








RGB-D-based Human Detection and Segmentation for Mobile Robot Navigation in Industrial Environments

Oguz Kedilioglu¹^a, Markus Lieret¹^b, Julia Schottenhamml²^c, Tobias Würfl²^d,
Andreas Blank¹^e, Andreas Maier²^f and Jörg Franke¹^g

¹*Institute for Factory Automation and Production Systems, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 91058 Erlangen, Germany*

²*Pattern Recognition Lab, Friedrich-Alexander- Universität Erlangen-Nürnberg (FAU), 91058 Erlangen, Germany*

Keywords: Image Segmentation, Object Recognition, Neural Networks, Deep Learning, Robotics, Autonomous Mobile Robots, Flexible Automation, Warehouse Automation.

Abstract: Automated guided vehicles (AGV) are nowadays a common option for the efficient and automated in-house transportation of various cargo and materials. By the additional application of unmanned aerial vehicles (UAV) in the delivery and intralogistics sector this flow of materials is expected to be extended by the third dimension within the next decade.


To ensure a collision-free movement for those vehicles optical, ultrasonic or capacitive distance sensors are commonly employed. While such systems allow a collision-free navigation, they are not able to distinguish humans from static objects and therefore require the robot to move at a human-safe speed at any time. To overcome these limitations and allow an environment sensitive collision avoidance for UAVs and AGVs we provide a solution for the depth camera based real-time semantic segmentation of workers in industrial environments. The semantic segmentation is based on an adapted version of the deep convolutional neural network (CNN) architecture FuseNet. After explaining the underlying methodology we present an automated approach for the generation of weakly annotated training data and evaluate the performance of the trained model compared to other well-known approaches.


1 INTRODUCTION


Within the last years collaborative robots and automated guided vehicles (AGV) found their way into industrial automation. As they share the working space with human workers, fencing become impractical and novel safeguarding technologies are required. While collaborative robots commonly provide inherent safety features such as force and momentum limitations or capacitive skins to detect approaching objects, AGVs capture their environment based on optical or ultrasonic sensors.


Those safeguarding technologies ensure a reliable collision avoidance but are based on the detection of any object within the robots working space. The more expedient approach is the detection of humans and an appropriate reaction of the robot based on the humans' movements and behaviour. Thereby non-human objects only need to be considered for collision avoidance but no additional safety distance or speed reduction is required when passing those objects. This applies also for the indoor navigation of UAVs that will be deployed inside of industrial production halls within the next decade.


To provide a next step to an environment-aware robot navigation we present an approach for the real-time human detection and segmentation in industrial environments. Based on the humans' three-dimensional point cloud representation resulting from the segmentation process an adaptive collision avoidance behaviour can be achieved.


^a  <https://orcid.org/0000-0002-3916-805X>


^b  <https://orcid.org/0000-0001-9585-0128>

^c  <https://orcid.org/0000-0002-9145-9102>

^d  <https://orcid.org/0000-0001-9086-0896>

^e  <https://orcid.org/0000-0002-5904-9680>

^f  <https://orcid.org/0000-0002-9550-5284>

^g  <https://orcid.org/0000-0003-0700-2028>

The paper is therefore structured as follows: Section 2 presents related research and existing solutions for the localization of human workers in industrial environments. Within section 3 we will dissociate our approach from existing solutions and present the underlying methodology and architecture. In section 4 we evaluate the reliability and accuracy of our approach and afterwards we discuss the results and outline the future research in section 5.

2 RELATED WORK

The reliable localization and tracking of humans for mobile robot navigation has already been the subject of various research projects and different approaches have been presented. The most common strategy is the equipment of workers with wireless transponders or optical markers that can be tracked by the application of suitable receivers and vision systems in combination with feature detectors and deep learning approaches.

Koch et al. (Koch et al., 2007) propose a localization method that relies solely on radio frequency identification (RFID) technology. Therefore about 4000 passive RFID tags are installed in the floor of a 60 m² test environment resulting in a grid density of 12.5 cm. To determine its current position, a wireless RFID reader is attached to the top of the humans foot. The resulting tracking accuracy is about 10 cm, however no precise localization is possible when the person walks too fast. Mosberger et al. (Mosberger and Andreasson, 2013) proposed the usage of a mono-camera system and reflective vests for human tracking adapted to industrial environments. A flash attached to the camera illuminates the scene and a Random Forest classifier is used to detect the reflections of the vests. Those reflections can be localized in 3D space within an accuracy in the decimeter range and a maximum detection distance of 10 meters.

However, whenever the human worker is not wearing his tracker or vest, he is not detectable by the robot and therefore exposed to potential harm. Therefore, different camera based solutions have been presented that allow a detection and pose determination of humans without any additional safety clothing or costly external sensor systems.

Munaro et al. (Munaro et al., 2015) present different RGB-D based human tracking pipelines for industrial environments. Using a combined method of applied consistency constraints, HOG-like descriptors and Haar-like feature extraction on disparity images they achieved a human detection with a false negative

rate of 21.95 % and a false positives per frames value of 0.17.

Different approaches focus on adaptive point cloud filtering to detect humans. After the removal of ground and ceiling plane the remaining objects are segmented based on their depth values and finally classified using a poselet-based human detector. Zhang et al. achieved the calculation of the human's bounding boxes at 7-15 frames per second with a false negative rate of approx. 3 % and a false positive rate of 2 %, based on the utilized dataset (Zhang et al., 2013). Shotton et al. achieve promising results on the real-time human pose recognition applied to single depth images. Using randomized decision trees and forests they reached a mean average precision of 0.731 on the locations of the individual body joints on a dataset of only human depth data without additional objects (Shotton et al., 2011).

Besides, it has also been shown that region of interest algorithms or the combination of depth-based shape detection, face and skin detection and motion estimation using reversible-jump Markov Chain Monte Carlo filters allow a reliable tracking of humans and are applicable for mobile robots ((Jafari et al., 2014), (Liu et al., 2015)).

While there are already many different approaches for the vision based human detection available, they commonly focus on the solely detection rather than the actual and precise segmentation. This renders them unsuitable for real-time navigation of mobile robots, thus, they do not work well in real-world industrial environments. To overcome this limitations we present in the following a real-time semantic segmentation of humans in industrial environments that is based on the convolutional neural network FuseNet. We investigate if using an RGB-D based segmentation model rather than a model that only takes color or depth information in account, improves segmentation results in commonly unicolored and evenly structured industrial environments.

3 RGB-D-BASED HUMAN SEGMENTATION

Object recognition methods are divided into four approaches. When the image is classified without determining the position of the individual objects, then the task is termed image classification. When the position of the objects is determined with a bounding box, then it corresponds to object detection. Semantic segmentation models return a semantic mask $M_{semantic}$ that assigns a semantic label, e.g. the label

'bottle' to each pixel of the input image. The semantic mask $\mathbf{M}_{semantic} \in \mathcal{L}_{semantic}^{H \times W}$ with image height H , image width W and semantic label-set $\mathcal{L}_{semantic} = \{0, 1, \dots, C\}$ where C is the number of semantic categories, is a 2-dimensional array with the pixel values designating the semantic category a pixel belongs to. Background pixels are indicated by 0. Instance segmentation returns separate labels for different instances of the same class. The instance mask $\mathbf{M}_{instance} = \{\mathbf{M}_i \in \mathcal{L}_{instance}^{H \times W} | i = 1, \dots, n\}$ with instance label-set $\mathcal{L}_{instance} = \{0, 1\}$ and number of instances n , is a set of 2-dimensional binary arrays \mathbf{M}_i where 0 denotes background pixels and 1 denotes object instance pixels ((Garcia-Garcia et al., 2017)).

3.1 FuseNet Architecture

To achieve a reliable and fast 3D segmentation we investigate using FuseNet (Hazirbas et al., 2016), an RGB-D semantic segmentation model that achieves competitive results compared to the state-of-the-art methods on the SUN RGB-D benchmark (Song et al., 2015) with 37.29 % mIoU. It has an encoder-decoder structure with two encoder branches (figure 1). One branch extracts features from the RGB image and the other one from the corresponding depth image.

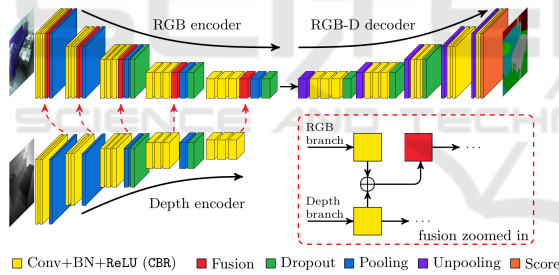


Figure 1: FuseNet architecture representing the encoder-decoder structure for RGB and depth images (figure adapted from (Hazirbas et al., 2016)).

The feature maps from the depth branch are fused into the feature maps of the RGB branch. This fusion layer is implemented as an element-wise summation of the two feature maps. The authors note that the depth information helps the model with the segmentation especially in cases where objects have similar color properties.

They provide a PyTorch implementation of their FuseNet architecture (Hazirbas et al., 2017), which we use for our analysis. They also provide the weights of their model that was pretrained with the NYU RGBD dataset (Nathan Silberman and Fergus, 2012), which consists of 1449 labeled RGBD images from indoor scenes. The original NYU dataset has 894 different categories, but Hazirbas et al. use only 40 cate-

gories to train their model. To map the original labels into 40 classes, they employ the mapping by (Gupta et al., 2013).

To optimize the pretrained model for the reliable segmentation of humans in industrial environments, we train the model with a task-specific, weakly annotated data set. The corresponding data acquisition process and the methodology used for the automated annotation of the captured images and details on the actual training process are presented in the following.

3.2 Data Acquisition

With an Intel RealSense D415 RGB-D camera around 20k images have been acquired. All images were recorded within the laboratories of the FAPS institute at the University of Erlangen-Nuremberg. It is filled with production machines, robots, tools, tables etc. Many small objects, buckets, boxes, shelves and machine parts help to simulate a challenging industrial environment. The prevailing colors are gray, beige and green, thus constituting a representative mixture of colors. One side of the hall consists of windows that are facing outside and covering more than half of the wall. This facilitates good illumination thanks to sunlight. But it has to be noted that this is not always the case in industrial environments. In order to simulate a more representative illumination situation the acquisition was realized in the morning and in the afternoon.

Besides the environment, the motion of the camera also plays an important role in determining the quality of the images. Therefore the movement velocities were varied and the camera motions comprise continuous steady paths as well as sideways turning and tilting motions to generate a challenging dataset. Fast motions lead to blurred images, which are nonetheless kept in the dataset in order to examine the behavior of the segmentation algorithms with difficult images. From scene to scene, the paths, the starting positions and the motion styles are varied to generate a diverse dataset. The resulting images are split into 11 scenes, each with approximately 2000 images. Each scene represents a coherent set of images representing a short video snippet with a duration of about 1 minute.

3.3 Training Data Preparation

As the manual annotation of the images would imply significant time investment and therefore is not expedient we make use of the fact, that color and depth image provided by the camera are aligned and of the same resolution. We therefore use a Mask R-CNN

model that is trained on the COCO dataset (Lin et al., 2014) to calculate segmentation masks for the individual humans that are solely based on the color image and afterwards additionally assign those masks to the depth image to generate the annotated RGB-D data set. As Mask R-CNN works with instance masks while FuseNet requires semantic masks, the masks have to be adapted, before they can be used for the training. Figure 2 shows an overview of the required steps needed to start training FuseNet.

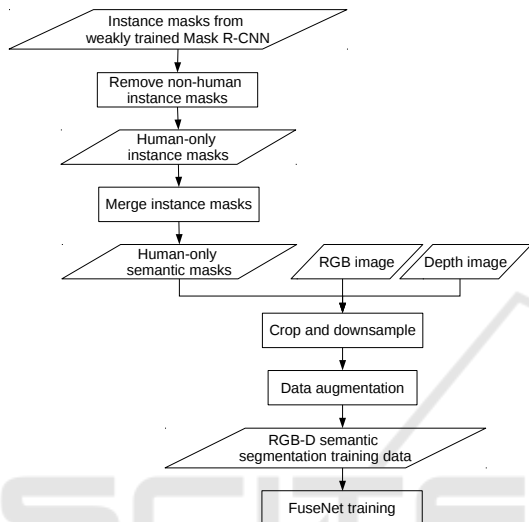


Figure 2: Preprocessing steps for FuseNet training. The weakly trained human only masks received from Mask R-CNN are multiplied with the RGB and depth images to create the required training data.

An instance mask $\mathbf{M}_{instance} = \{(\mathbf{M}_i, c_i) | \mathbf{M}_i \in \mathcal{L}_{binary}^{H \times W}, \mathcal{L}_{binary} = \{0, 1\}, c_i \in \mathcal{L}_{semantic} = \{0, 1, \dots, C\}\}$ has height H , width W , class c_i of instance i represented by single-instance mask \mathbf{M}_i and total number of classes C (0 corresponds to background). In our case $C = 80$, because of the COCO dataset Mask R-CNN was pretrained with.

As we only require a semantic class ('human'), which means that $\mathcal{L}_{semantic} = \mathcal{L}_{binary}$ we can extract the human-only semantic mask $\mathbf{M}_{semantic}$ by merging all single-instance masks \mathbf{M}_i that represent the category 'human'. The individual semantic segmentation masks $\mathbf{M}_{semantic}$ and the RGB-D images \mathbf{I}_{RGB-D} are then converted from their original resolution to the target resolution of 320×240 required by the FuseNet implementation. To retain the target image ratio the width of the original images and masks are cropped to 960 pixels before downsampling. This resizing of the images is computed dynamically during runtime and also includes a reshaping of the image's array representation to coincide with the Mask R-CNN implementation.

3.4 Data Augmentation

After the images and masks have been transformed into the right format their number can be increased by data augmentation methods. We apply two simple data augmentation methods, left-right flip and lower brightness to increase the model's invariance w.r.t. the location of humans in the image and disturbing effects in images. By applying different combinations of the two data augmentation methods we can quadruple the size of our dataset from 17,505 to 70,020 elements. The four different combinations of the two data augmentation methods, left-right flip and lower brightness, are 1) original dataset, 2) only left-right flip, 3) only lower brightness and 4) left-right flip and lower brightness. These four versions of our dataset are shuffled to create independent but identically distributed samples.

3.5 Training Details

In advance to the training process the PyTorch implementation has to be modified. The model originally provides 40 output channels which would require segmentation masks $\mathbf{M}_{semantic}$ with 40 categories for training. As our masks are binary we modified the last layer of FuseNet to only keep the first of the 40 output channels and to discard the remaining.

The total number of training samples is 70,020 and the total number of validation samples is 393. For the validation dataset data augmentation is not used. FuseNet uses VGG-16 (Simonyan and Zisserman, 2014) as backbone that was pretrained with the ImageNet dataset (Deng et al., 2009).

The model's output is a probability mask $\mathbf{M}_{sigmoid}$ with continuous values between 0 and 1, but for the localization of humans the output needs to be a binary mask $\mathbf{M}_{semantic}$. We therefore utilize hysteresis thresholding to maintain high confidence areas in the mask and to suppress the noise of low confidence pixels within these areas. We use the hysteresis thresholding implementation of the scikit-image library (van der Walt et al., 2014) with a low-threshold of 0.3 and a high-threshold of 0.7.

Overall, we trained FuseNet five times, each time with different training hyper-parameters, but always starting from the same initial NYU-pretrained model, resulting in 5 different model versions. For all model versions stochastic gradient descent is utilized as the learning algorithm with momentum, weight decay and learning rate decay. Table 1 lists all applied training hyper-parameters. The best results were achieved with the parameter set used for model version 3 which has a mean validation loss value of 0.041 and will

Table 1: FuseNet training hyper-parameters used for the different training versions. To achieve optimal results initial learning rate, learning rate decay, frequency of learning rate decay, epochs and batch size were evaluated.

Model Version	Initial Learning Rate	Learning Rate Decay	Frequency of Learning Rate Decay after	Epochs	Batch Size
1	0.001	0.9	10,000 iter.	3	1
2	0.001	0.5	10,000 iter.	7	2
3	0.01	0.6	10,000 iter.	8	2
4	0.1	0.5	10,000 iter.	7	2
5	0.001	0.5	50,000 iter.	7	2

therefore be used for the subsequent evaluation. Version 4 has the worst performance with a mean validation loss of 0.071.

Figure 3 shows the training loss function as binary cross entropy (BCE) loss after the individual iterations while Figure 4 shows the corresponding validation loss.

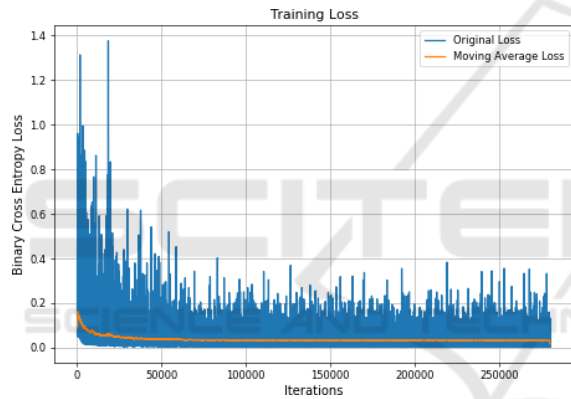


Figure 3: Loss of model version 3 on the training set after each iteration.

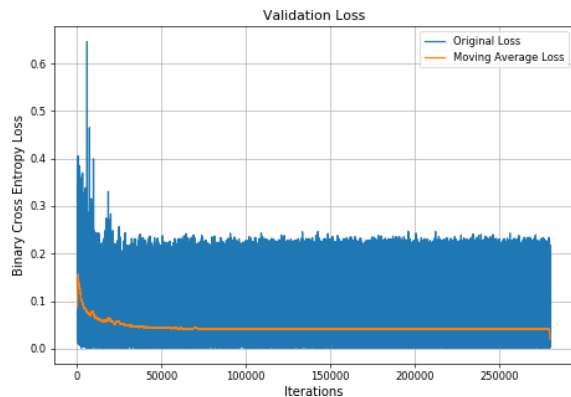


Figure 4: Loss of model version 3 on the validation set after each iteration.

Both training and validation loss decrease rapidly at the beginning and settle at a constant value range after approx. 50000 iterations. After approx. 100000

iterations the moving average loss for the training reaches a constant value of 0.038 while the validation loss moving average stays constant at a value of 0.040.

3.6 3D Point Cloud Generation

The 3D localization of humans determines their (x, y, z) position relative to the camera mounted on the UAV or AGV. Thereby the human is represented by a 3D point cloud consisting of points that belong to humans which is created by combining the segmentation mask and the depth image.

FuseNet takes the whole RGB-D image $\mathbf{I}_{RGB-D} \in \{(\mathbf{I}_{RGB}, \mathbf{I}_D) | \mathbf{I}_{RGB} \in \mathbb{N}^{H \times W \times 3}, \mathbf{I}_D \in \mathbb{N}^{H \times W}\}$ with image height H , image width W and 3 color channels as input. The pixel values in the depth image \mathbf{I}_D thereby correspond to the distance in millimeters between objects and the camera. The mask $\mathbf{M} \in \mathcal{L}^{H \times W}$ with labels $\mathcal{L} = \{0, 1\}$ returned by the segmentation model is a 2-dimensional array, where pixels that represent a human have label 1 and all the other pixels have label 0. This mask is multiplied elementwise with the depth image \mathbf{I}_D and the resulting human-only depth image \mathbf{I}_{Dh} is then transformed into a point cloud \mathbf{I}_{PC} by using the camera's inverted intrinsic matrix \mathbf{K}^{-1} .

The resulting point cloud \mathbf{I}_{PC} allows the differentiation between humans and the static environment and can be used as input for human-aware and velocity constrained path and motion planning as proposed by Sisbot et al. (Sisbot et al., 2007) or Collins et al. (Shi et al., 2008). Such path planning solutions allow a faster movement of the individual robots while humans in the environment still feel comfortable when the robot passes them with reduced velocity.

4 EVALUATION OF THE SEGMENTATION RESULTS

In this section we evaluate the different segmentation models on our dataset. The performance of the individual segmentation models is evaluated based on the mean Intersection over Union (mIoU) metric and the inference runtime speed. The corresponding values for the compared models are listed in Table 2 below. The mIoU values are calculated w.r.t. 30 manually annotated ground truth masks, the used desktop computer contains a i7-8700 CPU, 32 GB RAM and a GTX 1080 GPU, while the laptop contains a i5-7200U CPU, 16 GB RAM and no additional GPU. The entire implemented software stack is based on the Robot Operating System (ROS), a widely used soft-

ware framework which offers solutions for common tasks in robotics applications (Quigley et al., 2009).

Table 2: Runtime in milliseconds for CUDA-GPU desktop computer (GPU) and CPU-only laptop (CPU). mIoU values for segmentation models.

Model	CPU Time	GPU Time	mIoU
Mask R-CNN	13103	820	0.5871
DeepLab-v3+	6845	389	0.6098
DeepLab-v3+ MobileNet	637	79	0.5139
FuseNet	-	37	0.6244

The segmentation masks calculated by our trained FuseNet model (figure 6) provide a slightly higher accuracy compared to the ones calculated with the publicly available and pretrained models of Mask R-CNN (He et al., 2017), DeepLab-v3+ and DeepLab-v3+ MobileNet (Chen et al., 2018), shown in figure 5.

While it has been shown that the adaption of the original FuseNet model to industrial environments is possible and shows promising results, it must be mentioned that the referenced models solely use color information and a further comparison with different RGB-D segmentation models is required. When comparing the computing time, it must also be taken into account that FuseNet works with a significantly lower resolution than the other networks (FuseNet: 320x240, Mask R-CNN: 1024x1024 DeepLab-v3+ (MobileNet): 512x288), which results in a direct reduction of the computing time.

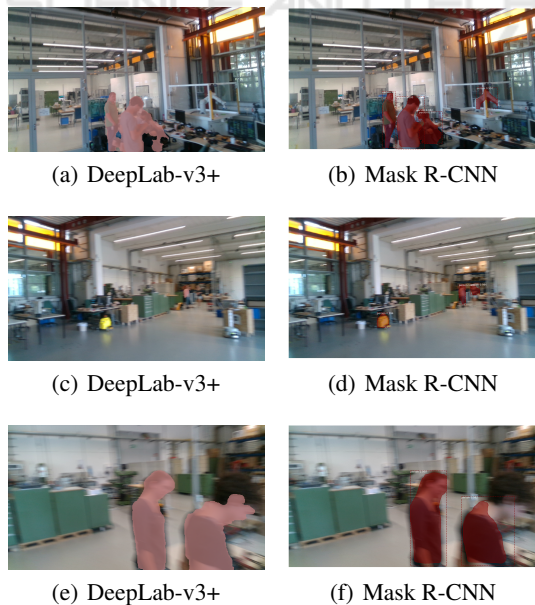


Figure 5: Segmentation results of DeepLab-v3+ and Mask R-CNN.

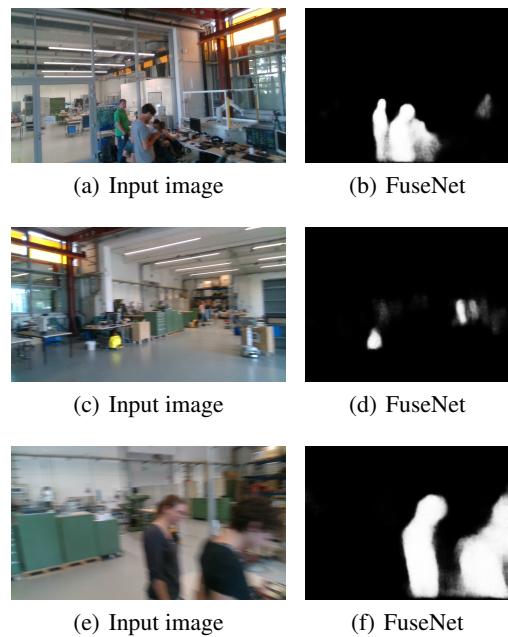


Figure 6: Segmentation results of FuseNet, sigmoid output without threshold.

The effects of the hysteresis threshold on the FuseNet output can be seen in figure 7. For example, in the images of the first row the robot arm disappears from the mask thanks to thresholding.

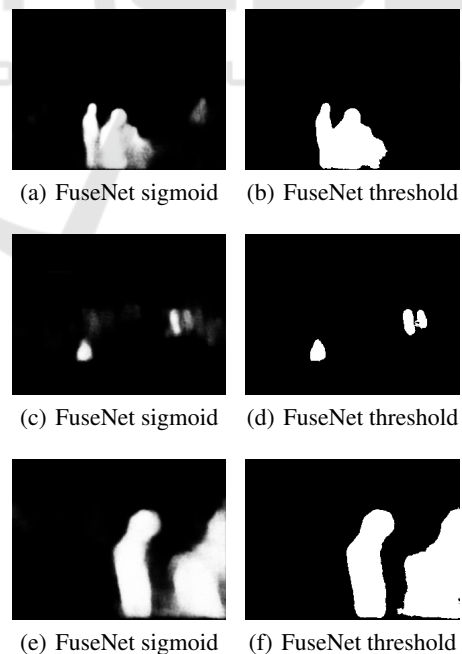


Figure 7: FuseNet comparison of sigmoid output and threshold output.

5 CONCLUSION AND OUTLOOK

Within this paper, we presented an approach to use FuseNet for the 3D segmentation of humans in industrial environments to create corresponding point clouds for the adaptive path planning of mobile robots. To automatically generate the training data annotations required for the FuseNet training, Mask R-CNN pre-trained with the COCO dataset was used to annotate the color and depth information acquired and superimposed by the camera using the segmentation masks calculated from the color image. On an evaluation dataset with manually annotated ground truth masks, our trained FuseNet model was able to achieve a higher mean intersection over union value at a reduced computing time compared to competing pre-trained segmentation models. Due to the low computation time and good recognition quality, the model is suitable for real-time 3D segmentation of persons to enable human-aware path planning for mobile robots from the resulting point cloud.

For future improvements, we intend to compare our model which was trained using a weakly supervised strategy with the NYU trained FuseNet model which is trained on expensive groundtruth annotation. Additionally we are interested in evaluating CNNs like DA-RNN (Xiang and Fox, 2017) and STD2P (He et al., 2016) which take the temporal aspect of the data into account, as tracking humans from frame to frame instead of segmenting each frame in an isolated way could yield better results.

ACKNOWLEDGEMENTS

The project receives funding from the German Federal Ministry of Education and Research under grant agreement 05K19WEA (project RAPtOr).

REFERENCES

- Chen, L., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611.
- Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., and Rodríguez, J. G. (2017). A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857.
- Gupta, S., Arbeláez, P., and Malik, J. (2013). Perceptual organization and recognition of indoor scenes from rgb-d images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571.
- Hazirbas, C., Ma, L., Domokos, C., and Cremers, D. (2016). Fusetnet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian Conference on Computer Vision*.
- Hazirbas, C., Ma, L., Domokos, C., and Cremers, D. (2017). Fusetnet. https://github.com/zanilzanza/FuseNet_PyTorch.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.
- He, Y., Chiu, W., Keuper, M., and Fritz, M. (2016). RGBD semantic segmentation using spatio-temporal data-driven pooling. *CoRR*, abs/1604.02388.
- Jafari, O. H., Mitzel, D., and Leibe, B. (2014). Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In *In ICRA*.
- Koch, J., Wettach, J., Bloch, E., and Berns, K. (2007). Indoor localisation of humans, objects, and mobile robots with rfid infrastructure. In *7th International Conference on Hybrid Intelligent Systems (HIS 2007)*, pages 271–276.
- Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.
- Liu, J., Liu, Y., Zhang, G., Zhu, P., and Qiu Chen, Y. (2015). Detecting and tracking people in real time with rgb-d camera. *Pattern Recognition Letters*, 53:16–23.
- Mosberger, R. and Andreasson, H. (2013). An inexpensive monocular vision system for tracking humans in industrial environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 5850–5857.
- Munaro, M., Lewis, C., Chambers, D., Hvass, P., and Menegatti, E. (2015). Rgb-d human detection and tracking for industrial environments. In *Intelligent Autonomous Systems 13*, pages 1655–1668.
- Nathan Silberman, Derek Hoiem, P. K. and Fergus, R. (2012). Indoor segmentation and support inference from rgb-d images. In *ECCV*.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.
- Shi, D., Collins Jr, E. G., Goldiez, B., Donate, A., and Dunlap, D. (2008). Human-aware robot motion planning with velocity constraints. In *2008 International Symposium on Collaborative Technologies and Systems*, pages 490–497.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304.
- Simonyan, K. and Zisserman, A. (2014). Very deep con-

- volutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.
- Sisbot, E. A., Marin-Urias, L. F., Alami, R., and Simeon, T. (2007). A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883.
- Song, S., Lichtenberg, S. P., and Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576.
- van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ*, 2:e453.
- Xiang, Y. and Fox, D. (2017). DA-RNN: semantic mapping with data associated recurrent neural networks. *CoRR*, abs/1703.03098.
- Zhang, H., Reardon, C., and Parker, L. E. (2013). Real-time multiple human perception with color-depth cameras on a mobile robot. *IEEE Transactions on Cybernetics*, 43(5):1429–1441.

