Transfer Learning and Hyperparameter Optimization for Instance Segmentation with RGB-D Images in Reflective Elevator Environments

Lukas Reithmeier
Research Group
Advanced Information
Systems and Technology
Softwarepark 11
4232 Hagenberg, Austria
lukas.reithmeier@fh-hagenberg.at

Oliver Krauss
Research Group
Advanced Information
Systems and Technology
Softwarepark 11
4232 Hagenberg, Austria
oliver.krauss@fh-hagenberg.at

Gerald Adam Zwettler
Dept. of Software Eng.,
University of Applied
Sciences Upper Austria
Softwarepark 11
4232 Hagenberg, Austria
gerald.zwettler@fh-hagenberg.at

ABSTRACT

Elevators, a vital means for urban transportation, are generally lacking proper emergency call systems besides an emergency button. In the case of unconscious or otherwise incapacitated passengers this can lead to lethal situations. A camera-based surveillance system with AI-based alerts utilizing an elevator state machine can help passengers unable to initiate an emergency call. In this research work, the applicability of RGB-D images as input for instance segmentation in the highly reflective environment of an elevator cabin is evaluated. For object segmentation, a Region-based Convolution Neural Network (R-CNN) deep learning model is adapted to use depth input data besides RGB by applying transfer learning, hyperparameter optimization and re-training on a newly prepared elevator image dataset. Evaluations prove that with the chosen strategy, the accuracy of R-CNN instance segmentation is applicable on RGB-D data, thereby resolving lack of image quality in the noise affected and reflective elevator cabins. The mean average precision (mAP) of 0.753 is increased to 0.768 after the incorporation of additional depth data and with additional FuseNet-FPN backbone on RGB-D the mAP is further increased to 0.794. With the proposed instance segmentation model, reliable elevator surveillance becomes feasible as first prototypes and on-road tests proof.

Keywords
Instance Segmentation, RGB-D data, Transfer Learning, Reflective Environments

1 INTRODUCTION

Elevators are an essential part of modern urban life to enable accessibility in buildings with multiple stories. In Austria, public buildings with more than one story require an integrated elevator system, equipped with an emergency button to establish a call center voice connection with minimal effort [Nat06a, Mos18a]. These emergency buttons show a key drawback. When the person in need of help is no longer able to push the button, e.g. an elderly person has fallen and can't get up, or a person has fallen unconscious. Thus, a viable demand for camera-driven and preferably fully-automated emergency detection arises.

In the context of an industry-oriented project, research on the algorithmic and technological foundation for a reliable and automated emergency system is being conducted. With the camera systems mounted in corners of the elevator cabin, unorthodox views on the passengers arise. This point of view is rarely trained in current deep learning (DL) models for instance segmentation. Furthermore, the generally metallic and thus highly reflective nature of the elevator cabins presents a significant challenge for the task of analysing RGB video feed.

The research questions are:

- *Can depth information be used to improve the performance of instance segmentation in highly reflective and noisy elevator environments?*
- *Does incorporation of depth data generally increase the performance of instance segmentation?*
- *Can the use of a FuseNet-FPN as a backbone network together with transfer learning and hyperparameter optimization improve the performance of Mask R-CNN adapting to RGB-D?*

DOI:10.24132/CSRN.2021.3101.30

ISBN 978-80-86943-34-3

1.1 State of the Art

Instance segmentation, i.e. the detection of all pixels belonging to a specific class instance, is a highly challenging and relevant task in computer vision that has been a focus in research since the very first days of digital imaging. With the use of static camera systems, object segmentation is achievable from background modelling to calculate a difference image from the actual frame [Wu10a] with local intensity variance, motion and overlapping color spectrum hardening practical applicability. In the past, the focus was laid on semi-automated single-feature approaches such as flood-fill [Gon01a, Gom07a] incorporating the image intensities or Live-Wire [Bar97a] contour tracing based on the image edges. With Grab Cut a highly relevant tool for instance segmentation on RGB images was introduced, reducing the demand for user-interaction to zero in optimal scenarios [Rot04a, Tan15a].

Incorporating the expected object shape besides the image characteristics, as proposed for active shape [Gin02a] and active appearance models [Coo01a], allowed advances e.g. in the field of face detection and instance segmentation in general. Thus, knowing the statistical shape of the target structure, segmentation can be implicitly defined as registration problem [Xia16a, All18a]. With registration strategies, deep learning (DL) and frame-to-frame propagation, nowadays instance segmentation in RGB video sequences become feasible as object-tracking tasks [Hou19a].

To date, with the evolution of hardware and availability of machine learning frameworks, instance segmentation is generally considered as a machine learning topic. With more and more hidden layers introduced and the availability of training datasets and pre-trained models, convolutional neural networks became the gold standard in segmentation and classification. A broad range of published network types is available and applicable for various input data and application domains. With the progression of recurrent networks [Rum87a] allowing to construct a spatio-temporal memory of the processed input sequence, cf. long-short-term-memory (LSTM) [Hoc97a], significant advances in optical character recognition (OCR) and video processing in general have been possible [Sab18a].

For the task of image instance segmentation on visual input data, a broad range of different network types have been published in the past. With larger network structures increasing the number of hidden layers, application of simple layer-wise back-propagation leads to the so called vanishing gradient problem, i.e. update influence becomes marginal for the lower layers. This problem is counteracted by He et al. presenting the Residual Network (ResNet) [He16a] introducing bypassing of the updates and thus allowing an increased depth of 100+ layers. The common principle of filter pyramids for instance detection at varying scale is proposed by Lin et al. with the Feature Pyramid Networks (FPN) [Lin16a]. While FPN is practical for detection and classification tasks, pyramid up-scaling as introduced by Encoder/Decoder pattern of the U-net [Ron15a] finally features instance segmentation tasks. Besides using a filter pyramid for multi-resolution object localization, the Region Proposal Network (RPN) [Uij13a, Ren15a] detects objects invariant to translation and scale together with their bounding box and an objectness score as confidence metric. The RPN is thereby introduced by Ren et al. to speed up the R-CNN initially introduced by Graves et al. [Gra13a] then denoted as the Faster R-CNN [Ren15a].

The Fully Convolutional Instance-aware Semantic Segmentation (FCIS) introduced by Li et al. [Li16a] extends the Region-based Fully Convolutional Networks (R-FCN) introduced by Dai et al. [Dai16a] to perform instance segmentation besides object location. With the Mask R-CNN as evolution introduced by He et al. in 2017 [He17a], Faster R-CNN is adapted for instance segmentation, too. Instead of sliding windows, single-shot approaches utilize a scale bipyramid as introduced for TensorMask by Chen [Che19a]. Single-shot approaches are utilized for Single Shot MultiBox Detector (SSD) [Liu15a], RetinaNet [Lin17b] and YOLO [Red15a] respectively in a similar way, too.

Nevertheless, these approaches for instance detection and segmentation show high accuracy in case of good image quality only. With reflections and a bad SNR, even the sophisticated DL models for image processing often fail [Ahm11a]. Thus, specific pre-processing is necessitated, e.g. removing the shadows in traffic surveillance videos [Kil92a] or removing mirror-based reflections [Chi18a]. To overcome visual noise in image data and to improve accuracy in general, incorporation of depth data with RGB-D images is an adaequate strategy [Ye17a].

1.2 Related Work

With RGB-D data provided, Watershed transform [Beu79a] is applicable onto the varying local depth profiles leading to a first rough fragmentation of the scene. These pre-segmented regions can be utilized as input for semantic classifiers to detect humans and classify the body pose. As machine learning classifiers, Support Vectors [Cor95a], Random Forest [Ho95a] and XGBoost [Che16a] are applied.

Although most of the DL models only consider 3-channel RGB data as input, some CNNs incorporating depth data have already been published [Bo13a, Cou13a]. Thus, to incorporate arbitrary DL models for instance segmentation, transfer learning [Wei16a] or hyperparameter optimization [Los16a] are applicable to re-train the model with data augmentation or GAN

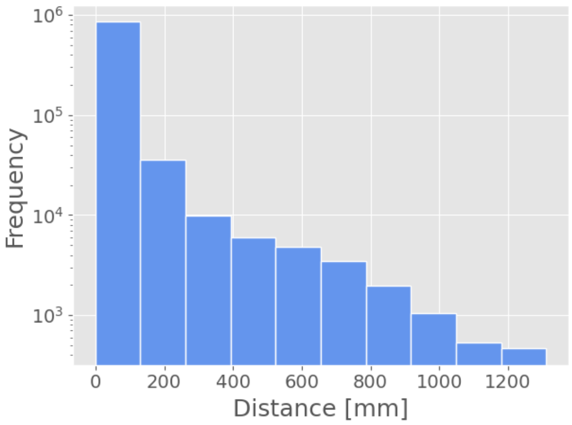


Figure 1: Pixel-wise standard deviation in the depth images of a recording of a closed elevator. The frequency is scaled logarithmically.

synthesis as a potential strategy in case of insufficient amount of training data in medicine [Zwe20a]. For the domain of elevator surveillance, lack of data generally is immanent due to the uncommon perspective.

Depth perception data has already been successfully applied to detect the doors of an elevator, including the difference between elevator and outside floor heights with precision $< 20mm$ and the opening state of the door as an accurate and generic alternative to mechanical sensor systems in the field of predictive maintenance. This is important for emergencies as this can mean that the elevator will not move to another floor if the door is blocked, or that first response help is being provided in case of a medical emergency [Ign99a].

An attempt has also been made to conduct classification purely with RGB data, omitting depth information utilizing different methods such as K-Nearest Neighbor, Support Vector Machine, Random Forest and Neural Networks. This has been shown to work at detection success of 92% in non-reflective environments. The noise introduced by reflections as well as movement of the elevator, i.e. vibrations also apply to the camera and greatly reduces the detection rate [Sti99a].

2 MATERIAL

To train the machine learning models we use two datasets. Since there are no RGB-D datasets containing samples recorded in highly reflective and noisy elevator scenes, we introduce the Elevator RGB-D dataset. In addition we use the SUN RGB-D dataset [Son15a]. The Elevator RGB-D dataset consists of 1138 samples, where each of these samples is composed of a three-channel RGB image, a one-channel depth image and labels. Each label consists of one or more 2D polygons with a corresponding class label.

The RGB and depth images are derived from 21 recorded RGB-D videos from scenes of humans and objects in four different elevators. The RGB-D videos are recorded using the Intel RealSense D-400 RGB-D camera [Kes17a]. One sample per second of recorded video is extracted. Samples without a label are removed from the dataset. The noise in the depth images varies between each elevator. The majority of the pixels in a recording have a very minor standard deviation, as seen in Figure 1. But some pixels in the recording have a standard deviation of over one meter. This recording contains RGB-D images of a closed elevator without changes in lighting conditions.

The depth images are aligned with the RGB image, so that the position of objects in the depth image corresponds to the position of objects in the RGB image. The depth image contains values from 0 to 65535, where the value of a pixel represents the distance from an object to the camera in millimeters.

The samples of the Elevator RGB-D dataset are labeled to contain objects of nine different classes { *human_standing*, *human_lying*, *human_sitting*, *human_other*, *object*, *box*, *bag*, *chair*, *plant*}, where *object* can be any object that is not a *box*, *bag*, *chair* or *plant*. Class *human_other* describes humans in any pose different to standing/lying/sitting such as e.g. crouching or indistinct/unknown poses. Furthermore, *human_standing* also includes walking humans, as long as they are in an upright pose. The number of objects per class are not distributed evenly, as seen in Figure 2. The object categories such as bag or box are utilized to detect potential hazardous objects left in the elevator cabin, too. Especially at airports static objects need to be treated as a threat and thus triggering an emergency. The label with the most occurrences *human_standing* is thrice as common as the label with the second most occurrences. Samples of this dataset can be seen in Figure 3.

The Elevator RGB-D dataset is inherently biased, since the RGB-D recordings only contained images of 10 different male persons with light skin color wearing various different outfits, also including beards or glasses. This bias can prevent generalization of image recognition models trained using this dataset [Tor11a, Tom15a].

The samples have different sizes, therefore all RGB and depth images are resized and zero-padded to a size of 512×512. The depth images are normalized to a range of 0 to 255 using min-max normalization as described by Patro and Sahu [Pat15a]. Furthermore, to allow processing the Mask R-CNN, each of the polygons of the n labels of a sample is converted to a binary mask per label. The n masks of a sample are combined to an n-channel image. Additionally, a n-dimensional vector with an integer representation c of the class label is derived from the labels with zero encoding background.

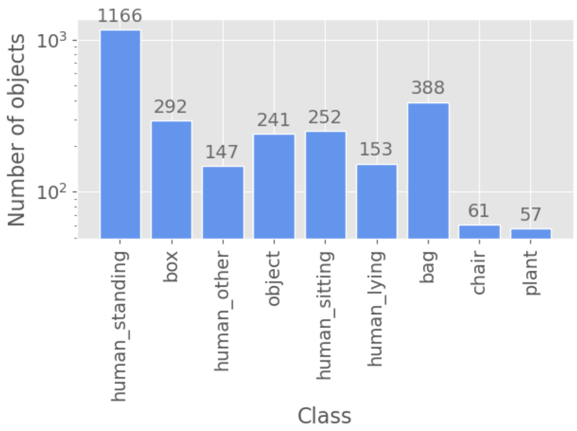


Figure 2: Number of objects per class in the Elevator RGB-D dataset, scaled logarithmically with *human_lying* and *human_sitting* theerby replicating medical emergency situations.

Both the Elevator RGB-D dataset and the SUN RGB-D dataset are randomly split into three different categories for training, validation and testing. The SUN RGB-D dataset has 5285 training samples, 475 validation samples and 4575 test samples. The Elevator RGB-D dataset has 797 training samples, 228 validation samples and 113 test samples, all randomly assigned.

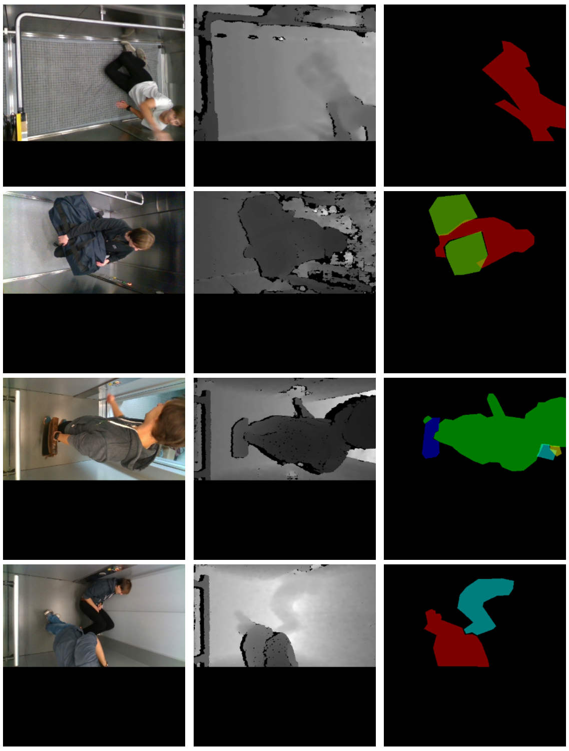


Figure 3: Samples of the Elevator RGB-D dataset. Each sample consists of an RGB image (left), a depth image (middle) and labels (right).

3 METHODS

To answer the question, whether depth data can be used to improve the performance of instance segmentation in highly reflective and noisy elevator environments, three different approaches of the Mask R-CNN model [He17a] are compared. The approach RGB uses solely the three-channel RGB images as input of the Mask R-CNN, the approach D3 uses the depth image duplicated over three channels to achieve tensor conformity at the cost of redundancy, the approach RGBD uses a four-channel image consisting of the three channels of the RGB image and one corresponding depth channel. These three approaches use a ResNet-FPN [He16a] as backbone network. In addition to these three approaches the approach RGBD-F also uses a four-channel RGB-D image as input, but also uses an adapted FuseNet-FPN as a backbone network to answer the question whether this usage can improve the performance of Mask R-CNN with RGB-D images in highly reflective noisy elevator environments.

The adapted FuseNet-FPN consists of the encoder part of the FuseNet introduced by Hazirbas et al. [Haz16a] and thus consists of five different stages. The filter kernels have a size of 7×7 in the first stage and 3×3 in the following stages. The ResNet and the FuseNet are used as a feature pyramid networks (FPN) [Lin17a].

Usually transfer learning using weights of a ResNet-101 or a ResNet-50 trained with the MS COCO dataset [Lin14a] is applied to initialize the Mask R-CNN model [He17a]. But since four different approaches with RGB and depth information are used and the MS COCO dataset only contains RGB information, initializing the weights would undermine the comparability of these approaches. Therefore, the training procedure consists of the following three steps:

1. A hyperparameter optimization is performed using the SUN RGB-D dataset.

2. The models are pre-trained using the optimized hyperparameters and the SUN RGB-D dataset.

3. Transfer learning performed by transferring the weights learned training the models with the SUN RGB-D dataset and further training the models with the Elevator RGB-D dataset.

3.1 Hyperparameter Optimization

The approaches differ in input data and backbone architecture. Thus, their hyperparameters are optimized separately. The hyperparameter space consists of the backbone architecture, the number of regions of interest per image, the minimum detection confidence and choice of the optimizer.

The backbone architecture determines the kind of CNN that is used to extract features from the input image.

The approaches RGB, D3 and RGBD can either use a ResNet101 with a mini-batch size of 1, a ResNet50 with a mini-batch size of 1 or a ResNet50 with a mini-batch size of 2. The backbone architecture of the approach RGBD-F is a FuseNet that is separated into five different stages. The number of filters in each Conv layer for the five stages $(f_1, f_2, f_3, f_4, f_5)$ can be one of the three different configurations:

$$(f_1, f_2, f_3, f_4, f_5) \in \begin{array}{c} \{(32,32,64,128,256), \\ (64,64,128,256,512), \\ (128,128,256,512,1024)\}. \end{array} \quad (1)$$

The minimum detection confidence determines whether regions of interest are considered to be a valid result. Regions of interest with a class score below the minimum detection confidence are discarded. The empirically chosen minimum detection confidence ranges from 0.6 to 0.8. The maximum number of regions of interest per image can either be 50, 100 or 200. As optimizer either a SGD [Kie52a] with mini-batches and momentum or an ADAM optimizer [Kin14a] can be used. In total, the hyperparameter space consists of 54 different configurations.

Tree-structured Parzen Estimators (TPE) [Ber11a] is used to minimize the total number of evaluations needed for a use-able result. TPE is a sequential model-based optimization algorithm (SMBO) commonly used to optimize hyper parameters of machine learning algorithms. TPE searches a minimum in a surrogate model of the hyperparameter space, which it evaluates by training a machine learning model with said minimum as hyperparameter configuration. The resulting score value is used to adapt the surrogate model. The number of evaluations per approach is chosen to be 10. To evaluate a hyperparameter configuration a model is trained for 100 epochs, however an epoch during hyperparameter optimization is set to use only 500 samples. Afterwards the sample-wise F1 scores are calculated using all the validation samples. The mean of these sample-wise F1 scores is used as metric that the TPE algorithm has to maximize. F1 scores are calculated using the bounding boxes with an intersection over union (IoU) threshold of 0.5.

3.2 Pre-training

In order to perform transfer learning, the models are pre-trained using the SUN RGB-D dataset and the optimized hyperparameters. One model per approach is trained for 50 epochs. One epoch during training using the SUN RGB-D dataset consists of 2643 mini-batches. After 25 epochs learning rate decay reduces the learning rate from 0.005 to 0.001. The Region Proposal Network of the Mask R-CNN uses 15 anchor boxes that are defined with scales $s \in \{16, 32, 64, 128, 256\}$ and aspect ratios $ar \in \{0.5, 1, 2\}$. The approach RGBD-F uses a dropout rate of 0.3.

Stochastic Gradient Descent (SGD) [Rob51a] and Adaptive Moment Estimation (ADAM) [Kin14a] are optimizers commonly used to train convolutional neural networks. We used ADAM and SGD as choices in the hyperparameter optimization. Although ADAM optimizer produced the best results during the hyperparameter optimization, SGD with mini-batch and momentum is used to train the models, since training with the ADAM optimizer performs worse on the validation samples. The ADAM optimizer initially has a lower validation loss than the SGD optimizer. But the validation loss of the ADAM optimizer stagnates over the course of the training, whereas the validation loss of the SGD optimizer drops, while both training losses converge towards a minimum, indicating overfitting.

3.3 Transfer Learning and Training

The Elevator RGB-D dataset is only a tenth in size compared to the SUN RGB-D dataset. Therefore, transfer learning is applied, so that the pre-trained weights can be reused and the models trained on the Elevator RGB-D dataset converge towards a minimum faster and produce better results.

The model weights learned during the pre-training are reused to initialize the models for the training. For each approach a model is trained for another 50 epochs using the SGD optimizer. An epoch during training using the Elevator RGB-D dataset consists of 399 mini-batches. Learning rate decay reduces the learning rate from 0.005 to 0.0001 after 25 epochs.

3.4 Evaluation

The four approaches are evaluated after the training by calculating the mean average precision, the mean recall and the mean F1 score. The mean average precision (mAP) is calculated as described by Padilla et al. [Pad20a]. Similar to the mAP, the mean average recall (mAR) is calculated by averaging the average of the recall for each class for each sample. The mean F1 score is calculated by averaging the sample-wise F1 scores. The sample-wise F1 score is calculated using the average precision and the average recall of a sample. Additionally a precision-recall curve is calculated for each approach as described by Padilla et al. [Pad20a]. To derive these metrics the bounding boxes generated by the Mask R-CNN, the ground-truth bounding boxes, an IoU threshold of 0.5 and the test samples of the Elevator RGB-D dataset are used.

3.5 Implementation Details

To annotate the Elevator RGB-D dataset with labels the Label Studio software by Tkachenko et al. [Tka20a] is used. OpenCV [Bradski00a] is used to implement

	RoIs	DMC	Backbone
RGB	100	0.6	ResNet50 - BS2
D3	200	0.7	ResNet50 - BS2
RGBD	50	0.8	ResNet50 - BS2
RGBD-F	50	0.8	[32,32,64,128,256]

Table 1: Optimal configurations for each of the four approaches all utilizing ADAM optimizer.

Input Resolution	PTE	mAP	mAR	F1 score
256 × 256	50	0.031	0.272	0.129
512 × 512	50	0.131	0.219	0.268

Table 2: A comparison of models of the approach RGB using input images of different resolution.

Approach	mAP	mAR	F1 score
RGB	0.131	0.219	0.268
D3	0.155	0.241	0.312
RGBD	0.126	**0.271**	0.320
RGBD-F	**0.156**	0.258	**0.331**

Table 3: Mean average precision (mAP), their mean average recall (mAR) and F1 score for the approaches after pre-training.

Pre-trained on	none	SUN RGB-D	MS COCO
Epochs	0	50	320
ResNet layers	50	50	101
mAP	0.661	0.753	**0.862**
mAR	0.569	0.560	**0.585**
F1 score	0.535	0.579	**0.636**

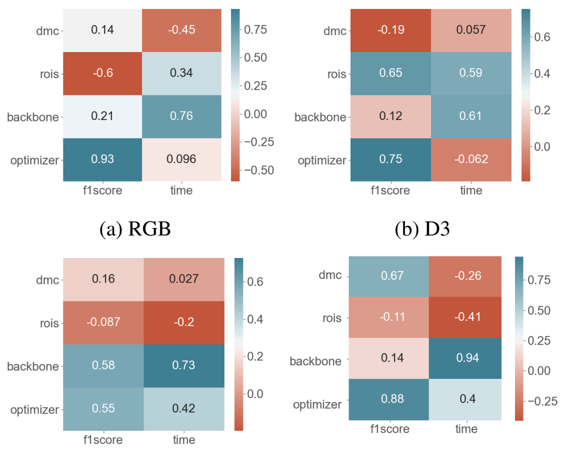
Table 4: A comparison of models of the approach RGB using transfer learning and without transfer learning.

the preprocessing steps. The Matterport Mask R-CNN implementation by Abdulla [Abd17a] is used to train and evaluate a Mask R-CNN model. The Matterport Mask R-CNN is altered to be compatible with TensorFlow version 2.0 made by Abadi et al. [Aba15a]. To perform the hyperparameter optimization using Tree-structured Parzen Estimators, *hyperopt* by Bergstra et al. [Ber13a] is used. Models are trained on a single *NVIDIA GeForce RTX 2070* with 8GB of memory.

4 RESULTS

Table 1 shows the optimal hyperparameters for each approach. In this table each configuration consists of the maximum number of regions per image (RoIs), the minimum detection confidence (DMC), the backbone architecture and the optimizer. The correlations between the hyperparameters, as well as the F1 score and the runtime of each evaluation run (time) for each approach can be seen in Figure 4. These correlations are calculated using Pearson's R [Pear96a].

Since the mini-batch size is limited by the GPU memory, only a mini-batch size of 2 is possible with 512 × 512 sized input images. As seen in Table 2 reducing the size of the images to 256 × 256 to increase the size of the mini-batches to 8 worsens the results, at least for the approach RGB. Table 2 shows models of the approach RGB using input images of different resolution, with the number of epochs that they were pre-trained (PTE), their mean average precision (mAP), their mean average recall (mAR) and their F1 score after 50 epochs of training using the SUN RGB-D dataset. The scores mAP, mAR and F1 are calculated using the test data of the SUN RGB-D dataset.



(a) RGB

(b) D3

(c) RGBD

(d) RGBD-F

Figure 4: The correlations between the hyperparameters for the four approaches.

The approaches that use RGB-D images have better F1 scores than the approaches that solely rely on the RGB image and the depth image. The depth images of the SUN RGB-D dataset do not contain as many reflections and noise as the depth images of the Elevator RGB-D dataset. Therefore, one cannot conclude that the approaches RGBD and RGBD-F perform similarly well on the Elevator RGB-D dataset, see Table 3.

Using transfer learning to initialize the models with weights pre-trained with the SUN RGB-D dataset leads to better results, as seen in Table 4. This table shows models with and without pre-trained weights the dataset that they were pre-trained on, the number of epochs that they were pre-trained, their mean average precision (mAP), their mean average recall (mAR) and their F1 score after 50 epochs of training using the Elevator RGB-D dataset. mAP, mAR and F1 score are calculated using the test data of the Elevator RGB-D dataset. As seen in this table using transfer learning with weights pre-trained on the MS COCO dataset, provided by Abdulla [Abd17a], yields the best results after training for additional 50 epochs, although, this validation is only done for the approach RGB.

The RGB-D images utilizing a FuseNet-FPN backbone network perform best on the Elevator RGB-D dataset with a mean average precision of 0.794, a mean average recall of 0.565 and an mean F1 score of 0.599, see Table 5. Whereas relying on solely the RGB images or

Approach	mAP	mAR	F1 Score
RGB	0.753	0.560	0.579
D3	0.707	0.558	0.569
RGBD	0.768	0.558	0.578
RGBD-F	**0.794**	**0.565**	**0.599**

Table 5: Approaches evaluated w.r.t. their mAP, mAR and their mean F1 score.

the RGB images in combination with the depth image but with a ResNet50-FPN backbone network leads to similar results. Both are worse than the results of the approach RGBD-F. The approach that solely uses the depth images achieves the worst results. The precision-recall curves, as seen in Figure 5, supports this observation.

Thus, one can conclude that using RGB and RGB-D data for instance segmentation with Mask R-CNN in an elevator environment leads to equal results, whereas relying solely on depth images achieves worse results. Therefore, depth information does not improve the performance of instance segmentation with Mask R-CNN in highly reflective and noisy elevator environments. Furthermore, the usage of a FuseNet-FPN as a backbone network improves the resulting model in a highly reflective elevator environment. Figure 6 shows images segmented with these models.

5 CONCLUSION AND OUTLOOK

In this paper a comparison is done on whether instance segmentation using the Mask R-CNN algorithm can be improved by the usage of depth images. The images are recorded in highly noisy and reflective elevator environments. To perform this comparison four different approaches are compared. The four approaches use:

1. RGB image with a ResNet-FPN backbone
2. depth image duplicated on three channels with a ResNet-FPN backbone network
3. RGB-D image with a ResNet-FPN backbone
4. RGB-D image with an adapted FuseNet-FPN backbone network.

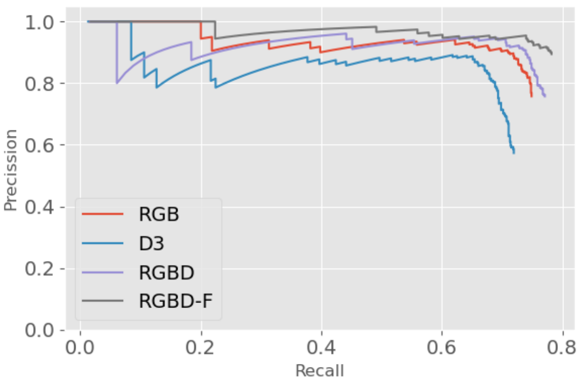


Figure 5: Precision recall curves of the four approaches.

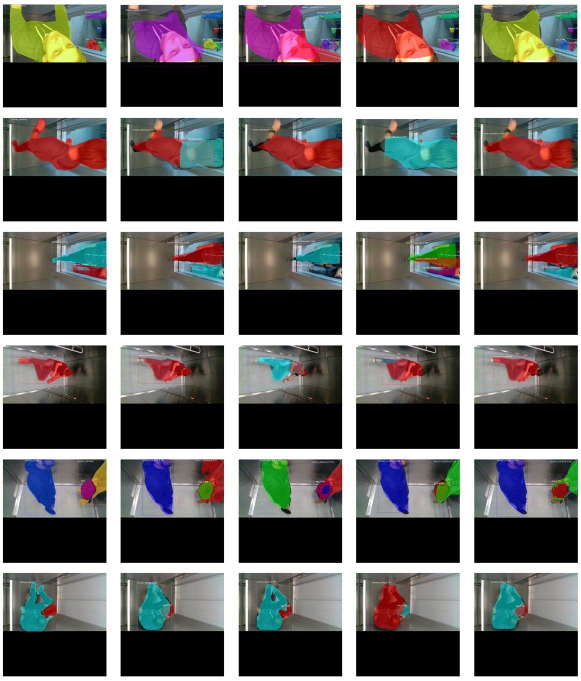


Figure 6: Samples of the Elevator RGB-D dataset instance-wise segmented with the pre-trained models. From left to right: ground truth, RGB, D3, RGBD, RGBD-F. The colors correspond to instances.

For each of these four approaches first the hyperparameters are optimized, pre-training is performed using the SUN RGB-D dataset, which contains images from indoor scenes and using transfer learning the pre-trained weights are further trained using images from an elevator environment. This comparison is done to answer the following research questions:

- *Can depth information be used to improve the performance of instance segmentation in highly reflective elevator environments which generally show a bad signal-to-noise ratio (SNR)?*

  The approach using an RGB image produces comparable results to the approach using RGB-D images, when training the Mask R-CNN models using a ResNet50-FPN as backbone network with a difference in the F1 score of 0.001. Solely relying on the depth image produces worse results than these two approaches.

- *Does incorporation of depth data generally increase the performance of instance segmentation?*

  After the pre-training using images from indoor scenes, the approach using the RGB image leads to the worst F1 score of all four approaches. The approach using the depth image in addition to the RGB image results in a better F1 score. The approach using solely the depth image leads to a F1 score between these two approaches.

- *Can the use of a FuseNet-FPN as a backbone network together with transfer learning and hyperparameter optimization improve the performance of Mask R-CNN adapting to RGB-D images?*

  The approach using a FuseNet-FPN as a backbone network and RGB-D images results in a better F1 score than the approach using a ResNet50-FPN and RGB-D images using the images from an elevator environment as well as images from an indoor scene.

5.1 Outlook

Hyperparameter optimization using an SMBO algorithm can lead to configurations that tend to increase overfitting, since only one score metric is optimized. To improve the hyperparameter optimization, the evaluation runs during hyperparameter optimization can be performed for as long as the pre-training. This negates the necessity of the pre-training, since each iteration of the SMBO algorithm produces the pre-trained weights. Although this will also increase the time needed for the hyperparameter optimization by the increase in the total number of learned training data.

This problem that occurred during the hyperparameter optimization lead to the switch from the ADAM optimizer to the SGD optimizer, since the SGD optimizer is less likely to converge towards a local minimum [Wu16a, Kes17a, Aki17a]. Alternatively, this problem can be avoided by switching from ADAM to SGD during training [Wu16a, Kes17a] or using an optimizer with this switching behavior [Luo19a].

Since the mini-batch size is limited by the GPU memory only a mini-batch size of 2 is possible with 512×512 sized input images. The training process thus can be further improved by using multiple GPUs in parallel. He et al. used 8 GPUs in parallel to train the Mask R-CNN on the MS COCO dataset [He17a]. Loshchilov et al. use 30 GPUs in parallel to perform hyperparameter optimization of CNNs [Los16a].

The models are each only trained for 50 epochs to provide a fair comparison between them. They can be improved by increasing the number of epochs and use early-stopping if overfitting occurs. Additionally, pretraining can also be extended, which as shown in table 4 can improve the performance of the resulting model.

6 ACKNOWLEDGMENTS

Our thanks to the Austrian Research Promotion Agency FFG for facilitating project EDEN with basic funding program number 875284, fed by research budget of the Federal Republic of Austria.

7 REFERENCES

[Aba15a] Abadi, Martin et al. TensorFlow: Large-scale machine learning on heterogeneous systems, Software available from tensorflow.org, 2015.

[Abd17a] Abdulla, Waleed. Mask R-CNN for object detection and instance segmentation on Keras and Tensorflow. https://github.com/matterport/Mask_RCNN.

[Ahm11a] Ahmed, Mohamed, and Pitie, F.. Reflection detection in image sequences. In Proc. of the IEEE Comp. So. Conf. on Computer Vision and Pattern Recognition, pages 705-712, 2011.

[Aki17a] Akiba, Takuya, Suzuki, Shuji, and Fukuda, Keisuke. Extremely large minibatch SGD: training Resnet-50 on Imagenet in 15 minutes. CoRR, abs/1711.04325, 2017.

[All18a] Alldieck, Thiemo, Magnor, Marcus A., Xu, Weipeng, Theobalt, Christian, and Pons-Moll, Gerard. Video based reconstruction of 3d people models. CoRR, abs/1803.04758, 2018.

[Bar97a] Barrett, William A., and Mortensen, Eric N. Interactive livewire boundary extraction. Medical Image Analysis, 1(4):331-341, 1997.

[Ber11a] Bergstra, James, Bardenet, Remi, Bengio, Yoshua, and Kegl, Balazs. Algorithms for hyper-parameter optimization. In Proc. of the 24th Int. Conf. on Neural Information Processing Systems, NIPS'11, page 2546-2554, 2011.

[Ber13a] Bergstra, James, Yamins, D., and Cox, D.D.. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Proc. of the 30th International Conf. on Machine Learning - Volume 28, ICML'13. JMLR.org, 2013.

[Beu79a] Beucher, Serge, and Lantuejoul, Christian. Use of watersheds in contour detection. In Int. Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, 132, 1979.

[Bo13a] Bo, Liefeng, Ren, Xiaofeng, and Fox, Dieter. Unsupervised Feature Learning for RGB-D Based Object Recognition, pages 387-402. Springer International Publishing, Heidelberg, 2013.

[Bradski00a] Bradski, G.. The opencv library. Dr. Dobb's Journal of Software Tools, 2000.

[Che16a] Chen, Tianqi, and Guestrin, Carlos. Xgboost: A scalable tree boosting system. In Proc. of the 22nd ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining, pages 785-794, 2016.

[Che19a] Chen, Xinlei, Girshick, Ross B., He, Kaiming, and Dollar, Piotr. Tensormask: A foundation for dense object segmentation. CoRR, abs/1903.12174, 2019.

[Chi18a] Chi, Zhixiang, Wu, Xiaolin, Shu, Xiao, and Gu, Jinjin. Single image reflection removal using deep encoder-decoder network. CoRR, abs/1802.00094, 2018.

[Coo01a] Cootes, T.F., Edwards, G.J., and Taylor, Christopher. Active appearance models. IEEE Trans. on Pattern Analysis and Machine Intelligence, 23:681-685, 2001.

[Cor95a] Cortes, Corinna, and Vapnik, Vladimir. Support-vector networks. Mach. Learn., 20(3):273-297, 1995.

[Cou13a] Couprie, Camille, Farabet, Clement, Najman, Laurent, and LeCun, Yann. Indoor semantic segmentation using depth information, 2013.

[Dai16a] Dai, Jifeng, Li, Yi, He, Kaiming, and Sun, Jian. R-FCN: Object detection via region-based fully convolutional networks, in CoRR, 2016.

[Gin02a] Ginneken, B. van, Frangi, A. F., Staal, J. J., ter Haar Romeny, B. M., and Viergever, M. A.. Active shape model segmentation with optimal features. IEEE Trans. on Medical Imaging, 21(8):924-933, 2002.

[Gom07a] Gomez, Octavio, Gonzalez, Jesus A., and Morales, Eduardo F.. Image segmentation using automatic seeded region growing and instance-based learning. In: Progress in Pattern Recognition, Image Analysis and Applications, pages 192-201, Springer, Berlin Heidelberg, 2007

[Gon01a] Gonzalez, Rafael C., and Woods, Richard E.. Digital Image Processing. Addison-Wesley Longman Publishing, USA, 2nd ed., 2001.

[Gra13a] Graves, Alex. Generating sequences with recurrent neural networks. CoRR, abs/1308.0850, 2013.

[Haz16a] Hazirbas, Caner, Ma, Lingni, Domokos, Csaba, and Cremers, Daniel. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In ACCV, 2016.

[He16a] He, K., Zhang, X., Ren, S., and Sun, J.. Deep residual learning for image recognition. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 770-778, 2016.

[He17a] He, Kaiming, Gkioxari, Georgia, Dollar, P., and Girshick, Ross B.. Mask r-cnn. IEEE International Conf. on Computer Vision (ICCV), pages 2980-2988, 2017.

[Ho95a] Ho, Tin Kam. Random decision forests. In Proc. of the 3rd Int. Conf. on Document Analysis and Recognition, volume 1 of ICDAR '95, page 278, USA, IEEE Computer Society, 1995.

[Hoc97a] Hochreiter, Sepp, and Schmidhuber, Jurgen. Long short-term memory. Neural Comput., 9(8):1735-1780, Nov. 1997.

[Hou19a] Hou, Rui, Chen, Chen, Sukthankar, Rahul, and Shah, Mubarak. An efficient 3d CNN for action/object segmentation in video. CoRR, abs/1907.08895, 2019.

[Ign99a] ignace20192019 Jordens. State classification of elevator doors to assist emergency detection in elevator networks, 2019.

[Kes17a] Keselman, L., Woodfill, J. I., Grunnet-Jepsen, A., and Bhowmik, A.. Intel(r) realsense(tm) stereoscopic depth cameras. In IEEE Conf. on Computer Vision and Pattern Recognition Workshops, pages 1267-1276, 2017.

[Kes17a] Keskar, Nitish Shirish, and Socher, Richard. Improving generalization performance by switching from adam to SGD. CoRR, abs/1712.07628, 2017.

[Kie52a] Kiefer, J., and Wolfowitz, J.. Stochastic estimation of the maximum of a regression function. Ann. Math. Statist., 23(3):462-466, 09 1952.

[Kil92a] Kilger, M.. A shadow handler in a video-based real-time traffic monitoring system. In Proc. IEEE Workshop on Applications of Computer Vision, pages 11-18, 1992.

[Kin14a] Kingma, Diederik, and Ba, Jimmy. Adam: A method for stochastic optimization. International Conf. on Learning Representations, 12 2014.

[Li16a] Li, Yi, Qi, Haozhi, Dai, Jifeng, Ji, Xiangyang, and Wei, Yichen. Fully convolutional instance-aware semantic segmentation, 2016.

[Lin14a] Lin, Tsung-Yi, Dollar, Piotr, Girshick, Ross B., He, Kaiming, Hariharan, Bharath, and Belongie, Serge J.. Feature pyramid networks for object detection. CoRR, abs/1612.03144, 2016.

[Lin16a] Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S.. Feature pyramid networks for object detection. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 936-944, 2017.

[Lin17a] Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross B., He, Kaiming, and Dollar, Piotr. Focal loss for dense object detection. CoRR, abs/1708.02002, 2017.

[Lin17b] Lin, Tsung-Yi, Maire, Michael, Belongie, Serge J., Bourdev, Lubomir D., Girshick, Ross B., Hays, James, Perona, Pietro, Ramanan, Deva, Dollar, Piotr, and Zitnick, C. Lawrence. Microsoft COCO: common objects in context. CoRR, abs/1405.0312, 2014.

[Liu15a] Liu, Wei, Anguelov, Dragomir, Erhan, Dumitru, Szegedy, Christian, Reed, Scott E., Fu, Cheng-Yang, and Berg, Alexander C.. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015.

[Los16a] Loshchilov, Ilya, and Hutter, Frank. Cma-es for hyperparameter optimization of deep neural networks. In Conference Track Proceedings, 4th Int. Conf. on Learning Representations, 2016.

[Luo19a] Luo, Liangchen, Xiong, Yuanhao, Liu, Yan, and Sun, Xu. Adaptive gradient methods with dynamic bound of learning rate. In Proc. of the 7th International Conf. on Learning Representations, New Orleans, Louisiana, 2019.

[Mos18a] Moser, Herbert, and Rebel, Johann. Leitfaden fuer personenaufzuege und personenhebeeinrichtungen. Technical report, MA 34 Bau -und Gebaeudemanagement, Stadt Wien, 2018.

[Nat06a] Nationalrat Oesterreich. Bundesgesetz ueber die Gleichstellung von Menschen mit Behinderungen (Bundesbehindertengleichstellungsgesetz - bgstg), 2006.

[Pad20a] Padilla, R., Netto, S. L., and da Silva, E. A. B.. A survey on performance metrics for object-detection algorithms. In International Conf. on Systems, Signals and Image Processing (IWSSIP), pages 237-242, 2020.

[Pat15a] Patro, S. Gopal, and Sahu, Kishore Kumar. Normalization: A preprocessing stage. IARJSET, 03 2015.

[Pear96a] Pearson, Karl. Mathematical contributions to the theory of evolution. Philos Trans R Soc Lond Series A, 187:253-318, 1896.

[Red15a] Redmon, Joseph, Divvala, Santosh Kumar, Girshick, Ross B., and Farhadi, Ali. You only look once: Unified, real-time object detection. CoRR, abs/1506.02640, 2015.

[Ren15a] Ren, Shaoqing, He, Kaiming, Girshick, Ross B., and Sun, Jian. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR, abs/1506.01497, 2015.

[Rob51a] Robbins, Herbert, and Monro, Sutton. A stochastic approximation method. Ann. Math. Statist., 22(3):400-407, 09 1951.

[Ron15a] Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. CoRR, abs/1505.04597, 2015.

[Rot04a] Rother, Carsten, Kolmogorov, Vladimir, and Blake, Andrew. "grabcut": Interactive foreground extraction using iterated graph cuts. In ACM SIGGRAPH 2004 Papers, SIGGRAPH ’04, page 309-314, New York, NY, USA, 2004.

[Rum87a] Rumelhart, D. E., and McClelland, J. L.. Learning Internal Representations by Error Propagation, pages 318-362. MIT Press, 1987.

[Sab18a] Sabir, Ekraam, Rawls, Stephen, and Natarajan, Prem. Implicit language model in LSTM for OCR. CoRR, abs/1805.09441, 2018.

[Son15a] Song, S., Lichtenberg, S. P., and Xiao, J.. Sun rgb-d: A rgbd scene understanding benchmark suite. In IEEE Conf. on Computer Vision and Pattern Recognition, pages 567-576, 2015.

[Sti99a] Stigler, Daniel. Evaluation of classifiers for use in emergency detection systems, bachelors thesis, University of Applied Sciences Upper Austria, 2018.

[Tan15a] Tang, M., Ayed, I. B., Marin, D., and Boykov, Y.. Secrets of grabcut and kernel k-means. In IEEE International Conf. on Computer Vision (ICCV), pages 1555-1563, 2015.

[Tka20a] Tkachenko, Maxim, Malyuk, Mikhail, Shevchenko, Nikita, Holmanyuk, Andrey, and Liubimov, Nikolai. Label Studio: Data labeling software, Open source software available from https://github.com/heartexlabs/label-studio, 2020.

[Tom15a] Tommasi, Tatiana, Patricia, Novi, Caputo, Barbara, and Tuytelaars, Tinne. A Deeper Look at Dataset Bias, pages 37-55. Springer, 2017.

[Tor11a] Torralba, A. and Efros, A. A.. Unbiased look at dataset bias. In CVPR, pp. 1521-1528, 2011.

[Uij13a] Uijlings, Jasper, Sande, K., Gevers, T., and Smeulders, Arnold. Selective search for object recognition. International Journal of Computer Vision, 104:154-171, 09 2013.

[Wei16a] Weiss, Karl, Khoshgoftaar, Taghi, and Wang, DingDing. A survey of transfer learning. Journal of Big Data, 3, 12 2016.

[Wu10a] Wu, Mingjun, and Peng, Xianrong. Spatiotemporal context for codebook-based dynamic background subtraction. AEU - International Journal of Electronics and Communications, 64(8):739-747, 2010.

[Wu16a] Wu, Yonghui et al.. Google’s neural machine translation system: Bridging the gap between human and machine translation. CoRR, abs/1609.08144, 2016.

[Xia16a] Xiang, Yu, Kim, Wonhui, Chen, Wei, Ji, Jingwei, Choy, Christopher, Su, Hao, Mottaghi, Roozbeh, Guibas, Leonidas, and Savarese, Silvio. Objectnet3d: A large scale database for 3d object recognition. In ECCV 2016, pages 160-176, Springer Int. Publishing, 2016.

[Ye17a] Ye, L., Liu, Z., and Wang, Y.. Depth-aware object instance segmentation. In IEEE Int. Conf. on Image Processing (ICIP), pages 325-329, 2017.

[Zwe20a] Zwettler, Gerald Adam, Holmes III, David R., and Backfrieder, Werner. Strategies for training deep learning models in medical domains with small reference datasets. Journal of WSCG, 28 (1-2):37-46, 2020.