Convolution kernel adaptation to calibrated fisheye

Bruno Berenguel-Baeta*
berenguel@unizar.es
Maria Santos-Villafranca*
m.santos@unizar.es
Jesus Bermudez-Cameo
bermudez@unizar.es
Alejandro Perez-Yus
alperez@unizar.es
Jose J. Guerrero
josechu.guerrero@unizar.es

Instituto de Investigacion en Ingenieria de Aragon (I3A)
Universidad de Zaragoza
Zaragoza, Spain

Abstract

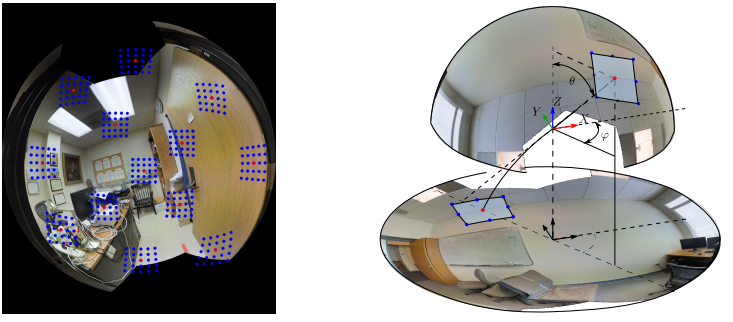
Convolution kernels are the basic structural component of convolutional neural networks (CNNs). In the last years there has been a growing interest in fisheye cameras for many applications. However, the radially symmetric projection model of these cameras produces high distortions that affect the performance of CNNs, especially when the field of view is very large. In this work, we tackle this problem by proposing a method that leverages the calibration of cameras to deform the convolution kernel accordingly and adapt to the distortion. That way, the receptive field of the convolution is similar to standard convolutions in perspective images, allowing us to take advantage of pre-trained networks in large perspective datasets. We show how, with just a brief fine-tuning stage in a small dataset, we improve the performance of the network for the calibrated fisheye with respect to standard convolutions in depth estimation and semantic segmentation. The code of the calibrated deformable kernels is publicly available at https://github.com/Sbrunoberenguel/CalibratedConvolutions.

1 Introduction

Nowadays, Neural Networks are the standard and globally adopted solutions for many machine learning approaches. Among them, Convolutional Neural Networks (CNNs) are the state-of-the-art approaches to handle images and understand what a computer can see. Following classical computer vision algorithms, the first CNNs worked on conventional images (i.e. perspective images). These images provide information of the environment in a

* Equal contribution

This work was supported by projects PID2021-125209OB-I00 and TED2021-129410B-I00 (MCIN/AEI/10.13039/501100011033 and FEDER/UE and NextGenerationEU/PRTR), and DGA 2022-2026 grant.



(a) (b)

Figure 1: Overview of how a standard convolution is deformed by the Kannala-Brandt's projection model in a fisheye image. a) Several convolutional kernels adapted to the calibrated fisheye image. b) How the convolutional kernels are computed with the Kannala-Brandt's projection model.

relatively small field of view, which is usually enough for the CNNs to achieve great performance in many tasks. This is possible due to the large number of labelled datasets of perspective images, which provide an excellent foundation for these solutions.

On a new trend, unconventional cameras with wider fields of view are becoming popular in many applications and devices such as autonomous vehicles or augmented reality. In particular, we focus on fisheye cameras, which provide several advantages in scene understanding problems. The wide field of view (possibly wider than 180°) allows us to achieve a better understanding of the device's surrounding with fewer images. Compared to perspective cameras, with fisheyes it is possible to gather much more spatial information at once, providing more context to the network predictions, especially when the task involve a global understanding of the scene. This is particularly useful in tasks such as semantic segmentation, depth estimation, or room layout estimation.

However, these cameras have some important drawbacks yet to be addressed, which mainly are the reduced number of labelled datasets (mostly because of the difficulty of manually label these images), and the strong distortions induced in large field of view cameras. The last reason causes that existing CNN-based approaches do not transfer well to fisheye images, since the appearance of the elements in the scene is very different to what was learned with perspective datasets. Besides, the contextual information has drastically changed with the much wider field of view, which consequently deteriorates the network's ability to understand the scene. There is definitely a domain gap between perspective and fisheye cameras that needs to be addressed to fully exploit the potential of these devices.

In this work, we propose a method to reduce this domain gap and easily train and use CNNs with highly distorted cameras. We take advantage of the classical perspective CNNs trained with massive datasets, which already provide impressive performance in many different tasks, and adapt these networks to calibrated fisheye cameras. To do so, we propose to substitute the standard convolution operation with deformable convolutions pre-computed with the camera calibration. During the convolution operation, kernels are deformed to accommodate to the distortion depending on the position in the image (see Fig. 1). With minimal fine-tuning on a small set of data, we can achieve the good performance of well-known CNNs from perspective cameras to fisheye cameras, not needing to create new large

datasets specific to each desired camera calibration in order to help the network to learn the distortions. The main contributions of this work are:

- We present a novel implementation of calibrated deformable convolution for fisheye cameras under the Kannala-Brandt projection model, which could be used with any fisheye camera (even with a field of view wider than 180°).
- We propose a set of experiments of domain adaptation of well known CNNs for several tasks with fisheye cameras. Particularly, we show results with different fields of view, showing that our method allows great flexibility in the camera configurations with minimal effort.

2 Related work

In recent years, there has been a surge of using fisheye and 360° cameras since they introduce more information within a single image, which is advantageous when tackling different computer vision tasks such as: scene understanding [11], depth estimation [4, 19, 20], semantic segmentation [4, 7], object [12] and pedestrian detection [13], or autonomous driving [29], among others [3, 22, 23]. There is also an increasing presence of wide-angle cameras such as fisheyes in mobile devices such as phones, or VR headsets and stereo cameras, due to the improved robustness in localization and mapping from the larger field of view.

However, when it comes to deep learning methods applied to wide angle images, one of the most common operations, the convolution, is flawed. The space-varying distortion caused by the image projection models for omnidirectional and wide-angle cameras makes the translational weight sharing ineffective [5]. Objects appear differently distorted depending on where they are in the image, which makes more difficult for the network to learn each plausible configuration, especially considering different camera calibrations. Therefore, there is still an open challenge about how to use and train traditional CNN architectures with these kinds of images, also considering the lack of large datasets compared to perspective images. Some researchers have focused on adapting CNNs to the spherical domain. For instance, Cohen et al. [5] proposed Spherical CNNs, studying convolutions on the sphere using spectral analysis. Jian et al. [16] replace conventional convolution kernels with linear combinations of differential operators weighted by learnable parameters on unstructured grids. Su and Grauman [27] aims to adapt a CNN trained on perspective images to the equirectangular domain adjusting the sizes of the rectangular kernels depending on the elevation angle. UniFuse [17] proposed to fuse features from Cubemap projection with regular convolutions on equirectangular images on the decoding stage for depth estimation. In an attempt to make more efficient the computation, [33] use spherical attention masks to make the model aware of its spherical nature.

On the other hand, different approaches have been proposed to enable CNNs to be more dynamic, improving the performance for specific tasks, as well as extending their applicability to new domains [6, 15, 31]. For example, [15] and [6] focus on convolution units with no fixed shape and learned offsets for the convolution kernel. Jeon and Kim [15] use these convolutions to obtain better receptive field for object classification, whereas [6] incorporates spatial deformations into the convolutional operation to be able to handle objects with significant variations in shape or appearance. The deformable convolutions [6] were used in [10] for layout estimation, with not learned but fixed offsets, pre-computed to account for the image distortion that occurs in equirectangular projections. Thus, the receptive field of

the convolution filter is undistorted. The idea of making the convolution "on the sphere" and project the convolution kernel with the equirectangular projection model was also proposed by [28] with their *distortion-aware convolutions*, introducing a pipeline of transfer learning from learned convolution filters in perspective images applied for depth estimation in equirectangular panoramic images. Strategies like transfer learning or domain adaptation have been successfully used in the past [8, 26], and we believe it could alleviate the absence of datasets for our task. This approach was recently explored for distortion-aware convolutions in [7].

However, most works disregard the specific calibration parameters, mostly because they use very simple image projections such as equirectangular projection, that directly map azimuth and elevation angles in pixel locations. Thus, the research on these distortion-aware convolutions for other configurations such as fisheyes is scarce and very specific to the camera pose [21]. Only a few works directly deal with the camera calibration parameters into the convolutions, like CAMConvs [9]. In this work, we aim to breach that gap, introducing novel convolutions for radial-distortion models, like the Kannala-Brandt's projection model [18], that adapts to the specific calibration of the camera and is apt for any fisheye camera. Drawing inspiration from [10], our approach uses deformable convolutions [6] to adapt the convolution filters to the distortion caused by the projection. Up to our knowledge, this is the first work that explicitly deals with calibrated convolutions for radially distorted cameras. We show how, with just a little fine-tuning in a relatively small dataset, classical CNN methods can be adapted to any calibrated camera. Our approach with calibration specific kernel offsets could also be extended to other calibration models and account for their distortion.

3 Fisheye convolution

Convolutions are the keystone of CNNs and current computer vision algorithms. The work [6] presents a learned deformable kernel to improve the performance of several CNNs. In this work, we propose to use calibration-based deformable kernels. We use the camera calibration of fisheye cameras to compute the offsets of the kernel positions and adapt the convolution to the distortion of these images. In this section we summarize the projection model used and how we apply the distortion to the kernels.

3.1 Fisheye projection model

In the literature of non-conventional cameras we find many works that propose mathematical models of several projections. Considering fisheye cameras, we can also find several models such as the *equidistance*, *stereographic*, *orthogonal* or *equisolid angle*. Each of these models propose a different non-linear function to fit the distortion of the projecting rays of different lens configurations and geometries. In our case, we aim to cover a wider set of projection models, so we use empirical models, which are more flexible to different or unknown projection models.

In the field of empirical models for camera calibration, two rise above the others: Scaramuzza's [25] and Kannala-Brandt's [18]. Both models propose a high-order polynomial function to model the camera distortion. In this work, we use the second one, Kannala-Brandt's [18], since it is the most extended in the calibration of commercial devices.

The full Kannala-Brandt model is explained in [18], where they present a radially symmetric model and a full model were the radial asymmetry of lenses is taken into account. For

this work we use the first one, assuming that the radial asymmetry error is much lower than the pixel precision that can be obtained from an image. So, assuming a radially symmetric model, [18] define the forward projection model as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = d(\theta) \begin{pmatrix} f_x \cos\varphi \\ f_y \sin\varphi \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix} \tag{1}$$

where (u,v) are the pixel coordinates in the image, (c_x,c_y) are the pixel coordinates of the optical center, (f_x,f_y) is the focal length on each axis, (θ,φ) are the spherical coordinates (see Fig. 1 (b)) of the incoming ray, $d(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^9$ is the high order polynomial function and $[k_1,k_2,k_3,k_4]$ the Kannala-Brandt calibration parameters.

The back projection model is computed as:

$$\varphi = \arctan\frac{m_y}{m_x}; \quad \theta = d^{-1}\left(\sqrt{m_x^2+m_y^2}\right), \tag{2}$$

where $m_x = (u-c_x)/f_x$, $m_y = (v-c_y)/f_y$ and d^{-1} is computed iteratively.

3.2 Fisheye calibrated kernel

Deformable convolutions were presented in [6], where the authors propose a method to learn offsets in the kernels for a better adaptation of the CNN to the task at hand. On the other hand, using the calibration of perspective cameras on CNNs has also been addressed by [9], obtaining improvements in the performance of the network. In this section, we present our implementation of a camera-calibrated kernel for non-linear projection models, adapting the kernel to the fisheye distortion of the Kannala-Brandt projection model.

Let $\mathcal{K}$ be a $(k_i \times k_j)$ rectangular kernel where $(k_i,k_j) \geq 1$ are an odd number and (u_0,v_0) is the anchor pixel around which we will apply the convolution kernel. We define the coordinates of each element of $\mathcal{K}$ as: $\hat{p}_{ij} = (i,j,d)^T$ where i is in range $\left[-\frac{k_i-1}{2}, \frac{k_i-1}{2}\right]$; j is in range $\left[-\frac{k_j-1}{2}, \frac{k_j-1}{2}\right]$; and d is the focal distance of $\mathcal{K}$. We assume that the standard kernel has the same behaviour as in a perspective camera, so we compute the focal distance as a function of the field of view of the kernel, α, which we linearly map from the fisheye camera field of view, Φ, as: $d = \frac{k_i}{2\tan\frac{\alpha}{2}}$ where $\alpha = \frac{k_i}{W}\Phi$ where W is the size of the feature map.

We project each kernel point into the unit sphere surface by normalizing the vectors. Then, we want to go back to the pixel domain using the forward projection model of the camera, Eq. 1. However, the resolution of the feature maps of a network (almost) always differs from the input image of the network, which is the resolution of the calibration parameters of the camera. Besides, we have to align the anchor of the kernel with the pixel we want to apply the kernel.

To solve the multi-scale/multi-resolution problem, we compute an scaling factor for each resolution that relates the calibration resolutions with the current feature map. The scaling factor is defined as $s = \frac{W_c+p_w}{W_{FM}}$, where W_c is the camera resolution width, W_{FM} is the feature map width and p_w is the width of an additional padding to the input image. We use this padding to set a fixed input resolution to our network in case of different image resolutions. With the scaling factor, we re-compute the calibration parameters for the kernel as:

$$\begin{pmatrix} cx_k \\ cy_k \end{pmatrix} = \begin{pmatrix} c_x \frac{W_{FM}-p_w/s}{W_c} \\ c_y \frac{H_{FM}-p_h/s}{H_c} \end{pmatrix}; \quad \begin{pmatrix} fx_k \\ fy_k \end{pmatrix} = \begin{pmatrix} f_x \frac{W_{FM}-p_w/s}{W_c} \\ f_y \frac{H_{FM}-p_h/s}{H_c} \end{pmatrix}, \tag{3}$$

where $(cx_k, cy_k), (fx_k, fy_k), (W_{FM}, H_{FM})$ are the coordinates of the optical center, focal lengths and resolution of the feature map and (W_c, H_c) the resolution of the fisheye camera.

Once the calibration parameters are adjusted to the feature map resolution, we compute the projecting ray of the anchor pixel (u_0, v_0) using Eq.2 and rotate the projecting rays of $\mathcal{K}$ to meet the orientation. With the $\mathcal{K}$ in the correct position, we project again each element of the kernel into the fisheye plane with Eq.1, obtaining the new locations of the kernel in the fisheye image (or feature map). From this implementation, we obtain a convolution kernel that adapts its shape with the distortion of the camera following the radially symmetric projection model (see Fig. 1).

4 Experiments

For the experimental part, we evaluate the calibrated convolutions against the standard convolutions on fisheye cameras. For that purpose, we use a well known CNN, U-Net [24], on two different tasks to evaluate the performance of the proposed kernels. We want to avoid current architectures where convolutions are mixed with other components as recurrent blocks [14] or attention mechanisms [30] to evaluate only the impact of the convolutions in the overall performance. For a fair comparison between convolutions, we propose the following set-up for the experiments: 1) We train the CNN, with standard convolutions, on perspective images of the Stanford dataset [1] and use these weights as baseline; 2) We evaluate the network with standard convolutions and calibrated convolutions on fisheye images obtained from the Stanford dataset [1]; 3) With the baseline as pretrained weights, we fine tune the network with fisheye images in two different situations: one fine tune is made with standard convolutions and other with the proposed calibrated convolutions; 4) After fine tuning, we evaluate again both networks on the fisheye images. To extend our comparison, we also fine tune and evaluate the network with standard convolutions on rectified images (from fisheye to perspective), an alternative approach only applicable when field of view is <180º. More details of the rectification process and full experiment is available in the supplementary material.

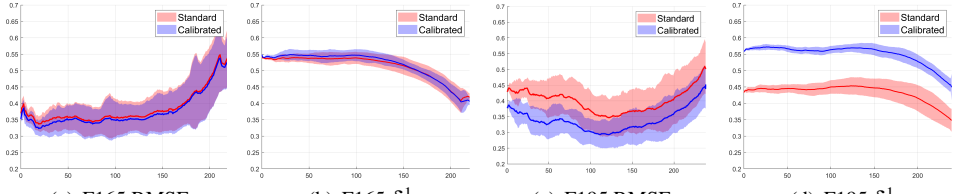
Training and fine tuning is made in the Stanford dataset following the #1 folder split, taking the *Area 5* only for evaluation and the others as training and validation sets. Perspective images are taken from the original dataset, where we find around 70k images with depth and semantic information. For simplicity, we define this dataset as *Pers* in the experiments. Fisheye images are randomly synthesized from the panoramic dataset in different orientations and with known calibration. We have generated 11.2k images for two different fisheye calibrations (i.e. 5.6k images of each calibration). For simplicity, we define these datasets by the field of view of the camera, such as: *F165* defines the dataset of fisheye images with a field of view of 165° and *F195* with a field of view of 195°.

4.1 Monocular depth estimation

The first task we evaluate is monocular depth estimation from single images. The convolutional network has been trained for 50 epochs on the Stanford dataset with perspective images, which has taken around 100 hours to complete. This network is our baseline (BL) for the experiment. Then, the fine tuning (FT) has been made on top of this training for less than 10% of the training time. The network has been fine tuned for 20 epochs in the

	Dataset	Kernel	MRE ↓	MAE ↓	RMSE ↓	$RMSE_{log}$ ↓	δ^1 ↑	δ^2 ↑	δ^3 ↑
BL	*Pers*	Standard	0.1538	0.2462	0.4908	0.1146	0.6391	0.8678	0.9407
	F195	Standard	0.2726	0.3914	0.4943	0.2408	0.4186	0.7027	0.8517
		Calibrated	0.2922	0.4133	0.5262	0.2604	0.3839	0.6745	0.8371
	F165	Standard	0.2670	0.3790	0.5005	0.2324	0.4377	0.7198	0.8579
		Calibrated	0.2798	0.3971	0.5216	0.2441	0.4019	0.6998	0.8500
		Rectified	0.8595	0.6412	0.9714	0.4178	0.3000	0.5509	0.7305
FT	*F195*	Standard	0.2432	0.3729	0.4023	0.2022	0.4241	0.7277	0.8827
		Calibrated	**0.2017**	**0.3159**	**0.3418**	**0.1575**	**0.5450**	**0.7972**	**0.9075**
	F165	Standard	0.2508	0.3582	0.4040	0.1899	0.4962	0.7628	0.8879
		Calibrated	**0.2505**	**0.3561**	**0.3875**	**0.1865**	**0.4992**	**0.7648**	**0.8884**
		Rectified	0.7758	0.5999	0.8933	0.3661	0.3016	0.5710	0.7618

Table 1: Monocular depth estimation with standard and calibrated convolutions for U-Net neural network. BL: Base Line; FT: Fine Tuned. Best metric for each fisheye calibration is in bold.



(a) F165 RMSE (b) F165 δ^1 (c) F195 RMSE (d) F195 δ^1

Figure 2: Comparison and results of depth estimation with U-Net neural network with standard (red) and calibrated (blue) convolutions. The x-axis defines the distance of the pixels to the optical center and the y-axis the computed error, defined as mean and one standard deviation.

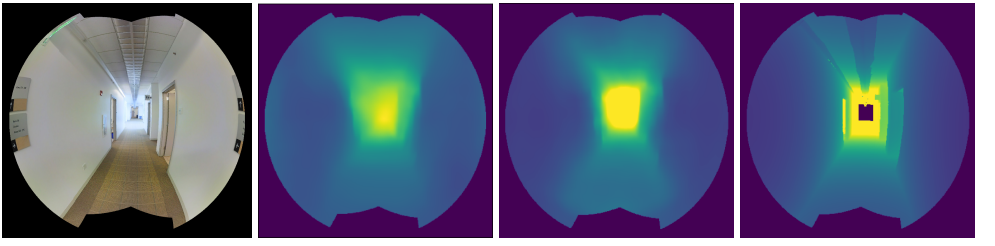
fisheye dataset, taking between 2-4 hours. Fine-tuning time changes with the resolution of the fisheye images, being different for each field of view.

Results of this experiment are shown in Tab. 1. We use the standard metrics for depth estimation presented in [32]. We also compute the metrics with respect the distance of each pixel to the principal point of the camera (i.e. $d(\theta)$ from equation 1) to observe the behaviour of each convolution with the increasing distortion of the image. These results are presented in Fig. 2 for both fisheye datasets. Additionally, we present qualitative results of monocular depth estimation in Fig. 3 and a 3D reconstruction on Fig. 4.

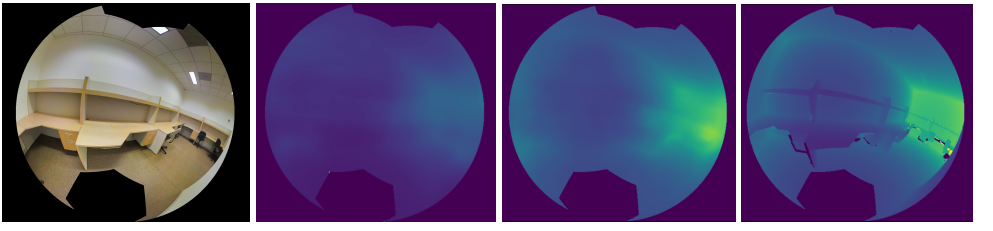
4.2 Semantic Segmentation

The second task that we evaluate with U-Net [24] is semantic segmentation. We use the same set-up and dataset than in the previous experiment. We train the network for 50 epochs, taking 75 hours to train, and then fine tune it 20 more epochs, taking between 5-7 more hours.

Results of this experiment are shown in Tab. 2. The metrics used are the mean Intersection over Union (mIoU) and the mean Accuracy (mAcc) over all the classes, except the unknown class. We also compute the metrics with respect the distance of each pixel to the principal point of the camera to evaluate the behaviour of the network with the increasing distortion of the image. These results are presented in Fig. 5 and qualitative results of semantic segmentation are presented in Fig. 6.



(a) F165 RGB (b) F165 Standard (c) F165 Calibrated (d) F165 GT



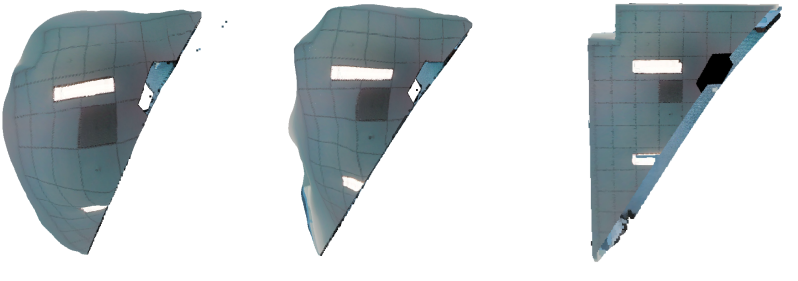
(e) F195 RGB (f) F195 Standard (g) F195 Calibrated (h) F195 GT

Figure 3: Qualitative results of monocular depth estimation on different fisheye calibrations. Distance is in a color scale, from colder colors (closer distances) to warmer colors (farther distances).

5 Discussion

The experiments and results presented show that the transfer learning problem is still an open topic and difficult to achieve in the conversion from perspective to omnidirectional images. Both monocular depth estimation and semantic segmentation results present a great decrease of performance with the baseline weights of the network, being in some cases more accentuated when we use calibrated convolutions. However, after a short fine tune of the network, these results improve significantly, particularly with calibrated convolutions.

From the results of depth estimation, we observe that the fine-tuned networks have slightly worse performance than the baseline with perspective images. The main difference

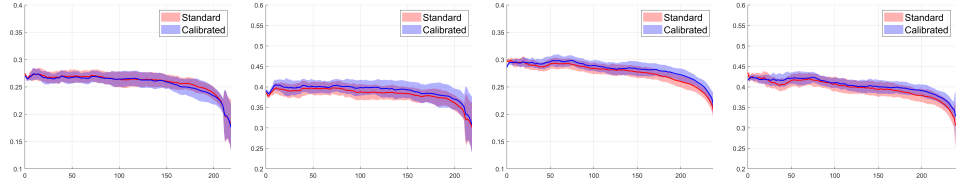


(a) Standard Conv (b) Calibrated Conv (c) Ground Truth

Figure 4: Qualitative results of depth estimation for FOV of 195º, top view of a 3D point cloud generated from depth data.

	Dataset	Kernel	mIoU	mAcc
BL	*Pers*	Standard	33.32	42.35
	F195	Standard	15.05	24.41
		Calibrated	13.73	22.55
	F165	Standard	15.12	24.36
		Calibrated	13.23	21.46
		Rectified	11.81	19.82
FT	*F195*	Standard	29.48	43.40
		Calibrated	**30.90**	**46.90**
	F165	Standard	27.70	36.51
		Calibrated	**27.89**	**36.71**
		Rectified	21.99	29.61

Table 2: Semantic segmentation with standard and calibrated convolutions for U-Net neural network. BL: Base Line; FT: Fine Tuned. Best metric for each fisheye calibration is in bold.



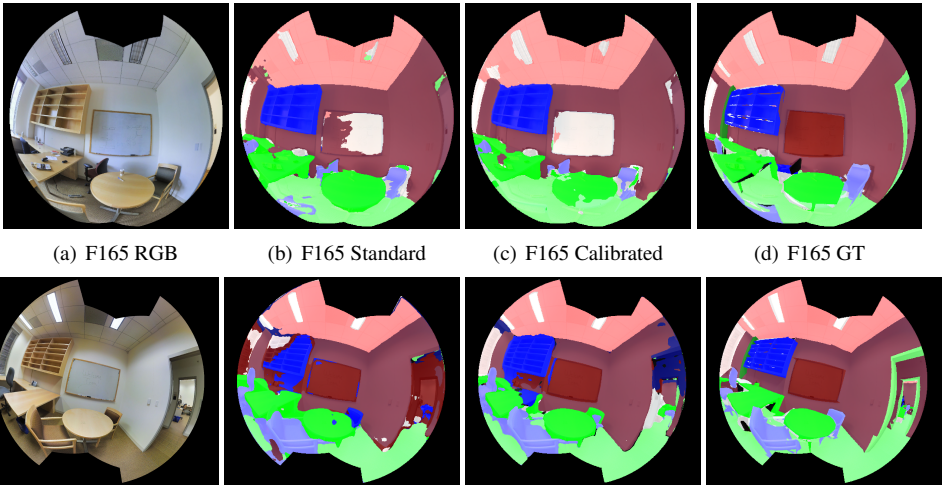
(a) F165 mIoU (b) F165 mAcc (c) F195 mIoU (d) F195 mAcc

Figure 5: Comparison and results of semantic segmentation with the U-Net like network with standard (red) and calibrated (blue) convolutions. The x-axis defines the distance of the pixels to the optical center and the y-axis the computed error, defined as mean and one standard deviation.

is that with the fisheye images we cover a wider field of view, obtaining more information of the scene with the same number of images. In the comparison of convolutions, we observe that with wider fields of view, the calibrated convolutional kernels provide better performance, while with the smaller field of view, the performance is quite similar. However, when we make a deeper analysis of these results, in the Figure 2 we observe the error distribution of the standard and calibrated convolutions. These results show that the estimation of the calibrated convolutions is more precise, with less dispersed error than the results of standard convolutions. Even if they are also affected by the increasing distortion of the fisheye images, the prediction is closer to the average error.

Regarding semantic segmentation experiments, the quantitative results from Tab. 2 show that the performance with fisheye images decreases significantly from the perspective image case. This is to be expected, since the segmentation problem difficulty increases with the field of view, including more objects to segment in the same image, and distortion, changing the appearance of the same object in different locations in the image. However we mitigate the second problem with the calibrated convolutions, obtaining better results with our proposal than with standard convolutions consistently along the radius in most metrics, particularly in larger radius (see Fig. 5). In addition, the qualitative results from Fig. 6 show significant differences in the performance between standard and calibrated convolutions. We can observe how the boundaries of objects and some details are better obtained with the calibrated convolutions than with the standard ones.

These results and conclusions led us to believe that calibrated convolutions provide a faster domain adaptation of CNNs, that means, in the same training conditions with limited



(a) F165 RGB (b) F165 Standard (c) F165 Calibrated (d) F165 GT

(e) F195 RGB (f) F195 Standard (g) F195 Calibrated (h) F195 GT

Figure 6: Qualitative results of semantic segmentation on different fisheye calibrations. Each color represent a different class from the dataset.

data, the calibrated convolutions provide better performance that the standard ones. The adaptation of networks trained on perspective images can be done with small datasets of fisheye images, achieving similar performance.

6 Conclusion

In this article we have presented a novel implementation of deformable convolutional kernels taking into account the intrinsic calibration of fisheye cameras. Integrating the Kannala-Brandt projection model for revolution symmetry cameras in the kernel of convolutional neural networks, we obtain a domain adaptation mechanism to take advantage of previous works on perspective images and adapt these networks to work with fisheye cameras. On a similar approach, this work could also be extended to other projection models that take into account the calibration of omnidirectional cameras, such as the Scaramuzza's model [25].

Results of the performed experiments show that the calibrated convolutions perform better than standard convolutions for domain adaptation. Besides, the impossibility of rectifying omnidirectional images of more than 180 degrees of field of view and the poor results obtained on the rectified ones increases the interest in studying how to adapt current deep learning methods to omnidirectional devices as the fisheye cameras.

A comparison with other methods, as CAM-Convs [9], is not trivial. These works should be extended to other projection models, since currently only work on the pin-hole camera model. With a naive approach, including directly the Kannala-Brandt model to the CAM-Convs proposal, we observe abrupt changes in the feature maps, which make us believe that further research is needed. This new approach, the extension of methods as CAM-Convs to omnidirectional images, remains as future work.

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[2] Charles-Olivier Artizzu, Guillaume Allibert, and Cédric Demonceaux. Omni-conv: Generalization of the omnidirectional distortion-aware convolutions. *Journal of Imaging*, 9(2):29, 2023.

[3] Iljoo Baek, Albert Davies, Geng Yan, and Ragunathan Raj Rajkumar. Real-time detection, tracking, and classification of moving and stationary objects using multiple fisheye images. In *Intelligent vehicles symposium*, pages 447–452. IEEE, 2018.

[4] Bruno Berenguel-Baeta, Jesus Bermudez-Cameo, and Jose J. Guerrero. Fredsnet: Joint monocular depth and semantic segmentation with fast fourier convolutions from single panoramas. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6080–6086, 2023. doi: 10.1109/ICRA48891.2023.10161142.

[5] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018.

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the International Conference on Computer Vision*, pages 764–773. IEEE, 2017.

[7] Liuyuan Deng, Ming Yang, Yeqiang Qian, Chunxiang Wang, and Bing Wang. Cnn based semantic segmentation for urban traffic scenes using fisheye camera. In *Intelligent Vehicles Symposium*, pages 231–236. IEEE, 2017.

[8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[9] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Cam-convs: Camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 11826–11835. IEEE/CVF, 2019.

[10] Clara Fernandez-Labrador, Jose M Facil, Alejandro Perez-Yus, Cedric Demonceaux, Javier Civera, and Jose J Guerrero. Corners for layout: End-to-end layout recovery from 360 images. *Robotics and Automation Letters*, pages 1255–1262, 2020.

[11] Shaohua Gao, Kailun Yang, Hao Shi, Kaiwei Wang, and Jian Bai. Review on panoramic imaging and its applications in scene understanding. *Transactions on Instrumentation and Measurement*, 71:1–34, 2022.

[12] Julia Guerrero-Viu, Clara Fernandez-Labrador, Cédric Demonceaux, and Jose J Guerrero. What's in my room? object recognition on indoor panoramic images. In *International Conference on Robotics and Automation*, pages 567–573. IEEE, 2020.

[13] Olfa Haggui, Hamza Bayd, Baptiste Magnier, and Arezki Aberkane. Human detection in moving fisheye camera using an improved yolov3 framework. In *International Workshop on Multimedia Signal Processing*, pages 1–6. IEEE, 2021.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 4201–4209. IEEE/CVF, 2017.

[16] Chiyu Jiang, Jingwei Huang, Karthik Kashinath, Philip Marcus, Matthias Niessner, et al. Spherical cnns on unstructured grids. *arXiv preprint arXiv:1901.02039*, 2019.

[17] Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. Unifuse: Unidirectional fusion for 360 panorama depth estimation. *Robotics and Automation Letters*, 6 (2):1519–1526, 2021.

[18] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *Transactions on Pattern Analysis and Machine Intelligence*, pages 1335–1340, 2006.

[19] Varun Ravi Kumar, Stefan Milz, Christian Witt, Martin Simon, Karl Amende, Johannes Petzold, Senthil Yogamani, and Timo Pech. Monocular fisheye camera depth estimation using sparse lidar supervision. In *International Conference on Intelligent Transportation Systems*, pages 2853–2858. IEEE, 2018.

[20] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2801–2810. IEEE/CVF, 2022.

[21] Ming Meng, Likai Xiao, Yi Zhou, Zhaoxin Li, and Zhong Zhou. Distortion-aware room layout estimation from a single fisheye image. In *International Symposium on Mixed and Augmented Reality*, pages 441–449. IEEE, 2021.

[22] Dejing Ni, Peng Ji, and Aiguo Song. Vanishing point detection in corridor for autonomous mobile robots using monocular low-resolution fisheye vision. *Advances in Mechanical Engineering*, 11(10):1687814019884767, 2019.

[23] Alejandro Perez-Yus, Gonzalo Lopez-Nicolas, and Jose J Guerrero. Scaled layout recovery with wide field of view rgb-d. *Image and Vision Computing*, 87:76–96, 2019.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[25] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *International Conference on Computer Vision Systems*, pages 45–45. IEEE, 2006.

[26] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition workshops*, pages 806–813. IEEE/CVF, 2014.

[27] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360 imagery. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision*, pages 707–722. Springer, 2018.

[29] Marin Toromanoff, Emilie Wirbel, Frédéric Wilhelm, Camilo Vejarano, Xavier Perrotton, and Fabien Moutarde. End to end vehicle lateral control using a single fisheye camera. In *International Conference on Intelligent Robots and Systems*, pages 3613–3619. IEEE/RSJ, 2018.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems*, 2017.

[31] Chuanqing Zhuang, Zhengda Lu, Yiqun Wang, Jun Xiao, and Ying Wang. Acdnet: Adaptively combined dilated convolution for monocular panorama depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3653–3661, 2022.

[32] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *European Conference on Computer Vision*, pages 448–465. Springer, 2018.

[33] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, Federico Alvarez, and Petros Daras. Spherical view synthesis for self-supervised 360 depth estimation. In *International Conference on 3D Vision*, pages 690–699. IEEE, 2019.