IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 20, 2019, accepted September 26, 2019, date of publication September 30, 2019, date of current version October 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944766

Hexagonal Convolutional Neural Networks for Hexagonal Grids

JUNREN LUO, WANPENG ZHANG, JIONGMING SU, AND FENGTAO XIANG
College of Intelligence Science and Technology, National University of Defense and Technology, Changsha 410073, China
Corresponding author: Wanpeng Zhang (wpzhang@nudt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61806212 and Grant 61603403.

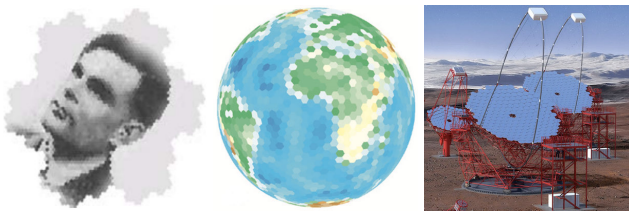
ABSTRACT Hexagonal grids use a hierarchical subdivision tessellation to cover the entire plane or sphere. Due to the 6-fold rotational symmetry, hexagonal grids have some advantages (e.g. isoperimetry, equidistant neighbors, and uniform connectivity) over quadrangular and triangular girds, which makes them suitable to tackle tasks of geospatial information processing and intelligent decision-making. In this paper, we first introduce some applications based on the hexagonal grids. Then, we introduce the planer and spherical hexagonal grids and analyze the group representations for them, we review geometric deep learning, some Convolutional Neural Networks (CNNs) for hexagonal grids, and group-based equivariant convolution. Next in importance, we propose the HexagonNet for hexagonal grids, and define a new convolution operator and pooling operator. Finally, in order to evaluate the effectiveness of the proposed HexagonNet, we perform experiments on two tasks: aerial scene classification on the aerial image dataset (AID), and 3D shape classification on the ModelNet40 dataset. The experimental results verify the practical applicability of the HexagonNet given some fixed parameter budgets.

INDEX TERMS Hexagon, discrete grids, CNN.

I. INTRODUCTION

Hexagonal grids have some advantages (e.g. isoperimetry, additional equidistant neighbors, and uniform connectivity) over quadrangular and triangular girds [1], which have implications for processing information defined on them. Inspired by these advantages and the ubiquitous occurrence in nature (e.g. the visual systems of insect and human, honey-combs), hexagonal grids have been extensively used in the area of astrophysical and cosmological research [2], and visual systems [3], etc. As illustrated in Fig. 1, the past few years have witnessed the widespread application of hexagonal grids in digital image processing [1], [4], [5], geospatial information subdivision organization [6], combat environment modeling [7], new economic geography [8], and intelligent transportation system [9], etc.

Convolutional neural network (CNN) as an extensively-used representation learning method for Computer Vision and Big Data, and has attracted enormous attention in the past decade. Remarkable progresses have been achieved in the design of network architectures and convolution operations, in which increasing efforts have been made to generalize the CNN to multiview [11], non-Euclidean spaces [12] and smooth manifolds [13]. Also, CNN is capable of learning equivariant/invariant representations to common object/context transformations, some equivariant/invariant CNNs have been investigated by some researchers based on Group-based convolution operations [14]. Besides, combined with reinforcement learning, CNN as the feature representation learning part has shown great practicability in deep reinforcement learning.



(a) (b) (c)

FIGURE 1. Widespread applications of hexagonal grids: (a) Hexagon Image Processing [1], (b) Discrete Global Grids [6], (c) Cherenkov Telescope Array [10].

In this paper, we concentrate on designing CNNs for hexagonal grids and developing some novel frameworks to process the information defined on them. As for planar

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Xiao.

hexagonal representations, we rely on a specific indexing scheme for hexagonally sampling information, which allows for Cartesian tensors computation. As for spherical hexagonal representations, we use an icosahedron-based spherical grids as the initial representation of the spherical information. This efficient geodesic grid representation supports fast computation of the locally-supported spherical convolution operation. We start by introducing some applications based on the hexagonal grids. Then, we present the planer and spherical hexagonal grids and analyze the group representations for them, we review geometric deep learning, some new convolutions for hexagonal CNNs, and group-based equivariant convolution. Next in importance, HexagonNet, one novel CNN framework is defined to learn hexagonal representations. The hexagonal convolution kernel is decomposed into multiple rectangular sub-kernels, and the convolution operation is redesigned as a kernel expansion (KE) step followed by a planar hexagonal convolution (HexConv). Finally, in order to evaluate the feasibility of the proposed HexagonNet with increased degree of symmetry, we perform experiments on two tasks: aerial scene classification on the aerial image dataset (AID) [15], and 3D shape classification on the ModelNet40 dataset. The experimental results verify the practical implications of the HexagonNet.

The rest of the paper is organized as follows: Sec.II introduces the background and related work. Subsequently, in Sec.III, the main framework of HexagonNet is introduced and the detail implementation of hexagonal convolution are demonstrated. Sec.IV presents the experimental environment, settings, and results of aerial scene classification, and 3D shape classification using the HexagonNet. Finally, conclusion and future work are summarized in Sec.V.

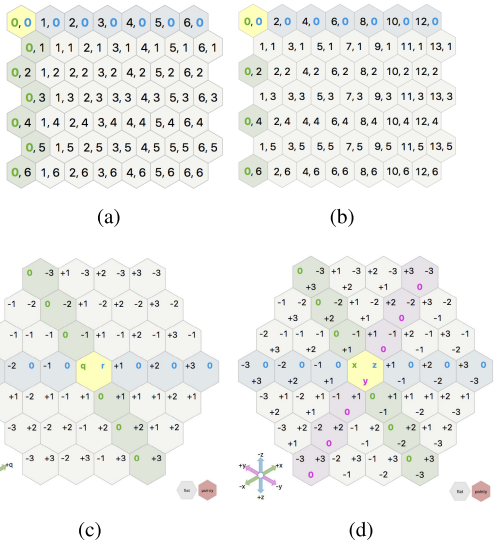


FIGURE 2. Coordinate systems [17]: (a) Offset, (b) Doubled, (c) Axial, (d) Cube.

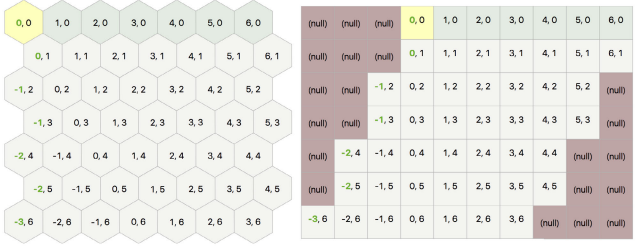


FIGURE 3. One typical storage scheme for hexagonal grids based on the Axial coordinate system.

II. BACKGROUND AND RELATED WORK

Hexagonal grids have gained extensive attention due to the advantages of reduced anisotropy and increased degree of symmetry.

A. HEXAGONAL GRIDS

The isoperimetry and uniform connectivity of hexagonal grids makes it a naturally excellent method to tile a plane [1]. Current research on hexagonal grids has made considerable progress in grid construction, encoding, indexing, storage, and related operations [16]. Many coordinate systems have been proposed for addressing hexagonal structure, such as 2-Axes "Offset, Doubled, Axial" coordinates and 3-Axes symmetrical "Cube" coordinate. The coordinate systems mentioned above are shown in Fig. 2.

Hexagonal grids can be indexed using the above coordinate systems. These systems differ by three important characteristics: memory efficiency when stored in a rectangular memory region, the availability of a mapping between convolution in the coordinate system and conventional 2D convolution, the possibility of reusing square convolution kernels for hexagonal convolution, and the ease of computing rotations and flips in the coordinate system [5].

Many indexing schemes have been proposed to process hierarchical data, such as GBT (Generalized Balanced Ternary) [18], ASA (Array Set Addressing) [19], HIP (Hexagonal Image Processing) [1], PYXIS [20], HQBS (Hexagonal Quadternary Balanced Structure) [21], and radix-based algebraic encoding scheme [16], etc. Many of these structures are based on group arithmetic with redefined add and multiply tables.

Hexagonal grids have many kinds of storage schemes. Usually, a rectangular area in the hexagonal grid corresponds to a parallelogram in memory. Fig. 3 shows one typical storage scheme for hexagonal grids based on the Axial coordinate system.

As shown in Fig. 4, three kinds of subdivision aperture proposed by previous researchers are mostly used to generate multiple resolution discrete grids, i.e. aperture 3, 4 and 7 with centroid-aligned and vertex-aligned variants.

As for spherical hexagonal grids, the traditional discrete grids include latitude and longitude grid, Yin-Yang grid [22], triangle grid (QTM) [23], diamond grid (HEALPix) [24], hexagon grid (PYXIS) [25], and Voronoi/TIN grid [26]. Geodesic polyhedron-based subdivision bound with projection into a sphere is the existing most-frequently employed

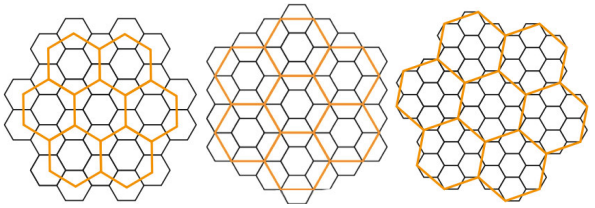


FIGURE 4. Centroid-aligned (left) and vertex-aligned (right) variants of hexagonal grids subdivision: (a) Aperture 3, (b) Aperture 4, (c) Aperture 7.

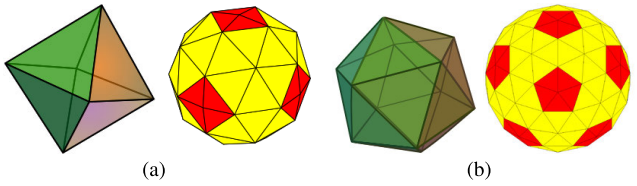


FIGURE 5. Geodesic polyhedron: (a) Octahedron (b) Icosahedron.

routine. As shown in Fig. 5, octahedron, and icosahedron are highly-adopted alternatives for spherical hexagonal grids.

B. GROUP REPRESENTATION

1) GROUP

A mathematical abstraction with compositional structure and mathematical operators, such as transformations. For group elements $g, h, k \in G$, it has four main properties:

- **closure**: chained transformations are transformations, e.g. $gh \in G$.
- **associativity**: $g(hk) = ghk = (gh)k$.
- **identity**: there exists a transformation $e \in G$ (sometimes written 0) such that $eg = ge = g, \forall g \in G$.
- **invertibility**: every transformation g has an inverse g^{-1}, so $gg^{-1} = g^{-1}g = e$, for instance, rotations and translations are both examples of groups.

2) PLANAR HEXAGONAL GRID

An infinite hexagonal grid discretizes the 2 dimensional uniform space, the symmetry of the finite $n \times n$ hexagonal girds can be described by the group composed of the dihedral group D_6 expressing local regular-hexagonal symmetry and the group $\mathbb{Z}_2 \times \mathbb{Z}_2$ (direct product of two cyclic groups of order n) expressing translational symmetry in two directions. In this paper, the oblique basis vectors are adopted to represent the grid:

$$\ell_1 = d\begin{bmatrix}1\\0\end{bmatrix}, \quad \ell_2 = d\begin{bmatrix}-1/2\\ \sqrt{3}/2\end{bmatrix}. \tag{1}$$

where $d > 0$ means the vector length. The infinite and finite hexagonal grids are expressed as:

$$\mathcal{H} = \{n_1\ell_1 + n_2\ell_2 | n_1, n_2 \in \mathbb{Z}\}, \tag{2a}$$
$$\mathcal{H}_n = \{n_1\ell_1 + n_2\ell_2 | n_1, n_2 \in \mathbb{Z}, 0 \le n_1, n_2 \le n-1\}. \tag{2b}$$

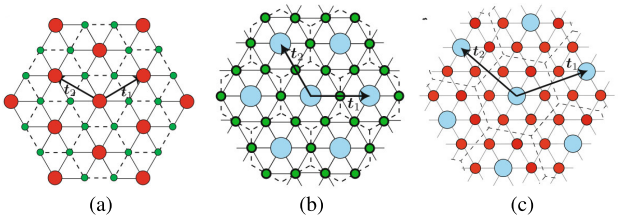


FIGURE 6. The finite hexagonal grids and different upsampled grids with aperture 3, 4, or 7.

To represent different aperture of the hexagonal grids $\mathcal{H}$, we consider the upsampled hexagonal grids spanned by:

$$t_1 = \alpha\ell_1 + \beta\ell_2, \quad t_2 = -\beta\ell_1 + (\alpha - \beta)\ell_2. \tag{3}$$

where α and β are integer parameters with $(\alpha, \beta) \neq (0, 0)$. The upsampled hexagonal gird $\mathcal{H}(\alpha, \beta)$ can be expressed as:

$$\begin{aligned}
&\mathcal{H}(\alpha, \beta)\\
&= \{n_1t_1 + n_2t_2 | n_1, n_2 \in \mathbb{Z}\}\\
&= \{(n_1\alpha - n_2\beta)\,\ell_1 + (n_1\beta + n_2(\alpha - \beta))\,\ell_2 | n_1, n_2 \in \mathbb{Z}\}\\
&= \left\{\begin{bmatrix}\ell_1 & \ell_2\end{bmatrix}\begin{bmatrix}\alpha & -\beta\\ \beta & \alpha-\beta\end{bmatrix}\begin{bmatrix}n_1\\n_2\end{bmatrix} | n_1, n_2 \in \mathbb{Z}\right\}
\end{aligned} \tag{4}$$

The symmetry of the finite hexagonal grid $\mathcal{H}_n$ is characterized by invariance:

- r : counterclockwise rotation around the origin at an angle of $\pi/3$,
- s : reflection $y \mapsto -y$,
- t_1: preiodic translation along the l_1 axis, and
- t_2: preiodic translation along the l_2 axis.

Fig. 6 portrays the finite hexagonal grids and the upsampled grids with different apertures.

Consequently, the symmetry of the hexagonal grid H_n can be described by the group:

$$G = \langle r, s, t_1, t_2 \rangle, \tag{5}$$

where G contains the dihedral group $\langle r, s \rangle \simeq \mathcal{D}_6$, cyclic groups $\langle t_1 \rangle \simeq \mathbb{Z}_n$, $\langle t_2 \rangle \simeq \mathbb{Z}_n$, and G has the semi-direct product structure of D_6 by $\mathbb{Z}_n \times \mathbb{Z}_n$, which means, $G = D_6 \ltimes (\mathbb{Z}_n \times \mathbb{Z}_n)$.

Here, the dihedral group D_6 refers to the dihedral symmetry, which contains six rotations and one flip, the rotation cyclic symmetry can be represented as one cyclic group C_6.

3) SPHERICAL HEXAGONAL GRIDS

A spherical hexagonal grid can be achieved by projecting the sphere into five platonic polyhedra (tetrahedron, cube, octahedron, dodecahedron, and icosahedron). Usually, an icosahedron-based spherical hexagonal grid is adopted as the discrete representations of some spherical signals. As shown in Fig. 7, the icosahedron is a regular solid with 20 faces, 30 edges, and 12 vertices. Icosahedral-based spherical meshes are the most uniform and accurate discretizations of the sphere manifold [27], in which we can progressively

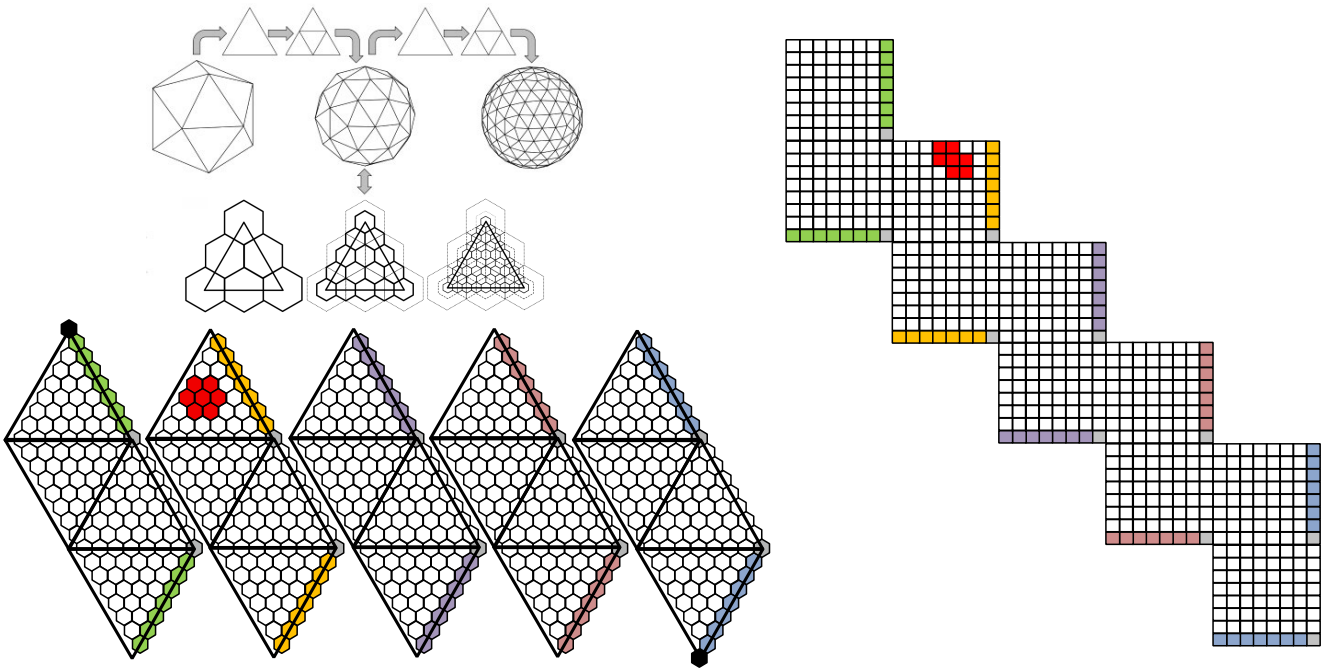


FIGURE 7. The icosahedron-based spherical hexagonal grid $\mathcal{H}_3$, which is covered by five charts $\{C_i\}_{i=1}^5$. The original spherical hexagonal grids is upfolded onto a 2D plane with five charts, each chart is covered with 4 triangles. The colored hexagonal grid are shared by the neighbor chart. For shared hexagonal grids $p \in S_i \cap S_j'$, the transition can be depicted by a gauge transformation between charts. By rotating the skew axes into orthogonal axes, the flattened icosahedron can be stretched into a rectilinear grid structure represented by five 2D matrices with shape $2^{i+1} \times 2^i$.

subdivide each face of the unit icosahedron into hexagonal grids and re-project the mesh vertex to the unit sphere.

The subdivision scheme provides a natural multi-resolution (coarsening and refinement) representation way for the hexagonal grids, which allows for easy implementations of pooling and upsampling routines. Here we adopt aperture 4 for subdivision and begin with a spherical hexagonal grid $\mathcal{H}_0$ consisting of the corners of the unit icosahedron itself. Each progressive spherical hexagonal grid $\mathcal{H}_{i+1}$ is one level above the previous $\mathcal{H}_i$. Hence, for a level-i spherical hexagonal grid, the numbers of vertices, edges, and faces (hexagonal or pentagonal) are as follows:

$$n_v = 10 \cdot 2^{2i+1} \tag{6a}$$
$$n_e = 15 \cdot 2^{2i+1} \tag{6b}$$
$$n_f = n_e - n_v + 2 = 5 \cdot 2^{2i+1} + 2 \tag{6c}$$

The information reorganization procedure of spherical hexagonal grids produces a hierarchical subdivision of a spherical surface where the central point (pixel) of each grid covers the same surface area as every other pixel. For each non-corner hexagonal pixel, it has 6 neighbors, but each corner pentagonal pixel has 5 neighbors.

We can employ $\mathcal{I}$ (a non-abelian subgroup of the rotation group $SO(3)$, which is isomorphic to the alternating group A5) to denote the set of global orientation-preserving symmetries (60 rotations). The symmetries of the $\mathcal{I}$ can be divided into sets of rotations around the axes (5 rotations around each axis passing through vertices or 3 rotations around each axis passing through its face centers). In addition, the 2D Cartesian coordinate system is employed to store the icosahedron-based spherical grid as one regular data structure. The icosahedral group $\mathcal{I}$ is the group of symmetries on the icosahedron, which consists of 60 rotations, the Cayley Table is shown in Fig. 8, we can see that the group $\mathcal{I}$ is non-abelian since the table is not symmetric.

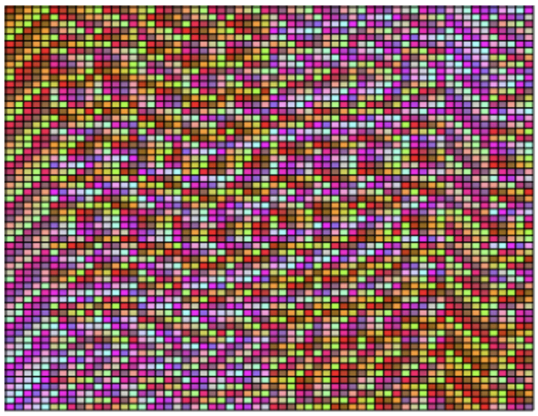


FIGURE 8. Cayley table for the icosahedral group $\mathcal{I}$.

In this paper, the icosahedron is unfolded with five charts $\{C_i\}_{i=1}^5$, each chart is covered with 4 triangles. Each chart is an invertible map $\varphi_i : S_i \to V_i$, where $S_i \subset \mathcal{H}_i$ and $V_i \subset \mathbb{Z}^2$. For shared hexagonal grids $p \in S_i \cap S_i'$, the transition can be depicted by a gauge transformation between charts. By

rotating the skew axes into orthogonal axes, we define the maps $\varphi_i(c_i) = x_i$ between the corner points c_i in $\mathcal{H}_i$ and points x_i in the planar grid $\mathbb{Z}^2$. This transformation has two main advantages: within each chart, shifting kernels on the sphere is approximately equivariant to the translation in the skew axes; features on the planar hexagonal grids can be naturally expressed using tensors. The flattened icosahedron can be stretched into a rectilinear grid structure represented by 2D matrices with shape $2^{i+1} \times 5 \cdot 2^i$.

C. GEOMETRIC DEEP LEARNING

Many network architectures and convolution operators have been proposed for sets [28], homogeneous spaces [29] and other non-Euclidean manifolds [30]. Geometric representation-based CNNs (e.g. Geometric CNN (gCNN) [31], ShapeNet [32], SplatNet [33], VoxNet [34], OctTree [35], PointNet [36] and PointNet++ [37]) provide information processing defined on regular mesh, grid, voxel, or raw irregular point cloud. View-based CNNs (e.g. Multi-View [38], GVCNN [11], DeepPano [39], EMVN [40], Omnidirectional CNN [41]) represent the scenes by sets of views or rendered panoramic views, these novel approaches aggregate views ''outside-in'' for 3D shapes and ''inside-out'' for panoramic scene understanding. Spherical CNNs (e.g. Spherenet [42], SphConv [43], Spherical CNN [44], Clebsch-Gordan Net [45], Kernel Transformer Network [46], DeepSphere [24], S^2CNN [47], a^3S-CNN [44], Spherical U-Net [48]) focus on processing the data defined on the sphere. Group-based CNNs (e.g. CubeNet [49], HEXACONV [5]) utilize symmetrical equivariance and invariance, which enable them to considerably outperform other methods.

As for planar representations, the property of non-alignment with Cartesian coordinates drives the increasing efforts to design convolution operations for the non-Cartesian data defined on them. By exploiting groups of symmetries and replacing convolutions with groups convolution, in which a filter transformation (FT) step is followed by a planar hexagonal convolution, HEXACONV shows better accuracy than square filters [5]. Combined with a special indexing scheme, HexagDLy facilitates access to CNN [50]. H-CNN devises a framework which combines the local prediction with hexagon-based ensemble mechanism [9]. IndexedConv [51] presents indexed convolution and pooling for the indexed hexagonal grids.

As for spherical representations, most recent researchers adopted icosahedron subdivision based representation for spherical information. The icosahedron is one regular convex polyhedron with 20 triangular faces, which can be progressively subdivided. Triangle is the dual graph of hexagon, some researchers adopted triangle for convolution design. UGSCNN [52] implements mesh based convolution with a reparameterized differential operator on unstructured grids. SpherePHD [53] defines convolution on the triangle faces subdivided by one icosahedron. HexUNet [54] introduces fast interpolation for orientation-aware convolutions on the sphere. a^3S-CNN [44] proposes an azimuth-zenith anisotropic CNN for 3D shape classification and retrieval. EMVN [40] takes advantage of the finite nature of multiple views and the finite rotation groups like the icosahedron to design equivariant multiview network. Gauge CNN [55] designs one gauge equivariant CNN by enforcing weight sharing across multiple orientations.

D. GROUP EQUIVARIANT CONVOLUTION

Equivariance to symmetry transformations drives the equivariant networks to show great performance and data efficiency on medical image segmentation, 3D shape classification, and climate pattern prediction. Recent research has shown that CNN can exploit other sources of invariance, such as rotation invariance, by using group convolutions instead of planar convolutions. For the sake of performance and ease of implementation, it is necessary to limit the group convolution with transformations. There are four kinds of equiviriant networks: adopting new filter representations [56], changing to equivariant coordinate [40], employing the group based equivariant filter [49], and using gauge equivariant transformation [55]. Previous research has shown that convolutional networks can be made equivariant by exploiting richer groups of symmetries [57]–[59]. For a hexagonal tiling of the plane with 6-fold rotational symmetry, only integer translations, rotations by multiples of 60 degrees, and reflections are admissible.

1) EQUIVARIANCE

Given a space $\mathcal{X}$ and a transformation group G. Consider a function map $\Phi : \mathcal{X} \to \mathcal{Y}$. For any $g \in G$, we can define the group action applied to the space, $\mathcal{L}_g^{\mathcal{X}} : \mathcal{X} \to \mathcal{X}$, which has property of homomorphism, $\mathcal{L}_g^{\mathcal{X}} \mathcal{L}_h^{\mathcal{X}} = \mathcal{L}_{gh}^{\mathcal{X}}$. So the function Φ is said to be equivariant to G if

$$\Phi(\mathcal{L}_g^{\mathcal{X}}(x)) = \mathcal{L}_g^{\mathcal{Y}}(\Phi(x)), \quad \forall x \in \mathcal{X},\ g \in G. \tag{7}$$

where $\mathcal{L}^{\mathcal{X}}$ denotes a group action in the corresponding space, $\mathcal{X}$ and $\mathcal{Y}$ are the spaces of input and feature representations respectively. If $\mathcal{L}^{\mathcal{Y}} = id$, then equivariance is invariance and Φ is invariant to G without preserving information about g. In this paper, such degenerate case is not considered.

2) GROUP CONVOLUTION

As for planar convolution between $f, h : \mathbb{R}^2 \mapsto \mathbb{R}$, the main operation of CNN can be seen as an operation over the group of translations on the plane, where the group action is an addition of coordinate values expressed as:

$$[f \star h](x) = \int_{y \in \mathbb{R}^2} f(y) h(x - y)\, dx. \tag{8}$$

Planar convolution operation is equivariant to translation. The operation can be generalized to any group G and $f, h : G \mapsto \mathbb{R}$, which is equivariant to group actions from G, as expressed by:

$$[f \star h](x) = \int_{g \in G} f(x) h(g^{-1} x)\, dg, \tag{9}$$

By changing the translation to a transformation from a larger group G, given the input space X (e.g. the square or hexagonal grids), a group convolution is defined by:

$$[f \star_g h](g) = \sum_{x \in X} \sum_{k} f_k(x) \psi_k(g^{-1}x), \tag{10}$$

where k denotes the input channel, f_k and ψ_k are the signals defined on X, and g is a transformation in G.

So, given the translate action $\mathcal{L}_t$ that translates a feature map $f : \mathbb{Z}^2 \to \mathbb{R}^K$ by $t \in \mathbb{Z}^2$, and given a kernel h. Translation equivariance is then expressed as a translation followed by convolution equals convolution followed by a translation:

$$[[\mathcal{L}_t f] \star h](x) = [\mathcal{L}_t [f \star h]](x). \tag{11}$$

However, convolution is not rotation equivariant. If we instead consider the feature map $f : G \to G$, the group convolution is equivariant to transformations.

$$[[\mathcal{L}_r f] \star h](x) = [\mathcal{L}_r [f \star [\mathcal{L}_{r^{-1}} h]]](x). \tag{12}$$

$$[[\mathcal{L}_t f] \star_g h](x) = [\mathcal{L}_t [f \star_g h]](x). \tag{13}$$

III. HEXAGONNET

In order to use hexagonal grids in convolutional neural networks, a number of challenges must be addressed. Firstly, resampling is performed on hexagonal grids, as for planar quadrangular, triangular or non-regular girds, that can be easily achieved using bilinear interpolation. Secondly, in order to improve computational efficiency, there is a need to employ efficient storage, indexing schemes that assist a practical implementation of hexagonal convolution. In fact, we can leverage highly-optimized square convolution routines to design hexagonal convolution. In this paper, we mainly focus on the image classification task on real datasets, where there are usually three steps for preparation:

- Hexagonal reorganization: as for planar hexagonal grids, we can resample the square grids-based images by using bilinear interpolation, as for spherical hexagonal grids, we can project the spherical images to the icosahedron-based grids;
- Choosing of coordinate system, indexing and storage scheme: choose the memory efficient indexing and storage scheme for image operations;
- Convolution operations design: convolution kernel decomposition and square convolution routines design for fast hexagonal convolution implementation.

In addition, one typical CNN is composed of four main building blocks, i.e. (1) a convolution operation, (2) a nonlinear or optional activation function, (3) a pooling operation, and (4) a normalization. Among the four building blocks, the convolution and pooling operations need to be designed. Many methods define convolutional operations on the Fourier domain with point-wise operations, spherical harmonics or whole spherical domain, whereas spatially-localized or locally-supported kernels show more computational efficiency and allow for the learning of hierarchical features as stacked layers. In this paper, we define the HexagonNet with locally-supported kernels on groups.

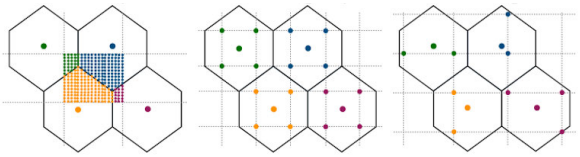


FIGURE 9. The planar hexagonal resampling schemes from rectangular grid: (a)Re-bining, (b)Over-sampling, (c)Interpolation.

A. HEXAGONAL REORGANIZATION

From the perspective of information organization, data can be hexagonal organized through hexagonal quantifying or resampling. Such as the Cherenkov Telescope Array, the hexagonal collected data are quantified for every hexagon. But for non-hexagonal organized data, in order to employ the hexagonal girds, we usually need to resample the dataset for hexagonal reorganization with bilinear interpolation.

As shown in Fig.9, there are many resampling schemes, such as re-bining, over-sampling, and interpolation. In this paper, as for planar hexagonal grids resample, we chose the interpolation scheme with half pixel shift [60]. Compared to a rectangular grid, the one line of two in a hexagonal grid is shifted by half a pixel. So, we only need to interpolate the hexagonal grid with an average of two consecutive horizontal square grids.

As for icosahedron-based spherical hexagonal grids, spherical parameterization [61] and spherical projection [62] are two different methods widely employed as one preporcessing step to generate spherical representation. Such as for 3D objects, we can generate representations of level-r spherical hexagonal grids, which are neither too redundant (voxel) nor too sparse (multiview). Spherical parameterization, such as area-preserving mapping using optimal mass transport and angle-preserving map using conformal map, can be used to generate some intrinsic shape descriptors [63]: principal curvatures (PC), average geodesic distance (AGD), Laplace-Beltrami operator (LBO), and heat kernel signature (HKS). But, this method only works for genus-0 closed object, in which the high genus of many mesh-based objects is the main hurdle needs repairing. Spherical projection by ray-mesh intersection can be used to project the 3D object onto an enclosing sphere using ray casting for every grid from the center. Two extrinsic spherical descriptors are often used [44], [63]: spherical extent function (SEF), normal deviation function (NDF).

B. HEXAGONAL CONVOLUTION

Equivariant networks have shown excellent performance and data efficiency in problems that exhibit symmetries. The main hurdle of a practical implementation of equivariant

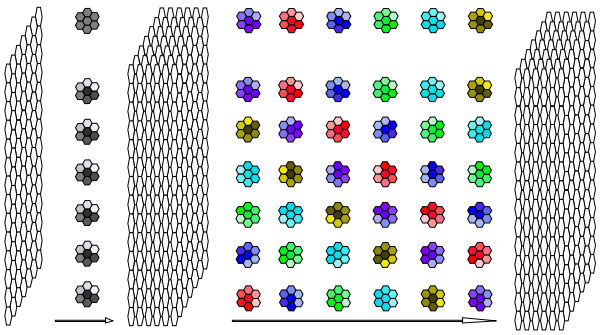


FIGURE 10. Group C_6 based kernel expansion (KE) step for weight sharing.

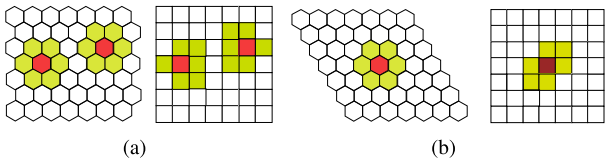


FIGURE 11. Hexagonal convolution in the two relevant coordinate system:(a)Offset, (b)Axial. The memory matrix in Cartesian coordinate (left), the hexagonal grids (right). Standard 2D convolution uses both feature map and kernel stored according to the coordinate system.

convolution is the requirement of interpolation. As for hexagonal grids, we can transform the skew coordinate to Cartesian coordinate and redesign the convolution operations. It is possible to rotate a kernel by any multiple of 60 degrees without interpolation, which allows us to define group convolutions that contains integer translation, rotation with multiples of 60 degrees, and flip.

Group based hexagonal convolution (HexConv) is defined on the group representations of the hexagonal grids. As for group $\mathcal{G}$, the group convolution can implemented as one kernel expansion in $\mathcal{G}$ followed by one planar convolution in $\mathbb{Z}^2$. HexConv generally should tackle two kinds of group convolutions: one for planar convolution to group represented feature layer and one for group represented feature layers. The unified formation can be illustrated by introducing $\mathcal{G}_{in}$ and $\mathcal{G}_{out}$. Usually for the first convolution layer, $\mathcal{G}_{in} = \{e\}$ is the trivial group containing only the identity expansion of kernel, while $\mathcal{G}_{out} = \mathcal{G}$ is typically a group of discrete rotations. As for higher convolution layers, we have $\mathcal{G}_{in} = \mathcal{G}_{out} = \mathcal{G}$.

As shown in Fig. 10, group C_6 based kernel expansion step for weight sharing. The kernels $K : \mathbb{R}^2 \to \mathbb{R}^{R_{\text{in}}\mathcal{G}_{\text{in}} \times R_{\text{out}}\mathcal{G}_{\text{out}}}$ are stored in an array of shape $(R_{\text{in}}\mathcal{G}_{\text{in}}, R_{\text{out}}\mathcal{G}_{\text{out}})$, with a 1-ring hexagonal kernel. The complete convolution operation can be formulated based on hexagonal reorganization (HR) and kernel expansion (KE) as follows:

$$\text{HexConv(f, k)=Conv2D (HR(f), KE(k))} \tag{14}$$

where f and HR(f) both have shape $(\mathcal{G}_{\text{in}}R_{\text{in}}, B)$, the kernel k have shape $(\mathcal{G}_{\text{in}}R_{\text{in}}, \mathcal{G}_{\text{out}})$, and the expanded kernel has shape $(\mathcal{G}_{\text{in}}R_{\text{in}}, \mathcal{G}_{\text{out}}R_{\text{out}})$. The output of HexConv has shape $(\mathcal{G}_{\text{out}}R_{\text{out}}, B)$.

Many CNNs employ decomposed convolution operators [50], [64]. Fig. 11 shows the hexagonal convolution kernel and the memory matrix in Cartesian coordinate.

The natural shape of the locally-supported kernel related to the hexagonal grid is one kernel with different ring sizes (As for spherical hexagonal grid, the locally-supported kernel corresponds to the local cap region). To improve the computational efficiency, there is a need to design an efficient data structure which is self-sufficient to support convolution,

Algorithm 1 Hexagonal Convolution (HexConv)

if *planar hexagonal grids* **then**
- $\{H_i^j\}_{j=1}^3 \leftarrow \text{Padding}(H_i)$
- $\mathbf{W}_1 \leftarrow \begin{bmatrix} w_3 & w_4 & w_5 \end{bmatrix}$, $\mathbf{W}_2 \leftarrow \begin{bmatrix} w_1 & w_2 \\ 0 & 0 \\ w_6 & w_7 \end{bmatrix}$
- **for** $j \leftarrow \{1, \ldots, 3\}$ **do**
  - $H_i^1 \leftarrow \text{Conv2D}(H_i^1, \mathbf{W}_1)$
  - $H_i^2 \leftarrow \text{Conv2D}(H_i^2, \mathbf{W}_2)$;
  - $H_i^3 \leftarrow \text{Conv2D}(H_i^3, \mathbf{W}_2)$;
  - $H_i \leftarrow H_i^1 \oplus H_i^2 \oplus H_i^3$
- **end**
- **Return:** H_i

if *spherical hexagonal grids* **then**
- $H_i = \{C_i\}_{i=1}^5 \leftarrow \text{Padding}(H_i = \{C_i\}_{i=1}^5)$
- $\mathbf{W_g} \leftarrow \begin{bmatrix} w_1 & w_2 & 0 \\ w_3 & w_4 & w_5 \\ 0 & w_6 & w_7 \end{bmatrix}, \quad g \in C_6, D_6$
- **for** $i \leftarrow \{1, \ldots, 5\}$ **do**
  - **for** $j \leftarrow \{1, \ldots, |G|\}$ **do**
    - $H_i^j \leftarrow \text{Conv2D}(C_i, \mathbf{W_g})$
    - $H_i \leftarrow \{H_i^1, \ldots, H_i^{|G|}\}$
  - **end**
- **end**
- **Return:** $\{H_i\}_{i=1}^5$

pooling, and other CNN operators. Efficient planar hexagonal convolution) is tightly related with the hexagonal kernel. In this paper, the hexagonal kernel is decomposed into multiple rectangular sub-kernels. So, the hexagonal convolution operation can be performed by a combination of multiple square convolution routines of the input tensor with sub-kernels. The decomposition of ring size 1 kernel and the convolution operator are illustrated in Fig. 12. For every square convolution operation with one sub-kernel, different paddings and strides are applied, and the non-violating symmetry convolution operator is realized by combining the decomposed sub-convolutions.

Our hexagonal convolution operation is given in Alg. 1, in which the planar hexagonal convolution can be computed with sub-kernels, the spherical hexagonal convolution can be

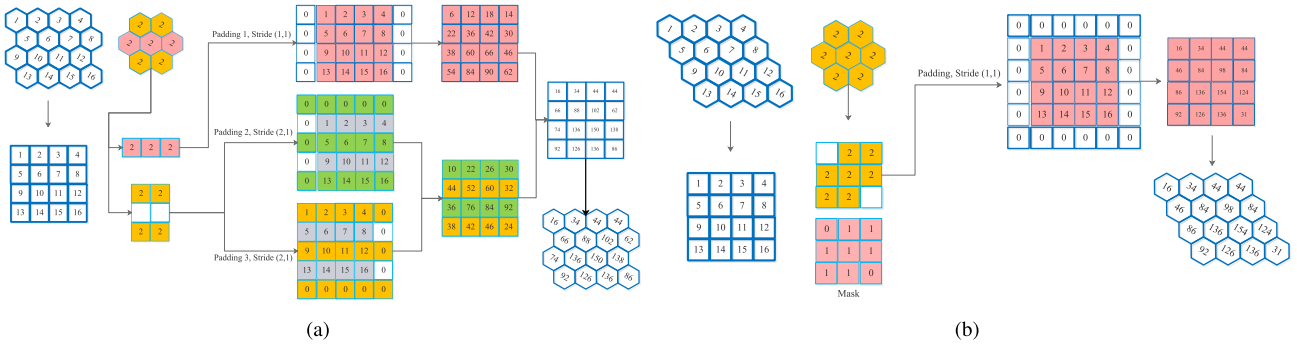


FIGURE 12. The hexagonal convolution operation (HexConv) with a kernel of ring size 1: (a) planar hexagonal grid, the kernel is divided into two sub-kernels, different paddings and strides are applied to the input tensor for every sub-kernel, the resulting sub-convolutions are then merged and added to receive the hexagonal convolved tensor; (b)spherical hexagonal grids, the kernel with one mask operator that zeros the top-right and bottom-left.

Algorithm 2 Padding With Cross Chart

Input: $\{C_i\}_{i=1}^{5}$ with width W and hight $2W$.

for $i \leftarrow \{0, \ldots, 4\}$ **do** // pad each chart

$C_l \leftarrow C_{\mathrm{mod}(i-1,5)}$; // left chart

$C_r \leftarrow C_{\mathrm{mod}(i+1,5)}$; // right chart

$$T \leftarrow \left[C_l(W, W) \text{ to } C_l(1, W)\ 0 \right];$$

$$L \leftarrow \begin{bmatrix} \left[C_l(W+1, W) \text{ to } C_l(2W, W) \right]^{\mathrm{T}} \\ \left[C_l(2W, W-1) \text{ to } C_l(2W, 1) \right]^{\mathrm{T}} \\ 0 \end{bmatrix};$$

$$B \leftarrow \left[0\ C_r(2W, 1) \text{ to } C_r(W+1, 1) \right];$$

$$R \leftarrow \begin{bmatrix} \left[C_r(1, W) \text{ to } C_r(1, 1) \right]^{\mathrm{T}} \\ \left[C_r(1, 1) \text{ to } C_r(W+1, 1) \right]^{\mathrm{T}} \end{bmatrix};$$

$$P_i \leftarrow \begin{bmatrix} T \\ \left[L\ C_i \right] \end{bmatrix}; \quad \text{// top and left}$$

$$C_i \leftarrow \left[\begin{bmatrix} P_i \\ B \end{bmatrix} R \right]; \quad \text{// bottom and right}$$

end

Return: $\{C_i\}_{i=1}^{5}$ with shape $(2^{i+1}+2) \times (2^i+2)$.

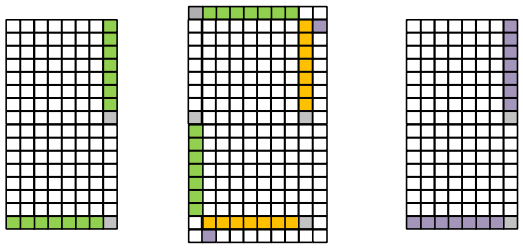


FIGURE 13. The padding operation with cross-neighbor.

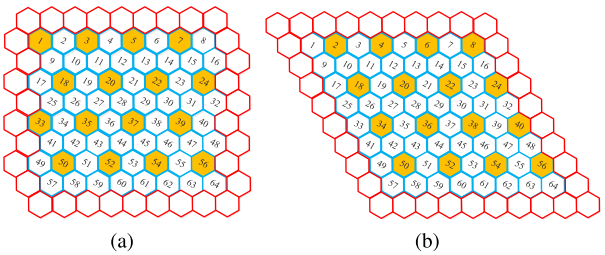


FIGURE 14. The hexagonal pooling (HexPool) operation with a kernel of ring size 1 and a stride of 2: (a)planar hexagonal grids, (b)spherical hexagonal grids.

computed with one mask step to zero the right-top and left-bottom of the kernel.

As shown in Fig. 7, to prepare for convolution operation, we need to pad every chart with shared hexagonal grids. In this paper, we use convolution kernel with ring size 1, as described in Fig. 13, the details of cross-neighbor padding operation are illustrated in Alg. 2, where we pad all boundaries with cross charts.

C. HEXAGONAL POOLING

The function of HexPool (hexagonal pooling) operator is to progressively reduce the spatial size and reduce the number of parameters. The most common form of a square pooling layer applies a kernel size of 2×2 with a stride of 2 to downsample the input. Compared to max pooling or even L2-norm pooling, average pooling has shown to be permutation invariant. The HexPool operator with a kernel of size 1 and a stride 2 for hexagonal grids is performed as illustrated in Fig. 14.

IV. EXPERIMENTAL EVALUATION

This section provides a concise and precise description of the experimental results on some datasets. In order to validate our implementation, the accuracy and parameters of our HexagonNet had been evaluated with state-of-the-art methods on the different datasets. The label function of each image is invariant to rotations, that is the labels dose not change when the image is rotated. We focus on classification tasks.

A. EXPERIMENTS SETUP

All the experiments are executed on one Alienware that runs Ubuntu 16.04 with 4 CPU cores, NVIDIA GeForce

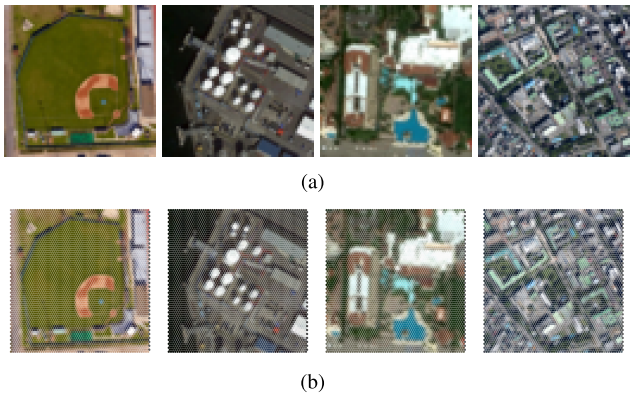


FIGURE 15. Samples of the hexagonal resampled AID dataset: (a)original example images, (b)hexagonal sampled images.

GTX1080 GPUs with 16GB of RAM, CUDA 8.0, and cuDNN 5.0. Our algorithms are implemented in Python.

B. AERIAL SCENE CLASSIFICATION

For sensory perception tasks, deep neural networks have mostly replaced handcrafted features. The evaluation of our HexagonNet was performed on the AID dataset, and the details are as follows:

1) AID

The AID (Aerial Image Dataset) [15] is a dataset consisting of 10000 satellite images, each with pixels of 400 × 400, and 30 classes. These images are collected from the Google Earth imagery under varying conditions (time and seasons) and cover different countries. After resampling each images to 64 × 64 pixels, the dataset was transformed to a hexagonal dataset. All the images are standardized and whitened using a PCA. Fig. 15 shows some original and resampled samples of the AID dataset. The task is aerial scene classification.

2) CLASSIFICATION

In deep neural networks, convolution layers are often associated with pooling layers.

a: ARCHITECTURE

The Resnet [65] architecture of the HexagonNet that we use in this paper consists of 1 convolution layer, 3 ResBlocks, 1 pooling layer between ResBlocks, 1 fully-connected layer and 1 softmax layer. Each ResBlock has 2 residual blocks, and 1 skip connection from the input to the output. The pooling operation between 2 stacks was achieved with a stride of 2. Some parameters of the architectures are shown in Tab. 1.

TABLE 1. HexagonNet-C_6 architecture, where *a*, *b*, *c* stands for input channels, bottleneck channels, and output channels. *s* stands for strides.

Stage	a	Block	b	c	s
–	1	HexConv	–	16	1
1	16	ResBlock	64	64	2
2	64	ResBlock	256	256	2
3	256	ResBlock	1024	1024	2
–	1024	MaxPool	–	–	1
–	1024	MLP	–	30	1

TABLE 2. The comparision of the networks evaluated on the AID dataset.

Model	Accuracy (%)	Parameters
VGG-VD-16 [15]	89.6	-
BPECNN [66]	95.7	-
TSDFCNN [67]	94.6	-
Resnet-Our	89.3	91.4k
HEXACONV [10]	91.3	91.6k
IndexedConv [51]	79.8	72.8k
HexagonNet-C_6	91.5	70.1k
HexagonNet-D_6	**92.2**	70.2k

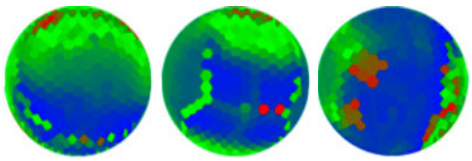


FIGURE 16. Some samples of the spherical hexagonal projections from ModelNet40 dataset.

b: RESULTS AND ANALYSIS

The baseline models we choose include normal VGG-VD-16 [15], binary patterns encoded CNN [66], two-stream deep fusion CNN [67], and hexagon based HEXACONV [10], IndexedConv [51]. Our experiments are conducted with 80%/20% splits to obtain better estimates and fair comparison.

As shown in Tab. 2, the BPECNN with mid-level features and TSDFCNN with saliency detection show higher accuracy improvement, which can be attributed to the fusion of high-level feature information, but our HexagonNet outperforms these hexagon based methods which mainly benefit from weight sharing across different kernel. Last, the Resnet-Our with same architecture without original images resampling is outperformed by our HexagonNet, which can be attribute to owing the increase of symmetries

C. 3D SHAPE CLASSIFICATION

For 3D shape classification, the performance is dependent on the input representation. Recently, by rendering multiview of the 3D input, we can leverage the image oriented CNNs to 3D shape classification. Here, we evaluate our HexagonNet on the ModelNet40 dataset.

1) ModelNet40

The ModelNet40 contains 12,311 non-aligned CAD objects of 40 categories ranging from airplane to xbox, where the orientations are not aligned. Fig. 16 shows some samples of the ModelNet40 dataset.

2) CLASSIFICATION

We adopt spherical projection to generate spherical descriptors for 3D shapes as many 3D objects are not genus-0.

TABLE 3. Two hexagonNet architecture used in 3D shapes classification experiments. *a*, *b*, *c* stands for input channels, bottleneck channels, and output channels. *s* stands for strides.

Level	a	Block	b	c	s
5	4	HexConv	–	8	1
4	8	ResBlock	8	16	2
3	16	ResBlock	16	64	2
2	64	ResBlock	64	256	2
1	256	MaxPool	–	–	1
–	256	MLP	–	40	1

TABLE 4. The accuracy of the networks evaluated on the ModelNet40 dataset.

Model	Input	Views	Accuracy (%)
DeepPano [39]	panoramic view	1	77.6
Spherical CNN [44]	spherical views	1	88.9
UGSCNN [52]	unstructured grids	20+	90.5
a^3S-CNN [63]	spherical views	20+	89.6
HexagonNet	projected views	20+	**91.4**

we augment the channels with re-generating spherical descriptors on the convex hull of the input 3D shapes.

a: SPHERICAL DESCRIPTORS

We employ the level-5 spherical hexagonal grids to encapsulate the 3D models. Let d be the distance from the center to the last point of intersection for a ray at direction $\{\phi_j, \psi_j\}$. We may employ two spherical descriptors: SEF, one spherical function that represents the maximal distance d from the object's center as a function of the spherical angle, $f(\phi_j, \psi_j) = d$; NDF, one spherical function that represents the sine angle between surface normal at the intersecting face and the ray direction, $f(\phi_j, \psi_j) = sin\alpha$.

b: ARCHITECTURE

The architecture contains the same network structure as we use in AID, We use four channels of the spherical descriptors as the input. Some parameters of the architectures are shown in Tab. 3.

c: RESULTS AND ANALYSIS

Here, we use the default training and testing split (9843 models for training and 2468 models for testing). The baseline models we choose include DeepPano [39], Spherical CNN [44], UGSCNN [52], and a^3S-CNN [63], which are mainly view-based paradigms.

As shown in Tab. 4, our proposed HexagonNet compares favorably with some view-based models, and achieves relative well performance.

V. CONCLUSION AND FUTURE WORK

This paper presents the HexagonNet for hexagonal grids by leveraging highly-optimized square convolution routines. In order to provide a native solution to process hexagonal grids sampled data, one indexing scheme is employed for the storage structure, which allows 6-fold rotations without the need for interpolation and facilitates the access to CNN for any hexagonally-sampled data. The effectiveness of our HexagonNet is evaluated through tasks of classification on the AID dataset and the ModelNet40 dataset, the experimental results verify the applicability.

Although rotation equivariance is essential in some applications, orientation information plays a crucial role in the prediction capability, such as distinguishing digits "6" and "9". That is the orientation information from cameras attached to vehicles or drones are very important cues that can be exploited in object detection, semantic segmentation and etc.

In fact, locally-supported and orientation-preserving convolution operators are sometimes indispensable for many vision tasks, but many equivariant CNNs still have difficulties in distinguishing. In the future, more attention will be paid to designing orientation-preserving CNN and extending the HexagonNet to some other potential applications with different task architectures, such as medical image segmentation, taxi supply-demand gap forecasting, astronomical telescope information process, and wargame situation awareness.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers who gave me plenty valuable advices to improve the paper.

REFERENCES

[1] L. Middleton and J. Sivaswamy, *Hexagonal Image Processing: A Practical Approach*. Berlin, Germany: Springer, 2006.

[2] I. Shilon, M. Kraus, M. Büchele, K. Egberts, T. Fischer, T. L. Holch, T. Lohse, U. Schwanke, C. Steppa, and S. Funk, "Application of deep learning methods to analysis of imaging atmospheric Cherenkov telescopes data," *Astropart. Phys.*, vol. 105, pp. 44–53, Feb. 2019.

[3] F. D. Tschopp, M. B. Reiser, and S. C. Turaga, "A connectome based hexagonal lattice convolutional network model of the drosophila visual system," 2018, *arXiv:1806.04793*. [Online]. Available: https://arxiv.org/abs/1806.04793

[4] B. Gardiner, S. Coleman, and B. Scotney, "Fast multiscale operator development for hexagonal images," in *Proc. Joint Pattern Recognit. Symp.* Berlin, Germany: Springer, 2009, pp. 282–291.

[5] E. Hoogeboom, J. W. T. Peters, T. S. Cohen, and M. Welling, "Hexaconv," Mar. 2018, *arXiv:1803.02108*. [Online]. Available: https://arxiv.org/abs/1803.02108

[6] K. Sahr, D. White, and A. J. Kimerling, "Geodesic discrete global grid systems," *Cartography Geograph. Inf. Sci.*, vol. 30, no. 2, pp. 121–134, 2003.

[7] R. Ellen, W. Bernie, and K. Edward, "Hex size in JTLS," *Rolands Associates Corp.*, vol. 30, Apr. 2002.

[8] K. Ikeda and K. Murota, *Bifurcation Theory for Hexagonal Agglomeration in Economic Geography*. Berlin, Germany: Springer, 2014.

[9] J. Ke, H. Yang, H. Zheng, X. Chen, Y. Jia, P. Gong, and J. Ye, "Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[10] B. S. Acharya *et al.*, "Science with the cherenkov telescope array," Sep. 2017, *arXiv:1709.07997*. [Online]. Available: https://arxiv.org/abs/1709.07997

[11] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: Group-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 264–272.

[12] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[13] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2016, pp. 5115–5124.

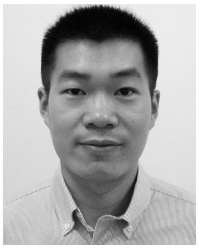
[14] M. Weiler, F. A. Hamprecht, and M. Storath, "Learning steerable filters for rotation equivariant CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 849–858.
[15] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.
[16] J. Ben, Y. Li, C. Zhou, R. Wang, and L. Du, "Algebraic encoding scheme for aperture 3 hexagonal discrete global grid system," *Sci. China Earth Sci.*, vol. 61, no. 2, pp. 215–227, Feb. 2018.
[17] R. B. Games. (2000). *Hexagonal Grids*. [Online]. Available: https://www.redblobgames.com/grids/hexagons/
[18] J. W. van Roessel, "Conversion of Cartesian coordinates from and to generalized balanced ternary addresses," *Photogramm. Eng. Remote Sens.*, vol. 54, no. 11, pp. 1565–1570, 1988.
[19] N. I. Rummelt, "Array set addressing: Enabling efficient hexagonally sampled image processing," Ph.D. dissertation, Dept. Graduate School Univ. Florida, Univ. Florida, Gainesville, FL, USA, 2010.
[20] P. Peterson, "Close-packed uniformly adjacent, multiresolutional overlapping spatial data ordering," U.S. Patent 8 018 458 B2, Sep. 13, 2011.
[21] X. Tong, J. Ben, and Y. Wang, "A new effective hexagonal discrete global grid system: Hexagonal quad balanced structure," in *Proc. 18th Int. Conf. Geoinform.*, Jun. 2010, pp. 1–6.
[22] A. Kageyama and T. Sato, "Yin-Yang grid: An overset grid in spherical geometry," *Geochem., Geophys., Geosyst.*, vol. 5, no. 9, pp. 1–15, Sep. 2004.
[23] G. H. Dutton, *A Hierarchical Coordinate System for Geoprocessing and Cartography*. Berlin, Germany: Springer, 1999.
[24] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, "DeepSphere: Efficient spherical convolutional neural network with HEALPix sampling for cosmological applications," *Astron. Comput.*, vol. 27, pp. 130–146, Apr. 2019.
[25] L. Meng, X. Tong, S. Fan, C. Cheng, B. Chen, W. Yang, and K. Hou, "A universal generating algorithm of the polyhedral discrete grid based on unit duplication," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 3, p. 146, Mar. 2019.
[26] M. A. Mostafavi, C. Gold, and M. Dakowicz, "Delete and insert operations in Voronoi/Delaunay methods and applications," *Comput. Geosci.*, vol. 29, no. 4, pp. 523–530, May 2003.
[27] J. R. Baumgardner and P. O. Frederickson, "Icosahedral discretization of the two-sphere," *SIAM J. Numer. Anal.*, vol. 22, no. 6, pp. 1107–1115, 1985.
[28] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3391–3401.
[29] T. Cohen, M. Geiger, and M. Weiler, "A general theory of equivariant CNNs on homogeneous spaces," 2018, *arXiv:1811.02017*. [Online]. Available: https://arxiv.org/abs/1811.02017
[30] R. Chakraborty, J. Bouza, J. Manton, and B. C. Vemuri, "Manifoldnet: A deep network framework for manifold-valued data," 2018, *arXiv:1809.06211*. [Online]. Available: https://arxiv.org/abs/1809.06211
[31] S.-B. Seong, C. Pae, and H.-J. Park, "Geometric convolutional neural network for analyzing surface-based neuroimaging data," *Frontiers Neuroinform.*, vol. 12, p. 42, Jul. 2018.
[32] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.
[33] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539.
[34] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognitio," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2015, pp. 922–928.
[35] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3577–3586.
[36] R. Q. Charles, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2016, pp. 652–660.
[37] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5099–5108.
[38] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3779–3788.
[39] B. Shi, S. Bai, Z. Zhou, and X. Bai, "DeepPano: Deep panoramic representation for 3-D shape recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2339–2343, Dec. 2015.
[40] C. Esteves, Y. Xu, C. Allen-Blanchette, and K. Daniilidis, "Equivariant multi-view networks," 2019, *arXiv:1904.00993*. [Online]. Available: https://arxiv.org/abs/1904.00993
[41] T.-H. Wang, H.-J. Huang, J.-T. Lin, C.-W. Hu, K.-H. Zeng, and M. Sun, "Omnidirectional CNN for visual place recognition and navigation," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 2341–2348.
[42] B. Coors, A. P. Condurache, and A. Geiger, "SphereNet: Learning spherical representations for detection and classification in omnidirectional images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 518–533.
[43] Y.-C. Su and K. Grauman, "Learning spherical convolution for fast features from 360° imagery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 529–539.
[44] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning SO(3) equivariant representations with spherical CNNs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 52–68.
[45] R. Kondor, Z. Lin, and S. Trivedi, "Clebsch–Gordan nets: A fully Fourier space spherical convolutional neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10117–10126.
[46] Y.-C. Su and K. Grauman, "Kernel transformer networks for compact spherical convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9442–9451.
[47] T. S. Cohen, M. Geiger, J. Köehler, and M. Welling, "Spherical CNNs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Feb. 2018, pp. 1–15.
[48] F. Zhao, S. Xia, Z. Wu, D. Duan, L. Wang, W. Lin, J. H. Gilmore, D. Shen, and G. Li, "Spherical U-net on cortical surfaces: Methods and applications," in *Proc. Int. Conf. Inf. Process. Med. Imag.* New York, NY, USA: Springer, 2019, pp. 855–866.
[49] D. Worrall and G. Brostow, "CubeNet: Equivariance to 3D rotation and translation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 567–584.
[50] C. Steppa and T. L. Holch, "HexagDLy—Processing hexagonally sampled data with CNNs in pytorch," *SoftwareX*, vol. 9, pp. 193–198, Jan./Jun. 2019.
[51] M. Jacquemont, L. Antiga, T. Vuillaume, G. Silvestri, A. Benoit, P. Lambert, and G. Maurin, "Indexed operations for non-rectangular lattices applied to convolutional neural networks," in *Proc. 14th Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, Feb. 2019, pp. 1–10.
[52] C. M. Jiang, J. Huang, K. Kashinath, Prabhat, P. Marcus, and M. Niessner, "Spherical CNNs on unstructured grids," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–16.
[53] Y. Lee, J. Jeong, J. Yun, W. Cho, and K.-J. Yoon, "SpherePHD: Applying CNNs on a spherical polyhedron representation of 360deg images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9181–9189.
[54] C. Zhang, S. Liwicki, W. Smith, and R. Cipolla, "Orientation-aware semantic segmentation on icosahedron spheres," in *Proc. ICCV*, Jul. 2019, pp. 1–9.
[55] T. S. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, "Gauge equivariant convolutional networks and the icosahedral CNN," in *Proc. 36th Int. Conf. Mach. Learn.*, May 2019, pp. 1321–1330.
[56] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5028–5037.
[57] S. Dieleman, J. De Fauw, and K. Kavukcuoglu, "Exploiting cyclic symmetry in convolutional neural networks," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 1–10.
[58] J. Li, Z. Yang, H. Liu, and D. Cai, "Deep rotation equivariant network," *Neurocomputing*, vol. 290, pp. 26–33, May 2018.
[59] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2990–2999.
[60] B. Gardiner, S. Coleman, and B. Scotney, "Comparing hexagonal image resampling techniques with respect to feature extraction," in *Proc. 14th Int. Mach. Vis. Image Process. Conf.*, Jul. 2011, pp. 102–115.
[61] A. Sinha, J. Bai, and K. Ramani, "Deep learning 3D shape surfaces using geometry images," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 223–240.

[62] Z. Cao, Q. Huang, and R. Karthik, "3D object classification via spherical projections," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 566–574.
[63] M. Liu, F. Yao, C. Choi, S. Ayan, and K. Ramani, "Deep learning 3D shapes using alt-az anisotropic 2-sphere convolution," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Sep. 2018, pp. 1–14.
[64] Q. Qiu, X. Cheng, R. Calderbank, and G. Sapiro, "Dcfnet: Deep neural network with decomposed convolutional filters," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2018, pp. 1–14.
[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[66] R. M. Anwer, F. S. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen, "Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 138, pp. 74–85, Apr. 2018.
[67] Y. Yu and F. Liu, "A two-stream deep fusion framework for high-resolution aerial scene classification," *Comput. Intell. Neurosci.*, vol. 2018, Dec. 2018, Art. no. 8639367.

JUNREN LUO received the B.Eng. degree in command automation engineering from Information Engineering University, Zhengzhou, China, in 2012. He is currently pursuing the M.S. degree in control science and engineering from the National University of Defense and Technology (NUDT), Changsha, China. His current research interests include intelligence decision-making and control, adversarial planning, and human-machine teaming.

WANPENG ZHANG received the B.Eng., M.S., and Ph.D. degrees in automatic control from the National University of Defense and Technology (NUDT), Changsha, China, in 2004, 2007, and 2012, respectively, where he is currently an Associate Professor with the College of Intelligence Science and Technology. In 2012, he joined the College of Mechatronics and Automation, NUDT. He has directed five research projects. He has coauthored two books. His research interests include the intelligence decision, mission planning, automation and control engineering, and human-machine teaming.

JIONGMING SU received the B.Eng., M.S., and Ph.D. degrees in control science and engineering from the National University of Defense and Technology (NUDT), Changsha, China, in 2007, 2009, and 2014, respectively, where he is currently a Lecturer with the College of Intelligence Science and Technology. In 2014, he joined the College of Mechatronics and Automation, NUDT. He has directed two research projects, including the National Science Foundation of China and Hunan Province. He has coauthored a book and has published over ten articles in refereed international journals and academic conferences proceedings. His research interests include computer vision, pattern recognition, and intelligence decision-making.



FENGTAO XIANG received the M.Sc. and D.Sc. degrees in control science and engineering from the National University of Defense and Technology (NUDT), Changsha, China, in 2010 and 2014, respectively, where he is currently a Lecturer. His main research interests include sparse representation, computer vision, intelligent systems, and pattern recognition.