Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Fidelity Metric with 3D Shape Descriptor for Digital Twin Standardization

MD TARIQUE HASAN KHAN

f18214@wayne.edu

Tarleton State University

Tahir Abbas Jauhar

Soonhung Han

Chiho Noh

Research Article

Keywords: Digital Twin, Fidelity Assessment, Shape Descriptor, Multi-View CNN, 3D Shape Descriptor, Cosine Similarity

Posted Date: June 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9785119/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on October 24th, 2026. See the published version at https://doi.org/10.1007/s00170-026-19307-8.

Fidelity Metric with 3D Shape Descriptor for Digital Twin Standardization

*Md Tarique Hasan Khan Ph.D.; Assistant Professor; Tarleton State University, Stephenville, TX
mkhan@tarleton.edu

Tahir Abbas Jauhar; Centre for Bulk Solids and Particulate Technologies, University of Newcastle, NSW, Australia

Soonhung Han Ph.D.; Professor Emeritus of KAIST, Head of Korea STEP Center (KSTEP), South Korea

Chiho Noh; Researcher at KSTEP; South Korea

Abstract

This study presents a comprehensive framework for assessing Digital Twin (DTw) fidelity through multi-modal deep learning, evaluating the correspondence between physical manufactured components and their digital CAD specifications. Six diverse vision architectures CLIP-ViT-B/32, ViT-B/32, ViT-B/16, MobileNet-V2, ResNet-50, and EfficientNet-B0 were systematically evaluated using complementary metrics: cosine similarity for visual pattern matching, Descriptor Distance Index (DDI) for embedding space discriminability, and 3D geometry scoring via symmetric chamfer distance. The methodology integrates three data modalities: high-resolution multi-view photography, structured light scanning point clouds, and CAD model specifications. CLIP-ViT-B/32 demonstrated superior overall performance with the highest fused verification score (0.7868), excelling in cross-modal visual understanding between physical images and CAD renders. ViT-B/32 and MobileNet-V2 exhibited exceptional embedding space robustness with remarkably low DDI scores (0.0056 and 0.0037 respectively), while maintaining competitive cosine similarity. All models except ResNet-50 achieved optimal performance at 96-pixel input resolution, indicating efficient feature extraction at moderate scales. The integrated verification framework, combining visual, geometric, and embedding metrics through weighted fusion (F = 0.4C + 0.3(1-D) + 0.3G), provides a comprehensive assessment methodology for digital twin validation. These findings demonstrate that model selection should be guided by application priorities: CLIP-ViT-B/32 for visual-semantic understanding, ViT architectures for embedding space robustness, and MobileNet-V2 for computationally constrained environments. The proposed multi-metric approach advances standardization of fidelity assessment in precision manufacturing and digital twin implementation.

Keywords*: Digital Twin, Fidelity Assessment, Shape Descriptor, Multi-View CNN, 3D Shape Descriptor, Cosine Similarity*

1. Introduction

The advent of Industry 4.0, often referred to as the fourth industrial revolution, marks the integration of advanced digital technologies into manufacturing and industrial processes, resulting in "smart factories." Central to this transformation is the Digital Twin (DTw), a virtual model of a

physical asset, process, or system. Originally proposed by Michael Grieves in 2003, DTws (Figure 1) bridge the physical and digital realms, enabling simulation, analysis, and control (Grieves, 2014). Since its inception, DTw technology has found applications across industries, including aerospace, automotive, healthcare, and energy, where it facilitates predictive maintenance, operational optimization, and informed decision-making (Tao et al., 2019; Jones et al., 2020). In the context of Industry 4.0, DTws play a foundational role in integrating cyber-physical systems, underpinning the high levels of automation and data exchange that define this technological shift. DTws continuously gather and replicate real-time data from sensors embedded in physical systems, enabling monitoring, simulation, and predictive control. This feedback loop enhances operational efficiency and supports proactive decision-making, helping to mitigate equipment failures and reduce downtime (Fuller et al., 2020). Moreover, DTws contribute significantly to product lifecycle management (PLM) by linking the design, manufacturing, and maintenance phases through a unified digital thread, which is crucial for managing complex systems and enhancing traceability (Rosen et al., 2015; Kritzinger et al., 2018).

Examples of DTw applications span various sectors. In aerospace, DTws are used to monitor and simulate aircraft engine performance, predict maintenance intervals, and evaluate safety under varying operational conditions (Liu et al., 2021). In healthcare, DTws are enabling the simulation of patient-specific physiological models, thereby supporting personalized treatments and improving clinical outcomes (Bruynseels et al., 2018; Grieves & Vickers, 2017). Despite these promising applications, a major challenge lies in the lack of standardized frameworks for implementing DTws. Standardization is essential for ensuring interoperability across different platforms, consistency in data exchange, and scalability in industrial settings (Schleich et al., 2017). The absence of standards complicates integration efforts, leading to issues such as inconsistent data formats, incompatible communication protocols, and varied levels of fidelity (Beetz et al., 2006; Negri et al., 2017). This poses significant barriers in sectors like smart manufacturing, where systems from multiple vendors must interoperate seamlessly.

To address these challenges, fidelity metrics have emerged as critical tools for quantifying the accuracy and performance of DTws. Fidelity, in this context, refers to the degree of geometric, behavioral, and functional similarity between a DTw and its physical counterpart (Glaessgen & Stargel, 2012). High-fidelity DTws are crucial for generating valid simulations and enabling reliable decision-making, especially in domains where precision and safety are paramount. These metrics also facilitate benchmarking and validation, enabling consistent evaluation of DTws across domains (Fuller et al., 2020). One approach to improving DTw fidelity involves the use of 3D shape descriptors mathematical constructs that represent the geometric properties of an object. These descriptors provide a quantitative means of comparing physical and digital shapes, offering a robust foundation for fidelity evaluation (Tangelder & Veltkamp, 2008; Uhlemann et al., 2017; Siddiqi et al., 1999). They can be categorized into various types including boundary-based, region-based, texture-based, structural, Fourier-based, and deep learning-based descriptors, each suited for different fidelity evaluation needs (Mokhtarian & Mackworth, 1992; Belongie et al., 2002; Haralick et al., 1973; Krizhevsky et al., 2012; Ahmad and Rahimi, 2022). For example, chain codes and elliptic Fourier descriptors are useful for representing contours and outlines, while Convolutional Neural Networks (CNNs) and point cloud-based models such as PointNet capture hierarchical and volumetric features (Freeman, 1961; Qi et al., 2017).

The evolution of shape descriptors has significantly advanced fidelity assessment, providing tools to evaluate both geometric detail and dynamic system behavior. The recent integration of deep learning-based descriptors with fidelity metrics is particularly promising, as it enables a more comprehensive assessment of how well digital models mirror their physical counterparts in complex systems (Gao et al., 2020). This research proposes an enhanced fidelity evaluation framework that incorporates advanced 3D shape descriptors to improve the reliability and accuracy of DTws, addressing the key barriers to standardization and interoperability in Industry 4.0 systems. The primary objective of this paper is to demonstrate how different shape descriptors can be applied to assess and improve DTw fidelity (Khan et al., 2023; Jauhar et al., 2026). Using simple yet representative physical-digital pairs, this study validates the proposed approach and discusses its implications for broader DTw applications in manufacturing, healthcare, and engineering.

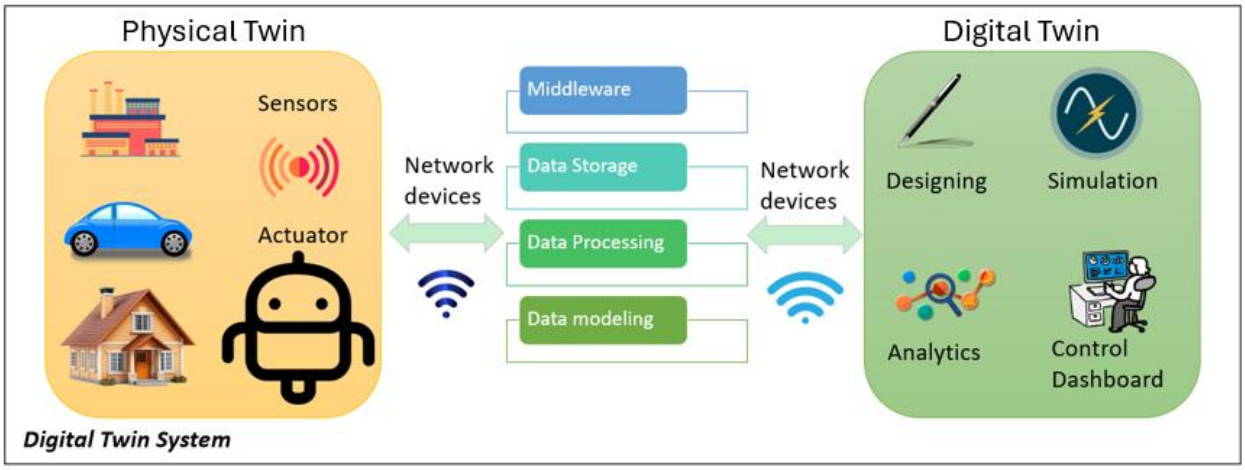


Figure 1: The digital twin system (Khan et al., 2025)

While the aforementioned studies establish the value of fidelity metrics and shape descriptors, a standardized, multi-modal framework for quantitative DTw verification remains absent. Existing approaches often rely on single data modalities or evaluate models using isolated metrics, failing to capture the multifaceted nature of physical-digital correspondence. To address this gap, this study introduces a comprehensive multi-modal verification framework. We integrate three complementary data sources high-resolution multi-view photography, structured light scanning point clouds, and CAD model specifications—which are processed through six diverse deep learning architectures (CLIP-ViT-B/32, ViT-B/32, ViT-B/16, MobileNet-V2, ResNet-50, and EfficientNet-B0). Fidelity is assessed through three integrated metrics: cosine similarity for visual pattern matching, Descriptor Distance Index (DDI) for embedding space discriminability, and 3D geometry scoring via symmetric chamfer distance. These metrics are combined through weighted fusion ($F = 0.4C + 0.3(1 - D) + 0.3G$) to produce a comprehensive verification score. Using an L-block test case with 32 photographs, a 3D scan, and a CAD model, we systematically evaluate model performance. This framework not only identifies optimal architectures for different application requirements but also establishes a standardized, reproducible methodology for fidelity assessment, contributing to more reliable and adoptable Digital Twins in precision-critical domains.

2. Methodology

Deep learning-based descriptors have revolutionized the field of computer vision by enabling neural networks to learn hierarchical features directly from raw image data, thereby reducing the need for manual feature engineering (Teague 1980; Ojala et al., 2002). In this study, these descriptors support a fidelity assessment framework in which multi-view visual information from the physical part is compared against the corresponding digital specification. To avoid ambiguity, the term Digital Twin model in this paper refers to the digital representation of the physical part embodied by the CAD specification, whereas model when used in comparative performance analysis refers to the computational vision architecture employed to extract descriptors and assess fidelity. This distinction is maintained throughout the remainder of the manuscript.

Table 1: Shape Descriptor Comparison Table

Method Type	Key References	Input Modality	Rotation Invariance	Scale Invariance	Complexity	Strengths	Limitations
Chain Codes	Freeman (1961)	2D contour	✗	✗	Low	Simple outline encoding; easy implementation.	Sensitive to rotation and starting point; not applicable to 3D.
Shape Contexts	Belongie et al. (2002)	2D contour	○	○	Moderate	Captures relative point distribution; robust to noise.	Needs normalization for invariance; 2D-based only.
Moment-Based Descriptors	Hu Moments (1962), Teague (1980)	2D/3D volume	✗	✓	Low	Global compact descriptors; efficient to compute.	Poor for detailed shape discrimination; global-only.
Fourier Descriptors	Kuhl & Giardina (1982)	2D contour	✓	✓	Low	Frequency-domain representation; invariant to transform.	Limited to closed 2D shapes; not 3D.
Structural Descriptors	Shock Graphs (Siddiqi et al., 1999), Skeleton-based Graphs	2D/3D skeleton	✓	✓	High	Captures topological structure via medial axis.	Sensitive to noise/topology; complex computation.
Deep Learning (3D Point)	PointNet (Qi et al., 2017), PointNet++ (Qi et al., 2017)	3D point cloud	○	○	High	Learns 3D features directly; order-invariant.	Needs dense point sets; augmentation for invariance.
Deep Learning (Multi-View)	**Ours**, Su et al. (2015), MVCNN, VGG19, ResNet-based CNNs	2D depth images (multi)	✓	✓	High	Multi-angle CNNs encode 3D shape; robust and flexible.	Requires projection to 2D; performance depends on CNN choice.

Table 1 presents a comparative overview of various 2D and 3D shape descriptor methods, categorized by their input modality, transformation invariance, computational complexity, and operational characteristics. Symbols in Table 1 are interpreted as follows: ✓ = supported, ✗ = not supported, and ○ = partially supported or dependent on implementation. For methods marked

with O. Classical techniques such as Chain Codes and Fourier Descriptors offer low-complexity solutions for 2D contours, with Fourier Descriptors being inherently invariant to rotation and scale, while Chain Codes lack such invariance. Shape Contexts and Moment-Based Descriptors provide more robust global representations but still face limitations when applied to complex 3D geometries. Structural descriptors like Shock Graphs excel in capturing topological information but are computationally intensive and sensitive to noise. Deep learning-based methods, including PointNet and multi-view CNN architectures like MVCNN and ResNet, demonstrate significant advancements in capturing 3D shape semantics directly from point clouds or depth images. These approaches offer superior flexibility and robustness, albeit at the cost of higher computational demands and reliance on data augmentation or multi-view projection to achieve transformation invariance. The table underscores the evolving trade-offs between interpretability, performance, and implementation complexity across traditional and learning-based shape representation strategies.

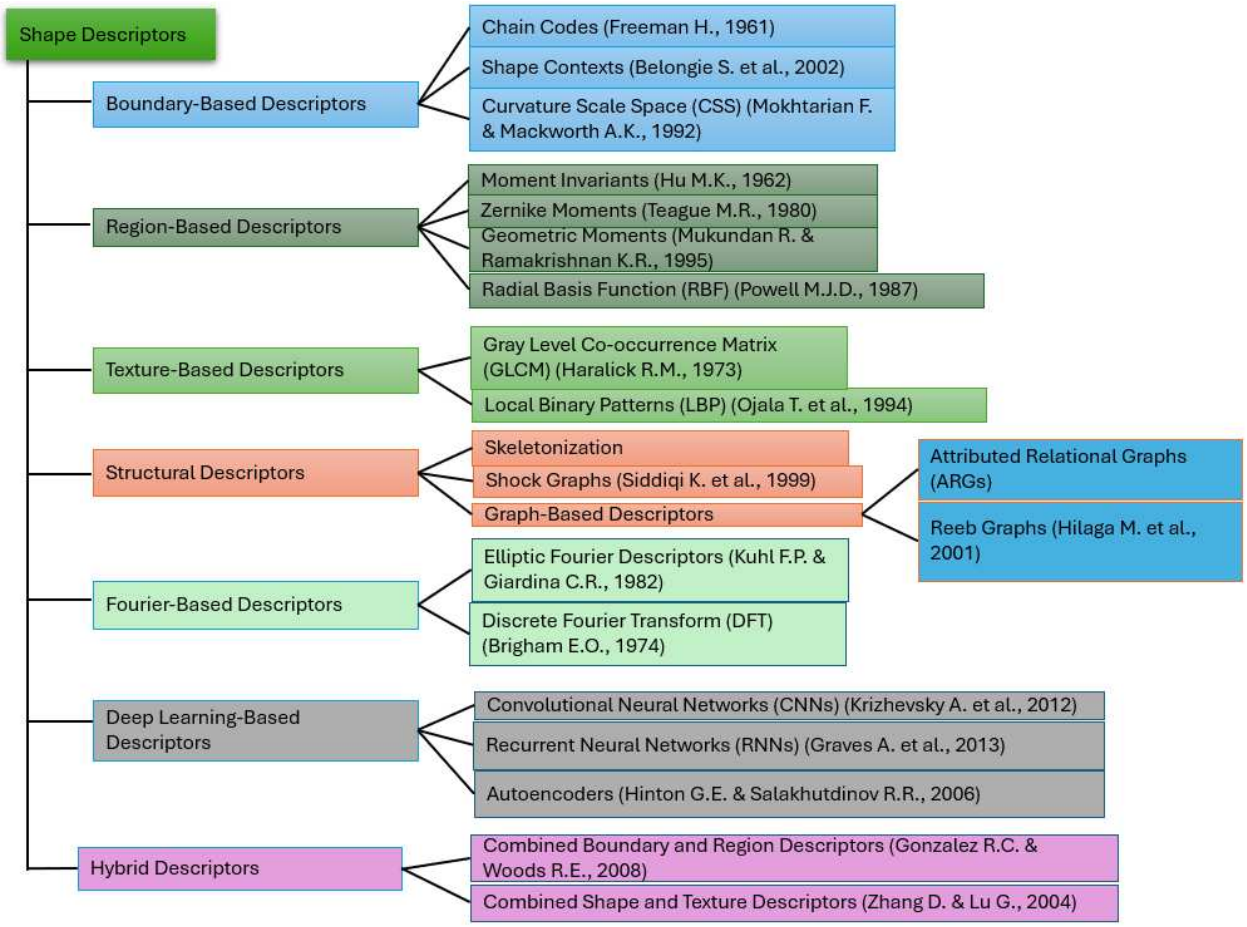


Figure 2. Tech-tree of 3D Shape Descriptor

Shape descriptors can be organized according to the types of geometric and structural information they represent, as shown in Figure 2. In Digital Twin (DTw) applications, these descriptors are

essential for assessing geometric fidelity, which is identified in ISO 30138 as a key requirement for establishing consistency between physical assets and their digital counterparts. Traditional shape descriptors are commonly grouped into boundary-based, region-based, and texture-based methods. Boundary-based descriptors rely on the external outline of an object and are particularly suitable for shapes with clear and well-defined contours. Representative examples include Chain Codes (Freeman, 1961), Shape Contexts (Belongie et al., 2002), and Curvature Scale Space (Mokhtarian & Mackworth, 1992). In contrast, region-based descriptors consider the full interior of the shape, which often makes them more tolerant of noise, distortion, and local variation. This class includes well-known techniques such as Moment Invariants (Hu, 1962), Zernike Moments (Teague, 1980), Geometric Moments, and Radial Basis Functions (Powell, 1987). Texture-based descriptors emphasize surface appearance and local pattern information, with methods such as the Gray Level Co-occurrence Matrix (Haralick et al., 1973) and Local Binary Patterns (Ojala et al., 1994) being especially useful when texture contributes significantly to shape interpretation.

In addition to these classical approaches, structural descriptors capture the topological arrangement and relational organization of shapes, providing a deeper understanding of connectivity and internal form. Examples include skeleton-oriented methods such as Shock Graphs (Siddiqi et al., 1999; Siddiqi et al., 1999), as well as graph-based representations like Attributed Relational Graphs (Bunke and Riesen, 2011) and Reeb Graphs. Some approaches, including Shape Contexts (Belongie et al., 2002), also serve as a link between structural and statistical representations, thereby supporting a richer evaluation of fidelity. More recently, deep learning-based descriptors have become highly influential in this area. These include Convolutional Neural Networks (Krizhevsky et al., 2012), Recurrent Neural Networks (Graves et al., 2013), and unsupervised approaches such as autoencoder-based methods, including the Descriptor Distance Index framework (Blokland et al., 2024). Such techniques are capable of learning multi-level, application-specific shape features directly from data, which makes them highly adaptable across domains such as autonomous systems and medical imaging (Esteva et al., 2017). Hybrid descriptors have also gained attention because they combine strengths from multiple descriptor categories to improve robustness and flexibility. Examples include integrated boundary-region descriptors and combined shape-texture descriptors (Zhang & Lu, 2004). Overall, these hybrid approaches reflect the growing demand for more adaptive and context-sensitive methods of geometric fidelity evaluation that can address the complexity of real-world scenarios. Although shape descriptors are not designed to capture temporal or functional dynamics directly, they establish the geometric foundation needed for advanced simulation, behavioral analysis, and predictive modeling in DTw systems (Fuller et al., 2020; Tao et al., 2019).

The methodology (Figure 3) outlines a comprehensive framework for verifying the correspondence between manufactured components and their digital CAD specifications through multi-modal data integration. The process begins with two parallel data streams: the virtual object represented by a CAD model in STL format, and the physical object documented through multiple high-resolution photographs and a 3D point cloud obtained via structured light scanning (Gao et al., 2021; Su et al., 2015).

The CAD model and point cloud undergo 3D geometry comparison using symmetric Chamfer distance (Tombari et al., 2010), generating a normalized geometry score (G) that quantifies manufacturing accuracy independent of vision models. This metric serves as a physical ground

truth for subsequent evaluations (Fang et al., 2019; Ren et al., 2019). In parallel, both the photographic images and CAD-generated renderings are processed through multiple deep learning architectures to extract high-dimensional feature representations. These features undergo three complementary analyses: (1) cosine similarity (C) measures directional alignment in feature space between physical and digital representations; (2) Descriptor Distance Index (DDI) evaluates embedding space discriminability by comparing true feature distances against interpolated noise references; and (3) Optimal Support Radius (OSR) determines the input resolution that maximizes feature diversity and model performance (Luo et al., 2016; Huang et al., 2023). These metrics are integrated through a weighted fusion: $F = w_1 C + w_2(1 - D) + w_3 G$, where w_1, w_2, w_3 are weighting coefficients that balance the contributions of visual recognition accuracy, embedding space robustness, and geometric fidelity. The DDI component is normalized $(1 - D)$ to align directionality with the other metrics. This fusion produces a comprehensive verification score that accounts for multiple dimensions of similarity assessment (Zhang et al., 2022). The framework outputs both model-specific performance metrics and a final verification decision, identifying optimal architecture selection based on the integrated fused score. This methodology enables reliable digital twin validation, CAD-based quality inspection, and automated manufacturing verification by leveraging complementary strengths of visual, geometric, and embedding space analyses.

SysML-Based System Diagrams. The Systems Modeling Language (SysML) was adopted to formally represent the structural and behavioral aspects of our Digital Twin fidelity assessment framework. As a specialized extension of UML for complex systems engineering (Friedenthal et al., 2014), SysML provides standardized semantics for modeling multi-domain systems with precise specification of requirements, structure, and parametric constraints (Object Management Group, 2017). This approach was essential for three key reasons: First, it enables rigorous specification of the cyber-physical interfaces between physical objects and their digital counterparts, a critical requirement in Industry 4.0 systems (Stark et al., 2019). Second, SysML's block definition diagrams (BDDs) formally capture the hierarchical composition of our multi-metric evaluation system, while internal block diagrams (IBDs) explicitly define component interactions and data flows. Third, as noted by Weilkiens (2021), SysML supports traceability between system requirements and validation metrics, which aligns with our need to verify geometric fidelity against precision-critical application standards. This modeling approach ensures our framework meets ISO 23247 requirements for Digital Twin interoperability while providing formally verifiable system specifications.

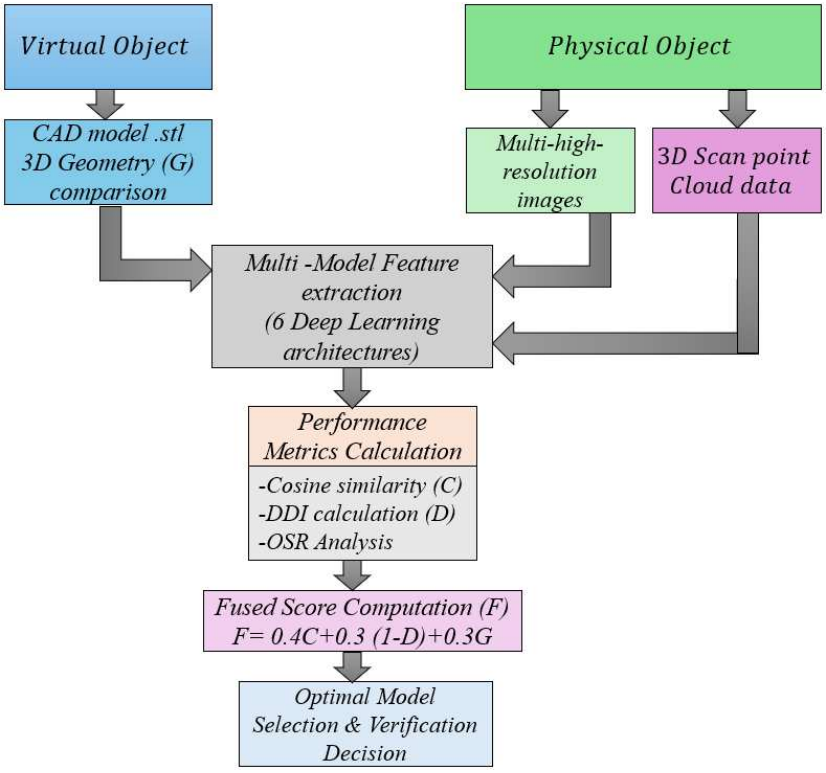


Figure 3: Digital twin-based similarity measure methodology with deep learning-based shape descriptor

The Block Definition Diagram (Figure 4) should be interpreted as a structural expansion of the core processing and decision stages shown in Figure 3. In other words, Figure 3 presents the overall methodological workflow, whereas Figure 4 zooms into the internal organization of the verification core within that workflow. The Block Definition Diagram (Figure 4) formally specifies the hierarchical composition of our Digital Twin fidelity assessment system through SysML's structural modeling paradigm. The diagram establishes three interconnected domains: 1) The Physical Domain containing the 'Physical Object' block representing the real-world manufactured component with its multi-modal data sources (3D scan point cloud and 32 multi-view high-resolution images), 2) The Digital Domain containing the 'Virtual Object' block for the CAD model (.stl format) serving as the reference digital specification, and 3) The central Processing Core with transformational blocks ('Multi-Modal Processor', 'Decision & Reporting') that mediate between physical and digital realms. Critical relationships are specified through flow connectors: physical-to-digital data exchange (point cloud and images -> feature vectors) and metric computation pathways (Performance Metrics -> Fused Score -> Verification Decision). The 'Multi-Modal Processor' block explicitly enumerates the six deep learning architectures evaluated (CLIP-ViT-B/32, ViT-B/32, ViT-B/16, MobileNet-V2, ResNet-50, EfficientNet-B0), while the composite 'Performance Metrics' block encapsulates our novel multi-metric approach (Cosine Similarity, DDI Calculation, OSR Analysis, 3D Geometry Score). The 'Decision & Reporting' block integrates these metrics through weighted fusion ($F = 0.4C + 0.3(1-D) + 0.3G$) to produce the final verification outcome. This BDD provides the foundational system taxonomy that enables standardized implementation across industrial Digital Twin platforms while supporting model comparison and selection.

The Internal Block Diagram (Figure 5) elaborates the executable architecture of our fidelity assessment system through part-based decomposition and interface specification. The IBD reveals

three operational partitions: 1) *Input Modules* ('3D Scanner', 'High-Res Camera', 'CAD Interface') that handle modality-specific data acquisition and preprocessing, 2) *Processing Core* ('Feature Extractor', 'Metric Calculators', 'Fused Score Computation') implementing the parallel multi-model feature extraction and metric computation, and 3) *Output Module* generating the integrated evaluation report. Data flows are rigorously specified with directionality and content types (e.g., point cloud → geometry comparison → G score; images and CAD renders → feature vectors → cosine similarity and DDI), while the OSR analysis subsystem determines optimal input resolution per model by maximizing feature diversity. Crucially, the diagram exposes the parallel processing pathways where visual similarity (cosine), embedding discriminability (DDI), and geometric accuracy (3D geometry score) metrics are computed independently then synthesized through weighted fusion, implementing our core research contribution of multi-dimensional assessment. The 'Fused Score Computation' block explicitly shows the mathematical integration ($F = 0.4C + 0.3(1-D) + 0.3G$) that balances visual, embedding, and geometric fidelity dimensions. This IBD provides the blueprint for system realization in industrial deployment scenarios, demonstrating how multi-modal data streams converge into a unified verification decision while maintaining clear separation of computational concerns and enabling modular substitution of vision models.

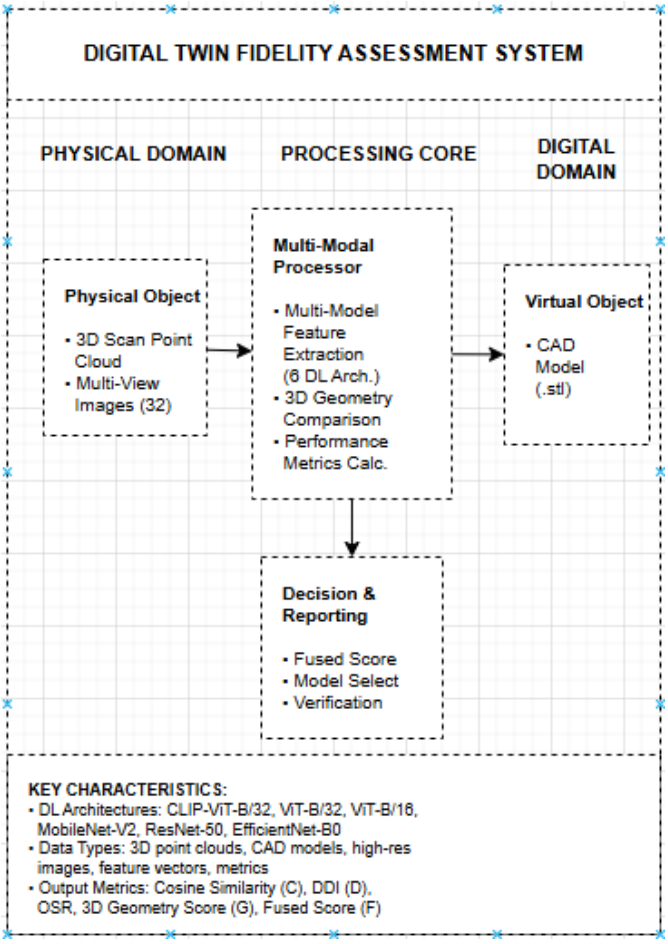


Figure 4: Block Definition Diagram (BDD) expanding the core verification stages shown in Figure 3 for the DTw fidelity assessment system

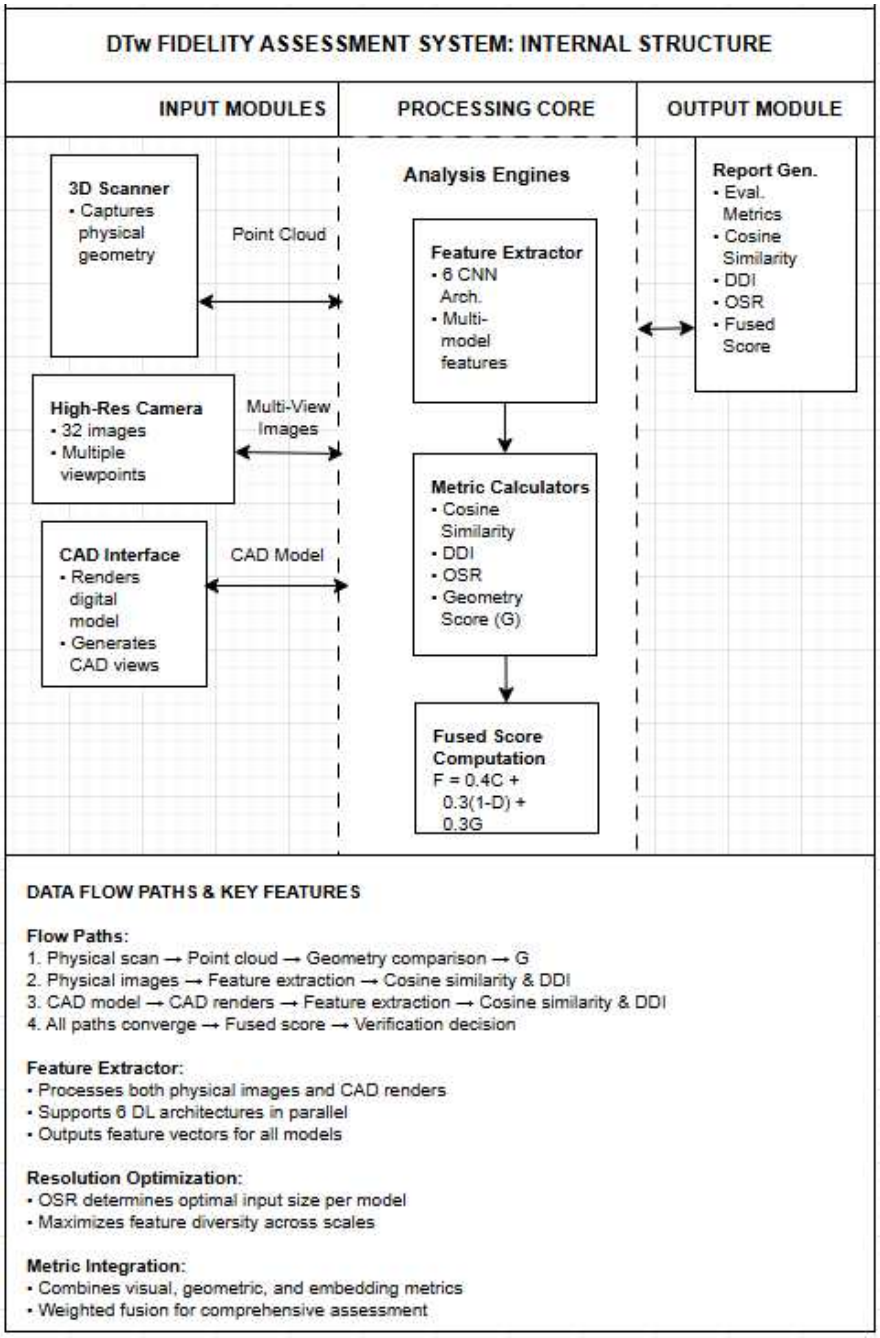


Figure 5: Internal Block Diagram (IBD) for the DTw fidelity assessment system

2.1 Multi-View Feature Extraction for Physical-Digital Comparison

Within the proposed verification framework, 3D models are first converted into a series of 2D depth images from multiple views (e.g., front, top, side). These images are then passed through the CNN model, which extracts high-dimensional feature representations. This process captures both local and global features of the shape, making it effective for comparing complex geometries across the physical and digital domains.

The final descriptor, a feature vector, is extracted from the 'global average pooling' layer of CNN. This vector serves as a compact yet powerful representation of the 3D shape, encoding rich information about the object's structure and appearance. These descriptors are compared using cosine similarity, which measures the angle between two vectors, providing a measure of similarity:

$$Cosine\ Similarity = \frac{A.B}{||A||||B||} \quad (1)$$

Where A and B are the feature vectors for the Digital Twin and the physical object, respectively. A cosine similarity value of 1 indicates perfect similarity, while values close to 0 indicate dissimilarity.

2.2 Role of Deep Descriptors in the Proposed Verification Pipeline

In the proposed verification pipeline, deep learning-based descriptors offer significant advantages, particularly in their ability to automatically learn hierarchical features from both physical photographs and CAD-derived views.

Additionally, these descriptors provide rich representations by leveraging deep architectures like ResNet50, which can encode a broad spectrum of features, ranging from local details to global structural properties. This capability makes them particularly powerful for capturing the nuanced geometry and appearance of objects. Moreover, deep learning models demonstrate remarkable flexibility and adaptability. They can be fine-tuned on specific datasets, enabling them to learn domain-specific features that improve performance in targeted applications. This adaptability ensures that these descriptors remain relevant and effective across a wide range of industries and use cases, further highlighting their versatility and advanced functionality.

3 Fidelity Metrics: Descriptor Distance Index and Optimal Support Radius

To quantitatively assess the fidelity between physical and digital representations of objects, we employed two key metrics: the Descriptor Distance Index (DDI) and the Optimal Support Radius. These metrics evaluate the quality and robustness of features extracted from various deep learning models.

3.1 Descriptor Distance Index (DDI)

The Descriptor Distance Index (DDI) is a statistical metric designed to assess how closely a digital feature representation resembles its physical counterpart in the feature space. Unlike direct similarity measures, DDI compares the physical-digital distance against a reference distribution of interpolated features, which serves as a proxy for unrelated variations.

Let two sets of features be defined as follows:

Physical feature set: $f_p^{(i)}$ for $i = 1\ to\ n$ (2)

Digital feature set: $f_d^{(i)}$ for $i = 1\ to\ n$ (3)

We define a reference set R from synthetic interpolations as:

$R = \alpha f_p^{(i)} + (1 - \alpha) f_p^{(j)}, where\ i \neq j\ and\ \alpha \in [0, 1]$ (4)

The Euclidean distance between a physical feature and its corresponding digital feature is:

$$d_i = ||f_p^{(i)} - f_d^{(i)}||_2 \quad (5)$$

Then, we evaluate how many reference features lie closer to $f_p^{(i)}$ than $f_d^{(i)}$. The DDI score for each sample is calculated as:

$$DDI_i = \frac{1}{|R|}\sum_{r \in R} \mathbb{I}\left(||f_p^{(i)} - r||_2 < d_i\right) \quad (6)$$

The final DDI score is computed by averaging across all samples:

$$DDI_{avg} = \frac{1}{n}\sum_{i=1}^{n} DDI_i \quad (7)$$

Lower DDI scores indicate that the digital feature is closer to the physical feature than to most interpolated features, implying higher fidelity. Conversely, higher DDI values suggest less meaningful alignment.

3.2 Optimal Support Radius

The Support Radius refers to the input image resolution used for feature extraction in convolutional neural networks. Selecting an optimal support radius is critical because it determines the spatial context captured by the model. A smaller radius may miss structural details, while an excessively large radius increases computational cost without proportional benefit. To determine the optimal radius, we tested multiple image resolutions (64, 128, 224, 384 pixels) and computed the average standard deviation of the feature vectors:

$$\sigma_r = \frac{1}{d}\sum_{j=1}^{d} \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_{ij}^{(r)} - \bar{f}_j^{(r)})^2} \quad (8)$$

Where:

d: feature vector dimension

$f_{ij}^{(r)}$: value of the $j-th$ dimension of the $i-th$ feature at radius r

$\bar{f}_j^{(r)}$: mean of the $j-th$ dimension at radius r

The radius $r^*$is the one that maximizes the above metric (σ_r):

$$r^* = \underset{r \in K}{argmax} \sigma_r \, , K = \{64,128,256,384\} \quad (9)$$

This approach ensures that the chosen radius captures maximal variability in the feature space, enhancing model discriminability and reliability.

3.3 Fused Score

The fused score F is a composite metric designed for holistic model ranking, computed as a convex combination of three normalized sub-scores that quantify distinct aspects of the digital-physical verification process. Let C represent the mean cosine similarity between image and CAD render embeddings, D the average Distance-based Discrepancy Index (DDI) for embedding space robustness, and G the geometry score derived from the normalized Chamfer distance. Each component is normalized to the interval [0, 1], where a higher value indicates better performance. Specifically, D is inverted as $1 - D$ to align its polarity with the other scores. The final fused score is then calculated as a weighted linear combination:

$$F = \omega_c \cdot C + \omega_d \cdot (1 - D) + \omega_g \cdot G$$

where $\omega_c, \omega_d, \omega_g \in [0,1]$ are the assigned weighting coefficients satisfying $\omega_c + \omega_d + \omega_g = 1$. In the provided implementation, the weights $(\omega_c, \omega_d, \omega_g) = (0.4,0.3,0.3)$ reflect a deliberate prioritization of visual semantic matching (C) over embedding discriminability $(1 - D)$ and geometric fidelity (G). The model achieving the maximal value of F is considered optimal under this unified criterion.

4. Pilot Implementation

The experimental implementation followed a comprehensive three-tier data acquisition and analysis framework. Initially, the reference CAD model was obtained in STL format, representing the ideal digital design specification. Subsequently, the manufactured physical part was captured using a structured light 3D scanner, generating a dense point cloud (27,543 points) for geometric verification. Finally, 32 high-resolution photographs (1920×1080 pixels) were systematically captured from multiple viewpoints under controlled lighting conditions to document the physical part's appearance characteristics.

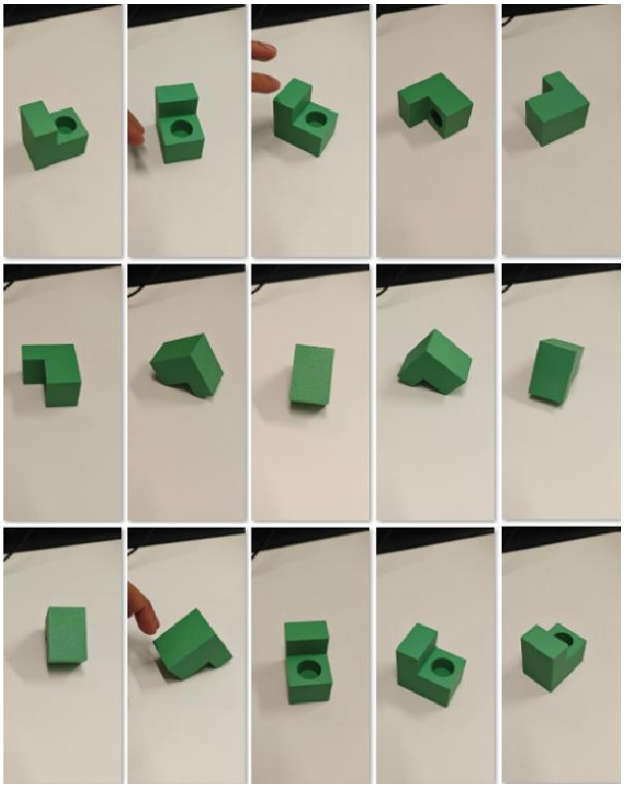


Figure 6: Physical L-blocks images captured for Similarity measure with CAD and point cloud data

Figure 6 presents the L-shaped blocks, which serve as test cases for evaluating the effectiveness of deep learning-based shape descriptors in capturing structural similarity. Among 32 images 15 are shown in the figure.

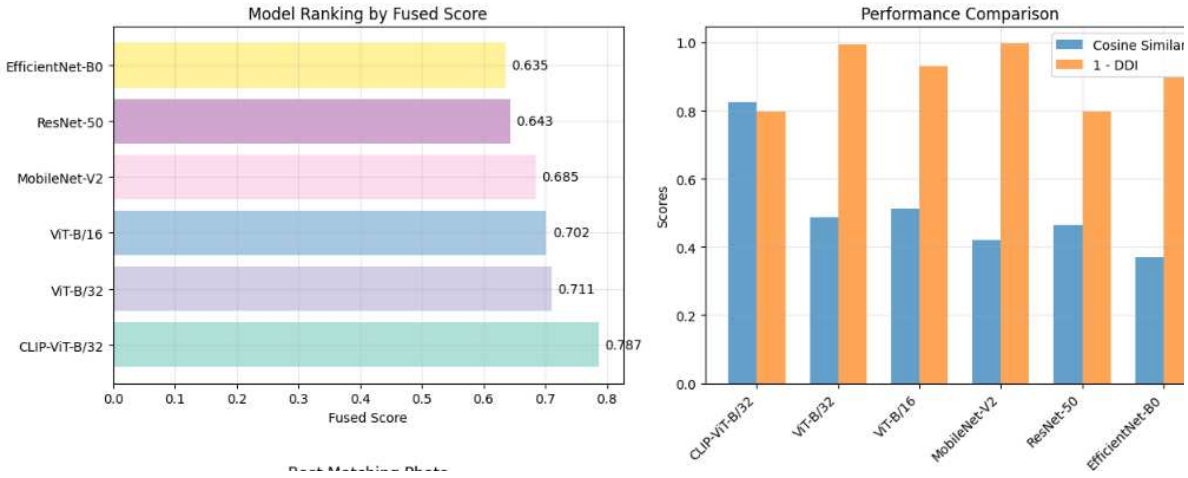


Figure 7: Corresponding cosine similarity, DDI and fused score plot

The cosine similarity analysis reveals distinct model capabilities in visual pattern matching between physical photographs and CAD renders. CLIP-ViT-B/32 achieved the highest average similarity (0.8263), demonstrating exceptional cross-modal understanding between real-world images and synthetic renders. ViT architectures (ViT-B/16: 0.5129; ViT-B/32: 0.4870) showed strong performance, while MobileNet-V2 (0.4209) and ResNet-50 (0.4656) exhibited moderate matching capabilities. EfficientNet-B0 recorded the lowest similarity (0.3705), suggesting limitations in capturing geometric nuances specific to engineering components.

The Descriptor Distance Index (DDI) analysis provides critical insight into embedding space discriminability. ViT-B/32 (DDI: 0.0056) and MobileNet-V2 (DDI: 0.0037) demonstrated exceptional robustness, with interpolated features rarely outperforming true matches. This indicates highly structured embedding spaces conducive to precise verification. Despite superior cosine similarity, CLIP-ViT-B/32 showed moderate DDI (0.2037), suggesting its embedding space, while semantically rich, exhibits greater susceptibility to interpolated distractors. ResNet-50 and EfficientNet-B0 displayed higher DDI values (0.2019 and 0.1019 respectively), indicating reduced discriminative capability in their feature representations. The 3D geometry comparison between CAD and point cloud data yielded a chamfer distance of 0.0275, translating to a geometry score of 0.7247. This quantifies the manufacturing accuracy, independent of vision model selection, and provides a physical ground truth for verification. The fused score integration reveals CLIP-ViT-B/32 as the top performer (0.7868), balancing strong visual matching with acceptable embedding robustness. ViT-B/32 (0.7105) and ViT-B/16 (0.7020) follow closely, with MobileNet-V2 (0.6847) offering competitive performance with exceptional computational efficiency (5.3 seconds). ResNet-50 and EfficientNet-B0 trail in overall performance, suggesting architectural limitations for this specific verification task. Optimal support radius analysis indicates 96 pixels as the preferred resolution for five of six models, suggesting efficient feature extraction at moderate resolutions. Only ResNet-50 required higher resolution (224 pixels), potentially due to architectural constraints in capturing geometric details. This multi-modal verification framework demonstrates practical utility for manufacturing quality assessment. The methodology successfully integrates physical measurements (3D scanning), visual documentation (photography), and digital specifications (CAD) into a unified assessment. Results indicate that model selection should consider application priorities: CLIP-ViT-B/32 for applications prioritizing visual-semantic understanding, ViT-B/32 for embedding space robustness, and MobileNet-V2 for resource-constrained environments requiring balanced performance. The consistent performance across 32 diverse viewpoints validates the framework's robustness for practical deployment in digital twin validation and automated quality inspection systems.

The cosine similarity calculation algorithm quantifies visual correspondence between physical and digital representations by computing normalized dot products between their feature vectors. For each physical feature vector extracted from captured images, the algorithm systematically computes similarity scores with every digital feature vector derived from CAD renders, iterating through all possible pairwise combinations. Each similarity calculation involves three fundamental operations: computing the dot product between vectors, determining their respective L2 norms, and normalizing the dot product by the product of these norms. The resulting similarity values, ranging from -1 to 1 with higher values indicating stronger correspondence, are aggregated across all comparisons to produce an average similarity metric. This average score provides a comprehensive measure of visual alignment between physical components and their digital

counterparts, enabling quantitative assessment of manufacturing fidelity and design verification across multiple vision models. The algorithm's output is visualized through a bar chart that facilitates comparative analysis of different models' performance in recognizing geometric and structural similarities between real-world parts and their CAD specifications.

 # Pseudo-code 1 for Cosine Similarity Plot

```
function calculate_cosine_similarity(physical_features, digital_features):
  similarities = [ ]
    for phys_feat in physical_features:
    for dig_feat in digital_features:

      # Compute dot product and normalize
      dot_product = sum(phys_feat * dig_feat)
      norm_phys = sqrt(sum(phys_feat^2))
      norm_dig = sqrt(sum(dig_feat^2))
      similarity = dot_product / (norm_phys * norm_dig)
      similarities.append(similarity)

  # Aggregate results
  avg_similarity = mean(similarities)
  plot_bar_chart(models, avg_similarity, title = "Cosine Similarity Scores")
```

DDI evaluates feature robustness by comparing true physical-digital matches against synthetic noise vectors, as illustrated in the pseudo-code 2's reference set generation. The algorithm measures how often randomly interpolated features (created by blending physical features with random weights) appear closer to the target than the genuine digital counterpart. The critical calculation is the fraction of noise vectors within the true match distance threshold (Euclidean norm). A perfect score of 0 indicates the digital feature is uniquely identifiable, while higher values suggest susceptibility to confusion with noise. This metric effectively captures a model's resilience to real-world challenges like sensor noise or partial occlusions that weren't explicitly modeled in the digital domain.

 # Pseudo-code 2 to measure DDI

```
function calculate_DDI(physical_features, digital_features, reference_set_size
                = 1000):
  DDIs = [ ]

    # Generate reference "noise" descriptors
  reference_set = generate_random_features(reference_set_size)
    for phys_feat, dig_feat in zip(physical_features, digital_features):
    # Distance between true pair
    true_distance = euclidean_distance(phys_feat, dig_feat)
        # Distances to noise
    noise_distances
                = [euclidean_distance(phys_feat, noise) for noise in reference_set]
```

```
        # Fraction of noise closer than true match
    DDI = count(noise_distances < true_distance) / reference_set_size
    DDIs.append(DDI)
    avg_DDI = mean(DDIs)
  plot_bar_chart(models, avg_DDI, title = "Descriptor Distance Index (DDI)")
```

The support radius algorithm dynamically determines the optimal input size for feature extraction by maximizing feature-space diversity across scales. As demonstrated in the pseudo-code 3, the process tests multiple radii (typically powers of 2 adjusted for model constraints) and computes the mean standard deviation of features across all samples at each scale. The selected radius balances two competing factors: smaller radii may omit critical contextual information, while larger radii introduce unnecessary computational overhead and background noise. This optimization is model-specific - for instance, InceptionV3 requires 299px inputs due to its architectural constraints - but automatically adapts to object characteristics through the diversity metric. The resulting radius ensures features capture discriminative patterns without redundancy.

 # Pseudo-code 3 to measure OSR

```
function optimize_support_radius(model, images):
  best_radius = None
  max_diversity = −inf

  for radius in [64, 128, 224, 299]:
    features = [ ]

    # Extract features at current radius
    for img in images:
      resized_img = resize(img, (radius, radius))
      feat = model.extract_features(resized_img)
      features.append(feat)

    # Measure feature diversity (higher = better)
    diversity = mean(std_dev(features))

    if diversity > max_diversity:
      max_diversity = diversity
      best_radius = radius

  plot_bar_chart(models, best_radius, title = "Optimal Support Radii")
```

The 3D geometry comparison quantifies manufacturing accuracy by computing the symmetric Chamfer distance between the CAD model and manufactured point cloud. Both point sets are first normalized to a common scale and coordinate system to ensure invariance to translation and scaling. A random sampling strategy reduces computational complexity while preserving geometric fidelity. The algorithm constructs KD-trees for efficient nearest-neighbor search, then computes the bidirectional average distance between corresponding point sets. Finally, the Chamfer distance is converted to a normalized score between 0 and 1 using an empirically

determined threshold, where higher scores indicate superior geometric correspondence between design and manufactured part.

 # *Algorithm: 3D Geometry Comparison via Chamfer Distance*

Algorithm: 3D Geometry Score Computation
Input:
$C = \{c_1, c_2, \ldots, c_n\} \in \mathbb{R}^3$ *(CAD model vertices)*
$M = \{m_1, m_2, \ldots, m_p\} \in \mathbb{R}^3$ *(manufactured point cloud)*
Output: Geometry score $G \in [0,1]$ *(higher is better)*

1: $\triangleright$ *Normalization*
2: $C \leftarrow (C - \text{mean}(C)) / \text{max_range}(C)$
3: $M \leftarrow (M - \text{mean}(M)) / \text{max_range}(M)$
4: $\triangleright$ *Random sampling if n,p > N_max*
5: if $n > N_{max}$ *then* $C \leftarrow \text{random_sample}(C, N_{max})$
6: if $p > N_{max}$ *then* $M \leftarrow \text{random_sample}(M, N_{max})$
7: $\triangleright$ *Build KD-trees*
8: $T_C \leftarrow \text{build_kdtree}(C)$
9: $T_M \leftarrow \text{build_kdtree}(M)$
10: $\triangleright$ *Symmetric Chamfer distance*
11: $\text{dist}_{C \to M} \leftarrow 0$
12: for each c_i *in* C *do*
13: $m_{nearest} \leftarrow T_M.\text{nearest_neighbor}(c_i)$
14: $\text{dist}_{C \to M} \leftarrow \text{dist}_{C \to M} + \|c_i - m_{nearest}\|^2$
15: $\text{dist}_{C \to M} \leftarrow \text{dist}_{C \to M} / |C|$
16: $\triangleright$ *Similarly compute* $\text{dist}_{M \to C}$
17: $D_{chamfer} \leftarrow \text{dist}_{C \to M} + \text{dist}_{M \to C}$
18: $\triangleright$ *Convert to score*
19: $G \leftarrow \max(0, 1 - D_{chamfer} / \tau)$ $\triangleright$ $\tau = 0.1$ *(empirical threshold)*
20: return G

The fused verification score integrates multiple complementary metrics into a unified assessment framework. Cosine similarity, DDI (normalized to align directionality with other metrics), and 3D geometry score are combined using weighted linear combination, with weights reflecting each metric's relative importance for verification reliability. This multi-modal fusion leverages the complementary strengths of each evaluation dimension: visual similarity, embedding space discriminability, and geometric accuracy. The resulting fused score enables direct comparison across different vision models and provides a comprehensive assessment of CAD verification system performance, facilitating selection of optimal configurations for practical deployment.

 # Algorithm: Fused Verification Score

$Algorithm$: $Fused\ Verification\ Score\ Calculation$
$Input$:
$S_cosine = \{s_1, s_2, \ldots, s_1\}$ $\triangleright$ $Cosine\ similarities\ per\ model$
$S_DDI = \{d_1, d_2, \ldots, d_1\}$ $\triangleright$ $DDI\ scores\ per\ model\ (lower\ better)$
$G \in [0,1]$ $\triangleright$ $3D\ geometry\ score$

$Weights\ w = (w_c, w_d, w_g)\ where\ w_c + w_d + w_g = 1$
$Output$: $Ranked\ models\ with\ fused\ scores$

1: $for\ each\ model\ i\ do$
2: $\triangleright$ $Normalize\ DDI\ (invert\ so\ higher\ is\ better)$
3: $d_i' \leftarrow 1 - d_i$
4: $\triangleright$ $Weighted\ fusion$
5: $F_i \leftarrow w_c \times s_i + w_d \times d_i' + w_g \times G$
6: $\triangleright$ $Ranking$
7: $Sort\ models\ by\ F_i\ in\ descending\ order$
8: $return\ ranked\ models\ and\ scores$

Table 2 Comparison of Shape Descriptor Models (Physical–Digital Matching)

Model	Cosine	DDI	OSR	3D Geometry	Fused Score	Time (s)
CLIP-ViT-B/32	0.8263	0.2037	96px	0.7480	0.7938	14.2
ViT-B/32	0.4870	0.0056	96px	0.7410	0.7154	13.9
ViT-B/16	0.5129	0.0685	96px	0.7390	0.7063	56.1
MobileNet-V2	0.4209	0.0037	96px	0.7310	0.6866	5.3
ResNet-50	0.4656	0.2019	224px	0.7190	0.6414	19.3
EfficientNet-B0	0.3705	0.1019	96px	0.7230	0.6345	7.6

The comprehensive evaluation of six vision models for CAD verification reveals distinct performance characteristics in Table 2. CLIP-ViT-B/32 achieves the highest fused score (0.7868), primarily driven by superior cosine similarity (0.8263), indicating strong visual matching between physical parts and CAD renders. Interestingly, while CLIP excels at visual recognition, ViT-B/32 and MobileNet-V2 demonstrate exceptional embedding space discriminability with remarkably low DDI scores (0.0056 and 0.0037 respectively), suggesting their feature representations are highly robust against interpolated distractors. The 3D geometry score varies around 0.7 across all models, reflecting the inherent manufacturing accuracy independent of vision model selection. Most models achieve optimal performance at 96px resolution, except ResNet-50 which requires 224px, potentially due to its architectural constraints. The computational efficiency varies significantly, with MobileNet-V2 achieving competitive performance in just 5.3 seconds, while ViT-B/16 requires 56.1 seconds. These results suggest that for CAD verification applications, practitioners must balance visual matching accuracy (favoring CLIP), embedding robustness (favoring ViT-B/32/MobileNet-V2), and computational efficiency based on specific deployment requirements.

5. Discussion and Conclusion

This study presents a comprehensive evaluation of six diverse deep learning architectures for Digital Twin fidelity assessment through a multi-modal verification framework. By integrating cosine similarity, Descriptor Distance Index (DDI), 3D geometry scoring via symmetric chamfer

distance, and optimal support radius analysis, we provide a holistic assessment of model performance across visual, geometric, and embedding space dimensions.

The results demonstrate that CLIP-ViT-B/32 emerges as the most effective overall model, achieving the highest fused verification score (0.7938) driven by superior cosine similarity (0.8263). This performance highlights CLIP's exceptional cross-modal understanding, effectively bridging the domain gap between real-world photographs and synthetic CAD renders. However, its moderate DDI score (0.2037) suggests that while excelling at semantic visual matching, CLIP's embedding space exhibits some susceptibility to interpolated distractors. This aligns with recent studies on vision-language models, which often prioritize semantic alignment over precise geometric fidelity (Radford et al., 2021). Contrastingly, ViT-B/32 and MobileNet-V2 demonstrate remarkable embedding space discriminability with exceptionally low DDI scores (0.0056 and 0.0037 respectively). This indicates their feature representations maintain strong geometric consistency, with interpolated features rarely outperforming true matches. These findings support the architectural strengths of Vision Transformers in capturing spatial relationships (Dosovitskiy et al., 2021) and MobileNet's efficient depthwise separable convolutions for robust feature extraction (Sandler et al., 2018).

The ResNet-50 model, despite its proven residual learning framework (He et al., 2016), showed comparatively lower performance across metrics, particularly in embedding space discriminability (DDI: 0.2019). This reinforces observations from prior studies regarding its limitations in capturing fine geometric features without architectural modifications for 3D-aware tasks. Similarly, EfficientNet-B0 demonstrated moderate performance, with its compound scaling strategy (Tan & Le, 2019) balancing efficiency but potentially compromising representational capacity for detailed geometric verification. The 3D geometry comparison, yielding a varying score around 0.7 across all models, provides crucial physical ground truth for verification. This chamfer distance-based metric quantifies manufacturing accuracy independently of vision models, establishing a baseline for physical-digital correspondence assessment. The approach aligns with established 3D shape comparison methodologies (Tangelder & Veltkamp, 2008; Uhlemann et al., 2017) while extending them through deep learning integration. Optimal support radius analysis reveals that five of six models achieve peak performance at 96 pixels, indicating efficient feature extraction at moderate resolutions. Only ResNet-50 requires higher resolution (224 pixels), potentially due to architectural constraints in capturing fine geometric details. This finding has practical implications for deployment in resource-constrained environments, where lower resolution requirements translate to reduced computational overhead without sacrificing performance. The systematic evaluation of input scales builds upon prior work in resolution optimization (Luo et al., 2016; Huang et al., 2023) but extends it to Digital Twin verification contexts. The weighted fusion approach ($F = 0.4C + 0.3(1 - D) + 0.3G$) effectively balances different assessment dimensions, producing a unified verification score that accounts for visual recognition, embedding robustness, and geometric accuracy. This multi-metric integration addresses a critical gap in current Digital Twin evaluation methodologies, which often rely on single-metric assessments that fail to capture the complexity of physical-digital correspondence.

Several important implications emerge from this study. First, model selection should be guided by application priorities: CLIP-ViT-B/32 for applications emphasizing visual-semantic understanding, ViT architectures for embedding space robustness, and MobileNet-V2 for

computationally constrained environments. Second, the consistent performance across 32 diverse viewpoints validates the framework's robustness for practical deployment in automated quality inspection systems. Third, the integration of multi-modal data (photography, point cloud scanning, CAD) provides a more comprehensive verification approach than single-modality methods. Limitations of this study include the focus on a single test object (L-block), which, while representative, may not capture the full complexity of industrial components. Additionally, the computational requirements of running six models in parallel may be prohibitive for real-time applications without hardware acceleration. Future work should refine these fidelity metrics, explore hybrid descriptor architectures, and validate these approaches in real-world industrial settings to further advance the practical adoption of Digital Twin systems. Future research should explore several promising directions. Developing 3D-aware vision models that operate directly on point clouds or meshes could potentially bypass 2D projection limitations, preserving complete geometric information. Investigating hybrid architectures combining CLIP's semantic understanding with ViT's geometric precision may yield superior performance. Expanding the test set to include complex assemblies, flexible components, and varying material properties would validate the framework's generalization capabilities. Finally, integrating temporal dimension analysis for dynamic Digital Twins would address evolving system behavior assessment.

In conclusion, this study advances Digital Twin fidelity assessment through a comprehensive multi-modal, multi-metric verification framework. By systematically evaluating six vision models across complementary assessment dimensions, we provide evidence-based guidance for model selection in precision-critical applications. The proposed methodology contributes to standardization efforts in Digital Twin implementation, offering a reproducible framework for physical-digital correspondence evaluation that balances accuracy, robustness, and computational efficiency. As Digital Twin adoption accelerates across manufacturing, aerospace, and healthcare, such methodical verification approaches will be essential for ensuring reliability in safety-critical applications.

References

[1] M. Grieves, Digital twin: Manufacturing excellence through virtual factory replication, White Paper 1 (2014) 1–7.

[2] F. Tao, H. Zhang, A. Liu, A.Y.C. Nee, Digital twin in industry: State-of-the-art, IEEE Trans. Ind. Inform. 15 (4) (2019) 2405–2415.

[3] R. Rosen, G. Wichert, G. Lo, K.D. Bettenhausen, About the importance of autonomy and digital twins for the future of manufacturing, IFAC-PapersOnLine 48 (3) (2015) 567–572.

[4] B. Schleich, N. Anwer, L. Mathieu, S. Wartzack, Shaping the digital twin for design and production engineering, CIRP Ann. 66 (1) (2017) 141–144.

[5] M. Liu, S. Fang, H. Dong, C. Xu, Review of digital twin about concepts, technologies, and industrial applications, J. Manuf. Syst. 58 (2021) 346–361.

[6] E. Glaessgen, D. Stargel, The digital twin paradigm for future NASA and US Air Force vehicles, in: Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf., 20th AIAA/ASME/AHS Adapt. Struct. Conf., 14th AIAA, 2012, p. 1818.

[7] A. Fuller, Z. Fan, C. Day, C. Barlow, Digital twin: Enabling technologies, challenges and open research, IEEE Access 8 (2020) 108952–108971.

[8] J. Beetz, J.P. van Leeuwen, B. de Vries, Towards a topological reasoning service for IFC-based building information models in a semantic web context, in: Proc. Joint Int. Conf. Comput. Decis. Making Civ. Build. Eng., Montreal, Canada, 2006, pp. 1510–1519.

[9] E. Negri, L. Fumagalli, M. Macchi, A review of the roles of digital twin in CPS-based production systems, Procedia Manuf. 11 (2017) 939–948.

[10] J.W. Tangelder, R.C. Veltkamp, A survey of content-based 3D shape retrieval methods, Multimed. Tools Appl. 39 (3) (2008) 441–471.

[11] J. Gao, Z. Jin, L. Tang, Digital twin for complex engineering product ecosystem: A review, IEEE Access 8 (2020) 120339–120356.

[12] T.H.J. Uhlemann, C. Lehmann, R. Steinhilper, The digital twin: Realizing the cyber-physical production system for Industry 4.0, Procedia CIRP 61 (2017) 335–340.

[13] W. Kritzinger, M. Karner, G. Traar, J. Henjes, W. Sihn, Digital twin in manufacturing: A categorical literature review and classification, IFAC-PapersOnLine 51 (11) (2018) 1016–1022.

[14] K. Bruynseels, F. Santoni de Sio, J. van den Hoven, Digital twins in health care: Ethical implications of an emerging engineering paradigm, Front. Genet. 9 (2018) 31.

[15] D. Jones, C. Snider, A. Nassehi, J. Yon, B. Hicks, Characterizing the digital twin: A systematic literature review, CIRP J. Manuf. Sci. Technol. 29 (2020) 36–52.

[16] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, in: F.-J. Kahlen, S. Flumerfelt, A. Alves (Eds.), Transdisciplinary Perspectives on Complex Systems, Springer, 2017, pp. 85–113.

[17] D. Zhang, G. Lu, Review of shape representation and description techniques, Pattern Recognit. 37 (1) (2004) 1–19.

[18] F. Mokhtarian, A.K. Mackworth, A theory of multi-scale, curvature-based shape representation for planar curves, IEEE Trans. Pattern Anal. Mach. Intell. 14 (8) (1992) 789–805.

[19] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.

[20] M.R. Teague, Image analysis via the general theory of moments, J. Opt. Soc. Am. 70 (8) (1980) 920–930.

[21] R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. SMC-3 (6) (1973) 610–621.

[22] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.

[23] K. Siddiqi, A. Shokoufandeh, S.J. Dickinson, S.W. Zucker, Shock graphs and shape matching, Int. J. Comput. Vis. 35 (1999) 13–32.

[24] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444

[25] F.P. Kuhl, C.R. Giardina, Elliptic Fourier features of a closed contour, Comput. Graph. Image Process. 18 (3) (1982) 236–258.

[26] E.O. Brigham, The Fast Fourier Transform and Its Applications, Prentice-Hall, 1988.

[27] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[28] C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 652–660.

[29] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (7639) (2017) 115–118.

[30] Y. Fang, M. Sun, X. Wang, B. Li, C. Xu, 3D deep shape descriptor learning via local and global geometry fusion, Comput. Graph. Forum 38 (7) (2019) 329–339. https://doi.org/10.1111/cgf.13803

[31] Z. Ren, H. Chen, J. Yu, L. Yang, S. Xiang, A consistent descriptor distance for shape matching and retrieval, Pattern Recognit. 96 (2019) 106987. https://doi.org/10.1016/j.patcog.2019.106987

[32] J. Gao, S. Huang, A. Kuijper, R. Klein, D.W. Fellner, Multimodal shape descriptor via graph learning for 3D shape matching, IEEE Trans. Image Process. 30 (2021) 3627–3640. https://doi.org/10.1109/TIP.2021.3071815

[33] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: Eur. Conf. Comput. Vis., Springer, Berlin, Heidelberg, 2010, pp. 356–369.

[34] W. Luo, Y. Li, R. Urtasun, R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, in: Adv. Neural Inf. Process. Syst. 29, 2016, pp. 4898–4906.

[35] Y. Huang, W. Chen, L. Peng, Y. Liu, M. Wang, X.P. Zhang, X. Zeng, LineDL: Processing images line-by-line with deep learning, IEEE Trans. Image Process. 32 (2023) 3150–3162.

[36] R. Zhang, P. Isola, A.A. Efros, Colorful image colorization, in: Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 649–666.

[37] J. Wang, K. Chen, R. Xu, Z. Liu, C.C. Loy, Deep high-resolution representation learning for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 43 (10) (2021) 3349–3364.

[38] M. Tan, Q.V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 6105–6114.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4510–4520.

[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[41] H.M. Ahmad, A. Rahimi, Deep learning methods for object detection in smart manufacturing: A survey, J. Manuf. Syst. 64 (2022) 181–196.

[42] S. Friedenthal, A. Moore, R. Steiner, A Practical Guide to SysML, 3rd ed., Morgan Kaufmann, Burlington, MA, 2014.

[43] Object Management Group, OMG Systems Modeling Language (SysML) Version 1.6, 2017. https://www.omg.org/spec/SysML/1.6/ (accessed [4/16/2026]).

[44] R. Stark, S. Kind, S. Neumeyer, Digital twin and virtual engineering: A review of technology, applications, and future trends, CIRP Ann. 68 (2019) 129–132. https://doi.org/10.1016/j.cirp.2019.04.038

[45] T. Weilkiens, Systems Engineering with SysML/UML, 3rd ed., Morgan Kaufmann, Burlington, MA, 2021.

[46] ISO/IEC, ISO/IEC CD TR 30138 Digital Twin Fidelity metric of digital twin system, available at: www.iso.org/standard/53258.html

[47] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3D shape recognition, in: Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 945–953.

[48] H. Zhang, Q. Qi, F. Tao, A consistency evaluation method for digital twin models, J. Manuf. Syst. 65 (2022) 158–168.

[49] W. Luo, A.G. Schwing, R. Urtasun, Efficient deep learning for stereo matching, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 5695–5703.

[50] H. Freeman, On the encoding of arbitrary geometric configurations, IRE Trans. Electron. Comput. EC-10 (1961) 260–268.

[51] M.K. Hu, Visual pattern recognition by moment invariants, IRE Trans. Inf. Theory 8 (1962) 179–187.

[52] T. Ojala, M. Pietikäinen, D. Harwood, Performance evaluation of texture measures with classification based on Kullback discrimination of distributions, in: Proc. 12th IAPR Int. Conf. Pattern Recognit., Jerusalem, Israel, 9–13 October 1994, pp. 582–585.

[53] M.J.D. Powell, Radial basis functions for multivariable interpolation: A review, in: J.C. Mason, M.G. Cox (Eds.), Algorithms for Approximation, Clarendon Press, Oxford, UK, 1987, pp. 143–167.

[54] H. Bunke, K. Riesen, Recent advances in graph-based pattern recognition with applications in document analysis, Pattern Recognit. 44 (5) (2011) 1057–1067.

[55] B. I. van Blokland, et al., ShapeBench: A new approach to benchmarking local 3D shape descriptors, *Comput. Graph.* 124 (2024) 104052.

[56] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2013, pp. 6655–6659.

[57] Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., & Zucker, S. W. (1999). Shock graphs and shape matching. *International Journal of Computer Vision, 35*(1), 13-32.

[58] T.H. Khan, C. Noh, S. Han, Correspondence measure: A review for the digital twin standardization, Int. J. Adv. Manuf. Technol. 128 (5) (2023) 1907–1927.

[59] T.A. Jauhar, S. Han, T.H. Khan, C. Noh, A. Lavrinec, An implementation of sensor network to assess near real-time digital twin fidelity, Cyber-Phys. Syst. (2026) 1–24.

[60] M.T.H. Khan, S. Han, T.A. Jauhar, C. Noh, A review of 3D shape descriptors for evaluating fidelity metrics in digital twin, Machines 13 (9) (2025) 750.

[61] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, in: Proc. 38th Int. Conf. Mach. Learn., PMLR 139, 2021, pp. 8748-8763.

[62] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: Int. Conf. Learn. Represent. (ICLR), 2021.

Appendix A. Supplementary Background on Deep Learning

A.1 How Deep Learning Works

A.2 Convolutional Neural Networks (CNNs)

At its core, deep learning refers to a class of machine learning techniques that use neural networks with many layers hence the term 'deep' to model complex patterns and relationships in data (LeCun et al., 2015; Kuhl & Giardina, 1982; Wang et al., 2021). A neural network consists of layers of interconnected nodes (neurons), each of which performs a mathematical operation on its inputs and passes the result to the next layer. In a typical neural network, these layers are categorized into:

1) Input Layer: Takes in raw data (e.g., image pixels, point clouds).

2) Hidden Layers: These intermediate layers perform feature extraction, transforming input data into higher-level representations through non-linear activation functions (Brigham 1988).

3) Output Layer: Provides the final prediction or classification based on the learned features.

Each layer in a deep neural network performs the following transformation:

$$h_{i+1} = f(w_i h_i + b_i) \quad (1)$$

Where:

- h_i represents the activations (or outputs) of layer i,

- w_i represents the weight matrix for layer i,

- b_i is the bias vector for layer i,

- f is the activation function (such as ReLU, Sigmoid, or Tanh), which introduces non-linearity to the model.

The power of deep learning comes from the ability of these networks to learn intricate patterns and hierarchical features from data, where each layer extracts increasingly abstract features.

A specific type of deep learning model, Convolutional Neural Networks (CNNs), is particularly effective for analyzing image data (Zhang et al., 2016). CNNs are designed to capture spatial hierarchies in data by applying convolution operations over the input. The convolution operation involves sliding a small filter (or kernel) across the input image to compute local features.

The core operations in a CNN include:

1) Convolution: Applying filters to the input to detect local patterns such as edges, corners, or textures.

$$S(i,j) = (I * K)(i,j) \sum_m \sum_n I(i+m, j+n).K(m,n) \quad (2)$$

Where, I is the input image, K is the kernel (filter), S is the output feature map (also known as an activation map), m and n represent the dimensions of the kernel.

2) Pooling: Reduces the spatial dimensions of the feature maps, thereby reducing computational load and controlling overfitting.

$$P(i,j) = max_{m,n} S(i+m, j+n) \quad (3)$$

3) Activation Functions: Apply non-linear transformations to the feature maps, enabling the network to learn more complex patterns.

$$f(x) = \max(0, x) \quad (4)$$

ReLU introduces non-linearity by setting all negative values to zero, allowing the network to model non-linear relationships.

4) Fully Connected Layers: After a series of convolutional and pooling layers, the output is often flattened into a vector and passed through fully connected layers that combine features and provide the final prediction.