Semantic Segmentation of Fisheye Images

Gregor Blott[1,2][0000−0003−3329−0529]✉,
Masato Takami[1], and Christian Heipke[2][0000−0002−7007−9549]

[1] Computer Vision Research Lab, Robert Bosch GmbH, Hildesheim, Germany
firstname.surname@de.bosch.com
[2] Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany surname@ipi.uni-hannover.de

Abstract. Semantic segmentation of fisheye images (e.g., from action-cameras or smartphones) requires different training approaches and data than those of rectilinear images obtained using central projection. The shape of objects is distorted depending on the distance between the principal point and the object position in the image. Therefore, classical semantic segmentation approaches fall short in terms of performance compared to rectilinear data. A potential solution to this problem is the recording and annotation of a new dataset, however this is expensive and tedious. In this study, an alternative approach that modifies the augmentation stage of deep learning training to re-use rectilinear training data is presented. In this way we obtain a considerably higher semantic segmentation performance on the fisheye images: +18.3% intersection over union (IoU) for action-camera test images, +8.3% IoU for artificially generated fisheye data, and +18.0% IoU for challenging security scenes acquired in bird's eye view.

Keywords: Semantic Segmentation, Fisheye Images, Deep Learning

1 Introduction

Semantic segmentation (SemSeg) of images is a research topic of increasing interest. Several tasks, e.g. in the automotive domain [3], in action localization [13], person re-identification [19], background modeling [16], and remote sensing [18] address SemSeg in a pre-processing step before the actual domain-specific work is conducted. Various SemSeg approaches incl. those from the knowledge-based domain, graphical models, and machine learning have been published in recent years, see the survey [17]. A large and representative amount of training data is required before such approaches can be successfully applied to unseen data. Obtaining this required level of training data is expensive and tedious, since all images have to be annotated before conducting supervised learning.

We address a SemSeg procedure for ultra-wide-angle view images with fisheye (FE) effects. FE lenses are used in the automotive, robotic, consumer, and security domains to obtain a larger field of view with a single camera. Examples of corresponding sensors are action-cameras, recently published smartphones,

and security-cameras. As a drawback of using such lenses, rectilinearity in images is not maintained and the projection depends heavily on the lens design. In particular, the ratio of pixels per degree for equidistant fisheye projection is constant, whereas the ratio for central projection depends on the angle between the optical axis and the ray of the observed image point in space. Therefore, object shape in images obtained using FE lenses depends on the distance to the principal point and the position in the image. Consequently, training material, e.g., from an object located next to the principal point in image space will look different than the same object located next to the image border. It is thus not recommended for FE training to use rectilinear data since the model can then never learn the FE peculiarities, especially those towards the image border.

Our approach for FE SemSeg (cf. Figure 1) exploits the projection model underlying FE images and a publicly available dataset containing central projection images and annotations (MS COCO [10]) and transforms those images and annotations into FE geometry before training. Thus, the focus of our study is on obtaining rectilinear dataset performance for FE images without ever having seen an image actually captured in FE geometry during training and validation.

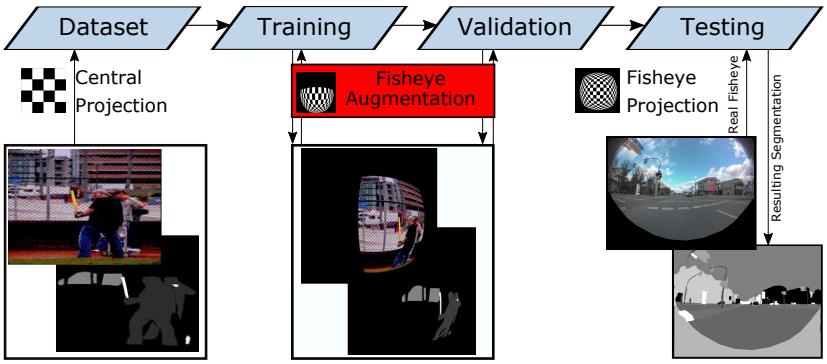


Fig. 1. Our approach for FE SemSeg. Images and annotations using central projection are augmented into FE images for training and validation with six degree of freedom. Real FE images are used only for testing. Credit central projection images [10].

The rest of the paper is structured as follows: work related to this study and the state of-the-art are discussed further in this section. The methods we developed are described in Section 2, and the results of the experiments are reported and discussed in Section 3. Finally, conclusions are provided in Section 4.

1.1 State-of-the-Art and Related Work

Three approaches can be designed for FE SemSeg. These are: (1) separate recording, annotation (labeling), and training with a FE dataset using well-known SemSeg approaches. However, this task is expensive and tedious, and is therefore not

addressed in this paper. (2) Pre-training of a classifier (typically a deep learning architecture would be employed today) on rectilinear data and re-training of the last layers on a diverse amount of FE images and annotations. In this approach, the effort related to generating data and creating annotations is much lower but it is still tangible for the purpose of this study. (3) Although generic in nature, a FE image can be re-projected (de-warped) into a rectilinear view, resulting in an equivalent to an image taken by a virtual camera with central projection (cf. Fig. 2). Here, a trade-off has to be found between image quality, field of view, and de-warping artifacts. The outer FE image areas are frequently suppressed,

Fig. 2. De-warping of a FE image: Left to right: original FE image and rectilinear de-warped images with decreasing focal length of the virtual camera. By increasing the focal length of the virtual camera, the field of view decreases and information content from the FE image is lost.

since squeezed FE regions cannot be de-warped with sufficient quality into a rectilinear image. A SemSeg model pre-trained on rectilinear data can subsequently be run on the generated rectilinear data as long as de-warping artifacts are acceptable.

To the best of our knowledge there exists only one reference dealing directly with FE SemSeg. This approach, which is related to the goal of our work is published in [5]. In this work, the authors focus on finding a specialized architecture for handling FE images. Due to a lack of FE images with provided SemSeg annotations, images from *Cityscapes* dataset [3] are transformed into images taken using a virtual FE camera exploiting a theoretical FE approximation. The resulting FE images are classified pixel wise via a Convolutional Neural Network (CNN) approach. Additionally *Zoom Augmentation* (one degree of freedom), by varying the focal length of the virtual FE camera, is employed. Validation of real FE data is not performed.

Our work differs fundamentally from the above described approach, since we want to train a network, which achieves superior performance on FE images, without the need of creating an expensive training dataset of annotated FE images. Furthermore, we want to avoid de-warping and enable segmentation also in the outer FE image area where de-warping cannot be performed. In contrast to [5], we use six degrees of freedom (DoF) for augmentation, focus on obtaining rectilinear SemSeg performance on FE images and our FE model is adapted for real manufactured camera lenses.

2 Methods

Our approach is based on the fact that a rectilinear image taken under central projection and its corresponding annotation can be de-warped (transformed) to a FE image by exploiting a commonly used projection model. Additionally, by varying the exterior camera orientation (pose), various artificial (augmented) FE images can be created which will look like a real camera image. After a transformation into FE geometry, vignetting effects resulting in black areas, which are typical for FE images, occur towards the corners and borders and also for non-illuminated pixels caused by the augmentation. These pixels are masked out (we call them "ignore pixels"), which means, that they will not be used for parameter optimization in training and neither be evaluated in validation nor in testing. Our augmentation contains six DoF (cf. Figure 3 (c)-(h)); the resulting

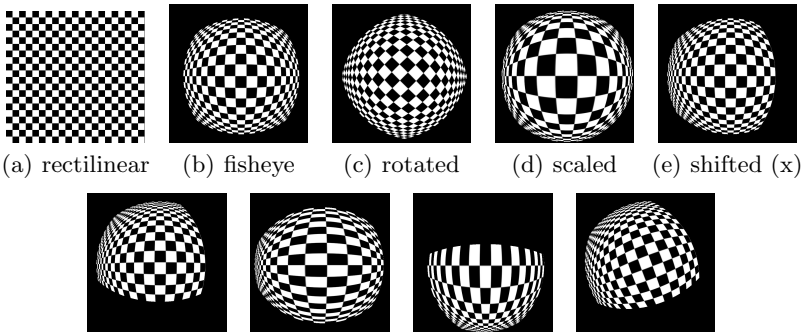


Fig. 3. DoF effects of augmentation. (a) Original rectilinear input image, (b): augmented and centered FE image, (c): (b)+ rotated by 45°, (d): (b)+ scaled, (e): (b)+ horizontally shifted, (f): (b)+ horizontally and vertically shifted, (g): (b)+ paned, (h): (b)+ tilted, (i) 6 degrees of freedom randomly applied.

non-illuminated pixel coordinates can be pre-computed for every augmented image. In theory, we can extend our DoF using additional parameters describing the interior camera orientation. However, in this paper, we use six parameters and train exactly for the camera model that we will use for evaluation later (Note that our FE augmentation differs from central projection augmentation (e.g. scaling or shifting) since object shape is distorted differently with increasing distance to the principal point. Using central projection, the shape itself is consistent and, typically, only similarity transformation, flipping, and cropping are used for augmentation [6]).

For the augmentation, we use the projection model introduced by Mei [11] including his notation, which is an extension of [1, 7]. As described in the following equations, up to a certain scale, points in 3D space can be transformed into a FE

image and vice versa. We use this model to enable indirect coordinate mapping of the rectilinear image and the related annotation via a look-up-table (LUT). Bilinear interpolation for the image and nearest neighbor interpolation for the annotation are used to keep values consistent. For every tuple consisting of an image and its corresponding annotation, 25 randomly chosen augmentations are created. Coordinates from a source image (rectilinear image) are mapped to the destination image (FE) by applying the following transformation (cf. Fig. 4), [11][3]:

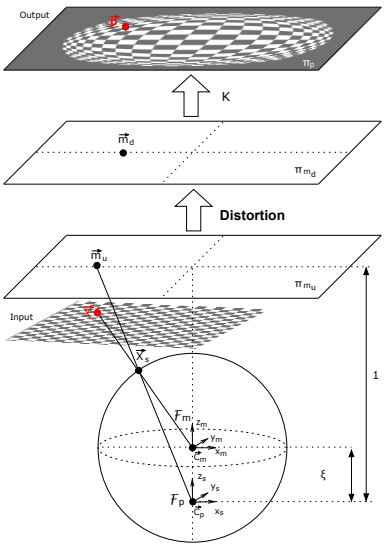


Fig. 4. Projection model from [11][1] adapted for this study

1. An image plane (rectilinear image) is located in the coordinate system (C_m), whereby the exact position of the plane is varied due to the augmentation. The plane is randomly rotated (three DoF) and shifted (three DoF) with respect to the coordinate system origin and axes. This allows to move the rectilinear image content to different locations in the FE image, depending on the randomly chosen plane position and orientation.
2. Points from the image plane are projected onto a unit sphere,

$$(\vec{X})_{\mathcal{F}_m} \rightarrow (\vec{X_S})_{\mathcal{F}_m} = \frac{\vec{X}}{||\vec{X}||} = (X_S, Y_S, Z_S). \tag{1}$$

3. The points are then changed to a new reference frame centered at $\vec{C}_p = (0, 0, \xi)$, where ξ is a lens depending parameter (cf. [11]),

$$(\overrightarrow{X_S})_{\mathcal{F}_m} \rightarrow (\overrightarrow{X_S})_{\mathcal{F}_p} = (X_S, Y_S, Z_S + \xi). \tag{2}$$

[3] `http://www.robots.ox.ac.uk/~cmei/articles/projection_model.pdf`

4. The points are then projected onto the normalized image plane (π_{m_u}). The coordinates on the normalized image plane are given by:

$$\vec{m}_u = (\frac{X_S}{Z_S + \xi}, \frac{Y_S}{Z_S + \xi}, 1). \tag{3}$$

5. Radial and tangential distortions are added with the distortion model introduced by Brown [2] with three radial and two tangential coefficients. The coordinates on the distortion affected image plane (π_{m_d}) are then:

$$\vec{m}_d = \vec{m}_u + D(\vec{m}_u, \mathbf{V}), \tag{4}$$

where D describes the coordinate depending distortion with the distortion coefficients $\mathbf{V}$.

6. The final projection involves a generalized camera projection matrix $\mathbf{K}$ (with the generalized focal length f, (u_0, v_0) as the principal point, s as the skew, and r as the aspect ratio). The coordinates on the fisheye image plane (π_p) are finally:

$$\vec{p} = \mathbf{K} \cdot \vec{m}_d. \tag{5}$$

In our study, we used the interior calibration parameters (ξ, u_0, v_0, f, s, r, $\mathbf{V}$) obtained from the target FE camera based on [15].

For the actual augmentation, the inverse transformation is implemented to enable indirect mapping (see above).

3 Experimental Evaluation

In this section, the results of experiments to investigate the described method are reported. The goals of the experiments can be divided into three parts and are summarized in Table 1:

Table 1. Structure of this section

Section	Goal
3.1	**Training and validation** - Baseline: Training on public MSCOCO dataset separately with (w/A) and without (wo/A) rectilinear augmentation. - Training on the same dataset with FE augmentation (w/FEA).
3.2	**Testing on FE images (consumer-camera images)** - Testing the models trained in Sec. 3.1 on real and artificially generated FE images.
3.3	**Re-training and testing on FE images (security-camera images)** - Re-training and testing the models trained in Sec. 3.1 with 400 security-camera images using central projection to obtain domain adaptation (the public dataset does not contain a security-camera pose).

Semantic segmentation methodology: We choose the commonly used architecture from the Visual Geometry Group, *VGG16* [14], to create baseline results against which we evaluate our experiments. The feature extractor is initialized with weights, which we obtain by pre-training on the ImageNet Dataset [4]. To output a semantic segmentation the last two fully connected layers are converted into fully convolutional layers and skip layers are introduced up to FCN8s as described in [12]. We freeze the weights from all layers prior to the originally fully connected layers and therefore only re-train the last layers of the network. Training is run with a batch size of five samples, using the adaptive optimizer ADAM [9].

While there are several other approaches, which potentially deliver better results on SemSeg, the scope of this work is to introduce and evaluate a method for improving SemSeg on FE images compared to a SemSeg output from a network, which is solely trained on rectilinear images. For this purpose the absolute accuracy and thus the choice of deep learning network architecture is seen as less important.

Key performance indicator: We optimize our network with the cross entropy loss and evaluate the resulting SemSeg image with the intersection over union (IoU) as used in many SemSeg benchmarks, e.g. the Cityscapes Dataset [3]. Furthermore, we report the average IoU, which is the average IoU over all class IoUs and the F1-score, followed by the true positive and true negative rates.

Test images: Since training and testing material is limited in the FE domain, especially in the SemSeg domain, we create and annotate three datasets for our study. MSCOCO-FE (Section 3.2) constitutes 37,504 images and annotations from the original MSCOCO [10] dataset, which we transform to 937,600 artificially generated FE images with pre-defined interior orientation parameters. GoPro-FE (Section 3.2) constitutes 50 real images taken using a *GoPro Hero 4* full frame, ultra-wide-angle view FE camera (4000x3000 pixels resolution, down-sampled to 640x480 pixels). The dataset consists of persons, animals, cars, bicycles, and furniture. The interior orientation of this camera is used for MSCOCO-FE; a bundle adjustment based on [15] is used to determine these parameters. However, since the GoPro camera does not have a fixed focal length, we use the averaged camera parameters for our projection model and do not consider variations. Security-Dome (Section 3.3) constitutes 12 challenging real FE images (640x640 pixels) taken using a circular FE security-camera in bird's eye view pose. While being a dataset with only a small number of images, each image comprises a lot of information with 25 - 55 persons being present per image.

Augmentation: Performances between networks where augmentation was used during training can not be fairly compared to networks trained without augmentation. By using augmentation during training, a bigger variability is introduced

to the training data and if correctly used, it will be beneficial for the network. Therefore, we not only implemented a FE augmented training as used in Section 3.1, but also an augmentation for the rectilinear images. By applying this to our baseline experiment trained on rectilinear images without augmentations (wo/A), we get a model trained on rectilinear images with augmentations (w/A), which can be better compared to our FE augmented model (w/FEA). This augmentation on the baseline experiment are arbitrary combinations of similarity transformations like translation (2D) or flipping. For both augmentation approaches (baseline and FE), the angle of rotation (see Fig. 3(c)) is limited to $-20° < \alpha < +20°$, because the shape of objects in the test set is expected to be in this range.

3.1 Training and validation

Training data: Microsoft COCO (MSCOCO) [10], is a diverse dataset containing consumer-camera images collected from the Flickr website. The dataset provides 80 object classes and one background class for instance-based SemSeg. The official test set of the dataset is not published and evaluation is only possible by submitting to the evaluation servers. Due to our final goal of training networks specialized for FE images such an evaluation is not of interest. Instead, we reduce the class set to the classes, which are relevant for the FE test dataset and subdivide the validation set of the publicly available dataset including annotations into a customized validation and test set. We remap the 80 to 16 coarser classes[4] for non-instance-based SemSeg. Training is performed based on 82,783 images, 3,000 images for validation, and 37,504 images for testing (the original MSCOCO validation image size minus our used validation images). As we do not carry out instance-based SemSeg, we do not apply the official evaluation metric, which is the average precision and average recall on instance-based segmentation, but use the IoU evaluation following the Cityscapes [3] evaluation protocol instead, because we are primarily interested in the impact of our method on FE images. Furthermore, all images are normalized by subtraction of the mean and division by the standard deviation.

Validation procedure: The standard procedure is to only apply augmentations during training and to leave the validation images untouched to measure the improvements in every subsequent epoch. For rectilinear augmentations this is reasonable, because the validation set consists of real images. However, in our case, we are not searching for the best performance on real images. Instead, we want to validate the performance on virtual FE images, where all transformations from rectilinear images represent a 'real' FE image on its own. Therefore, we perform random FE transformations also on the validation images. Due to reasons of comparability we do the same for the experiment using rectilinear augmentations.

[4] background, person, bicycle, car, motorcycle, airplane, bus, train, truck, boat, traffic light, furniture, animal, bagpack, handbag, and suitcase

Validation results: In Figure 5 the average IoU as a function of the training epochs on the validation data for the different training strategies is shown, where we choose an epoch size of 8000 images. Because of the reduction to 16 classes, we

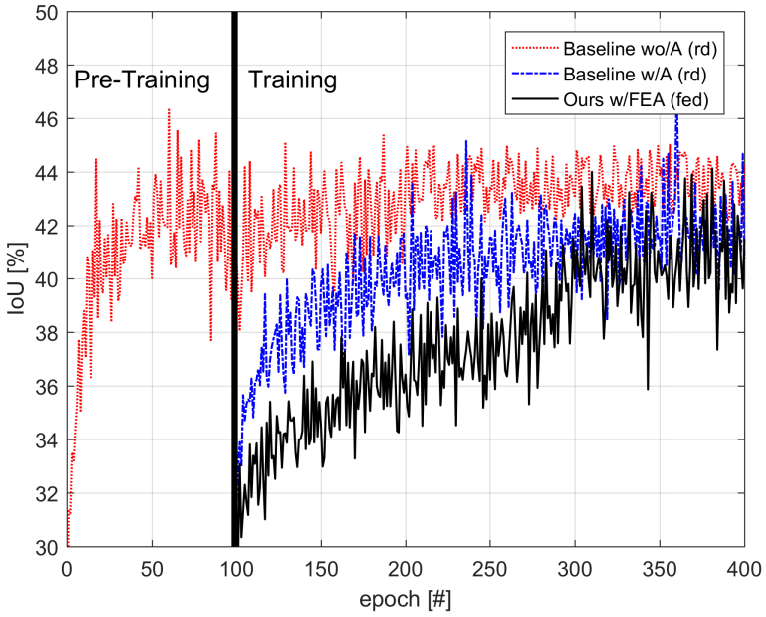


Fig. 5. Average IoU as function of training epoch. Training is performed on rectilinear data (rd) and artificially (augmented) generated FE data (fed).

have to deal with an over-representation of the background class in the MSCOCO dataset, which is why training tends to get stuck in a local minimum resulting in background as the only output. To overcome this issue, we carry out training and validation for the first 100 epochs using the original training and validation images (without augmentation) and subsequently use the resulting weights to initialize the networks of the augmentation based experiments. Another option would have been to introduce class weights.

The red curve in Figure 5 shows the IoU for the rectilinear training without augmentation (wo/A), the blue curve shows the IoU for rectilinear augmentation (w/A). The drop in IoU percentage (red vs. blue curve) results in the difference between training and validation images caused by the augmentation introduced after 100 epochs. On average, both models obtain around 42% IoU during the last epochs on the respective validation data. While the baseline model without augmentation converges after approximately 70 epochs, the training incorporating augmentation needs considerably longer to converge. This is not surprising if

taking into account that here, in contrast to the standard procedure, the validation images are also augmented. The black curve shows the IoU of the approach presented in this work, using augmented FE images. After around 300 epochs, no significant improvement is noted for both augmentation-based models. In our experiment the FE augmented model and the rectilinear augmented model eventually end up at roughly the same performance. This shows that regardless of the different augmentations, the model performances does not seem to differ as long as the same type of transformations are used in training and validation.

3.2 Testing on FE images

In this section, we evaluate our SemSeg performance on real fisheye images for models trained with rectilinear images against models trained with our augmented FE images and present qualitative and quantitative results.

Qualitative evaluation: Figure 6 shows examples for the two models (w/A and w/FEA) trained on MSCOCO deployed on our new dataset (GoPro-FE). The

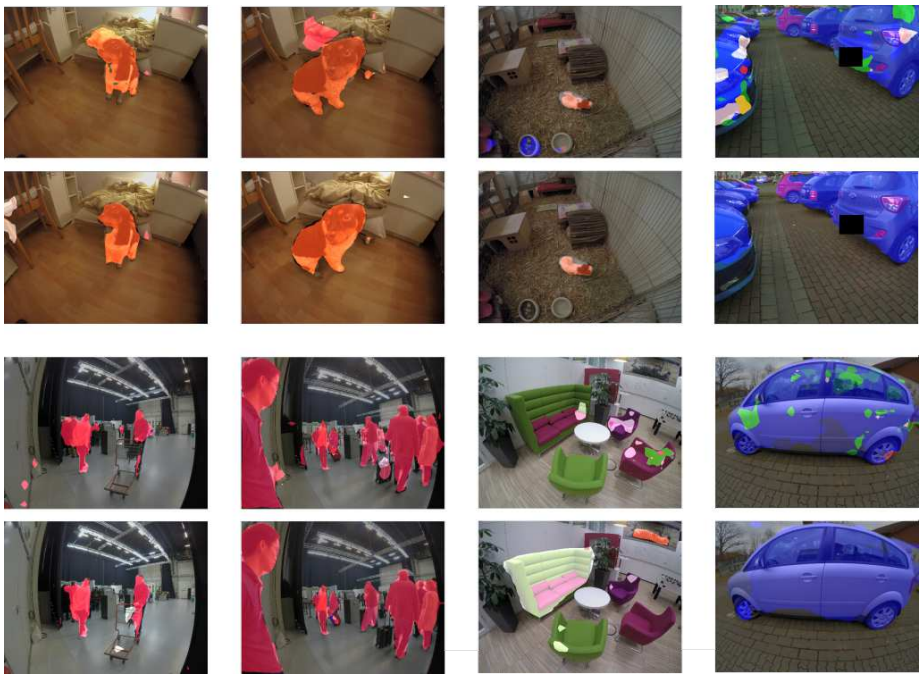


Fig. 6. Qualitative results on images randomly picked from our dataset. Odd line numbers show the result with the best rectilinear model; even line numbers correspond to the results for the model trained with our FE model and augmentation. Class visualization: person - red, animal - orange, car - blue, furniture - white.

images obtained using the baseline approach suffer from mis-classifications in the outer image area. Since this model is not trained on FE images, the number of incorrectly segmented pixels rises with increasing distance to the principal point due to stronger FE effects. Images segmented from the model trained with our approach show improved results in particular in the outer image areas (1st and 2nd line, 1st image from the right). In the image showing the furniture (3rd and 4th line, 2nd image from the right) the result of explicitly learned FE effects can be observed for the curved couch and the zebra image in the back.

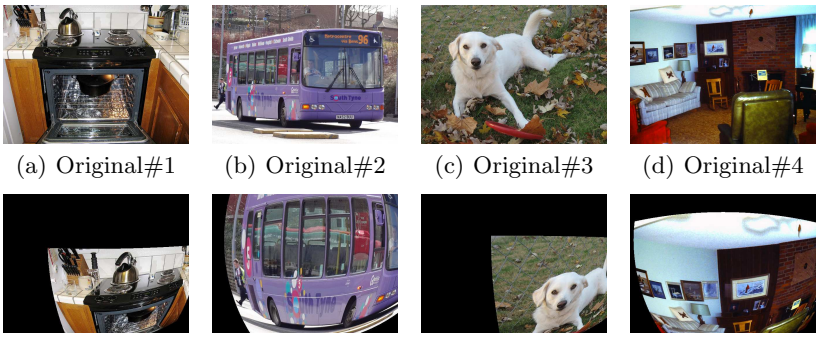
Quantitative evaluation: Table 2 shows the quantitative results for our 50 real FE test images (GoPro-FE). Our model considerably outperforms the two baseline models (+18.3% $\overline{IoU}$) while using the same raw training material as the baseline with augmentation. All classes are better classified by our approach, evidenced by an F1-score of +18.8%, a true positive rate of +27.8%, and a true negative rate of +7.7%. The large margin compared to the baseline is not surprising since the baseline model is not trained for FE data. Notably, we observe that we can use rectilinear images plus augmentation to obtain much higher IoU on FE images.

Table 2. Quantitative evaluation on our own FE dataset. $\star$ indicates not countable.

	#instances	Baseline wo/A	Baseline w/A	Ours w/FEA
IoU (average)	175	33.9	37.3	**55.6**
background	$\star$	90.5	90.3	**94.7**
person	49	51.0	59.8	**75.5**
bicycle	28	12.1	13.9	**27.6**
car	75	30.0	29.8	**45.4**
sitting furniture	6	0.0	4.9	**39.1**
animal	17	19.9	25.1	**51.1**
F1	175	42.8	50.0	**68.8**
True Positive Rate	175	31.4	34.2	**62.0**
True Negative Rate	175	89.0	87.9	**95.6**

Evaluation on artificial FE images: In this section, we report the performance for the FE SemSeg obtained for images that are from the same domain than the ones used during training (MSCOCO-FE), but not used for training and validation. We use artificially generated images since the test material is limited. To do so we also transform the MSCOCO test dataset, which is split (original validation dataset minus the 3,000 images that we only use for our validation), into FE geometry. We transform all of the 37,504 images and the label images each to 25 randomly generated FE image versions using the projection model introduced in Section 2, resulting in 937,600 test images.

Figure 7 shows four example images from MSCOCO augmented to FE images, whereas Table 3 lists the SemSeg results. The results show that our aug-



(a) Original#1 (b) Original#2 (c) Original#3 (d) Original#4

(e) Augmented#1 (f) Augmented#2 (g) Augmented#3 (h) Augmented#4

Fig. 7. Four randomly picked original images from the MSCOCO dataset and corresponding fisheye images.

Table 3. Quantitative evaluation on an artificially generated **FE** test set. The used augmented test dataset constitutes 937,600 images.

	Baseline wo/A (↑)	Baseline w/A(↑)	Ours w/FEA (↑)
IoU (average)	34.0	34.7	**43.0**
background	90.1	89.9	**92.9**
person	65.1	58.9	**73.1**
bicycle	23.5	24.7	**30.2**
car	24.4	30.5	**35.2**
motorcycle	36.8	48.1	**48.9**
airplane	33.3	34.8	**43.5**
bus	48.5	48.4	**66.4**
train	46.8	44.4	**63.1**
truck	28.0	26.3	**41.1**
boat	15.3	18.2	**22.8**
traffic light	22.5	25.4	31.2
furniture	24.6	19.7	**26.4**
animal	64.9	57.8	**75.3**
bagpack	0.0	6.6	**16.0**
handbag	**4.1**	1.9	0.0
suitcase	16.2	19.6	**23.2**
F1-score	46.5	48.0	**56.1**
True positive rate	42.2	42.9	**52.1**
True negative rate	97.8	97.5	**98.5**

mentation is very powerful. We obtain +8.3% more IoU than the other two tested approaches on the same test images with our augmentation. Using the proposed method, all classes, except the handbag class, are considerably better classified in comparison to using the baseline approaches. The class handbag is the class with the lowest number of pixels, and therefore an adequate training of this class is challenging. Moreover, our true positive rate is +9.2% and F1-score +8.1% higher than those obtained for the other two approaches. It is also noticeable, that the two baseline models show almost identical performance. This indicates, that it is not sufficient to add variability by any arbitrary augmentation. To gain performance on the target images, it is crucial to choose augmentations which suit the camera model. We will underline this further in Section 3.3.

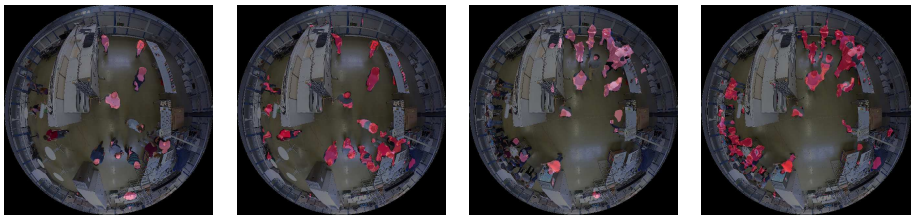
3.3 Testing on FE images in the security-camera domain

In this Section, we show that it is crucial to choose the correct underlying camera model when applying our FE augmentation method. Therefore, we evaluate our approach on our Security-Dome test dataset, which consists of challenging real dome security-camera images. Compared to the GoPro-FE dataset, the FE effect is much stronger and we are dealing with a bird's eye view. First, to create our baseline experiment, we evaluate the baseline model (w/A) and the model trained with FE-augmentations (w/FEA) on the Security-Dome dataset. As expected, the w/FEA model achieved higher performance, but especially facing some issues towards the borders of the image. Since not having many persons captured from the bird's eye view in the MSCOCO dataset, we decided to use an already annotated dataset Security-Recti which consists of 400 rectilinear images also from the security domain. To evaluate the importance of correct augmentation, we now re-train our w/FEA model for 100 epochs with the same FE-augmentation as used for the GoPro and a FE-augmentation, which uses the transformation parameters from the dome security-camera. For completion, we also re-trained the w/A-model with the Security-Recti images. In Table 4, the high impact of the augmentations can be observed. With the baseline approach for rectilinear images an IoU of 17.4% is achieved, while the use of our FE-augmentation method is already giving a 10% gain even with the parameters for the GoPro-camera. Now applying the correct augmentation, the performance reaches 35.4% IoU. This is quite impressive on the Security-Dome with its very

Table 4. Measures for class person in the challenging Security-Dome dataset.

Class Person	w/A	w/A +re-train	w/FEA (GoPro)	w/FEA +re-train (GoPro)	w/FEA +re-train (Sec.-Cam.)
	Baseline	Baseline	(wrong cam.)	(wrong cam.)	(correct cam.)
IoU	15.1	17.4	20.4	27.1	**35.4**
F1-score	26.2	29.7	35.2	42.6	**52.3**
True positive rate	16.4	19.5	22.3	29.0	**38.2**
True negative rate	99.0	98.7	99.1	99.2	**99.2**

strong distortions and extreme viewpoint considering that not a single image from this domain was was recorded and annotated. Qualitative results can be seen in Figure 8.



(a) Baseline#1 (b) Ours#1 (c) Baseline#2 (d) Ours#2

Fig. 8. SemSeg for FE images. Two input images are segmented with w/A (Baseline) and with w/FEA (Ours) both re-trained on Security-Recti.

4 Conclusion

Training data for fisheye (FE) semantic segmentation is limited and recording and annotation for supervised learning is expensive and tedious. Additionally, much more training data is required than for rectilinear data, since, depending on the employed lens, the shape of objects varies with the position of the object in the image. We presented an approach to re-use rectilinear training material to enable semantic segmentation on FE images obtained from action cameras, smartphones or security cameras. In particular, we introduce rectlinear-to-FE transformations to the augmentation stage in training. Additional annotations or specially tailored deep learning architectures are not necessary. On average our approach is +18.3% IoU (trained on MSCOCO) better on real full-frame fisheye images (n=50 images) and +8.3% IoU better on artificially generated FE images (n=937,600 images) when using the same raw training material as the baseline with rectilinear augmentations. Furthermore, we obtained +18.0% *IoU* on a new, very challenging dome security-camera dataset (circular FE) where the camera is mounted in bird's eye view.

One further effect, which we realized was that in our experiment the FE augmented model and the rectilinear augmented model eventually end up roughly at the same performance on the respective dataset. This shows that regardless of the different augmentations, the model performance does not seem to differ as long as the same type of transformation is used in training and validation.

In the future, we will increase the DoF for augmentation and investigate, how the deep learning model architecture can be tailored for further improvements on FE SemSeg. Another direction, we want to explore is the use of Generative Adversarial Networks (GANs) [8] to do semantic segmentation on FE images: instead of employing an explicit sensor model (see equations (1) to (5)) we want

to train a GAN with the aim to transform arbitrary FE images to rectilinear images and the segmentation back to the FE image. The segmentation network is then only trained with the original training set of rectilinear images. We will compare obtained segmentation performance with our FE augmented model.

References

1. Barreto, J.P., Araujo, H.: Issues on the geometry of central catadioptric image formation. In: CVPR. pp. 422–427. IEEE (2001)
2. Brown, D.C.: Decentering Distortion of Lenses. Photogrammetric Engineering **130** (1966)
3. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: CVPR. pp. 3213–3223. IEEE (2016)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR. pp. 248–255. IEEE (2009)
5. Deng, L., Yang, M., Qian, Y., Wang, C., Wang, B.: CNN based semantic segmentation for urban traffic scenes using fisheye camera. In: Intelligent Vehicles Symposium (IV). pp. 231–236. IEEE (2017)
6. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Rodríguez, J.G.: A Review on Deep Learning Techniques Applied to Semantic Segmentation. CoRR **abs/1704.06857** (2017)
7. Geyer, C., Daniilidis, K.: A unifying theory for central panoramic systems and practical implications. In: ECCV. pp. 445–461. Springer (2000)
8. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative Adversarial Networks. CoRR **abs/1406.2661** (2014)
9. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. CoRR **abs/1412.6980** (2014)
10. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: ECCV. pp. 740–755. Springer (2014)
11. Mei, C.: Couplage Vision Omnidirectionnelle et Télémétrie Laser pour la Navigation en Robotique/Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping. Ph.D. thesis, INRIA Sophia Antipolis, Project-team ARobAS (2007)
12. Shelhamer, E., Long, J., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. CoRR **abs/1605.06211** (2016)
13. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.: CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. CoRR **abs/1703.01515** (2017)
14. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR **abs/1409.1556** (2014)
15. Strauß, T., Ziegler, J., Beck, J.: Calibrating multiple cameras with non-overlapping views using coded checkerboard targets. In: ITSC. pp. 2623–2628. IEEE (2014)
16. Su, T.F., Chen, Y.L., Lai, S.H.: Over-Segmentation Based Background Modeling and Foreground Detection with Shadow Removal by Using Hierarchical MRFs. In: ACCV. pp. 535–546. Springer (2011)
17. Thoma, M.: A Survey of Semantic Segmentation. CoRR **abs/1602.06541** (2016)

18. Wei, X., Guo, Y., Gao, X., Yan, M., Sun, X.: A new semantic segmentation model for remote sensing images. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS). pp. 1776–1779. IEEE (2017)
19. Yang, Y., Yang, J., Yan, J., Liao, S., Yi, D., Li, S.Z.: Salient Color Names for Person Re-identification. In: ECCV. pp. 536–551. Springer (2014)