Instance segmentation in fisheye images

Rémi Dufour, Cyril Meurie, Clément Strauss, Olivier Lezoray

Instance segmentation in fisheye images

Rémi Dufour[1] remi.dufour@railenium.eu
Cyril Meurie[2,1] cyril.meurie@univ-eiffel.fr
Clément Strauss[1] clement.strauss@railenium.eu
Olivier Lézoray[3,1] olivier.lezoray@unicaen.fr

[1]FCS Railenium, F-59300 Famars, France
[2]COSYS-LEOST, Université Gustave Eiffel, IFSTTAR, Univ. Lille, Villeneuve d'Ascq, France
[3]Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, France

***Abstract*—In this paper we propose a data augmentation method for instance segmentation in fisheye images. A lot of progress has been made on the task of instance segmentation in the last few years, particularly for rectilinear images. In fisheye images, detection tasks have mostly been explored as a semantic segmentation task. Instance segmentation in fisheye images is challenging and has not yet been fully explored. There is also much interest in the development of deep neural networks that can handle both rectilinear and fisheye images. Indeed, this can be interesting to have control on computing resource requirements, of paramount importance for real-time systems e.g., in transportation systems. This paper aims to explore these two challenges using Mask R-CNN trained with a data augmentation method designed to provide good performance on both rectilinear and fisheye images. We show that performance on fisheye augmented images can be increased by 9% while only decreasing performance on rectilinear images by 2%, and that performance on wide angle fisheye cameras can be increased by 18.4% compared to the reference, which provides more benefits than a simple vertical flip augmentation.**

***Index Terms*—fisheye images, instance segmentation, deep learning, data augmentation**

I. Introduction

Instance segmentation is a particularly useful detection task in computer vision. The fine-grained localization of objects allows for a vast range of applications but it can also be seen as a useful first step for bounding box object detection.

In transportation environments, and more particularly for trains, detecting and tracking persons is of particular interest for surveillance and safety purposes. Detecting intrusions, crowds, agitation, violence, faintings: these are some of the potential uses of a human detection and tracking system. In this context, human instance segmentation is a first essential step to these different detection and tracking tasks.

Both rectilinear and fisheye cameras are helpful for surveillance tasks in transportation contexts. Depending on the location, one kind of camera would be better suited than the other. Since rectilinear and fisheye images are very different, it is natural to develop dedicated systems for each. However, for limited computing reasons, it might be more interesting to have a single detection system that can handle both kinds of images. To achieve this goal, we apply a data augmentation method which transforms rectilinear images into fisheye effect (FE) images. Experimenting this method for the task of instance segmentation is the main contribution of this paper.

This paper is organized as follows. Related works of the literature are presented in Section II. The considered methods are detailed in Section III. Experiment results are reported and discussed in section IV. Finally, section V concludes.

II. Related Works

Within the last few years, several algorithms stood out for the task of instance segmentation such as Mask R-CNN [1] and YOLACT [2].

Mask R-CNN is an algorithm derived from Faster R-CNN [3]. Mask R-CNN is currently the most widely used reference work for the task of instance segmentation and it provides state-of-the-art performance.

YOLACT is a recent Fully Convolutional model that can perform real-time instance segmentation thanks to its efficient architecture. Indeed, it does not relies on the heavy "feature localization" step present in the Mask R-CNN architecture.

Progress in the task of instance segmentation has also been enabled thanks to the release of important instance segmentation datasets. The most commonly used instance segmentation datasets are MS COCO [4] and CityScape [5].

Fisheye images are not well represented in these large scale datasets. This lack of suitable training data presents a challenge for training Convolutional Neural Network (CNN) and performing well on fisheye images. Creating such a dataset using manual annotation is a tedious and delicate task that is very expensive and time consuming. CNN that have been trained on classical rectilinear images learn to recognize objects in rectilinear settings and consequently offer lower performances when used on fisheye images. This is particularly the case near the image borders, where the fisheye distortions are the highest. Recently, the WoodScape dataset [6] was proposed but it remains relatively small as compared to the MS COCO dataset used for training state-of-the-art instance segmentation algorithms. Approaches to the aforementioned problem can be broadly separated into two categories: The first category consists in correcting the fisheye distortions from the images so that existing algorithms can be used. However fisheye dewarping can also lead to artifacts on the border of the image. This requires to find a compromise between camera field-of-view, dewarping artifacts, and computation time. The second

978-1-7281-8750-1/20/$31.00 ©2020 IEEE

category consists in adapting the algorithm in order to directly process fisheye images. This second category is particularly interesting because such an algorithm could potentially be used for both rectilinear and fisheye images. Another advantage is that this approach doesn't require additional processing on top of the instance segmentation algorithm.

A common way of dealing with the lack of specific fisheye dataset is to create synthetic training data from an existing segmentation dataset. This strategy consists in applying a transformation to the rectilinear images so that the result harbors the challenging properties (distortions) of fisheye images. In [7], real-time semantic segmentation in the context of autonomous driving is achieved using a customized neural network architecture combined with a FE data augmentation strategy. In [8], barrel and pillow distortions are used as part of a data augmentation strategy to robustify the training of a semantic segmentation algorithm in order to improve performance on panoramic images.

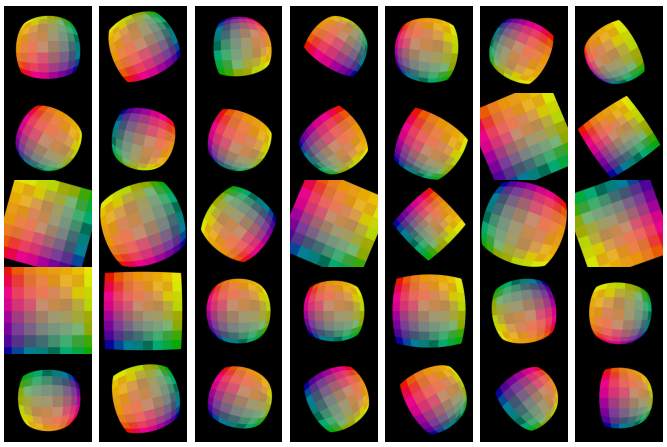
In [9], a FE data augmentation method based on a fisheye camera model is proposed and evaluated on a fisheye dataset for the task of semantic segmentation. The authors show that the FE augmentation improves the performances on fisheye dataset as compared to those obtained with classical rectilinear augmentations (such as translation, flipping and rotation).

In this paper, we follow a similar data augmentation strategy in order to obtain satisfactory segmentation results on both rectilinear and fisheye images acquired in transportation environments.

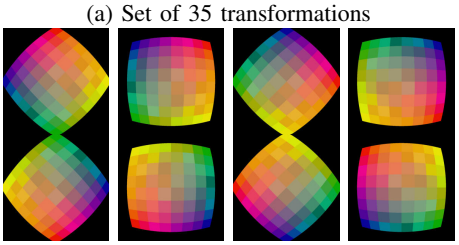
III. Method

The proposed approach uses an existing instance segmentation algorithm, Mask R-CNN [1], as a baseline. We seek to improve it in order to be able to deal with both rectilinear images and fisheye images. The fisheye domain adaptation approach consists in training Mask R-CNN on the COCO2017 dataset, using FE data augmentation. The data augmentation strategy uses the same transformation model used in [9], and originally presented in [10]. This projection model works by projecting the image on a unit sphere, changing the reference frame, and projecting the points on the normalized image plane. At this stage, the radial and tangential distortions are added according to the model introduced by Brown [11], and finally, the projection is done using the camera's parameters.

This projection model is used to create a set of transformations that can be applied efficiently for data augmentation dedicated to training. In our study, two sets of transformations were conceived in order to study the effect of the transformation diversity. The first set contains 35 different complex FE transformations (as shown in Figure 1a). And the second one contains 8 different simple rotated transformations (as shown in Figure 1b). The first set has the following transformation properties: translations in the range [0;0.5] in normalized scale, arbitrary rotations on the z axis of unconstrained magnitude, scaling in the range [50%;120%], radial distortions in the range [-0.5;-1.0], and ξ factor in the range [0;1]. The second set has the following transformation properties: no translation, 8 different rotations with a step of $\frac{\pi}{4}$, no scaling, no distortion, ξ factor of 1.



(a) Set of 35 transformations



(b) Set of 8 transformations

Fig. 1: Two transformation sets

The FE data augmentation is carried out as follows: an augmentation ratio is defined to apply fisheye augmentation to a portion of training examples. When an image is picked to be fisheye augmented, a specific transformation is randomly picked from the precomputed augmentation set and applied to the image. This data augmentation method was chosen in order to optimize the speed of the transformation and therefore the speed of the training phase.

Mask R-CNN training is done using the default parameters of the Detectron framework [12], with adjustments, such as a higher batch size per gpu, and a smaller number of training steps, in order to get good performance on our hardware (one Nvidia V100 GPU). Two different training schedules are used: a long one for the large resnet101 architecture, and a short one for the small resnet50 architecture. The long schedule is used to obtain the maximum detection performance, while the short one is used to execute more training runs for comparison of FE augmentation settings. The long schedule consists of 90k steps and a base learning rate of 0.02 which is divided by 10 at steps 50k and 70k. The short schedule consists of 40k steps and a base learning rate of 0.02 which is divided by 10 at steps 20k and 30k.

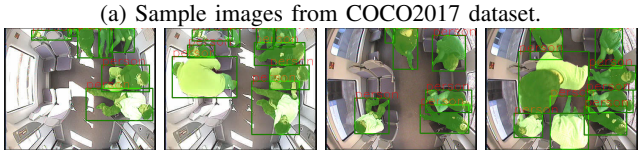
IV. Experimental results

A. Evaluation datasets

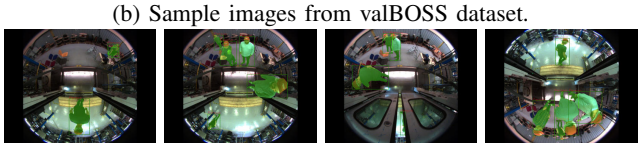
The detection results are evaluated qualitatively and quantitatively using three datasets: the validation subset of COCO2017, valBOSS and trainDoor.

For the creation of valBOSS, we sampled 60 frames from two video sequences from the BOSS dataset [13]. These frames were then annotated manually using CVAT [14] for human instance segmentation (i.e. only humans were labelled). We illustrate this dataset with sample images in Figure 2b.

(a) Sample images from COCO2017 dataset.



(b) Sample images from valBOSS dataset.



(c) Sample images from trainDoor dataset.

Fig. 2: Sample images from COCO2017, valBOSS and trainDoor datasets.

For the creation of the trainDoor dataset, we sampled a video dataset featuring scenes meant to resemble the passage of pedestrians through a train door. 121 frames were annotated for human instance segmentation. We name this evaluation dataset "trainDoor" and show sample images in Figure 2c.

To evaluate the performance of our strategy, we use the official tools of the COCO2017 dataset. We consider the following metrics: APall (mean average precision over many different thresholds), AP50 (threshold of 50% confidence), AP75 (threshold of 75% confidence), APS (mean average precision for small objects), APM (medium objects), APL (large objects). Concerning the valBOSS and trainDoor datasets, only the "person" label outputed from Mask R-CNN is considered. The APS metric is not considered, because of the lack of small objects in these datasets.

B. Importance of COCO2017 pretraining

Mask R-CNN is normally trained starting from a backbone pretrained on ImageNet [15], and then trained on COCO2017. Transfer learning from backbones pretrained on ImageNet provides a significant performance boost on different domains. However it has not been proven that using an entire Mask R-CNN neural network pretrained on COCO2017 provides an advantage over starting only from an imagenet backbone when it comes to transfer learning on the fisheye domain. In order to confirm this hypothesis, we trained a Mask R-CNN network starting from an ImageNet backbone, and another starting from a complete neural network pretrained on COCO2017. Table I illustrates the performance comparison between the two approaches. The result confirms the usefulness of using Mask R-CNN network pretrained on COCO2017 for transfer learning on the fisheye domain with a data augmentation of 35 transformations. This shows clearly that the learned filters on COCO2017 rectilinear images do constitute a good initialization for fisheye images. In the rest of the paper, all Mask R-CNN trainings are done by starting from a Mask R-CNN network completely pretrained on COCO2017.

TABLE I: Impact of complete COCO pretraining vs starting from only an ImageNet backbone, evaluated on COCO2017val augmented with 35 transformations.

	APall	**AP50**	**AP75**	**APS**	**APM**	**APL**
ImageNet start	0.12	0.22	0.12	0.05	0.15	0.20
COCO2017 start	**0.16**	**0.28**	**0.16**	**0.07**	**0.19**	**0.27**

C. Importance of FE augmentation ratio

The augmentation ratio is an important parameter of our model. With 100% of the images being transformed to FE during training, the neural network doesn't see any rectilinear image anymore, potentially hurting its performance on rectilinear images. Since we seek a good performance on both rectilinear and fisheye images, we experimented our model with augmentation ratios of 0%, 25%, 50%, 75% and 100%. For this experiment, we used the backbone architecture resnet50 with a short training schedule. Results are illustrated in Figure 3. From this experiment, one can notice that even a 25% ratio already provides a huge jump in performance on FE images. Above 75% of augmentation, the performance on classical rectilinear images starts degrading significantly. For the purpose of finding a compromise between performance on both domain, we consider that a 50% ratio is acceptable. This ratio leads to a 9% improvement in FE augmented images, while only decreasing performance by 2% on rectilinear images.

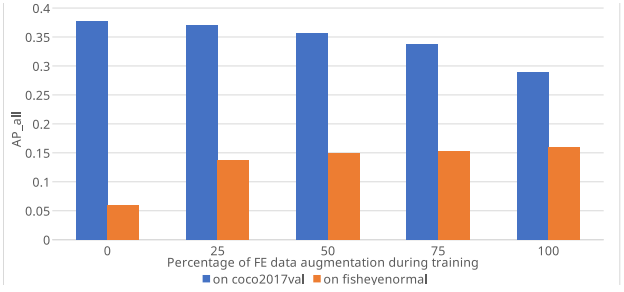


Fig. 3: Evolution of AP_all on rectilinear (Blue) and transformed (Orange) COCO2017val depending on the ratio of FE augmentation during training.

In order to confirm the good performance of the 50% ratio on classical images, Mask R-CNN is trained using this ratio with a resnet101 backbone and a long training schedule. Results are shown in Table II. One can notice that the difference in terms of performance on classical images is negligible, which means that the network performs much better on FE augmented COCO2017 images while not losing performance

on rectilinear images. This first result shows therefore that, with a careful data augmentation, it is possible to design a CNN that performs well on both rectilinear and fisheye images.

D. Evaluation on valBOSS and trainDoor

After evaluating the performance on rectilinear images, we endeavoured to confirm the good performance of the chosen training settings on FE datasets taken in the context of transportation systems, but for which it was not trained onto: valBOSS and trainDoor, as shown in Table III.

Concerning the valBOSS dataset, one can notice a small improvement, but not enough to conclude on the benefit of the strategy.

TABLE II: COCO2017 Performance of 50% fisheye augmented **resnet101** Mask R-CNN vs Reference.

	APall	**AP50**	**AP75**	**APS**	**APM**	**APL**
Reference	40.1	61.9	44.0	22.6	43.6	52.6
50% FE aug	39.5	60.7	43.2	22.1	43.5	51.7

TABLE III: Performance of **resnet101** Mask R-CNN trained with 50% FE augmentation vs reference on the valBOSS, trainDoor and trainDoorAug datasets.

	APall	**AP50**	**AP75**	**APM**	**APL**
valBOSS					
Reference	18.0	43.9	12.1	**26.6**	10.0
50% FE aug	**18.0**	**46.0**	**12.8**	25.1	**15.1**
trainDoor					
Reference	55.8	78.5	66.3	49.7	59.9
50% FE aug	**70.3**	**90.6**	**83.6**	**59.1**	**77.6**
trainDoorAug					
Reference	43.5	64.7	50.4	31.5	52.2
50% FE aug	**67.0**	**90.7**	**79.0**	**54.7**	**74.5**

TABLE IV: Performance of **resnet50** Mask R-CNN trained with 50% FE augmentation (with two different transformation sets) vs reference on the valBOSS and trainDoorAug datasets.

	APall	**AP50**	**AP75**	**APM**	**APL**
valBOSS					
Reference	14.6	38.1	9.3	21.5	8.8
halfvflip	12.0	35.2	6.6	19.1	8.5
35FE transformations	13.9	39.7	7.8	20.6	9.2
8FE transformations	**22.8**	**52.8**	**17.8**	**29.6**	**9.5**
trainDoorAug					
Reference	43.6	67.5	50.5	31.5	51.0
halfvflip	60.6	86.4	69.4	**53.9**	65.3
35FE transformations	61.0	87.6	71.7	51.2	67.0
8FE transformations	**62.0**	**87.7**	**74.2**	51.4	**68.5**

For the trainDoor dataset, a large improvement can be noticed. This dataset presents a lot of persons that do appear "upside down" (i.e. head down and feet up), and this is probably the main reason for the large difference in terms of performance: the reference has been trained on COCO2017 that mainly contains persons displayed upright (i.e. head up and feet down). The reference has thus not been properly trained to recognize upside down people, since CNN are not rotationally invariant [16]. Figure 4 shows some examples of upside down persons being detected by FE augmented Mask R-CNN but not by the reference Mask R-CNN. Moreover, the dataset, as is, has an unbalanced number of upright persons and upside down persons. This could give an unfair advantage to one algorithm depending on what each one is best suited for. In order to fix this, we propose to create an augmented trainDoor dataset, consisting of the trainDoor dataset concatenated with a vertically flipped version of itself. This also has the advantage of doubling the size of the evaluation dataset, to 242 images. We name this new dataset trainDoorAug. The results obtained on trainDoorAug are shown in Table III. The reference performs a lot worse on this augmented dataset while our approach obtains similar performance. This confirms that the vertical orientation of the dataset leads to a huge bias in the evaluation. With this in mind, in order to better evaluate the impact of the FE augmentation, our approach should be evaluated against a reference trained with vertically flipped images.

E. Importance of FE augmentation vs vertical flip augmentation

Given the previous results, we also trained a Mask R-CNN neural network with a 50% augmentation ratio of vertically flipped images in order to have a more fair reference. We use the short training schedule with a small resnet50 architecture for the sake of speed. The reference and the network trained with FE augmentations are also using the short schedule and resnet50 architecture so that the comparison is valid. The results of this new comparison are shown in Table IV, under the name "halfvflip". The results confirms that a simple vertical flip augmentation is already interesting to get a large performance increase on the augmented trainDoor dataset. Yet, we still notice an improvement of 1.9% APall on valBOSS, and 1.4% APall on trainDoorAug, when comparing 35FE augmentation with halfvflip augmentation. This difference shows the usefulness of FE augmentation, but we can now consider if the set of transformations can be made simpler.

F. Impact of the size of the transformation set

We now study if the 35 transformations set is overkill for the task of instance segmentation in FE images. To do so, we created a smaller transformation set, containing only 8 different rotations with a minimal fisheye effect. We trained Mask R-CNN with this augmentation, using a 50% transformation ratio. As before we evaluate this neural network on our two datasets captured in a transportation context: the valBOSS and augmented trainDoor datasets. Considering the results presented in Table IV, one can notice a substantial performance increase on valBOSS dataset, and a slight performance increase in trainDoorAug dataset. In conclusion, it might not be necessary to use 35 different highly distorted transformations for data augmentation, since 8 transformations seem to be sufficient. These final results are illustrated qualitatively in Figure 5.

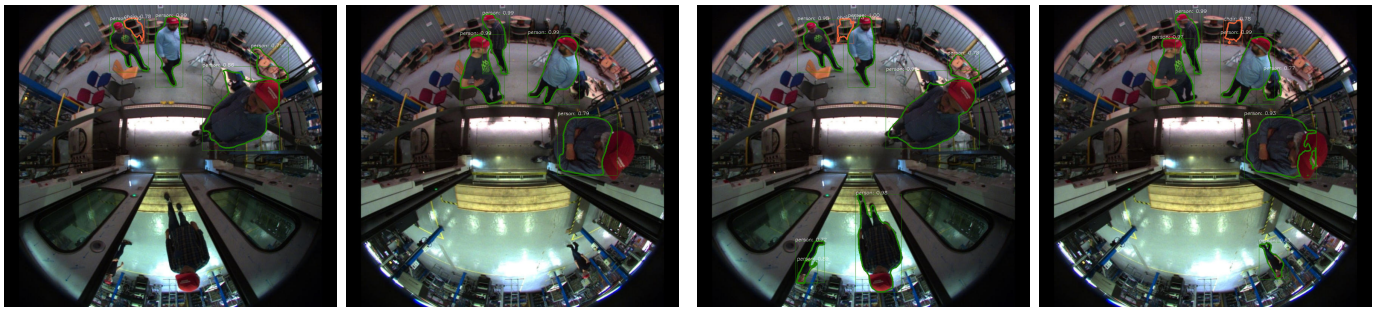


Fig. 4: First two images: results with resnet101 Mask R-CNN reference. Last two images: results with resnet101 Mask R-CNN with FE augmented training.

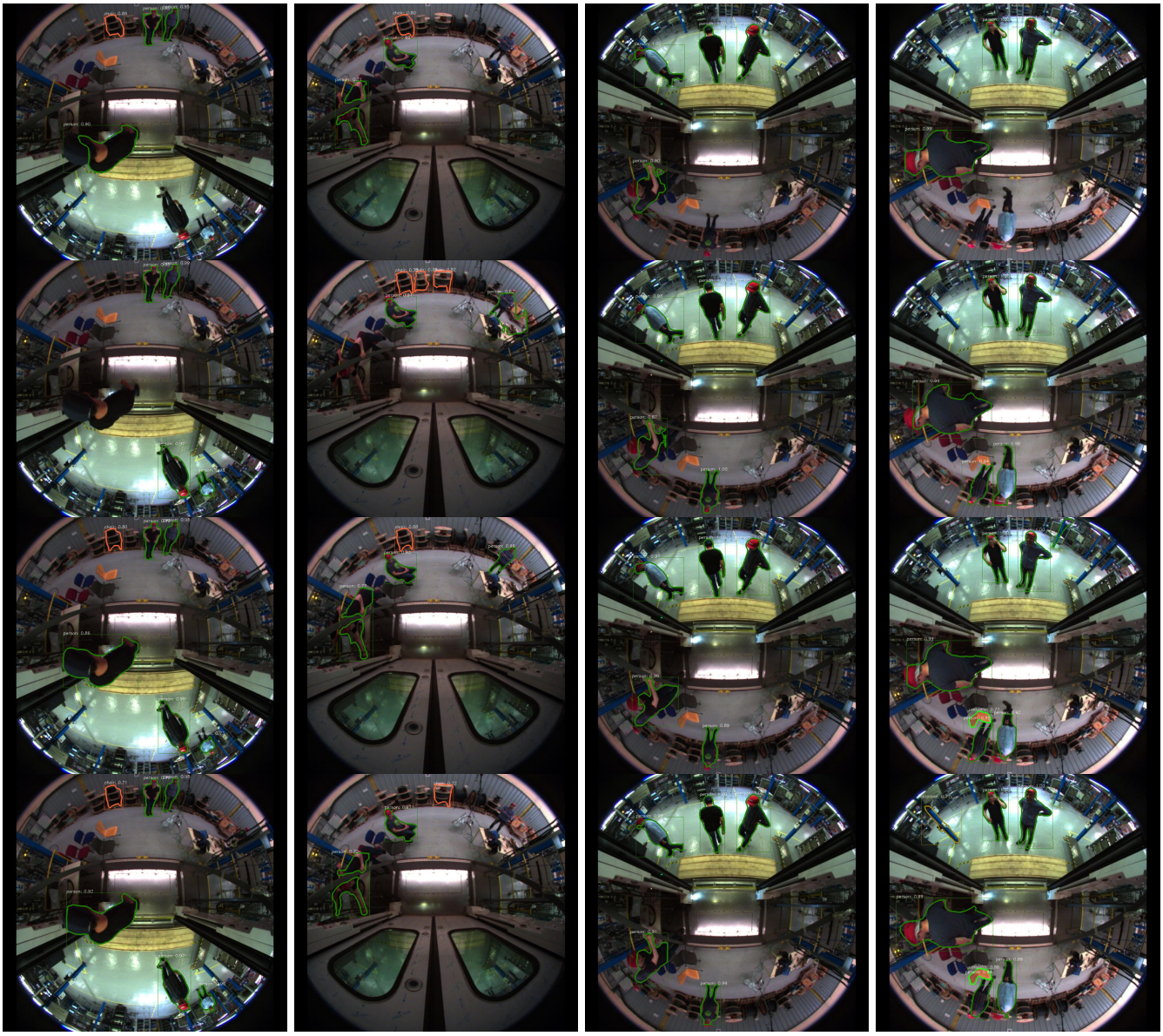


Fig. 5: Top line: results with resnet50 Mask R-CNN reference. Second line: results with resnet50 Mask R-CNN with vertical flip augmentation. Third line: results with resnet50 Mask R-CNN with 35 transformations augmentation. Fourth line: results with resnet50 Mask R-CNN with 8 transformations augmentation.

In light of all these experimental results, one can observe that a large part of the challenge associated with FE images is due to the object rotations that objects are prone to, that Mask R-CNN is not trained to handle. On the other hand, this shortcoming is mainly related to the nature of existing large scale training datasets such as COCO2017. The experiments we led have shown that it is possible to overcome these shortcomings for fisheye images by using a FE data augmentation strategy. Moreover, it is possible to do so while not hurting performance on classical rectilinear images. This opens the door to the possibility of using the same system to deal with both rectilinear and fisheye images.

For the two transportation datasets we considered, the large majority of human instances to segment were in the center of the images, which means they were not very distorted. This is probably the main reason why a few image transformations, mainly consisting of rotations, help so much on these datasets. More annotated fisheye images, in particular presenting humans near the border, with high distortion, could be necessary to really know how current algorithms deal with high distortions in fisheye images.

V. Conclusion

In this paper we have proposed a way to train Mask R-CNN so that it can provide good instance segmentation both in rectilinear and fisheye images. We have shown that data augmentation with specific fisheye transformations can render Mask R-CNN able to deal with both normal and fisheye images. We studied the impact of the types and magnitude of the transformations used as well as the augmentation ratio on both rectilinear and fisheye datasets. In conclusion, a small set of rotated fisheye transformations is enough to get an improvement, and a ratio of around 50% of data augmentation is a good compromise to obtain good performance on both rectilinear and fisheye images, with the same neural network. We also have shown this on 2 transportation datasets, the first, valBOSS, containing 60 images showing high distortion, and the second, trainDoorAug, made from 121 images taken with a wide angle fisheye camera. The proposed strategy offers satisfactory results on these two datasets, improving APall by 8.2% compared to the reference on valBOSS and 18.4% compared to the reference on trainDoorAug.

Acknowledgments

This research work is developed in the framework of Autonomous Train - Passenger Service project and co-financed by the European Union with the European Regional Development Fund (Hauts-de-France region).

References

[1] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE.

[2] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT++: Better real-time instance segmentation," 2019. [Online]. Available: http://arxiv.org/abs/1912.06218

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[4] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," 2015. [Online]. Available: http://arxiv.org/abs/1405.0312

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," 2016. [Online]. Available: http://arxiv.org/abs/1604.01685

[6] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Chennupati, S. Nayak, S. Mansoor, X. Perroton, and P. Perez, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," 2019. [Online]. Available: http://arxiv.org/abs/1905.01489

[7] A. Saez, L. Bergasa, E. Lopez-Guillen, E. Romera, M. Tradacete, C. Gomez-Huélamo, and J. del Egido, "Real-time semantic segmentation for fisheye urban driving images based on ERFNet," vol. 19, no. 3, p. 503, 2019. [Online]. Available: http://www.mdpi.com/1424-8220/19/3/503

[8] K. Yang, X. Hu, L. M. Bergasa, E. Romera, and K. Wang, "PASS: Panoramic annular semantic segmentation," pp. 1–15, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8835049/

[9] G. Blott, M. Takami, and C. Heipke, "Semantic segmentation of fisheye images," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds. Springer International Publishing, 2019, vol. 11129, pp. 181–196. [Online]. Available: http://link.springer.com/10.1007/978-3-030-11009-3_10

[10] M. Christopher, "Laser-augmented omnidirectional vision for 3d localisation and mapping," Ph.D. dissertation, École Nationale Supérieure des Mines de Paris, 2009. [Online]. Available: https://pastel.archives-ouvertes.fr/pastel-00004652

[11] E. Sanz-Ablanedo, J. R. Rodriguez-Pérez, J. Armesto, and M. F. A. Taboada, "Geometric stability and lens decentering in compact digital cameras," vol. 10, no. 3, pp. 1553–1572, 2010. [Online]. Available: http://www.mdpi.com/1424-8220/10/3/1553

[12] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[13] S. A. Velastin and D. A. Gómez-Lira, "People detection and pose classification inside a moving train using computer vision," in *International Visual Informatics Conference*. Springer, 2017, pp. 319–330.

[14] N. Manovich and B. Sekachev, "Powerful and efficient computer vision annotation tool (cvat)," https://github.com/opencv/cvat, 2019.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

[16] D. Marcos, M. Volpi, and D. Tuia, "Learning rotation invariant convolutional filters for texture classification," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 2012–2017. [Online]. Available: http://ieeexplore.ieee.org/document/7899932/