Accurate semantic segmentation of RGB-D images for indoor navigation

Sudeep Sharan,[a,b,*] Peter Nauth,[a] and Juan-José Domínguez-Jiménez[b]
[a]Frankfurt University of Applied Sciences, Faculty of Computer Science and Engineering, Frankfurt am Main, Germany
[b]University of Cadiz, School of Engineering, Department of Computer Science and Engineering, UCASE Software Engineering Group, Cadiz, Spain

Abstract. We introduce an approach of semantic segmentation to detect various objects for the mobile robot system "ROSWITHA" (RObot System WITH Autonomy). Developing a semantic segmentation method is a challenging research field in machine learning and computer vision. The semantic segmentation approach is robust compared with the other traditional state-of-the-art methods for understanding the surroundings. Semantic segmentation is a method that presents the most information about the object, such as classification and localization of the object on the image level and the pixel level, thus precisely depicting the shape and position of the object in space. In this work, we experimented with verifying the effectiveness of semantic segmentation when used as an aid to improving the performance of robust indoor navigation tasks. To make the output map of semantic segmentation meaningful, and enhance the model accuracy, points cloud data were extracted from the depth camera, which fuses the data originated from RGB and depth stream to improve the speed and accuracy compared with different machine learning algorithms. We compared our modified approach with the state-of-the-art methods and compared the results when trained with the available dataset NYUv2. Moreover, the model was then trained with the customized indoor dataset 1 (three classes) and dataset 2 (seven classes) to achieve a robust classification of the objects in the dynamic environment of Frankfurt University of Applied Sciences laboratories. The model attains a global accuracy of 98.2%, with a mean intersection over union (mIoU) of 90.9% for dataset 1. For dataset 2, the model achieves a global accuracy of 95.6%, with an mIoU of 72%. Furthermore, the evaluations were performed in our indoor scenario. © *2022 SPIE and IS&T* [DOI: 10.1117/1.JEI.31.6.061818]

Keywords: semantic segmentation; machine learning; object detection; mobile robot navigation; robot vision.

Paper 220231SS received Mar. 2, 2022; accepted for publication Jun. 29, 2022; published online Aug. 3, 2022.

1 Introduction

In recent years, robotics has emerged as a fast-developing field in industrial and research contexts. The advancement in the hardware accelerators in many fields, such as graphical processing units (GPUs) and tensor processing units (TPUs), and the algorithms in computer vision, machine learning, artificial intelligence (AI), notably in deep learning fields have achieved many remarkable breakthroughs.[1,2]

Recognizing the potential, researchers recently have tried to corporate components of deep learning in the computer vision field. Their objective is to enhance robots' abilities, namely the ability to see, feel, and understand the real world like a human, through sensors, actuators, and cameras. Self-driving automobile companies, such as Tesla and NVIDIA,[3] are examples of an industry trying to make their machines more intelligent by powering them with AI to make them fully discernible regarding surrounding objects and to navigate and localize themselves in new environments.

*Address all correspondence to Sudeep Sharan, s.sharan@fb2.fra-uas.de

Indoor navigation is one of the fields that has lots of room for improvement. Indoor navigation requires a robot to fully understand and precisely navigate in a complex environment while avoiding different types of obstacles and finding the correct navigating paths.[4]

A conventional approach uses the two-dimensional (2D) laser scanner or the ultrasonic sensor. It allows for detecting obstacles around robots, from which a suitable navigation path can be calculated to avoid objects/obstacles. However, these sensors only detect the environment on a flat plane at the sensor level. Therefore, these sensors have some limitations, such as not being able to recognize all types of obstacles/objects. For example, a 2D laser scanner can only detect four legs of a table but not the whole surface of the table and has difficulty navigating paths (for instance, 2D laser scanner cannot see that there is a staircase in front of it). Not only are such devices limited with the scanning ability, but they are also unable to tell the objects' classes, which is a significant drawback for complex tasks that require flexibility, such as path planning when encountering different types of obstacles.

To deal with such issues, rich visual and space information retrieved from the RGB-D camera can be used to make the robots fully aware of the indoor spaces and obstacles. The robots can react differently from these data when encountering static obstacles versus dynamic obstacles. Semantic segmentation is one of many solutions that we suggest to serve this task.[5] In this paper, Sec. 2 discusses the related works. Section 3 presents the model overview. Section 4 presents the implementation and testing. Section 5 presents the results and the discussions of the experiments. Finally, conclusions and future improvements are given in Sec. 6.

2 Related Work

Merging visual data and depth data streams to improve the accuracy of the RGB-D image semantic segmentation algorithms is not a new approach in the research field. Because depth images are invariant to lighting conditions, they will help the model detect objects in extreme and variant lighting environments.

Traditional RGB-D image segmentation[6] uses the basic features of the image, such as contour and color homogeneity, to divide images into smaller clusters of superpixels regions. Regional higher-level features can be extracted via handcrafted filtering operations before being fed into a traditional machine learning classifier such as a support vector machine to classify the class of each region. Afterward, a further post-processing algorithm, such as conditional random field, is run over the image to refine the output of the classification stage. Finally, the segmented regions with corresponding labels are the output. Reliably implementing tradition segmentation in real-world or industrial scenarios is more complicated due to the involvement of multiple separate stages, as well as the fact that the filtering operations during the feature extraction stage are mostly handcrafted, thus leading to the liability of not being able to capture all of the data presented in the image and not being robust to different environmental scenarios. Also, the classifiers in such methods are naive. Finally, by dividing the image into multiple different regions, the algorithms cannot learn the interconnected features between different adjacent regions.

Image segmentation with deep learning models[7] uses new deep learning techniques to produce the segmentation map directly, without iterating through the segmenting, feature extracting, classification, and refining steps compared with traditional ways. Such deep learning methods commonly utilize the powerful learning capability of convolutional neural networks (CNNs). In addition, the data from the RGB camera and depth camera are usually extracted in different streams (encoder branches) before being fused in later steps.

Segmentation with deep learning models executes most of the stages in a one-shot fashion, leading to higher flexibility and feasibility for implementation. In addition, we have found and tested that CNN models tend to outperform traditional computer vision methods due to their ability to generalize different types of images and capture low- and high-level (features from faraway regions on the images) precisely and automatically. Therefore, deep learning methods are becoming the leading research direction of robust object detection. Some notable works that represent this trend are fused squeeze-and-excitation network (FuseNet),[8] multi-modality and multi-scale attention fusion network (MMAF-Net),[9] and attention complementary network.[10] However, some drawbacks include the high number of resulting hyperparameters cannot be

implemented into mobile and embedded devices,[9] they are prone to overfitting, they requiring more time and power to train, and non-optimal approaches leverage the depth data into the deep learning model.

In this work, we evaluate the viability of applying an efficient architecture in a real robot to solve the following challenges:

R1 The robot distinguishes the surrounding obstacles by implementing a machine learning model with a lightweight size (with better accuracy and precision); it is easy to train and moves in real time.

R2 The robot operates under various indoor light conditions.

3 Model Overview

The backbone model that we use is fully connected harmonic DenseNet[11] (FC-HarDNet), an upgraded version of the fully connected DenseNet model.[12] Because the main source code is written in the Pytorch framework, which is used mainly in research, we rewrote the code in the TensorFlow framework to make it easier to deploy on a TPU for training and to be optimized on a GPU for inference. There are some advantages of applying FC-HarDNet:

- It has high efficiency on the benchmark outdoor dataset with IoU score up to 75% and speed up to 75 FPS.[11]
- It is relatively small size to avoid overfitting in our Frankfurt University of Applied Sciences (FRA-UAS) indoor dataset.
- To tackle the issue of vanishing gradient while maintaining the depth,[13] it employs the skip connection wisely. Skip connection is a concatenation or elementwise addition of layers, allowing for gradients of later layers to flow directly to earlier layers without getting attenuated.
- With the reordering of the skip connections resembling the harmonic waves, the authors of FC-HarDNet prove that such organization would lead to faster input/ouput operation on the dynamic random access memory (DRAM), resulting in a faster processing speed of up to 36%. The author also injected a parameter k to control the depth of each convolution layer in the HarDBlock.

To fuse the data from the RGB camera and depth camera, we used the architecture from MMAF-Net,[9] in which there are two separate encoders to extract data from two cameras, and we fuse them before each downsampling stage to create a main branch. The main branch is then used to upsample the feature map back to the original. Our model used a slightly different attention module from the MMAF-Net. "The attention modules are used to help any neural network models to learn and focus on the most important information rather than irrelevant background data. For example, the human brain tends to focus on specific part of the image and view the non-useful information in ways that may aid perception."[14,15] Our attention module is taken from the Google brain's image classification model, EfficientNet.[16] EfficientNet makes use of the attention mechanism (or so-called squeeze–excitation network[17]), in which the module helps the model know which features are useful and which features are not for the learning process by multiplying an added weight to each feature map. The weight will be zero if the feature map is useless and positive or negative depending on the usefulness of the feature map. Figure 1 shows the overall architecture of our models.

4 Implementation and Testing

Our model is implemented to guide a mobile robot system named "ROSWITHA" (RObot System WITH Autonomy)[18] by running the robot randomly in the Laboratory of Autonomous Systems and Intelligent Sensors at FRA-UAS, Germany. By segmenting captured images into classes, we differentiate between objects that are static or dynamic. Figure 2 shows the structure of the ROSWITHA robot and the mounted Intel RealSense depth camera.

Before feeding our depth data into the model, we first converted the depth image into a point cloud. The reason for this is that the model can infer more features from the environment by

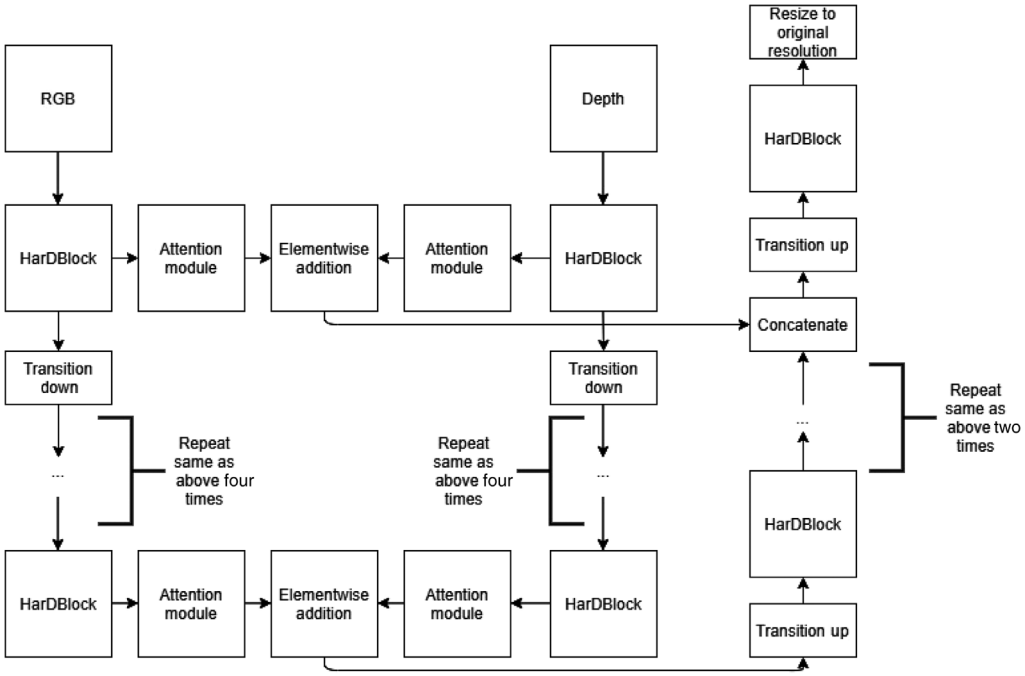

Fig. 1 Overall architecture of our modified RGB-D semantic segmentation combined with the attention module.

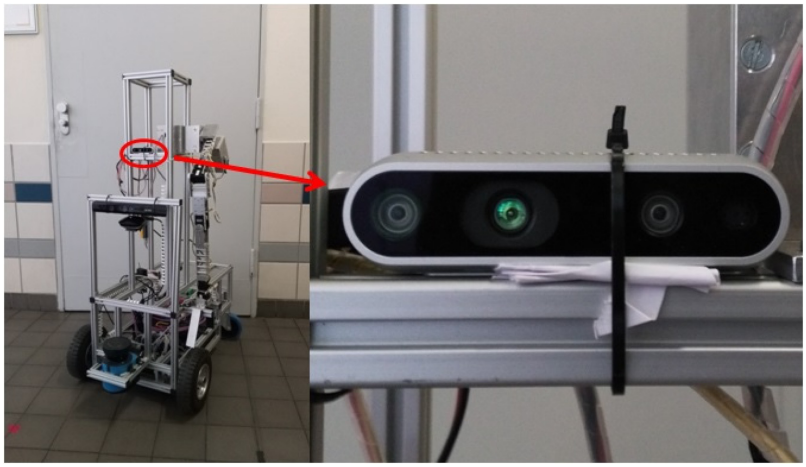

Fig. 2 ROSWITHA robot equipped with an Intel RealSense depth camera (located in red box).

having point cloud data. For example, walls tend to have the same relative horizontal distance to the camera, whereas the ceiling tends to have a higher height than the camera. Meanwhile, suppose we use only the original depth image. In that case, the Euclidian distance from the point to the camera or the distance of the projection of the point onto the center axis of the camera can be extracted. To convert into point clouds, we use the following imaging model[19]:

$$x = (x_d - cx_d) * \frac{\text{depth}(x_d, y_d)}{fx_d}, \tag{1}$$

$$y = (y_d - cy_d) * \frac{\text{depth}(x_d, y_d)}{fy_d}, \tag{2}$$

$$z = \text{depth}(x_d, y_d), \tag{3}$$

where x, y, and z are the real-world coordinates; (x_d, y_d) are the pixel coordinates; cx_d and cy_d are the pixel coordinate of the principal point (center of projection); fx_d and fy_d are the focal length of the image. The multiplication of the pixel width, and depth (x_d, y_d) is the real-world depth at the pixel coordinate (x_d, y_d). This imaging model improves our model mean intersection over union (mIoU) from 0.31 to 0.40. Because the JPG image file extension only allows for positive integers, whereas the point clouds are real and float numbers, our pipeline also scales the point clouds' frames back into integers in the range [0,255] to save them as images and save the disk memory. This also offers an advantage that the model will generalize the relative position of points in space rather than learning their exact coordinates.

Our data came from two main sources: an external synthetic indoor RGB-D dataset with a segmentation label and our generated dataset at FRA-UAS laboratories. The reason for choosingthe first source is to give our model good initial weights, which would bring good results when we apply it to retrain with our custom dataset. The synthetic datasets have been proven to work in many computers vision tasks[20] and are more precise than human-labeled ones. The dataset that we used is SceneNet RGB-D,[21] which has up to 5 million images in 14 classes [background, bed, books, ceiling, chair, floor, furniture, objects, picture, sofa, table, television (TV), wall, and window] at a resolution of 240×320. Some examples of the SceneNet RGB-D dataset are shown in Fig. 3.

We generated a dataset at FRA-UAS laboratories. SceneNet does not include some classes that we are interested in, such as humans, backpacks, robots, and obstacles in general. Meanwhile, there are many irrelevant classes, such as bed, toilet, or ceiling. The FRA-UAS laboratories environment is much different from the scenarios presented in the SceneNet dataset. Therefore, to make the model work well with our scenarios, we must collect more images around robotics laboratories. The images were gathered by a D435 Intel RealSense attached to the mobile robot ROSWITHA. The images were captured at the resolution of 640×480 at 90 FPS, and the point clouds converting speed was 30 FPS. The robot was driven around the corridor and the robotics laboratories on the second floor of building eight at FRA-UAS, Germany, to collect the images. The generated dataset is divided into two subdatasets.

Fig. 3 Examples synthetic images from the SceneNet dataset (from top to bottom: RGB, depth, and label images).

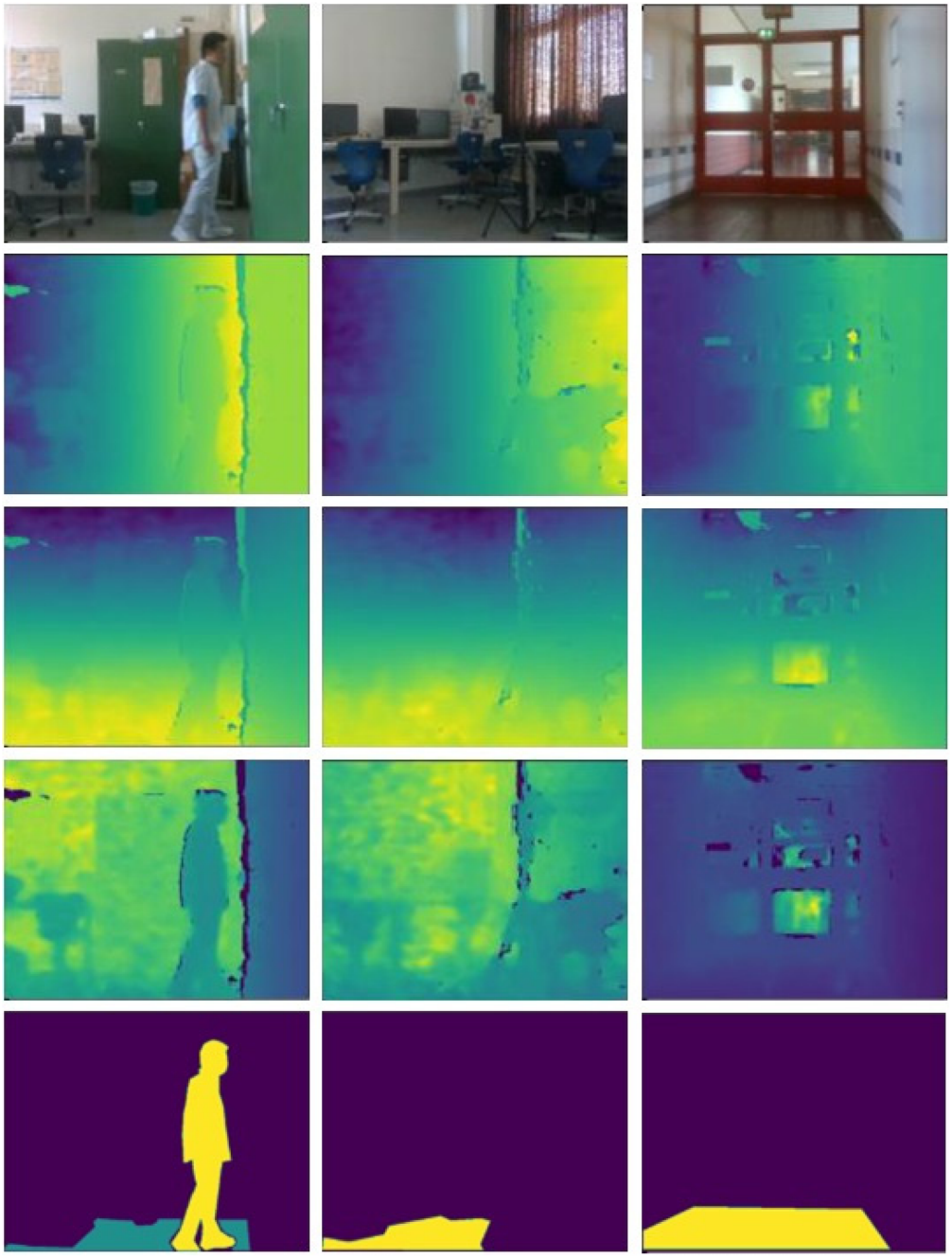

Fig. 4 Three-classes FRA-UAS dataset (from top to bottom: RGB, x, y, and z coordinates, and labeled images).

The first dataset has ∼1400 images of three classes: "obstacles," "free space," and "human" (Fig. 4). These images are created and then tested on the performance of our model in the real world. After achieving better results, we increased the number of classes. We created the second dataset with ∼390 images of seven different classes, including "free space," "human," "chair," "table," "robot," "backpack," and "other obstacles" (Fig. 5).

The model training process is divided into three stages:

- Training the model with the SceneNet dataset to give good initial weights to the model.
- Training the model with the FRA-UAS dataset 1 with three classes.
- Training the model with the FRA-UAS dataset 2 with seven classes.

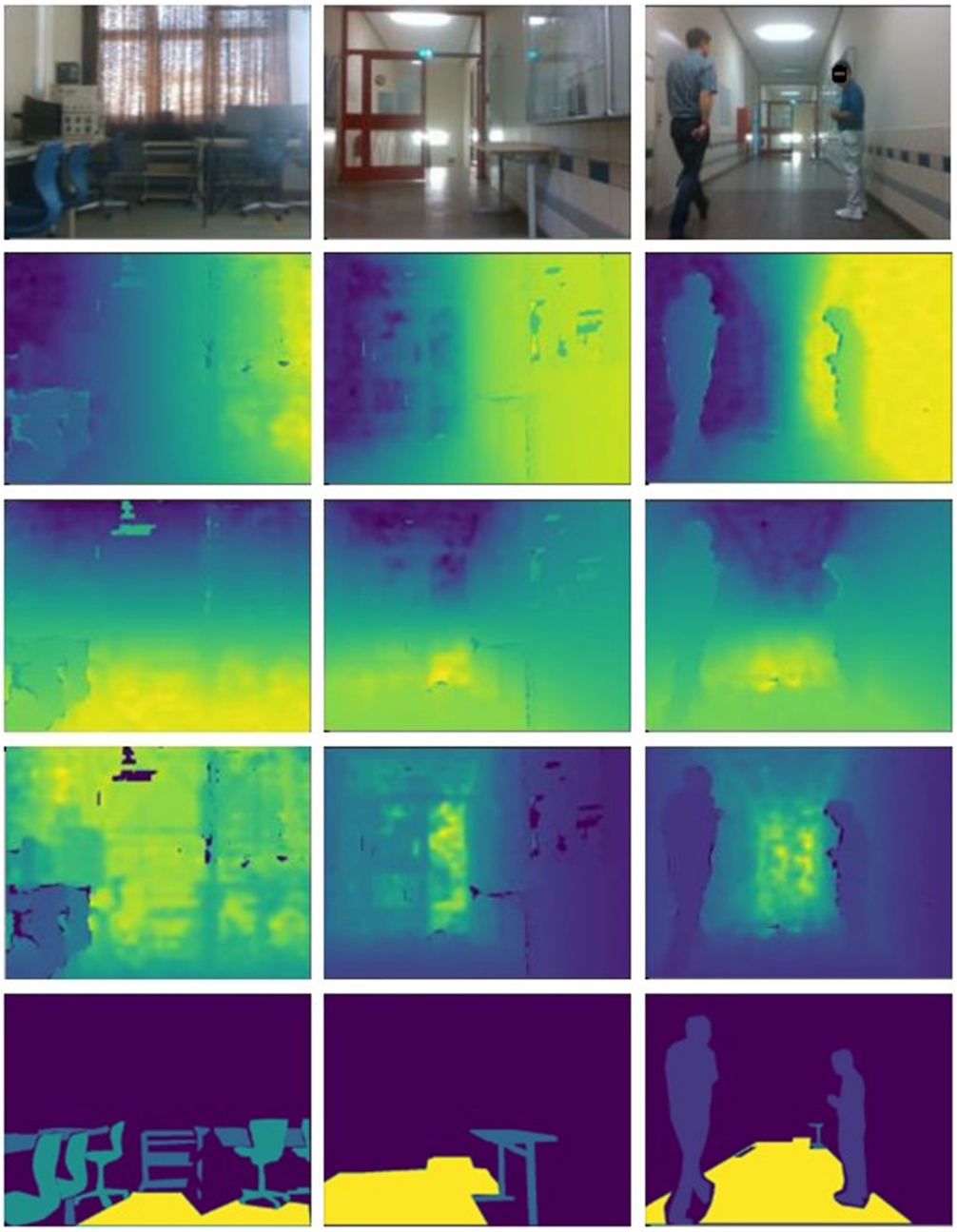

Fig. 5 Seven-classes FRA-UAS dataset (from top to bottom: RGB, *x, y*, and *z* coordinates, and labeled images).

Normally, when training for other domain tasks, only the weights of some of the last layers are trained, whereas the previous layer's weights are frozen. Early layers are trained very well on the large dataset to extract the best features from any images. In contrast, later layers are more specific to a particular task. (Edges features will be necessary for any classification task and extracted correctly in early layers. However, complex features, such as body parts, are important for human classification but not vehicle classifications, and those features need to be learned in later layers.)

In our scenario, we froze the encoder branches of the model and only trained on the decoder because the decoder is the part of the model responsible for inferring from the data extracted by the encoder. As the training progressed, once we saw no improvement in the validation accuracy, we unfroze the encoder branches and fully trained the model with a small learning rate (10^{-4}) to

Table 1 Comparison between without pretrained and pretrained model for mIoU, iterations until convergence, and required RAM.

	Without pretrained	Pretrained
mIoU score	0.67	0.70
Training iterations until convergence	10,000	8000
%RAM required when train with batch size = 28	100%	78%

squeeze some more accuracy gain from the model while also ensuring that the weights did not drift too much from the original numbers.

To verify the advantages of the transfer learning approach versus learning from scratch, we conducted an experiment between our model training on the seven-classes FRA-UAS dataset with two different sets of weights: completely random weights (train from scratch) and pretrained weights of the SceneNet dataset and three-classes FRA-UAS dataset. In our pretrained version, we only continue the training process of the decoder while freezing the encoder. As Table 1 shows, the model when trained from scratch achieved a lower accuracy, took longer to converge, and occupied more RAM compared with when trained by transfer learning.

5 Result and Discussion

This evaluation aims to solve the challenges R1 and R2 raised in Sec. 2.

R1: The robot distinguishes the surrounding obstacles well by implementing a machine learning model with lightweight size (with better accuracy and precision) that is easy to train and runs in real time.

In this section, first, we presented and justified the results of our model, which is a modified model with having small size parameters. We compared it with other deep learning RGB-D segmentation models. We tested the model on the renowned benchmark indoor dataset NYU-Depth V2,[22] which contains 1449 labeled images of indoor scenes with 41 classes. We compared the model size (number of parameters in millions), global accuracy precision (gAP), mean accuracy precision (mAP), and mIoU. The comparison results are shown in Table 2. In Table 2, pretrained denotes that the model has been trained in advance on the 1-million-image 1000 class dataset ImageNet before being trained on the NYUv2 dataset. HHA encoding is a geocentric embedding for each pixel in addition to the horizontal disparity depth images that encodes height above ground and angle with gravity for each pixel. A model's name with HHA denotes that the depth data were pre-preprocessed and converted to HHA encoding[23] before being fed to the model.

Therefore, according to the results, we can see that the proposed model with a lightweight size achieves a results for gAP, mAP, and mIoU of 65.1%, 46.1%, and 34.1%, respectively, which are acceptable compared with other deep learning methods. Additionally, the model is easy to train and deploy, which makes it run in real time with sufficient hardware strength, and it achieved acceptable results regarding accuracy and processing time within our customized FRA-UAS laboratory dataset.

R2: The robot operates under various indoor light conditions.

Furthermore, to achieve robust results for our application of indoor detection, we include the light condition in the test.

First, we trained our model with the SceneNet dataset, which achieved the global accuracy and mIoU of 0.73 and 0.39, respectively, with respect to per-class IoU, as shown in Table 3.

Some examples are shown in Fig. 6 after the SceneNet data have been processed by our model; the segmented objects are expressed with different colors.

Similarly, after retraining with the FRA-UAS dataset 1 (three classes), our model achieved the results for global accuracy and mIoU of 0.982 and 0.909, respectively, with respect to

Table 2 Performance comparison between our model and others.

Model name	Size (in millions)	gAP (%)	mAP (%)	mIoU (%)
DeepLab[24]	>21	50	23.9	15.9
HAA-CNN VGG16[24]	>42	59.1	30.8	21.9
D-CNN + HHA VGG16[24]	>42	61.4	35.6	26.2
D-CNN VGG16[24]	>21	60.3	39.3	27.8
FCN-32s[25]	>138	61.5	42.4	30.5
(Pretrained) Facebook AlexNet[26]	>63	62.9	41.3	30.8
(Pretrained) FuseNet[8,27]	>30	66.0	43.3	32.7
(Pretrained) FCH-32s + HHA[25]	>138	64.3	44.9	32.8
(Pretrained) FCN-16s + HHA[25]	>70	65.4	46.1	34.0
(Pretrained) Facebook VGG16[26]	>138	65.6	45.1	34.1
Our model	6.5	65.1	46.1	34.1
(Pretrained) DeepLab-L + HHA[28]	>37	68.4	49.0	37.6
(Pretrained) MMAF-Net-152[9]	122.3	72.2	59.2	44.8
(Pretrained) 3M2RNet[29]	225.4	76.0	63.0	48.0

Table 3 mIoU per class of the SceneNet dataset.

Global and mean metrics	Global accuracy	0.73
	mIoU	0.39
Per-class IoU	Background	0.92
	Bed	0.36
	Books	0.00
	Ceiling	0.69
	Chair	0.34
	Floor	0.66
	Furniture	0.22
	Objects	0.4
	Picture	0.4
	Sofa	0.05
	Table	0.33
	TV	0.25
	Wall	0.67
	Window	0.20

Fig. 6 Segmentation samples on the validation set of the SceneNet dataset (top row: RGB image, middle row: ground truth, and last row: predictions).

Table 4 mIoU per class of FRA-UAS dataset 1.

Global and mean metrics	Global accuracy	0.982
	mIoU	0.909
Per-class IoU	Obstacles	0.9801
	Free space	0.8503
	Human	0.8966

per-class IoU, as shown in Table 4. The examples of segmented objects with different colors are shown in Fig. 7.

Finally, the model is retrained with the FRA-UAS dataset 2 (seven classes) and achieved the results for global accuracy and mIoU of 0.956 and 0.72, respectively, with respect to per-class IoU. The examples of segmented object detection are shown in Table 5 and Fig. 8, respectively.

We can see (Fig. 8) that the performance of our model in different lighting conditions is very robust. For example, we can see from our results that the camera detected the objects (static and human) when the camera was exposed directly to the sunlight. The color of the human's face cannot be differentiated from the color curtains as well in the indoor light condition. However, by relying on the information provided by the depth camera, the human and the obstacles were detected correctly. We conclude that our model also detects the obstacles and humans in different indoor light conditions.

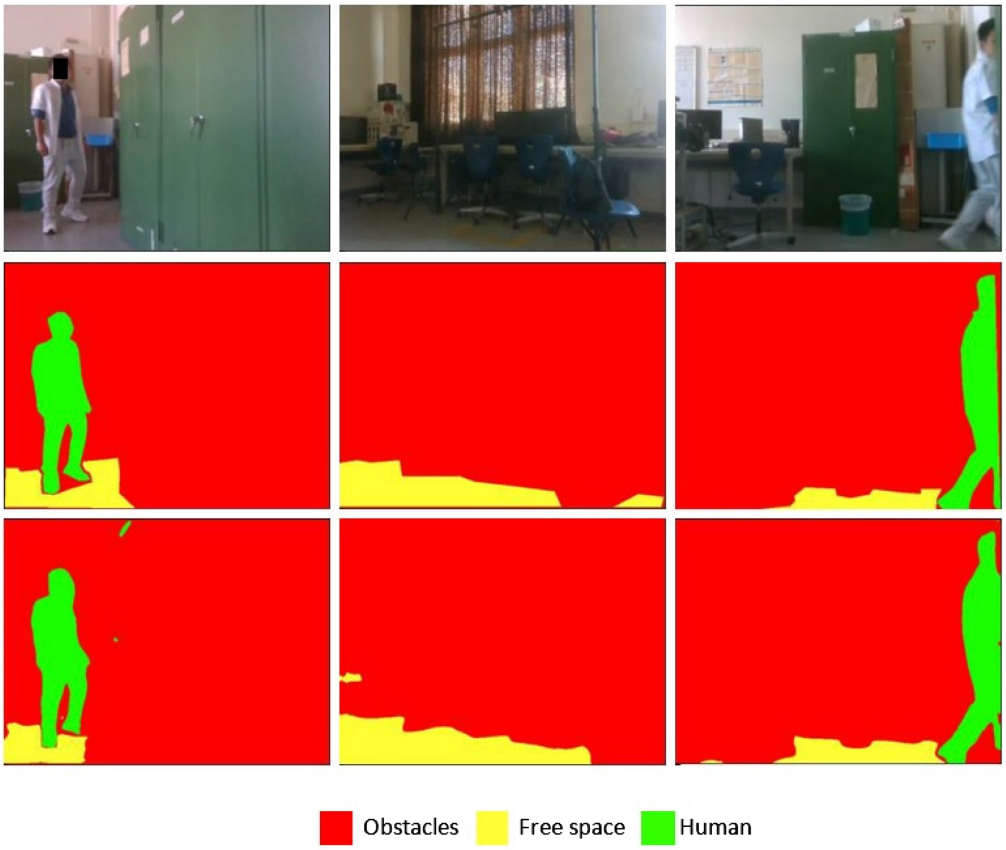

Fig. 7 Segmentation samples from the three-classes FRA-UAS dataset (top row: RGB image, middle row: ground truth, and last row: predictions).

Table 5 mIoU per class of FRA-UAS dataset 2.

	Global accuracy	0.9564
Global and mean metrics	mIoU	0.7203
Per-class IoU	Other obstacles	0.9506
	Human	0.8524
	Table	0.6628
	Chair	0.7632
	Robot	0.7085
	Backpack	0.2129
	Free space	0.8914

6 Conclusion and Future Work

In this work, we have solved the problem of object detection by applying our modified deep learning model during the operation of the mobile system in the FRA-UAS robotic laboratory. Compared with other deep learning RGB-D segmentation models, our model achieved comparative results when trained with the available NYUv2 dataset.

Furthermore, the modified model was trained by our customized indoor FRA-UAS dataset 1 with three classes and FRA-UAS dataset 2 with seven classes and achieved the global accuracy

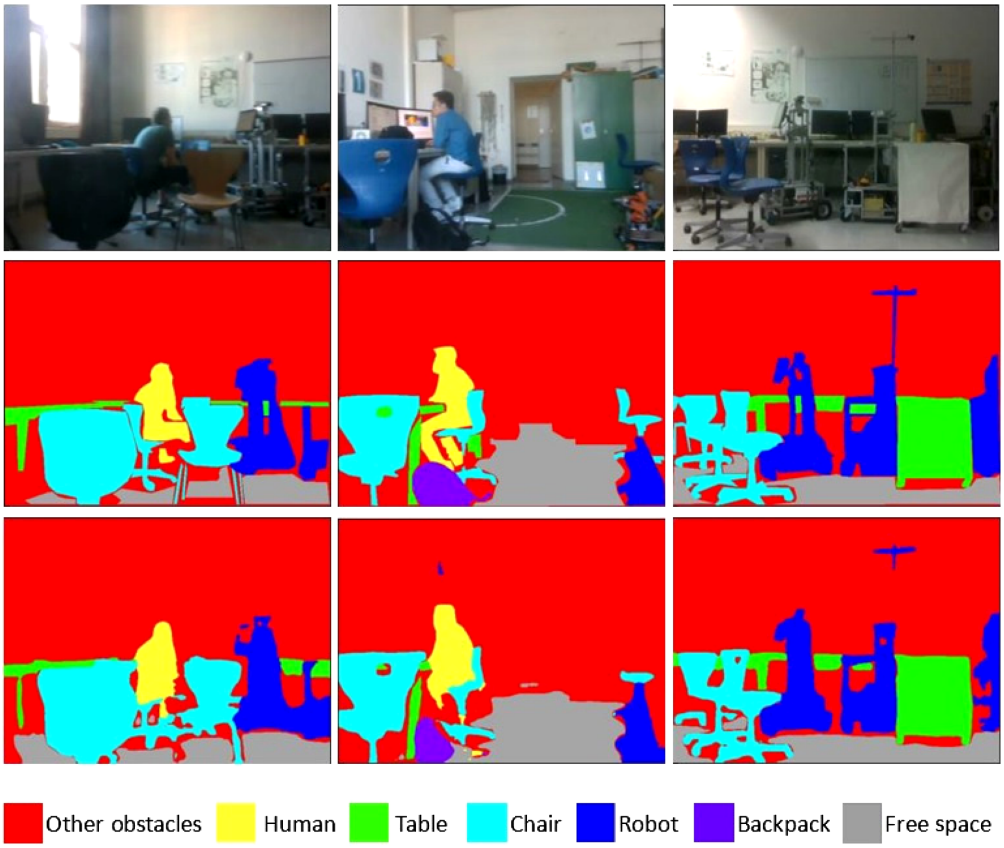

Fig. 8 Segmentation samples from the seven-classes FRA-UAS dataset (top row: RGB image, middle row: ground truth, and last row: predictions).

of ~98% and 95%, respectively. We have obtained better results than other traditional models for the application of mobile robot ROSWITHA in our indoor environment laboratory.

Our model segmentation output was used during the testing scenario to create the mobile robot's local cost map for its mobility. The results were optimal in the detection of difficult-to-detect objects, such as tables. We checked that our model was enough for the robots to navigate in the indoor environment without using other sensors for obstacle detection. Therefore, the performance of our model was robust in indoor light conditions.

Moreover, after achieving better results, we still found some limitations of our model. When the image is extremely blurred, the model does not detect the objects correctly. To overcome this limitation, we can reduce the confidence in the point cloud sent by the model during the fast turning of the robot or use another method to exploit the time component of the video stream. The model may rely more on the information of previous frames to infer the current blurred frames. To solve these issues, some researchers have tried to incorporate optical flow or a long short-term memory network into the CNN model to capture the relationship between adjacent frames.[21,30]

One of many possible directions for future development is to convert the original image segmentation problem into a video segmentation problem. The video segmentation problem will help the model become more robust to blur in the image and fast-moving scenes and objects as it can incorporate the time-series features of the data into the learning process. Also, learning images as a video will enhance the stability of the prediction because two adjacent frames tend not to differ very much, thus reducing the outliers in the predictions of the model.

Another possible direction to increase the robustness of the model would be enriching the real-world indoor dataset used to train and validate the model. Because our collected and labeled dataset is relatively small, collecting more data will make the model robust to different environments, and the model's performance evaluation result will be more trustworthy.

With segmenting, the robot captures images into classes to differentiate whether objects are static or stationary, and we can create the mobile robot's local cost map for its mobility. By applying navigation algorithms in later steps, the robot can move to the designed position and avoid collision with obstacles along its way. In future work, we will integrate the model with the navigation method of the mobile robot to understand the dynamic and static objects and apply it to the real-world robot's (ROSWITHA) application of maneuvering in a dynamic environment.

Acknowledgments

This work was funded by the European Commission (FEDER) and the Spanish Ministry of Science, Innovation and Universities under the project FAME (RTI2018-093608-B-C33).

References

1. Y. Xu et al., "Artificial intelligence: a powerful paradigm for scientific research," *The Innovation* **2**(4), 100179 (2021).
2. B. Brik et al., "ThermCont: a machine learning enabled thermal comfort control tool in a real time," in *15th Int. Wireless Commun. Mob. Comput. Conf. (IWCMC)*, pp. 294–300 (2019).
3. N. Cvijetic, "Ride in Nvidia's self-driving car [film]," Youtube (2019).
4. R. Barber et al., "Mobile robot navigation in indoor environments: geometric, topological, and semantic navigation," in *Applications of Mobile Robots*, E. G. Hurtado, Ed., ch. 5, IntechOpen, Rijeka (2019).
5. D. Teso-Fz-Betoño et al., "Semantic segmentation to develop an indoor navigation system for an autonomous mobile robot," *Mathematics* **8**, 855 (2020).
6. F. Fooladgar and S. Kasaei, "A survey on indoor RGB-D semantic segmentation: from hand-crafted features to deep convolutional neural networks," *Multimedia Tools Appl.* **79**, 4499–4524 (2020).
7. S. Minaee et al., "Image segmentation using deep learning: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(7), 3523–3542 (2022).
8. C. Hazirbas et al., "FuseNet: incorporating depth into semantic segmentation via fusion-based CNN architecture," in *Asian Conf. Comput. Vision (ACCV)* (2016).
9. F. Fooladgar and S. Kasaei, "Multi-modal attention-based fusion model for semantic segmentation of RGB-depth images," CoRR abs/1912.11691 (2019).
10. X. Hu et al., "ACNet: attention based network to exploit complementary features for RGBD semantic segmentation," in *IEEE Int. Conf. Image Process. (ICIP)*, pp. 1440–1444 (2019).
11. P. Chao et al., "HarDNet: a low memory traffic network," in *IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, pp. 3551–3560 (2019).
12. S. Jégou et al., "The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation," in *IEEE Conf. Comput. Vision and Pattern Recognit. Workshops (CVPRW)*, pp. 1175–1183 (2017).
13. K. He et al., "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 770–778 (2016).
14. S. Chaudhari et al., "An attentive survey of attention models," *ACM Trans. Intell. Syst. Technol.* **12**(5), 1–32 (2021).
15. K. Xu et al., "Show, attend and tell: neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn.*, Vol. 37, pp. 2048–2057 (2015).
16. M. Tan and Q. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, Vol. 97, pp. 6105–6114 (2019).
17. J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 7132–7141 (2018).
18. P. Nauth, "Improvement of assistive robot behavior by experience-based learning," in *6th Int. Conf. Hum. Syst. Interact. (HSI)*, pp. 363–367 (2013).
19. K. Michael, "Performance evaluation for full 3D projector calibration methods in spatial augmented reality," Master's Thesis, Temple University, Pennsylvania (2011).

20. S. I. Nikolenko, *Synthetic Data for Deep Learning*, Springer Optimization and Its Applications, Springer, Reading, Massachusetts (2021).
21. M. Ding et al., "Every frame counts: joint learning of video segmentation and optical flow," in *Proc. AAAI Conf. Artif. Intell.,* Vol. 34, No. 7, pp. 10713–10720 (2020).
22. N. Silberman et al., "Indoor segmentation and support inference from RGBD images," in *Comput. Vision – Europ. Conf. Comput. Vision (ECCV) 2012*, pp. 746–760 (2012).
23. S. Gupta et al., "Learning rich features from RGB-D images for object detection and segmentation," in *Eur. Conf. Comput. Vision (ECCV)*, pp. 345–360 (2014).
24. W. Wang and U. Neumann, "Depth-aware CNN for RGB-D segmentation," in *Comput. Vision – Eur. Conf. Comput. Vision (ECCV)*, pp. 144–161 (2018).
25. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 3431–3440 (2015).
26. D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *IEEE Int. Conf. Comput. Vision (ICCV)*, pp. 2650–2658 (2015).
27. C. Hazirbas et al., "Fusenet: incorporating depth into semantic segmentation via fusion-based CNN architecture," https://github.com/tum-vision/fusenet (2020).
28. B. Kang, Y. Lee, and T. Q. Nguyen, "Depth-adaptive deep neural network for semantic segmentation," *IEEE Trans. Multimedia* **20**(9), 2478–2490 (2018).
29. F. Fooladgar and S. Kasaei, "3M2RNet: multi-modal multi-resolution refinement network for semantic segmentation," in *Adv. Comput. Vision*, pp. 544–557 (2019).
30. A. Pfeuffer, K. Schulz, and K. Dietmayer, "Semantic segmentation of video sequences with convolutional LSTMs," in *IEEE Intell. Veh. Symp. (IV)*, pp. 441–1447 (2019).

Sudeep Sharan is currently working as a research staff in the research group of Autonomous Systems and Intelligent Sensors at Frankfurt University of Applied Sciences (FRA-UAS), Germany. He received his ME degree in engineering in 2014. His research work focuses on autonomous mobile robots with an emphasis on navigation in dynamic environments by means of deep learning, robot vision, and humanoid manipulators, including mechanics and human robot interaction.

Peter Nauth is a professor at Frankfurt University of Applied Sciences, Frankfurt, Germany, and director of the research group of Autonomous Systems and Intelligent Sensors. He received his PhD from Johannes Gutenberg University, Mainz, Germany, in 1985. His research interests include autonomous mobile systems, cognitive robots, assistive robots, artificial intelligence, computer vision, and smart sensors. He has authored 1 book, has 1 registered patent, and has published more than 50 research papers.

Juan-José Domínguez-Jiménez is an associate professor at the University of Cadiz (UCA), Spain. He received his PhD in computer science from UCA in 2009 and was honored with the extraordinary PhD award from UCA. His research interests include software verification, software testing, search-based software engineering, artificial intelligence, evolutionary algorithm, and cybersecurity. He has directed several PhD candidates in software testing. He has coordinated and participated in the development of several open-source testing tools.