially in classes where classification is more challenging.

To validate robustness and variation, five independent experiments with different random seeds were conducted. Detailed results are provided in Appendix 5.

The experimental results validate that a two-stage classification approach leads to a more accurate and robust manufacturing process determination.

6.4 Computational efficiency and inference speed

In addition to classification accuracy and robustness, inference time is a critical factor for real-world integration of automated decision-support tools.

Table 12 summarizes the total runtime and average inference time per sample for four representative models: the proposed AutoSplit two-stage architecture and three one-stage baselines (random forest, custom CNN + MLP, and ResNet50V2 + MLP). All measurements were conducted on the same dual-GPU system with a batch inference enabled and TensorFlow mixed-precision optimization.

While traditional models such as random forest offer extremely fast inference times, they fall short in classification precision, particularly when distinguishing between overlapping geometric features across manufacturing processes. AutoSplit maintains a competitive runtime—outperforming both CNN-based baselines—and delivers a significantly higher accuracy by incorporating an initial standard-part filter and process-specific classification refinement. This balance between computational efficiency and predictive performance confirms the practical applicability of AutoSplit in industrial design environments, where real-time or near-real-time feedback is desirable.

6.5 Error analysis

Examination of misclassification patterns reveals several significant challenges within the classification process. The analysis shows that the most frequent confusion occurs predominantly between AM and machining categories, likely due to their overlapping geometric characteristics and similar feature sets. Sheet metal components demonstrate relatively low misclassification rates, though when errors do occur, they are primarily confused with machining processes, possibly due to shared geometric features such as flat surfaces and linear edges. Additionally, the asymmetric nature of these misclassifications suggests distinct feature overlaps between different manufacturing process pairs, indicating that certain manufacturing methods share more common characteristics than others, which can complicate the classification process. These patterns highlight the inherent complexity of distinguishing between different manufacturing processes, particularly when components exhibit features that could reasonably be produced through multiple manufacturing methods.

6.6 Discussion

While AutoSplit demonstrates a strong classification performance—achieving 88.84% accuracy (AUC: 0.94) in standard part detection and 82% accuracy in process-specific classification—the observed confusion between machining and additive manufacturing highlights the practical challenge of overlapping manufacturing feasibility. For example, parts with moderate complexity and tolerancing can often be produced by either method, depending on specific requirements such as surface finish or production volume. These borderline cases reflect not only modeling limitations but also real-world ambiguity in design-to-manufacturing transitions.

Furthermore, it is important to emphasize that AutoSplit functions as a recommendation system, not a prescriptive decision engine. Its output is intended to support engineers in narrowing down viable manufacturing routes during the early design stage, rather than replacing expert judgment or cost-based trade-off analysis.

Lastly, while machining achieves the highest $F1$-score (0.85), followed by sheet metal (0.82) and additive manufacturing (0.74), process evaluation should not be based solely on accuracy or $F1$. In safety–critical applications or cost-sensitive domains, recall may be of higher importance to avoid overlooking feasible manufacturing strategies. Future versions of AutoSplit could incorporate configurable risk preferences, allowing users to adjust the model's sensitivity to false negatives or false positives depending on context.

Table 12 Summarize the total runtime

Model	Architecture type	Avg. inference time/sample	Total runtime (full set)
Random forest (1-stage)	Traditional ML	~0.04 s	~40 s
Custom CNN + MLP (1-stage)	Deep learning hybrid	~0.78 s	~595 s
ResNet50V2 + MLP (1-stage)	Deep learning hybrid	~0.71 s	~542 s
AutoSplit (2-stage)	CNN + MLP + RF (for both stages)	~0.61 s	~482 s

6.6.1 Classification challenges and solutions

A significant challenge in manufacturing process classification stems from inherent overlaps between different manufacturing methods (Figure 14). Despite establishing clear classification criteria, expert disagreement in the labeling process reveals the ambiguous nature of process selection for certain components.

This ambiguity is particularly pronounced between additive manufacturing and machining processes, as evidenced by the confusion matrix:

- 150 AM components misclassified as machining
- 144 machining components misclassified as AM.

To manage inherent classification ambiguities, AutoSplit generates probabilistic outputs rather than definitive categorical predictions. This method provides a probability distribution across all potential manufacturing processes, supporting more nuanced decision-making.

For instance, a component may be classified with a 75% probability for additive manufacturing, 20% for machining, and 5% for sheet metal. This probabilistic approach aligns more closely with real-world manufacturing decision-making, where multiple feasible methods may be applicable to a single component.

6.6.2 Comparative advantages

The two-stage architecture demonstrates significant improvements over single-stage approaches:

- 15.6% improvement in standard part classification
- 18.0% enhancement in AM component recall
- 4.8% improvement in machining identification
- 24.6% increase in sheet metal component recall
- 7.4% overall accuracy improvement.

6.6.3 Practical relevance and industrial applicability

The practical significance of the results is particularly evident in the substantial time savings during development through automated pre-classification and rapid identification of ST-Parts. The early detection of AM-suitable components enables an optimized part design and avoids costly wrong decisions in manufacturing the process selection. The high classification accuracy confirms the reliability of the systematic decision support, while the easy integration into existing CAD and PLM systems through the STEP format enables real-time recommendations during design and underscores the practical value for industrial product development.

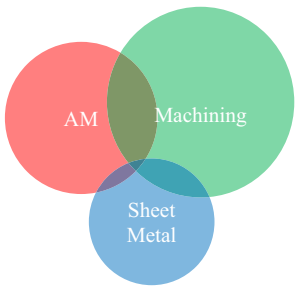


Fig. 14 Overlapping class definitions

6.6.4 Limitations

However, it is important to note some limitations. The current model may struggle with geometries or novel manufacturing techniques not well-represented in the training data. Additionally, the classification does not yet account for manufacturing constraints such as cost-effectiveness or production time, which are crucial factors in industrial applications.

A significant limitation is the reliance on publicly available CAD libraries (GrabCAD, Fusion360 gallery, TraceParts), which may not fully represent industrial manufacturing complexity found in proprietary datasets. Furthermore, the statistical significance of reported performance improvements requires a formal validation through hypothesis testing to ensure observed gains are not due to random variation.

7 Conclusion

AutoSplit demonstrates significant improvements in manufacturing process classification through its innovative two-stage approach. The hybrid CNN-MLP model achieves 88.84% $\pm$0.6 (SD) accuracy (AUC : 0.94) in a standard part identification, while the random forest classifier successfully categorizes non-ST-Parts into specific manufacturing processes with 82.0% $\pm$1.1 (SD) accuracy. These results represent clear advancements over traditional one-stage approaches, particularly with regard to process-level recall and $F1$-score.

Key contributions include the development of an effective two-stage classification framework and the creation of a comprehensive 20,000 CAD file dataset. Future work should focus on integrating cost-time constraints, developing advanced feature extraction methods, and expanding the dataset to include emerging manufacturing technologies and multi-process components. Integration with the CAD software for real-time process recommendations presents another promising direction for development.

8 Appendix 1. The utilized features for the first stage:

Table 13 Utilized features stage 1

Feature	Description
num_faces	Total number of faces in the model
num_edges	Total number of edges in the model
Volume	Volume of the component
surface_area	Total surface area of the component
x_dim	Dimension in *x*-direction (width)
y_dim	Dimension in *y*-direction (height)
z_dim	Dimension in *z*-direction (depth)
num_line_edges	Number of straight-line edges
num_circle_edges	Number of circular edges
num_ellipse_edges	Number of elliptical edges
num_bsplinecurve_edges_ratio	Proportion of B-spline curve edges to total edges
thickness_complexity	Complexity measure based on wall thickness to faces ratio
num_torus_surfaces_ratio	Proportion of torus surfaces to total surfaces
num_bsplinecurve_edges	Number of B-spline curve edges
num_line_edges_ratio	Proportion of straight edges to total edges
num_plane_surfaces	Number of planar surfaces
num_cylinder_surfaces	Number of cylindrical surfaces
log_num_edges	Logarithmic transformation of edge count
average_edge_length	Average length of all edges
aspect_ratio_xy	Aspect ratio in *xy*-plane (width/height)
aspect_ratio_xz	Aspect ratio in *xz*-plane (width/depth)
aspect_ratio_yz	Aspect ratio in *yz*-plane (height/depth)
volume_to_xyz	Ratio of actual volume to bounding box volume ($x \times y \times z$)
num_bspline_surfaces_ratio	Proportion of B-spline surfaces to total surfaces
num_unique_normals	Count of unique normal vectors (diversity of face orientations)
wall_thickness	Average wall thickness of the component
x_center_mass_relative	Relative position of center of mass in *x*-direction
y_center_mass_relative	Relative position of center of mass in *y*-direction
z_center_mass_relative	Relative position of center of mass in *z*-direction
shape_factor	Form factor (surface area/volume$^{(2/3)}$)
overall_complexity	Average complexity score from edge, surface, and thickness complexity

And the utilized features for the second stage:

Table 14 Utilized features stage 2

Feature	Description
num_faces	Total number of faces in the model
num_edges	Total number of edges in the model
Volume	Volume of the component
surface_area	Total surface area of the component
x_dim	Dimension in *x*-direction (width)
y_dim	Dimension in *y*-direction (height)
z_dim	Dimension in *z*-direction (depth)
num_line_edges	Number of straight-line edges
num_circle_edges	Number of circular edges
num_ellipse_edges	Number of elliptical edges
num_hyperbola_edges	Number of hyperbolic edges
num_parabola_edges	Number of parabolic edges
num_beziercurve_edges	Number of Bezier curve edges
num_bsplinecurve_edges	Number of B-spline curve edges
num_othercurve_edges	Number of other curve types
num_plane_surfaces	Number of planar surfaces
num_cylinder_surfaces	Number of cylindrical surfaces
num_cone_surfaces	Number of conical surfaces
num_sphere_surfaces	Number of spherical surfaces
num_torus_surfaces	Number of torus (donut) surfaces
num_bezier_surfaces	Number of Bezier surfaces
num_bspline_surfaces	Number of B-spline surfaces
num_revolution_surfaces	Number of revolution surfaces
num_extrusion_surfaces	Number of extrusion surfaces
num_other_surfaces	Number of other surface types
num_unique_normals	Count of unique normal vectors (diversity of face orientations)
wall_thickness	Average wall thickness of the component
x_center_mass_relative	Relative position of center of mass in *x*-direction
y_center_mass_relative	Relative position of center of mass in *y*-direction
z_center_mass_relative	Relative position of center of mass in *z*-direction
volume_to_surface_ratio	Ratio of volume to surface area (measure of compactness)
edge_to_face_ratio	Ratio of edges to faces (measure of geometric complexity)
average_face_area	Average area of individual faces
edge_complexity	Complexity measure based on edge-to-face ratio
surface_complexity	Complexity measure based on unique normals-to-faces ratio
thickness_complexity	Complexity measure based on wall thickness to faces ratio
average_edge_length	Average length of all edges
num_line_edges_ratio	Proportion of straight edges to total edges
num_circle_edges_ratio	Proportion of circular edges to total edges
num_ellipse_edges_ratio	Proportion of elliptical edges to total edges
num_bsplinecurve_edges_ratio	Proportion of B-spline curve edges to total edges
num_plane_surfaces_ratio	Proportion of planar surfaces to total surfaces
num_cylinder_surfaces_ratio	Proportion of cylindrical surfaces to total surfaces
num_cone_surfaces_ratio	Proportion of conical surfaces to total surfaces
num_sphere_surfaces_ratio	Proportion of spherical surfaces to total surfaces
num_torus_surfaces_ratio	Proportion of torus surfaces to total surfaces
num_bspline_surfaces_ratio	Proportion of B-spline surfaces to total surfaces
aspect_ratio_xy	Aspect ratio in *xy*-plane (width/height)

Table 14 (continued)

Feature	Description
aspect_ratio_xz	Aspect ratio in *xz*-plane (width/depth)
aspect_ratio_yz	Aspect ratio in *yz*-plane (height/depth)
volume_to_xyz	Ratio of actual volume to bounding box volume ($x \times y \times z$)
surface_to_xyz	Ratio of surface area to bounding box volume
center_mass_distance	Distance of center of mass from geometric center
log_volume	Logarithmic transformation of volume
log_surface_area	Logarithmic transformation of surface area
log_num_faces	Logarithmic transformation of face count
log_num_edges	Logarithmic transformation of edge count
shape_factor	Form factor (surface area/volume$^{(2/3)}$)
overall_complexity	Average complexity score from edge, surface, and thickness complexity

9 Appendix 2. Top 6 most important features.

The table below lists the ten most influential features based on cumulative Gini importance, as used by the random forest classifier in stage 2. These features collectively account for 44.1% of total decision weight.

Table 15 Top 10 feature importance

Rank	Feature name	Importance (%)
1	shape_factor	0.085678
2	thickness_complexity	0.080241
3	wall_thickness	0.053853
4	num_bsplinecurve_edges	0.033427
5	volume_to_xyz	0.031898
6	num_bsplinecurve_edges_ratio	0.029119
7	num_bspline_surfaces_ratio	0.028144
8	aspect_ratio_xy	0.023305
9	overall_complexity	0.023135
10	num_faces	0.022878

With feature category breakdown as follows:

- Geometric: three features (avg. importance: 3.39%)
- Edge/surface complexity: two features (avg. importance: 4.70%)
- Efficiency metrics: one feature (4.09%)
- Interaction metric: one feature (8.26%)
- Other shape descriptors: three features (avg. importance: 4.07%)

10 Appendix 3. Micro-tuned SMOTE optimization summary.

Table 16 shows the results of fivefold cross-validation after applying SMOTE in Stage 2. The model achieved an average test accuracy of $81.43\% \pm 0.75\%$**,** with a consistent overfitting gap of $4.91\% \pm 0.89\%$**.** These results confirm that SMOTE effectively improved class balance while maintaining strong generalization across folds.

Table 16 5-Fold validation SMOTE optimization

Fold	Train acc	Test acc	Overfit gap
1	0.8524	0.8252	0.0372
2	0.8516	0.8213	0.0403
3	0.8552	0.8070	0.0582
4	0.8519	0.8109	0.0510
5	0.8561	0.8073	0.0588

11 Appendix 4. Cross-validation accuracy and variance across classifiers.

Cross-validation analysis (Fig. 15) revealed that while random forest attained the highest mean accuracy (0.7903), it exhibited moderate variability (CV Std: 0.0282). Neural network (MLP) demonstrated the lowest variance (0.0058), followed by *K*-nearest neighbors (0.0096) and gradient boosting (0.0190). Gradient boosting achieved nearly comparable performance (CV Mean: 0.7896) with superior stability.

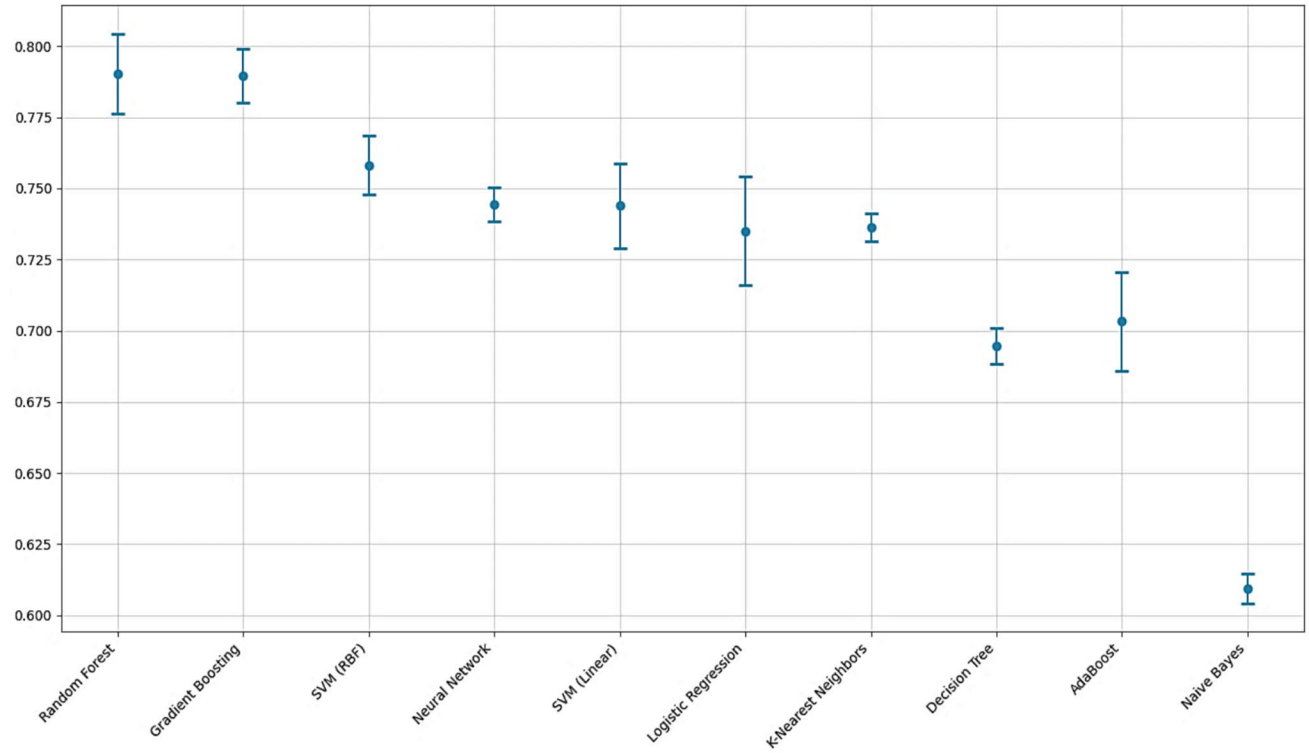


Fig. 15 Cross-validation accuracy with standard deviation across classifiers

Table 17 Cross-validation mean and standard deviation of various ML models for 4-class manufacturing process classification

Model	CV Mean	CV Std
Random forest (4-classes)	0.7903	0.0282
Gradient boosting	0.7896	0.0190
SVM (RBF)	0.7582	0.0207
Neural network (MLP)	0.7444	0.0058
SVM (linear)	0.7439	0.0297
Logistic regression	0.7351	0.0383
K-nearest neighbors	0.7363	0.0096
Decision tree	0.6946	0.0124
AdaBoost	0.7032	0.0347
Naive Bayes	0.6094	0.0104

Logistic regression and SVM (linear) displayed notably higher variability, as evidenced by their larger error bars, making them less reliable for consistent performance across different data splits. Naive Bayes, while showing the poorest accuracy, actually demonstrated relatively low variability, indicating consistent but suboptimal performance.

Random forest's combination of highest mean performance and low-moderate variability, along with its ensemble-based learning mechanism, ability to handle nonlinear interactions, and insensitivity to feature scaling, makes it particularly well-suited for CAD-derived, imbalanced data. However, it should be noted that gradient boosting achieved nearly comparable performance with similar stability. Given random forest's performance advantage and proven robustness in similar applications, it was selected as the final classifier for stage 2.

12 Appendix 5. Micro-tuned model robustness and statistical validation

Performance gains from SMOTE and training stability across folds are detailed in Appendix 3, while seed-based robustness and significance testing are reported in Appendix 5.

To assess robustness and significance, the model was retrained over five independent runs with varied seeds (42, 123, 456, 789, 999). A conservative SMOTE strategy was applied in all folds. Results are showed the following:

- Mean accuracy: 0.8085 ± 0.0068
- Mean $F1$ (macro): 0.7918 ± 0.0072
- Mean recall (macro): 0.7855 ± 0.0065
- Overfitting gap**:** 0.0491 ± 0.0089

Table 18 Mean ± SD and 95% CI of metrics for stage 2 over five random seeds

Metric	Mean ± Std	95% CI
Accuracy	0.8085 ± 0.0068	[0.8000, 0.8170]
$F1$-score (macro)	0.7918 ± 0.0072	[0.7829, 0.8008]
Precision (macro)	0.7991 ± 0.0086	[0.7885, 0.8098]
Recall (macro)	0.7855 ± 0.0065	[0.7774, 0.7937]

A one-sample t-test confirmed that improvements in the $F1$-score were statistically significant ($p = 0.0002$), with a large effect size (Cohen's $d = 5.79$). Performance across seeds was highly consistent, with a maximum $F1$ range of only 0.0182.

The final micro-tuned model achieved 82% test accuracy and 81.6% macro $F1$, outperforming baseline by + 1.3%.

Acknowledgements The authors express their sincere thanks to the Federal Ministry of Economics and Climate Protection and the Project Management Jülich for providing funding and support for the associated project.

Author contributions Mehdi Nazarian: conceptualization, methodology, software implementation, data labeling, writing—original draft; Rafael Neves: software implementation, data mining and analysis, data labeling, testing and validation; Léon Klick: data analysis, resources, review and editing; Fabian Schöfer: initial concepts for the project, data labeling; Robert Lau: supervision, data labeling, formal analysis, review and editing; Arthur Seibel: Project proposal, initial concepts for the project; Felix Weigand: review and editing.

Funding Open Access funding enabled and organized by Projekt DEAL. The collaboration between Fraunhofer IAPT and Autoflug GmbH was purely academic in nature. For this, research was obtained from publicly available sources, including GrabCAD, Fusion360 Gallery, and TraceParts, following their respective terms of use. TraceParts explicitly granted permission for academic use of their CAD models, and the Open CASCADE Technology (OCCT) library was used under its appropriate licensing terms.

Data availability To ensure reproducibility and transparency, the research code and dataset preparation scripts are publicly available at [https://github.com/MNazarian/AutoSplit.git].

Declarations

Competing interests The authors declare no competing interests.

References

1. Zhu Z, Dhokia V, Nassehi A, Newman S (2013) A review of hybrid manufacturing processes – state of the art and future. Int J Comput Integr Manuf 26:596–615
2. Lauwers B, Klocke F, Kink A, Tekkaya A, Neugebauer R, Mcintosh D (2014) Hybrid processes in manufacturing. CIRP Int Acad Prod Eng 63:561–583
3. Dilberoglu UM, Gharehpapagh B, Yaman U, Dolen M (2021) Current trends and research opportunities in hybrid additive manufacturing. Int J Adv Manuf Technol 113:623–648
4. Huang W, Li S (1998) A two-stage hybrid flowshop with uniform machines and setup times. Math Comput Model 27:27–45
5. Wuest T, Weimer D, Irgens C, Thoben K-D, Irgens C (2016) Machine learning in manufacturing: advantages, challenges, and applications. Prod Manuf Res: Open Access J 4:23–25
6. Manda B, Bhaskare P, Muthuganapathy R (2021) A convolutional neural network approach to the classification of engineering models. IEEE Access 9:22711–22723
7. Song Z, Luo S (2024) Application of machine learning and data mining in manufacturing industry. Int J Comput Sci Inform Technol 2:425–436
8. Shi Y, Yicha Z, Xia K, Harik R (2020) A critical review of feature recognition techniques. Comput-Aided Design Appl 17:861–899
9. Blessing L, Chakrabarti A (2009) DRM, a design research methodology. Springer, London, pp 22711–22723
10. Wu H, Lei R, Peng Y, Gao L (2024) AAGNet: A graph neural network towards multi-task machining feature recognition. Robot Comput-Integr Manuf 86:102661
11. Zhang, Zhibo R, Rahul J, Prakhar (2018) FeatureNet: Machining feature recognition based o 3D convolution neural network. Comput-Aided Des 101:12–22
12. Colligan AR, Robinson T, Trevor CND, Hua Y, Cao W (2022) Hierarchical CADNet: learning from B-reps for machining feature recognition. Comput-Aided Des 147(103226):2022
13. ElMaraghy W, ElMaraghy H, Tomiyama T, Monostori L (2012) Complexity in engineering design and manufacturing. CIRP Int Acad Prod Eng 61:793–814
14. Yiu lp C, Lapadat D, Sieger L, Regli WC (2002) Using shape distributions to compare solid models. ACM Symposium on Solid modeling and applications (SMA '02), ACM, Saarbruecken, Germany, pp 273–280
15. Johan H, Li B (2011) 3D model retrieval using hybrid features and class information. Multimed Tools Appl 62:821–846
16. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
17. Xu Y, Mo T, Feng Q, Zhong P, Lai M, I-Chao Chang E (2014). Deep learning of feature representation with multiple instance learning for medical image analysis. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Florance, Italy, pp 1626–1630
18. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. IEEE 86:2278–2324
19. Hinton GE, Sutskever I, Krizhevsky A (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 25:84–90
20. Aditya B, Sambit G, Kin Gwn L, Gavin Y, Adarsh K, Soumik S (2017) Learning localized geometric features using 3D-CNN: an application to manufacturability analysis of drilled holes. arXiv:1612.02141v2
21. Liang M, Hu X (2015) Recurrent convolutional neural network for object recognition. IEEE Conference on computer vision and pattern recognition (CVPR), Boston, MA, pp 3367–3375
22. Su H, Maji S, Kalogerakis E, Learned-Miller E (2015) Multi-view convolutional neural networks for 3D shape recognition. Proceeding

of the IEEE International Conference on Computer Vision (ICCV), Santiago, pp 945–953
23. Hornik K, Maxwell S, Halbert W (1989) Multilayer feedforward networks are universal approximators. Neural Netw (ELSEVIER Access) 2:359–366
24. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. Vol. 1. No 2. Cambridge, Massachusetts: the MIT press, Cambridge, MA
25. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2:303–314
26. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536
27. Breiman L (2001) Random forests. Mach Learn 45:5–32
28. Tombari F, Salti F, Di Stefano L (2010) Unique signatures of histograms for local surface description. In: Daniilidis K, Maragos P, Paragios N (eds) Computer Vision - ECCV 2010. Lect Notes Comput Sci 6313:356–369
29. Zhang H, Wang C, Tian S, Lu B, Zhang L, Ning X, Bai X (2023) Deep learning-based 3D point cloud classification: a systematic survey and outlook. arXiv:2311.02608
30. Yiu Ip C, Regli WC (2005) Content-based classification of CAD models with supervised learning. Computer-Aided Design and Applications 2(5):597–606
31. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (2021) Understanding deep learning (still) requires rethinking generalization. Commun ACM 64:107–115
32. Qi CR, Su H, Mo K, Guibas JL (2017) PointNet: deep learning on point sets for 3D classification and segmentation. IEEE conference on computer vision and pattern recognition (CVPR). pp 652–660
33. Jayaraman PK, Sanghi A, Lambourne JG, Willis KDD, Davies T, Shayani H (2022) UV-Net: Learning from boundary representations. CVPR 2021:26
34. Lambourne JG, Willis KDD, Jayaraman PK, Sanghi A, Meltzer P, Shayani H (2021) BRepNet: A topological message passing system for solid models. Proceeding of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp 12773–12782
35. Hennessy JL, Patterson DA (2017) Computer architecture: a quantitative approach. California, USA: 6th ed. Morgan Kaufmann, Cambridge, MA
36. Autodesk AI Lab, "Fusion360 Gallery Dataset", Github [Online]. Available: https://github.com/AutodeskAILab/Fusion360GalleryDataset. Accessed Nov-Apr 2023–2024
37. Grabcad-Liberary [Online]. Available: https://grabcad.com/library . Accessed Nov-Apr 2023–2024
38. Traceparts [Online]. Available: https://www.traceparts.com/de/. Accessed 04 Nov 2024
39. Open CASCADE Technology (OCCT3D). Open-source full-scale 3D geometry library, [Online]. Available: https://dev.opencascade.org/. Accessed 04 Nov 2024
40. Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W (2019) ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: Proceeding of the International Conference on Learning Representations (ICLR 2019)
41. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks - ECCV 2014 – European Conference on Computer Vision 8689:818–833
42. Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor Newsl 6:20–29
43. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
44. Tomek I (1976) Two modifications of CNN. IEEE Transactions on systems, man, and cybernetics (SMC) 6(11):769–772
45. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21:1263–1284
46. Fernández A, García S, Galar M, Prati RC, Krawczyk B, Herrera F (2018) Learning from imbalanced data sets. Springer International Publishing
47. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp 770–778
48. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. Computer Vision (ECCV 2016), 9908:630–645
49. Kornblith S, Shlens J, Le QV (2019) Do better imagenet models transfer better?. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp 2661–2671
50. Simonyan K, Zisserman A (2015) Very Deep Convolutional Networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR 2015), San Diego, pp 1–14, arXiv:1409.1556
51. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
52. Baltrušaitis T, Ahuja C, Morency L-P (2017) Multimodal machine learning: A survey and taxonomy. IEEE Trans Pattern Anal Mach Intell 41:423–443
53. Rasmussen CE, Ghahramani Z (2000) Occam's razor. In: Leen T, Dietterich T, Tresp V (eds) Advances in neural information processing systems 13 (NIPS 2000), MIT Press, Cambridge, MA, pp 294–300
54. Bengio Y (2012) Practical recommendations for gradient-based training of deep architectures. In: Montavon G, Orr GB, Mueller KR (eds) Neural networks: tricks of the trade, lecture notes in computer science, 7700:437–478
55. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
56. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423
57. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. J Mach Learn Res - Proceedings Track 9:249–256
58. Zhang C, Almpanidis G, Zhang X, Liu C (2017) An up-to-date comparison of state-of-the-art classification algorithms. Expert Syst Appl 82:128–150
59. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13:281–305
60. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. ICML 2015: Proceedings of the 32nd International Conference on Machine Learning, 37:448–456
61. Powers D (2020) Evaluation: from precision, recall and F-Measure to ROC, informedness, markedness & correlation. Mach Learn Technol 2:37–63
62. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th international joint conference on artificial intelligence (IJCAI), Montreal, Quebec, Canada, pp 1137–1143
63. Drummond C, Holte RC (2006) Cost curves: An improved method for visualizing. Mach Learn 65(1):95–130