

Industrial Fisheye Image Segmentation Using Neural Networks

A. Beloshapko[1(✉)], V. Korkhov[1], C. Knoll[2], and U. Iben[2]

[1] Saint-Petersburg State University, 7/9 Universitetskaya nab., St. Petersburg 199034, Russia
beloshapko-alexey@rambler.ru, v.korkhov@spbu.ru

[2] Robert Bosch GmbH, Robert-Bosch-Campus 1, 71272 Renningen, Germany
{Christian.Knoll,Uwe.Iben}@de.bosch.com

Abstract. Fisheye cameras have recently became very popular in computer vision applications due to their wide field of view. In addition to a better overview of the surrounding area, they enable to capture objects at extremely close ranges. These advantages come at a cost of strong image distortion, which cannot be removed completely maintaining image continuity. This complicates the use of traditional computer vision algorithms, which expect a single image as an input. This paper presents a performance evaluation of neural network algorithms for object detection and segmentation on fisheye camera images. Three approaches are evaluated: semantic image segmentation with Fully Convolutional Network (FCN) [13], a fully convolutional approach to instance segmentation with U-Net [18] and a region-based approach to instance segmentation with Mask R-CNN [10]. All of these networks successfully solved the task. However, as they were designed to different purposes, each of them has its own strengths and shortcomings. These three approaches are used to perform euro container image segmentation task. An image dataset was created in order to train and evaluate these algorithms. Huge part of this dataset was generated artificially, which simplified the task of ground truth labeling. The power of neural networks enable for fast and reliable image segmentation. As to our knowledge, this is the first neural networks application for euro container fisheye image detection and segmentation.

Keywords: Fisheye cameras · Euro container detection · Image semantic segmentation · Image instance segmentation · FCN · Mask R-CNN · U-Net

1 Introduction

Warehouse automation is an area with a huge potential. The use of robots that handle euro containers can both improve workplace ergonomics and safety, as well as reduce operational expenses. One major problem that has to be solved is the development of an automatic box picking and carrying system, which uses optical devices to detect, localize and track boxes using optical cameras as

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11622, pp. 678–690, 2019.
https://doi.org/10.1007/978-3-030-24305-0_50

sensors. Thus, there is a need for a reliable real-time box detection algorithm that will process images from cameras and locate objects of interest on them. Despite that, there are classical computer vision algorithms that are capable to detect objects of known shapes or properties; they are not robust enough to the appearance change or occlusion. The same applies to euro container objects. Introduction of a new type of a container may require the algorithm manual fine-tuning or even its complete reimplementation.

On the other hand, machine learning approaches offer a flexibility to tune the algorithm automatically when new object types become available. Use of machine learning based algorithm can make system setup much faster and more intuitive. The most promising type of machine learning algorithm is a convolutional neural network.

In this paper, we evaluate performance of three neural network-based approaches applied to the task of euro container image detection and segmentation. One of the methods does image semantic segmentation, while the other two perform image instance segmentation.

Semantic segmentation algorithm assigns a class label to each pixel on the image. Thus, an object can be detected as a continuous blob of pixels of the same class label. However, this approach cannot distinguish between different objects of the same class. This can give misleading results in case when more than one box is present on the same image. Instance segmentation algorithms additionally perform object detection, which enable to distinguish between multiple objects captured on the same image.

In this paper, we use Fully Convolutional Network (FCN) as a basis for semantic segmentation algorithm. Though some newer approaches exist (like DeepLab, [4], PSPNet [28]), it still outperforms many others in terms of speed (see [27]). U-Net and Mask R-CNN networks are evaluated to solve the instance segmentation task.

2 Related Work

There are several approaches to object detection in warehouse environment. In [14] Faster R-CNN neural network [17] is used for pallet detection on 2D range images. Master thesis of [21] describes an implementation of box detection and segmentation algorithm based on Point Cloud Library. Neither of them uses images acquired with fisheye camera, and thus impose a distance limitation on a system. Also, a typical object of interest of [21] has a relatively simple shape (a rectangular box with planar sides). These assumptions are violated in case of euro containers.

Fisheye cameras are a very popular choice in automotive applications, as they enable to capture the whole surrounding area with four cameras only. Autonomous driving task may benefit of urban scenes image segmentation. Authors of [19] developed a neural network algorithms applied to urban scenes fisheye image segmentation. Authors of [8] introduced Overlapping Pyramid Pooling and zooming image augmentation technique to train an OPP-net to

perform fisheye image segmentation of traffic scenes. These works are similar to ours because camera calibration is not assumed. In [7] a novel approach with deformable convolutions was used to solve a similar task. Deformable convolutions are meant to compensate for severe image distortion, but they has to be trained at each pixel location. We have shown that in the task of euro pallets image segmentation this feature is not strictly required.

There are several works related to fisheye image processing with neural networks. The work of [16] presents a fairly simple neural network architecture for robot navigation using a single fisheye camera. Works of [23] and [5] present approaches to spherical image processing with neural networks. These works require a calibrated camera, which imposes an additional limitation on the algorithm application.

As to our knowledge, this paper present a first attempt to use fisheye camera for object detection and segmentation in warehouses. We are not going to use any specialized network for fisheye image segmentation to avoid the need of camera calibration. Instead, we focus on using neural network algorithms that are initially designed to handle images captured with commonly used narrow-angle cameras.

There is a plenty of neural networks designed to handle object image segmentation. They fall into two distinct groups: neural networks for image semantic segmentation and neural networks for instance segmentation.

FCN is one of the simplest and most effective networks for image semantic segmentation. It uses a backbone classification network to create a coarse image segmentation map, and then iteratively upscales the result and refines it with skip-connections. SegNet network [1] improves this approach by introducing non-linear upsampling. Work of [26] introduced dilated convolutions to image segmentation neural network. Dilated convolutions replaced pooling layers typical to image classification network to enable dense multi-scale feature computation. In the work [3] this architecture was further developed by adding Fully Connected CRFs to refine the final segmentation result. PSPNet [28] used pyramid pooling to iteratively refine the segmentation mask, starting from high-level context and combining the intermediate result with lower-level features. The authors of [4] combined ideas of dilated convolutions, residual blocks and spatial pooling in a single network DeepLab v3. This network showed a great performance improvement over the original FCN network.

Instance segmentation neural networks additionally enable to distinguish between object instances of the same class. Most of them are region-based, that is, they use an object detection network like Faster R-CNN [17] to separate different object instances. One of the most notable implementation is the Mask R-CNN [10] which is based on Faster R-CNN object detection network and provides one additional output for dense image segmentation. The latest version of this network uses pyramid pooling as well. Some improvement was shown by the latest MaskLab [2] network, which is also built on top of the Faster R-CNN architecture, but also generates direction map, which enables to further separate objects from each other. Other region-based neural network architectures include SharpMask [15], OMN [9], RNN for Semantic Instance Segmentation [20].

On the other hand, there are several existing approaches that do not rely on a pre-generated object regions. U-Net [18] improves the idea of a Fully Convolutional Network with a more general skip-connection type and a weighted loss function that improves segmentation accuracy near image edges. Thus, distinct objects appear separated with a thin boundary. TernausNetV2 [11] adopts this idea to satellite imagery. Authors of [25] suggested to augment FCN architecture with the MaskSplitter strategy that penalize the loss when more than one object correspond to a single mask region (and vice versa).

The biggest challenge of the task of euro container image segmentation is to distinguish between an actual container and the industrial environment on the highest semantic scale. Both environment and the object of interest lack of visual features, contain a lot of parallel straight lines. In most of the cases, the object of interest is well aligned with the environment, which makes it hard to extract. Fisheye camera distortion introduces additional difficulty to the task.

We aim to overcome these problems with the use of neural networks, which can learn high-level features in an automatic manner. However, care should be taken in order to avoid model overfitting. Usually, this is done through either enhancement of the training dataset or restriction of the model structure.

Our target is to fulfill the following requirements:

- High box detection rate ($\geq$99%) and high segmentation accuracy ($\geq$0.95 IoU metric value)
- Fast inference (less than 1s per image on a GPU)
- Low memory consumption during training (11 GB or less)

We assume that objects of interest have a nearly-rectangular shape. Given these requirements, we have decided to avoid dilated convolutions and extensive pyramid pooling as they raise memory consumption during training. Also, CRF and RNN strategies could potentially violate the fast inference requirement. Thus, we evaluate three network architectures that are lightweight, fast and accurate:

- FCN for image semantic segmentation
- Mask-RCNN as a region-based instance segmentation network
- U-Net as a fully-convolutional instance segmentation network

3 Methods

3.1 Classification Neural Network

Each of the networks for image segmentation uses a classification network for image pre-processing. A typical classification network like AlexNet [12], VGG [22], or GoogLeNet [24] consist of two parts, e.g. Fig. 1. First part consists of blocks of convolutional layers followed by pooling layers. This block is connected to an input image. Convolutional layers compute feature maps of the image while

pooling layers reduce spatial dimensionality. Second part of a network consists of a stack of fully connected layers. It takes the last feature map from the lower part and process it to generate a classification hypothesis. The input of such a classification network is an image, and the output is usually a vector of scores for each of image classes.

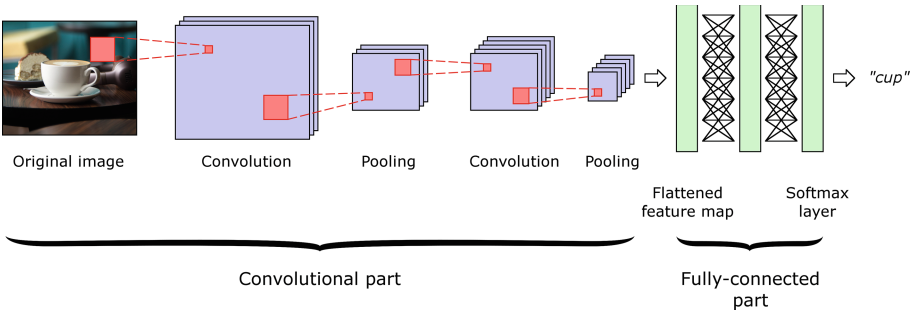

Fig. 1. Structure of a simple classification network which does cup image classification. This network consists of two convolution layers followed by spatial pooling, then two fully-connected layers and a softmax block. Blue rectangles represent 2D image feature maps, while green ones represent 1D layer activation (Color figure online)

3.2 Fully Convolutional Network

The Fully Convolutional Network [13] idea is fairly simple. First, consider a classification network that takes a small part of an image as an input and generates a score that a particular object is present on this image patch. A convolution of an input image with a classification network would generate a score map, e.g. Fig. 2.

This can be achieved by turning a classification network into a fully-convolutional one. Fully-connected layers of a classification network can be represented with convolutional layers with large kernel size.

FCN uses such a network to generate a raw image segmentation. Then, it additionally performs intelligent score map upsampling. It does a raw score map upsampling with a bilinear interpolation and combines upscaled map with a feature map generated by the classifier, e.g. Fig. 3 for an example. This operation is meant to propagate low-scale details to the upsampled score map and is repeated several times to achieve the same score map resolution as of the input image.

We use VGG-19 as a classification network to implement the FCN. The number of classes is reduced to two (object of interest and the background). The classification network weights are pre-trained on the imagenet [6] dataset.

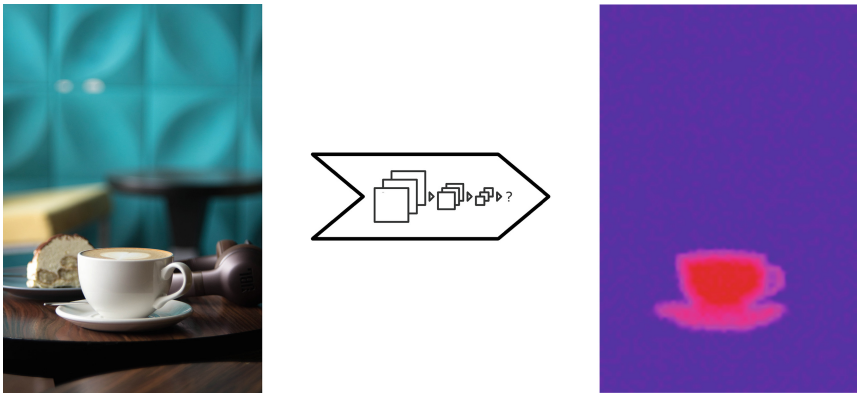

Fig. 2. Image convolution with a classification network generates a score "heatmap". Higher values (red) mean higher probability that an object of interest occupies this particular pixel. (Color figure online)

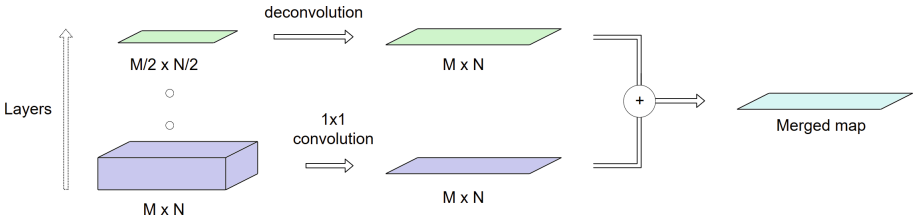

Fig. 3. A scheme of FCN skip-connection. When upsampling the raw score map, a classifier-generated feature maps are used to enhance localization. First, a low-level feature layer is convolved with a kernel of shape [$1 \times 1 \times$ channels] to reduce the number of channels to 1. The score map being upsampled is deconvolved with 2×2 deconvolution kernel. Finally, these score maps are combined elementwise. The resulting score map has double resolution while being more detailed than the original one.

3.3 U-Net

U-Net network architecture is very similar to the FCN one. A classification network is used in a fully convolutional manner to generate intermediate feature maps. The subsequent part performs an iterative upsampling combined with feature concatenation.

The main difference between U-Net and FCN architectures is a different skip-connection structure. While FCN keeps the number of channels of upsampled feature maps equal to one, U-Net directly concatenate features from the lower-part layer to the upper-part layer, thus increasing total number of output channels. This leads to a more general skip-connection structure.

The structure of our version of U-Net is shown on the Fig. 4. There are several changes to the original version, which were done to compensate for higher image resolution. The main changes to original U-Net network structure are:

- The encoder part of the network was modified so that VGG-19 network could be used as a backbone
- Total of 5 max pooling layers were used instead of 4
- One additional convolutional layer is used in each decoding block
- A pretrained VGG-19 network was used to initialize encoder weights
- Batch normalization layer was inserted after each non-initialized convolutional layer

3.4 Mask R-CNN

Mask R-CNN [10] is a region based network for instance segmentation. It uses the Faster R-CNN network for object detection prior to image segmentation. Figure 5 illustrates basic pipeline stages. It is as follows:

1. Compute a feature map with a classification network used in a fully convolutional manner (like the FCN network does)
2. Generate bounding box proposals. This is done via Faster-RCNN pipeline. First, a predefined set of anchor boxes are tested against "objectness" metric. Then, bounding box proposals are refined.
3. Perform classification and segmentation of each image patch defined by a bounding box proposals.

The result of this procedure is a set of classified and segmented image patches. These patches can possibly overlap while belonging to the same class. Thus, the network tries to distinguish between different instances of a single object class.

We used a Mask-RCNN implementation with Feature Pyramid Network integrated. The input to the FPN is downsampled to the size of 56×56 pixels (this technique is called "minimask"). This is done to limit memory consumption during training. The ResNet-101 was used as a classification backbone.

Table 1 summarises differences between these 3 networks.

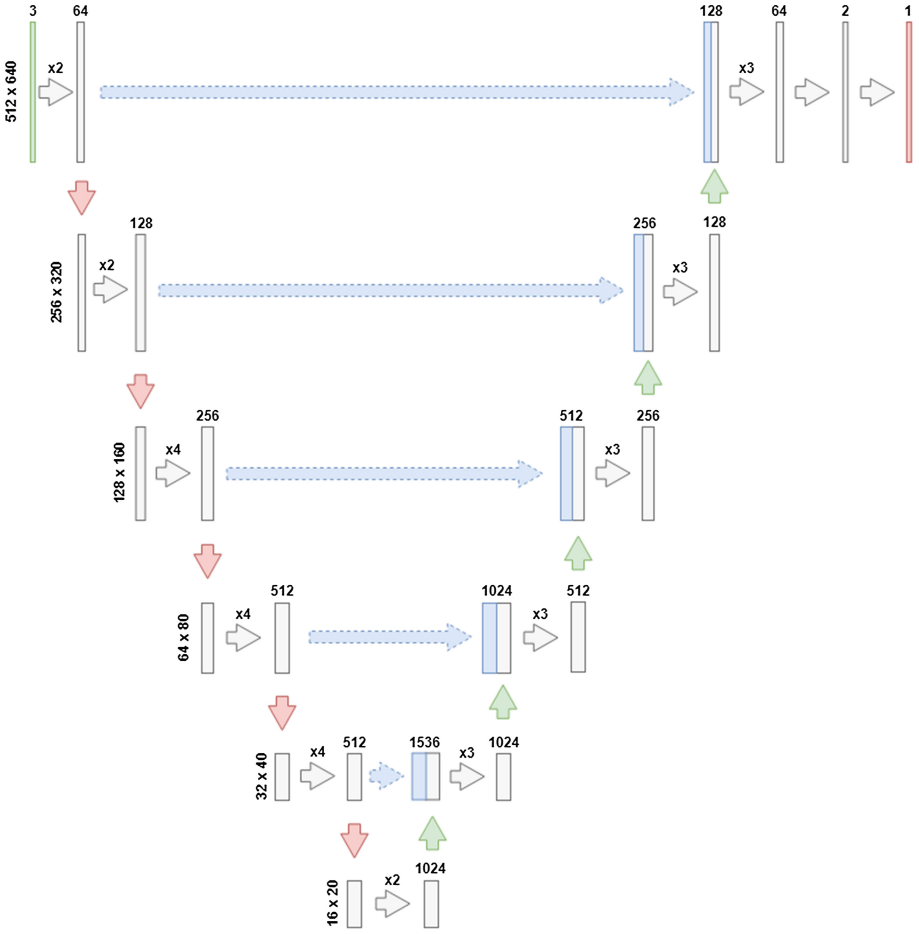

Fig. 4. A scheme of U-Net used in this paper. An input frame is a 3-channel image with resolution of 640×512 pixels. Red arrows depict downsampling procedure with the stride of 2, green arrows do 2×2 deconvolution. Gray arrows perform 3×3 convolution multiple times. Blue lines represent feature map concatenation (U-Net skip-connection). (Color figure online)

Table 1. Feature comparison of the three neural network architectures

Feature	FCN	U-Net	Mask R-CNN
Image segmentation	Yes	Yes	Yes
Object detection	No	Yes	Yes
Overlapping objects detection	No	No	Yes
Pixel-to-pixel segmentation	Yes	Yes	No (with minimask feature enabled)

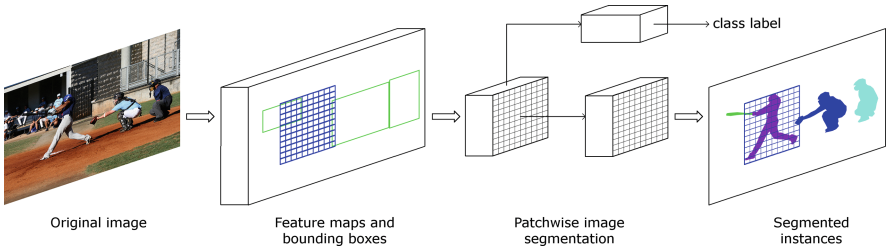

Fig. 5. Mask-RCNN pipeline. First, high-level feature maps are generated and bounding box proposals are computed. Then, feature maps get cropped and processed with two convolutional layers to classify and segment each image patch.

4 Dataset

The dataset consists of images of euro pallets captured in an industrial environment. We use synthetic images alongside with real images in order to simplify dataset labeling, e.g. Fig. 6 for an example.

Synthetic images were obtained by copying a real image of an object onto a different background image. First, the object was put in front of some simple background and captured. Second, a segmentation image was generated using morphology operation followed by a graph-cut based refinement. After this, the segmentation mask was used to crop out the object image to put it on some more sophisticated background.

Real-life images are captured in an industrial environment with boxes being close to the camera.

Training dataset consists of 9507 synthetic images and 199 real-live images. Test dataset is split into two parts: 1030 synthetic images and 72 real-life images. Each part is evaluated separately.

5 Experiments

The networks were trained on a machine with 4 Nvidia GeForce 1080 Ti GPUs. The input images have size of 512×640 pixels. It took around one day to fully train the FCN network while U-Net and Mask-RCNN took 3 days to train.

The accuracy of image segmentation was evaluated with the IoU metric. Assume that A represents the ground-truth entity of pixels that belong to box region, and B is the pixel entity predicted with a neural network. The IoU metric is calculated with the following formulae:

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

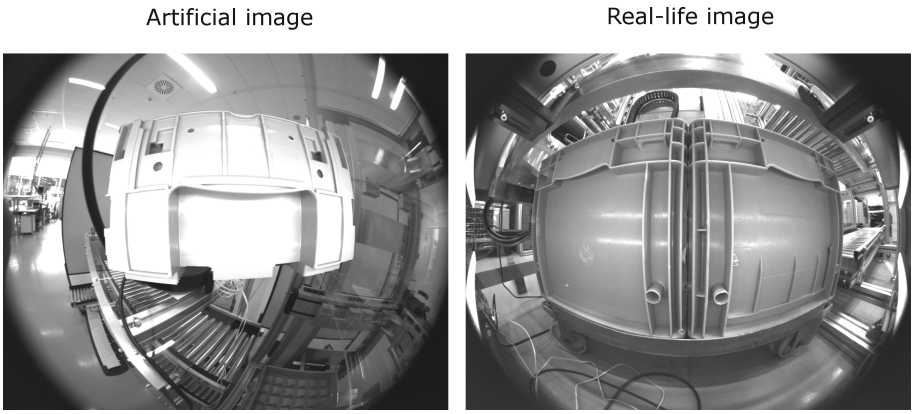

Fig. 6. An example of artificial and real-live test images

Thus, this metric captures how well the ground-truth and the predicted region overlap with each other. The metric is normalized to be in the range from 0 to 1, where higher is better.

In Table 2 you can see averaged IoU metric calculated over both synthetic and real-life parts of the test image set.

Table 2. Intersection over union score of the trained models evaluated on synthetic and real-life part of the test dataset

IoU	FCN	U-Net	Mask R-CNN
Synthetic images (1030)	0.9898	0.9921	0.9785
Real-life images (72)	0.9739	0.9544	0.9731

FCN and Mask R-CNN show comparable accuracy. The U-Net network shows better results on synthetic dataset, but performs poorly on real-life images. This might indicate overfitting of the U-Net network.

Figure 7 shows some cases of test images processing with these three networks.

Table 3 shows average processing time for each of the networks. FCN shows the best performance with processing more than 10 frames per second. Mask-RCNN is able to process around 5–6 frames per second. U-Net has the longest processing time.

Table 3. Average processing time for each of the networks

Time, seconds	FCN	U-Net	Mask R-CNN
Real-life images (72)	0.08785	0.31466	0.170926

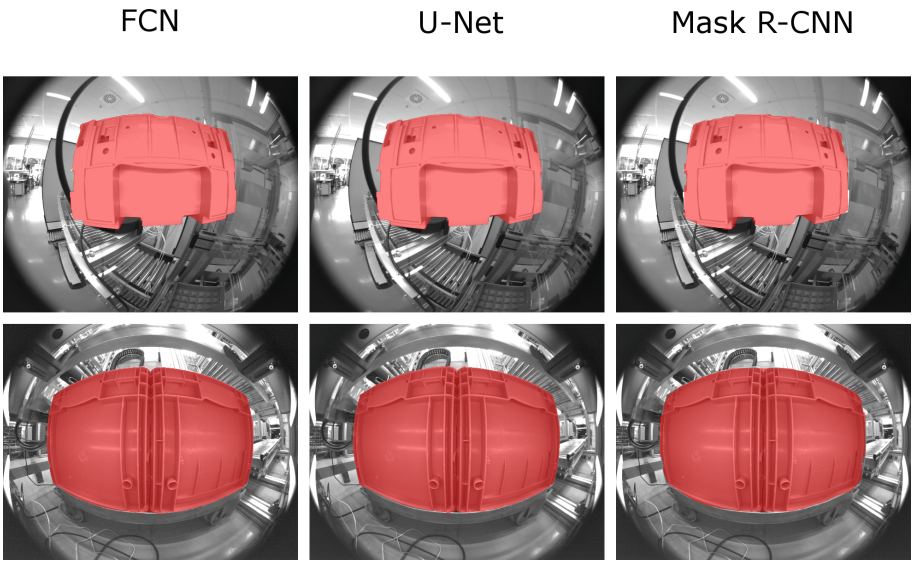

Fig. 7. Some examples of images processed with the trained models. Red area is an estimated box region. Upper row of images were obtained through processing an artificial image, while lower row contains processed real-life images. (Color figure online)

6 Discussion

All the networks reach the desired accuracy. FCN provides both best accuracy and computation time, but lacks of object detection capabilities. Mask-RCNN is twice slower than FCN and has slightly lower segmentation accuracy, but enables for object detection. U-Net based on VGG-19 has the worst performance.

Given that, it is better to use Mask-RCNN when object detection is required, and to use FCN for pure semantic segmentation tasks. U-Net architecture could be further improved to approach the accuracy of FCN network by changing its hyperparameters or by choosing a different backbone network.

7 Conclusion

All of the networks (FCN, Mask R-CNN and U-Net) reach the required performance and are able to handle fisheye image segmentation directly without any rectification. FCN provide box image segmentation of an excellent quality. Mask R-CNN can be used to additionally distinguish between several box instances. U-Net has to be further constrained to reach the performance of Mask R-CNN neural network.

Acknowledgments. This work is supported by the German-Russian Interdisciplinary Science Center (G-RISC) funded by the German Federal Foreign Office via the German Academic Exchange Service (DAAD).

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. CoRR abs/1511.00561 (2015). http://arxiv.org/abs/1511.00561
2. Chen, L., Hermans, A., Papandreou, G., Schroff, F., Wang, P., Adam, H.: Masklab: instance segmentation by refining object detection with semantic and direction features. CoRR abs/1712.04837 (2017). http://arxiv.org/abs/1712.04837
3. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. CoRR abs/1606.00915 (2016). http://arxiv.org/abs/1606.00915
4. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR abs/1706.05587 (2017). http://arxiv.org/abs/1706.05587
5. Coors, B., Condurache, A.P., Geiger, A.: Spherenet: learning spherical representations for detection and classification in omnidirectional images. In: European Conference on Computer Vision (ECCV), September 2018. http://www.cvlibs.net/publications/Coors2018ECCV.pdf
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR 2009 (2009)
7. Deng, L., Yang, M., Li, H., Li, T., Hu, B., Wang, C.: Restricted deformable convolution based road scene semantic segmentation using surround view cameras. CoRR abs/1801.00708 (2018). http://arxiv.org/abs/1801.00708
8. Deng, L., Yang, M., Qian, Y., Wang, C., Wang, B.: CNN-based semantic segmentation for urban traffic scenes using fisheye camera, pp. 231–236, June 2017. https://doi.org/10.1109/IVS.2017.7995725, https://www.researchgate.net/publication/318805934_CNN_based_semantic_segmentation_for_urban_traffic_scenes_using_fisheye_camera
9. Hayder, Z., He, X., Salzmann, M.: Shape-aware instance segmentation. CoRR abs/1612.03129 (2016). http://arxiv.org/abs/1612.03129
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. CoRR abs/1703.06870 (2017). http://arxiv.org/abs/1703.06870
11. Iglovikov, V.I., Seferbekov, S.S., Buslaev, A.V., Shvets, A.: Ternausnetv2: fully convolutional network for instance segmentation. CoRR abs/1806.00844 (2018). http://arxiv.org/abs/1806.00844

12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates Inc., New York (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
13. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. CoRR abs/1411.4038 (2014). http://arxiv.org/abs/1411.4038
14. Mohamed, I.S., Capitanelli, A., Mastrogiovanni, F., Rovetta, S., Zaccaria, R.: Detection, localisation and tracking of pallets using machine learning techniques and 2D range data. CoRR abs/1803.11254 (2018). http://arxiv.org/abs/1803.11254
15. Pinheiro, P.H.O., Lin, T., Collobert, R., Dollár, P.: Learning to refine object segments. CoRR abs/1603.08695 (2016). http://arxiv.org/abs/1603.08695
16. Ran, L., Zhang, Y., Zhang, Q., Yang, T.: Convolutional neural network-based robot navigation using uncalibrated spherical images. Sensors **17**, 1341 (2017)
17. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR abs/1506.01497 (2015). http://arxiv.org/abs/1506.01497
18. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. CoRR abs/1505.04597 (2015). http://arxiv.org/abs/1505.04597
19. Saez, A., Bergasa, L., Romeral, E., Guillén, M., Barea, R., Sanz, R.: CNN-based fisheye image real-time semantic segmentation, pp. 1039–1044, June 2018. https://doi.org/10.1109/IVS.2018.8500456
20. Salvador, A., Bellver, M., Baradad, M., Marqués, F., Torres, J., Giró i Nieto, X.: Recurrent neural networks for semantic instance segmentation. CoRR abs/1712.00617 (2017). http://arxiv.org/abs/1712.00617
21. Shipitko, O.: 3D pose estimation algorithm for intelligent box picking of warehouse automation robot (2018). https://www.researchgate.net/publication/325780223_3D_pose_estimation_algorithm_for_intelligent_box_picking_of_warehouse_automation_robot
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014). http://arxiv.org/abs/1409.1556
23. Su, Y., Grauman, K.: Flat2sphere: learning spherical convolution for fast features from 360° imagery. CoRR abs/1708.00919 (2017). https://arxiv.org/pdf/1708.00919.pdf
24. Szegedy, C., et al.: Going deeper with convolutions. CoRR abs/1409.4842 (2014). http://arxiv.org/abs/1409.4842
25. Ter-Sarkisov, A., Ross, R.J., Kelleher, J.D., Earley, B., Keane, M.: Beef cattle instance segmentation using fully convolutional neural network. CoRR abs/1807.01972 (2018). http://arxiv.org/abs/1807.01972
26. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. CoRR abs/1511.07122 (2015). http://arxiv.org/abs/1511.07122
27. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: ICNet for real-time semantic segmentation on high-resolution images. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 418–434. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_25
28. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. CoRR abs/1612.01105 (2016). http://arxiv.org/abs/1612.01105