Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 8674641, 33 pages
https://doi.org/10.1155/2023/8674641

Review Article

A State-of-the-Art Computer Vision Adopting Non-Euclidean Deep-Learning Models

Sakib H. Chowdhury, Md. Robius Sany, Md. Hafiz Ahamed, Sajal K. Das, Faisal Rahman Badal, Prangon Das, Zinat Tasneem, Md. Mehedi Hasan, Md. Robiul Islam, Md. Firoj Ali, Sarafat Hussain Abhi, Md. Manirul Islam, and Subrata Kumar Sarker

Department of Mechatronics Engineering, Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh

Correspondence should be addressed to Subrata Kumar Sarker; subrata@mte.ruet.ac.bd

Received 7 November 2022; Revised 17 April 2023; Accepted 18 April 2023; Published 9 May 2023

Academic Editor: B. B. Gupta

A distance metric known as non-Euclidean distance deviates from the laws of Euclidean geometry, which is the geometry that governs most physical spaces. It is utilized when Euclidean distance is inappropriate, for as when dealing with curved surfaces or spaces with complex topologies. The ability to apply deep learning techniques to non-Euclidean domains including graphs, manifolds, and point clouds is made possible by non-Euclidean deep learning. The use of non-Euclidean deep learning is rapidly expanding to study real-world datasets that are intrinsically non-Euclidean. Over the years, numerous novel techniques have been introduced, each with its benefits and drawbacks. This paper provides a categorized archive of non-Euclidean approaches used in computer vision up to this point. It starts by outlining the context, pertinent information, and the development of the field's history. Modern state-of-the-art methods have been described briefly and categorized by application fields. It also highlights the model's shortcomings in tables and graphs and shows different real-world applicability. Overall, this work contributes to a collective information and performance comparison that will help enhance non-Euclidean deep-learning research and development in the future.

1. Introduction

For decades, machine learning has been enriched in many dimensions in terms of rich data inputs, nobler algorithms, and output optimization. The computer vision's primary goal is to analyze, process, and give meaning to digital images. To achieve this feat, machine-learning (ML) algorithms as per different methods are being developed day by day [1]. For text-based models or 1D inputs, *K*-nearest neighbor (KNN) models are being used with proficiency [2, 3]. Text-based inputs are more straightforward as only one dimension is enough to assess features. Then, came the 2D or image-based algorithms which are segmented with grids to analyze. Convolutional neural network (CNN) [4], artificial neural network (ANN) [5], and recurrent neural network (RNN) [6] models are established as an almost saturated model for image processing [7–9]. Until this point, these were directed with structured parameters and traceable features. While 2D models are tamed, 3D models have been the main focus in the computer vision department for the last decade. Modern GPU-based computers' increasing processing power, the accessibility of the size of training datasets, and effective stochastic optimization techniques have all made it possible in recent years to design and successfully train complex network frameworks with numerous degrees of freedom. This sparked the field's growth by enabling deep neural networks to significantly improve productivity with a large range of applications, starting from processing speech and language for machine processing to image processing and computer vision.

3D models are more potent in terms of extracting analytics, simply containing more information. Analyzing 3D datasets became a necessity with the requirements of analyzing massive information from social media with many

parameters added with each node [10], observations of physical objects found in nature with archaeological values [11], analyzing the protein structures [12], 3D scanning in the medical department [13], and many more. Virtual reality is becoming more embedded in the world with each passing day, but still, it is in its infancy stage, with its usage only in the entertainment department. For practical embedding of virtual reality, flawless accuracy must be achieved in analyzing the 3D surroundings [14]. That said, 3D models come with non-Euclidean geometry, which brings a lot of difficulties while measuring features.

As 1D and 2D models are easy to organize and connectivity among nodes is prioritized, the CNN model can easily recognize and analyze the provided input patterns. These inputs are called Euclidean inputs that feature Euclidean geometry that can be defined with 2D shapes and figures following explainable mathematical rules. The main difference lies in Euclidean geometry being only on the same planes, whereas non-Euclidean geometry is the geometry in 3D places with infinite planes. Thus, conventional 2D geometry that has been used till now is useless in non-Euclidean geometry. That said, non-Euclidean geometry uses complex structures that hold more data. Euclidean geometry has only one plane to store data and, therefore, has less freedom for inputs. Existing models for 2D assessment are being upgraded for 3D analysis and unlocking their full potential to analyze data of multiple planes. Thus, new algorithms are being created to recognize the 3D analysis by non-Euclidean deep learning, also known as "geometric deep learning" [15], taking on a challenge to give structure to unstructured mesh grids, manifolds, and point clouds.

Data having a non-Euclidean spatial structure is of interest to many scientific disciplines. Social networks in the field of computational social sciences, sensor networks in information exchange, functional networks in neuroimaging, regulatory networks in genetics, and meshed surfaces in 3D modeling are just a few examples.

In social networking sites, user characteristics may be replicated using signals on the vertex points of the social graph. Distributed, interconnected sensors make up sensor networks, and their measurements are shown on graphs as time-varying signals. 3D objects are modeled in computer vision and graphics as Riemannian manifolds [16] with attributes like color, texture, and motion fields such as dynamic meshes. Since these data are non-Euclidean, it follows that they lack well-known characteristics like global parametrization, a standard set of coordinates, graph-based structure, and shift invariance. As a result, fundamental operations like linear combination and convolution, which are assumed to be clearly defined in the Euclidean context, are considerably less, so in non-Euclidean domains. This is a significant barrier that has prevented the application of effective deep-learning techniques, like convolution or recurrent neural networks, to non-Euclidean geometric data up to this point. As a result, fields like computer graphics and computational sociology have not yet experienced the practical and theoretical breakthroughs that deep-learning models have given to voice recognition, natural language, and computer vision. So the evolution of this deep learning begins.

Figure 1 describes practices of deep geometrical learning that started a while back starting with recursive neural networks (RvNN) in 1997 and are used on directed graphs [17]. A breakthrough in machine learning came in 1998 with the convolution of a neural network, which is further developed to assist in the creation of many more models for understanding 3D objects [18]. The reason for the possibility is local spatial feature extraction in multiscale. The first modified version of a graph neural network (GNN) [19] to calculate graph data started its journey in 2005 and fully came to attention in 2009 with improved performance using the SL algorithm [20]. Following that, the graph convolutional neural network (GCNN) [21] model was introduced to counter the difficulties of analyzing non-Euclidean manifolds. GCNN was followed by lookup-based CNN (LCNN) [22], which used a learned lexicon feature to encode CNN convolution. The diffusion CNN (DCNN) [23] also known as diffusion-convolutional neural network was the following method put forth to develop a diffusion-based model of nodes from network data to categorize them. A common technique for extracting the local feature from the graph was proposed in the work of GNN, which is comparable to convolutional networks that work based on images and on the inputs connected by local regions. ChebNet [24] was first proposed in 2016. Following that, an anisotropic CNN (ACNN) [25] model was introduced that added a shape factor to CNN for analyzing using shapes as a unit. PointNet [26] models that already existed were improved using the recursive structure the following year and established PointNet++ [27]. Afterward, GCN was suggested as a more specific variant. CayleyNet [28] was proposed a year later that used Cayley polynomials to existing GCN [29]. Two recently proposed models are anisotropic Chebyshev spectral CNN (ACSCNN) [30] which aggregates local feature values for effective signal collection in 2020 and UV-Net [31] which combines image data with GCN for low memory overhead cost and computational cost in 2021. Convolutional networks constituted the foundation for the research findings mentioned above for graph-based methods. Similarly, manifold or voxel-based algorithms saw their development over the years side by side using 3D CNN as the base model. Working inherently with geometric forms is commonplace in the field of computer graphics [32]. Here, 3D objects are often treated as Riemannian manifolds and discretized using meshes.

Many image-based machine-learning methods have been directly applied to 3D geometric data, with varying degrees of success, with the data being represented as range pictures [33, 34] or rasterized volumes [35, 36]. The primary problem with these methods is that they incorrectly assume that geometric data can be represented as Euclidean structures. To explain Figure 2, when dealing with complicated 3D objects, representations based on Euclidean geometry, such as depth pictures or voxels, may distort or even destroy the object's topological structure, resulting in a considerable loss of information. Second, when an item is

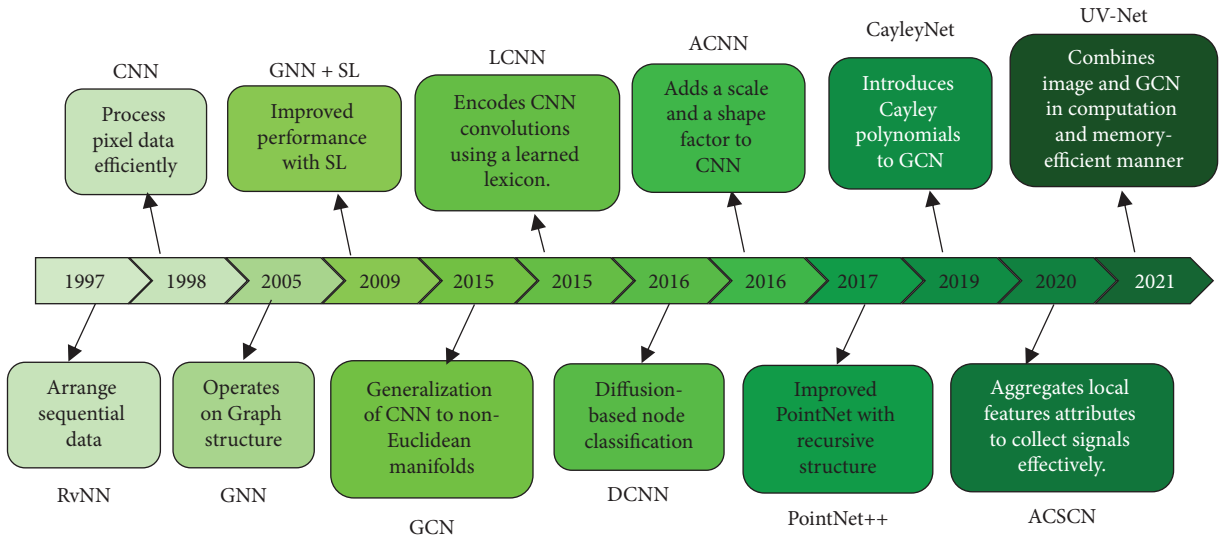


Figure 1: Evolution timeline of the graph convolutional network (GCN).

posed differently or deformed, its Euclidean representation will change [37].

There have been some reviews on geometric deep learning in the past few years. Bronstein et al. gave a sophisticated overview of this context. Quite a few general models with corresponding mathematical derivations with some wide extensive application in 2017 were described in detail [15]. Zhang et al. [38] reviewed the different types of deep-learning methods on graphs only [39, 40]. Later on, Cao et al. [41] reviewed some models with their mathematical derivations on graphs and manifolds.

This paper shifts the emphasis from a broad overview to computer vision exclusively, since the previous publications all concentrate on their models in general use cases across disciplines. Models like UV-Net [31] and MDGCN [42] presented more recently improved computer vision in an intuitive manner with effective visual representations for hobbyist computer vision researchers. In contrast to the cited publications, the strengths and weaknesses in terms of the performances of these models are discussed and summarized. Unlike previous works published in the modern age of computer vision, this one covers a broad range of cutting-edge applications. The sole contributions of this study are statistical findings, trends, obstacles, and recommendations for further research. This article provides a detailed examination of current deep-learning architectures in computer vision, together with an exhaustive account of their methodology and their applications.

Our contributions can be summarized as follows:

(i) This paper offered a quick summary of mostly used and efficient non-Euclidean deep-learning models suggested during the last two decades. In addition, it offered theoretical and operational analyses of the model. It has visualizations, explanations, and mathematical reasoning.

(ii) This literature categorized the models according to their underpinnings, such as spectral and spatial types. Newer hybrid frameworks, such as a spatial-spectral-based model, are also included. Each model was outlined together with its underlying logic and operational principles.

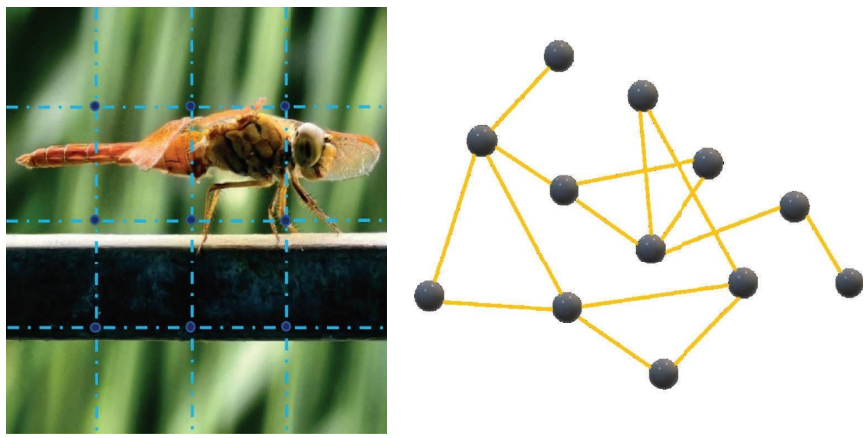
(iii) To fully comprehend these models, it has summarized their key features, such as their uniqueness and their limits, into a performance table. Various numerical findings demonstrating performance metrics on their respective datasets are also included.

(iv) To help put the spotlight on the current trend in this area, it has compiled a table summarizing the most recent uses of these graph-based frameworks in a variety of contexts.

(v) To emphasize the importance, it provides data from recent research showing the current trajectory and promising future of the field.

(vi) In light of these most recent work patterns and technical challenges, this paper analyzed potential future scopes.

This paper reviews the most recent deep-learning methods for computer vision on graphs with manifolds and the applications that traverse them. It started with the introduction in Section 1 of our article, where it gave a succinct history and evolution of the various models put forth over the years. It also discussed the importance and influence of our field in this section. The background research in the linked fields of the point cloud, graph, and manifold is detailed in Section 2, along with relevant theories. In Section 3, it has categorized every suggested methodology for non-Euclidean fields with simple math and descriptions. The models were divided into GCNs and manifolds, which were then divided into spectral and spatial subcategories. Section 4 is devoted to outlining several current algorithmic limitations, applications, and prospective future opportunities for the progress of our



(a) (b)

Figure 2: (a) Image in the Euclidean space and (b) graph data in the non-Euclidean space.

profession. In Section 5, it closes with a summary and nobility of our paper.

2. Background

Geometric deep-learning models use point clouds, shapes, graphs, and manifolds whose mathematical operations are described.

2.1. Point Cloud. Commonly used ways to analyze 3D objects are to use a laser scanner to gather a lot of data as a point cloud. Providing them inputs, analysis becomes much more complicated as all the nodes are undirected and independent [43]. The points are non-Euclidean and require feature segmentation for further calculation. Object recognition and detection take point clouds as inputs in computer vision tasks due to their availability.

In point clouds, a single point from the point cloud P can be defined as $R - (P_x, P_y, P_z) \in R^3$, where R is the reference point and P_x, P_y, P_z represent the 3D position of the point P [44]. As R^3 represents the three-dimensional space, navigating between two points is performed by radial rotation from a fixed point. The connecting line is defined by r, θ, in which r is the distance between two locations and θ represents the angle between the axes.

While the point cloud needs a reference point to be located and used, the mesh feature allows the points to be referred through another point. Mesh form is built by point-by-point addition and reduction of some vertices by compression.

The data's complexity necessitates using a B-spline curve for this kind of mathematical manipulation, as stated above. In this part, the B-spline curve will be discussed.

2.1.1. B-Spline Curve. Approximate 3D segmentation comes with a problem of denoting features as fixed or defined as on continuous curvature surfaces, and there are no reference points [45]. To counter the problem, the B-spline curve is presented, which is visualized in Figure 3. At the maximum

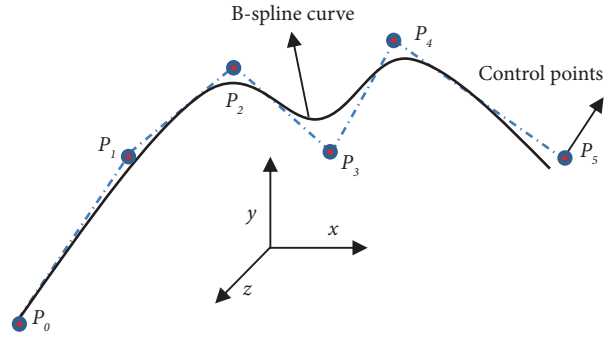


Figure 3: B-spline curve.

bending portion, a V-shaped joint is provided called control points, whereas the corresponding points on the curve are presented as knot vectors [46]. The control points can be used as the reference point, and local features can be obtained.

The B-spline curve of degree d in R^3 is exemplified by [47]:

$$H(t) = \sum_{i=0}^{n} S_{i,d}(t).C_i, \tag{1}$$

where B_t, d are the B-spline basis function defined on the knot vectors and functions on knot vectors are denoted as $S_{i,d}$ [47]:

$$V = \{0, \ldots, 0, v_{d+1}, \ldots, v_n, 1, \ldots, 1\}, \tag{2}$$

where similar end-to-end points are denoted as $d + 1$; the control point [48] $P(t)$ is exemplified by

$$C = \{C_i R^2, i = 0, \ldots, n\}. \tag{3}$$

A B-spline curvature fits only on a fixed knot vector. Assuming for any point, when cloud $Z = \{Z_k, k = 0, \ldots, N\}$, the B-spline curve $P(t)$ is [48]:

$$\min C\left[\frac{1}{N+1}\sum_{k=0}^{N} d^2(Z_k, P(t)) + \frac{1}{n+1}\lambda.f_s\right]. \tag{4}$$

The minimal distance between Z_k and $P(t)$ is stated as $d(Z_k, P(t))$, in which f_s is the term for regularization and λ is the error coefficient. This regularization is defined as in [49]:

$$f_s = \alpha \int_0^1 \left\| P^{'}(t) \right\| \mathrm{d}t + (1-\alpha) \int_0^1 \left\| P^{''}(t) \right\|^2 \mathrm{d}t. \tag{5}$$

For $0 \le \alpha \le 1$, multiplicands of α and $(1-\alpha)$ are denoted as f_1 and f_2. Assuming $P(t_k)$ to be the projection point on $P(t)$ of Z_k, the local Frenet frame on $P(t)$ at $C(t_k)$ is $\{T_k, N_k\}$ given that T_k is unit tangential and N_k is a unit normal vector of curvature $P(t_k)$.

Let $W_k = Z_k - \mathrm{P}(\mathrm{t_k})$. The quadratic approximation for $d^2(Z_k, P(t))$ is given by [48]:

$$d^2(Z_k, P(t)) \approx \gamma_k . \left({W_k}^T T_k \right)^2 + \left({W_k}^T N_k \right)^2. \tag{6}$$

Thus, the local quadratic model [47] is

$$F = \frac{1}{N+1} \sum_{k=0}^{N} \left(\gamma_k . \left({W_k}^T T_k \right)^2 + \left({W_k}^T N_k \right)^2 \right) + \frac{1}{n+1} \lambda f_s. \tag{7}$$

The approximation of F matches to the Gauss–Newton method for $\gamma_k = 0$, and the approximation leads to intrinsic parametrization for $\gamma_k = 1$.

2.2. Graph Theory. Graphs have been used for a long time for analyzing and enhancing grid inputs from pictures. For 2D pictures, grid inputs are directed graphs with components of link with direction [50]. Being successful in 2D analysis, 3D observation comes with a few difficulties to handle undirected graphs and different types of input of mesh grids. The undirected graphs are structured by using the discrete Laplacian and Fourier transform using eigenvectors to add local features.

Graph theories come with these challenges while analyzing:

(a) Node classification
(b) Graph categorization
(c) Clustering of nodes
(d) Prediction of link
(e) Influence maximization

A graph [51] is defined by $G(V, E)$ given that V and E represent vertices and edges, respectively. We assume $v_i \in V$ and $a_{ij} = (v_i, v_j) \in E$. Here, v_i denotes the nodes and e_{ij} denotes edges among v_i and v_j. For $N = |V|$, the adjacency matrix A is given by an $N \times N$ matrix stated as edge weights. Here, $A_{ij} = a_{ij} > 0$; $e_{ij} \in \mathrm{E}$ and $A_{ij} = 0$; $e_{ij} \notin E$. Graphs can be of two types: directed and undirected, based on the types of edges. The edges are connected with a direction for directed graphs, and undirected graphs have connected edges without a direction. Thus, $A_{ij} = A_{ji} = 1$ for the undirected and $A_{ij} \neq A_{ji}$ for the directed graphs.

By analyzing the eigenvalues of the graph Laplacian matrix, spectral graph theory may provide insight into whether a graph is linked and the quality of that connection. The Fourier transform is utilized for eigendecomposition or the breakdown of a matrix into its parts for the graph's Laplacian matrix. The term "convolution" is used to describe the process of multiplying the input neurons by a set of weights, sometimes called "filters" or "kernels," in a graph. In this part, discrete Laplacian, Fourier transform, and convolutional operations on the graph will be discussed.

2.2.1. Discrete Laplacian on Graph Theory. Discrete Laplacian (also known as the Laplacian matrix) is a spectral graph machine-learning algorithm. The Laplacian matrix allows the creation of a link between discrete inputs of graphs and manifolds. The function provides a mathematically tractable solution to graph localization limitations. The eigenvalues at the adjacency matrix are an essential factor in localizing two graph vectors as similar graphs provide the same eigenvalues [52]. Thus, graph Laplacian is a must to understand the undirected graph.

The Laplacian matrix is given by $L = D - \mathrm{A}$. The degree matrix is denoted as D such that $D_{ii} = \sum_j A_{ij}$. The Laplacian matrix can be of three forms [53]:

$$\begin{aligned} \text{combinatorial: } & L = D - A, \\ \text{symmetric normalized: } & L^{\mathrm{sym}} = D^{-1/2} L D^{-1/2} \\ & \quad = I - D^{-1/2} A D^{-1/2}, \\ \text{rando m walk normalized: } & L^{rw} = D^{-1} L = I - D^{-1} A. \end{aligned} \tag{8}$$

Assuming a graph G that is undirected and straightforward, the adjacency matrix will only contain 1 and 0. Thus, the value of L is given by [53]:

$$L_{ij} = \begin{cases} \deg(v_i); i = j, \\ -1; i \neq j, v_i \text{ is adj acent to } v_j, \\ 0; \text{else}, \end{cases}$$

$$L_{ij}^{\mathrm{sym}} = \begin{cases} 1; i = j, \deg\ (v_i) \neq 0, \\ \dfrac{-1}{\sqrt{\deg(v_i) \deg\ (v_j)}} \quad ; i \neq j, v_i \text{ is adj acent to } v_j, \\ 0; \text{else}, \end{cases}$$

$$L^{rw} = \begin{cases} 1; i = j, \deg\ (v_i) \neq 0, \\ \dfrac{(-1)}{\deg\ (v_i)}; i \neq j, v_i \text{ is adj acent to } v_j, \\ 0; \text{else}, \end{cases} \tag{9}$$

where $\deg(v_i)$ denotes the degree of the node i. The different forms of the Laplacian matrix are given by the type of the degree of the node.

2.2.2. Graph Fourier Transform. Fourier analysis on the graph is possible as eigenvectors of the Laplacian matrix represent similar values on the Fourier basis. The Fourier transform and its reverse enable a node to be present in two different scales [54]. Given that for any N no of nonnegative, mutually orthogonal, and independent eigenvalues, the graph's Laplacian matrix may be expressed as in [55]:

$$L = U \wedge U^{-1}. \tag{10}$$

Given that λ is the eigenvalue and $\wedge$ is the eigenvalue matrix such that $\wedge_{ij}. = \lambda_i;\ 0 \le \lambda_1 \le \lambda_{2\ldots} \le \lambda_{N-1}$, the significance of the graph's vertices is represented via the values of λ. Also, $U = [u_0, u_1 \ldots, u_{N-1}] \in R_{N\times N}$ and is a matrix of eigenvectors. Since U is orthogonal, it can be written as $U^{T-1} = U^T$. Thus, L will be [54]:

$$L = U \wedge U^T. \tag{11}$$

For using the eigenvectors in the Fourier transform, the eigenfunction must the transformed via the basis function $e^{-\mathrm{iwt}}$. For any graph signal, $g \in R^N$ such that g_i is denoted as the value of i^{th} node.

The Fourier transform can be expressed as $\hat{\mathrm{g}} = \mathrm{U}^{\mathrm{T}}$ and the inverse Fourier transform as $g = U\,\hat{\mathrm{g}}$. Here, eigenvectors are used as the basis, and the transform projects the input graphs on the orthogonal space.

2.2.3. Convolutional Operations on Graphs. Using the Fourier transform of a graph in the frequency domain, the undirected graph becomes well-directed and component-wise multiplication transforms. Convolution requires a filter to find out the compact on convolution layers. Now, for any input b and using a filter g, the result is [55] as follows:

$$b * g = U\hat{\mathrm{g}}(\wedge)\mathrm{U}^{\mathrm{T}}\mathrm{b} = \hat{\mathrm{g}}(\mathrm{L})\mathrm{b}. \tag{12}$$

Although the filters bring complexity with the number of nodes and do not clarify a lot, we use parametrization of $\hat{\mathrm{g}}$. Thus, the resultant is [55] as follows:

$$\hat{\mathrm{g}}_\theta(L) = \sum_{d=0}^{D-1} \theta_d L^d, \tag{13}$$

where $\hat{\mathrm{g}}_\theta$ is the polynomial parametrization, whereas $\hat{\mathrm{g}}$ has a degree of D. The complexity becomes clearer as $\hat{\mathrm{g}}$ is D-localized with known relation $\theta_0, \ldots, \theta_{\{D-1\}}$.

2.3. Manifold Geometry. A manifold can be defined as curved, but locally, it can be seen as flat, as in Figure 4. Thus, although a manifold is a non-Euclidean shape, it can be treated locally as a Euclidean model.

Considering a manifold of the topological space $\mathbf{M}$ with the dimension number of $\mathbf{n}$, Also, let $\mathbf{m_1}, \mathbf{m_2} \in \mathbf{M}$ such that $\mathbf{m_1}$ and $\mathbf{m_2}$ are neighboring points. The inner product of the tangent space is denoted by $(.,.)$: $\mathbf{T_{m1}M} \times \mathbf{T_{m2}M} \in \mathbf{R}$ given that $\mathbf{T_m}$ represents the tangential space of the Euclidean part and $\mathbf{R}$ is an abstract manifold capable of comparable measurements. For describing 3D data objects, computer vision takes two neighboring surfaces as input and embeds them in the $\mathbf{R^3}$ space. Although the model is not completely taken as 3D, the completion of embedding creates a 3D shape to analyze.

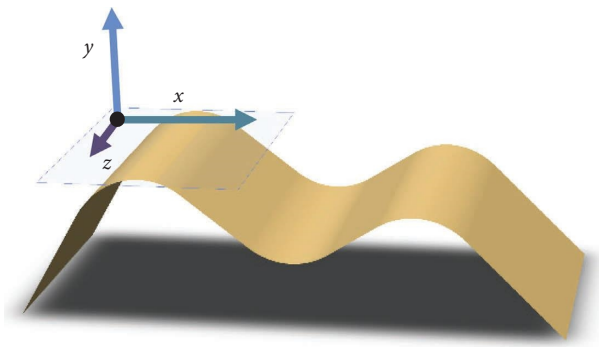


Figure 4: Manifold curve with a local plane.

2.3.1. Calculus Operation Manifold. Being manifold as non-Euclidean surfaces, calculus cannot be performed directly as declaring variables is not possible. However, various methods have been present to create smooth surfaces on a manifold that is known as a differentiable manifold. A differential manifold is just a space on a manifold where calculus can be performed. That said, manifold data can be segmented into many more spaces, and the calculus function must be applied to all of them. Thus, the difference between normal and manifolds is normal calculus works on only one dimension, whereas manifold calculus is structured with higher dimensions to adapt to multiple spaces.

We state that $f: M \longrightarrow R$, such that f is a smooth function of manifolds defined in the scalar field, where the mapping function is given by $M \longrightarrow TM$. $F(x) \in T_xM$, such that $F(x)$ is a tangent vector at the point x. Considering Hilbert space fields of $L^2(M)$ as a scalar field and $L^2(TM)$ as a vector field, thus multiplication results [56] are

$$\begin{aligned}(f,g)_{L^2(M)} &= \int f(x)g(x)\mathrm{d}x,\\ (F,G)_{L^2(TM)} &= \int (F(x),G(x))_{T_xM}\mathrm{d}x.\end{aligned} \tag{14}$$

With dx being the area element, differentiating f results in $df: TM \longrightarrow R$. Again, differentiation on the closest neighbor points is defined as $df(x) = (\nabla f(x),.)_{T_xM}$. Now, applying the operation to the tangent vector, $df(x) = (\nabla f(x), F(x))_{T_xM}$. Now, a small displacement of x results in $\nabla f: L^2(M) \longrightarrow L^2(TM)$ which is the gradient operator. Thus, divergence [56] becomes div: $L^2(TM) \longrightarrow L^2(M)$:

$$(F, \nabla f)_{L^2(TM)} = (-\mathrm{div}\,F, f)_{L^2(M)}. \tag{15}$$

The relation between the gradient and divergence is as follows [56]:

$$\Delta f = -\mathrm{div}(\nabla f). \tag{16}$$

Laplacian is found symmetric as in [56]:

$$(\nabla f, \nabla f)_{L^2(TM)} = (-\mathrm{div}(\nabla f), f)_{L^2M} = (f, \Delta f)_{L^2M}. \tag{17}$$

For compact manifolds, the functions of x work similarly to the spectral analysis for graphs. Also, it can be seen that Laplacian stands as a vital part of analyzing non-Euclidean space.

2.3.2. Discretization of Manifold. Discretization is required for data conversion, such as from point clouds to manifolds. For discretization, a manifold [57] can be sampled by N points, where the positions of nodes are stated as $x_1, x_2, \ldots, x_N$, thus creating a graph accordingly. The undirected graphs can be difficult to pose for the increase in nodes of a unit area. Again, the surface can be created as the mesh (V, E, F), where V is the vertex, E is the edge, and F is the triangular face [57]: $V = \{1, \ldots, N\}$, $ijk, ijh \in F$, and $\{(i, j), (i, k), (j, k)\} \in E$. The triangular mesh must have the designated manifold boundary, and the edge length should be $l_{ij} > 0$, satisfying the inequity of a triangle. For $l_{ij} = \|x_i - x_j\|_2$, cotangent weights are given by $w_{ij} = 1/2\,(\cot \alpha_{ij} + \cot \beta_{ij})$ for the manifold's Laplacian mesh.

3. Methods

Because of our familiarity with the background, we can now analyze many models, each of which uses non-Euclidean geometric data for a specific reason. This geometry, as said previously, can be characterized by graphs as well as manifolds. In network structures like social media or in general, it learns embedding that integrates knowledge about its surroundings [58]. Graphs are employed in these network structures. In addition, manifolds are used in the process of three-dimensional form, as well as on various complicated contour surfaces and in model analysis. Convolutional neural networks (CNNs) are the foundation, on which the vast majority of these models are typically built, as shown in Figure 5.

The primary applications of this network include image processing, classification, and segmentation, in addition to the processing of various types of autocorrelated data, although it contains GANs and GGNs along with GAEs, respectively.

3.1. Methods Based on Graph Convolutional Networks (GCNs). In the actual world, graphs are the most popular form of data organization, so to deal with it, GCNs fit very well in a situation like social media connection analysis, protein model analysis, and traffic control. Every image and video can be presented as some grids or grid structure data. To manipulate it, we need the solution of a graph convolutional network. In modern technology, deep learning greatly impacts various developments like games, not just social networks. Image analysis is a testament to the effectiveness of deep learning, and it is effective in computer vision as well. Recently, researchers have been trying to evolve the architecture based on the graph using some traditional models like the CNN, long short-term memory (LSTM), attention mechanism (AM), and autoencoder (AE) for more efficient performance.

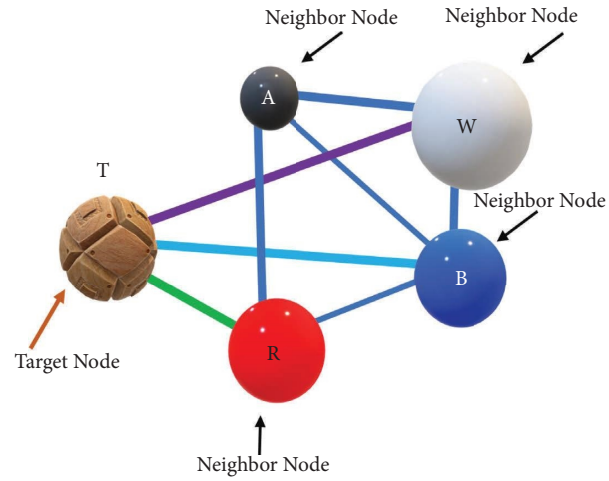


Figure 5: Input graphs of GCNs.

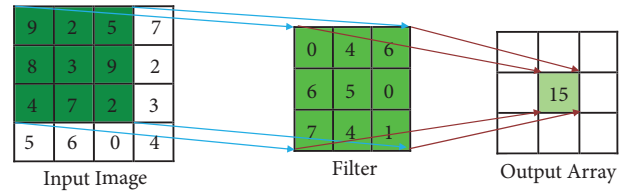


Figure 6: Convolutional layer in traditional convolutional neural networks.

However, it incorporates some problems. As images could be of different types, their complexities also vary. As the image contains more complex data (i.e., different lighting conditions, light, shades, and complex contour structures in various colors of light), it becomes more difficult to extract the actual shape from the image. Technically, nodes of the graphs may vary in large numbers. As a result, the convolution operation is very likely difficult in this situation, and this includes another problem, like data size, as the nodes of the graphs vary. This introduces new rising problems each time to these algorithms.

Now, this paper will classify this graph-based model into three different parts according to their basis of analysis or method of working:

(i) Spectral-based GNN

(ii) Spatial-based GNN

(iii) Spatial and spectral-based

Researchers try to conclude a common form of convolutional network that may work in any scenario. The main theme of this network is shown in Figure 6. It tries to make an input graph from the image, set the target, learn from its neighbor node, and aggregate those to form usable data like in Figure 7.

This GCN is divided into two kinds. Graph signal processing is inspired by spectral graph theory, which is the foundation of graph convolution [59]. Alternately, spatial domain convolution is the second.

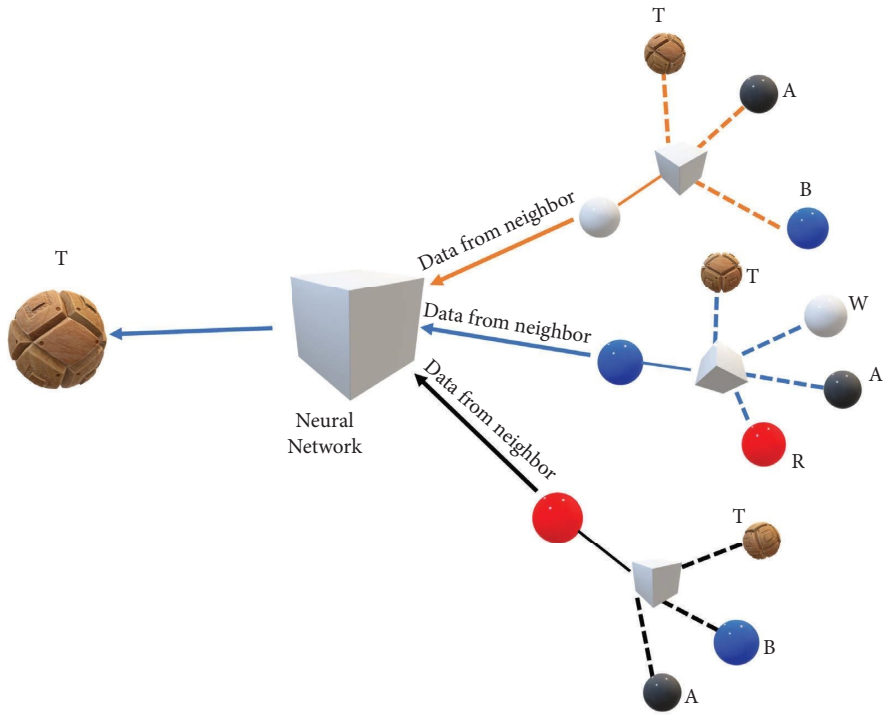


Figure 7: Architecture of graph networks.

Based on the context, it expressed the graph Fourier Transform and its inverse equation. The equations distinguish between spectral and special domains of a graph. $g = U\hat{g}$ represents the spatial domain, while $\hat{g} = U\hat{T}$ indicates the spectral domain. Consequently, if a signal g is used in the spatial domain, $\hat{g}$ is likewise significant in the spectral domain. These signals are typically referred to as kernels, and the Fourier coefficient of a graph often decays fast. The signal is compressible since its Fourier coefficients may be calculated from a few graph coefficients.

3.1.1. Spectral-Based GCNs. The spectral-based GCN model that has been constructed cannot be applied directly to graphs; nevertheless, because of its effective feature extraction capacity, it may be highly beneficial to extract features [60]. It is possible to define non-Euclidean convolution using it, and by analogy, it may be used to describe the relation in terms of the frequency domain. In recent times, the graph Laplacian matrix has been used directly because its convolutional layer architecture is so effective.

(1) Spectral CNN. The "specifying" architectural cluster in spectral CNNs includes additional inputs. A vector representing the network's most recent output distribution is sent to the specified input for each training dataset on each training step. The propagation of this input then proceeds in a conventional feed-forward manner, with the specification of a cluster and network layer at the conclusion. This extracluster is likewise subject to a learning rule. This strategy comes in a variety of architectural forms: using a single-layer structure or a multilayer structure, we link the outputs of the specified cluster with the output layer and a hidden level of the network [61]. With the use of this model, convolution filters are altered to provide greater optimization capabilities via complex-coefficient spectral parameterization. Competitive outcomes on classification and approximation tasks were accomplished without the need for dropout or max pooling thanks to a more recent method of randomized change of resolution.

(2) CayleyNets. The CayleyNet model is a modified model of ChebNet [24]. ChebNet uses Chebyshev filters that avoid expensive computation without using eigenvectors. The main drawback of the model is that it cannot produce narrow-band filters. It occurs when there are eigenvalues clustered around minimal frequencies and the spectral gap is high [62]. CayleyNets add a new type of filter that takes the simplicity of the Chebyshev filters and can also produce narrow-band filters to counter disadvantages. The real value of a complex function is determined as the Cayley polynomial of order r [24]:

$$g_{v,z}(\lambda) = v_0 + 2Re\left\{\sum_{j=1}^{r} v_j (z\lambda - i)^j (z\lambda + i)^{-j}\right\}, \quad (18)$$

where $v = (v_0, v_1, \ldots, v_r)$ is given as a vector for a single real coefficient and r is the coefficient of complex. $Z > 0$ is denoted as the zoom parameter. For a real signal f, the Cayley filter is given as G_f and defined by [24]:

$$\begin{aligned} G_f &= g_{v,z}(\Delta) f \\ &= v_0 f + 2Re\left\{\sum_{j=1}^{r} v_j (z\Delta - iI)^j (z\Delta) + iI)^{-j} f\right\}. \end{aligned} \quad (19)$$

Parameters of v and z are optimized in training. The filter works with the basic calculation of matrix operations similar

to ChebNet. Thus, no eigendecomposition is required for the G_f filter to work. Cayley filters are based on the rational function of Laplacian and named ARMA filters. As general ARMA filters require matrix inversion, there is no way to ensure stable inversion as the training path is unknown. The Cayley filters to ensure inversion are stable. Also, the general ARMA filter uses a larger number of parameters which are overfitting for the objective, whereas the Cayley filter uses a moderate number of parameters.

The model presents a new class of extremely regular, localized, complicated rational Cayley filters that can represent any smooth spectral transfer function. The fundamental characteristic of the model is its ability to maintain localization in the spatial domain while specializing in narrow frequency bands with a limited number of filter parameters.

(3). UV-Net (Boundary Representations). U and V parameters of curves and surfaces are clearly expressed, while an adjacency graph explicitly defines topology in a boundary-representation data model. That happens when the user combines convolutional neural networks with image processing to create UV-Net [31], a network that both memories and computes economically.

Numerous topological components, such as faces, edges, half-edges, vertices, and their connections, compose the boundary-representation data model. With a few clicks, it extracts the most critical geometric and topological data from the boundary-representation and transforms it into a format suitable for current neural network architectures [31].

The UV-Net representation offers some benefits:

(1) For both primitive and geometric surface types, curve assessment of parameters can be applied quickly and easily [31]

(2) The representation is sparse and proportional to the number of B-rep contour surfaces

(3) The grid is mostly independent of precise parametrization

Using graph convolutions, local curves as well as surface characteristics are conveyed over the whole boundary representation. Curve and surface convolution is performed by taking 2D UV-grids. For message passing, contour CNNs are hidden features considered input edges and node features of the GNN. We calculate the hidden node features $g_v(h)$ in the graph layer $h \in 1 \cdots H$ by combining all the input features of the node $g_v(h-1)$ from a one-hop neighborhood $u \in A(v)$ while conditioning them on the edge features [31] $g_{uv}(h-1)$:

$$g_v^{(h)} = \alpha^{(h)}\left[\left(1 + e^{(h)}\right)g_v^{(h-1)} + \sum_{u \in A(v)} \left(l_\theta\left(g_{uv}^{(h-1)}\right) \odot g_u^{(h-1)}\right)\right]. \tag{20}$$

Here, $\alpha^{(h)}$ is an MLP, or in other words, multilayer perceptron along with two FC (fully connected) layers, $e^{(h)}$ is a parameter to differentiate the center nodes from the neighbors, and l_θ represents linear projection from the edge to the node feature space [31]. End-point features influence a concealed edge feature. The following recursive model underpins this learning process [31]:

$$g_{uv}^{(h)} = \beta^{(h)}\left[\left(1 + f^{(h)}\right)g_{uv}^{(h-1)} + l_{\theta'}\left(g_u^{(h-1)} + g_v^{(h-1)}\right)\right], \tag{21}$$

where $\beta^{(h)}$ is also an MLP having 2 layers. Final shape embedding is obtained by projecting (linearly) these characteristics into 128 D vectors and summing them [31]:

$$g_X = \sum_{h=1}^{H} y^{(h)}.g^{(h)} + c^{(h)}. \tag{22}$$

The model utilizes existing image and graph convolutional neural networks and can operate on B-rep data. On both supervised and self-supervised tasks spanning five B-rep datasets, advantages and adaptability are demonstrated, outperforming other representations such as point clouds, voxels, and meshes. A fresh synthetic B-rep dataset with differences in geometry and topology was once more introduced as SolidLetters.

(4) CurvaNet. Analyzing 2D images with the regular grid is much less challenging than 3D images using mesh surfaces or manifold input. Although traditional GNN models segment and use smaller surfaces, considering them flat, the model cannot differentiate higher surface changes due to a lack of data. CurvaNet modifies the GNN model by integrating differential geometry [63]. Data accuracy is ensured by downsampling by mesh pooling and upsampling by unpooling operation using an encoder and decoder. Thus, minimization of classification error is done by considering more input properties. The architecture is quite similar to the U-Net model, which runs through the curvature filter (CF) and graph convolution filter (GC) for segmentation. Skip connection is used to preserve precise boundaries [64].

A directional curvature filter negates fixed curvature limitations such as data loss and underfits or overfits. Addressing weight parameter direction is easier by segmenting all directions with many tangent vectors. The vectors must have a unique origin to point out null values. To ensure fixed parameters are provided for pool rotation for different angles. Graph convolution layers are used for sampling a neighborhood using the graph Laplacian matrix. ChebyNet24 and graph attention network (GAT) 56 are used for the function. Afterward, the properties of curvature are conserved by downsampling and upsampling. Let σ be the nonlinear activation function and b and d be the shared kernels. The feature matrix of curvature yields C, where k is the interval number and m is the maximum number [63]:

$$\begin{aligned} c_i^{(k)} &= \sigma\left(c_i^T A_i^{(k)} b^{(k)} + d^{(k)}\right), \\ c_i &= \max\ \left(c_i^{(1)}, \ldots, c_i^{(k)}, \ldots, c_i^{(m)}\right). \end{aligned} \tag{23}$$

To learn the directional curvature features at each vertex on a mesh surface, the model offers a unique convolutional filter. The mesh surface's curvature features are sent using graph convolutional methods. A U-Net-like hierarchical structure that downsamples and upsamples a mesh surface

dependent on mesh simplification was presented to make use of multiscale curvature features.

(5). Anisotropic Chebyshev Spectral CNNs (ACSCNNs). Anisotropic Chebyshev spectral CNN [30] is a new shape correspondence architecture based on manifold convolution. Extended convolution operators combine local signal characteristics by a series of directed kernels around each point, capturing additional signal information. Based on multiple anisotropic Laplace–Beltrami operator (LBO) eigendecomposition, spectral filtering is used to train kernels. To decrease computing difficulties, trainable Chebyshev polynomial expansion coefficients are used to represent spectrum filters [30].

The manifold X [30] is defined by

$$M^2(X) = \left\{ g\colon X \longrightarrow \mathbb{R}, \int g(x)^2 \mathrm{d}x < \infty \right\}, \tag{24}$$

and dx is the area element.

Vallet and Lévy [65] observed that the eigenvalues and eigenfunctions of the LBO are comparable to the frequency as well as Fourier basis in the Euclidean space. The LBO can be described [30] as

$$\Delta_X g(x) = -\mathrm{div}_X(\nabla_X g(x)). \tag{25}$$

The inner product $\hat{g}(\gamma_p) = < g, \sigma_p >_X$ is known as the Fourier transform (coefficient) for manifolds, because the eigenvalues and eigenfunctions of LBO have periodic properties. Its inverse Fourier transform for the manifold $g \in M^2(x)$ can be expressed [30] as

$$\begin{aligned} g(x) &= \sum_{p \geq 0} < g, \sigma_p >_X \sigma_p(x) \\ &= \sum_{p \geq 0} \hat{g}(\gamma_p)\sigma_p(x). \end{aligned} \tag{26}$$

To define the convolution theorem based on manifolds [30],

$$(g * h)(x) = \sum_{\hat{g}(\gamma_p)} \hat{g}(\gamma_p)\hat{h}(\gamma_p)\sigma_p(x). \tag{27}$$

Anisotropic LBO [66] can be defined as

$$\Delta_X g(x) = -\mathrm{div}_X(\mathbf{T}(x)\nabla_X g(x)). \tag{28}$$

As illustrated in the following model, ALBO is redefined [66] as

$$\Delta_{\beta\lambda} g(x) = -\mathrm{div}_X\left(\mathbf{C}_{\boldsymbol{\lambda}}\mathbf{T}_\beta(x)\mathbf{C}_\lambda{}^{Q}\Delta_X g(x)\right). \tag{29}$$

Here, $\mathbf{C}_\lambda$ is a revolution about a surface with a normal angle λ on the direction of the tangent, $\mathbf{T}(x)$ is a thermal conductivity tensor, and parameter β controls the anisotropic level [30].

Instead of employing anisotropic heat kernels in [25], it aims to learn kernels that depend on tasks by learning their parameterized filters $\hat{h}(\gamma)$. As stated in [24, 67], a polynomial filter may be used to solve these problems. Chebyshev polynomials are adopted to the filter $\hat{h}(\gamma)$, and due to its properly functioning, repetitive relation eliminates the eigendecomposition of ALBO and learning becomes easier [24, 68]. If the Chebyshev polynomial is $L_n(\gamma)$ of order n, the filter $\hat{h}(\gamma)$ can be expressed [30]:

$$\hat{h}(\gamma) = \sum_{n=0}^{n-1} b_n L_n(\gamma). \tag{30}$$

An extension of the manifold convolution operator, the model suggests the anisotropic convolution operator. Due to its direction-based consideration, this sort of anisotropic convolution enables a more thorough capture of the intrinsic local information of signals when compared to earlier works. To simplify the computation, Chebyshev polynomials are used to express the filters with trainable coefficients. In certain cases, the achieved outcome was superior to that of the earlier models.

3.1.2. Spatial-Based GCN. Unlike the spectral base, this convolution can be used directly because the kernel size is fixed. So it must need to select the neighbor of concern to be convoluted in a traditional manner. It uses pseudocoordinates by the filter function. These pseudocoordinates accumulated at the time of convolution. The most difficult aspect of developing CNNs that function with core nodes that have a variety of neighboring nodes is establishing local invariance for such CNNs.

The first intrinsic version of CNNs was introduced by Masci et al. [16]. Then, evolution began by Boscaini et al. [25] by introducing anisotropic heat kernels. The more general framework (MoNet) was then introduced by Monti et al. [69] to develop the deep convolutional architecture on graphs and manifolds. Then, the B-spline-based filter was introduced by Fey et al. [70], which works quite efficiently in the input of arbitrary dimensionality.

(1) Diffusion CNN (DCNN). The motivation for diffusion CNN is that a form encompassing graph diffusion can serve as a more reliable foundation for forecasting than a graph alone. A simple method for including contextual information about things that are calculated in polynomial time and effectively used on the GPU is provided by graph diffusion, which may be repressed. Many methods, such as probabilistic structural models and kernel methods, incorporate depth information in classification tasks; DCNNs offer a supplementary strategy that significantly improves predictive performance at node classifications [23]. When performing node classification tasks, diffusion-convolutional neural networks outperform probabilistic relational models and kernel approaches thanks to the representation that captures the effects of graph diffusion. Diffusion processes perform a good job of representing nodes, but they are ineffective in summarizing complete graphs.

(2) Graph Neural Network (GNN). The graph neural network is a supervised neural architecture that works well in terms of the graph and node-based applications. Two existing

concepts are combined into a single framework by this model. The neural network model will be referred to as the GNN. Both random walk models and recursive neural networks are shown to be extensions of the GNN and to retain their features. The model expands repetitive neural networks because it can handle node-focused tasks with no processing stage and can analyze a wider range of graphs, covering cyclic, oriented, and undirected graphs. The method broadens the range of processes that may be described and adds a learning mechanism to random walk theory [19].

The model offers a cutting-edge neural network architecture that can handle inputs from cyclic, directed, and undirected graphs or a combination of these. The diffusion of information and relaxation mechanisms are the model's foundations. Analysis of the outcome shows that the strategy is also appropriate for huge datasets.

(3) GraphSAGE. To accomplish the objective of node categorization, GraphSAGE models fully utilize the attribute information, structure, and knowledge of nodes in social networks, as well as mine the implicit mutual information among nodes. The best performance of a graph neural network may be comparable to that of the graph WL ISOMORPHISM test when the data structure (update feature plus aggregate feature) in graph networks is singular. An enhanced GraphSage [71] method is used to create the model for learning about GraphSAGE.

The model offers a revolutionary method that makes it possible to effectively create embeddings for invisible nodes. GraphSAGE successfully balances performance and runtime via sampling node neighborhoods, regularly outperforms state-of-the-art baselines, and offers a theoretical analysis that sheds light on understanding local graph structures.

(4) Large-Scale Graph Convolution Network (LGCN). The algorithmic structure cooptimization to speed up large-scale GCN inference on FPGA is provided to combat the significant expense of external storage access when evaluating the graph-structured dataset. To comply with on-chip storage restriction, first, data splitting is executed. Then, to decrease computational complexity and improve data locality, a two-phase preprocessing approach is created. The main computational kernels are mapped on an FPGA during hardware design, and data transmission for pipelined execution occurs through on-chip memory [72]. Varied GCN architectures and various analytic orders are supported by the data path.

The suggested architecture allows for the application of standard convolutional methods while transforming generic graphs into data with grid-like patterns. Transformation is carried out using a brand-new k-largest node selection method that ranks the values of node features.

(5) Mixture Model CNN (MoNet). For analyzing non-Euclidean geometries of GCNN and ACNN for graphs, GCN and DCNN models are proposed, but they come with some shortcomings. For analyzing, each segment of the shape requires a separate local function, while the functions are not learnable to use on similar locals. Using parametric construction which can be used in similar localities, a different framework of a mixture model network is presented [69]. Operations are convolution working on spatial-domain methods using local parameters of 'patches' of manifolds or graphs [73]. The patches create a local function that can be represented as a Gaussian kernel mixture.

A general spatial framework is used in mixture model CNNs for graphs and manifolds. For any point on the manifold x and the vertex y of the neighborhood N such that $y \in N\ (x)$, a vector of dimension d and pseudocoordinates are denoted as $u(x, y)$. A weighting function is achieved using learnable parameters. Using fixed parameters of Gaussian kernels and geodesic coordinates, the mixture model CNN can be reverted to GCNN, ACNN, and GCN models. The mixture model is based on parametric kernels by using learnable parameters denoted as follows [69]:

$$g_j(u) = \exp\left(-\frac{1}{2}\left(u - w_j\right)^T \sum_j^{-1}\left(u - w_j\right)\right), \tag{31}$$

where Σj is the covariance matrix and w_j is a mean vector constructed from a Gaussian kernel. Covariances are restricted to 2D and 2jD for patch operators.

To achieve deep learning, the aggregate ensemble CNN model integrates two distinct convolutional neural network architectures. Two deep-learning networks—AlexNet66 and NIN67—are integrated to calculate the weighted average of feature vectors. Based on AECNN modeling runs, we see that the aggregate model outperforms the single-CNN ensemble model in terms of classification accuracy and retrieval precision for images.

(6) SplineCNN. SplineCNN is a modified version of CNN for countering the non-Euclidean geometry focused on using B-spline. The model uses the convolution of spline bases, and the convolution layer takes undirected data with a directed graph as input [70].

A trainable set is used to aggregate the node features in the spatial layer. Represented by $n(i)$, the node features are weighted by the continuous kernel function [74]. The spatial relation of the nodes is stated by pseudocoordinates in U. Presenting no restrictions on U, no values are lost in a local neighborhood as it can contain edge weights, features of nodes, and local data.

A continuous kernel function is used for the convolution operation that uses B-spline bases. Constant values of trainable sets are used for parametrizing the function. For computation efficiency, all input outside the preset interval is set to zero. Now, considering a B-spline curvature of degree d and a trainable variable $t_{p,l} \in \mathrm{T}$ for every element p formed from the Cartesian product [70], $P = (N_{1,i}^m)_i \times \ldots \times (N_{d,i}^m)_i$ while l is the input feature. For I_{in} as input, the trainable parameter can be defined as $K = I_{in}.\prod_{i=1}^{d} k_i$. Now, defining the continuous convolution function [70] $g_l[a_1, b_1] \times \ldots \times [a_d, b_d] \longrightarrow R$ as

$$g_l(u) = \sum_{p \in P} t_{p,l}.B_p(u), \tag{32}$$

where B_p represents the product of basic functions. Relating to traditional CNN, SplineCNN uses a normalization factor as an extra and a differently functioned filter. Otherwise, SplineCNN can also take 2D inputs with a few kernels to analyze the dataset.

On irregularly structured, geometric input data, the model learns. The proposed convolution filter combines nearby information in the spatial domain by using a trainable continuous kernel function with trainable B-spline control values. SplineCNN is the first architecture that enables robust end-to-end deep learning directly from geometric data.

3.1.3. Spectral- and Spatial-Based Models. In this method, models are benefited from both spectral and spatial characteristics. Their combined features use for better data extraction.

(1) Multiscale Dynamic Graph Convolutional Network (MDGCN). GCN is capable of performing convolution on non-Euclidean data and is ideal for irregular image regions represented by graph topological data [42], so that these two stages may work together to produce discriminative embedded features and a revised graph, and the graph must be constantly changed while the graph convolution technique is running. Simple linear iterative clustering (SLIC) [75] is utilized when a hyperspectral image is supplied as input. This approach creates homogeneous superpixels. Then, at different spatial scales, graphs are constructed on the top of these superpixels. Following that, the input graphs are further refined by performing convolutions on them, which simultaneously acquire and accumulate multiscale spectral-spatial features. In an ideal embedding space, superpixels possibly belonging to the same class will be grouped. Finally, the well-trained network produces the categorization results in [42].

Hyperspectral image categorization is the name of the proposed model. During the convolution process, MDGCN utilizes dynamic graphs that are gradually refined. As a result, the graphs can accurately encode the inherent similarities between image regions and aid in the discovery of precise region representations. To completely utilize the multiscale information and gain hidden spatial context with superior results, many graphs with various neighborhood scales are built.

3.1.4. Comparison between Spectral- and Spatial-Based Models. The model that is based on spectral analysis is far more effective than the spatial one. Although the intricacy of the calculations grows more difficult as the size of the graph rises, it uses the eigendecomposition technique inside this convolution which may modify the results [67]. A summary of their main points is shown in Figure 8.

On the other hand, spatially based models are more beneficial in terms of huge graph-based applications. These models accomplish localization by grouping the nodes in their immediate surroundings.

The most significant thing to note is that it may be applied directly to the graph data in contrast to the spectral one. While a model based on spectral analysis has difficulty dealing with graph-based input data, a model based on spatial analysis can very effectively handle many sources of graph input (such as edge features and edge directions).

Because it functions well only on certain preset graphs, the spectral-based approach only has a restricted range of applications. To put it another way, it is improbable that the model trained for one application would function well for another, because its graphs and Laplacian nature are unique.

On the other hand, on a spatial basis, it is somewhat reliant on each node of graphs. Because of this, it is used extensively in three-dimensional form and structural analysis (e.g., shape correspondence on the FAUST dataset). Because of this, it is relatively easy to apply this model to a variety of roles and structures. Because of this, the spatial model is broader and getting more appreciation day by day.

3.1.5. Apart from GCN Architecture. This section will provide an overview of additional graph neural networks. GAN and GNN are described here.

(1) Graph Attention Networks (GANs). An attention-based architecture called graph attention networks is used to classify nodes in graph-structured data. The goal is to use a self-attention method to monitor each node's neighbors to compute each node's hidden representations. The attention structure has several intriguing characteristics, such as efficient operation, because it may be parallelized over node neighbor sets. By assigning arbitrary values to neighbors, it may also be used to graph nodes with varying degrees, and the method is capable of direct inductive learning issues, including challenges where the algorithm must generalize to wholly unknown graphs [76].

(2) Graph Generative Network (GGN). The difficulty of creating a graph structure for expanding graphs having different nodes that are disconnected from the previously observed graph is overcome by graph generative networks [77]. The slow response issues in social platforms and recommendation systems it has significant significance. The fundamental generating process is assumed to be stationary during growth. Neither node characteristics nor natural extension to new, isolated nodes is utilized by graph RNN [6]. Similar problems plague the majority of alternative graph representation learning techniques; notably, the separation from the existing graph makes it difficult to apply aggregation or pass messages. Understanding how graph architectures are generated consecutively for situations where node characteristics and topological information are both present and for situations in which only node characteristics are accessible solves this problem.

A sequential generative model for developing graphs is proposed that combines graph representation learning and graph convolutional networks. Scalability, however, is still

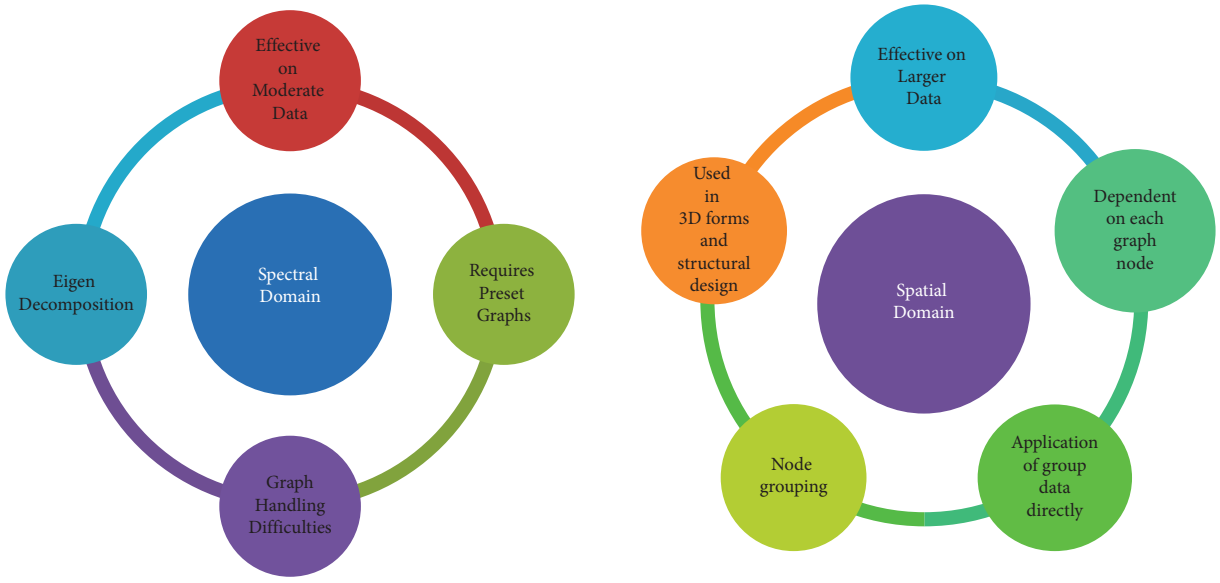


Figure 8: Spectral and spatial model features show distinctive characteristics.

a significant problem because it depends on the size of the entire graph.

3.2. Method Based on Manifolds. As discussed earlier, manifolds are nonconventional geometry having a complex shape. Figure 9 describes the manifold differently as its smallest portion can be explained as conventional geometry like a rectangle. The manifold-based model works based on this rule. Manifolds can be of different complex shapes so does nature have.

3.2.1. Voxel-Based. Bounded by small boxes, 2D space images cannot figure the depth of 3D spaces or be further analyzed as the data become blurry in smaller portions as in Figure 10. Voxel-based data create an object by segmenting them piece by piece with a 3D object, thus creating a richer dataset capable of in-depth analysis with higher accuracy.

(1) ShapeNet. ShapeNet advances CNNs to non-Euclidean manifolds, demonstrating how to use them to create invariant shape descriptors. Utilizing a local network of geodesic coordinates, ShapeNet produces "patches" that are then put to a range of techniques and linear as well as nonlinear operators [78]. These filters' parameters are optimization parameters that can be trained to reduce a loss function that depends on the task at hand. Because of the framework's considerable flexibility, different descriptors can be obtained depending on the requirements by combining several layers with various configurations. CNNs are expanded to manifolds by using the idea of geodesic convolution. The design, known as ShapeNet [78], is made up of numerous tiers that are applied in succession, meaning that the result of one layer advances as the input for the next

Figure 9: Manifold (torus) shows the analysis of manifold and assumes segments as Euclidean geometry.

process. The depth of the model is measured by the number of "hidden" layers that exist between both the input and output levels. The levels are followed as fully connected, ReLU, convolution of geodesic, angular max pooling, and Fourier transform magnitude.

The Siamese neural network [79], a well-liked architecture that has been extensively employed in metric learning tasks, is how ShapeNet is trained. A Siamese network comprises two identical models with the same parameterization and is fed by pairs of data that are purposefully similar or distinct. The loss is minimized in the model [78]:

$$l(\varnothing) = (1 - \gamma)l + (\varnothing) + \gamma l - (\varnothing), \tag{33}$$

where $\gamma \in [0, 1]$ denotes the differential parameter between the losses [78],

(a) (b)

Figure 10: (a) Voxel data using 3D segment and (b) image data using the 2D grid. Because of higher dimensions, quality becomes higher than that of the right one.

$$l_+(\varnothing) = \frac{1}{2}\sum_{i=0}^{|T_+|}\left\|F_\varnothing\left(g^i\right) - F_\varnothing\left(g_+^i\right)\right\|^2,$$

$$l_-(\varnothing) = \frac{1}{2}\sum_{i=0}^{|T_-|}\max\ \left\{0, \mu - \left\|F_\varnothing\left(g^i\right) - F_\varnothing\left(g_-^i\right)\right\|\right\}^2, \tag{34}$$

where $\varnothing$ is the variable of the ShapeNet model. The set of layers is given as $T_\pm = \{(g^2, g_\pm^2)\}$, and the negative parts are pulled at margin μ.

With the use of non-Euclidean manifolds, the model suggests generalizing convolutional neural networks to learn hierarchical task-specific features. The model is extremely flexible and general, and it may be made arbitrarily complicated by stacking additional layers. The model improves on various prior shape descriptor approaches by energizing them.

3.2.2. Multiview-Based. If 2D views for different surfaces are combined, a good idea of the 3D object can be obtained, as shown in Figure 10. Based on this fact, the multiview model takes different 2D features as input, as in Figure 11(a), and combines them by view pooling for analyzing 3D objects, as in Figure 11(b).

(1) Multiview Convolutional Neural Networks (MVCNNs). A conventional CNN architecture taught to detect forms' generated viewpoints that are not linked to one another shows a 3-dimensional shape that can be detected from a single view with greater accuracy than using 3-dimensional form descriptors of the highest quality [33]. Multiple shapes improve recognition rates. This new CNN architecture integrates many perspectives of 3D geometry into a single description for improved recognition. This multiview depiction of 3D forms is useful for many activities. First, we utilize existing 2D picture attributes to create a view description. This is the simplest way to use multiview. Multiple 2D image descriptors per 3-dimensional form, one per view, must be integrated for recognition jobs [33].

For image descriptors, two types of image descriptors for each 2D view: a state-of-the-art "hand-crafted" image descriptor based on Fisher vectors [80] with multiscale SIFT and CNN activation features [81] are used in this model. One-versus-rest linear SVM was trained to categorize forms using picture information [33]. A measurement of distance or likeness is essential for retrieval tasks. Taking shape as x along with r_x as image descriptors and shape y with r_y image descriptors, the space around them is calculated as in equation (42). The space between two 2D images can be expressed as z_2, and the space between their feature vectors is $\|p_i - q_j\|_2$. So it is as follows [33]:

$$S(p,q) = \frac{\left(\sum_j \min_i \left\|p_i - q_j\right\|2\right)}{\left(2r_y\right)} + \frac{\left(\sum_i \min_j \left\|p_i - q_j\right\|2\right)}{\left(2r_x\right)}. \tag{35}$$

The model suggests using these several 2D projections, which produce excellent discrimination performance. Compactness, efficiency, and improved accuracy can be attained by creating descriptors that are aggregations of data from many viewpoints. Additionally, these 3D shapes can be

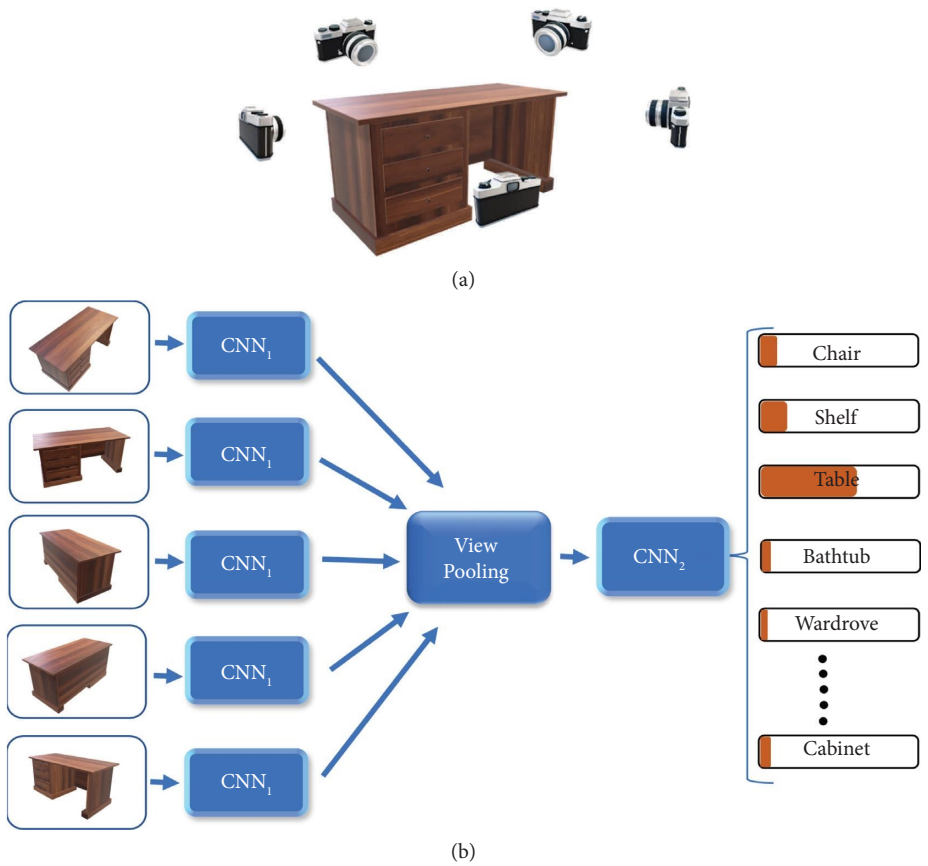


Figure 11: (a) Shape analysis with different views; (b) the multiview model architecture takes views as 2D input to analyze 3D objects.

recovered using sketches with high precision and take advantage of the implicit understanding of 3D shapes included in their 2D views by connecting the information of 3D shapes to 2D representations like sketches.

3.2.3. Difference between Volumetric CNN and MVCNN. A 3D form is encoded in the volumetric representation as a 3-dimensional tensor of binary or real values. Oppositely, the multiview representation organizes a 3-dimensional form as an accumulation of multiple perspective representations. It seems intuitive that the volumetric representation should be able to input extra information about the characteristics of three-dimensional structures rather than the multiview representation. The most important highlights are shown in Figure 12.

However, Qi et al. [36] replicated the tests using a grid of 30 voxels with 3D ShapeNets with multiview CNNs on the ModelNet40 dataset. According to the data, the categorization performance of individual algorithms suggests that the volumetric CNN with voxel-dependent performance is 7.3% less accurate than the MVCNN [82]. There are at least two probable explanations, including the input data performance and the diversity in network architecture [36]. However, when both networks are fed equal levels of information, the accuracy of the classification of MVCNNs is much higher (89.5%) than that of 3-dimensional

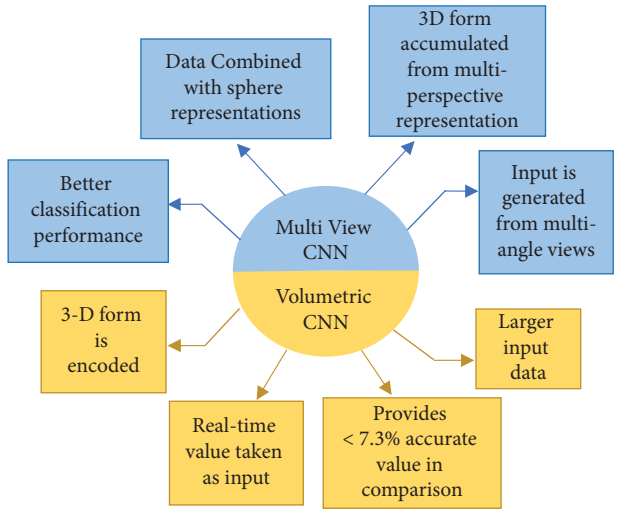


Figure 12: MVCNN vs. volumetric CNN model features.

ShapeNets (84.7%). In this experiment, data from MVCNNs are combined with sphere representations of the grid of 30 equal in height, width, and length. Even with lower features (resolution) of input data, the classification performance of the MVCNN is much greater than that of 3D ShapeNets. This shows that the design of VCNNs has many opportunities for improvement.

3.2.4. Point-Based. The advantage of the point-based approach is that it may take point cloud data as input without first transforming it to voxel, mesh, or another type of 3D representation. A point cloud is a combination of geometrically important points that form a structure. In place of a large number of benefits, there are certain difficulties, such as sparsity and the unpredictability of the geometric data. Because it may be used effectively in various contexts, researchers are becoming more interested in this model. As a direct consequence of this, there is persistent progression [83, 84].

(1) PointNet++. PointNet++ [27] is a hierarchical structure of the neural network that performs recursion of PointNet [26] on layered partitioning of the input. It is a direct successor of PointNet. Although PointNet was the very first DNN capable of manipulating 3D point clouds natively, several networks have since been developed. It learns the spatial coding of every location in the input cloud and then, by aggregating all the features, it determines the global characteristics of a point cloud. However, PointNet++ removes its shortcoming by solving how a local division of a point cloud can be carried out and how local characteristics of a point cloud can be extracted. In other words, it learns about local characteristics with increasing contextual scales [27]. Using an analysis of metric space lengths, Qi et al. [27] claimed that it is capable of learning features very robustly, even in nonuniformly sampled point sets. The way it extracts features can be briefed into three sections:

(a) *Sampling Layer.* The sampling method is known as FPS, or farthest point sampling, which begins its work by picking a random series of points out from point cloud functioning as its input.

(b) *Grouping Layer.* The objective of the grouping layer is to construct local areas before extracting characteristics. More specifically, this research uses the neighborhood ball approach rather than the KNN algorithm since it is feasible to ensure a set area scale. Multiple subpoint clouds are created by employing surrounding points surrounding centroid points (within a defined radius) [27].

(c) *PointNet Layer.* As described earlier, it uses the PointNet algorithm to originate preliminary assumption and then run it through some iterations to get closer to the extraction of features.

In terms of 3D shape, there may be an ununiform density of sampling points like the perspective effect, and radial density variations cause great trouble learning features. Qi et al. [27] proposed an abstraction layer that may aggregate information from multiple aspects according to local point densities. By this algorithm, the author comes up with an accuracy of 90.7% in ModelNet40 [83].

The model is suggested for handling sampled point sets in the metric space. PointNet++ efficiently learns hierarchical features concerning the distance metric by performing recursive operations on nested partitioning of the input point set. Two unique sets were suggested abstraction layers that intelligently aggregate multiscale data following local point densities, producing improved results to address the problem of nonuniform point sampling.

(2) Taylor GMM Convolutional Network (TGNet). The Taylor GMM convolutional network constructs a graph pyramid through clustering point clouds. In each layer of the pyramid, local regions are abstracted progressively. For learning local features, TGConv is applied and the targeted data are again interpolated at a finessed scale at each layer. On a similar scale, the features are interconnected [85]. TGNet uses limited computation, and information losses occur. To counter the issue, MLP can be applied to input for conserving information along the process. Finally, sampled features and the finessed scale are combined for per-point segmentation.

The TGNet model is driven by TGConv [86]. Let a graph $G = (V, E, D)$ formed of a point cloud $C = \{c_1, c_2, \ldots, c_n\} \subseteq R^3$, where $V = \{1, 2, \ldots, n\}$ is the collection of vertices and $E \subseteq V \times V$ is the collection of edges [86]. Every directed edge $(x, y) \in E$ has 3D pseudocoordinates defined as $d(x, y)$ given that D is the set of the coordinates. We consider all point $y \in H(x)$, where S is the neighbor set vertex $x, d(x, y)$ which indicates a 3D vector of coordinates of y. Let $a = \{a_1, a_2, \ldots, a_N\}$ be the set of input features of the vertex. For F as a feature dimension, the features $a_i \in R^F$ are associated with a related graph $i \subseteq V$. Local coordinates are generated from input characteristics using Taylor kernel functions [86] with Gaussian weighting. The learnable weighted functions are denoted by D. The convolution function is performed by the aggregation of the feature sets and is given by [86]:

$$(f * g)(x) = \mathrm{Agg}\left(g_\theta\left(\sum_{s=1}^{S} D_s(x) a_y\right)\right), y \in H(x). \quad (36)$$

The model suggested altering the linear combination of convolutional feature maps in the conventional convolutional operation of CNNs to collect detailed high-frequency and low-frequency information. It is shown that TaylorNets have a nonlinear combination of the convolutional feature maps based on Talyor expansion 87. The steerable module created by TaylorNets is generic, making it simple to include in various deep architectures and to be taught using the same backpropagation algorithm pipeline. This results in a higher representational capacity.

(3) MongeNet. In the ShapeNet model, triangular meshes are used for sampling 3D surfaces, but sampling gets irregular as the surface angle changes and clamping or undersampling occurs. The problem can be defined as a transport problem of discrete measures and simplex. MongeNet is a neural network working as a uniform mesh sampler to counter the mentioned problem [87]. Computation is performed on GPUs and batchwise across triangles. This model's direct competitor is the current cutting-edge sampler PyTorch3D. Test findings suggest that, for a moderate increase in computational cost, the model provides greater performance than the previous model.

MongeNet is proposed to replace the already established sampling technique using a triangle mesh. The model minimizes 2-Wasserstein distance which implies favorable transport distance for segmented Euclidean metrics [87]. MongeNet approximation relies on solving convex which relies on Laguerre tessellation for computing the optimal distance.

Generally, solving the resulting point cloud should consume a lot of time with a high cost, but the model proposes learning optimal positional points using a feed-forward neural network with satisfactory approximation and fast calculation. Stating that the network is denoted as f_θ, where θ is the learnable parameter. The input is the triangle t with limited output points such that $o \in [a, b]$ and random noise $n \in R$ that follows the normal distribution, and the output provided is a random order of $f_\theta(t, o, n) \in R^{3\times o}$. For training set D with components $[t, S]$, the sampled points [87] are

$$\operatorname{argmin} \theta \sum_{i=a}^{N} \sum_{o=a}^{b} L\left(f_\theta\left(t_i, o, n_i\right), S_i\right). \tag{37}$$

Stating W_2^ε as optimal transport and $n_i \sim N(0, 1)$, the loss function L is given as follows [88]:

$$\begin{aligned} L(t, o, n, S) = {} & W_2^\varepsilon\left(f_\theta(c, o, n), S\right) \\ & - \alpha W_2^\varepsilon\left(f_\theta(c, o, n), f_\theta\left(c, o, n^{'}\right)\right). \end{aligned} \tag{38}$$

The model overcomes drawbacks of the common mesh sampling approach used by most 3D deep-learning models, such as its proneness to erroneous sampling and clamping, which leads to noisy distance estimates. For a small additional investment in computer deep learning, MongeNet outperforms already used methods, such as widely used random uniform sampling.

3.2.5. Spatial-Based. Spatial convolution is the implementation of graph-based convolution processes directly. Being the size of the standard convolution kernel predefined, a predetermined-length neighborhood must be selected for convolution if standard convolution is performed on a graph. However, graph nodes often contain a variable number of neighbors despite data with a normal grid form. The graph convolution technique mimics the image convolution process and is constructed from spatial node relationships. Similar to the central pixel in normal CNN 3×3 filters, the presentation of a center node depends on the aggregate output of its neighboring nodes.

(1) Geodesic Convolutional Neural Networks (GCNN). The GCNN [16] model is an extension of non-Euclidean manifolds of the convolutional neural network (CNN) paradigm. This local geodesic framework of polar coordinates is used to extract "patches," which pass through a series of filters including linear and nonlinear processes. To reduce a task-specific cost function, the values of the filters with linear combination weights are optimized variables. Utilizing the diagonal results of heat-like operators yields a variety of very well spectral-shaped descriptors. Thus, spectral descriptors may be utilized as heat kernel signature (HKS), wave kernel signature (WKS), and optimal spectral descriptors (OSDs) [16]:

$$(D(x)f)(\rho, \theta) = \int_X v_{\rho,\theta}\left(x, x^{'}\right) f\left(x^{'}\right) \mathrm{d}x^{'}. \tag{39}$$

Mapping is performed by $(D(x)f)(\rho, \theta)$, and the value of the function f is in the range of $x \epsilon \mathbf{X}$ to the neighbor polar coordinates (ρ, θ), in which $v_\rho(x, x^{'})$ is the radial interpolation weights having a geodesic distance from x revolving around ρ, $v_\theta(x, x^{'})$ represents the angular weights derived from a collection of geodesics radiating from x in direction θ, $dx^{'}$ indicates the surface component of the Riemannian measure, and $v_{\rho,\theta}$ localizes a weighting function around $v_{\rho,\theta}$ [16]. $(D(x)f)(\rho, \theta)$ in GCNN converts the values for the function f around the node x into the regional polar coordinates ρ, θ, hence forming the geodesic convolution [16]:

$$(f * h)(x) = \max_{\Delta\theta\epsilon\{0,)2\pi} \int h(\rho, \theta + \Delta\theta)(D(x)f)(\rho, \theta) \mathrm{d}\rho \, \mathrm{d}\theta, \tag{40}$$

where $h(\rho, \theta + \Delta\theta)$ acts as a filter. GCNN is composed of numerous consecutively applied layers. The layer is differentiated as follows:

(1) Typically, the linear layer comes after the input layer and before the output layer to modify the input and output sizes by a linear function.

(2) The usual Euclidean convolutional layer is replaced with the geodesic convolution (GC) layer.

(3) The angular max pooling layer combines the GC layer to estimate the optimum filter rotation [16].

(4) The FTM layer is an extra constant layer that performs the patch operation to every input dimension, proceeded by rotational coordinates as well as actual value Fourier transform.

(5) The covariance (COV) layer is utilized in recovery applications that need the aggregation of point-wise descriptors into something like a descriptor of global shape [89].

The model was developed for uses like shape correspondence or retrieval to learn hierarchical task-specific features on non-Euclidean manifolds. Our model is extremely flexible and general, and by stacking additional layers, it may be made arbitrarily complicated. By altering the local geodesic charting process, GCNN could be used for different form of representations, such as point clouds.

(2) Anisotropic Convolutional Neural Networks (ACNNs). These networks are generalization of standard CNNs to non-Euclidean entities in which traditional convolutions are substituted by projections over a set of centered approach anisotropic diffusion kernels [25]. As spatial scaling functions, anisotropic heat kernels retrieve the inherent regional representation of a function defined on the manifold. This

ACNN architecture is a CNN. This patch operator design is far easier than those of GCNN, irrespective of the manifold's subsurface diameter, which is not limited to triangular meshes. The basis of this method is the generation of regional geodesic polar coordinates utilizing a technique that may have been used for fundamental shape context descriptors [90].

ACNN translates heat kernels as a regional weighted function and builds the patch operator as follows [25]:

$$\left(D_{\alpha}(x)f\right)(\theta,t)=\left(\frac{\int_{X}h_{\alpha\theta 1}(x,y)f(y)\mathrm{d}y}{\int_{X}h_{\alpha\theta 1}(x,y)\mathrm{d}y}\right), \tag{41}$$

and for some anisotropy level, $\alpha>1$. $h_{\alpha\theta 1}(x,y)$ is the anisotropic heat kernel that indicates the quantity of heat transmitted at the period t from the point x to the point y [25].

Convolution [25] can be described as

$$(f*b)(x)=\int b(\theta,t)\left(D_{\alpha}(x)f\right)(\theta,t)\mathrm{d}t\,\mathrm{d}\theta. \tag{42}$$

The major curvature direction is primarily used as the reference $\theta=0$ in ACNN's creation. The most potential future work path is the use of ACNN to graph learning. GCNN and ACNN approaches work in spatial domains; avoiding the limitations of standard spectral approaches with varied domains, these techniques have proven to be more successful than traditional hand-crafted methods in locating deformable shapes.

Convolutional neural networks are generalized to non-Euclidean domains in the proposed model, which enables deep learning on geometric data. The work, which is currently the most generic intrinsic CNN model, continues the very recent trend of applying machine-learning techniques to computer graphics and geometry processing applications.

The provided models are fairly efficient and beneficial for a variety of reasons; therefore, their results and analyses are quite valuable for surveying their efficacy on various datasets. The accuracy and error of the previously stated models are summarized in Table 1 together with their related datasets.

A second perspective was studied, based on the Scopus string TITLE-ABS-KEY (geometric AND deep AND learning, OR graph, OR manifold) AND (LIMIT-TO (PUBSTAGE, "final")) AND (LIMIT-TO (PUBYEAR, 2022) OR LIMIT-TO (PUBYEAR, 2021) OR LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018) OR LIMIT-TO (PUBYEAR, 2017) OR LIMIT-TO (PUBYEAR, 2016) OR LIMIT-TO (PUBYEAR, 2015)) AND (LIMIT-TO (SUBJAREA, "COMP")) AND (LIMIT-TO (EXACTKEYWORD, "Deep Learning") OR LIMIT-TO (EXACTKEYWORD, "Geometry") OR LIMIT-TO (EXACTKEYWORD, "Deep Neural Networks") OR LIMIT-TO (EXACTKEYWORD, "Convolution") OR LIMIT-TO (EXACTKEYWORD, "Convolutional Neural Networks") OR LIMIT-TO (EXACTKEYWORD, "Computer Vision") OR LIMIT-TO (EXACTKEYWORD, "Convolutional Neural Network")) which is shown in Table 2. It describes the most advanced uses of computer vision applications of these models, including object detection, medical imaging, face detection, action, and activity detection, human pose detection, network detection, and pedestrian trajectory. These applications are stated along with their specific employments.

3.2.6. Comparative Analysis of Described Models. Table 1 provides a comparative analysis of described models in terms of numerical values, novelty, and their limitations, which is sorted according to the proposed year. Initially, ShapeNet [78] was proposed in 2015 in the field of non-Euclidean geometry which was successful in terms of generalizing CNN to learn specific features. However, it is only limited to mesh features. With continuous improvements, in the same year, MVCNN [33] was proposed which is compact and efficient with improved accuracy. However, the 3D descriptor is untested. Later on, in 2017, TGNet [85] changed the linear convolution of CNN in the feature map, but its TGConv has a very narrow dynamic range. In 2017, MoNet [69] was proposed to ensemble the CNN model, but this method does not follow segmentation and weighted categorical cross-entropy outcomes.

In the next year, SplineCNN [70] was proposed which uses trainable continuous kernel functions and B-spline values to extract local features. However, its global behavior becomes worse along the large geodesic error. In 2019, CayleyNet [28] was proposed, which specializes in small frequency bands with few filter parameters while maintaining spatial localization. However, its bidirectional line cost is more, and the routing method is more sophisticated. In this continuation, CurvaNet [63] proposed in 2020 for a U-Net-like hierarchical structure is shown to exploit multiscale curvature characteristics but does not have structural regularization such as segment class topology. In this succession, MDGCN [42] initiated in 2020 utilizes dynamic graphs that are gradually refined and can accurately encode the inherent similarities between image regions. LSTM is used which requires high computational cost. The introduced GC unit has lower-order approximations of spectral graph convolution.

Again in 2020, ACSCNN [30] offers anisotropic convolution enabling a more thorough capture of the intrinsic local information of signals. However, the model needs shape segmentation and classification. In 2021, MongeNet [87] was proposed. The gap between the target point cloud and the sample point cloud from the mesh shrunk more quickly. So, at a given optimization time, the input point cloud was represented more accurately. However, it is a costly and time-consuming computation. This year, UV-Net [31] utilizes existing image and graph convolutional neural networks and can operate on B-rep data. However, the model did not use the B-rep curve and surface types, edge convexity, half-edge ordering, etc. Moreover, UV-grid features do not rotate.

Table 1: Performance summary of the described models with their corresponding datasets as well as their limitations and complexity.

Architectures	Year	Category	Datasets	Result (accuracy)	Novelty	Limitations
ShapeNet [78]	2015	Voxel-based	FAUST [91] SCAPE [92] TOSCA [93]	4.04 (EER %) 3.34 (EER %) 5.61 (EER %)	Suggests generalizing CNN to learn hierarchical task-specific features	Experiments were limited to meshes only
MVCNN [33]	2015	Multiview-based	ModelNet40 [94]	90.1%	Compactness, efficiency, and improved accuracy are attained by creating descriptors that are aggregations of data from many viewpoints	Building compact and discriminative 3D descriptors is untested
			SketchClean [95]	87.2%		Time-consuming, large datasets
TGNet [85]	2017	Point cloud-based	ScanNet [96]	62.2%	Changing CNNs' linear convolutional feature map combination	Multiobject linked area labeling failure, mostly due to the TGConv's narrow dynamic range
			S3DIS [97]	88.5 (OA) 57.8(mIoU)		
			Paris-Lille-3D [98]	96.97 (OA) 68.17 (mIoU)		
MoNet [69]	2017	Spatial graph convolution	MNIST [99]	95.05%	The ensemble CNN model combines two CNNs	This method underestimates segmentation, and weighted categorical cross-entropy improves outcomes
			Graph: CORA [100]	81.69 ± 0.48%		
			Graph: PubMed [101]	78.81 ± 0.44%		
			FAUST [91]	0 cm error (at 90% points) 4 cm error (at 99% points)		
SplineCNN [70]	2018	Spatial graph convolution	MNIST [99]: Grid	99.22 %	Using a trainable continuous kernel function and B-spline control values combines local spatial information	Global behavior worsens with larger geodesic error bounds. Geodesic misclassifications are infrequent
			MNIST [99]: Superpixels	95.22 %		
			CORA [100]	89.48 ± 0.31%		
CayleyNets [28]	2019	Spectral graph convolution	MNIST [99]	99.18 %	Specializes in small frequency bands with few filter parameters while maintaining spatial localization	Bidirectional lines cost more, and the routing method is more sophisticated
			CORA [100] (standard)	81.0 ± 0.5%		
			CORA [100] (extended)	88.09 ± 0.6%		
CurvaNet [63]	2020	Spectral graph convolution	SCAPE [92]	0.875	U-Net-like hierarchical structure is shown to exploit multiscale curvature characteristics	No structural regularization such as segment class topology
			FAUST [91]	0.952		
			MIT	0.911		
			Adobe fuse	0.943		
			All datasets	0.934		
			PSB: Human	0.876		
			PSB: Teddy	0.904		
			PSB: Fish	0.931		

Table 1: Continued.

Architectures	Year	Category	Datasets		Result (accuracy)	Novelty	Limitations
MDGCN [42]	2020	Spectral-spatial-based	Indian Pines [102]	OA	93.47 ± 0.38 %	Utilizes dynamic graphs that are gradually refined and can accurately encode the inherent similarities between image regions	LSTM is used which requires high computational cost. The introduced GC unit has lower-order approximations of spectral graph convolution
				AA	96.24 ± 0.21 %		
				Kappa	92.55 ± 0.43 %		
			The University of Pavia [103]	OA	95.68 ± 0.22 %		
				AA	93.15 ± 0.28 %		
				Kappa	94.25 ± 0.29 %		
			Kennedy Space Center [104]	OA	99.79 ± 0.01 %		
				AA	99.61 ± 0.02 %		
				Kappa	99.77 ± 0.01 %		
ACSCNN [30]	2020	Spectral graph convolution	FAUST [91]	SHOT	98.06% (geodesic distance = 0) 99.26% (geodesic distance = 0.01)	Provided anisotropic convolution enables a more thorough capture of the intrinsic local information of signals	The model needs shape segmentation and classification
				1	99.56% (geodesic distance = 0) 99.87% (geodesic distance = 0.01)		
MongeNet [87]	2021	Point cloud-based	The gap between the target point cloud and the sample point cloud from the mesh shrunk more quickly. So, at a given optimization time, the input point cloud was represented more accurately				Costly and time-consuming computations
UV-Net [31]	2021	Spectral graph convolution	Machining feature [105]		99.94 ± 0.00%	Solid model classification	Utilizes existing image and graph convolutional neural networks and can operate on B-rep data	The model did not use the B-rep curve and surface types, edge convexity, half-edge ordering, etc. UV-grid features do not rotate
			FabWave [106]		94.51 ± 0.10%		
			SolidLetters		97.24 ± 0.10%		
			MFCAD [107]		99.95 ± 0.02 % (Per-face)	Solid face segmentation	
			ABC [108]		88.87 ± 0.70 % (Per-face)		

EER = equal error rate; mIoU = mean intersection over union; OA = overall accuracy; AA = average accuracy; Kappa = Kappa coefficient; geodesic distance = shortest path between the vertices. FAUST [91] includes 300 high-resolution scans of 10 human participants in 30 positions and is used for 3D mesh reconstruction, shape analysis, virtual try-on, animation, and gaming. SCAPE [92]: the dataset contains 71 registered meshes of a particular person in different poses and is used for 3D mesh reconstruction, shape analysis, virtual try-on, animation, and gaming. TOSCA [93]: TOSCA (the object shape capture archive) is a repository of 3D scanned models of real-world objects and used for shape analysis, 3D object reconstruction, and deformation analysis. ScanNet [96]: it contains high-resolution 3D scans of over 1,500 indoor spaces and is used for scene understanding, 3D reconstruction, robotics, and virtual reality. S3DIS [97]: the dataset includes 6 large-scale indoor areas and is used for scene segmentation, object detection and recognition, and robotics. Paris-Lille-3D [97] is a large-scale dataset of dense point clouds representing urban environments in France. The dataset contains over 2 billion points, with a point density of approximately 20 points per square meter, and is used for urban planning, autonomous driving, object detection and recognition, and environmental monitoring. MINST [99] includes a total of 70,000 grayscale images of handwritten digits from zero to nine, each of which is 28 × 28 pixels in size, and is used for digit recognition and data augmentation. CORA: the dataset includes a total of 2,708 research papers in the field of computer science, each of which is represented by a bag-of-words vector of its abstract, and is used for citation network analysis, link prediction, text classification, and graph convolutional networks. PubMed [101]: the dataset includes information on over 32 million articles and is used for literature review, text mining, natural language processing, and biomedical informatics. PSB: the dataset consists of a set of protein structures and associated information that is used to evaluate and compare methods for predicting protein structure and function. Machining feature [105]: a synthetic labeled, balanced dataset representing machining features such as chamfers and circular end pockets applied to a cube. FabWave [106]: a small labeled, imbalanced collection of 5,373 3D shapes split into 52 mechanical part classes, such as brackets, gears, and o-rings. MFCAD [107]: a synthetic segmentation dataset of 15,488 3D shapes, similar to the machining feature dataset, but with multiple machining features. SolidLetters consists of 96 k 3D shapes generated by randomly extruding and filleting the 26 alphabets (a–z) to form class categories across 2002 style categories from fonts. ABC [108]: a real-world collection of millions of 3D shapes. The dataset is unlabeled and imbalanced and has many duplicates. ModelNet40 [94]: the dataset contains 12,311 CAD models of 40 different object categories and is used for object recognition, 3D recognition, and shape analysis. SketchClean [95] contains 160 categories, on which humans can achieve 93% recognition accuracy, and is used for sketch recognition, sketch synthesis, and human-computer interaction. Indian Pines [102]: it consists of 145 × 145 pixels with a spatial resolution of 20 m × 20 m, has 220 spectral channels covering the range from 0.4 to 2.5 μm, and is used for remote sensing, environmental monitoring, and agricultural analysis. University of Pavia [103]: the dataset captured Pavia University in Italy with the ROSIS sensor in 2001. It consists of 610 × 340 pixels with a spatial resolution of 1.3 m × 1.3 m, has 103 spectral channels in the wavelength range from 0.43 to 0.86 μm after removing noisy bands, and is used for urban analysis and remote sensing. Kennedy Space Center [104]: the Kennedy Space Center dataset was taken by the AVIRIS sensor in Florida with a spectral coverage ranging from 0.4 to 2.5 μm. This dataset contains 224 bands and 614 × 512 pixels with a spatial resolution of 18 m and is used for remote sensing and space exploration.

Table 2: Various applications of computer visions based on geometric deep learning.

Applications	References	Employment
Object detection	[109]	Salient object detection
	[110]	Agile formation control of drone flocking
	[111]	Online semantic mapping system
	[112]	Nonlinear estimation in robotics and vision
	[113]	Degree of disease in citrus fruits
	[114]	Network model suitable for indoor mobile robots
	[115]	3D object detection using point clouds
	[116]	Detecting visual relations and scene parsing
	[117]	Detecting 3D objects from noisy point clouds
	[118]	Robust label propagation for saliency detection
	[119]	Weakly supervised object detection
	[120]	Detection of distinct objects like seed selection
	[121]	Segmentation of rivers for autonomous surface vehicles
	[122]	3D scene perception
	[123]	City object detection from airborne LiDAR data
	[124]	Nonmaximal suppression product detection on the shelf
	[125]	LiDAR-based three-dimensional mapping in urban environments
	[126]	3D multiobject tracking in point clouds
2D image processing	[127]	Topic scene graphs for image captioning
	[128]	Text-based visual question answering
	[129]	Single-image 3D reconstruction
	[130]	Laser-based surface damage detection and quantification
	[131]	Smooth manifold triangulation
	[132]	Deep virtual stereo odometry
Hyperspectral imaginary	[133]	Enhanced discriminative broad learning system
	[134]	Spectral-spatial clustering of hyperspectral image
	[135]	Geometric knowledge embedding
	[136]	Hyperspectral and multispectral image fusion
	[137]	Hyperspectral unmixing
	[138]	Remote sensing image scene graph generation
Medical imaging	[139]	Brain graph synthesis
	[140]	Personalized brain-computer interfaces
	[141]	Estimating connectional brain templates
	[142]	Postmenstrual age prediction based on the neonatal white matter cortical surface
	[143]	Detecting brain state changes
	[144]	Predicting isomorphic brain graph
	[145]	Detecting and forecasting spatiotemporal anomalies
	[146]	Large lung motion in COPD patients
	[147]	Data-driven personalized cervical cancer risk prediction
	[148]	Potential drug-virus interactions against SARS-CoV-2
	[149]	Epileptic seizure prediction using scalp EEG signals
	[150]	Human connectome project multimodal cortical parcellation
	[151]	Intelligent health state diagnosis

Table 2: Continued.

Applications	References	Employment
Face recognition	[152]	Skin lesion segmentation from dermoscopic images
	[153]	Learning to cluster faces
	[154]	Facial expression recognition method for identifying and recording emotion
	[155]	Occluded face detection
	[156]	Face anonymization with pose preservation
	[157]	Consumer affect recognition using thermal facial ROIs
	[158]	Criminal person recognition
	[159]	Facial action unit detection
	[160]	Masked face detection
	[161]	Drunkenness face detection
	[162]	Face detection and recognition
	[163]	Driver drowsiness detection
	[164]	Large-scale face clustering
	[165]	Detection of facial action units
	[166]	Facial expression recognition
Action and activity recognition	[167]	Multiactor activity detection
	[168]	One-shot video graph generation
	[169]	Online graph depictions for tracking multiple 3D objects
	[170]	Event stream classification
	[171]	LiDAR-based 3D video object detection
	[172]	Salient superpixel visual tracking
	[173]	Video event recognition and elaboration from the bottom up
	[174]	Multiobject tracking with embedded particle flow
	[175]	Video scene graph generation
	[176]	Video action detection
	[177]	Multiobject tracking in autodriving
	[178, 179]	Skeleton-based action recognition
	[180]	Video distinct object recognition by extraction of robust seeds
	[181]	Video saliency detection
	[182]	Close-to-real-time tracking in congested scenes
Human pose detection	[183]	Human-object interaction detection
	[184]	Railway driver behavior recognition system
	[185]	Framework for object identification based on human local attributes
	[186]	Human object interaction detection
	[187]	Online top-down human pose tracking
	[188]	Multiperson pose grouping
	[189]	Behavior of the human body
	[190]	Pose recognition algorithm for sports players
	[191]	Interactive reasoning for human-object interaction detection

Table 2: Continued.

Applications	References	Employment
Network detection	[192]	Detection of botnets in IoT networks
	[193]	Optical flow traffic management using augmented detection
	[194]	Trajectory-user linking via graph neural network
	[195]	Message passing network for dense captioning
	[196]	Change detection in noisy dynamic networks
	[197]	Analyzing heterogeneous data from social networks
	[198]	Phishing detection on Ethereum
	[199]	Fake news detection
	[200]	Anomaly detection in dynamic networks
	[201]	Adversary situation awareness
Pedestrian trajectory	[202]	Pedestrian detection
	[203]	Obstruction to pedestrian detection
	[204]	Pedestrian trajectory prediction in public buildings
	[205]	Pedestrian movements near an amenity in walkways of public buildings

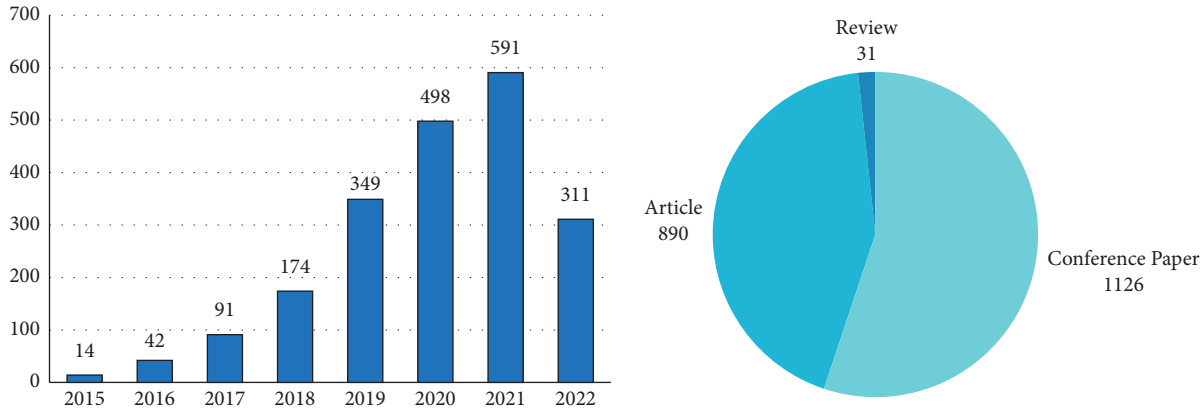


Figure 13: Graph shows documents published by year (till 2022), and charts show published documents by type.

4. Future Prospects and Challenges

These models are meant to deal with data that cannot be conveniently represented in the Euclidean space and have shown exceptional performance in tasks like shape identification, chemical design, and social network analysis. These non-Euclidean deep-learning models have several promising applications, including computer vision, robotics, natural language processing, and drug discovery. As a result, the future scope of non-Euclidean deep-learning models is very promising, and the research area is expanding in this field at an impressive rate. In addition to computer vision, it has already been used in applications such as AI pathfinding, 3D mapping, medical diagnostics, molecular analysis, VR-based applications, and even big data classification.

The extracted data by the string are plotted as "Document Published by Year" and "Documents by Type" in the following graphs and pie chart in Figure 13.

The bar graph gives information about documents published by year (till 2022). According to the graph, it can be seen that the number of publications started to increase from 2015 to 2021. In 2015, there were 14 documents published. Then, 42 were published in 2016 (an increased 121%), 91 were published in 2017 (an increased 116%), 174 were published in 2018 (an increased 91%), 349 were published in 2019 (an increased 100%), and 498 were published in 2020 (an increased 42%). It was highest in 2021 when the document published the most, 591 published (an increased 18%). 311 documents were published till the first half of 2022.

From the pie chart, it is clear that conference paper is published at the highest rate at 1126, and 890 articles were published in the document. However, the review rate is too poorer than the article and conference paper. There are a total of 31 reviews. In 100% of the pie chart, the total type of documents published is as follows: conference paper 55% (1126) + article 43% (890) + review 1.5% (31) = 100% (2047).

By evaluating the statistical future, it may be concluded that demand for geometrically based deep learning will increase. However, it leads us to newer challenges and difficulties day after day. Some difficulties are solved, and more of these are just ahead of us.

Various usages and challenges are illustrated in Figure 14. As challenges tend to be resolved, our future will progress toward these advanced applications. Key points of challenges are visualized and explained as follows:

(1) *Computational Difficulties.* Despite the success of GNN in several disciplines, the high cost of computing remains a challenge for academics and applications. A DNN includes a high set of variables, which makes testing and training stages computationally costly. It requires higher hardware computer resources like GPUs. GNN exhibits the same characteristics. In addition, computation is difficult due to the complex relationship among graph nodes as well as the nongrid structure. In contrast, the great majority of existing deep-learning systems deal with normalized datasets in the Euclidean space, such as a 1D or 2D grid, which may make use of the strong processing capacity of contemporary GPUs. In most instances, however, geometric data may not have a grid-like structure, necessitating other approaches for efficient and sophisticated computing. Therefore, accelerating graph neural network performance is a pressing demand.

(2) *Complex Architecture.* Concerning the issue of Euclidean data, the CNN architecture has achieved remarkable progress in the area of deep learning. The model gets more sophisticated as the number of network levels grows. Empirically, neural networks with more variables have the potential to perform well. However, stacked multilayered GNNS will exacerbate the smoothness issue. The nodes in a graph are connected to their surroundings, as well as the graph function in a network is a stream of data dispersion and consolidation. The more layers that are stacked, the more data from nodes will be integrated, resulting in an identical picture for all nodes. Consequently, each vertex will fall to the same value. For instance, the majority of complex GCN architectures include little or more than three or four layers. The study [206] sought to use a more sophisticated network architecture; however, its

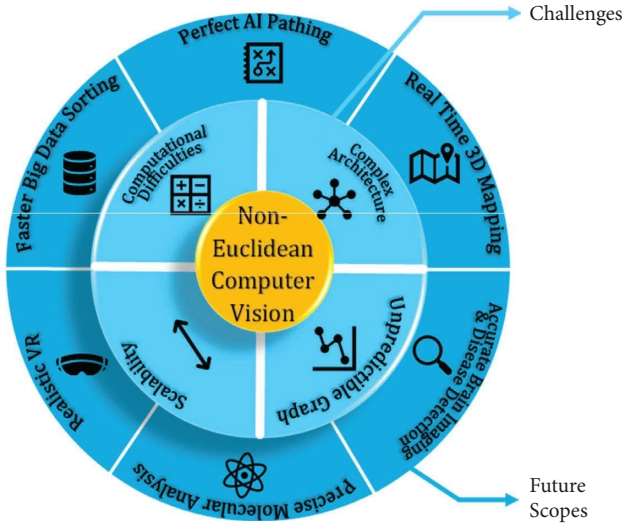


Figure 14: Challenges and scopes of non-Euclidean deep learning.

efficacy is insufficient. Nonetheless, in 2019, DeepGCNs [207] were able to construct a 56-layer network by leveraging the concepts of residual connections as well as dilated convolutions. Building a neural network based on a deep graph is an interesting yet difficult challenge.

(3) *Unpredictable Graph.* Most known techniques are used to edit static graphs, and many datasets are also static, though many graphs change over time. In social media platforms, the reduction of existing users and the addition of new users may occur at any time, as well as the relationship among members can vary considerably. The question of how to adequately describe the development of dynamic graphs is unresolved, which has some impact on the applicability of graph neural networks. There are several efforts to resolve this issue such as [208–210].

(4) *Scalability.* It is a challenging challenge to apply graph neural networks on huge graphs. On the one side, each node seems to have its neighborhood topology, which includes the hidden layer of surrounding nodes, making it challenging to train using the batch technique [39]. In contrast, while dealing with millions of nodes and edges, the Laplacian matrix of the network was challenging to compute for researchers. There are additional approaches to increase the performance of the model by quick sampling [211, 212] and subgraph training [213, 214] although the results are not particularly impressive.

5. Conclusion

Even while deep learning has traditionally relied on Euclidean geometry, the modern era's complex structural geometry demands the use of non-Euclidean geometry. The current world is aiming to include this non-Euclidean geometry in these deep-learning methods to satisfy the need for 3D complex structure analysis. Learning from complicated data, such as graphs and manifolds, is made easier using this geometric deep learning. Here, this paper explores the field of deep learning for manifolds and graphs at length. An in-depth look at the history, context, state-of-the-art and efficient mathematical deep-learning models, performance analysis, and pros and cons of deep networks used in computer vision on graphs and manifolds is presented in this paper. In addition, it is expensive and requires significant resources to deploy. Moreover, dynamic graphs and shapes demand a sophisticated real-time system, although it needs a great deal of more development for everyday usage. At this rate, however, it is reasonable to anticipate a more widely available advanced model for a variety of challenging classification tasks.

Abbreviations

P_x, P_y, P_z: Single points of point cloud P
$\hat{g}$: Fourier transform
$\Delta_X g(x)$: Laplace–Beltrami operator
R: Reference point
g: Inverse Fourier transform
$M^2(X)$: Manifold
r: Distance between two locations
f: Smooth function of manifolds
$L_n(\gamma)$: Chebyshev polynomial
θ: The angle between the axes
$df(x)$: Closest neighbor points
$\hat{h}(\gamma)$: Filter
$H(t)$: B-spline curve
$g_{v,z}(\lambda)$: Cayley polynomial
α: Anisotropy level
$S_{i,d}$: Knot vectors
G_f: Cayley filter for f signal
$n(i)$: Node features
$P(t)$: Control points
$g_v(h)$: Hidden node features
B_p: Product of basic functions
f_s: Regularization
$\alpha^{(h)}$: Multilayer perceptron
$l(\varnothing)$: Minimized loss
λ: Error coefficient
$e^{(h)}$: Differentiate center nodes and neighbors
$T_{\pm}$: Set of layers
V: Vertices
U: Eigenvector matrix
$\|p_i - q_j\|$: Space between feature vectors
E: Edges
l_θ: Linear projection
r_x, r_y: Image descriptors
$G(V, E)$: Graph function
$\beta^{(h)}$: MLP (2 layers)
S: Neighbor set vertex
L: Laplacian matrix
σ: Nonlinear activation function
(ρ, θ): Neighbor polar coordinates
D: Degree matrix
C: Feature matrix of curvature
$v_\rho(x, x')$: Interpolation weight

A: Adjacency matrix
k: Interval number
$(D(x)f)$: Mapping function
$\wedge$: Eigenvalue matrix
m: Maximum number
$h_{\alpha\theta 1}(x, y)$: Anisotropic heat kernel.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] P. Menghani, S. Barthwal, and S. Bansal, "An extreme helping hand for handicap people: using computer vision," in *Proceedings of the 2016 International Conference on Recent Advances and Innovations in Engineering, ICRAIE*, Jaipur, India, Decembe 2016.

[2] J. Huang, Y. Wei, J. Yi, and M. Liu, "An improved knn based on class contribution and feature weighting," in *Proceedings of the 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 313–316, Changsha, China, February 2018.

[3] D. Bajpai and L. He, "Evaluating KNN performance on WESAD dataset," in *Proceedings of the 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, Bhimtal, India, September 2020.

[4] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, August 2017.

[5] J. Zou, Y. Han, and S. S. So, "Overview of artificial neural networks," *Methods in Molecular Biology*, vol. 458, pp. 15–23, 2008.

[6] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[7] H. Li, J. Choi, S. Lee, and J. H. Ahn, "Comparing BERT and XLNet from the perspective of computational characteristics," in *Proceedings of the 2020 International Conference on Electronics, Information, and Communication (ICEIC)*, Barcelona, Spain, January 2020.

[8] M. Malik and R. Kamra, "A novel PV based ANN optimized converter for off grids locomotives," in *Proceedings of the 2021 International Conference on Technological Advancements and Innovations (ICTAI)*, Tashkent, Uzbekistan, November 2021.

[9] R. Bai, J. Zhao, D. Li, X. Lv, Q. Wang, and B. Zhu, "RNN-based demand awareness in smart library using CRFID," *China Communications*, vol. 17, no. 5, pp. 284–294, 2020.

[10] L. Han and Y. Shen, "Design of social media user satisfaction evaluation system from the perspective of Big data services," in *Proceedings of the 2021 International Conference on Big Data Analysis and Computer Science (BDACS)*, Kunming, China, June 2021.

[11] A. Toheed, M. H. Yousaf, and J. A. Rabnawaz, "Physical adversarial attack scheme on object detectors using 3D adversarial object," in *Proceedings of the 2022 2nd International Conference on Digital Futures and Transformative Technologies*, Rawalpindi, Pakistan, May 2022.

[12] S. Srhar, A. Arshad, and A. Raza, "Protien-DNA binding sites Prediction," in *Proceedings of the 4th International Conference on Innovative Computing ICIC*, Lahore, Pakistan, November 2021.

[13] M. A. Seif, R. Umeda, and H. Higa, "An attempt to control a 3D object in medical training system using leap motion," in *Proceedings of the ICIIBMS 2017 2nd International Conference on Intelligent Informatics and Biomedical Sciences*, Okinawa, Japan, November 2017.

[14] M. J. Blanchard, M. Reisch, and V. M. Yepes, "Swing: 2D and 3D animation in virtual reality," in *Proceedings of the 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, Lisbon, Portugal, March 2021.

[15] M. M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[16] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Santiago, Chile, USA, February 2015.

[17] P. Frasconi, M. Gori, and A. Sperduti, "A general framework for adaptive processing of data structures," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 768–786, 1998.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[20] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 729–734, Montreal, QC, Canada, July 2005.

[21] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Comput Soc Netw*, vol. 6, no. 1, pp. 11–23, 2019.

[22] H. Bagherinezhad, M. Rastegari, and A. Farhadi, "Lcnn: Lookup-based convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7120–7129, Honolulu, HI, USA, July 2017.

[23] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," *Adv Neural Inf Process Syst*, vol. 29, 2016.

[24] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[25] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, Montreal, QC, Canada, July 2017.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: deep hierarchical feature learning on point sets in a metric space,"

Advances in Neural Information Processing Systems, vol. 30, 2017.

[28] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "CayleyNets: graph convolutional neural networks with complex rational spectral filters," *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 97–109, 2019.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, Toulon, France, April 2016.

[30] Q. Li, S. Liu, L. Hu, and X. Liu, "Shape correspondence using anisotropic Chebyshev spectral CNNs," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 14646–14655, July 2020.

[31] P. K. Jayaraman, A. Sanghi, J. G. Lambourne et al., "UV-net: learning from boundary representations," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, June 2021.

[32] S. Arshad, M. Shahzad, Q. Riaz, and M. M. Fraz, "DPRNet: deep 3D point based residual network for semantic segmentation and classification of 3D point clouds," *IEEE Access*, vol. 7, pp. 68892–68904, 2019.

[33] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 945–953, IEEE Computer Society, Washington, DC, USA, 2015.

[34] L. Wei, Q. Huang, D. Ceylan, E. Vouga, and H. Li, "Dense human body correspondences using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1544–1553, Las Vegas, NV, USA, June 2016.

[35] Z. Wu, S. Song, A. Khosla et al., "3d shapenets: a deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, IEEE Computer Society, Washington, DC, USA, 2015.

[36] C. R. Qi, H. Su, M. Niebner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Washington, DC, USA, pp. 5648–5656, 2016.

[37] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[38] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2022.

[39] J. Zhou, G. Cui, S. Hu et al., "Graph neural networks: a review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[40] N. A. Asif, Y. Sarker, R. K. Chakrabortty et al., "Graph neural network: a comprehensive review on non-euclidean space," *IEEE Access*, vol. 9, Article ID 60588, 2021.

[41] W. Cao, Z. Yan, Z. He, and Z. He, "A comprehensive survey on geometric deep learning," *IEEE Access*, vol. 8, Article ID 35929, 2020.

[42] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3162–3177, 2020.

[43] J. Yu, C. Yu, C. Lin, and F. Wei, "Improved iterative closest point (ICP) point cloud registration algorithm based on matching point pair quadratic filtering," in *Proceedings of the 2021 International Conference on Computer, Internet of Things and Control Engineering, CITCE 2021*, pp. 1–5, Guangzhou, China, November 2021.

[44] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[45] Wikipedia, "B-Spline - wikipedia," 2022, https://en.wikipedia.org/wiki/B-spline.

[46] M. Ammad and A. Ramli, "Cubic B-spline curve interpolation with arbitrary derivatives on its data points," in *Proceedings of the 2019 23rd International Conference in Information Visualization - Part II*, pp. 156–159, Adelaide, SA, Australia, July 2019.

[47] W. Boehm, "Inserting new knots into B-spline curves," *Computer-Aided Design*, vol. 12, no. 4, pp. 199–201, 1980.

[48] H. Yang, W. Wang, and J. Sun, "Control point adjustment for B-spline curve approximation," *Computer-Aided Design*, vol. 36, no. 7, pp. 639–652, 2004.

[49] Towards data science, "Understanding regularization in machine learning | by ashu prasad | towards data science," 2023, https://towardsdatascience.com/understanding-regularization-in-machine-learning-d7dd0729dde5.

[50] M. Xu, W. Dai, C. Li, J. Zou, H. Xiong, and P. Frossard, "Graph neural networks with lifting-based adaptive graph wavelets," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 63–77, 2022.

[51] W. T. Tutte and W. T. Tutte, *Graph Theory*, Vol. 21, Cambridge University Press, Cambridge, UK, 2001.

[52] T. Wang, B. Li, J. Zou, F. Sun, and Z. Zhang, "New bounds of the Nordhaus-Gaddum type of the Laplacian matrix of graphs," in *Proceedings of the 2012 8th International Conference on Computational Intelligence and Security, CIS 2012*, pp. 411–414, Guangzhou, China, November 2012.

[53] R. B. Bapat, S. J. Kirkland, and S. Pati, "The perturbed laplacian matrix of a graph," *Linear and Multilinear Algebra*, vol. 49, no. 3, pp. 219–242, 2001.

[54] J. Domingos and J. M. F. Moura, "Graph fourier transform: a stable approximation," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4422–4437, 2020.

[55] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," 2023, https://proceedings.mlr.press/v48/niepert16.html.

[56] S. Fiori, "Manifold calculus in system theory and control—fundamentals and first-order systems," *Symmetry*, vol. 13, no. 11, p. 2092, 2021.

[57] F. Demengel and J. M. Ghidaglia, "Inertial manifolds for partial differential evolution equations under time-discretization: existence, convergence, and applications," *Journal of Mathematical Analysis and Applications*, vol. 155, no. 1, pp. 177–225, 1991.

[58] Analytics Vidhya, "What are Graph Neural Networks, and how do they work?," 2022, https://www.analyticsvidhya.com/blog/2022/03/what-are-graph-neural-networks-and-how-do-they-work/.

[59] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: extending high-dimensional data analysis to

networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[60] Y. Ma, J. Hao, Y. Yang, H. Li, J. Jin, and G. Chen, "Spectral-based Graph Convolutional Network for Directed Graphs," 2019, https://arxiv.org/abs/1907.08990.

[61] K. Kochetov and E. Putin, "SpecNN: the specifying neural network," in *Proceedings of the 2016 International Symposium on INnovations in Intelligent SysTems and Applications, INISTA 2016*, Sinaia, Romania, August 2016.

[62] Z. L. Wang, X. O. Song, and X. R. Wang, "Spectrum sensing detection algorithm based on eigenvalue variance," in *Proceedings of the conference 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference, ITAIC 2019*, pp. 1656–1659, Chongqing, China, May 2019.

[63] A. A. M. Muzahid, W. Wan, F. Sohel, L. Wu, and L. Hou, "CurveNet: curvature-based multitask learning deep networks for 3D object recognition," *IEEE/CAA Journal of Automatica Sinica*.vol. 8, no. 6, pp. 1177–1187, 2021.

[64] W. Zhang, P. Tang, L. Zhao, and Q. Huang, "A comparative study of U-nets with various convolution components for building extraction," in *Proceedings of the 2019 Joint Urban Remote Sensing Event (JURSE) 2019*, Vannes, France, May 2019.

[65] B. Vallet and B. Lévy, "Spectral geometry processing with manifold harmonics," *Computer Graphics Forum*, vol. 27, no. 2, pp. 251–260, 2008.

[66] M. Andreux, E. Rodolá, M. Aubry, and D. Cremers, "Anisotropic laplace-beltrami operators for shape analysis," *Lecture Notes in Computer Science*, vol. 8928, pp. 299–312, 2015.

[67] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014*, Scottsdale, AZ, Maricopa, December 2013.

[68] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[69] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5115–5124, Hangzhou, China, December 2017.

[70] M. Fey, J. E. Lenssen, F. Weichert, and H. Müller, "Splinecnn: fast geometric deep learning with continuous b-spline kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 869–877, IEEE Computer Society, Washington, DC, USA, 2018.

[71] L. Xiao, X. Wu, and G. Wang, "Social network analysis based on graph SAGE," in *Proceedings of the 2019 12th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, December 2019.

[72] B. Zhang, H. Zeng, and V. Prasanna, "Hardware acceleration of large scale GCN inference," in *Proceedings of the 2020 IEEE 31st International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, Manchester, UK, July 2020.

[73] X. Gao and H. Xiong, "A hybrid wavelet convolution network with sparse-coding for image super-resolution," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP) 2016*, pp. 1439–1443, Phoenix, AZ, USA, August 2016.

[74] K. Lazri, A. Blin, J. Sopena, and G. Muller, "Toward an in-kernel high performance key-value store implementation," in *Proceedings of the 2019 38th Symposium on Reliable Distributed Systems (SRDS)*, p. 268, Lyon, France, October 2019.

[75] T. Sederberg, "Computer aided geometric design," 2012, https://scholarsarchive.byu.edu/facpub/1.

[76] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks," in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April 2018.

[77] D. Xu, D. Zheng, F. Chen, E. Korpeoglu, S. Kumar, and K. Achan, "Studying the effect of carrier type on the perception of vocoded stimuli via mismatch negativity," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 2019, pp. 3167–3170, Berlin, Germany, July 2019.

[78] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "ShapeNet: Convolutional Neural Networks on Non-euclidean Manifolds," 2015, http://infoscience.epfl.ch/record/204949.

[79] D. Chicco, "Siamese neural networks: an overview," *Artificial Neural Networks. Methods in Molecular Biology*, Vol. 2190, Humana, New York, NY, USA, 2021.

[80] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.

[81] J. Donahue, Y. Jia, O. Vinyals et al., "Decaf: a deep convolutional activation feature for generic visual recognition," in *Proceedings of the International Conference on Machine Learning PMLR*, pp. 647–655, Beijing, China, June 2014.

[82] ModelNet, "Princeton ModelNet," 2022, https://modelnet.cs.princeton.edu/.

[83] A. Nguyen and B. Le, "3D point cloud segmentation: a survey," in *Proceedings of the 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 225–230, Manila, Philippines, November 2013.

[84] Y. Ishikawa, R. Hachiuma, N. Ienaga, W. Kuno, Y. Sugiura, and H. Saito, "Semantic segmentation of 3D point cloud to virtually manipulate real living space," in *Proceedings of the 2019 12th Asia Pacific Workshop on Mixed and Augmented Reality, APMAR 2019*, Ikoma, Japan, March 2019.

[85] Y. Li, L. Ma, Z. Zhong, D. Cao, and J. Li, "TGNet: geometric graph CNN on 3-D point cloud segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3588–3600, 2020.

[86] M. Yuan, Y. Chen, H. Guan, S. Shi, X. Wei, and B. Li, "An improved analytical probabilistic load flow method with GMM models of correlated der Generation," in *Proceedings of the 2020 IEEE Sustainable Power and Energy Conference (iSPEC)*, pp. 73–78, Chengdu, China, November 2020.

[87] L. Lebrat, R. S. Cruz, C. Fookes, and O. Salvado, "MongeNet: efficient sampler for geometric deep learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*IEEE Computer Society, Washington, DC, USA, Article ID 16664, 2021.

[88] K. Jothiramalingam, N. R. Dhineshbabu, and S. R. Srither, "Study the Complex Dielectric and Energy loss function of Graphene oxide nanostructures using linear optical constant," in *Proceedings of the 2018 Conference on Emerging Devices and Smart Systems (ICEDSS)*, pp. 209–212, Tiruchengode, India, March 2018.

[89] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," *Lecture Notes in Computer Science*, vol. 3952, pp. 589–600, Springer, Berlin, Germany, 2006.

[90] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[91] Faust Dataset, "MPI Faust Dataset," 2023, https://faust-leaderboard.is.tuebingen.mpg.de/.

[92] Stanford, "Drago Anguelov," 2023, http://ai.stanford.edu/drago/Projects/projects.html.

[93] Tosca, "OpenDataLab," 2023, https://opendatalab.com/TOSCA.

[94] ModelNet, "Princeton ModelNet," 2023, https://modelnet.cs.princeton.edu/.

[95] R. G. Schneider and T. Tuytelaars, "Sketch classification and classification-driven analysis using Fisher vectors," *ACM Transactions on Graphics*, vol. 33, no. 6, pp. 1–9, 2014.

[96] D. Angela, X. C. Angel, S. Manolis, H. Maciej, F. Thomas, and N. Matthias, "ScanNet | richly-annotated 3D reconstructions of indoor scenes," 2023, http://www.scan-net.org/.

[97] Stanford, "S3DIS dataset | papers with code," 2023, https://paperswithcode.com/dataset/s3dis.

[98] X. Roynard, J. E. Deschaud, and F. Goulette, "Paris-Lille-3D: a large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification," *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 545–557, 2018.

[99] Yann, "MNIST handwritten digit database, yann LeCun, corinna cortes and chris burges," 2023, http://yann.lecun.com/exdb/mnist/.

[100] Cora, "Dataset," 2023, https://relational.fit.cvut.cz/dataset/CORA.

[101] NCBI, "PubMed," 2023, https://pubmed.ncbi.nlm.nih.gov/.

[102] M. Graña, M. A. Veganzons, and B. Ayerdi, "Hyperspectral Remote sensing scenes - grupo de Inteligencia computacional (GIC)," 2023, https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

[103] Pavia University, "V7 Open Datasets," 2023, https://www.v7labs.com/open-datasets/pavia-university.

[104] Kennedy Space Center, "V7 Open Datasets," 2023, https://www.v7labs.com/open-datasets/kennedy-space-center.

[105] Z. Zhang, P. Jaiswal, and R. Rai, "FeatureNet: machining feature recognition based on 3D convolution neural network," *Computer-Aided Design*, vol. 101, pp. 12–22, 2018.

[106] A. Angrish, E. P. Fitts, B. Craver, and B. Starly, "FabSearch": a 3D CAD model based search engine for sourcing manufacturing services," 2018, https://arxiv.org/abs/1809.06329.

[107] W. Cao, T. Robinson, Y. Hua, F. Boussuge, A. R. Colligan, and W. Pan, "Graph representation of 3D CAD models for machining feature recognition with deep learning," in *Proceedings of the ASME Design Engineering Technical Conference*, pp. 11A–2020A, St Louis Missouri, USA, November 2020.

[108] S. Koch, A. Matveev, Z. Jiang et al., "ABC: a Big CAD model dataset for geometric deep learning," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9593–9603, Long Beach, CA, USA, June 2019.

[109] Y. Wang, T. Zhou, Z. Li, H. Huang, and B. Qu, "Salient object detection based on multi-feature graphs and improved manifold ranking," *Multimedia Tools and Applications*, vol. 81, Article ID 27551, 2022.

[110] P. Zhang, G. Chen, Y. Li, and W. Dong, "Agile Formation control of drone flocking enhanced with active vision-based relative localization," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6359–6366, 2022.

[111] T. Hempel and A. Al-Hamadi, "An online semantic mapping system for extending and enhancing visual SLAM," *Engineering Applications of Artificial Intelligence*, vol. 111, Article ID 104830, 2022.

[112] P. Antonante, V. Tzoumas, H. Yang, and L. Carlone, "Outlier-robust estimation: hardness, minimally tuned algorithms, and applications," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 281–301, 2022.

[113] P. Dhiman, V. Kukreja, P. Manoharan et al., "A novel deep learning model for detection of severity level of the Disease in citrus fruits," *Electronics*, vol. 11, p. 495, 2022.

[114] L. Jiang, W. Nie, J. Zhu, X. Gao, and B. Lei, "Lightweight object detection network model suitable for indoor mobile robots," *Journal of Mechanical Science and Technology*, vol. 36, no. 2, pp. 907–920, 2022.

[115] M. A. Ansari, M. Meraz, P. Chakraborty, and M. Javed, "Angle-based feature learning in GNN for 3D object detection using point cloud," *Lecture Notes in Electrical Engineering*, vol. 858, pp. 419–432, 2022.

[116] X. Liu, X. Jing, Z. Zheng, W. Du, X. Ding, and Q. Zhu, "A symmetric fusion learning model for detecting visual relations and scene parsing," *Scientific Programming*, vol. 2022, pp. 1–9, 2022.

[117] P. Jiang, X. Deng, L. Wang, Z. Chen, and S. Zhang, "Hypergraph representation for detecting 3D objects from noisy point clouds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1, 2022.

[118] C. Xia, X. Gao, X. Fang, K. C. Li, S. Su, and H. Zhang, "RLP-AGMC: robust label propagation for saliency detection based on an adaptive graph with multiview connections," *Signal Processing: Image Communication*, vol. 98, Article ID 116372, 2021.

[119] R. Ji, Z. Liu, L. Zhang et al., "Multi-peak graph-based multi-instance learning for weakly supervised object detection," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[120] H. Wang, C. Zhu, J. Shen, Z. Zhang, and X. Shi, "Salient object detection by robust foreground and background seed selection," *Computers & Electrical Engineering*, vol. 90, Article ID 106993, 2021.

[121] K. Meier, S. J. Chung, and S. Hutchinson, "River segmentation for autonomous surface vehicle localization and river boundary mapping," *Journal of Field Robotics*, vol. 38, no. 2, pp. 192–211, 2021.

[122] N. Gothoskar, M. Cusumano-Towner, B. Zinberg et al., "3DP3: 3D scene perception via probabilistic programming," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9600–9612, 2021.

[123] B. Mao and B. Li, "City object detection from airborne Lidar data with OpenStreetMap-tagged superpixels," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 23, Article ID e6026, 2020.

[124] B. Santra, A. K. Shaw, and D. P. Mukherjee, "Graph-based non-maximal suppression for detecting products on the rack," *Pattern Recognition Letters*, vol. 140, pp. 73–80, 2020.

[125] R. Ren, H. Fu, X. Hu, H. Xue, X. Li, and M. Wu, "Towards efficient and robust LiDAR-based 3D mapping in urban environments," in *Proceedings of the 2020 3rd International Conference on Unmanned Systems (ICUS)*, pp. 511–516, Harbin, China, November 2020.

[126] J. Poschmann, T. Pfeifer, and P. Protzel, "Factor graph based 3D multi-object tracking in point clouds," in *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Article ID 10343, Las Vegas, NV, USA, October 2020.

[127] M. Zhang, J. Chen, P. Li, M. Jiang, and Z. Zhou, "Topic scene graphs for image captioning," *IET Computer Vision*, vol. 16, no. 4, pp. 364–375, 2022.

[128] X. Li, B. Wu, J. Song, L. Gao, P. Zeng, and C. Gan, "Text-instance graph: exploring the relational semantics for text-based visual question answering," *Pattern Recognition*, vol. 124, Article ID 108455, 2022.

[129] W. Gao, L. Yu, Y. Du, and S. Lu, "Single image 3D reconstruction based on attention mechanism and graph convolution network," in *Proceedings of the 8th International Conference on Computing and Artificial Intelligence*, vol. 18, pp. 670–676, Tianjin, China, March 2022.

[130] B. Guldur Erkal and J. F. Hajjar, "Using extracted member properties for laser-based surface damage detection and quantification," *Structural Control and Health Monitoring*, vol. 27, no. 11, Article ID e2616, 2020.

[131] J. Li, J. Yi, L. Hua et al., "Development and validation of a predictive score for venous thromboembolism in newly diagnosed non-small cell lung cancer," *Thrombosis Research*, vol. 208, pp. 45–51, 2021.

[132] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: leveraging deep depth prediction for monocular direct sparse odometry," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 817–833, Munich, Germany, September 2018.

[133] T. Graph, "Graph convolutional enhanced discriminative broad learning system for hyperspectral image classification," *IEEE Access*, vol. 10, Article ID 90299, 2022.

[134] M. Zeng, Y. Cai, X. Liu, Z. Cai, and X. Li, "Spectral-spatial clustering of hyperspectral image based on laplacian regularized deep subspace clustering," in *Proceedings of the IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2694–2697, Yokohama, Japan, July 2019.

[135] H. Wu, Y. Yan, Y. Ye, M. K. Ng, and Q. Wu, "Geometric Knowledge Embedding for unsupervised domain adaptation," *Knowledge-Based Systems*, vol. 191, Article ID 105155, 2020.

[136] Q. Yang, Y. Xu, Z. Wu, and Z. Wei, "Hyperspectral and multispectral image fusion based on deep attention network," in *Proceedings of the 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Amsterdam, Netherlands, September 2019.

[137] H. Fang, A. Li, T. Wang, H. Chang, and H. Xu, "Hyperspectral Unmixing Using Graph-Regularized and Sparsity-Constrained Deep NMF," in *Proceedings of the Volume 10846, Optical Sensing and Imaging Technologies and Applications*, pp. 664–672, Beijing, China, December 2018.

[138] Z. Lin, F. Zhu, Q. Wang et al., "RSSGG_CS: Remote sensing image scene graph generation by fusing contextual information and statistical knowledge," *Remote Sensing*, vol. 14, p. 3118, 2022.

[139] A. Bessadok, M. A. Mahjoub, and I. Rekik, "Brain graph synthesis by dual adversarial domain alignment and target graph prediction from a source graph," *Medical Image Analysis*, vol. 68, Article ID 101902, 2021.

[140] F. P. Kalaganis, N. A. Laskaris, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "A data augmentation scheme for geometric deep learning in personalized brain–computer interfaces," *IEEE Access*, vol. 8, Article ID 162218, 2020.

[141] M. B. Gurbuz and I. Rekik, "Deep graph normalizer: a geometric deep learning approach for estimating connectional brain templates," *Lecture Notes in Computer Science*, vol. 12267, pp. 155–165, Springer Cham, New York, NY, USA, 2020.

[142] V. Vosylius, A. Wang, C. Waters et al., "Geometric deep learning for post-menstrual age prediction based on the neonatal white matter cortical surface," *Lecture Notes in Computer Science*, vol. 12443, pp. 174–186, Springer Cham, New York, NY, USA, 2020.

[143] Z. Huang, H. Cai, T. Dan, Y. Lin, P. Laurienti, and G. Wu, "Detecting brain state changes by geometric deep learning of functional dynamics on riemannian manifold," *Lecture Notes in Computer Science*, vol. 12907, pp. 543–552, Springer Cham, New York, NY, USA, 2021.

[144] A. Bessadok, M. A. Mahjoub, and I. Rekik, "Dual adversarial connectomic domain alignment for predicting isomorphic brain graph from a baseline graph," *Lecture Notes in Computer Science*, vol. 11767, pp. 465–474, Springer Cham, New York, NY, USA, 2019.

[145] Z. Zhen, Y. Chen, I. Segovia-Dominguez, and Y. R. Gel, "Tlife-GDN: detecting and forecasting spatio-temporal anomalies via persistent homology and geometric deep learning," *Lecture Notes in Computer Science*, vol. 13281, pp. 511–525, Springer Cham, New York, NY, USA, 2022.

[146] L. Hansen, D. Dittmer, and M. P. Heinrich, "Learning deformable point set registration with regularized dynamic graph CNNs for large lung motion in COPD patients," *Lecture Notes in Computer Science*, vol. 11849, pp. 53–61, Springer Cham, New York, NY, USA, 2019.

[147] V. C. Gogineni, S. R. E. Langberg, V. Naumova et al., "Data-driven personalized cervical cancer risk prediction: a graph-perspective," in *Proceedings of the 2021 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 46–50, Rio de Janeiro, Brazil, July 2021.

[148] B. Das, M. Kutsal, and R. Das, "A geometric deep learning model for display and prediction of potential drug-virus interactions against SARS-CoV-2," *Chemometrics and Intelligent Laboratory Systems*, vol. 229, Article ID 104640, 2022.

[149] T. Dissanayake, T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Geometric deep learning for subject independent epileptic seizure prediction using scalp EEG signals," *IEEE J Biomed Health Inform*, vol. 26, no. 2, pp. 527–538, 2022.

[150] L. Z. J. Williams, A. Fawaz, M. F. Glasser, A. D. Edwards, and E. C. Robinson, "Geometric deep learning of the human connectome project multimodal cortical parcellation," *Lecture Notes in Computer Science*, vol. 13001, pp. 103–112, Springer Cham, New York, NY, USA, 2021.

[151] B. Zhao, X. Zhang, Z. Zhan, Q. Wu, and H. Zhang, "Multiscale graph-guided convolutional network with node attention for intelligent health state diagnosis of a 3-PRR planar parallel manipulator," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 11, Article ID 11733, 2022.

[152] F. Bagheri, M. J. Tarokh, and M. Ziaratban, "Skin lesion segmentation from dermoscopic images by using Mask R-

CNN, Retina-Deeplab, and graph-based methods," *Biomedical Signal Processing and Control*, vol. 67, Article ID 102533, 2021.

[153] L. Yang, X. Zhan, D. Chen, J. Yan, C. C. Loy, and D. Lin, "Learning to cluster faces on an affinity graph," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2298–2306, Long Beach, CA, USA, June 2019.

[154] X. Wu, X. Ma, J. Zhang, and Z. Jin, "Salient object detection via reliable boundary seeds and saliency refinement," *IET Computer Vision*, vol. 13, no. 3, pp. 302–311, 2019.

[155] F. Albalas, A. Alzu'Bi, A. Alguzo, T. Al-Hadhrami, and A. Othman, "Learning discriminant spatial features with deep graph-based convolutions for occluded face detection," *IEEE Access*, vol. 10, pp. 35162–35171, 2022.

[156] N. Dall'Asen, Y. Wang, H. Tang, L. Zanella, and E. Ricci, "Graph-based generative face anonymisation with pose preservation," *Lecture Notes in Computer Science*, Springer Cham, vol. 13232, pp 503–515, New York, NY, USA, 2022.

[157] S. Nayak, A. Routray, M. Sarma, and S. Uttarkabat, "GNN based embedded framework for consumer affect recognition using thermal facial ROIs," *IEEE Consumer Electronics Magazine*, 2022.

[158] P. Karuppanan and K. Dhanalakshmi, "Criminal persons recognition using improved feature extraction based local phase quantization," *Intelligent Automation & Soft Computing*, vol. 33, no. 2, pp. 1025–1043, 2022.

[159] Z. Shao, Y. Zhou, B. Liu, H. Zhu, W. L. Du, and J. Zhao, "Facial action unit detection via hybrid relational reasoning," *The Visual Computer*, vol. 38, no. 9, pp. 3045–3057, 2022.

[160] A. Alguzo, A. Alzu'Bi, and F. Albalas, "Masked face detection using multi-graph convolutional networks," in *Proceedings of the 2021 12th International Conference on Information and Communication Systems (ICICS)*, pp. 385–391, Valencia, Spain, May 2021.

[161] V. B. Kamath, S. S. Pai, S. S. Poojary, A. Rastogi, and K. S. Srinivas, "Drunkenness face detection using graph neural networks," in *Proceedings of the CSITSS 2021 5th International Conference on Computational Systems and Information Technology for Sustainable Solutions*, Bangalore, India, December 2021.

[162] Z. Lu, C. Zhou, X. Xuyang, and W. Zhang, "Face detection and recognition method based on improved convolutional neural network," *International Journal of Circuits, Systems and Signal Processing*, vol. 15, pp. 774–781, 2021.

[163] J. Bai, W. Yu, Z. Xiao et al., "Two-stream spatial-temporal graph convolutional networks for driver drowsiness detection," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, Article ID 13821, 2022.

[164] Z. Wen, "Large-scale face clustering method research based on deep learning," in *Proceedings of the 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence, MLBDBI*, pp. 731–734, Taiyuan, China, December 2021.

[165] Z. Liu, J. Dong, C. Zhang, L. Wang, and J. Dang, "Relation modeling with graph convolutional networks for facial action unit detection," *Lecture Notes in Computer Science*, Springer Cham, vol. 11962, pp 489–501, New York, NY, USA, 2020.

[166] X. Xu, Z. Ruan, and L. Yang, "Facial expression recognition based on graph neural network," in *Proceedings of the 2020 IEEE 5th International Conference on Image, Vision and Computing, ICIVC 2020*, pp. 211–214, Beijing, China, July 2020.

[167] B. Zhang, J. Wan, Y. Zhao, Z. Tong, and Y. Du, "Multi-actor activity detection by modeling object relationships in extended videos based on deep learning," *Engineering Applications of Artificial Intelligence*, vol. 114, Article ID 105055, 2022.

[168] Y. Han, T. Zhuo, P. Zhang et al., "One-shot video graph generation for explainable action reasoning," *Neurocomputing*, vol. 488, pp. 212–225, 2022.

[169] J. N. Zaech, A. Liniger, D. Dai, M. Danelljan, and L. van Gool, "Learnable online graph representations for 3D multi-object tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5103–5110, 2022.

[170] B. Xie, Y. Deng, Z. Shao, H. Liu, and Y. Li, "VMV-GCN: volumetric multi-view based graph CNN for event stream classification," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1976–1983, 2022.

[171] Z. Xiong, H. Ma, Y. Wang, T. Hu, and Q. Liao, "LiDAR-based 3D video object detection with foreground context modeling and spatiotemporal graph reasoning," in *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC) 2021*, pp. 2994–3001, Indianapolis, IN, USA, September 2021.

[172] J. Zhan, H. Zhao, P. Zheng, H. Wu, and L. Wang, "Salient superpixel visual tracking with graph model and iterative segmentation," *Cognitive Computation*, vol. 13, no. 4, pp. 821–832, 2019.

[173] N. Gkalelis, A. Goulas, D. Galanopoulos, and V. Mezaris, "Objectgraphs: using objects and a graph convolutional network for the bottom-up recognition and explanation of events in video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3375–3383, Nashville, TN, USA, June 2021.

[174] W. Zhang and F. Meyer, "Graph-based multiobject tracking with embedded particle flow," in *Proceedings of the 2021 IEEE Radar Conference (RadarConf21) 2021*, Atlanta, GA, USA, May 2021.

[175] G. Jung, J. Lee, and I. Kim, "Tracklet pair proposal and context reasoning for video scene graph generation," *Sensors*, vol. 21, p. 3164, 2021.

[176] M. Tomei, L. Baraldi, S. Calderara, S. Bronzin, and R. Cucchiara, "Video action detection by learning graph-based spatio-temporal interactions," *Computer Vision and Image Understanding*, vol. 206, Article ID 103187, 2021.

[177] Y. Yin, X. Feng, and H. Wu, "Learning for graph matching based multi-object tracking in auto driving," *J Phys Conf Ser*, vol. 1871, no. 1, Article ID 012152, 2021.

[178] S. A. Kazakova, P. A. Leonteva, M. I. Frolova, J. V. Donetskaya, I. Y. Popov, and A. Y. Kuznetsov, "A study of human motion in computer vision systems based on a skeletal model," *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, vol. 21, no. 4, pp. 571–577, 2021.

[179] Y. M. Seo and Y. S. Choi, "Graph convolutional networks for skeleton-based action recognition with LSTM using tool-information," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pp. 986–993, Virtual Event, Korea, March 2021.

[180] M. Xu, B. Liu, P. Fu, J. Li, Y. H. Hu, and S. Feng, "Video salient object detection via robust seeds extraction and multi-graphs manifold propagation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2191–2206, 2019.

[181] M. Xu, B. Liu, P. Fu, J. Li, and Y. H. Hu, "Video saliency detection via graph clustering with motion energy and

spatiotemporal objectness," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2790–2805, 2019.

[182] Y. Zhang, H. Sheng, Y. Wu, S. Wang, W. Ke, and Z. Xiong, "Multiplex labeling graph for near-online tracking in crowded scenes," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7892–7902, 2020.

[183] Z. Su, Y. Wang, Q. Xie, and R. Yu, "Pose graph parsing network for human-object interaction detection," *Neurocomputing*, vol. 476, pp. 53–62, 2022.

[184] L. He and J. Zhang, "Railway driver behavior recognition system based on deep learning algorithm," in *Proceedings of the 2021 4th International Conference on Artificial Intelligence and Big Data, ICAIBD 2021*, pp. 398–403, Chengdu, China, May 2021.

[185] Y. Tang, B. Wang, W. He, and F. Qian, "PointDet: an object detection framework based on human local features in the task of identifying violations," in *Proceedings of the 2021 11th International Conference on Information Science and Technology (ICIST)*, pp. 673–680, Chengdu, China, May 2021.

[186] S. Zheng, S. Chen, and Q. Jin, "Skeleton-based interactive graph network for human object interaction detection," in *Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME). 2020*, London, UK, July 2020.

[187] G. Ning, J. Pei, and H. Heng, "Lighttrack: A generic framework for online top-down human pose tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1034-1035, Seattle, WA, USA, June 2020.

[188] J. Jiahao Lin and G. Hee Lee, "Learning spatial context with graph neural network for multi-person pose grouping," in *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA). 2021*, pp. 4230–4236, Xi'an, China, May 2021.

[189] T. Cui, W. Song, G. An, and Q. Ruan, "Prototype generation based shift graph convolutional network for semi-supervised anomaly detection," *Communications in Computer and Information Science*, Springer, vol. 1480, pp 159–169, Singapore, 2021.

[190] C. Zhang and H. He, "Research on Pose Recognition Algorithm for Sports Players Based on Machine Learning of Sensor Data," *Security and Communication Networks*, vol. 2021, 2021.

[191] D. Yang and Y. Zou, "A graph-based interactive reasoning for human-object interaction detection," in *Proceedings of the IJCAI International Joint Conference on Artificial Intelligence*, pp. 1111–1117, Messe Wien, Vienna, Austria, January 2020.

[192] P. R. K. Pranav, S. Verma, S. Shenoy, and S. Saravanan, "Detection of botnets in IoT networks using graph theory and machine learning," in *Proceedings of the 2022 6th International Conference on Trends in Electronics and Informatics, ICOEI 2022*, pp. 590–597, Tirunelveli, India, April 2022.

[193] I. Papakis, A. Sarkar, and A. Karpatne, "A graph convolutional neural network based approach for traffic monitoring using augmented detections with optical flow," in *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC) 2021*, pp. 2980–2986, Indianapolis, IN, USA, September 2021.

[194] F. Zhou, S. Chen, J. Wu, C. Cao, and S. Zhang, "Trajectory-user linking via graph neural network," in *Proceedings of the ICC 2021 - IEEE International Conference on Communications*, Montreal, QC, Canada, June 2021.

[195] A. A. Liu, Y. Wang, N. Xu, S. Liu, and X. Li, "Scene-Graph-Guided message passing network for dense captioning," *Pattern Recognition Letters*, vol. 145, pp. 187–193, 2021.

[196] I. U. Hewapathirana, D. Lee, E. Moltchanova, and J. McLeod, "Change detection in noisy dynamic networks: a spectral embedding approach," *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 14–22, 2020.

[197] A. Paul, M. Shamsul Arefin, and R. Karim, "Developing a Framework for Analyzing Heterogeneous Data from Social Networks," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, 2019.

[198] Z. Yuan, Q. Yuan, and J. Wu, "Phishing detection on ethereum via learning representation of transaction subgraphs," *Communications in Computer and Information Science*, vol. 1267, pp. 178–191, 2020.

[199] Y. Ren, B. Wang, J. Zhang, and Y. Chang, "Adversarial active learning based heterogeneous graph neural network for fake news detection," in *Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM) 2020*, pp. 452–461, Sorrento, Italy, November 2020.

[200] D. Zhu, Y. Ma, and Y. Liu, "A flexible attentive temporal graph networks for anomaly detection in dynamic networks," in *Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020*, pp. 870–875, Guangzhou, China, December 2020.

[201] R. Wen, J. Wang, C. Wu, and J. Xiong, "ASA: adversary situation awareness via heterogeneous graph convolutional networks," in *Proceedings of the Web Conference 2020 - Companion of the World Wide Web Conference, WWW 2020*, pp. 674–678, Taipei, Taiwan, April 2020.

[202] C. Shen, X. Zhao, X. Fan et al., "Multi-receptive field graph convolutional neural networks for pedestrian detection," *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1319–1328, 2019.

[203] J. Xie, Y. Pang, H. Cholakkal, R. Anwer, F. Khan, and L. Shao, "PSC-Net: learning part spatial co-occurrence for occluded pedestrian detection," *Science China Information Sciences*, vol. 64, no. 2, Article ID 120103, 2020.

[204] A. K. F. Lui, Y. H. Chan, and M. F. Leung, "Modelling of destinations for data-driven pedestrian trajectory prediction in public buildings," in *Proceedings of the 2021 IEEE International Conference on Big Data, Big Data 2021*, pp. 1709–1717, Orlando, FL, USA, December 2021.

[205] A. K. F. Lui, Y. H. Chan, and M. F. Leung, "Modelling of pedestrian movements near an amenity in walkways of public buildings," in *Proceedings of the 2022 8th International Conference on Control, Automation and Robotics, ICCAR 2022*, pp. 394–400, Xiamen, China, April 2022.

[206] Q. Li, Z. Han, and X. M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.

[207] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: can gcns go as deep as cnns?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9267–9276, Seoul, Korea, October 2019.

[208] M. Looks, M. Herreshoff, D. L. Hutchins, and P. Norvig, "Deep learning with dynamic computation graphs," in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, February 2017.

[209] Y. Ma, Z. Guo, Z. Ren, J. Tang, and D. Yin, "Streaming graph neural networks," in *Proceedings of the 43rd International ACM SIGIR Conference On Research And Development In*

Information Retrieval. SIGIR '20, pp. 719–728, Association for Computing Machinery, Xi'an, China, July 2020.

[210] F. Manessi, A. Rozza, and M. Manzo, "Dynamic graph convolutional networks," *Pattern Recognition*, vol. 97, Article ID 107000, 2020.

[211] J. Chen, J. Zhu, and L. Song, "Stochastic training of graph convolutional networks with variance reduction," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, vol. 3, pp. 1503–1532, Stockholm, Sweden, July 2018.

[212] J. Chen, T. Ma, and X. C. FastGCN, "Fast learning with graph convolutional networks via importance sampling," in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, January 2018.

[213] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[214] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, vol. 18, pp. 1416–1424, Washington DC, USA, August 2018.