



# Towards a Unified Benchmark for Monocular Radial Distortion Correction and the Importance of Testing on Real-World Data

Christoph Theiß<sup>1,2</sup>(✉) and Joachim Denzler<sup>1,2</sup>(✉)

<sup>1</sup> Friedrich Schiller University Jena, 00743 Jena, Germany  
{christoph.theiss,joachim.denzler}@uni-jena.de

<sup>2</sup> German Aerospace Center, Institute of Data Science, 07745 Jena, Germany

**Abstract.** Radial distortion correction for a single image is often overlooked in computer vision. It is possible to rectify images accurately when the camera and lens are known or physically available to take additional images with a calibration pattern. However, sometimes it is impossible to identify the camera or lens of an image, e.g., crowd-sourced datasets. Nonetheless, it is still important to correct that image for radial distortion in these cases. Especially in the last few years, solving the radial distortion correction problem from a single image with a deep neural network approach increased in popularity. This paper shows that these approaches tend to overfit completely on the synthetic data generation process used to train such networks. Additionally, we investigate which parts of this process are responsible for overfitting. We apply an explainability tool to analyze the trained models' behavior. Furthermore, we introduce a new dataset based on the popular ImageNet dataset as a new benchmark for comparison. Lastly, we propose an efficient solution to the overfitting problem by feeding edge images to the neural networks instead of the images. Source code, data, and models are publicly available at <https://github.com/cvjena/deeprect>.

**Keywords:** Radial distortion · Monocular images · Synthetic data

## 1 Introduction

The effects of lens distortion are often overlooked when knowledge about the geometry of 3D scenes or the pinhole camera model is integrated into deep learning-based approaches [10, 20, 31, 35]. Nevertheless, their effects are still visible on images taken with modern cameras, including mobile devices. In particular, wide-angle lenses, which are widely used due to their large field of view, suffer from geometric distortions. Additional difficulties arise when images are taken under uncontrolled conditions, e.g., crowd-sourcing or web-crawling scenarios. In these cases, conventional automatic algorithms for correcting radial distortions cannot be applied, and one would have to try correcting the image manually.

However, state of the art for radial distortion correction from a single image has improved dramatically recently with the ever-increasing success of machine learning and deep learning-based approaches [13, 14, 18, 22, 30, 33]. While these methods achieve remarkable performance on benchmark datasets, they fail to estimate the radial distortion for real-world data correctly.

We show that the generalization to real-world data can be improved when edge detections of the images are fed to the network instead of the distorted images themselves. This way, information that is unnecessary for the radial distortion correction is removed, reducing the complexity of the search space and removing unintentional artifacts. Moreover, we demonstrate the influence of high-resolution images and that a combination of edge detections and high-resolution images improves the generalization even further. For that, we investigate the behavior of three previously published methods on a new high-resolution dataset, which is a subset of the ImageNet dataset [23]. Beyond that, we analyze the behavior of a classification-based approach with an explainability tool. This way, we can study which input areas the models focus on for their prediction.

## 2 Related Work

Many approaches exist for radial distortion correction with the help of a calibration pattern, video sequences, or when the camera is known and physically available. However, in many cases, none of these pieces of information are available. Hence, we focus on correcting radial distortion using a single image and describe more recent approaches that apply deep neural networks to solve this task.

Previous works on radial distortion correction from a single image focus almost exclusively on barrel distortion and omit pincushion or mustache distortion, which requires the estimation of at least two distortion coefficients. Rong *et al.* [22] cast radial distortion correction as a classification problem where each class corresponds to a radial distortion coefficient. This significantly limits the distortions they can estimate. Lutz *et al.* [18] proposed a data-driven approach in which a network is trained on a large set of synthetically distorted images. As common in regression tasks, they minimize the mean squared error between prediction and ground truth distortion coefficients. Shi *et al.* [24] extend a ResNet-18 model [12] by adding a weight layer with so-called inverted foveal models after the final convolutional layer to emphasize the importance of pixels closer to the border of the images. In addition to the distortion coefficients, Li *et al.* [13] estimate the displacement field between distorted and corrected images and use it to generate the corrected image.

An approach to estimate more than a single distortion parameter is presented by López-Antequera *et al.* [16], which simultaneously predicts tilt, roll, focal length, and radial distortion. They recover estimations for two radial distortion parameters from a large dataset with the help of Structure from Motion [27] reconstruction. They show that the parameters of the cameras used to acquire the dataset lie close to a one-dimensional manifold. While some cameras have this property, many do not, severely limiting the method’s applicability.

Instead of predicting the radial distortion coefficients, Liao *et al.* [14] use generative adversarial networks (GANs) [11] to generate the undistorted image directly. The reconstructed images lack texture details, are partially distorted or differently illuminated, among other problems.

Other works focus on correcting fisheye images. Xue *et al.* [29] exploit that distorted lines generated by the fisheye projection should be straight after correction, or similarly that straight lines in 3D space should also be straight in the image plane [28]. However, both approaches require line segment annotations that are not easily acquirable, significantly limiting the available training data.

Finally, the current state of the art for correcting barrel distortion with a single image is achieved by Zhao *et al.* [33] integrating additional knowledge about the radial distortion model. They use a CNN to estimate the radial distortion coefficient and the inverse of the division model to calculate a sampling grid to rectify the distorted image by bilinear sampling. As a standard radial distortion model, we describe the division model in Sect. 3.1.

### 3 Data Generation

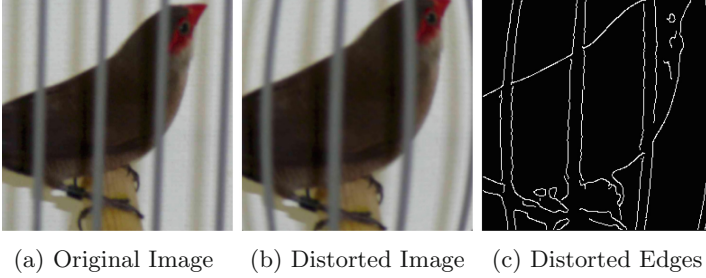
Ground truth data for radial distortion correction is difficult to obtain. Hence, previous work used synthetically distorted images for training. Following, we describe the division model [9], which is often used to obtain distorted images. Afterward, we describe what datasets are used for that purpose and why they are unsuitable for radial distortion correction. Instead, we propose to use a subset of the ImageNet dataset that only contains high-resolution images. Lastly, we describe the proposed preprocessing strategy applied after the synthetic distortion of the input images.

#### 3.1 Lens Distortion Models

A common way to model radial distortion is the division model

$$r^u = \frac{r^d}{1 + k_1(r^d)^2 + k_2(r^d)^4 + \dots} \quad (1)$$

w.r.t the radius  $r^d = \sqrt{(\bar{x}^d)^2 + (\bar{y}^d)^2}$ , proposed by [9], where  $\bar{x}^d = x^d - x^0$ ,  $\bar{y}^d = y^d - y^0$ , and the coordinates of the distorted point  $(x^d, y^d)$ . The distortion center is  $(x^0, y^0)$ . In practice, the model is parameterized with up to three parameters  $k_1, k_2, k_3$ . It is also common that models for specific cameras only have one parameter. Barrel distortion occurs when all coefficients  $k_1, k_2, \dots$  are negative and a pincushion distortion if all are positive. Mustache distortion can occur when the signs of the coefficients differ. It generally requires fewer coefficients to approximate large distortion compared to the radial model [4]. We assume that the center of distortion is the center of the image, and refer the interested reader to [26] for a more detailed comparison of various camera distortion models. A backward image warping is performed, which requires calculating the undistorted radius given the distorted radius when we distort an image.



**Fig. 1.** Example of the different types of input that are fed to the model. We compare the performance of models trained with images, and edge detection.

### 3.2 Preprocessing Strategy

We consider two input types to the neural network, as depicted in Fig. 1. As a baseline, we use natural images as is done in all previous deep neural network approaches for radial distortion correction. We show that models trained on synthetically distorted natural images overfit the data generation process and do not generalize to real-world data. Instead, we perform edge detection as the last preprocessing step with the Canny edge detector [6]. Previous non-parametric approaches [1, 5, 7, 8, 25, 32] demonstrate that edge detection is well suited as a preprocessing step to estimate radial distortion. They provide enough context in the image to correctly estimate the coefficients while simultaneously reducing the amount of data. In the case of edge detections, the resulting inputs only have one channel. We adjust the first layer of the network accordingly.

### 3.3 Synthetically Distorted Datasets

In the literature [13, 14, 18, 22, 33], the model in Eq. (1) is often used to generate distorted images where the rectified images are assumed to be given as ground truth.

Previous work uses already available datasets to collect sufficient training data and synthetically distorts them. To that end, Rong *et al.* [22] obtain a subset of the ImageNet dataset [23] with images that have a large amount of long straight lines relative to the image size. However, many images of the resulting dataset, counting roughly 68,000, have undesirable properties, like tiny images or images not taken with a camera like, *e.g.*, screenshots. Additionally, all images of this dataset are resized to a resolution of  $256 \times 256$  before conducting the synthetic radial distortion. In comparison, Li *et al.* [13] use the Places365-Standard dataset [34]. They use images with a resolution of  $512 \times 512$  as the original non-distorted images to generate a dataset of distorted images at a resolution of  $256 \times 256$ . Similarly, Lutz *et al.* [18] use the MS COCO dataset [15] but resize all images to 1024 on their longer side before distortion.

In contrast, we focus on high-resolution images. Like Rong *et al.* [22], we construct an ImageNet subset containing 126,623 the original 14.2 million images. However, we choose all images with a minimum width and height of at least 1200 pixels independent of straight lines. This results in a more diverse dataset with a vastly larger average resolution of  $2011 \times 1733$  than other datasets like MS COCO with  $577 \times 484$  or Places365-Standard with  $677 \times 551$ . We can leverage the high resolution to generate more realistically looking distortions because the resulting images appear significantly less blurry. We split the dataset into 106,582 training, 5,000 validation, and 10,000 test samples.

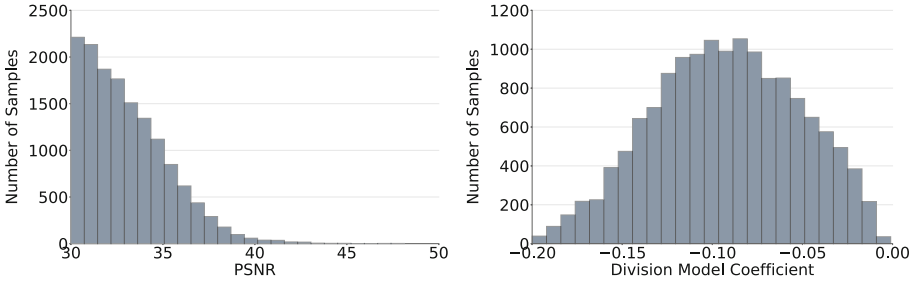
To reduce the possibilities for the network to exploit the data generation process, we propose increasing the resolution at which the images are distorted and applying an edge detection preprocessing step. This way, we can decrease the possibility for artifacts that are not visible by a human observer but force the network to overfit the synthetic data generation process instead of the actual radial distortion. We leverage the high-resolution images available in the ImageNet dataset to distort the images. This makes it possible to distort the images at a resolution of  $1024 \times 1024$  compared to  $512 \times 512$  or  $256 \times 256$  as is done in other work, which results in sharper, less blurry distorted images that more closely resemble the original undistorted image.

Besides evaluating the trained models qualitatively on real-world images, we also evaluate them quantitatively on a subset of a differently distorted dataset. For that, we use the dataset published by Bogdan *et al.* [3], who automatically generate distorted images based on panorama images and the unified spherical model [2]. Compared to Brown’s and division models used in other work, the unified spherical model can describe a wider variety of cameras, including perspective, wide-angle, fisheye, and catadioptric cameras. Hence, we need to determine the subset of distorted images that the division model can correct. We first compute the corrected image for each sample in the dataset with the unified spherical model. Afterward, we sample coefficients for the division model between 0.0 and  $-0.2$  to find the visually most similar-looking correction. We choose the subset of all images for which the peak signal-to-noise ratio (PSNR) between the two corrections is larger than 30, i.e., they are visually indistinguishable. Figure 2 shows the distribution of the model coefficients and PSNR of the resulting test dataset.

## 4 Experiments

### 4.1 Implementation Details

For the model, we follow previous works and choose a ResNet-18 [12] for all experiments. The models are initialized with ImageNet [23] weights. To avoid the unnecessary introduction of distortion, we resize all images while maintaining the aspect ratio. Then, the images are randomly cropped for training and center cropped for testing. We apply random horizontal and vertical flipping. Afterward, we randomly distort the images with the division model and distortion coefficients uniformly drawn between 0.0 and  $-0.2$ . We train each model



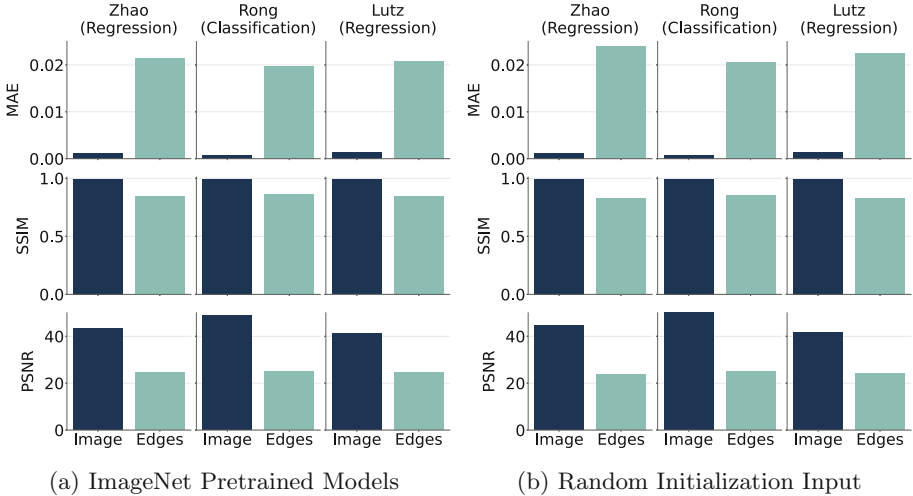
**Fig. 2.** PSNR and division model coefficient distributions for the subset of samples that are distorted by the unified spherical mode and can be corrected with the division model.

for 30 epochs and divide the learning rate by a factor of ten after 20 epochs. We used the AdamW optimizer [17] with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . The experiments are implemented with PyTorch [19].

Following previous works in this area [29, 30, 33], we adapt the PSNR and the structural similarity index measure (SSIM) as evaluation metrics. A larger value corresponds to better correction quality. Additionally, we report the mean absolute error (MAE) between estimated and ground-truth coefficients.

## 4.2 Radial Distortion Estimation

We analyze different input types and their impact on the model’s performance. We compare the models quantitatively on synthetically distorted validation data and qualitative on real-world images. The real-world samples are not synthetically distorted. Hence, they can give insight into whether the models learned to correct general radial distortion or overfitted to our data generation process. The quantitative results are shown in Fig. 3a. The results indicate that natural images without any further preprocessing outperform our preprocessing strategy on a synthetically distorted test dataset. However, the qualitative results on real-world images in Fig. 5 demonstrate that the models trained on natural images do not generalize to natural distortions. While models trained on edge images correctly predict a larger distortions coefficient when a severe distortion is visible in the image and do not overcorrect images with only a small visible distortion. This suggests that models trained on edge images can differentiate between severe and slight distortions.

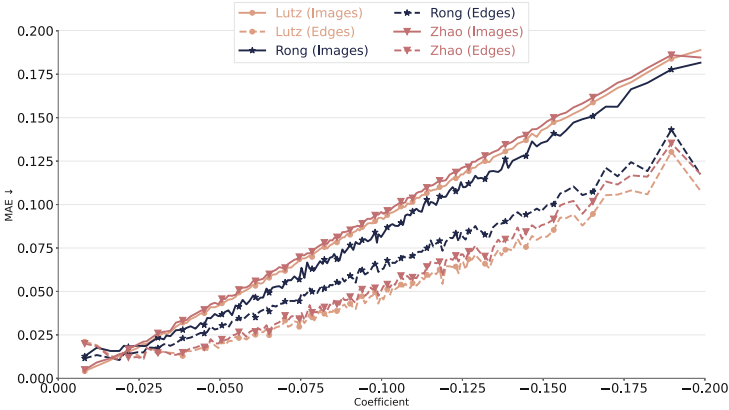


**Fig. 3.** Comparison between models trained on images and edge detections as inputs. It shows that natural images without any further preprocessing outperform the edges on a synthetically distorted test dataset. Moreover, the results are almost identical between pre-trained and randomly initialized weights.

Besides the qualitative evaluation of the images shown in Fig. 5, we quantitatively test the trained models on a subset of the dataset proposed by Bogdan *et al.* [3] that are distorted with the unified spherical model and can be corrected with the division. Because the coefficients are not uniformly distributed, we put them evenly sized bins of 100 based on the ground truth distortion coefficient and report the MAE for each group separately, as depicted in Fig. 4. The models trained on natural images cannot detect any distortion at all. Hence, they consistently predict a coefficient close to 0.0 resulting in an MAE almost identical to the ground truth distortion coefficient. On the other hand, models trained on edge detections have a significantly lower MAE.

### 4.3 Influence of Pre-trained Weights

All trained networks in Sect. 4.2 all use pre-trained weights wherever possible. The weights are obtained by solving a classification task on the original ImageNet dataset. To train the neural networks to estimate the radial distortion coefficients, we use a subset of the ImageNet dataset containing only high-resolution images. Hence, it is reasonable to assume that the weights are a significant cause of the observed overfitting phenomenon.



**Fig. 4.** MAE grouped by ground truth coefficients. While the MAE increases for both types of models with increasing distortion coefficients, models trained on edge detections are clearly more accurate.

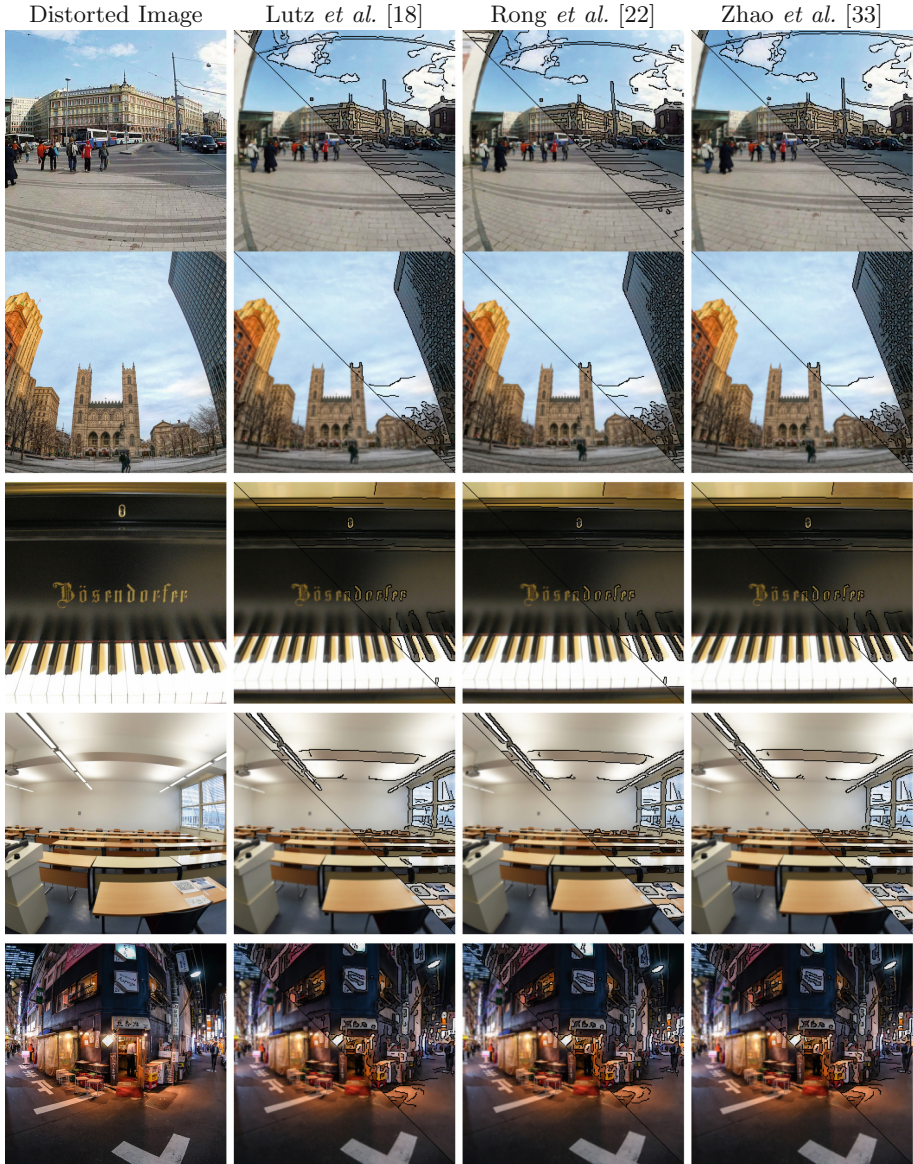
One possible explanation for the improved generalization when we train a network with edge images is the random initialization of weights for the first convolutional layer. Instead, the layer is trained from scratch because no weights are readily available for single-channel inputs with the same model architecture.

We confirm that the behavior is independent of the initialized weights by repeating the same experiment outlined above with randomly initialized weights. The results in Fig. 3b support that choosing weights pre-trained for the classification task on ImageNet is not the cause of the significant overfitting that can be observed for the radial distortion correction task, as can be seen in the negligible difference between Figs. 3a and 3b.

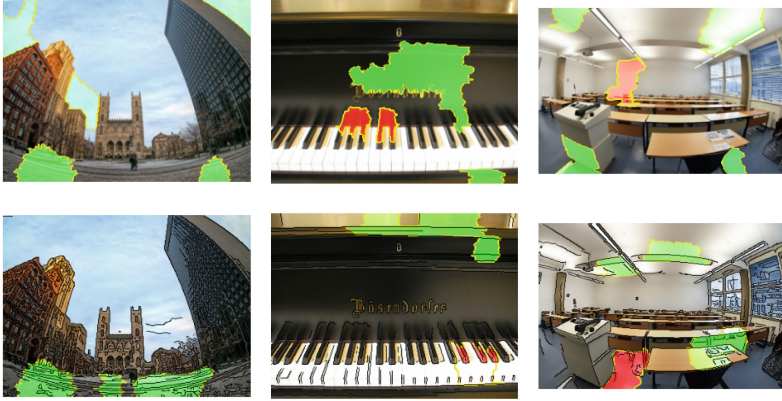
#### 4.4 Explainability of the Results

To better understand the behavior of our models, we apply local interpretable model-agnostic explanations (LIME) [21] to highlight areas of images that the models primarily use for their prediction. While applying this method to regression models is possible, the resulting explanations are not interpretable enough to be helpful, as stated by the authors. Hence, we apply LIME only to models trained with the method proposed by Rong *et al.* [22] because they are the only ones who cast radial distortion correction as a classification problem in a deep learning scenario. For radial distortion correction, areas with long lines are of particular interest. The results shown in Fig. 6 indicate that models trained on natural images specifically focus on homogenous areas without lines. In contrast, models that leverage edge detections prioritize areas with long edges.





**Fig. 5.** Qualitative results on various real-world examples. The top right of each image shows the correction with models trained on edge detections with the edges overlaid for visualization purposes. The bottom left models trained on natural images. From left to right, it shows the distorted real-world images and the models trained with the following methods: Lutz *et al.* [18], Rong *et al.* [22], and Zhao *et al.* [33]



**Fig. 6.** LIME explanations for models trained with the method proposed by Rong *et al.* The top row shows explanations for a model trained on natural images and the bottom row for a model trained on edge detection. The edges depicted in the bottom row indicate the input of models trained on edge detections. The model trained on natural image focuses on homogenous areas of the image which do not contain information about the radial distortion. On the other hand, the model trained on edge detection focuses on long, distinct lines.

## 5 Conclusion

This work investigated the effect of edge detection as additional preprocessing on the generalizability of deep neural networks for radial distortion correction. We analyzed the performance and behavior of three different methods that leverage deep learning for radial distortion correction. We investigated the overfitting for neural networks when trained with synthetically distorted images. To that end, we explored the influence of pre-trained weights and the resolution at which the images are synthetically distorted. In addition, we proposed a new dataset of high-resolution images based on the popular ImageNet dataset.

Moreover, we applied LIME to explain the results of the classification-based approach, which showed that models trained on natural images focus on homogeneous regions of the image, which should not be relevant for the task of radial distortion correction. We showed qualitatively on real-world examples and quantitatively on differently distorted images that edge detection as an additional preprocessing step is an effective measure to improve the generalization of these methods.

## References

1. Alvarez, L., Gómez, L., Sendra, J.R.: An algebraic approach to lens distortion by line rectification. *J. Math. Imag. Vis.* **35**(1), 36–50 (2009). <https://doi.org/10.1007/s10851-009-0153-2>
2. Barreto, J.P.: A unifying geometric representation for central projection systems. *Comput. Vis. Image Underst.* **103**(3), 208–217 (2006)
3. Bogdan, O., Eckstein, V., Rameau, F., Bazin, J.C.: DeepCalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In: Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production, CVMP 2018, pp. 1–10. Association for Computing Machinery, New York (2018). <https://doi.org/10.1145/3278471.3278479>
4. Brown, D.C.: Decentering distortion of lenses. *Photogrammetric Engineering and Remote Sensing* (1966). <https://ci.nii.ac.jp/naid/10022411406/>
5. Bräuer-Burchardt, C., Voss, K.: Automatic lens distortion calibration using single views. In: Sommer, G., Krüger, N., Perwass, C. (eds.) *Mustererkennung 2000. Informatik aktuell*, pp. 187–194. Springer, Heidelberg (2000). [https://doi.org/10.1007/978-3-642-59802-9\\_24](https://doi.org/10.1007/978-3-642-59802-9_24)
6. Canny, J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 679–698 (1986)
7. Claus, D., Fitzgibbon, A.W.: A plumbline constraint for the rational function lens distortion model. In: Proceedings of the British Machine Vision Conference 2005, pp. 10.1-10.10. British Machine Vision Association, Oxford (2005). <https://doi.org/10.5244/C.19.10>
8. Devernay, F., Faugeras, O.: Straight lines have to be straight. *Mach. Vis. Appl.* **13**(1), 14–24 (2001). <https://doi.org/10.1007/PL00013269>
9. Fitzgibbon, A.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I (2001). <https://doi.org/10.1109/CVPR.2001.990465>, ISSN 1063-6919
10. Garg, R., B.G., V.K., Carneiro, G., Reid, I.: Unsupervised CNN for single view depth estimation: geometry to the rescue. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016. LNCS*, vol. 9912, pp. 740–756. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46484-8\\_45](https://doi.org/10.1007/978-3-319-46484-8_45)
11. Goodfellow, I., et al.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc. (2014)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition, pp. 770–778 (2016)
13. Li, X., Zhang, B., Sander, P.V., Liao, J.: Blind geometric distortion correction on images through deep learning, pp. 4855–4864 (2019)
14. Liao, K., Lin, C., Zhao, Y., Gabbouj, M.: DR-GAN: automatic radial distortion rectification using conditional GAN in real-time. *IEEE Trans. Circ. Syst. Video Technol.* **30**(3), 725–733 (2020). <https://doi.org/10.1109/TCSVT.2019.2897984>
15. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014. LNCS*, vol. 8693, pp. 740–755. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
16. Lopez, M., Mari, R., Gargallo, P., Kuang, Y., Gonzalez-Jimenez, J., Haro, G.: Deep single image camera calibration with radial distortion, pp. 11817–11825 (2019)
17. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) [cs, math] (2019). [arXiv: 1711.05101](https://arxiv.org/abs/1711.05101)

18. Lutz, S., Davey, M., Smolic, A.: Deep convolutional neural networks for estimating lens distortion parameters. Session 2: Deep Learning for Computer Vision (2019). <https://doi.org/10.21427/yg8t-6g48>
19. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc. (2019)
20. Ranjan, A., et al.: Competitive collaboration: joint unsupervised learning of depth, camera motion, optical flow and motion segmentation, pp. 12240–12249 (2019)
21. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should I trust you?”: explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 13–17 August 2016, pp. 1135–1144 (2016)
22. Rong, J., Huang, S., Shang, Z., Ying, X.: Radial lens distortion correction using convolutional neural networks trained with synthesized images. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (eds.) *ACCV 2016*. LNCS, vol. 10113, pp. 35–49. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-54187-7\\_3](https://doi.org/10.1007/978-3-319-54187-7_3)
23. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015). <https://doi.org/10.1007/s11263-015-0816-y>
24. Shi, Y., Zhang, D., Wen, J., Tong, X., Ying, X., Zha, H.: Radial lens distortion correction by adding a weight layer with inverted foveal models to convolutional neural networks. In: *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1–6 (2018). <https://doi.org/10.1109/ICPR.2018.8545218>, ISSN 1051-4651
25. Strand, R., Hayman, E.: Correcting radial distortion by circle fitting. In: *Proceedings of the British Machine Vision Conference 2005*, pp. 9.1–9.10. British Machine Vision Association, Oxford (2005). <https://doi.org/10.5244/C.19.9>
26. Tang, Z., Grompone von Gioi, R., Monasse, P., Morel, J.M.: A precision analysis of camera distortion models. *IEEE Trans. Image Process.* **26**(6), 2694–2704 (2017). <https://doi.org/10.1109/TIP.2017.2686001>
27. Ullman, S., Brenner, S.: The interpretation of structure from motion. *Proc. R. Soc. Lond. Ser. B. Biol. Sci.* **203**(1153), 405–426 (1979). <https://doi.org/10.1098/rspb.1979.0006>, <https://royalsocietypublishing.org/doi/abs/10.1098/rspb.1979.0006>
28. Xue, Z.C., Xue, N., Xia, G.S.: Fisheye distortion rectification from deep straight lines. [arXiv:2003.11386](https://arxiv.org/abs/2003.11386) [cs] (2020). [arXiv: 2003.11386](https://arxiv.org/abs/2003.11386)
29. Xue, Z., Xue, N., Xia, G.S., Shen, W.: Learning to calibrate straight lines for Fisheye image rectification, pp. 1643–1651 (2019)
30. Yin, X., Wang, X., Yu, J., Zhang, M., Fua, P., Tao, D.: FishEyeRecNet: a multi-context collaborative deep network for Fisheye image rectification, pp. 469–484 (2018)
31. Yin, Z., Shi, J.: GeoNet: unsupervised learning of dense depth, optical flow and camera pose, pp. 1983–1992 (2018)
32. Ying, X., Hu, Z., Zha, H.: Fisheye lenses calibration using straight-line spherical perspective projection constraint. In: *Narayanan, P.J., Nayar, S.K., Shum, H.-Y.* (eds.) *ACCV 2006*. LNCS, vol. 3852, pp. 61–70. Springer, Heidelberg (2006). [https://doi.org/10.1007/11612704\\_7](https://doi.org/10.1007/11612704_7)
33. Zhao, H., Shi, Y., Tong, X., Ying, X., Zha, H.: A simple yet effective pipeline for radial distortion correction. In: *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 878–882 (2020). <https://doi.org/10.1109/ICIP40778.2020.9191107>, ISSN 2381-8549

34. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(6), 1452–1464 (2018). <https://doi.org/10.1109/TPAMI.2017.2723009>
35. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised Learning of depth and ego-motion from video, pp. 1851–1858 (2017)