Spherically-Weighted Horizontally Dilated Convolutions for Omnidirectional Image Processing

Romulo M. Stringhini, Thiago L. T. da Silveira, and Claudio R. Jung
Institute of Informatics, Federal University of Rio Grande do Sul
Porto Alegre, Brazil
{rmstringhini, tltsilveira, crjung}@inf.ufrgs.br

***Abstract*—Traditional convolutional neural networks (CNNs) face significant challenges when applied to omnidirectional images due to the non-uniform sampling inherent in equirectangular projection (ERP). This projection type leads to distortions, particularly near the poles of the ERP image, and fixed-size kernels in planar CNNs are not designed to address this issue. This paper introduces a convolutional block called Spherically-Weighted Horizontally Dilated Convolutions (SWHDC). Our block mitigates distortions during the feature extraction phase by properly weighting dilated convolutions according to the optimal support for each row in the ERP, thus enhancing the ability of a network to process omnidirectional images. We replace planar convolutions of well-known backbones with our SWHDC block and test its effectiveness in the 3D object classification task using ERP images as a case study. We considered standard benchmarks and compared the results with state-of-the-art methods that convert 3D objects to single 2D images. The results show that our SWHDC block improves the classification performance of planar CNNs when dealing with ERP images without increasing the number of parameters, outperforming peering methods. Code is available at: https://github.com/rmstringhini/SWHDC**

I. Introduction

Omnidirectional images, also known as spherical images, are captured by 360-degree cameras and provide a panoramic field-of-view of $180^\circ \times 360^\circ$, much broader than that of traditional pinhole cameras [1]. With its capability to capture the entire surrounding environment, omnidirectional images can be applied to various scenarios such as autonomous driving, augmented/virtual reality, and robotics [2].

Spherical images, which are signals defined on a spherical surface, are frequently mapped to a planar format using the equirectangular projection (ERP) [1], [2]. ERP samples the unit sphere non-uniformly and produces distortions when mapping data to the plane [1]. This means that pixels at higher latitudes represent smaller superficial areas on the sphere than those at the equator. Consequently, standard convolutional neural networks (CNNs) designed for structured planar imagery are not well-suited for omnidirectional images since they use standard convolutional kernels with fixed support [1], [2].

Many studies try to deal with distortions in ERP images, offering different solutions to improve the feature extraction capacity of CNNs, such as the Kernel Transformer Network [3] or SphereNet [4]. However, these approaches require high computational resources as kernels that sample irregularity are generally slower than traditional convolutions [1]. Dilated convolutions [5]–[8] are used to handle distortions in 360-degree images due to their wider receptive field and the possibility of capturing long-range dependencies. However, since the sphere's curvature causes variations in area and distance between points along different rows, the best fit for dilated convolutions in spherical images must consider different dilation rates for different rows [1].

The Vision Transformer (ViT) architecture [9] has also been adapted for spherical image processing tasks using different sampling strategies for extracting the tokens [10]–[13]. Despite the widespread of transformers in the past years, architectures based on CNNs or combinations of CNNs and ViTs can outperform pure ViT models, particularly when training data is not abundant [14], [15].

This paper proposes a convolutional block named Spherically-Weighted Horizontally Dilated Convolutions (SWHDC) designed to cope with the non-uniform sampling of ERP images. The SWHDC block contains multiple dilated convolutions along the horizontal dimension with a shared-weight kernel. The final output of the block is a linear combination of the multiple row-wise weighted feature maps of each convolution, where the row-dependent weights aim to select the optimal support based on the corresponding distortion. SWHDC block can be integrated into any planar CNN backbone to better extract features of spherical images without increasing the number of parameters. As a case study, we chose to evaluate the effectiveness of our block in the task of 3D object classification using spherical images that, like many others, exhibit limited data availability.

II. Related Work

Several different strategies have been used to deal with distortion in omnidirectional images. Su and Grauman [16] proposed a spherical convolution that adapts a planar network to handle 360-degree images by training distinct kernels for individual rows of the ERP, and a follow-up study [3] introduced the Kernel Transformer Network to transfer convolutional kernels from perspective images to ERP images. On the other hand, the idea of deforming convolutional filters to adapt their receptive fields according to the distortion levels was proposed in [17], [18]. Despite the promising results, these adaptations of planar filters are computationally demanding.

Another set of techniques explores convolutions defined on the sphere. Cohen et al. [19] explored the inner product between a spherical signal and a rotated spherical filter, using

the inherent rotational symmetry of spherical signals in a similar way to how standard convolutions networks leverage the translation symmetry in planar images. Esteves et al. [20] proposed a CNN that achieves 3D rotation invariance by performing convolutions on the sphere and pooling operations on the spectral domain to maintain equivariance. The spin-weighted version (SWSCNN) [21] avoids the need to lift data to $SO(3)$, and a fast implementation was introduced in [22]. Jiang and colleagues [23] explored linear combinations of differential operators to create a convolutional kernel that operates directly on icosahedral spherical meshes.

Yet, another class of approaches explores multiple representations or varying-size kernels to deal with panoramas. UniFuse [24] employs a unidirectional fusion approach to fuse and combine features from ERP and cube-map projections, while Bifuse [25] explores a two-branch network that incorporates both projections. Liu et al. [26] explored the HEALPix (Hierarchical Equal Area Iso Latitude Pixelation) representation to sample spherical data, and used pooling and convolution layers to perform different in the transformed domain. Zioulis et al. [27] proposed a direct use of omnidirectional images by transforming square convolutional filters into row-wise rectangles and adjusting filter sizes to be larger near the poles and smaller close to the equator. Pintore et al. [28] introduced a slice-based representation to exploit the characteristics of ERP along the vertical dimension directly, eliminating the need for distortion-aware convolutions. ACDNet [7] combines dilated convolutions with different horizontal and vertical dilation rates with a channel-wise fusion module to improve feature extraction. The transformable dilated convolution [8] dynamically adjusts the kernel size based on the distance of objects in spherical LiDAR data, using a larger kernel for closer objects and a smaller one for distant objects.

Finally, a recent trend is the adaptation of ViTs to the spherical domain. PanoFormer [12] extracts tangent patches to avoid distortions, using them as tokens. Similarly, OmniFusion [29] converts the ERP image to distortion-free patches with spherical and tangent plane center coordinates into an encoder-decoder network. Rey-Area et al. [30] projected the ERP input image onto a set of tangent planes to produce perspective views. HEAL-SWIN [13] proposes a modified version of the Swin Transformer [31] applied to a uniform HEALPix grid, while GLPanoDepth [10] proposes a Cubemap Vision Transformer (CViT) combined with a planar CNN to extract features directly from the ERP.

Even though several strategies aim to overcome the negative effects of non-uniform sampling in omnidirectional images, they typically demand considerably higher computational costs than their planar counterparts [1]. Furthermore, some of these approaches present similar or marginally superior results than traditional planar strategies [18]. Architectures based on ViTs have become very popular in the past years, and they can be adapted to the spherical domain by selecting adequate patches [10]–[13]. However, ViT-based approaches typically require larger training datasets to become effective [14], [15]. In fact, Goldblum et al. [15] recently evaluated several backbones, concluding that modern CNNs architectures pretrained via supervised learning perform better than ViTs on several vision tasks, whereas transformers benefit more from scale.

III. The Proposed Approach

In this section, we revisit the ERP mapping formulation and describe the proposed SWHDC block to handle the inherent ERP-induced distortions. Differently from [8], where kernel sizes are adjusted according to the distance of objects in LiDAR data, our approach spherically weights dilated convolutions according to the ideal support for each latitude in the ERP image (or feature map).

A. The Equirectangular Projection

The Spherical Camera Model projects a 3D world point $\boldsymbol{P} \in \mathbb{R}^3$ onto the unit sphere centered at $\boldsymbol{C} \in \mathbb{R}^3$ through central and spherical projections, resulting in the intersection point $\boldsymbol{p} \in \mathbb{S}^2$ [1]. Since $\boldsymbol{p}$ has unit distance from the camera center $\boldsymbol{C}$, it can be expressed as

$$\boldsymbol{p} = \begin{bmatrix} \cos\theta \sin\phi & \sin\theta \sin\phi & \cos\phi \end{bmatrix}^\top , \tag{1}$$

where $\phi \in [0, \pi)$ and $\theta \in [0, 2\pi)$ represent the latitudinal and longitudinal coordinates respectively.

Since this process can be applied to every angle pair (ϕ, θ), the spherical surface information can be arranged into a $[0, \pi) \times [0, 2\pi)$ rectangular grid [1]. This straightforward mapping is known as ERP [16], which represents a point $\boldsymbol{p} \in S^2$ in position (y, x) of a $h \times w$ 1:2 image (the ERP image) using

$$y = \left\lfloor \frac{\phi h}{\pi} \right\rfloor , \;\; x = \left\lfloor \frac{\theta w}{2\pi} \right\rfloor . \tag{2}$$

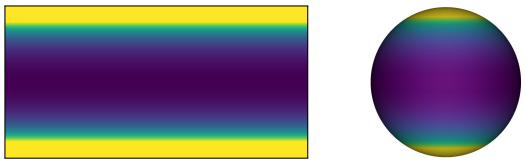
In ERP images, regions near the poles are more densely sampled than those near the equator [1]. The horizontal distance between two points on the sphere with longitudes θ_1 and θ_2, given a latitude ϕ, is expressed as $\sin\phi|\theta_1 - \theta_2|$, as detailed in Eq. (1). Conversely, for a fixed longitude 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, while the vertical support should remain the same.

B. Spherically-Weighted Horizontally Dilated Convolutions

In a traditional (planar) convolutional filter, the kernel has a fixed support. When applied to ERP images, it covers different areas of the sphere depending on its latitude. As shown in Fig. 1, the same kernel has a smaller support close to the poles. Our SWHDC block copes with the ERP distortions and can be integrated into any planar CNN to improve feature extraction. We adapt the idea of [32], where stacked parallel dilated convolutions are employed to process regular images. In [32], the outputs of the dilated convolutions (in both dimensions) are stacked to produce a multi-channel, multi-scale response. Our SWHDC block also relies on multiple parallel dilated convolutions but only dilates convolutions along the horizontal dimension. Instead of concatenating the filter responses, we

(a) (b)

Fig. 1. Applying regular fixed-support kernels at different latitudes to (a) ERP images covers uneven (b) sphere surface areas. Blue, green, and red squares represent the kernels (enlarged for visualization purposes).



(a) (b)

Fig. 2. Ideal dilation rates that should be adjusted depending on the latitude: blueish colors represent small dilation rates (not smaller than 1) while yellowish colors represent large dilation rates (in the limit case, infinity).

perform a row-dependent linear combination so that each latitude (row) of the ERP (or feature map) is influenced differently by each dilated convolution. This leads to a single-channel per kernel output for the SWHDC block.

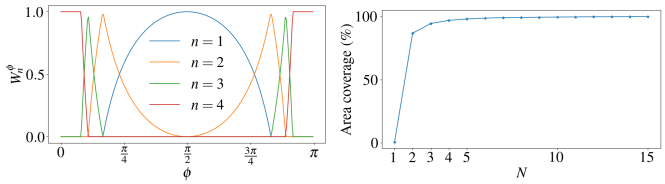
The SWHDC block employs N preset horizontally dilated convolutions with shared weights, each one with a horizontal dilation rate n that produces a feature map F_n, for $n = 1, \ldots, N$. This design choice allows a multi-support feature extraction, which can be appropriately weighted to capture ERP image information despite the distortions. We use circular padding to capture the full field-of-view of the input, preserving smooth horizontal continuity across spherical images. By sharing weights, the number of learnable parameters remains constant regardless of the number of dilated convolutions, providing computational efficiency.

Let us consider a kernel support relative to the equator line ($\phi = \pi/2$) of the ERP. As noted in Section III-A, the ideal kernel support for each latitude ϕ must be scaled by a factor $1/\sin\phi$ to cope with non-uniform sampling. Since the support is proportional to the dilation rate, we select the ideal row-wise dilation rate based on the factor $1/\sin\phi$ as illustrated in Fig. 2 (blueish colors represent smaller values while yellowish colors represent larger ones). Since this factor yields non-integer values, we interpolate between the two closest dilation rates by performing weighted averages, as explained next.

Each row index y of the SWHDC block's input (ERP image or feature map) relates to a latitude ϕ according to Eq. (2), for which the ideal scaling factor is

$$R_\phi = \min\{N, 1/\sin\phi(y)\}, \tag{3}$$

noting that we already limit the maximum scaling factor R_ϕ to the largest dilation N. The weight W_n^ϕ for dilation rate n and row index related to a latitude ϕ is given by interpolating



(a) (b)

Fig. 3. (a) Distribution of the weights W_n^ϕ according to ϕ when $N = 4$. (b) Percentage of area coverage on the spherical surface for N horizontally dilated convolutions.

the two closest integer scales, i.e.,

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

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

Finally, the combined output feature map F_* is given by a linear combination of the feature maps F_n resulting from the preset of horizontally dilated convolutions:

$$F_* = \sum_{n=1}^{N} H_B(W_n) \odot F_n \tag{5}$$

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

Our SWHDC block maintains the same number of parameters as regular convolutional blocks by using hardcoded row-dependent weights to mimic the optimal support for each horizontal dilated convolution. Despite employing multiple dilated convolutions, the number of output channels in the final combined feature map of the SWHDC block remains the same as in traditional convolutional blocks. This careful design choice ensures that the computational efficiency is preserved, allowing an improved feature extraction while mitigating distortions without adding any parameter overhead. Fig. 4 shows the SWHDC block architecture for $N = 4$.

In the design of our SWHDC block, we chose $N = 4$ as our preset amount of horizontally dilated convolutions. This decision is derived from the observation that the area covered by a dilated convolution in a spherical surface barely increases for $N > 4$. According to Fig. 3b, the area covered by dilated convolutions with an ideal kernel support ranging from rates 1 to 2 is notably higher compared to higher rates. For $N = 4$, the receptive field is sufficiently large to extract significant features and mitigate distortions effectively as when $N \geq 5$, since the additional coverage area becomes minimal ($\approx 1.1\%$). Thus, choosing $N = 4$ is sufficient to handle distortions near the poles of ERP images. In the particular case for $N = 1$, where the dilation rate is $n = 1$, the expansion of the kernel is not required and a traditional convolution is used. In this case, the feature extraction is performed directly along the

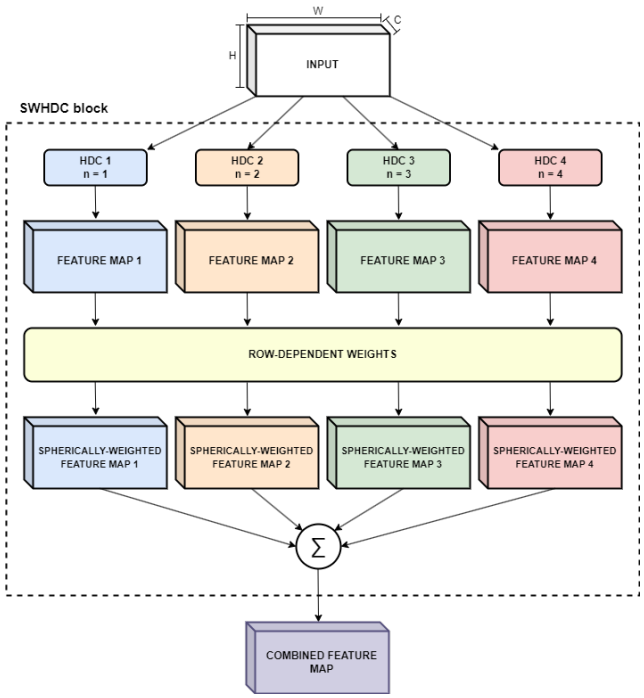


Fig. 4. Composition of our SWHDC block when $N = 4$. The input is processed by N horizontally dilated convolutions with different dilation rates n. Each feature map passes through a row-dependent weighting. Then, all N spherically-weighted feature maps are combined to generate the final combined feature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

equator line. The selection of N is based on our analysis of area coverage on a spherical surface and performance results, which are provided further in Section IV. Regardless the value of N, the number of trainable parameters remains the same.

The focus of this paper is to overcome the distortion-related issues inherent to ERP images. Then, we can embed our SWHDC block into existing planar architectures to effectively mitigate the impact of distortions in the feature extraction phase without increasing the number of trainable parameters. As detailed next, we evaluate our convolutional block in 3D object classification using spherical images as a case study.

IV. Experimental Results and Discussions

This section presents the case study application (3D object classification using spherical images), the datasets used, and the experiments and results, assessing different backbones and comparing the best-performing one with peering methods. Our goal is to show the effectiveness of our SWHCD block in the target case study application.

A. Classification of 3D Objects Using ERP Images

This paper aims to mitigate distortions inherent in ERP images by integrating our SWHDC block into planar CNNs. By doing so, we improve the performance of planar backbones for 3D object classification using spherical images.

To generate the 2D spherical views, we followed the approach used by other methods that explore panoramas for 3D

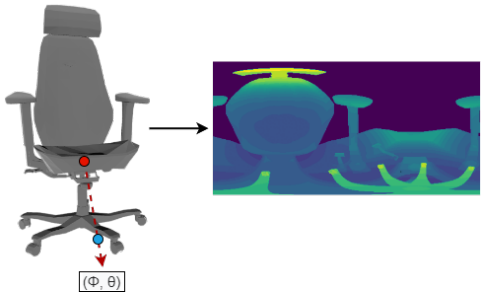


Fig. 5. Construction of the ERP image. Omnidirectional rays are cast from the object's centroid (red dot), with orientation defined by ϕ and θ. The distance between the centroid and the last intersection point (blue dot) is stored in a pixel position (y, x) to generate the external depth map of the object.

shape classification [20], [21], [26]. The core idea is to cast rays omnidirectionally from the object's centroid until they intersect the shape (or the convex hull, as in [26]), and retrieve local geometrical information at the intersection point.

The orientation of each ray is defined by ϕ and θ relative to the centroid of the object and mapped to an image pixel position (y, x) according to Eq. (2). In this work, we used $h = 256$ and $w = 512$, a common choice for panorama-based deep learning approaches [1]. The ERP image is obtained by calculating the distance between the origin of the ray and the position of the last intersection point in the object (i.e., the depth), which is then projected to a pixel position (y, x) on the ERP image. We encode the position (y, x) with zero distance if no hitting point exists between the ray and the object. The resulting ERP image can be interpreted as an external depth map of the object since we store information on the last intersection point between the ray and the object. To achieve scale invariance, we divide all distance values by the maximum distance within the object, yielding a normalized map. This process is illustrated in Fig. 5.

Our experiments are performed on the Princeton ModelNet [33] datasets, which contain CAD models divided into different categories. ModelNet10 is a subset that contains models from 10 categories divided into 3,991 training and 908 testing models. ModelNet40 contains 40 different categories with 9,843 and 2,468 models for training and testing, respectively. For both datasets, we randomly split the preset training set into 80% for training and 20% for validation.

We trained the evaluated backbones for a maximum of 200 epochs using early stopping with a patience of 25 epochs. We used the Adam optimizer 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 x and y axes, 0-45° for 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%.

B. Embedding SWHDC block into Planar CNNs

We analyzed three well-established planar CNN architectures: VGG [34], ResNets [35], and EfficientNets [36]. These

TABLE I
CLASSIFICATION RESULTS OF DIFFERENT BACKBONES AND CONVOLUTIONAL BLOCKS.

Backbone	ModelNet10	ModelNet40	Params.
VGG-16	90.30%	85.53%	138.3M
VGG-16+SWHDC	91.92%	88.21%	138.3M
ResNet-18	89.09%	86.66%	11.7M
ResNet-18+SWHDC	92.38%	89.41%	11.7M
ResNet-18+SPH	90.39%	87.83%	16.4M
ResNet-34	90.64%	87.44%	21.8M
ResNet-34+SWHDC	93.89%	90.87%	21.8M
ResNet-34+SPH	92.76%	89.52%	26.9M
ResNet-50	91.07%	87.88%	25.5M
ResNet-50+SWHDC	94.11%	91.89%	25.5M
ResNet-50+SPH	92.66%	90.13%	31.4M
EffNet-b0	90.22%	86.81%	52.8M
EffNet-b0+SWHDC	93.21%	89.95%	52.8M
EfftNet-b7	90.93%	87.21%	66.3M
EffNet-b7+SWHDC	93.80%	90.06%	66.3M

architectures were evaluated using both standard convolutions and our SWHDC block. To investigate and compare their performance on the target task, all models were trained from scratch using only our ERP images, without pretraining.

Table I shows the classification results of different backbones on the ModelNet10 and ModelNet40 datasets, comparing their performance with standard convolutions and our SWHDC block. As observed, integrating our convolutional block consistently improved the performance of *all* tested backbones on both ModelNet datasets without increasing the number of parameters. We also evaluated the integration of spherical convolutions (SPH) from SphereNet [4] into ResNet backbones as they outperformed the compared planar backbones. Although spherical convolutions improve the baseline results, they require more training parameters. Besides keeping the number of parameters unaltered, our SWHDC block is particularly advantageous when compared to spherical convolutions, which typically require more computational resources as they involve complex operations.

As stated in Section III-B, when the amount of horizontally dilated convolutions inside our block is $N = 4$, we cover $\approx 96.85\%$ of the spherical surface indicating effective feature extraction and distortion handling. As we increase the dilation rate to $N = 5$, there is minimal change in the covered area ($\approx 1.1\%$). It suggests that while higher dilation rates may expand the receptive field, they may not contribute substantially to increased coverage on the sphere, especially when the feature map is too small. Table II provides the results for the ModelNet40 dataset of a modified ResNet-18 backbone by integrating our SWHDC block containing different amounts of horizontally dilated convolutions.

C. Comparison with the State-of-the-Art

Several methods have been proposed to classify 3D objects, including approaches based on single or multiple views per object, as well as point- and voxel-based methods, which process raw point clouds or convert 3D objects into a grid of voxels, respectively. To ensure a fair comparison, our

TABLE II
RESULTS VARYING THE AMOUNT OF N HORIZONTALLY DILATED CONVOLUTIONS INSIDE OUR BLOCK ON THE MODELNET10 DATASET.

Backbone	ModelNet10
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%

TABLE III
SHAPE CLASSIFICATION RESULTS ON MODELNET. BEST RESULT IN BOLD, SECOND-BEST UNDERLINED.

Method/Backbone	ModelNet10	ModelNet40
DeepPano [37]	88.66%	82.54%
PVR [38]	<u>92.73%</u>	91.69%
Cao et al. [39] (from scratch)	-	86.09%
SPNet [40] (panoramic)	92.07%	-
Ding et al. [41]	91.18%	89.01%
Hoang et al. [42]	91.08%	85.82%
PanoFormer encoder [12] + FC	85.74%	79.71%
SWSCNN [21]	-	90.10%
STM [26]	-	**92.70%**
ResNet-50+SWHDC (Our)	**94.11%**	<u>91.89%</u>

study exclusively benchmarks against methods that employ a single view per object, focusing on a consistent evaluation of classification performance. We compared the best-performing ResNet model supplied with our SWHDC blocks (called *our* model) with the results of DeepPano [37], PVR [38], Cao et al. [39], SPNet [40], Ding et al. [41], Hoang et al. [42], SWSCNN [21], and STM [26]. Additionally, we trained a classifier with a recent transformer-based encoder to assess its accuracy with relatively small datasets such as ModelNet10 or ModelNet40. More precisely, we adapted the ViT-based encoder of PanoFormer [12] for the task of classification by adding a fully connected (FC) layer. Here, we feed our ERP images that encode the geometry of the objects to the PanoFormer encoder.

A summary of the results is provided in Table III. For ModelNet10, our approach achieved the best accuracy compared to all other methods that use a single image to represent 3D objects. For ModelNet40, our results were inferior only to STM [26]. However, STM takes as input a 6-channel panorama that includes information about depth values and normal vectors of the mesh and its convex hull[1], whereas our approach uses only a single-channel image (depth). Tests with PanoFormer yielded low accuracy values compared to other approaches. However, it is important to mention that we used the encoder originally designed for depth estimation in spherical images, so it might be over-dimensioned for the classification task with ModelNet. Another possible explanation for the low accuracy of the ViT-based PanoFormer encoder is the data-hunger nature of transformers.

[1]STM [26] also reports results combining single and multiple views, but the accuracy is marginally superior – 93.00%.

V. Conclusions

This work introduced the SWHDC block for CNNs. Our SWHDC block is designed to mitigate the adverse effects of non-uniform sampling in ERP images and improve the feature extraction capabilities of traditional planar backbones when processing omnidirectional images. By incorporating multiple weighted dilated convolutions that expand the horizontal receptive field only and mimic the optimal support according to the latitude of the ERP, our approach captures richer spatial information and mitigates distortions.

We have shown the effectiveness of our SWHDC block by plugging it into different planar backbones and comparing the results with their planar counterparts in the 3D object classification task using spherical images. We also compare results with a spherical convolution design. Additionally, we achieved better results than state-of-the-art methods that use a single image to represent 3D objects.

A key advantage of our SWHDC block is its ability to replace standard convolutions without increasing the number of parameters, ensuring that the enhanced performance does not come at the cost of additional trainable parameters. This efficiency makes our proposal highly practical and versatile, allowing easy integration into any existing CNN backbone.

In future work, we plan to assess the effectiveness of our SWHDC block in other applications. We plan to encompass different tasks such as depth estimation, object detection, semantic segmentation, and gravity alignment, to analyze our convolutional block's adaptability and generalizability comprehensively.

Acknowledgments

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).

References

[1] T. L. da Silveira and C. R. Jung, "Omnidirectional visual computing: Foundations, challenges, and applications," *Computers & Graphics*, 2023.

[2] H. Ai et al., "Deep learning for omnidirectional vision: A survey and new perspectives," *preprint arXiv:2205.10468*, 2022.

[3] Y. Su and K. Grauman, "Kernel transformer networks for compact spherical convolution," in *IEEE/CVF CVPR*, 2019, pp. 9442–9451.

[4] B. Coors et al., "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in *ECCV*, 2018, pp. 518–533.

[5] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *preprint arXiv:1511.07122*, 2015.

[6] F. Dai et al., "Dilated convolutional neural networks for panoramic image saliency prediction," in *IEEE ICASSP*, 2020, pp. 2558–2562.

[7] C. Zhuang et al., "Acdnet: Adaptively combined dilated convolution for monocular panorama depth estimation," in *AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 3653–3661.

[8] J.-S. Lee and T.-H. Park, "Transformable dilated convolution by distance for lidar semantic segmentation," *IEEE Access*, vol. 10, pp. 125 102–125 111, 2022.

[9] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *preprint arXiv:2010.11929*, 2020.

[10] J. Bai, H. Qin, S. Lai, J. Guo, and Y. Guo, "Glpanodepth: Global-to-local panoramic depth estimation," *IEEE Transactions on Image Processing*, vol. 33, pp. 2936–2949, 2024.

[11] S. Cho et al., "Spherical transformer," *preprint arXiv:2202.04942*, 2022.

[12] Z. Shen et al., "Panoformer: Panorama transformer for indoor 360° depth estimation," in *ECCV*, 2022, pp. 195–211.

[13] O. Carlsson et al., "Heal-swin: A vision transformer on the sphere," in *IEEE/CVF CVPR*, 2024, pp. 6067–6077.

[14] Z. Dai et al., "Coatnet: Marrying convolution and attention for all data sizes," *NeuIPS*, vol. 34, pp. 3965–3977, 2021.

[15] M. Goldblum et al., "Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks," *NeuIPS*, vol. 36, 2024.

[16] Y. Su and K. Grauman, "Learning spherical convolution for fast features from 360 imagery," *NeuIPS*, vol. 30, 2017.

[17] K. Tateno et al., "Distortion-aware convolutional filters for dense prediction in panoramic images," in *ECCV*, 2018, pp. 707–722.

[18] C. Fernandez-Labrador et al., "Corners for layout: End-to-end layout recovery from 360 images," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1255–1262, 2020.

[19] T. Cohen et al., "Convolutional networks for spherical signals," *preprint arXiv:1709.04893*, 2017.

[20] C. Esteves et al., "Learning so(3) equivariant representations with spherical cnns," in *ECCV*, 2018, pp. 52–68.

[21] C. Esteves, A. Makadia, and K. Daniilidis, "Spin-weighted spherical cnns," *NeuIPS*, vol. 33, pp. 8614–8625, 2020.

[22] C. Esteves, J.-J. Slotine, and A. Makadia, "Scaling spherical CNNs," in *ICML*, vol. 202, 2023, pp. 9396–9411.

[23] C. Jiang et al., "Spherical cnns on unstructured grids," *arXiv preprint arXiv:1901.02039*, 2019.

[24] H. Jiang et al., "Unifuse: Unidirectional fusion for 360 panorama depth estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1519–1526, 2021.

[25] F. Wang et al., "Bifuse: Monocular 360 depth estimation via bi-projection fusion," in *IEEE/CVF CVPR*, 2020, pp. 462–471.

[26] Y. Liu et al., "Spherical transformer: Adapting spherical signal to convolutional networks," in *PRCV*. Springer, 2022, pp. 15–27.

[27] N. Zioulis et al., "Omnidepth: Dense depth estimation for indoors spherical panoramas," in *ECCV*, 2018, pp. 448–465.

[28] G. Pintore et al., "Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation," in *IEEE/CVF CVPR*, 2021, pp. 11 536–11 545.

[29] Y. Li et al., "Omnifusion: 360 monocular depth estimation via geometry-aware fusion," in *IEEE/CVF CVPR*, 2022, pp. 2801–2810.

[30] M. Rey-Area et al., "360monodepth: High-resolution 360deg monocular depth estimation," in *IEEE/CVF CVPR*, 2022, pp. 3762–3772.

[31] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF CVPR*, 2021, pp. 10 012–10 022.

[32] R. Schuster et al., "SDC-stacked dilated convolution: A unified descriptor network for dense matching tasks," in *IEEE/CVF CVPR*, 2019, pp. 2556–2565.

[33] Z. Wu et al., "3D shapenets: A deep representation for volumetric shapes," in *IEEE/CVF CVPR*, 2015, pp. 1912–1920.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *preprint arXiv:1409.1556*, 2015.

[35] K. He et al., "Deep residual learning for image recognition," in *IEEE/CVF CVPR*, 2016, pp. 770–778.

[36] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*. PMLR, 2019, pp. 6105–6114.

[37] B. Shi et al., "Deeppano: Deep panoramic representation for 3-d shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[38] Y. Zhou et al., "3D shape classification and retrieval based on polar view," *Information Sciences*, vol. 474, pp. 205–220, 2019.

[39] Z. Cao et al., "3D object classification via spherical projections," in *International Conference on 3D Vision*, 2017, pp. 566–574.

[40] M. Yavartanoo et al., "Spnet: Deep 3D object classification and retrieval using stereographic projection," in *ACCV*, 2018, pp. 691–706.

[41] B. Ding et al., "3D shape classification using a single view," *IEEE Access*, vol. 8, pp. 200 812–200 822, 2020.

[42] L. Hoang et al., "A 3D shape recognition method using hybrid deep learning network cnn–svm," *Electronics*, vol. 9, no. 4, p. 649, 2020.