FisheyeHDK: Hyperbolic Deformable Kernel Learning for Ultra-Wide Field-of-View Image Recognition

Ola Ahmad[1], **Freddy Lecue**[1, 2]
[1] Thales, Thales Digital Solutions, Canada
[2] INRIA, France
ola.ahmad@thalesdigital.io, freddy.lecue@inria.fr

Abstract

Conventional convolution neural networks (CNNs) trained on narrow Field-of-View (FoV) images are the state-of-the art approaches for object recognition tasks. Some methods proposed the adaptation of CNNs to ultra-wide FoV images by learning deformable kernels. However, they are limited by the Euclidean geometry and their accuracy degrades under strong distortions caused by fisheye projections. In this work, we demonstrate that learning the shape of convolution kernels in non-Euclidean spaces is better than existing deformable kernel methods. In particular, we propose a new approach that learns deformable kernel parameters (positions) in hyperbolic space. FisheyeHDK is a hybrid CNN architecture combining hyperbolic and Euclidean convolution layers for positions and features learning. First, we provide intuition of hyperbolic space for wide FoV images. Using synthetic distortion profiles, we demonstrate the effectiveness of our approach. We select two datasets - Cityscapes and BDD100K 2020 - of perspective images which we transform to fisheye equivalents at different scaling factors (analogue to focal lengths). Finally, we provide an experiment on data collected by a real fisheye camera. Validations and experiments show that our approach improves existing deformable kernel methods for CNN adaptation on fisheye images.

Introduction

Fisheye cameras are designed with ultra-wide field of view (FoV) lenses to offer wide images. They are commonly used in many computer vision applications. In particular, autonomous vehicles heavily rely on perception tasks such as semantic segmentation and object detection from the environment surrounding the vehicle for path and motion planning, driving policy, and decision making (Deng et al. 2020; Yogamani et al. 2019). To optimize the load on autonomous vehicles, only four fisheye cameras each with FoV up to $180°$ can be used to provide the total $360°$ scene coverage (Yogamani et al. 2019). Automated drones (Hrabar et al. 2005), augmented reality (Schmalstieg and Höllerer 2017) and surveillance (Kim et al. 2015) are other interesting applications of large FoV cameras. However, unlike pinhole projection models of narrow FoV cameras (where straight lines in 3D world are projected to straight lines in image plane), fisheye images suffer from non-linear distortion in which straight lines are mapped to curvilinear (distorted lines) to provide large FoV on a finite spatial support. According to (Kannala and Brandt 2006), fisheye distortion does not obey one specific projection model. A general polynomial mapping of fourth order was proven to be an accurate approximation of fisheye camera model (Yogamani et al. 2019; Yin et al. 2018). This model was used to synthesize fisheye effects on perspective images. It was also used for calibration and distortion correction prior to fisheye image recognition tasks. However, estimating the parameters of the polynomial model of order four is an inverse problem which is ill-posed, in practice, under the lack of information and careful assumptions. In this paper, we address the challenge of ultra-wide FoV image recognition without the need for distortion correction. A challenge that is quite recent in computer vision research areas.

Convolutional neural networks (CNNs) have achieved state-of-the-art results on perception tasks when trained on perspective images. However, their performance significantly drops when applied on or transferred to fisheye images. Convolution networks rely on the translation invariant property and use fixed kernel shapes over all image plane. The translation invariance property makes their training harder on fisheye images because of the spatially varying distortion. Therefore, some CNN model adaptation methods based on kernel adaptation and learning were proposed to solve recognition tasks from large FoV images. Our work is inspired by the deformable kernel learning concept (Dai et al. 2017; Jeon and Kim 2017) which aims to learn local offset fields prior to convolution units in standard CNNs. Here, we propose a novel method that learns deformable kernels in hyperbolic spaces for wide FoV images; we generalize the original concept to provide a flexible solution for non-linear offsets.

Motivation We are motivated by learning CNNs aware to the geometric distortions tied to wide FOV cameras such as fisheye. We assume an existing analogy between hyperbolic space, more particularly the Poincaré Ball Model, and fisheye projections. The poincaré ball is a stereographic projection of the hyperbolic space on a disk in image plane (Fig. 2). The poincaré ball is a conformal mapping that preserves angles between distorted lines. We can thus obtain

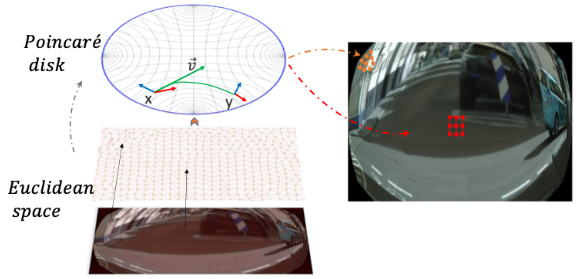

Figure 1: Learning deformable kernels in hyperbolic space for adapting CNNs to large FoV (fisheye) images. Image and input feature maps are represented on a graph and mapped to the poincaré disk for learning positions in a $(k \times k)$ receptive field K_x at every spatial location x.

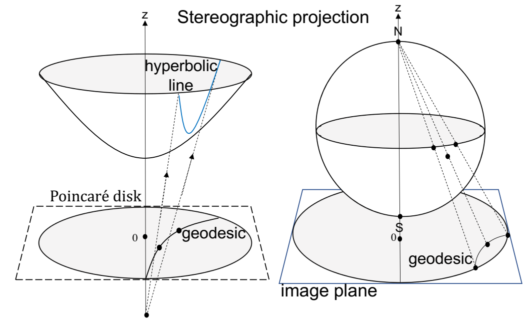

Figure 2: The stereographic projection of the second half sphere from north pole is analogue to the projection of hyperboloid on poincaré disk.

a one to one correspondance (bijective mapping) between Euclidean and hyperbolic spaces for kernel sampling. Our intuition is to learn local displacements of deformable kernels on diffeomorphisme using geodesic (Poincaré) metric of non-Euclidean space to better capture smooth non-linear distortions of fisheye effects.

Contributions In this work, we introduce *FisheyeHDK*, a hybrid neural network that combines hyperbolic and Euclidean convolution layers to learn the shape and weights of deformable kernels (Fig. 1), respectively. We build deformable kernel functions on top of existing convolution layers of Euclidean CNNs. Using synthetic fisheye data, we conduct extensive experiments and ablation studies to illuminate the intuition behind our approach and demonstrate its effectiveness. The data generation process implies converting narrow FoV datasets (perspective images) to distorted ones using fisheye polynomial model controlled by setting the focal length-like parameter at arbitrary values generating different severity levels of distortion. For this task, we selected Cityscapes [1] and BDD100K [2] datasets with pixel-level annotations. Through our experiments we provide in-depth analysis on the effect of learning deformable kernels in both Euclidean and hyperbolic spaces. We show that our method improves the performance of CNN semantic segmentation by an average gain of 2% on synthetic distortions. We also provide experimental results on a set of real-world images collected from fisheye camera and show that our method has better accuracy than baseline methods (more than 3%).

Related Works

Fisheye augmentation Recent research works have started studying how to directly adapt existing state-of-the-art object recognition models to fisheye images. In the context of object detection, Goodarzi *et al.* (Goodarzi et al. 2019) have optimized a standard CNN detector for fisheye cameras through data augmentation in which synthetic fisheye effect was generated on training data using radial distortion. Ye *et al.* (Ye et al. 2020) have recently learned a universal semantic segmentation CNN on urban driving images using a seven degrees of freedom geometric transformation as fisheye augmentation. Although, fisheye augmentation and fine-tuning are the simplest model adaptation techniques, there exist a fundamental limitation in applying these techniques with CNNs. The spatially variant (non-linear) distortion caused by large FoVs breaks the translation invariance property of the standard CNNs designed on regular grid structures (Yogamani et al. 2019).

Kernels adaptation Some works proposed to transfer planar CNNs on sphere assuming that fisheye lens produces spherical images (Coors, Condurache, and Geiger 2018; Su and Grauman 2019). To minimize distortion, SphereNet (Coors, Condurache, and Geiger 2018) adapts the sampling locations of convolution kernels using projection on sphere and tangent plane. Kernel Transformer Network (Su and Grauman 2019) uses equirectangular projection of spherical images to adapt the shape of the kernel based on matching activation maps between perspective and equirectangular images. These methods originally proposed for FoV cameras of 360° in which the resulted image can be perfectly projected on a sphere. The problem with fisheye cameras is that the FoV can vary between 180° and 280°. This makes the sphere adaptation not an optimal solution for fisheye images as reported by (Yogamani et al. 2019). Fisheye image could be the result of one of four projections: Stereographic, Equidistant, Equi-solid and Orthogonal (Kannala and Brandt 2006) based on the large FoV lens properties. An inverse mapping to a region on the sphere requires knowing a priori the FoV and center of distortion.

Learning deformable kernels Instead, learning deformable kernels is a projection-free concept which is generic and applicable to FoVs smaller than 360°. Our work is related to this line of research. Deformable Convolution Network (DCN) is originally proposed by (Dai et al. 2017) and applied on perspective images to improve recognition tasks. The DCN augments standard convolution layers with learnable 2D offsets to sample deformable kernels for convolution operations. Here, kernels are sampled at each location in the spatial support of input features. Later, (Deng

[1]https://www.cityscapes-dataset.com/

[2]https://bdd-data.berkeley.edu/

et al. 2020) have directly applied this idea on fisheye images for semantic segmentation. However, to preserve the spatial correspondence between input images and the predicted semantic maps, they proposed a slight modification. They restricted offsets learning only in the neighbour locations of the kernel's center which kept unlearnable. This modified version of deformable convolution was called RDC (Deng et al. 2020). We argue that fixing the center of the kernel cannot resolve the fundamental limitation inherit in learning offsets in Euclidean space for wide FoV images. Compared to them, our method is more generic and flexible. It learns those kernels in hyperbolic space, more particularly, Poincaré ball model of hyperbolic space, and show that this model better captures fisheye effect than Euclidean methods. A new insight that could inspire future research on non-perspective cameras. The shape of convolution kernels is learnable and change flexibly over the spatial support, which makes CNN model adaptation applicable on large to ultra-wide FoV cameras. We provide a figure illustrating these effects on a toy example in supplementary material (Ahmad and Lecue 2022).

Review of Hyperbolic Geometry and the Poincaré Ball

An d-dimensional hyperbolic space, denoted $\mathbb{H}^d$, is a homogenous, simply connected, n-dimensional Riemannian manifold with a constant negative curvature c. Analogous to sphere space (which has constant positive curvature), hyperbolic space is a space equipped with non-Euclidean (hyperbolic) geometry in which distances are defined by *geodesics* (shortest path between two points). Hyperbolic space has five isometric models: the Klein model, the hyperboloid model, the Poincaré half space model and the Poincaré ball model (Cannon et al. 1997). A mapping between any two of these models preserves all the geometric properties of the space. In this work, we choose the Poincaré ball model due to its conformal mapping properties (w.r.t. Euclidean space) and analogy to fisheye distortion. The Poincaré ball model is defined by the Riemannian manifold ($\mathbb{D}^d_c$, $g^{\mathbb{D}_c}$), where $\mathbb{D}^d_c := \{x \in \mathbb{R}^d | \|x\| < 1/\sqrt{c}\}$ is an open ball of radius $1/\sqrt{c}$ and its Riemannian metric is given by $g^{\mathbb{D}_c}_x = (\lambda^c_x)^2 g^{\mathbb{E}}$ such that $\lambda^c_x := \frac{2}{1-c\|x\|^2}$ and $g^E = \mathbf{I}_d$ denotes the Euclidean metric tensor (the dot product). The induced distance between two points $x, y \in \mathbb{D}^d_c$ is given by

$$d_{\mathbb{D}^c}(x,y) = \frac{1}{\sqrt{c}} \cosh^{-1}\left(1 + 2\frac{\|x-y\|^2}{(1-c\|x\|^2)(1-c\|y\|^2)}\right) \tag{1}$$

In hyperbolic space, the natural mathematical operations between vectors, such as vectors addition, subtraction and scalar multiplication are described with Möbius operations (Ungar 2008). The Möbius addition of x and y in $\mathbb{D}^d_c$ is defined as

$$x \oplus_c y := \frac{(1 + 2c\langle x, y\rangle + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c\langle x, y\rangle + c^2\|x\|^2\|y\|^2} \tag{2}$$

and the Möbius scalar multiplication of $x \in \mathbb{D}^d_c \backslash \{\mathbf{0}\}$, $c > 0$, by $a \in \mathbb{R}$ is defined as

$$a \otimes_c x := \frac{1}{\sqrt{c}} \tanh\left(a \tanh^{-1} \sqrt{c}\|x\|\right) \frac{x}{\|x\|} \tag{3}$$

Note that subtraction can be obtained by $x \oplus_c (-1 \otimes_c y) = x \oplus_c -y$. When c goes to zero, one recovers the natural Euclidean operations. The bijective mapping between the Riemannian manifold of the Poincaré ball ($\mathbb{D}^d_c$) and its tangent space (Euclidean vectors $\mathcal{T}_x\mathbb{D}^d_c \cong \mathbb{R}^d$) at given point is defined by the exponential and logarithmic maps. To do so, Ganea *et al.* (Ganea, Becigneul, and Hofmann 2018) derived a closed-form of the exponential map $\exp^c_x : \mathcal{T}_x\mathbb{D}^d_c \to \mathbb{D}^d_c$ and its inverse $\log^c_x : \mathbb{D}^d_c \to \mathcal{T}_x\mathbb{D}^d_c$ for $v \neq 0$ and $y \neq x$ such that

$$\exp^c_x(v) = x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda^c_x\|v\|}{2}\right)\frac{v}{\sqrt{c}\|v\|}\right) \tag{4}$$

$$\log^c_x(y) = \frac{2}{\sqrt{c}\lambda^c_x} \tanh^{-1}\left(\sqrt{c}\| -x \oplus_c y\|\right) \frac{-x \oplus_c y}{\| -x \oplus_c y\|} \tag{5}$$

As reported by (Ganea, Becigneul, and Hofmann 2018), the maps have nicer forms when $x = \mathbf{0}$. This makes the mapping between Euclidean and hyperbolic spaces obtained by $\exp^c_{\mathbf{0}}$ and $\log^c_{\mathbf{0}}$ more useful in practical point of view.

Proposed Approach

In this section, we introduce the proposed *FisheyeHDK* approach. Our method does not require a ground truth information of fisheye geometry but updates the parameters of deformable kernels from optimizing CNN features through back-propagation. During training, hyperbolic deformable kernels are mapped to the Euclidean space and used in CNN layers to compute the output feature map (see Fig. 3). Section explains how the input features are encoded in hyperbolic space to implement the HDK network. Section presents the architecture of the HDK network and CNN implementation with deformable kernels.

Input Representations for HDK Network

To leverage spatial information with feature vectors in hyperbolic space we represent input feature maps as graphs. Images can naturally be modeled as a graph. The simplest graph model is defined on regular grid, where vertices correspond to pixels encoding features information and edges represent their spatial relations. This representation, however, requires considerable computations and memory for large grids. To alleviate such complexity and reduce inputs dimensionality, we downsampled the resolution of spatial grids by a factor of 2^m ($m = 2$ as default). This allowed faster computations with insignificant effect on the performance, and enabled generating the graph from input features online. We used CUDA implementations and the open source library Pytorch-geometric [3] to generate graphs from

[3]https://github.com/rusty1s/pytorch_geometric

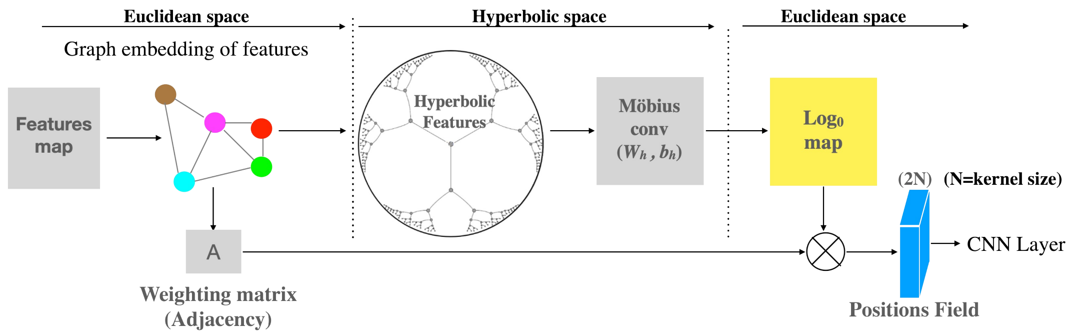

Figure 3: FisheyeHDK convolution layer. HDK network is one hyperbolic convolution layer predicting the positions in a deformable kernel of size $(k \times k)$ for each point of the spatial support. Conventional (Euclidean) convolution is applied between the weights sampled at predicted positions and input feature map.

grid feature maps. Consequently, the input to the HDK network are: vertices matrix $V \in \mathbb{R}^{N \times d}$, where N is the number of vertices and d is the feature dimension, and adjacency matrix A of size $N \times N$ encoding the spatial information. We refer to supplementary material for more details about image to graph generation.

FisheyeHDK Architecture

To implement FisheyeHDK, we build a hybrid architecture combining non-Euclidean (hyperbolic) convolution layers for kernel's positions learning and Euclidean convolution layers for features learning. Euclidean convolution layers belong to a conventional CNN model chosen from existing architectures. These layers apply convolution on an input feature map using deformable kernels sampled from the output of hyperbolic layers as shown in Figure 3. Hyperbolic convolution layers use preceding feature maps converted to graph to learn the shape of deformable kernels in hyperbolic space. We describe in the following the hybrid components of the proposed architecture.

HDK network comprises one hyperbolic convolution layer. Euclidean feature vectors are projected on hyperbolic space using Exp. map as described in Eq. (6):

$$H_v = \exp_{\mathbf{0}}^c(F_v), \tag{6}$$

where F_v is the Euclidean feature vector, and H_v is its projection on hyperbolic space for a give vertex v. A Möbius layer performs linear transformations on feature vectors inside Poincaré Ball (Eqs. (2) and (3)). Möbius features are mapped to the Euclidean space using the Log mapping as shown in Fig. 3, and the spatial information encoded by the adjacency matrix are aggregated with projected features on the tangent (Euclidean) space using an aggregation layer such as

$$K = \log_{\mathbf{0}}^c\left((W_h \otimes_c H) \oplus_c b_h\right) \odot A, \tag{7}$$

where $\odot$ denotes element-wise product, W_h and b_h are hyperbolic weights and bias vectors, K is a dense map of deformable kernels representing the positions inside $(k \times k)$ window at every point of the grid. We used the open source library geoopt [4] to build the HDK network. Additional details about the HDK network are provided in supplementary material (Ahmad and Lecue 2022).

Experiments

In this section, we present our experiments on the task of semantic segmentation. For the sake of generalization to other recognition tasks, we focus on the encoder components of the CNN model and demonstrate the effectiveness of hyperbolic deformable kernels versus their Euclidean counterparts for large FoV images. We tested our approach on three datasets. Two datasets of perspective images were transformed to fisheye using the general fisheye model. We rely on the model used in the open source library OpenCV [5]. The third dataset are real-world images collected by a fisheye camera.

Datasets

Cityscapes are perspective images with pixel-level annotations collected from urban German cities. The dataset comprises 5000 images divided into train, validation and test sets (2975, 500 and 1525 images respectively). Annotations consist of 30 classes but only 19 are defined as valid. The test set is provided without annotation maps, therefore we only used train and validation sets. The validation set was used as the test set and the original training set was split in two (0.9/0.1 ratio) for training and validation purposes. Their original resolution is (1024×2048) pixels. Three fisheye datasets were generated from perspective cityscapes images using the following focal lengths: 200, 125 and 50. Images and ground truth maps are resized to (512×1024) pixels in all three datasets.

BDD100K A large-scale divers dataset recently released for perception tasks in autonomous driving. It was collected from divers and complex environments in US cities. The dataset for the segmentation task is composed of 10000 perspective images with fine-grained pixel-level annotations

[4] https://github.com/geoopt/geoopt

[5] https://docs.opencv.org/3.4/db/d58/group__calib3d__fisheye.html

Method	# First layers (l)			# Last layers (l)			
	$l=1$	$l=3$	$l=6$	$l=1$	$l=3$	$l=6$	$l=all$
RDC (Deng et al. 2020)	57.9	58.0	57.9	57.8	57.7	57.9	56.8
FisheyeHDK (Ours)	58.3	58.4	58.5	58.5	59.6	59.3	58.0

Table 1: Test results of Mean intersection over union (mIoU (%)) on distorted cityscapes with $f = 200$. Deformable kernels are applied on ResNet101 module. First and last layers are convolutions starting from conv1 and layer4.1.conv3, respectively.

(including 40 object classes). Images and annotations are divided into train, validation and test sets (7000, 1000, 2000, respectively). Their original resolution is (720×1280) pixels. Annotations are not provided in the test set. Similar to Cityscapes, we use the validation set as test set and split the original train set into two sets: 6500 for training and 500 for validation. For the experiments conducted on this dataset, we generated two fisheye datasets using smaller focal lengths set to the values of 75 and 50. Images and ground truth maps are resized to (700×1024) pixels.

Real data A dataset collected using real fisheye cameras mounted in a driving vehicle. The dataset comprises 800 real-world images with pixel-level annotations similar to Cityscapes. Annotations include few set of classes which are: person, car, train, truck, traffic light, cyclist, motorcycle and bus. The resolution of fisheye images is (512×1024) pixels. We randomly split the dataset into three sets: 680 for training, 20 for validation and 100 images for testing. We kept the original resolution of images during all training phases. We apply data augmentation by randomly left-right flipping and by randomly changing color information (brightness, contrast, hue and saturation) during training.

Implementation Details

Training protocol Without loss of generality, we built hyperbolic deformable kernel components on top of the DeepLabV3 architecture. Any alternative semantic segmentation network can work. We trained the model on 2 GPUs using synchronized batch-norm. We implemented the pixel-wise weighted cross-entropy as a loss function. For baselines, we used Stochastic Gradient Decent (SGD) optimizer with momentum 0.9 and weight decay 5×10^{-4}. The learning rate was initialized to 1×10^{-3} for the encoder and 1×10^{-2} for the decoder; both are updated using the "poly" learning rate policy. Our approach comprises both Euclidean and hyperbolic parameters. For hyperbolic parameters, we adopted the Riemannian SGD (RSGD) (Bonnabel 2013) with a learning rate (1×10^{-2}) since they are manifold parameters and used usual SGD for the Euclidean parameters. We provide more details on optimization in supplementary material (Ahmad and Lecue 2022). We initialized the encoder and decoder parameters of the CNN layers with ImageNet weights. We initialized the hyperbolic weights W_h using the Xavier uniform distribution and hyperbolic biases b_h with zeros for HDK layers. Euclidean deformable kernel weights were initialized by zeros as in (Deng et al. 2020). For the synthetic fisheye dataset, we set the training batch size to 16 and the validation batch size to 4. For real fisheye data, we set the batch size to 8 during training and validation.

Distortion level (f)	Cityscapes Dataset mIoU/mAcc (%)			
	Rect+Seg	RegCNN	RDC	Ours
50	16.3/25.8	45.8/61.2	48.6/60.8	**49.8/61.5**
125	44.6/51.9	51.3/70.0	54.3/69.4	**55.9/70.0**
200	54.4/62.1	53.6/71.8	57.7/71.6	**59.6/72.1**
	BDD100K Dataset mIoU/mAcc (%)			
50	6.9/12.3	30.2/40.6	40.2/52.7	**40.5/53.0**
75	16.9/24.7	38.9/51.7	42.8/**57.6**	**44.0**/56.6

Table 2: Effect of distortion on the performance of segmentation model augmented with deformable kernels in last 3 layers of ResNet101. Smaller is f, stronger is the distortion. We report the results on the test sets of dataset transformed to fisheye.

Method	train/val speeds (on epoch)	test time (on one image)	Size (MB)
RegCNN	**634.3/33**s	0.270s	**237.9**
RDC (Deng et al. 2020)	656/34s	0.278s	238.9
FisheyeHDK-4	680/44s	0.318s	238.0
FisheyeHDK-8	650/35s	0.288s	238.0

Table 3: Comparative results using cityscapes dataset (2677 train images, 298 val images) converted to fisheye. All networks are trained on 2 Nvidia Tesla P100 GPUs each with 16Gb memory. Images resolutions is (512×1024) pixels

In all experiments, we trained the models for 100 epochs. We used per-class accuracy and the standard mean Intersection-Over-Union (mIoU) as evaluation metrics for validation (after each epoch) and testing on test sets after training. We report evaluation metrics on the "valid" classes provided with Cityscapes and BDD100K datasets. Void class, corresponding to unsegmented or irrelevant objects, and image's borders resulted from fisheye transformation were ignored.

Baseline Models

We compare our approach with conventional CNN models, and with existing deformable convolution methods (Dai et al. 2017; Deng et al. 2020). We choose one as both learn offsets in same way and our preliminary tests did not show significant differences (see supplementary material in (Ahmad and Lecue 2022) for details). We report in this paper the experiments conducted on the RDC method (Deng et al.

Method	Per-Class IoU / Per-Class Accuracy (%)							mIoU/mAcc
	traffic light	person	car	truck	bus	train	motorcycle	
Rect+Seg	-	26.7/29.2	57.5/70.6	-	-	-	-	42.5/49.9
RegCNN	65.6/88.6	86.7/88.3	77.8/93.4	72.7/79.7	58.6/1.00	25.8/26.3	69.7/69.8	65.3/78.0
RDC (Deng et al. 2020)	**64.5**/83.6	**96.5/97.8**	78.9/94.0	69.4/75.6	**71.8**/99.0	79.4/**85.9**	73.3/73.8	76.3/87.2
FisheyeHDK	63.0/**91.5**	96.2/97.4	**83.6/95.2**	**81.7/85.4**	71.7/99.0	**81.9**/83.2	**82.9/83.1**	**80.1/90.8**

Table 4: Per-class accuracy and IoU (%) on the test set of real fisheye images.

2020). Even though deformable convolutions can be built on any state-of-the-art CNN architecture, we limit our implementations to one CNN semantic segmentation model and focus on the effect of hyperbolic and Euclidean deformable kernels. We re-implement the RDC approach on fisheye datasets that we generated with different focal length values using same training protocols. As a baseline CNN segmentation model, we choose the DeepLabV3 architecture in which deformable kernels are applied in the backbone component of the encoder (ResNet101). We report additional results on other backbone architectures in supplementary material. Accordingly, two baseline models were used for comparisons: a regular CNN without deformable kernels (RegCNN) and a deformable one based on (Deng et al. 2020). Additional to these baselines, we compared with fisheye undistortion technique (based on inverted equidistant projection) prior to a standard pre-trained segmentation network (Rect+Seg). The segmentation network is a DeepLapV3 trained on original cityscapes images.

Ablation Study

We investigate the effect of using deformable kernels in CNN for distorted images. Then, we analyze the distortion level's impact on the CNN model's performance with an emphasize on the segmentation task. Finally, we analyze the efficiency of our approach compared to baselines.

Connection of deformable kernels and distortion In this study, we answer the question: ***"which CNN layers needed to be deformable to learn better representations from distorted inputs?"***. We hypothesize that low-level and high-level features are strongly correlated with fisheye distortion. Low-level layers capture spatial information such as distorted edges and lines, while high-level features represent semantic information of distorted objects. Typically, distortion is more visible in low-level features, however, it is unclear to what extent deformable kernels used only on first layers would improve learning in the following convolution layers. In baseline methods (Dai et al. 2017; Deng et al. 2020), convolutions with deformable kernels were limited to the few last layers of the CNN. To provide a deeper understanding on the relation between deformable kernels and fisheye distortion, we experiment all layers with emphasize on low-level and high-level layers. Therefore, both HDK and RDC networks were added prior to the CNN layers of the backbone module. We started with low-level layers and progressively added deformable kernels to l convolutional layers, ($l = \{1, 3, 6\}$), then we did the same protocol on the last layers and finally on all layers. For this analysis, we used perspective cityscapes dataset transformed to fisheye with $f = 200$. Table 1 shows the mIoU results of the segmentation model. As expected, augmenting low-level and high-level convolution layers with learned positions improves the performance better than using deformable kernels in all layers. According to these results, we can apply deformable convolution on few low-level or high-level layers and have almost similar performance. In the next of our experiments, we select as default the backbone variant with deformable convolution in the last three layers. Moreover, Table 1 shows that HDK networks improve the performance of the model compared to their Euclidean counterparts.

Distortion effect on CNN performance The level of distortion in large FoV images depends on fisheye lenses. Severe geometric distortions result from using lenses with small focal length (f) to compensate wide FoV angles (θ) when rays (R) projected on finite image plane ($R = f.\theta$). In this experiment, we test our approach and the baseline methods on perspective datasets transformed to fisheye using different distortion levels: $f \in \{50, 75, 125, 200\}$. Table 2 reports the quantitative results of the performance of segmentation model. Figure 4 shows qualitative results of different methods. As expected regular CNN is the worst-performing on distorted images. Our approach improves regular CNN better than RDC approach on both datasets (Cityscapes and BDD100K). Undistortion has bad influence on segmentation performance and leads to a loss of FoV.

Efficiency and model size As mentioned in Section , HDK inputs are represented as structure of graph with N nodes and $(N \times N)$ adjacency matrix. Feature maps should be converted to graph during training which adds additional computational cost to the model. The training speed decreases significantly when deformable kernels applied on high spatial resolutions. To keep efficiency comparable to regular CNN and RDC, we tested the performance versus training speed using two downsampling factors 2^m with $m = [2, 3]$. We refer to these networks by FisheyeHDK-4 and FisheyeHDK-8, respectively. Table 3 lists the comparative results with baseline methods. Our method is efficient and more accurate than the deformable convolution network when spatial inputs are downsampled by a factor of 4. Our method becomes significantly slower when using lower downsampling ($m = 1$). This is due to the computational cost needed to compute adjacency and upsample the kernels field up to the spatial resolution of the features input. Furthermore, strong downsampling leads to loss of information which could degrade the performance of our approach. Our approach does not significantly increase the size of the

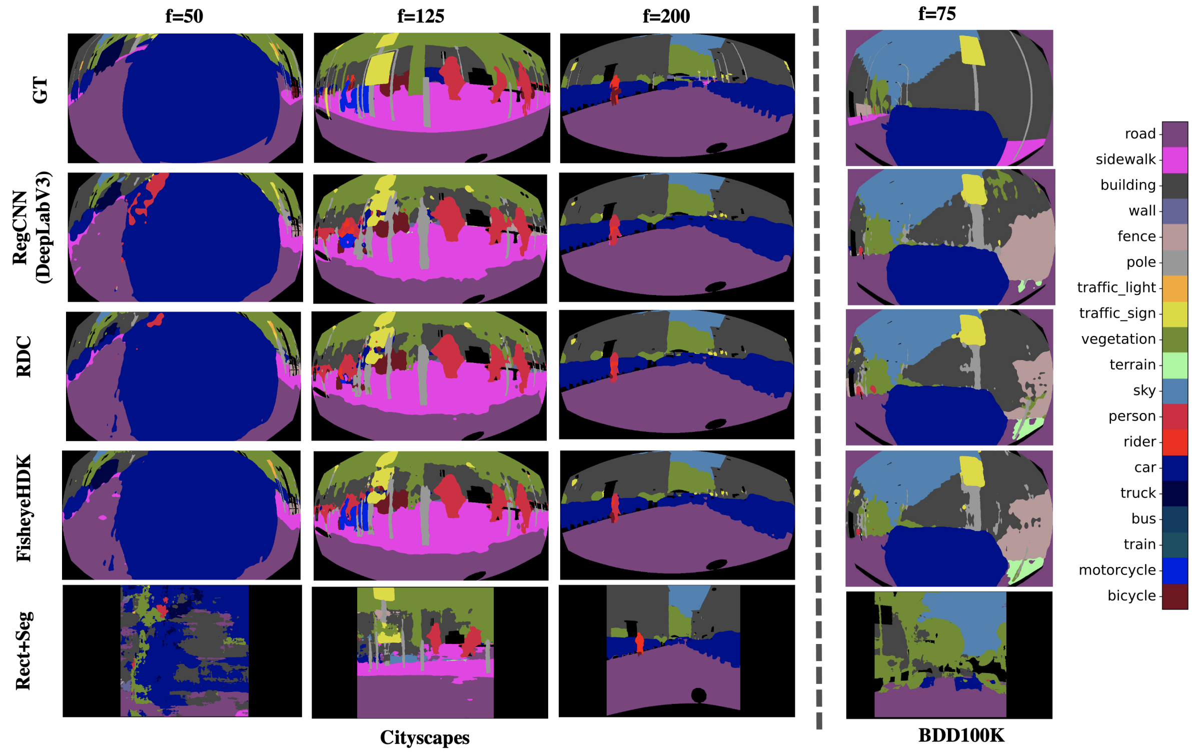

Figure 4: Qualitative results of different segmentation methods. On BDD100K dataset, we considered the road label as a void class.

baseline CNN architecture compared to the RDC. As illustrated in Table 3, all methods are close to the baseline CNN when deformable kernels applied on the last three layers. However, this result changes when deformable kernels networks are used prior to all backbone layers. Our model remains relatively light 238.5MB (+0.25%), while the model size of the of RDC method increases to 243.3MB (+2.2%). In supplementary (Ahmad and Lecue 2022), we provide a comparison between inference times of both methods.

Evaluation on Real Data

Given the small size of this dataset, we applied transfer learning of feature weights from the baseline segmentation model trained on perspective cityscapes to real fisheye dataset and trained FisheyeHDK network for only 20 epochs. Deformable kernel network parameters were initialized as explained in Section , and used only in the last 3 layers of the backbone module. Table 4 shows the quantitative results of per-class metrics (accuracy and IoU). The averaged metrics show that our method outperforms the baseline methods. On real fisheye distortion, the results prove that deformable kernel methods are better than using regular kernels in conventional CNN models. Segmentation after undistortion seems highly affected by the loss of FoV on this dataset. Some classes are ignored because they become less representative after undistortion. It is worth noting, for future work, that the small size of data reflects the reality of our world where the access and cost of annotating data is crucial. Further improvements and validations of our approach would consider this bottleneck.

Conclusion

In this work, we introduced FisheyeHDK, a method that adapts regular CNN models on large FoV images based on deformable kernel learning. We proposed a novel approach that learns the shape of deformable kernels (positions) in hyperbolic space and demonstrated its effectiveness on synthetic and real fisheye datasets. For the first time, we empirically demonstrated that hyperbolic spaces could be a promising approach for deformable kernel sampling and CNN adaptation to ultra-wide FoV images. Our goal is to provide new insights and inspire future research on non-Euclidean (hyperbolic) geometry for learning deformations from ultra-wide FOV images. We do not intend, in this work, to providing high metrics analogue to perspective images instead to show that deformable kernels learned in Euclidean space are not the optimal solution for fisheye images. We keep further improvements to future work and we will explore the feasibility of using our approach in object detection models. An exciting line of research is to examine the capability of FisheyeHDK model to adapt faster to different tasks without retraining the Euclidean part (features) of the network. That could be crucial due to the lack of annotated fisheye data.

Acknowledgments

The authors would like to acknowledge and thank Loic Messal from Jakarto (https://www.jakarto.com) for providing the private fisheye dataset and annotations. We also acknowledge Thales for funding this research work and the infrastructure team for their support on running our experiments on Kubernetes Platform.

References

Ahmad, O.; and Lecue, F. 2022. FisheyeHDK: Hyperbolic Deformable Kernel Learning for Ultra-Wide Field-of-View Image Recognition. *arXiv preprint arXiv:2203.07255*.

Bonnabel, S. 2013. Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Transactions on Automatic Control*, 58(9): 2217–2229.

Cannon, J. W.; Floyd, W. J.; Kenyon, R.; Walter; and Parry, R. 1997. Hyperbolic geometry. In *In Flavors of geometry*, 59–115.

Coors, B.; Condurache, A. P.; and Geiger, A. 2018. SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images. In *Computer Vision-ECCV 2018*, 525–541. Springer International Publishing.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable Convolutional Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 764–773.

Deng, L.; Yang, M.; Li, H.; Li, T.; Hu, B.; and Wang, C. 2020. Restricted Deformable Convolution-Based Road Scene Semantic Segmentation Using Surround View Cameras. *IEEE Transactions on Intelligent Transportation Systems*, 21(10): 4350–4362.

Ganea, O.; Becigneul, G.; and Hofmann, T. 2018. Hyperbolic Neural Networks. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 5345–5355. Curran Associates, Inc.

Goodarzi, P.; Stellmacher, M.; Paetzold, M.; Hussein, A.; and Matthes, E. 2019. Optimization of a CNN-based Object Detector for Fisheye Cameras. In *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 1–7.

Hrabar, S.; Sukhatme, G. S.; Corke, P.; Usher, K.; and Roberts, J. 2005. Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3309–3316.

Jeon, Y.; and Kim, J. 2017. Active Convolution: Learning the Shape of Convolution for Image Classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1846–1854.

Kannala, J.; and Brandt, S. 2006. A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 1335–1340.

Kim, H.; Chae, E.; Jo, G.; and Paik, J. 2015. Fisheye lens-based surveillance camera for wide field-of-view monitoring. In *2015 IEEE International Conference on Consumer Electronics (ICCE)*, 505–506.

Schmalstieg, D.; and Höllerer, T. 2017. Augmented reality: Principles and practice. In *2017 IEEE Virtual Reality (VR)*, 425–426.

Su, Y.; and Grauman, K. 2019. Kernel Transformer Networks for Compact Spherical Convolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9434–9443.

Ungar, A. A. 2008. A Gyrovector Space Approach to Hyperbolic Geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1): 1–194.

Ye, Y.; Yang, K.; Xiang, K.; Wang, J.; and Wang, K. 2020. Universal Semantic Segmentation for Fisheye Urban Driving Images. arXiv:2002.03736.

Yin, X.; Wang, X.; Yu, J.; Zhang, M.; Fua, P.; and Tao, D. 2018. FishEyeRecNet: A Multi-context Collaborative Deep Network for Fisheye Image Rectification. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision – ECCV 2018*, 475–490. Cham: Springer International Publishing. ISBN 978-3-030-01249-6.

Yogamani, S.; Hughes, C.; Horgan, J.; Sistu, G.; Varley, P.; O'Dea, D.; Uricar, M.; Milz, S.; Simon, M.; Amende, K.; Witt, C.; Rashed, H.; Chennupati, S.; Nayak, S.; Mansoor, S.; Perrotton, X.; and Perez, P. 2019. WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.