Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.5654

Adapting Convolutions for Effective Omnidirectional Image Processing

Romulo Marconato Stringhini [**Federal University of Rio Grande do Sul** | ***rmstringhini@inf.ufrgs.br***]
Thiago Lopes Trugillo da Silveira [**Federal University of Santa Maria** | ***thiago@inf.ufsm.br***]
Claudio Rosito Jung [**Federal University of Rio Grande do Sul** | ***crjung@inf.ufrgs.br***]

Institute of Informatics, Federal University of Rio Grande do Sul, Av. Bento Gonçalves, 9500, Agronomia, Porto Alegre, RS, 91509-900, Brazil.

Received: 07 March 2025 • **Accepted:** 13 November 2025 • **Published:** 29 May 2026

Abstract Omnidirectional images present unique challenges for traditional convolutional neural networks (CNNs) due to the non-uniform sampling inherent in the equirectangular projection (ERP). This projection introduces distortions, especially near the poles, and conventional fixed-size kernels in planar CNNs are not designed to handle these distortions effectively. To address this issue, we previously introduced the Spherically-Weighted Horizontally Dilated Convolutions (SWHDC) module which adjusts dilated convolutions by applying appropriate weights to account for the varying sampling density across ERP image rows. In this extended work, we provide a more comprehensive evaluation of the SWHDC module by benchmarking it against several state-of-the-art methods on the 3D object classification task. Additionally, we integrate SWHDCs into different backbones to further investigate their effectiveness in tackling the gravity alignment problem. Experimental results confirm that our approach not only improves classification accuracy but also enhances gravity alignment performances without increasing the amount of trainable parameters of the baseline backbones. These findings further establish SWHDC as a robust alternative for processing omnidirectional images in different visual computing applications.

Keywords: Omnidirectional Images, Convolutional Neural Network, Distortions, Dilated Convolution, Object Classification, Gravity Alignment

1 Introduction

Omnidirectional images, also referred to as spherical, 360-degree, or panoramic images, have grown in popularity thanks to advancements in 3D technology, the increasing affordability of capture devices, and the pursuit of more immersive experiences [Ai *et al*., 2022; da Silveira and Jung, 2023]. Unlike traditional 2D planar images, omnidirectional images are defined on the surface of a unit sphere, providing a $360° \times 180°$ field of view (FoV) and three degrees of freedom (3-DoF) [da Silveira *et al*., 2022]. As a result, omnidirectional imagery has found widespread applications in areas such as augmented and virtual reality, autonomous driving, and robotic navigation [Guo *et al*., 2020; Bello *et al*., 2020; da Silveira *et al*., 2022; da Silveira and Jung, 2023].

Spherical images are commonly projected to a 2D plane using the equirectangular projection (ERP) due to its simple relation between rectangular and spherical coordinates [Coors *et al*., 2018; Eder *et al*., 2020; Ai *et al*., 2022; da Silveira and Jung, 2023]. However, this representation maps the spherical surface to the plane through non-uniform sampling, introducing distortions, where areas near the poles are more densely sampled than those near the equator [da Silveira and Jung, 2023]. Consequently, standard convolutional neural networks (CNNs) designed for planar images are not the best choice to process omnidirectional images, as they rely on fixed-support convolutional kernels that do not account for spherical distortions [da Silveira and Jung, 2023; Ai *et al*., 2022].

Many deep learning approaches, ranging from CNNs to transformers, are designed to address the non-uniform sampling (distortion) in ERP. For instance, works such as KTN [Su and Grauman, 2019] and SphereNet [Coors *et al*., 2018] improve the feature extraction capabilities of CNNs by employing spherical convolutions. However, these methods demand significant computational resources, as kernels designed to handle irregular sampling are typically slower than standard convolutions [da Silveira and Jung, 2023]. Dilated convolutions [Yu and Koltun, 2015; Dai et al., 2020; Zhuang et al., 2022; Lee and Park, 2022] address distortions in 360-degree images due to their larger receptive field and ability to capture long-range dependencies. However, because the sphere's curvature leads to variations in area and distance between points along different rows, the optimal use of dilated convolutions in spherical images requires adjusting dilation rates for each row [da Silveira and Jung, 2023].

Recently, methods based on attention mechanisms [Vaswani, 2017] and Vision Transformers (ViTs) [Dosovitskiy *et al*., 2020] have emerged as an alternative to CNNs to enhance ERP image processing. ViTs rely on different strategies to treat an image as a sequence of patches like tokens in natural language processing (NLP) [Cho et al., 2022; Shen et al., 2022; Bai *et al*., 2024; Carlsson et al., 2024]. Although transformer-based architectures have become a trend, CNN-based models or hybrid architectures that combine the local feature extraction capabilities of CNNs with the global context modeling of

ViTs tend to outperform purely ViT-based models, especially when training data is limited [Dai et al., 2021; Goldblum et al., 2024].

This paper extends the method and analysis from our previous work [Stringhini *et al*., 2024a], where Spherically-Weighted Horizontally Dilated Convolutions (SWHDC) modules were proposed and considered for 3D object classification using panoramas. The SWHDC module is a convolutional module designed to address the non-uniform sampling inherent in ERP images and enhance feature extraction within any existing CNN. This module is structured as a block consisting of multiple horizontally dilated convolutions (HDCs) with varying dilation rates and shared-weight kernels. The output of this block is computed through a linear combination of the multiple row-wise weighted feature maps from each convolution, where row-dependent weights are used to select the optimal support for each HDC, effectively compensating for distortions present in the input data.

More precisely, the contributions of this extended paper are:

- A detailed discussion regarding the different approaches proposed for distortion handling;
- A more in-depth analysis of 3D object classification and the efficiency of the SWHDC module for this specific task (we increase the amount of channels of the ERP images to 12; we further evaluated recently proposed backbones for the task of 3D object classification and provided a detailed quantitative analysis);
- The evaluation of our SWHDC module in the task of gravity alignment of panoramas;
- An analysis regarding the integration of SWHDC module into modern CNNs for the task of gravity alignment;

The rest of this paper is organized as follows. Section 2 presents a detailed discussion of several methods proposed in the literature to handle ERP distortions. Section 3 reviews the ERP representation and details the SWHDC module proposed in our previous work [Stringhini *et al*., 2024a]. An in-depth evaluation of the proposed module for the tasks of 3D object classification and gravity alignment in panoramas is presented in Section 4, and Section 5 concludes this work.

2 Related Work

Different approaches have been proposed to handle distortion in omnidirectional images, ranging from convolutional to transformer-based methods, and they will be revised next.

2.1 Convolutional methods for handling distortions in omnidirectional images

Su and Grauman [2017a] adapt planar CNNs for processing ERP images. Their method reproduces flat kernel outputs on spherical data while addressing distortion effects across the sphere. By adjusting the kernel shape based on spherical coordinates, their approach mitigates distortions, particularly high in the polar regions of ERP images. However, the regular convolution weights are only shared along each row and can not be trained from scratch [Ai *et al*., 2022]. Cohen *et al*. [2017] introduced spherical convolutions defined by the inner product between a spherical signal and a rotated spherical signal. This approach leverages the inherent rotational symmetry of spherical signals, analogous to how standard convolutional networks exploit the translation symmetry in planar images.

SphereNet [Coors *et al*., 2018] extends local CNN operations, such as convolution and pooling, from the planar image domain to the spherical surface. It represents the kernel as a small tangent patch to the sphere, mitigating distortions. Similarly, Tateno *et al*. [2018] proposed a distortion-aware method that enhances network convolutions using geometric priors. This approach dynamically reshapes the filter to adjust the receptive field based on distortion, directly compensating for image distortions during convolution. Esteves *et al*. [2018] presented a CNN capable of 3D rotation invariance by performing convolutions on the sphere and applying spectral domain pooling to maintain equivariance. Their subsequent spin-weighted spherical convolutions [Esteves *et al*., 2020] eliminate the need to lift data to $SO(3)$, with a fast implementation introduced in [Esteves *et al*., 2023]. In a different approach, Jiang *et al*. [2019a] designed convolutional kernels based on linear combinations of differential operators, allowing them to operate directly on icosahedral spherical meshes.

The Kernel Transformer Network (KTN) [Su and Grauman, 2019] adapts convolution kernels from perspective images to ERP images. Unlike spherical convolutions from [Su and Grauman, 2017a], which modify kernel shapes to address distortions, KTN learns a function that generates distortion-corrected kernels based on spherical coordinates. Fernandez-Labrador *et al*. [2020] introduced *EquiConv*, a specialized convolution kernel for ERP images. It operates on spherical surface patches defined by angular parameters, dynamically adapting to the geometry of the equirectangular projection.

Liu *et al*. [2022a] explored the HEALPix approach for sampling spherical data and employed pooling and convolution layers to operate in the transformed domain. In a complementary effort, Zioulis *et al*. [2018] introduced RectNet, a network inspired by UResNet, to directly handle omnidirectional images. RectNet incorporates dilated convolutions to capture global context and uses a combination of rectangular and square filters to compensate for horizontal distortion at varying resolutions. Xu *et al*. [2021] proposed a spherical U-Net that replaces traditional convolutions and pooling operators with spherical counterparts, sharing kernels across all patches to handle varying sampling rates.

The Distortion-Aware Monocular Omnidirectional network [Chen *et al*., 2021] combines deformable convolutions for distortion learning and a strip pooling module to capture distortion along horizontal and vertical dimensions. Alternatively, Pintore *et al*. [2021] presented a slice-based representation to directly utilize ERP's vertical structure, bypassing distortion-aware convolutions. In ACDNet [Zhuang et al., 2022], dilated convolutions with variable horizontal and vertical dilation rates are fused through a channel-wise module to produce large receptive fields. Lee and Park [2022] proposed transformable-dilated convolutions, which dynamically adjust kernel sizes based on object distances in spherical light detection and ranging (LiDAR) data, using larger kernels for closer objects and smaller ones for distant ones.

2.2 Transformer-based methods for handling distortions in omnidirecional images

PanoFormer [Shen et al., 2022] introduced a novel approach by leveraging tangent patches derived from the spherical domain and learnable tokens to reduce distortions and enhance depth estimation accuracy. PanoFormer includes a spherical token-locating model for precisely sampling ERP regions and a Panoramic Structure-guided Transformer (PST) block to capture diverse geometric structures. Similarly, OmniFusion [Li *et al*., 2022] transforms the ERP into distortion-free tangent patches for patch-wise depth predictions. The results are merged using a geometry-aware feature fusion mechanism that integrates 3D geometric information with 2D features. The feature fusion mechanism is followed by a self-attention transformer for globally consistent aggregation of patch-wise predictions. Rey-Area *et al*. [2022] project the input ERP image onto a series of tangent planes to create perspective views. These tangent images are processed to generate detailed depth maps, which are then aligned using spatially varying deformation fields and blended with a gradient-based approach to ensure global consistency. PanoSwin Transformer [Ling *et al*., 2023], inspired by the Swin Transformer [Liu *et al*., 2021], addresses ERP distortions using a pano-style shift windowing technique, which involves splitting the ERP, rotating the right half counterclockwise, and applying pitch attention to enable effective cross-attention between corresponding windows. PCFormer [Xu *et al*., 2023] integrates the strengths of CNNs and transformers in a hybrid architecture. It consists of a transformer-based branch employing a Swin Transformer to capture global features and a convolutional branch to capture local features at multiple scales.

EGFormer [Yun *et al*., 2023] utilizes transformer blocks designed explicitly for ERPs, with horizontal and vertical self-attention mechanisms to capture information along these dimensions effectively. Additionally, it incorporates equirectangular relative position embeddings into the attention mechanism, ensuring attention scores accurately reflect spatial relationships within the ERP. GLPanoDepth [Bai *et al*., 2024] also employs a two-branch architecture. One branch transforms the ERP into a cube-map projection (CMP) and uses a vision transformer backbone for learning long-range dependencies. In contrast, the other branch uses planar CNNs to extract ERP features directly. SGFormer [Zhang *et al*., 2024] introduced the Spherical Geometry Transformer to capture local details and address ERP distortions. Its decoder features a bipolar reprojecting scheme to mitigate severe distortions by reprojecting these regions to the equator, followed by circular rotation for continuous panoramic representation and curved relative position embeddings to enhance spatial structure understanding. Finally, HEAL-SWIN [Carlsson et al., 2024] adapts the Swin Transformer [Liu *et al*., 2021] to operate on a uniform HEALPix grid, ensuring spherical compatibility.

While numerous distortion-aware strategies have been proposed to address the challenges of non-uniform sampling in ERP images, they often come with significantly higher computational costs compared to planar counterparts [da Silveira and Jung, 2023]. Also, some of these techniques need a vast amount of data to be trained on [da Silveira and Jung, 2023; Goldblum et al., 2024], or fail to deliver substantial improvements over conventional planar approaches [Fernandez-Labrador *et al*., 2020]. ViTs have gained widespread attention recently and can be adapted to the spherical domain by carefully selecting appropriate patches [Bai *et al*., 2024; Cho et al., 2022; Shen et al., 2022]. Despite that, ViT-based approaches typically rely on larger training datasets to perform effectively [Dai et al., 2021; Goldblum et al., 2024]. Notably, Goldblum et al. [2024] analyzed different architectures and concluded that modern CNNs pre-trained through supervised learning often outperform ViTs on various vision tasks. However, transformers tend to benefit more from increased dataset scale.

In this work, we present an extended version of our previous work [Stringhini *et al*., 2024a], where we proposed a novel convolutional module, named SWHDC, that can be integrated into any existing planar CNN to handle distortions and improve feature extraction in ERP images without increasing the number of trainable parameters. Next, we revise our proposed convolutional module.

3 Proposed Convolutional Module

In this section, we review the ERP representation and revise the SWHDC module [Stringhini *et al*., 2024a] specifically designed to address distortions in ERP images. Unlike the approach proposed by [Lee and Park, 2022], where kernel sizes are adjusted based on object distances in LiDAR data, our approach applies a spherical weighting scheme to a set of dilated convolutions, approaching the response of an optimal support for each latitude in the ERP image or feature map.

3.1 Equirectangular Projection of Spherical Surfaces

Spherical imaging projects a 3D point $\mathbf{P} \in \mathbb{R}^3$ onto a unit sphere centered at $\mathbf{C} \in \mathbb{R}^3$ by spherical projection, obtaining an intersection point $\mathbf{p} \in S^2$ [da Silveira *et al*., 2022]. Each point $\mathbf{p}$ has a unit distance from the camera center and can be rewritten as

$$\mathbf{p} = \begin{bmatrix} \sin\phi\cos\theta \\ \sin\phi\sin\theta \\ \cos\phi \end{bmatrix}, \tag{1}$$

where $\theta \in [0, 2\pi)$ represents the longitudinal angle and $\phi \in [0, \pi)$ denotes the latitudinal angle. The entire sphere can be mapped onto a rectangular grid with dimensions $[0, 2\pi) \times [0, \pi)$, creating a regular mapping called equirectangular projection [Su and Grauman, 2017b]. The spherical point $\mathbf{p} \in S^2$ can be projected to a position (v, u) in a $h \times w$ image, where v and u are the pixel coordinates being expressed as

$$v = \left\lfloor \frac{\phi h}{\pi} \right\rfloor, \quad u = \left\lfloor \frac{\theta w}{2\pi} \right\rfloor. \tag{2}$$

In this formulation, v and u correspond to the vertical and horizontal pixel positions in the image, respectively, making it simple to project a spherical surface onto a 2D image plane. This mapping, however, inherently introduces distortions due to the unequal spacing in latitude lines on the sphere, which is stretched when represented on a flat grid.

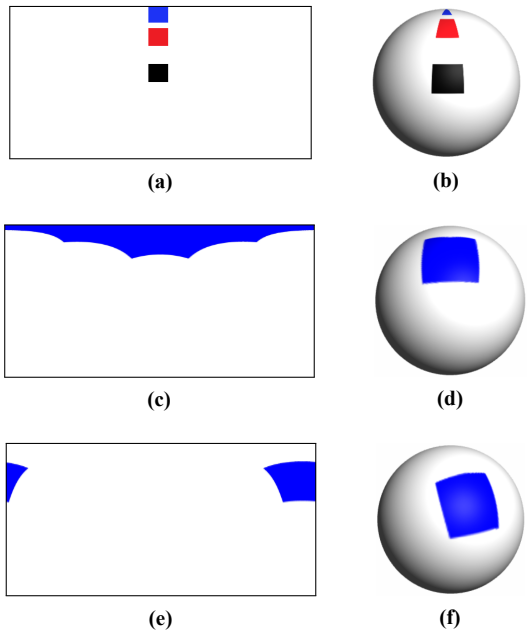


Figure 1. Applying regular fixed-support kernels at different latitudes to (a) ERP images covers uneven results in uneven coverage of the (b) spherical surface. The squares represent the kernels (enlarged for visualization purposes). The ideal kernels to ERPs and their coverage on the sphere are illustrated in (c), (d), (e), and (f).

In ERP images, regions near the poles are more densely sampled than those near the equator [da Silveira and Jung, 2023]. The horizontal distance between two points on the sphere with longitudes θ_1 and θ_2, given a latitude ϕ, is expressed as $\sin\phi|\theta_1 - \theta_2|$, as detailed in Eq. (1). Conversely, for a fixed longitude θ, the vertical distance between two points with latitudes ϕ_1 and ϕ_2 is simply $|\phi_1 - \phi_2|$, which remains constant. As a result, points closer to the poles require larger horizontal support (scaled by $1/\sin\phi$) than those at the equator. Note that the vertical kernel support should remain fixed.

3.2 Spherically-Weighted Horizontally Dilated Convolutions

Our SWHDC module can be integrated into any planar CNN to address ERP distortions and enhance omnidirectional feature extraction. Building on the approach from [Schuster et al., 2019], where stacked parallel dilated convolutions are employed for regular images, our SWHDC module adapts this concept for ERP images. In [Schuster et al., 2019], the outputs of the dilated convolutions (in both dimensions) are stacked to generate a multi-scale response. Similarly, our SWHDC module employs multiple dilated convolutions but restricts dilation to the horizontal dimension. Rather than concatenating the filter responses, we perform a row-dependent linear combination of them. This design choice allows each latitude (row) of the ERP (or feature map) to be influenced differently by each dilated convolution. This adaption enables convolutional filters to process omnidirectional images, overcoming the limitations of planar filters with fixed support, which cover varying areas of the sphere surface depending on the kernel's latitude (see Fig. 1).

Our SWHDC module consists of N predefined horizontally dilated convolutions (HDC) that share weights, each

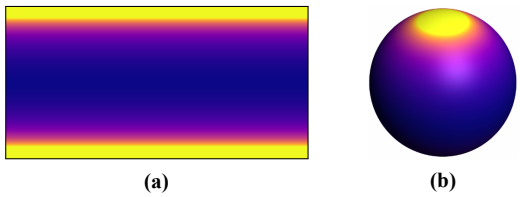


Figure 2. Ideal dilation rates that should be adjusted depending on the latitude: purplish colors represent small dilation rates (not smaller than 1) while yellowish colors represent large dilation rates (in the limit case, infinity). Sphere (b) slightly rotated for visualization purposes.

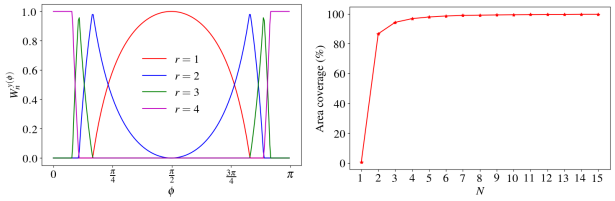


Figure 3. (a) Distribution of the weights W_r^{ϕ} according to ϕ when $N = 4$. (b) Percentage of area coverage on the spherical surface for N HDCs.

employing a different dilation rate r to generate a feature map F_r, for $r = 1, \ldots, N$. This particular design enables a multi-scale feature extraction and is well adapted to account for ERP image distortions. We use circular padding to tackle the cyclical property of ERPs, where the left and right boundaries represent points next to each other on the sphere. Also, using shared weight kernels, the number of trainable parameters remains fixed regardless of the number of HDCs, ensuring computational efficiency.

As mentioned in Section 3.1, due to the non-uniform sampling in ERPs, points closer to the poles require larger horizontal kernel support scaled by a factor of $\sin(\phi)^{-1}$ to address the irregular sampling in these regions. Because the support is directly proportional to the dilation rate r of the kernel, we determine the optimal row-wise r using the factor $\sin(\phi)^{-1}$ as illustrated in Fig. 2 (purplish colors represent smaller values while yellowish colors represent larger ones). As we move towards the poles, the dilation rate becomes larger in the ERP to cover the same surface area on the sphere. Since this factor produces non-integer values, we interpolate between the two nearest dilation rates r. For example, if the ideal row-wise dilation rate is $r = 2.7$, the nearest dilation rates are $r = 2$ and $r = 3$. Explicitly, the convolution with $r = 3$ is closer to the ideal value than the other; therefore, it will receive the higher weight. Conversely, the convolution with $r = 2$ will receive a lower weight. This weighting strategy ensures that the HDC with the dilation rate closest to the ideal factor has more importance for that given latitude (row). Also, it is essential to note that we do not modify the kernel size to change according to the latitude (row) to better fit the spatial characteristics of ERPs. Instead, we focus on applying different weights to each HDC based on how close each dilation rate is to the ideal factor for a given latitude (row) to adjust the influence of each dilation rate. The mathematical procedure for this spherical weighting approach is explained next.

The row index v in the input (ERP or feature map) to the SWHDC module maps to a latitude ϕ based on Eq. (2),

determining the ideal scaling factor S_ϕ as being

$$S_\phi = \min\{N, 1/\sin\phi(v)\}. \tag{3}$$

Note that the maximum scaling factor S_ϕ is limited to the largest dilation N. The weight W_r^ϕ for the HDC with dilation rate r and row index corresponding to a latitude ϕ is determined by interpolating between the two nearest integer scales, i.e.,

$$W_r^\phi = \begin{cases} 1, & \text{if } S_\phi \in \mathbb{N} \text{ and } r = S_\phi \\ \lceil S_\phi \rceil - S_\phi, & \text{if } S_\phi \notin \mathbb{N} \text{ and } r = \lfloor S_\phi \rfloor \\ S_\phi - \lfloor S_\phi \rfloor, & \text{if } S_\phi \notin \mathbb{N} \text{ and } r = \lceil S_\phi \rceil \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ denote rounding to the closest larger and smaller integers, respectively. Fig. 3a shows the weights W_r^ϕ for the case when the amount of HDCs is $N = 4$.

A linear combination of the feature maps F_r, obtained from the HDCs, yields the combined feature map F_* expressed as

$$F_* = \sum_{r=1}^{N} H_B(W_r) \odot F_r, \tag{5}$$

where W_r is the weight for all latitudes (rows), $H_B(\cdot)$ denotes horizontal broadcasting, and $\odot$ is the element-wise multiplication operator.

The proposed SWHDC module preserves the number of trainable parameters as standard convolutional modules by leveraging hardcoded row-wise weights to approximate the optimal support for each HDC. Although multiple dilated convolutions are used, the combined feature map F_* retains the same number of output channels as in planar convolutional counterparts. This design ensures computational efficiency, enabling enhanced feature extraction while addressing distortions without introducing parameter overhead. Our SWHDC module is illustrated in Fig. 4 for $N = 4$ HDCs.

Recall that our SWHDC module uses a set of N HDCs with varying dilation rates r, where $r = 1, \ldots, N$, producing distinct feature maps F_r. We chose $N = 4$ after analyzing the relative area coverage on a spherical surface. As shown in Fig. 3b, the area covered by a dilated convolution on a spherical surface shows a minimal increase for $N > 4$. It is also evident that dilated convolutions with an ideal kernel support raging from $r = 1$ to $r = 2$ cover significantly more area than higher rates. For $N = 4$, the receptive field is sufficiently large to extract meaningful features and effectively address distortions comparably to $N \geq 5$, as the additional coverage area becomes minimal ($\approx 1.1\%$). Therefore, setting $N = 4$ is adequate for addressing distortions near the poles of ERP images. In the case of $N = 1$, where the single dilation rate is $r = 1$, kernel expansion is unnecessary, and a standard convolution is applied. Moreover, the number of trainable parameters remains constant, regardless of the value of N. Further results are provided in Section 4.

4 Experimental Results and Discussions

As reported in [Stringhini *et al.*, 2024a], the SWHDC module was originally designed to handle the non-uniform sampling issues in ERP images, enhancing feature extraction in omnidirectional images in any existing CNN. In this work, we expand its validation by first extending the results of 3D object classification task to a comprehensive range of methods proposed in the literature. Additionally, we also extend the validation of the SWHDC module by applying it to the gravity alignment task, demonstrating its adaptability to new challenges beyond its original scope.

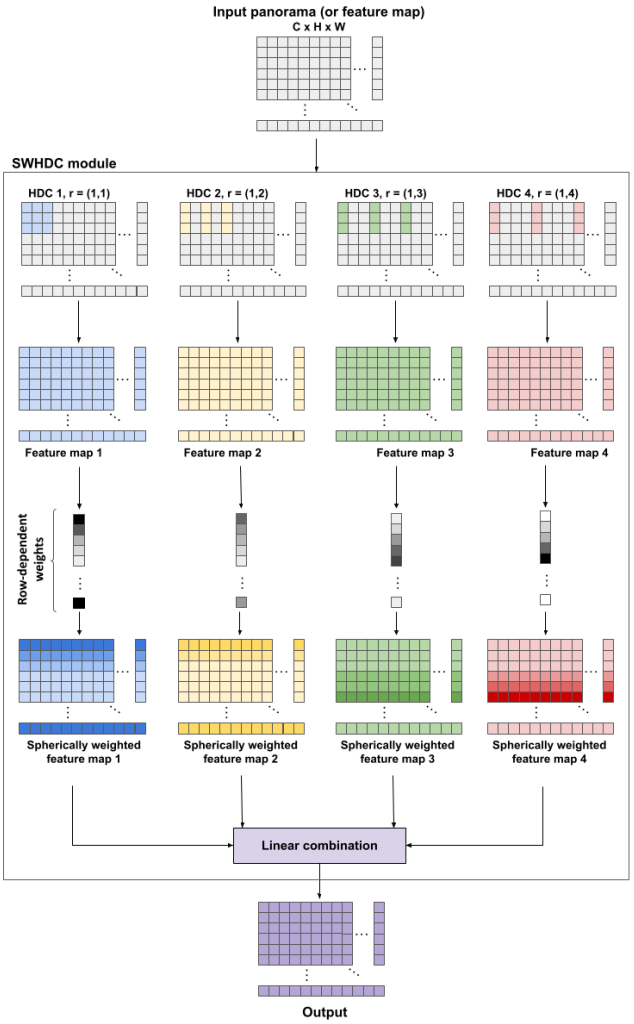


Figure 4. Composition of our SWHDC module when $N = 4$. The input is processed by N HDCs with different dilation rates r. Each feature map passes through a row-wise weighting. Then, all N spherically-weighted feature maps are combined to generate the final combined feature map. "H", "W", "C", and "HDC" stand for height, width, channels, and horizontally dilated convolution, respectively.

4.1 3D Object Classification

Object classification is a fundamental task in computer vision where the goal is to assign a predefined category label to an object within an image [Rawat and Wang, 2017]. This task can be straightforwardly extrapolated to 3D models [Muzahid *et al.*, 2024], and it is explored in this work.

Typically, methods for 3D object classification fall into three primary categories: point-based, volumetric-based (or voxel-based), and view-based strategies [Guo *et al.*, 2020; Bello *et al.*, 2020]. Point-based methods [Qi *et al.*, 2017a,b; Ma *et al.*, 2022; Wu *et al.*, 2023; Liu and Tian, 2024] extract features from 3D point clouds, and they are impacted by

the sampling strategy used to obtain sparse points from the meshes [Guo *et al*., 2020]. Volumetric-based methods [Wu *et al*., 2015; Maturana and Scherer, 2015; Riegler *et al*., 2017; Zhi *et al*., 2018] convert 3D data into regular voxel grids and typically find computational limitations when dealing with high-resolution data [Guo *et al*., 2020; Bello *et al*., 2020]. View-based methods explore 2D representations of 3D objects, enabling the use of established 2D CNNs [Qi *et al*., 2021]. Many works explore the view-based strategy using either multiple [Kanezaki *et al*., 2018; Sfikas *et al*., 2018; Han *et al*., 2019; Hamdi *et al*., 2021; Li *et al*., 2023] or single images [Shi *et al*., 2015; Yavartanoo *et al*., 2018; Zhou *et al*., 2019; Ding *et al*., 2020; Hoang *et al*., 2020] to represent 3D objects.

Point-based methods handle raw 3D point clouds and learn the features of each point. As a pioneering work, PointNet [Qi *et al*., 2017a] handles the unordered nature of point clouds and achieves permutation invariance through max pooling operations. PointNet++ [Qi *et al*., 2017b] extends PointNet by dividing points into regions that are further processed indepedently to extract local geometric features. Qian *et al*. [2022] used an inverted residual bottleneck architecture and separable multilayer perceptrons to improve PointNet++. Recent advancements in point cloud classification integrate attention mechanisms from NLP, as seen in Point Transformer [Zhao *et al*., 2021], Point-BERT [Yu *et al*., 2022], and G-Point++ [Liu and Tian, 2024]. Although point-based methods are constantly being improved, they depend on an efficient sampling strategy [Mirbauer *et al*., 2021].

Volumetric-based methods convert point clouds into 3D grids via voxelization for processing with 3D networks. 3DShapeNets [Wu *et al*., 2015] represent 3D shapes as probability distributions in voxel grids. VoxNet [Maturana and Scherer, 2015] applies 3D convolutions to occupancy grids to generate feature vectors. OctNet [Riegler *et al*., 2017] uses balanced octrees to partition data, reducing computation and memory but struggling with high-resolution data [Guo *et al*., 2020; Bello *et al*., 2020]. Also, voxel-based methods struggle to capture extensive contextual information due to the sparsity and varying density of 3D data, resulting in inefficient computation [Bello *et al*., 2020; Zhao *et al*., 2022].

In contrast, view-based methods projects 3D data into multiple (multi-view) or single 2D views (single-view), enabling the use of well-established 2D CNNs. Multi-view methods use a set of viewpoints and rotation angles to render 3D objects into multiple 2D images from different perspectives [Qi *et al*., 2021], as seen in MVCNN [Su *et al*., 2015], RotationNet [Kanezaki *et al*., 2018], MVCNN [Liu *et al*., 2020], and MVCLN [Liang *et al*., 2020], for instance. On the other hand, only one image representing the 3D object is processed in single-view methods. This single view passes through a feature extractor to be directly sent to a classifier for the final prediction, eliminating the need to combine different features from multiple views. Such methods project the 3D data into different 2D representations, using different types of projection such as cylindrical [Shi *et al*., 2015], stereographic [Yavartanoo *et al*., 2018], polar [Zhou *et al*., 2019], perspective [Ding *et al*., 2020; Hoang *et al*., 2020], or spherical [Liu *et al*., 2022a; Stringhini *et al*., 2024b].

In this work, our primary goal is to handle distortions in ERP images by integrating our SWHDC module into planar CNNs. For the task of 3D object classification, we opted to convert 3D objects represented as meshes into omnidirectional images using the ERP to, later on, validate the effectiveness of our convolutional module.

4.1.1 Converting 3D Meshes to ERP Images

To generate 2D omnidirectional images, we adopted an approach inspired by methods that explore panoramas to represent 3D shapes [Esteves *et al*., 2018, 2020; Liu *et al*., 2022a]. In our case, to obtain an omnidirectional image of the 3D shape (mesh), we cast rays omnidirectionally from the centroid C of the object outwards, retrieving local geometrical information at the intersection points between rays and object. The centroid of the mesh is computed as the average of the triangle centroids weighted by the area of each triangle. By doing so, we ensure total object coverage from all directions. The orientation of each ray is defined by θ and ϕ relative to the object's centroid and mapped to an image pixel position (v, u) according to Eq. (2). For our case, we used $h = 256$ and $w = 512$, a common choice for panorama-based deep learning approaches [da Silveira and Jung, 2023].

Once cast, rays may intersect none, one, or multiple triangles of the 3D object, producing multiple local geometric features at each intersection point. The number of intersections for a ray depends on its direction and the complexity of the object. After a careful analysis, we chose to retrieve only the first and last intersection points, thus capturing the "inner" and "outer" geometries. The point is duplicated for rays with a single intersection as the first and last hit. For rays with no intersections, the corresponding pixel value in the resulting ERP image is set to zero, indicating the absence of relevant ray-related information. Using zero to encode irrelevant information has also been adopted in other applications, such as omnidirectional scene depth estimation [Zioulis *et al*., 2018].

In this study, we build upon [Stringhini *et al*., 2024b] and incorporate additional geometric local features compared to [Stringhini *et al*., 2024a]. Beyond the depth map, which represents the distance from the centroid C to the object, we incorporate features such as surface normal map [Sfikas *et al*., 2018], orientation map [Sfikas *et al*., 2018], and gradient magnitude of the depth map [Sfikas *et al*., 2018]. By retrieving this information, a multi-channel representation is provided, enabling a more comprehensive analysis of each object's geometry and spatial properties.

A 2D surface normal map is an image that encodes the orientation of surface normals across a 3D object. Each pixel in the map represents the direction of the surface normal vector at the corresponding point on the object's surface. To compute it, we obtain the normalized cross-product of the edges of the triangular face intersected by the ray for each intersection point. An orientation map is an image that illustrates the alignment of surface normals relative to a set of predefined directions, which is derived from the cosine of the angle between the ray with direction (θ, ϕ) and the normal vector of the intersected surface. The depth map's gradient magnitude is obtained using Gaussian derivatives with a standard deviation, experimentally set to $\sigma = 2$, highlighting sharp transitions corresponding to significant local geometric

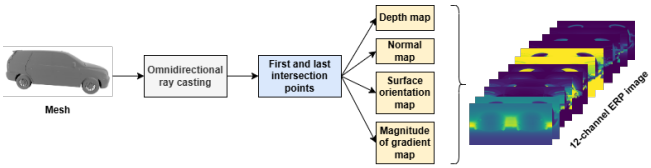


Figure 5. Employed pipeline for converting 3D meshes into multi-channel ERP images.

features. If a ray does not hit the object, we set the feature values to zero since there is no information to collect.

By retrieving these features as in [Stringhini *et al.*, 2024b], we obtain $N_{channels} = 12$ (6 per intersection point), which can be observed in Fig. 5. Thus, we can represent a 3D object as a 12-channel ERP image with 256×512 resolution.

4.1.2 Datasets and Training Details

To evaluate and validate the efficiency of the SWHDC module, we conduct several experiments for the task of 3D object classification. The most commonly used datasets are the ModelNet [Wu *et al.*, 2015] and ShapeNetCore (v2) [Chang *et al.*, 2015] datasets. ModelNet includes ModelNet10 (10 categories, 3,991 training, and 908 testing 3D models) and ModelNet40 (40 categories, 9,843 training, and 2,468 testing 3D models), with official splits. ShapeNetCore (v2) features 55 categories with greater complexity due to its unbalanced sample distribution. While it lacks official splits, we use standardized splits proposed in [Mirbauer *et al.*, 2021]. For all datasets, the training set is split into 80% for training and 20% for validation.

We trained different backbones for up to 200 epochs using early stopping with patience of 25 epochs. The Adam optimizer was used with an initial learning rate of 10^{-4}, decaying by 0.9 every 25 epochs to a minimum of 10^{-7}. Data augmentation included 3D rotations (0-15° for the x and y axes, 0-45° for the z axis), Gaussian blur with a random σ from 0.1 to 2, and Gaussian noise with a mean from 0 to 0.001 and σ from 0 to 0.03. All these primitives were applied to the training set with a probability of occurrence of 15%. Due to the nature of our ERP images, all backbones were trained from scratch, without pretrained weights.

4.1.3 3D Object Classification with SWHDC-based CNNs

Several experiments were conducted to better understand and analyze the SWHDC module's effectiveness in the task of 3D object classification. We evaluated several planar convolutional architectures, including VGG-16 [Simonyan and Zisserman, 2014], EfficientNets [Tan and Le, 2019], ResNets [He *et al.*, 2016], the encoder of Panoformer [Shen et al., 2022], ConvNeXts [Liu *et al.*, 2022b], and Conv2Former [Hou *et al.*, 2024]. By analyzing the performance of these planar backbones in processing ERP images, we can better validate the benefits of integrating the SWHDC module into these backbones. As input to these backbones, we used the 12-channel ERP images obtained from converting the 3D shapes into omnidirectional images with full resolution (256×512). As observed in Table 1, the residual-based and the recently proposed Conv2Former architectures yielded the best performance among the analyzed backbones.

Table 1. Performance of planar backbones in processing spherical images from the ModelNet10 dataset. The number of parameters is given in millions.

Backbone	Parameters	Accuracy
VGG-16	138.3	91.2%
EfficientNet-B0	5.3	89.8%
EfficientNet-B3	12.2	90.7%
ResNet-18	11.7	92.0%
ResNet-34	21.8	91.3%
ResNet-50	25.5	91.4%
PanoFormer encoder	13	85.74%
ConvNeXt-tiny	28.5	88.6%
ConvNeXt-small	50.2	89.7%
ConvNeXt-base	88.6	91%
ConvNeXt-large	197.8	90.4%
Conv2Former-N	15	91.9%
Conv2Former-T	27	91.7%

Table 2. Integration of the SWHDC module, and spherical convolutions ("SPH") into ResNets and Conv2Former backbones on the ModelNet10 dataset. The number of parameters is given in millions.

Backbone	Parameters	Accuracy
ResNet-18 + SPH	16.4	92.2%
ResNet-18 + SWHDC	11.7	96.9%
ResNet-34 + SPH	26.9	93.4%
ResNet-34 + SWHDC	21.8	**97.4%**
ResNet-50 + SPH	30.1	92.8%
ResNet-50 + SWHDC	25.5	96.2%
Conv2Former-N + SPH	21.6	93.1%
Conv2Former-N + SWHDC	15	95.1%
Conv2Former-T + SPH	36.8	93.3%
Conv2Former-T + SWHDC	28	95.4%

The subsequent experiment replaced the planar convolutions from the residual and Conv2Former architectures with our SWHDC module. Specifically for the Conv2Former variants (-N and -T), we replaced the convolutions in their patch embedding module and kept the same structure regarding stages and blocks as described in [Hou *et al.*, 2024]. To further evaluate the improvements brought by our convolutional module, we also replaced the planar convolutions with spherical convolutions from SphereNet [Coors *et al.*, 2018] for comparison. As noted in Table 2, integrating our convolutional module improved the performance of all backbones, with ResNet-34 combined with SWHDCs offering a good balance between accuracy and parameter count. Although the spherical convolution from SphereNet improved the planar backbones, they significantly increased the number of parameters and still resulted in lower accuracy than our module.

Given that the combination of ResNet-34 with our SWHDC module provided the best performance, we named the resulting network SWHDCNet. This network retains the overall structure of ResNet-34, preserving parameters such as kernel size, stride, and padding. The key difference lies in replacing traditional planar convolutions with our SWHDC module, allowing the network to adapt to omnidirectional data.

Table 3 presents the results for both the ModelNet10 and ModelNet40 datasets. SWHDCNet outperforms all voxel-based and single-view methods on both datasets. Compared to single-view methods, SWHDCNet achieves a 4% and

1.2% accuracy improvement on ModelNet10 and ModelNet40 datasets, respectively. These methods typically rely on cylindrical views that fail to capture the top and bottom parts of 3D models, perspective views from a single "optimal" viewpoint, or depth maps for each object. In contrast, our classifier uses a single spherical representation that captures multi-channel features from all angles of a given object. Furthermore, we address distortions by using a classifier based on SWHDCs. Regarding point-based methods, it is essential to note that their accuracy is influenced by the mesh sampling strategy and the number of points used [Mirbauer *et al*., 2021; Yu *et al*., 2022; Liu and Tian, 2024].

A more in-depth analysis of our method is provided in Figure 6, which displays the confusion matrices for ModelNet10 and ModelNet40 using SWHDCNet. Although our method achieved high accuracy rates (100% for some categories), a few pairs of categories displayed mutual confusion. Notably, pairs such as `table` vs. `desk` and `dresser` vs. `nightstand` in ModelNet10 yield mutual confusion, which is expected due to their visual similarities. In fact, these categories are hard to distinguish even for humans (see Fig. 7), as reported in previous studies [Garcia-Garcia *et al*., 2016; Gomez-Donoso *et al*., 2022]. When considering ModelNet40, we can observe that there were also confusions between `flower pot` vs. `plant` vs. `vase`.

The results for ShapeNetCore (v2) are presented in Table 4, with accuracy values from peer methods taken from [Mirbauer *et al*., 2021]. The SWHDCNet outperforms all point- and voxel-based methods, achieving an accuracy rate of 91.41%. Furthermore, compared to multi-view methdods [Kanezaki *et al*., 2018; Su *et al*., 2018; Han *et al*., 2018] that use multiple views per object as stated in [Mirbauer *et al*., 2021], our single-view classifier achieved competitive accuracy. The key strengths of SWHDCNet lie in its ability to address the non-uniform sampling (distortions), capture comprehensive features from a single ERP image, and balance accuracy with computational efficiency.

As discussed in Section 3.2, setting the number of HDCs in the SWHDC module to $N = 4$ (i.e., with dilation rates $r = 1, 2, 3, 4$) covers approximately 96.85% of the spherical surface, ensuring effective feature extraction and distortion handling. Increasing N to 5 provides only a slight coverage gain ($\approx 1.1\%$), indicating that while higher dilation rates expand the receptive field, they offer a little benefit, especially for small feature maps. Table 5 indicates $N = 4$ as optimal, as excessive dilation reduces sampled pixels, leading to spatial information loss. Moreover, as shown in our experiments and by Zhuang et al. [2022], higher dilation rates can degrade performance and increase complexity. This analysis was conducted using a ResNet-18 backbone with SWHDCs and a single-channel ERP image containing the external depth map of 3D objects as our prior work [Stringhini *et al*., 2024a] for simplicity.

4.2 Gravity Alignment in Panoramas

Gravity alignment, also known as upright vector correction, is an important task that aims to adjust the orientation of captured panoramas to align with the direction of gravity [Jung *et al*., 2019; Jeon *et al*., 2019; Bergmann *et al*., 2021]. This

Table 3. Overall classification accuracy results for the ModelNet10 (MN10) and ModelNet40 (MN40) datasets. In the input column, "Pts" stands for points, "Vx" for voxels, "MV" for multi-view, and "SV" for single-view. "Par." stands for the number of parameters (millions).

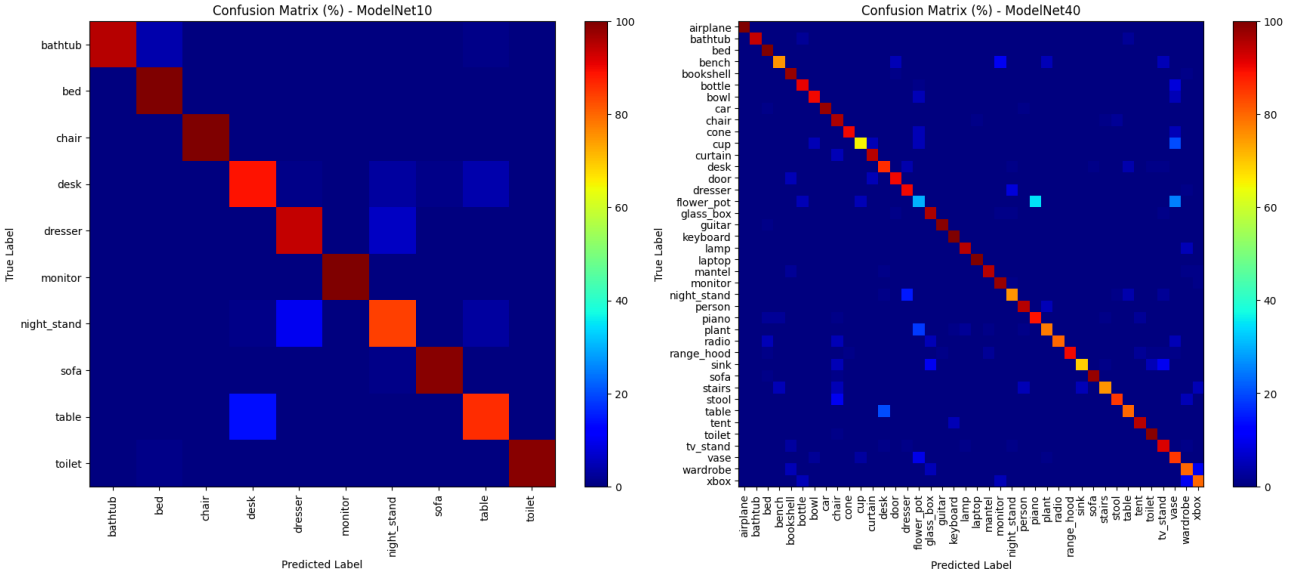
Method	Input	MN10	MN40	Par.
PointNet [Qi *et al*., 2017a]	Pts	-	89.2%	3.5
PointNet++ [Qi *et al*., 2017b]	Pts	-	91.9%	1.7
3D Capsule [Cheraghian and Petersson, 2019]	Pts	94.7%	92.7%	-
PointMLP [Ma *et al*., 2022]	Pts	-	94.1%	12.6
Point-BERT [Yu *et al*., 2022]	Pts	-	93.8%	-
3DCTN [Lu *et al*., 2022]	Pts	-	93.2%	-
SWPT [Guo *et al*., 2022]	Pts	-	93.5%	-
Point-PN [Zhang *et al*., 2023]	Pts	-	93.8%	0.8
PointConT [Liu *et al*., 2023]	Pts	-	93.5%	-
APES [Wu *et al*., 2023]	Pts	-	93.8%	0.4
G-PointNet++ [Liu and Tian, 2024]	Pts	95.5%	93.3%	-
3DShapeNet [Wu *et al*., 2015]	Vx	83.5%	77.3%	-
VoxNet [Maturana and Scherer, 2015]	Vx	92.0%	83%	0.9
OctNet [Riegler *et al*., 2017]	Vx	90.4%	-	-
ORION [Sedaghat *et al*., 2016]	Vx	93.8%	-	0.9
LightNet [Zhi *et al*., 2018]	Vx	93.9%	88.9%	-
MVCNN [Su *et al*., 2015]	MV	-	90.1%	42
Cao *et al*. [2017]	MV	-	94.2%	-
RotationNet [Kanezaki *et al*., 2018]	MV	98.5%	97.4%	42
Panorama-ENN [Sfikas *et al*., 2018]	MV	96.9%	95.6%	8.6
MHBN [Yu *et al*., 2018]	MV	95.0%	94.7%	-
SeqViews2SeqLabels [Han *et al*., 2018]	MV	94.8%	93.3%	-
3D2SeqViews [Han *et al*., 2019]	MV	94.7%	93.4%	-
MLVCNN [Jiang *et al*., 2019b]	MV	-	94.2%	-
G-CNN [Esteves *et al*., 2019]	MV	97.5	94.7%	-
View-GCN [Wei *et al*., 2020]	MV	-	97.6%	73.4
Ma *et al*. [2018]	MV	95.3%	91.1%	-
MVCLN [Liang *et al*., 2020]	MV	95.7%	93.5%	-
BPCN [Luo *et al*., 2020]	MV	-	95%	-
VWN [Huang *et al*., 2020]	MV	95.16%	93.8%	-
MVACPN [Liu *et al*., 2020]	MV	-	93.6%	-
DRCNN [Sun *et al*., 2020]	MV	99.3%	96.8%	-
MVTN [Hamdi *et al*., 2021]	MV	-	93.8%	11.2
PointView-GCN [Mohammadi *et al*., 2021]	MV	-	95.4%	-
3DSliceLeNet [Gomez-Donoso *et al*., 2022]	MV	94.4 %	79.9%	-
MVCVT [Li *et al*., 2023]	MV	98.6%	95.4%	19
DeepPano [Shi *et al*., 2015]	SV	88.7%	82.5%	3.3
SPNet [Yavartanoo *et al*., 2018]	SV	93.4%	88.6%	0.9
PVR [Zhou *et al*., 2019]	SV	92.7%	91.7%	-
Ding *et al*. [2020]	SV	91.2%	89%	-
Hoang *et al*. [2020]	SV	91%	85.8%	-
STM [Liu *et al*., 2022a]	SV	-	93%	-
SWHDCNet (Ours)	**SV**	**97.4%**	**94.2%**	**21.8**

Table 4. Overall classification accuracy results for the ShapeNetCore (v2) dataset. Results from peering methods borrowed from [Mirbauer *et al*., 2021].

Method	Input	Accuracy
PointNet [Qi *et al*., 2017a]	Pts	90.20%
PointNet++ [Qi *et al*., 2017b]	Pts	90.73%
Kd-Net [Klokov and Lempitsky, 2017]	Pts	87.36%
SO-Net [Li *et al*., 2018]	Pts	90.72%
O-CNN [Wang *et al*., 2017]	Vx	90.66%
Adaptive O-CNN [Wang *et al*., 2018]	Vx	87.35%
Voxception-ResNet [Brock *et al*., 2016]	Vx	88.30%
RotationNet [Kanezaki *et al*., 2018]	MV	91.48%
MVCNN2 [Su *et al*., 2018]	MV	92.98%
SeqViews2SeqLabels [Han *et al*., 2018]	MV	92.61%
SWHDCNet (Ours)	**SV**	**91.41%**

Table 5. Results varying the amount of N horizontally dilated convolutions in our SWHDC module on the ModelNet10 dataset.

Backbone	Accuracy
ResNet-18+SWHDC ($N = 2$)	90.41%
ResNet-18+SWHDC ($N = 3$)	91.40%
ResNet-18+SWHDC ($N = 4$)	92.38%
ResNet-18+SWHDC ($N = 5$)	91.96%



(a) **(b)**

Figure 6. Confusion matrices for (a) ModelNet10 and (b) ModelNet40 datasets.

(a) Desk **(b)** Table **(c)** Dresser **(d)** Nightstand

Figure 7. Similarity between two objects of different classes in ModelNet10.

process involves estimating a rotation matrix to align the image content so that the panorama's equator line is parallel to the ground plane. To this aim, one first projects the image onto a unit sphere, applies the rotation matrix, and then projects the adjusted intensities back onto a plane to obtain the corrected orientation [da Silveira and Jung, 2023]. This alignment ensures that the image appears as if the camera was perfectly vertical, compensating for any tilt or misalignment during capture [Jung *et al*., 2019; Jeon *et al*., 2019].

The main challenge with gravity alignment arises from the inherent distortions and projection models in spherical imagery, which can cause a wavy horizon and tilted structures if the camera is not upright, thereby significantly degrading visual quality and causing discomfort for viewers, especially when the image is displayed on a VR headset [Jung *et al*., 2019]. Besides being useful for AR and VR, gravity alignment can be used as an intermediate process for different computer vision tasks, such as layout and depth estimation, since they often apply to upright images only [da Silveira and Jung, 2023].

As a representative approach for upright alignment, Jung *et al*. [2019] propose the Deep360Up model, which employs a DenseNet121 backbone with the classification layer replaced by a regression layer to directly regress the two angles (elevation and azimuth) necessary to align the input image to the horizon. Jeon *et al*. [2019] performed automatic upright adjustment by employing a residual network to estimate rotation by analyzing 2D rotations projected from multiple narrow field-of-view images sampled from the panorama. Davidson *et al*. [2020] proposed to estimate the vertical axis by segmenting the unit sphere, where each point represents a potential direction, using a segmentation network applied on ERPs. By combining vanishing points (VPs) with learned semantic features, their method integrates a residual stream that propagates vanishing point likelihoods, guiding the segmentation and improving camera leveling. Bergmann *et al*. [2021] employed a DenseNet121 to output the three components of the normalized upright vector $\mathbf{v} = [x \quad y \quad z]^\top$ and minimize the squared L_2 error between the normalized version of $\mathbf{v}$ and the ground truth unit vector.

Liu *et al*. [2024] introduced a method that avoids explicit rotation or lookup tables, treating upright adjustment as a pixel-wise image-to-image mapping problem. Shan *et al*. [2024] estimated camera inclination angles within a range of $\pm 90°$ combining convolutions and attention mechanisms to focus on local and global information, and also geometric distortions caused by camera inclination.

Next, we describe the pipeline employed to validate the robustness of the SWHDC module in the task of gravity alignment in panoramas. We also show relevant evaluation and ablation studies to demonstrate the effectiveness of our convolutional module.

4.2.1 Dataset and Training Details

In line with previous works [Jung *et al*., 2019; Bergmann *et al*., 2021; Shan *et al*., 2024], we use the SUN360 dataset [Xiao *et al*., 2012], which is a large collection containing a variety of indoor and outdoor scene categories. However, this dataset does not provide pre-defined splits. For our study, we used the filtered version of the dataset provided by Bergmann *et al*. [2021] as well as their dataset splits, where the authors performed a filtration process to remove duplicates and avoid data leakage between the training, validation, and testing sets. The original SUN360 dataset contains 67,538 images,

however, 10,693 are duplicates (approximately 16%). By doing so, we provide a fair evaluation without data leakage that can potentially interfere with the results.

As the main goal is the integration of the SWHDC module into any existing backbone designed for different tasks, in this case, the gravity alignment, we followed the provided implementation of VectorUp [Bergmann *et al*., 2021] and modified their DenseNet-121 [Huang *et al*., 2017] backbone.

To ensure a fair evaluation, specially with VectorUp, we follow the same training procedure as mentioned in [Bergmann *et al*., 2021], with an input of 221×442. Gaussian noise and image blur were applied as data augmentation during training, with probabilities of 30% and 40%, respectively. The variance of the noise is randomly set between 10 and 50, and the blur is applied using a normalized box filter with kernel size randomly chosen from 3, 5, or 7. We train the models for 300 epochs with a learning rate of 10^{-3}, which reduces by a factor of 0.1 after 10 epochs without improvements in the angular error controlled by the "`ReduceLROnPlateau`" scheduler. All models were trained using pre-trained weights from ImageNet.

In the following, we provide an extensive analysis of the performance of the SWHDC module when integrated into different convolutional architectures for the task of gravity alignment, following the same procedure done in Section 4.1.3.

4.2.2 Gravity Alignment with SWHDC-based CNNs

We integrate our SWHDC module into the DenseNet121 backbone, the same used in [Jung *et al*., 2019, 2020; Bergmann *et al*., 2021] for fair comparison. The DenseNet121 architecture is modified by replacing each planar convolutional operator with kernel size $k > 1$ with one SWHDC block containing $N = 4$ dilated convolutions. Moreover, we did not replace convolution with kernel size $k = 1$, as dilations are not applicable since this kernel size is typically used as a linear transformation that adjusts the number of channels without modifying the spatial dimensions of the feature maps. We present the results of Davidson *et al*. [2020], Coarse2Fine [Shan and Li, 2019], Deep360Up [Jung *et al*., 2019], LUT-GAN [Chen *et al*., 2023], and VectorUp [Bergmann *et al*., 2021], even with possible different training procedures. As stated in Section 4.2.1, we used the filtered dataset and the splits provided by Bergmann *et al*. [2021], to ensure fair evaluation and avoid data leakage. The results on the testing set are presented in Table 6. We evaluated the performance using the average median error and the angular errors within thresholds of $5°$ and $12°$. According to a user study presented by Jung *et al*. [2019], these thresholds are regarded as "very satisfactory" and "satisfactory", respectively, based on the participants' ratings of image alignment quality when viewing synthetically unaligned VR images.

As stated in [Bergmann *et al*., 2021], VectorUp achieved an angular error smaller than $5°$ in 82.8% of the samples and an angular error smaller than $12°$ in 97.0% of the samples from the filtered SUN360 dataset. In comparison, the integration of SWHDCs on VectorUp resulted in improved results. More precisely, this SWHDC-based VectorUp version obtained an angular error smaller than $5°$ in 86.5% of the samples, and an angular error smaller than $12°$ in 98.6% of the samples. Also, the median angular error obtained by VectorUp was $3.61°$, while our convolutional module obtained a median angular error of $2.59°$. Qualitative comparison between VectorUp and its version coupled with SWHDCs can be observed in Figure 8. Our results demonstrate that the modified DenseNet121 with SWHDCs performs comparably to the state-of-the-art methods. Notably, our model outperforms VectorUp in the percentage of samples with angular error below $5°$ and $12°$, and achieves a lower maximum median error.

Table 6. Results comparison between with the state-of-the-art for the task of gravity alignment in panoramas. Methods marked with * were trained on the same splits.

Method	Median Error	5°	12°
Davidson *et al*. [2020]	-	**92.6%**	98.4%
Davidson *et al*. [2020] (w/o VP)	-	68.2%	97.4%
Coarse2Fine [Shan and Li, 2019]	-	79.1%	91.0%
Deep360Up [Jung *et al*., 2019]	5.2°	90.2%	96.4%
LUT-GAN [Chen *et al*., 2023]	-	89.2%	95.2%
VectorUp* [Bergmann *et al*., 2021]	3.61°	82.8%	97.0%
VectorUp + SWHDC*	**2.59°**	86.5%	**98.6%**

Table 7. Comparison between DenseNet-121 backbone in VectorUp and ConvNeXt variants.

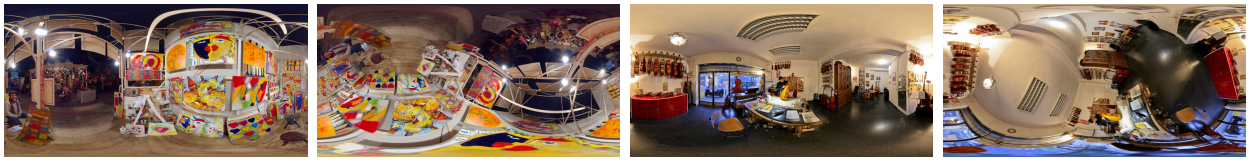
Method	Median Error	5°	12°
VectorUp	3.61°	82.7%	97%
ConvNeXt-tiny	1.68°	92.8%	99.2%
ConvNeXt-small	1.72°	93.1%	99.2%
VectorUp + SWHDC	2.59°	86.5%	98.6%
ConvNeXt-tiny + SWHDC	1.54°	94.7%	99.4%
ConvNeXt-small + SWHDC	1.62°	94.4%	99.5%

A comprehensive analysis of the effectiveness of our SWHDC module is provided in Table 7. This analysis includes an evaluation of the performance of the recent ConvNeXt architecture [Liu *et al*., 2022b], as well as its performance when combined with SWHDCs, using the same training procedure described in Section 4.2.1. The results show that all versions of ConvNeXt examined outperform DenseNet-121 (VectorUp) across all evaluation metrics. Moreover, replacing convolutions with kernel sizes of 3×3 or larger with SWHDCs results in further performance improvements, emphasizing the robustness of our module and its effectiveness when integrated into modern architectures for other tasks than 3D object classification task, such as the gravity alignment task.

5 Conclusion

This study presented and extended evaluation of the SWHDC module specifically designed to handle distortions and improve feature extraction in omnidirectional images initially introduced in Stringhini *et al*. [2024a]. This module leverages horizontally dilated convolutions, with dilation rates adjusted for each row of the ERP or feature map to account for varying levels of distortions. By optimizing the kernel support for each row, the SWHDC module effectively mitigates the adverse effects of distortions in ERP images.

We further extended the evaluation of the SWHDCNet to demonstrate better its competitive and superior performance in the task of 3D object classification amongst several relevant methods proposed in the literature. By converting



(a) Original ground truth **(b)** Rotated input **(c)** Original ground truth **(d)** Rotated input

(e) VectorUp **(f)** VectorUp + SWHDC **(g)** VectorUp **(h)** VectorUp + SWHDC

Figure 8. Qualitative comparison between VectorUp and its version with SWHDCs.

3D meshes into multi-channel ERP images, the proposed classifier achieved the best result for the single-view category. Compared to point-, voxel-, and multi-view-based approaches, we obtained competitive and superior results by using a single image to represent each object.

Additionally, we validate the capabilities of our SWHDC module in other tasks than 3D object classification task, such as the gravity alignment task. By replacing all planar convolutions with a kernel size larger than 1 in the VectorUp with SWHDCs, we observed consistent improvements across all evaluation metrics, showcasing the robustness and versatility of our approach. Nevertheless, our convolutional module can be integrated into modern backbones to better tackle this task.

A relevant aspect of our SWHDC module is its ability to replace any planar convolution without increasing the number of trainable parameters. This feature ensures that the improved performance is achieved without significant overhead making the approach efficient and straightforward to integrate into existing backbones. Moreover, the observed are not restricted to large architectures, as the SWHDC module also enhances the performance of smaller and more computationally efficient models, making it particularly attractive for applications involving constrained hardware resources.

Declarations

Authors' Contributions

RMS contributed to the investigation, methodology, validation, and writing. TLTS contributed to the methodology and writing, and CRJ contributed to the conceptualization, methodology, and writing. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq) -, and Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul - Brasil (FAPERGS).

Availability of data and materials

The code generated during the current study is available at `https://github.com/rmstringhini/SWHDC`

References

Ai, H., Cao, Z., Zhu, J., Bai, H., Chen, Y., and Wang, L. (2022). Deep learning for omnidirectional vision: A survey and new perspectives. *arXiv preprint arXiv:2205.10468*. DOI: 10.48550/arXiv.2205.10468.

Bai, J., Qin, H., Lai, S., Guo, J., and Guo, Y. (2024). Glpanodepth: Global-to-local panoramic depth estimation. *IEEE Transactions on Image Processing*, 33:2936–2949. DOI: 10.1109/tip.2024.3386403.

Bello, S. A., Yu, S., Wang, C., Adam, J. M., and Li, J. (2020). Deep learning on 3d point clouds. *Remote Sensing*, 12(11):1729. DOI: 10.48550/arXiv.2001.06280.

Bergmann, M. A., Pinto, P. G., da Silveira, T. L., and Jung, C. R. (2021). Gravity alignment for single panorama depth inference. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 41–48. IEEE. DOI: 10.1109/sibgrapi54419.2021.00015.

Brock, A., Lim, T., Ritchie, J. M., and Weston, N. (2016). Generative and discriminative voxel modeling with convolutional neural networks. *preprint arXiv:1608.04236*. DOI: 10.48550/arxiv.1608.04236.

Cao, Z., Huang, Q., and Karthik, R. (2017). 3d object classification via spherical projections. In *2017 international conference on 3D Vision (3DV)*, pages 566–574. IEEE. DOI: 10.1109/3dv.2017.00070.

Carlsson et al., O. (2024). Heal-swin: A vision transformer on the sphere. In *IEEE/CVF CVPR*, pages 6067–6077. DOI: 10.48550/arXiv.2307.07313.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., *et al*. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*. DOI: 10.48550/arxiv.1512.03012.

Chen, H., Li, S., and Li, J. (2023). An end-to-end network for upright adjustment of panoramic images. *Procedia Computer Science*, 222:435–447. DOI: 10.1016/j.procs.2023.08.182.

Chen, H.-X., Li, K., Fu, Z., Liu, M., Chen, Z., and Guo, Y. (2021). Distortion-aware monocular depth estimation for omnidirectional images. *IEEE Signal Processing Letters*, 28:334–338. DOI: 10.1109/lsp.2021.3050712.

Cheraghian, A. and Petersson, L. (2019). 3dcapsule: Extending the capsule architecture to classify 3d point clouds. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1194–1202. IEEE. DOI: 10.1109/wacv.2019.00132.

Cho et al., S. (2022). Spherical transformer. *preprint arXiv:2202.04942*. DOI: 10.48550/arXiv.2202.04942.

Cohen, T., Geiger, M., Köhler, J., and Welling, M. (2017). Convolutional networks for spherical signals. *arXiv preprint arXiv:1709.04893*. DOI: 10.48550/arxiv.1709.04893.

Coors, B., Condurache, A. P., and Geiger, A. (2018). Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 518–533. DOI: 10.1007/978-3-030-01240-3_32.

da Silveira, T. L. and Jung, C. R. (2023). Omnidirectional visual computing: Foundations, challenges, and applications. *Computers & Graphics*. DOI: 10.2139/ssrn.4350212.

da Silveira, T. L., Pinto, P. G., Murrugarra-Llerena, J., and Jung, C. R. (2022). 3d scene geometry estimation from 360 imagery: A survey. *ACM Computing Surveys*, 55(4):1–39. DOI: 10.1145/3519021.

Dai et al., F. (2020). Dilated convolutional neural networks for panoramic image saliency prediction. In *IEEE ICASSP*, pages 2558–2562. DOI: 10.1109/icassp40776.2020.9053888.

Dai et al., Z. (2021). Coatnet: Marrying convolution and attention for all data sizes. *NeuIPS*, 34:3965–3977. DOI: 10.48550/arxiv.2106.04803.

Davidson, B., Alvi, M. S., and Henriques, J. F. (2020). 360 camera alignment via segmentation. In *European Conference on Computer Vision*, pages 579–595. Springer. DOI: 10.1007/978-3-030-58604-1_35.

Ding, B., Tang, L., Gao, Z., and He, Y. (2020). 3d shape classification using a single view. *IEEE Access*, 8:200812–200822. DOI: 10.1109/access.2020.3035583.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., *et al*. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. DOI: 10.48550/arxiv.2010.11929.

Eder, M., Shvets, M., Lim, J., and Frahm, J.-M. (2020). Tangent images for mitigating spherical distortion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12426–12434. DOI: 10.1109/cvpr42600.2020.01244.

Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. (2018). Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68. DOI: 10.1007/s11263-019-01220-1.

Esteves, C., Makadia, A., and Daniilidis, K. (2020). Spin-weighted spherical cnns. *NeuIPS*, 33:8614–8625. DOI: 10.48550/arxiv.2006.10731.

Esteves, C., Slotine, J.-J., and Makadia, A. (2023). Scaling spherical CNNs. In *ICML*, volume 202, pages 9396–9411. DOI: 10.48550/arXiv.2306.05420.

Esteves, C., Xu, Y., Allen-Blanchette, C., and Daniilidis, K. (2019). Equivariant multi-view networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1568–1577. DOI: 10.1109/iccv.2019.00165.

Fernandez-Labrador, C., Facil, J. M., Perez-Yus, A., Demonceaux, C., Civera, J., and Guerrero, J. J. (2020). Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters*, 5(2):1255–1262. DOI: 10.1109/lra.2020.2967274.

Garcia-Garcia, A., Gomez-Donoso, F., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M., and Azorin-Lopez, J. (2016). Pointnet: A 3d convolutional neural network for real-time object class recognition. In *2016 International joint conference on neural networks (IJCNN)*, pages 1578–1584. IEEE. DOI: 10.1109/ijcnn.2016.7727386.

Goldblum et al., M. (2024). Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *NeuIPS*, 36. DOI: 10.52202/075280-1277.

Gomez-Donoso, F., Escalona, F., Orts-Escolano, S., Garcia-Garcia, A., Garcia-Rodriguez, J., and Cazorla, M. (2022). 3dslicelenet: Recognizing 3d objects using a slice-representation. *IEEE Access*, 10:15378–15392. DOI: 10.1109/access.2022.3148387.

Guo, X., Sun, Y., Zhao, R., Kuang, L., and Han, X. (2022). Swpt: Spherical window-based point cloud transformer. In *Proceedings of the Asian Conference on Computer Vision*, pages 3034–3050. DOI: 10.1007/978-3-031-26319-4_24.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M. (2020). Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364. DOI: 10.1109/tpami.2020.3005434.

Hamdi, A., Giancola, S., and Ghanem, B. (2021). Mvtn: Multi-view transformation network for 3d shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11. DOI: 10.1109/iccv48922.2021.00007.

Han, Z., Lu, H., Liu, Z., Vong, C.-M., Liu, Y.-S., Zwicker, M., Han, J., and Chen, C. P. (2019). 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. *IEEE Transactions on Image Processing*, 28(8):3986–3999. DOI: 10.1109/tip.2019.2904460.

Han, Z., Shang, M., Liu, Z., Vong, C.-M., Liu, Y.-S., Zwicker, M., Han, J., and Chen, C. P. (2018). Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing*, 28(2):658–672. DOI: 10.1109/tip.2018.2868426.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. DOI: 10.1109/cvpr.2016.90.

Hoang, L., Lee, S.-H., and Kwon, K.-R. (2020). A 3d shape

recognition method using hybrid deep learning network cnn–svm. *Electronics*, 9(4):649. DOI: 10.3390/electronics9040649.

Hou, Q., Lu, C.-Z., Cheng, M.-M., and Feng, J. (2024). Conv2former: A simple transformer-style convnet for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. DOI: 10.1109/tpami.2024.3401450.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708. DOI: 10.1109/cvpr.2017.243.

Huang, Q., Wang, Y., and Yin, Z. (2020). View-based weight network for 3d object recognition. *Image and Vision Computing*, 93:103828. DOI: 10.1016/j.imavis.2019.11.006.

Jeon, J., Jung, J., and Lee, S. (2019). Deep upright adjustment of 360 panoramas using multiple roll estimations. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V 14*, pages 199–214. Springer. DOI: 10.1007/978-3-030-20873-8_13.

Jiang, C., Huang, J., Kashinath, K., Marcus, P., Niessner, M., *et al*. (2019a). Spherical cnns on unstructured grids. *arXiv preprint arXiv:1901.02039*. DOI: 10.48550/arxiv.1901.02039.

Jiang, J., Bao, D., Chen, Z., Zhao, X., and Gao, Y. (2019b). Mlvcnn: Multi-loop-view convolutional neural network for 3d shape retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8513–8520. DOI: 10.1609/aaai.v33i01.33018513.

Jung, R., Cho, S., and Kwon, J. (2020). Upright adjustment with graph convolutional networks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1058–1062. IEEE. DOI: 10.1109/icip40778.2020.9190715.

Jung, R., Lee, A. S. J., Ashtari, A., and Bazin, J.-C. (2019). Deep360up: A deep learning-based approach for automatic vr image upright adjustment. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1–8. IEEE. DOI: 10.1109/vr.2019.8798326.

Kanezaki, A., Matsushita, Y., and Nishida, Y. (2018). Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5010–5019. DOI: 10.1109/cvpr.2018.00526.

Klokov, R. and Lempitsky, V. (2017). Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. In *IEEE ICCV*, pages 863–872. DOI: 10.48550/arXiv.1704.01222.

Lee, J.-S. and Park, T.-H. (2022). Transformable dilated convolution by distance for lidar semantic segmentation. *IEEE Access*, 10:125102–125111. DOI: 10.1109/access.2022.3225556.

Li, J., Chen, B. M., and Lee, G. H. (2018). So-net: Self-organizing network for point cloud analysis. In *IEEE/CVF CVPR*, pages 9397–9406. DOI: 10.1109/cvpr.2018.00979.

Li, J., Liu, Z., Li, L., Lin, J., Yao, J., and Tu, J. (2023). Multi-view convolutional vision transformer for 3d object recognition. *Journal of Visual Communication and Image Representation*, 95:103906. DOI: 10.1016/j.jvcir.2023.103906.

Li, Y., Guo, Y., Yan, Z., Huang, X., Duan, Y., and Ren, L. (2022). Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2801–2810. DOI: 10.1109/cvpr52688.2022.00282.

Liang, Q., Wang, Y., Nie, W., and Li, Q. (2020). Mvcln: multi-view convolutional lstm network for cross-media 3d shape recognition. *IEEE Access*, 8:139792–139802. DOI: 10.1109/access.2020.3012692.

Ling, Z., Xing, Z., Zhou, X., Cao, M., and Zhou, G. (2023). Panoswin: a pano-style swin transformer for panorama understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17755–17764. DOI: 10.1109/cvpr52729.2023.01703.

Liu, A.-A., Zhou, H.-Y., Li, M.-J., and Nie, W.-Z. (2020). 3d model retrieval based on multi-view attentional convolutional neural network. *Multimedia Tools and Applications*, 79:4699–4711. DOI: 10.1007/s11042-019-7521-8.

Liu, H. and Tian, S. (2024). Deep 3d point cloud classification and segmentation network based on gatenet. *The Visual Computer*, 40(2):971–981. DOI: 10.1007/s00371-023-02826-w.

Liu, J., Chen, H., Li, S., and Li, J. (2024). Generation of upright panoramic image from non-upright panoramic image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5261–5270. DOI: 10.1109/wacv57701.2024.00518.

Liu, Y., Tian, B., Lv, Y., Li, L., and Wang, F.-Y. (2023). Point cloud classification using content-based transformer via clustering in feature space. *IEEE/CAA Journal of Automatica Sinica*, 11(1):231–239. DOI: 10.1109/jas.2023.123432.

Liu, Y., Wang, Y., Du, H., and Cai, S. (2022a). Spherical transformer: Adapting spherical signal to convolutional networks. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 15–27. Springer. DOI: 10.1007/978-3-031-18913-5_2.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022. DOI: 10.1109/iccv48922.2021.00986.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022b). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986. DOI: 10.1109/cvpr52688.2022.01167.

Lu, D., Xie, Q., Gao, K., Xu, L., and Li, J. (2022). 3dctn: 3d convolution-transformer network for point cloud classification. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24854–24865. DOI: 10.1109/tits.2022.3198836.

Luo, W., Zhang, H., Ni, P., and Tian, X. (2020). Balanced principal component for 3d shape recognition using convolutional neural networks. *IET Image Processing*, 14(17):4468–4476. DOI: 10.1049/iet-ipr.2019.0844.

Ma, C., Guo, Y., Yang, J., and An, W. (2018). Learning multi-view representation with lstm for 3-d shape recognition and retrieval. *IEEE Transactions on Multimedia*, 21(5):1169–1182. DOI: 10.1109/tmm.2018.2875512.

Ma, X., Qin, C., You, H., Ran, H., and Fu, Y. (2022). Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*. DOI: 10.48550/arxiv.2202.07123.

Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE. DOI: 10.1109/iros.2015.7353481.

Mirbauer, M., Krabec, M., Křivánek, J., and Šikudová, E. (2021). Survey and evaluation of neural 3d shape classification approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8635–8656. DOI: 10.1109/tpami.2021.3102676.

Mohammadi, S. S., Wang, Y., and Del Bue, A. (2021). Pointview-gcn: 3d shape classification with multi-view point clouds. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3103–3107. IEEE. DOI: 10.1109/icip42928.2021.9506426.

Muzahid, A., Han, H., Zhang, Y., Li, D., Zhang, Y., Jamshid, J., and Sohel, F. (2024). Deep learning for 3d object recognition: A survey. *Neurocomputing*, page 128436. DOI: 10.1016/j.neucom.2024.128436.

Pintore, G., Agus, M., Almansa, E., Schneider, J., and Gobbetti, E. (2021). Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11536–11545. DOI: 10.1109/cvpr46437.2021.01137.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660. DOI: 10.1109/cvpr.2017.16.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30. DOI: 10.48550/arxiv.1706.02413.

Qi, S., Ning, X., Yang, G., Zhang, L., Long, P., Cai, W., and Li, W. (2021). Review of multi-view 3d object recognition methods based on deep learning. *Displays*, 69:102053. DOI: 10.1016/j.displa.2021.102053.

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., and Ghanem, B. (2022). Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204. DOI: 10.52202/068431-1685.

Rawat, W. and Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449. DOI: 10.1162/neco$_{a0}$0990.

Rey-Area, M., Yuan, M., and Richardt, C. (2022). 360monodepth: High-resolution 360deg monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3762–3772. DOI: 10.48550/arXiv.2111.15669.

Riegler, G., Osman Ulusoy, A., and Geiger, A. (2017). Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586. DOI: 10.1109/cvpr.2017.701.

Schuster et al., R. (2019). SDC-stacked dilated convolution: A unified descriptor network for dense matching tasks. In *IEEE/CVF CVPR*, pages 2556–2565. DOI: 10.1007/978-3-319-67199-4$_1$03425.

Sedaghat, N., Zolfaghari, M., Amiri, E., and Brox, T. (2016). Orientation-boosted voxel nets for 3d object recognition. *arXiv preprint arXiv:1604.03351*. DOI: 10.5244/c.31.97.

Sfikas, K., Pratikakis, I., and Theoharis, T. (2018). Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval. *Computers & Graphics*, 71:208–218. DOI: 10.1016/j.cag.2017.12.001.

Shan, Y., Chen, H., Zhang, J., Li, S., and Li, J. (2024). Multi-scale attention-based inclination angles estimation for panoramic camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1322–1330. DOI: 10.1109/cvprw63382.2024.00139.

Shan, Y. and Li, S. (2019). Discrete spherical image representation for CNN-based inclination estimation. *IEEE Access*, 8:2008–2022. DOI: 10.1109/ACCESS.2019.2962133.

Shen et al., Z. (2022). Panoformer: Panorama transformer for indoor 360° depth estimation. In *ECCV*, pages 195–211. DOI: 10.1007/978-3-031-19769-7$_1$2.

Shi, B., Bai, S., Zhou, Z., and Bai, X. (2015). Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343. DOI: 10.1109/lsp.2015.2480802.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *preprint arXiv:1409.1556*. DOI: 10.48550/arxiv.1409.1556.

Stringhini, R. M., da Silveira, T. L., and Jung, C. R. (2024a). Spherically-weighted horizontally dilated convolutions for omnidirectional image processing. In *2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 1–6. IEEE. DOI: 10.1109/sibgrapi62404.2024.10716273.

Stringhini, R. M., Lermen, T. S., Da Silveira, T. L., and Jung, C. R. (2024b). Single-panorama classification of 3d objects using horizontally stacked dilated convolutions. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 3436–3442. IEEE. DOI: 10.1109/icip51287.2024.10647442.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953. DOI: 10.1109/iccv.2015.114.

Su, J.-C., Gadelha, M., Wang, R., and Maji, S. (2018). A deeper look at 3D shape classifiers. In *ECCV Workshops*, pages 0–0. DOI: 10.48550/arXiv.1809.02560.

Su, Y. and Grauman, K. (2017a). Learning spherical convolution for fast features from 360 imagery. *NeuIPS*, 30. DOI: 10.48550/arxiv.1708.00919.

Su, Y. and Grauman, K. (2019). Kernel transformer networks

for compact spherical convolution. In *IEEE/CVF CVPR*, pages 9442–9451. DOI: 10.1109/cvpr.2019.00967.

Su, Y.-C. and Grauman, K. (2017b). Learning spherical convolution for fast features from 360 imagery. *NIPS*, 30. DOI: 10.48550/arxiv.1708.00919.

Sun, K., Zhang, J., Liu, J., Yu, R., and Song, Z. (2020). Drcnn: Dynamic routing convolutional neural network for multi-view 3d object recognition. *IEEE Transactions on Image Processing*, 30:868–877. DOI: 10.1109/tip.2020.3039378.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR. DOI: 10.48550/arxiv.1905.11946.

Tateno, K., Navab, N., and Tombari, F. (2018). Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 707–722. DOI: 10.1007/978-3-030-01270-0_43.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*. DOI: 10.65215/2q58a426.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., and Tong, X. (2017). O-cnn: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions On Graphics*, 36(4):1–11. DOI: 10.1145/3072959.3073608.

Wang, P.-S., Sun, C.-Y., Liu, Y., and Tong, X. (2018). Adaptive O-CNN: A patch-based deep representation of 3D shapes. *ACM Transactions on Graphics*, 37(6):1–11. DOI: 10.1145/3272127.3275050.

Wei, X., Yu, R., and Sun, J. (2020). View-gcn: View-based graph convolutional network for 3d shape analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1850–1859. DOI: 10.1109/cvpr42600.2020.00192.

Wu, C., Zheng, J., Pfrommer, J., and Beyerer, J. (2023). Attention-based point cloud edge sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343. DOI: 10.1109/cvpr52729.2023.00516.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920. DOI: 10.1109/cvpr.2015.7298801.

Xiao, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2012). Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2695–2702. IEEE. DOI: 10.1109/cvpr.2012.6247991.

Xu, C., Yang, H., Han, C., and Zhang, C. (2023). Pcformer: A parallel convolutional transformer network for 360 depth estimation. *IET Computer Vision*, 17(2):156–169. DOI: 10.1049/cvi2.12144.

Xu, Y., Zhang, Z., and Gao, S. (2021). Spherical dnns and their applications in 360° images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7235–7252. DOI: 10.1109/tpami.2021.3100259.

Yavartanoo, M., Kim, E. Y., and Lee, K. M. (2018). Spnet: Deep 3d object classification and retrieval using stereographic projection. In *Asian conference on computer vision*, pages 691–706. Springer. DOI: 10.1007/978-3-030-20873-8_44.

Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *preprint arXiv:1511.07122*. DOI: 10.48550/arxiv.1511.07122.

Yu, T., Meng, J., and Yuan, J. (2018). Multi-view harmonized bilinear network for 3d object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 186–194. DOI: 10.1109/cvpr.2018.00027.

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., and Lu, J. (2022). Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322. DOI: 10.1109/cvpr52688.2022.01871.

Yun, I., Shin, C., Lee, H., Lee, H.-J., and Rhee, C. E. (2023). Egformer: Equirectangular geometry-biased transformer for 360 depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6101–6112. DOI: 10.1109/iccv51070.2023.00561.

Zhang, J., Chen, Z., Lin, C., Nie, L., Shen, Z., Huang, J., and Zhao, Y. (2024). Sgformer: Spherical geometry transformer for 360 depth estimation. *arXiv preprint arXiv:2404.14979*. DOI: 10.1109/tcsvt.2025.3534220.

Zhang, R., Wang, L., Guo, Z., Wang, Y., Gao, P., Li, H., and Shi, J. (2023). Parameter is not all you need: Starting from non-parametric networks for 3d point cloud analysis. *arXiv preprint arXiv:2303.08134*. DOI: 10.48550/arxiv.2303.08134.

Zhao, H., Jiang, L., Jia, J., Torr, P. H., and Koltun, V. (2021). Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268. DOI: 10.1109/iccv48922.2021.01595.

Zhao, L., Xu, S., Liu, L., Ming, D., and Tao, W. (2022). Svaseg: Sparse voxel-based attention for 3d lidar point cloud semantic segmentation. *Remote Sensing*, 14(18):4471. DOI: 10.3390/rs14184471.

Zhi, S., Liu, Y., Li, X., and Guo, Y. (2018). Toward real-time 3d object recognition: A lightweight volumetric cnn framework using multitask learning. *Computers & Graphics*, 71:199–207. DOI: 10.1016/j.cag.2017.10.007.

Zhou, Y., Zeng, F., Qian, J., and Han, X. (2019). 3d shape classification and retrieval based on polar view. *Information Sciences*, 474:205–220. DOI: 10.1016/j.ins.2018.09.051.

Zhuang et al., C. (2022). Acdnet: Adaptively combined dilated convolution for monocular panorama depth estimation. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 3653–3661. DOI: 10.1609/aaai.v36i3.20278.

Zioulis, N., Karakottas, A., Zarpalas, D., and Daras, P. (2018). Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 448–465. DOI: 10.1007/978-3-030-01231-1_28.