2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Multi-Task Deep Learning for Depth-based Person Perception in Mobile Robotics

Daniel Seichter*, Benjamin Lewandowski*, Dominik Höchemer*, Tim Wengefeld and Horst-Michael Gross

***Abstract*— Efficient and robust person perception is one of the most basic skills a mobile robot must have to ensure intuitive human-machine interaction. In addition to person detection, this also includes estimating various attributes, like posture or body orientation, in order to achieve user-adaptive behavior. However, given limited computing and battery capabilities on a mobile robot, it is inefficient to solve all perception tasks separately, especially when using computationally expensive deep neural networks. Therefore, we propose a multi-task system for person perception, comprising of a fast, depth-based region proposal and an efficient, lightweight deep neural network. Using a single network forward pass, the system simultaneously detects persons, classifies their body postures, and estimates the upper body orientations while retaining almost the same computation time as a single-task network. We describe how to handle a real-world multi-task scenario and conduct an extensive series of experiments in order to compare various network architectures and task weightings. We further show that multi-task learning improves the networks' performance compared to their single-task baselines. For training and evaluation, we combine an existing dataset for orientation estimation and a new, self-recorded dataset, consisting of more than 235,000 depth patches that is made publicly available to the research community.**

I. Introduction

Mobile robots often rely on sequential processing architectures when several types of information are of interest. In our ongoing research projects, which cover public environments from supermarkets [1] to hospitals [2] and domestic applications [3], our robots require a robust person perception. This includes the sequential application of a person detector [4], a body posture classification, and an upper body orientation estimation [5] for standing persons in order to enable socially aware navigation behaviors. Since these tasks are based on machine learning and were trained successively, this pipeline does not follow the human concept of learning. Children are able to simultaneously learn how to speak, walk and do social interactions instead of learning one ability after another. In machine learning, this approach of learning several tasks at the same time is known as multi-task learning and offers two main advantages over learning separate tasks sequentially [6]. On the one hand, it may lead to performance improvements because knowledge is shared over all tasks, which may also increase generalization capabilities. On the other hand, it accelerates inference since feature redundancy is reduced, and all tasks are solved with a single model.

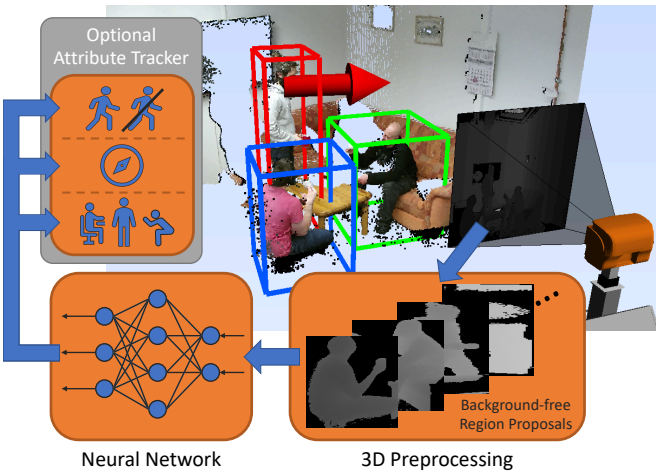


Fig. 1: Multi-task system overview: Regions of interest found in a 3D point cloud [7] are fed as depth patches into a fast deep neural network in order to detect persons and estimate their posture and upper body orientation (red arrow). The box color encodes the posture of the respective person: *standing* (red), *squatting* (blue), and *sitting* (green).

In this paper, we transform our person perception pipeline [4], [5] into a multi-task system, as shown in Fig. 1. Regions of interest found in a 3D point cloud are fed as background-free depth patches into a single deep neural network that solves the tasks person classification, posture classification, and orientation estimation in a single forward pass. Since, so far, only the orientation module was based on deep learning due to runtime issues, a deep learning-based approach leads to a significant performance increase for person detection and posture classification. However, due to 3D search space restriction, patch-based inference, and a lightweight neural network architecture, our system still runs in real time, even on CPU only. In our experiments, we examine various network architectures as well as training parameters and compare various multi-task networks to their deep learning single-task baselines. Thereby, we demonstrate that lightweight architectures can achieve the performance of more complex ones due to the applied multi-task approach.

We explicitly consider person and posture classification as separate tasks, since our robot should be aware of persons in its vicinity even if no posture could be derived due to heavy occlusions. Our system is able to distinguish between *standing*, *squatting*, and *sitting* and estimates the upper body

Authors are with Neuroinformatics and Cognitive Robotics Lab, Technische Universität Ilmenau, 98694 Ilmenau, Germany. daniel.seichter@tu-ilmenau.de

* Equal contribution.

This work has received funding from the German Federal Ministry of Education and Research (BMBF) to the project ROTATOR (grant agreement no. 03ZZ0437D) in the program Zwanzig20 and to the project FRAME (grant agreement no. 16SV7829K).

orientation as the continuous angle around the axis perpendicular to the ground. For training, validation, and test, we rely on our previously recorded orientation dataset [5] and a newly recorded dataset for person and posture classification.

In summary, our main contributions are:

1) an efficient multi-task system for person detection, posture classification, and upper body orientation estimation for real-time application on mobile robots with an exchangeable neural network depending on the application scenario and available hardware
2) a comparison of various network architectures regarding their applicability in a depth-based person perception system
3) a new dataset with more than 235,000 depth patches for person and posture classification
4) the publication of our new dataset, network code, and trained network weights to the research community[1]

II. Related Work

A. Person Detection

While classical image-based person detectors rely on handcrafted features [8], [9], [10], modern deep learning approaches, like [11], [12], [13], [14], learn relevant features from the data themselves. This has led to a great leap in performance. Unfortunately, color image-based detectors often rely on classifying at a lot of image scales in order to deal with persons in different distances and, therefore, are computationally intensive, especially when applying deep neural networks. More recent approaches address this problem by using a so-called region proposal network [15], [13] or by deriving both bounding boxes and classes directly from feature maps at different scales [12]. However, deep learning on full-sized images often still requires specialized hardware in order to run in real time on mobile platforms and a huge amount of data for training, best with high variance of the environments. With the advent of Kinect-like sensors, depth information was incorporated into color-based detectors to accelerate searching the scale space [16], [17]. Depth images or 3D point clouds were also used directly for detection since they often enable real-time application on CPU only [16], [7], [4]. Our multi-task system also relies on 3D point clouds to identify person candidates and classifies depth patches with a deep neural network in order to enable fast and accurate detections. As we will show, depth information further allows to quickly record a large amount of data for training.

B. Body Posture Classification

Human body posture estimation can be accomplished in different ways. To reduce the required amount of training data, an already trained skeleton estimation algorithm, such as [14], can be used to obtain a lower-dimensional feature vector, which subsequently can be classified [18], [19] more efficiently. However, estimating entire skeletons accurately from raw images is computationally intensive and not necessary if only the posture is of interest. In contrast, end-to-end posture classification approaches do not rely on skeleton estimation. In [4], a multi-class support vector machine is used to detect and distinguish standing and squatting persons in 3D point clouds. In [20], a Fast R-CNN [21] is applied to depth-based region proposals to categorize people according to their mobility aids in a clinical environment. Since we aim to solve all tasks in real time, we rely on an end-to-end approach as well.

C. Orientation Estimation

3D skeleton estimation approaches, like [22], [23], [24], also inherently provide an upper body orientation. The orientation can be derived by geometrical relations between 3D joints and bones. However, as we have shown in our previous work [5], it is not necessary to rely on such computationally intensive skeletons if only the upper body orientation is of interest. Instead, estimating the orientation directly is much faster and enables accurate social robot navigation. A direct estimation of the upper body orientation is very similar to estimating a head's orientation and can be implemented either as multi-class classification [25], [26], [27] or regression [28], [27], [5]. Since a classification introduces a systematic discretization error, we decided in favor of a regression for our multi-task system. Furthermore, in [5], we demonstrated that a lightweight neural network combined with depth-based image patches may be advantageous to using color patches while enabling a very accurate regression of the orientation in real time. Hence, we use this kind of lightweight network architecture as starting point in our experiments.

D. Multi-Task Deep Learning

Multi-task deep learning has shown excellent results for language processing [29] and computer vision problems [30], [31], [32], [33]. In this paper, we focus on heterogeneous multi-task learning. In contrast to homogeneous multi-task learning, i.e., learning similar tasks with the same output space, this allows combining different output spaces, such as regression and classification. Learning multiple tasks simultaneously can improve the overall system performance, as more task labels with different noise patterns can lead to more robust features and, thus, improve generalization [34]. However, the network's designer has to pay attention to additional training details, like higher requirements on the dataset, suitable task weightings [35], [36], [37], and the multi-task network architecture. The latter can be designed by loosely coupling several single-task networks [29], [30], but these so-called soft-parameter-sharing approaches neither reduce the computation time nor the network's complexity compared to their single-task counterparts. In contrast, hard-parameter-sharing approaches divide the architecture into several shared layers for feature extraction and multiple task-specific layers for calculating the final prediction for each task [31], [32]. Since most of the computations are done in the shared part, hard parameter sharing can significantly

[1]Our code and dataset are available at: https://www.tu-ilmenau.de/neurob/data-sets-code/depth-multi-task

reduce inference time while still benefiting from the multi-task scenario [6]. Hence, heterogeneous hard-parameter-sharing networks are well suited for deployment on a robotic platform and, thus, form the basis of our multi-task system.

III. System Overview

The idea of the proposed multi-task system (see Fig. 1) is to simultaneously detect persons, classify their body posture, and estimate their upper body orientation utilizing a single neural network on our mobile robot. For search space restriction, we apply the network only to regions of interest and not to the entire depth image of the robot's Kinect2.

The first step for determining regions of interest is to convert the depth image to a 3D point cloud. We then apply the candidate generator of [7] that labels each point individually as *ground plane*, *object*, or *fixed structure* by taking into account the assumptions that persons are always on the ground and have some free space above their heads. 3D points labeled as *objects* are then projected onto a 2D histogram in the ground plane, in which they are segmented into individual 3D point clusters. These clusters finally represent the regions of interest. Since deep learning on point clusters is challenging, we project each cluster back onto the Kinect2's image plane and crop it to the encasing bounding box, as displayed in Fig. 1. The whole preprocessing takes only 4ms on a single CPU core.

In [5], for orientation estimation, a system solely based on depth patches turned out to be superior to one using color patches or a combination of both. Hence, we rely on depth images for our multi-task system too. The preprocessed patches are free from background and form the input to our multi-task network. The network's output comprises whether a particular patch represents a person or not (is person) as well as the person's posture (*standing*, *squatting*, or *sitting*) and upper body orientation as a continuous angle.

On our robot, each person attribute is further tracked with our modular probabilistic tracker [38] (see Fig. 1). Since our derived network architectures use dropout before fully-connected layers, we are able to model the uncertainty for each task using dropout sampling [39]. This helps tracking and, thus, makes the robotic application even more robust.

The entire multi-task system and its insights are further visualized in the attached video to this paper[2]. Due to the preprocessing, patch-based inference, and a lightweight deep neural network, our multi-task system runs in real time either on an NVIDIA Jetson AGX Xavier or even on CPU only, depending on the chosen network architecture. In the following, we examine various network architectures suitable for the proposed multi-task system and compare their performance for all three tasks.

IV. Datasets

Training a multi-task network requires data with labels for all tasks. However, to the best of our knowledge, no depth image-based public dataset did meet all of our requirements of having labels for the body posture as *standing*, *squatting*, and *sitting*, the upper body orientation, and person classification, i.e. providing both person and non-person samples. Fortunately, the depth-based nature of our system allows us to record data easily. Since input patches to the neural network are background-free, persons can be recorded continuously at a single place. Therefore, we recorded a new person and posture classification dataset that we make available to the research community[1]. It complements our already published NICR RGB-D Orientation Dataset [5], which was captured in a similar manner. With both datasets combined, we have more than 340,000 depth patches for training, validation, and test. Statistics about the data are shown in Fig. 2.

	train.	valid.	test
Orientation [5]	57,717	19,368	31,420
Standing [5]	57,717	19,368	31,420
Standing	22,150	8,054	6,602
Squatting	9,308	2,831	5,056
Sitting	5,809	4,816	3,491
Person [5]	57,717	19,368	31,420
Person w/ posture	37,267	15,701	15,149
Person w/o posture	2,722	1,602	1,065
Non-person	86,781	17,594	59,423
Female/male [5]	7/14	3/4	5/4
Female/male	4/10	2/3	1/7
Intersections	0	1	3

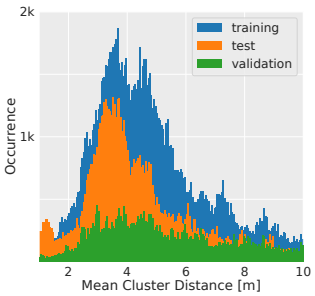


Fig. 2: Left: Number of patches per task (orientation, posture, is person), number of female and male persons, and number of persons appearing in both datasets. Right: Distance distribution of the new data.

A. NICR RGB-D Orientation Dataset

This dataset [5] was recorded for regression-based upper body orientation estimation. It consists of more than 105,000 RGB-D patches of 37 standing persons who were captured with five static Kinect2 devices simultaneously, placed in a half circle and in different distances. A learned background model was applied to each recorded depth frame in order to create background-free person patches. The upper body orientation has been automatically annotated using a highly precise external tracking system. The samples were divided into subsets for training, validation, and test with each person being assigned to exactly one of them. Since each sample represents a standing person, we can use this dataset for the posture and person classification tasks as well.

B. NICR Multi-Task Dataset

This new dataset was created using our robot's Kinect2 with focus on posture classification and person detection. Persons were recorded in distances of about 2m to 10m to the robot (see Fig. 2 right). To simplify labeling, only one body posture was recorded during a single session. For the *sitting* posture various kinds of chairs and stools were used. In order to introduce some occlusions, we also added various objects, such as shopping carts or cardboard boxes, to the scene. Since we had a static recording setup, background-free patches could be generated by simply subtracting a learned background model. To make the data being similar to the results after our preprocessing step, we further applied the candidate generator of [7] to the foreground point cloud

[2]The attached video is also available at: https://youtu.be/wRLk1kcsy5Y

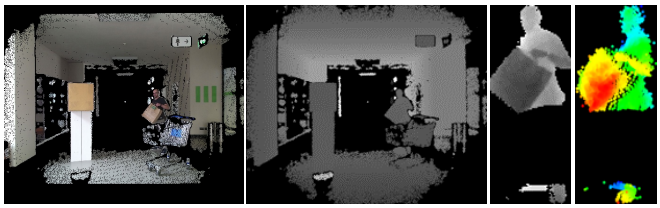


Fig. 3: Color and depth image of a recorded scene in our new dataset as well as the extracted patch and point cluster.

to generate patches. We verified all labels by hand after recording. Patches that could not be clearly assigned to one of the three body posture classes, e.g., due to heavy occlusions, were labeled as *person without posture* (see Fig. 2). Fig. 3 shows an example scene, the extracted foreground patch, and the corresponding point cluster. Negative samples were captured by manually driving the robot through three buildings of our university and a supermarket after closing time, ensuring that no person was included in the recordings. Afterwards, we extracted all regions of interest using the aforementioned preprocessing and labeled them as *non-person*. In total, our new dataset consists of more than 235,000 samples. Similar to the orientation dataset [5], we divided all samples into a training, validation, and test subset and assigned each person to exactly one of them. Furthermore, we ensured that persons who appear in both datasets are assigned to the same subset as their counterparts in the orientation data. *Non-person* samples were assigned to subsets based on the recorded building. Note that samples of the supermarket were assigned to the test set only in order to prevent overfitting to one of our operational environments.

V. Multi-Task Person Perception

In order to derive neural networks capable of handling all three tasks in our multi-task system, we conducted a series of experiments on the aforementioned datasets. To assess the networks' performance, we first trained several single-task baselines for each task independently. Subsequently, we gradually extended the number of tasks covered by the networks, taking into account the challenges when training multiple tasks at once.

A. Experimental Setup and Network Training

As the tasks are heterogeneous and, therefore, have different output spaces, require different loss functions and data, we start by summarizing the experimental setup for each task. Furthermore, we focus on network training and common hyperparameters.

Orientation estimation: For orientation estimation, we follow [5] and rely on biternion output encoding and von Mises loss function [28]. We used our NICR RGB-D Orientation Dataset for training and validation, and report the mean absolute angular error (MAE) on the test set.

Posture classification: Since the goal is to distinguish three classes, we use softmax output encoding in conjunction with cross-entropy loss. For network training, we joined the proposed NICR Multi-Task Dataset (persons with posture only) and the NICR RGB-D Orientation Dataset. Since the resulting dataset is not balanced, we report the balanced accuracy (bAcc) on the joined test sets.

Person classification: Due to the preprocessing pipeline, person detection is simplified to another classification task. Hence, we use softmax output and cross-entropy loss as well. Both datasets (NICR RGB-D Orientation Dataset and NICR Multi-Task Dataset) are joined for training, validation, and test. For evaluation, we report the F_1 score on the test set, as it is common practice.

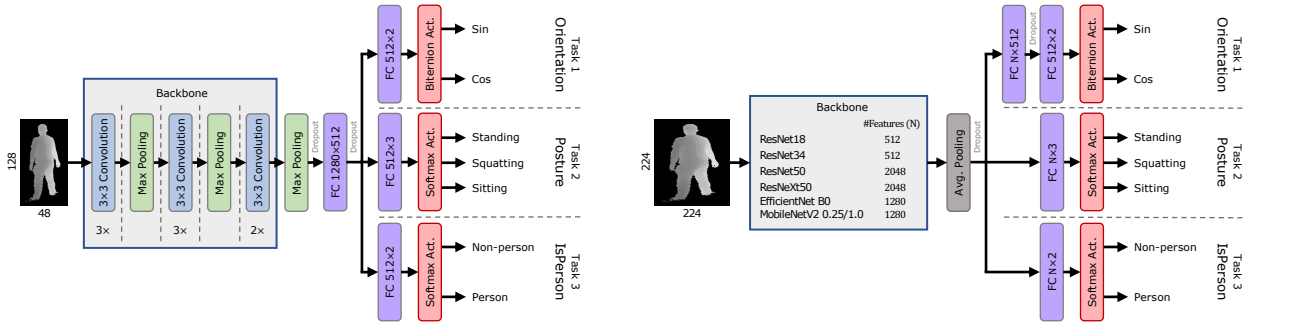
Network architectures: Due to restricted computational resources, in our previous work [5], we only focused on lightweight architectures, especially designed for mobile robotic applications, such as Deep Orientation Network (DONet) [5] and MobileNetV2 [40]. However, in this paper, we aim to solve multiple tasks at the same time using a single network. Therefore, we integrated more sophisticated backbone architectures, such as ResNet [41], ResNeXt [42], and recently published EfficientNet [43] in our study as well. Fig. 4 shows the network architectures used in this paper in detail. Note that the design for the task-specific layers is different for the DONet-based architecture (see Fig. 4a) compared to the ones that use a more sophisticated backbone (see Fig. 4b). These backbones are designed with regard to image classification. We found that a single fully-connected layer on top of the final average pooling of the backbone works well for posture and person classification. However, for orientation estimation another fully-connected layer is necessary. We assume that the number of weights in a single fully-connected layer is too small to adequately solve the regression problem. This finding coincides with [5].

Network training: To further increase the number of samples, we applied random horizontal flipping as data augmentation. For optimization, we used both SGD with momentum of 0.9 and Adam [44] with initial learning rates of $\{0.001, 0.01, 0.05, 0.1, 0.2\}$ and $\{0.0001, 0.0005, 0.001, 0.01\}$, respectively. During training, the learning rate was decreased after each batch of 128 samples using a polynomial decay. For multi-task training, due to our data handling, each batch always contained samples for all the tasks considered. The final weight configuration was chosen within 200 epochs based on the performance on the respective validation set. All networks were trained on NVIDIA GeForce 2080 Ti and TitanRTX GPUs using PyTorch [45]. For further details and other hyper parameters, we refer to the implementation[1].

B. Single-Task Baselines

To identify network architectures suitable for all tasks, we conducted extensive single-task experiments, varying the network architecture, the optimizer, and the initial learning rate. Only one of the heads shown in Fig. 4 was activated at the same time while training the single-task networks. Tab. I summarizes the best results obtained for each network architecture and presents a per-task ranking.

It is obvious that the performance increases as the network's depth and complexity increase. The lightweight net-



(a) based on DONet [5] (VGG-like [46])

(b) based on a more sophisticated backbone

Fig. 4: Network architectures used for multi-task learning as well as for single-task baseline training (only one active head).

work architectures DONet and MobileNetV2 0.25 (width multiplier of 0.25) are almost consistently ranked last. Surprisingly, for all tasks, the best result was not obtained using one of the complex residual networks but using MobileNetV2 1.0 (width multiplier of 1.0). Another mobile network architecture, the recently published EfficientNet B0 is ranked second or third. This result suggests that recent mobile network architectures can compete with complex residual networks. However, MobileNetV2 and EfficientNet B0 feature depthwise and grouped convolutions to reduce the number of parameters and computations. Currently, these operations cannot be optimized to the same level as vanilla convolutions used in DONet and ResNet. Therefore, we further examined the runtime on CPU and GPU for all network architectures. Fig. 5 shows the results when processing a batch of 20 samples (average number of candidates per image) at inference time. The results indicate that all network architectures allow real-time execution on GPU using a NVIDIA Jetson AGX Xavier with at least 20 Hz. Furthermore, it is obvious that the overall number of parameters has less influence on the runtime than the used type of convolution. ResNeXt50 and EfficientNet B0 heavily use grouped convolutions and, thus, result in a similarly slow runtime. Moreover, ResNet18 has far more parameters than MobileNetV2 but can be executed at the same speed. On CPU, the type of convolution is not as crucial as for GPU. Rather, it seems that the runtime is mostly influenced by the overall number of convolutions and feature maps in the network. However, only DONet and MobileNetV2 0.25 meet our real-time requirement on CPU with at least 5 Hz.

Model	Orientation MAE ↓	Posture bAcc ↑	IsPerson F_1 ↑	Ranking O - P - I
DONet [5]	5.211*	0.9242*	0.9962*	6 - 7 - 8
MobileNetV2 0.25 [40]	5.256	0.9201	0.9983	7 - 8 - 3
MobileNetV2 1.0 [40]	4.601	0.9488	0.9990	**1 - 1 - 1**
EfficientNet B0 [43]	4.729	0.9483	0.9986	3 - 3 - 2
ResNet18 [41]	5.344	0.9347	0.9970	8 - 6 - 6
ResNet34 [41]	5.031	0.9422*	0.9969	5 - 5 - 7
ResNet50 [41]	4.771	0.9433	0.9978	4 - 4 - 5
ResNeXt50 [42]	4.728	0.9485*	0.9982	2 - 2 - 4

TABLE I: Single-task evaluation metrics and task rankings obtained on the respective test set when training various network architectures for each task. * indicates that optimizing using classical SGD instead of Adam led to better results.

Based on the obtained single-task performances and in favor of a fast runtime on both CPU and GPU, we decided to stick to DONet and both versions of MobileNetV2 for our multi-task experiments. Both DONet and MobileNetV2 0.25 allow real-time execution even on CPU. MobileNetV2 1.0 performs best while being faster than non-mobile network architectures on GPU. The results for both the selected network architectures as well as the dropped ones help to assess the performance of the multi-task system presented in the following.

C. Dual-Task Experiments

As the output spaces for orientation estimation and both classification tasks are heterogeneous, we started by building a dual-task system in order to understand, how to combine heterogeneous tasks. We selected orientation estimation and posture classification for this study as posture classification is more challenging and the number of training examples is lower than for person classification.

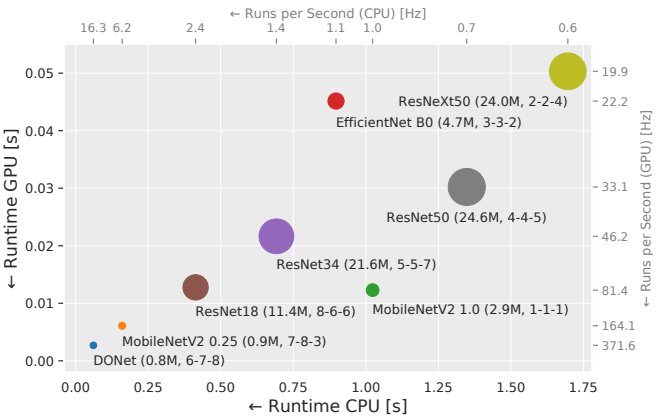


Fig. 5: Runtime comparsion of several multi-task network architectures on CPU (Intel Core i7-7700H, PyTorch, Intel MKL-DNN) and GPU (NVIDIA Jetson AGX Xavier, TensorRT, float16) when processing a batch of 20 samples. For each architecture, the overall number of parameters (circle diameter) and the single-task ranking is noted in brackets. Note that the runtimes were measured for the triple-task case but hardly differ for the single-task case.

Training a multi-task system requires handling and combining multiple losses. In general, each loss L_i for task i is weighted with a factor λ_i, before accumulating all losses to get the overall loss L. With Dynamic Weight Average (DWA) [37], GradNorm [36], and uncertainty weighting [35], several approaches for determining the weights λ_i automatically have been proposed recently. Unfortunately, in our scenario, none of them led to good results. With GradNorm, the loss diverged within few epochs. Using uncertainty weighting, a weight close to zero was assigned to the classification task in early epochs and kept throughout training. Only DWA led to a stable training. However, the results for one of the two tasks were always consistently worse compared to the single-task baselines. Therefore, we performed a grid search for determining suitable loss weights. For our dual-task system, the overall loss was calculated as:

$$L = \lambda_O \cdot L_O + \lambda_P \cdot L_P \tag{1}$$

Since several initial learning rates are examined, we decided to set $\lambda_O + \lambda_P = 1$.

Another problem to deal with is the selection of the best epoch. Due to the distinct task-specific layers, the best epoch can differ between the tasks. Unfortunately, most of the related works do not explain their procedure for selecting the best training epoch. We found that selecting the best epoch based on the accumulated per-task rankings over all epochs works best.

The results of our grid search for DONet and selected initial learning rates are depicted in Fig. 6. For all initial learning rates, similar trends are shown. First, posture classification heavily benefits from multi-task training. Second, for orientation estimation, $\lambda_O \geq 0.8$ is required to reach the single-task baseline. Unfortunately, through multi-task learning, an improvement compared to the single-task baseline cannot be achieved for orientation estimation.

Fig. 7 further summarizes the results for the top-5 networks for each architecture and their corresponding loss weightings. These results support our findings across all

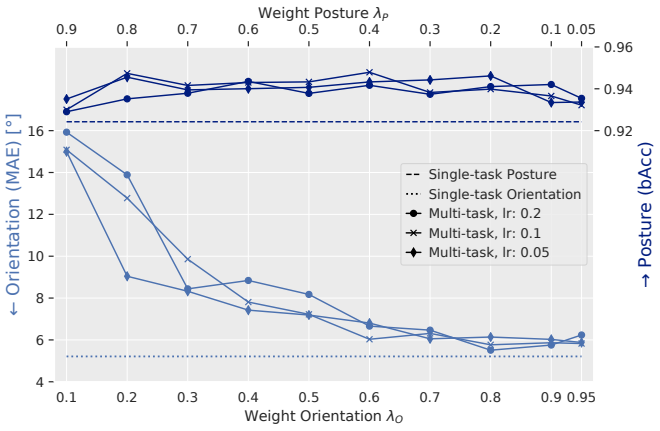


Fig. 6: Results of dual-task loss weighting grid search for DONet and various initial learning rates.

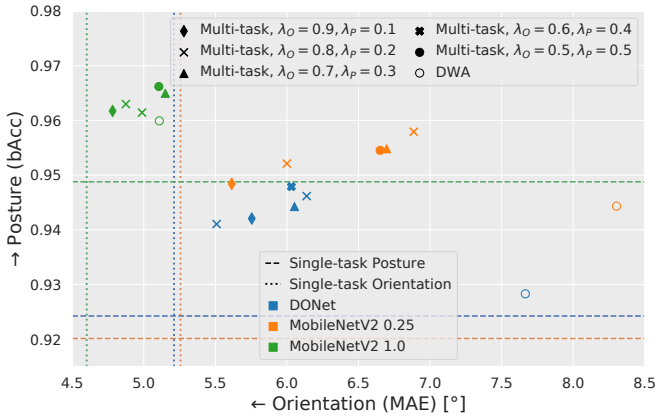


Fig. 7: Top-5 results over all learning rates and loss weightings for the considered network architectures DONet and MobileNetV2 in our multi-task system. In addition, the best result using DWA [37] is depicted for each architecture.

network architectures and confirm the effectiveness of multi-task learning for our scenario. Moreover, it becomes obvious that tuning the loss weights manually leads to better results than using DWA in our multi-task scenario.

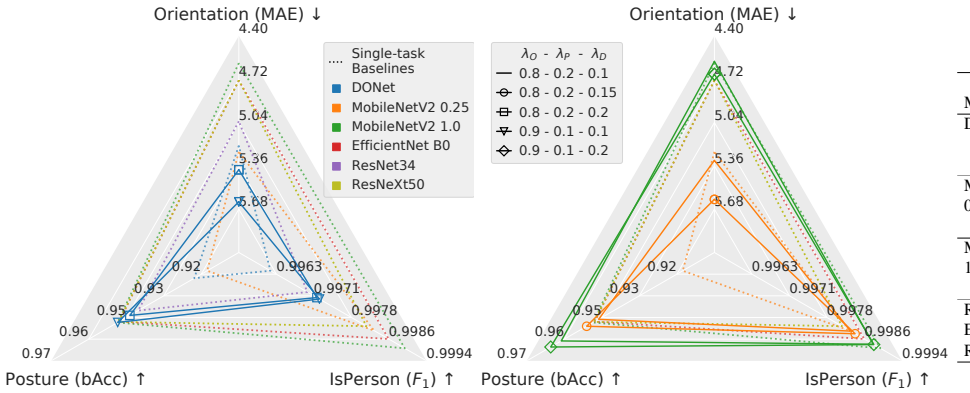
D. Triple-Task Experiments

With the findings of the previous subsection, we extended our dual-task system and included the person classification task as well. Since we know that the orientation task requires a much greater weight without affecting the posture task too much, we fixed the ratio between both tasks according to best results of our dual-task experiments. To consider the third task, we took a similar approach as before and modified our loss function as follows to include the is-person loss L_I:

$$L = (1 - \lambda_I)(\lambda_O \cdot L_O + \lambda_P \cdot L_P) + \lambda_I \cdot L_I \tag{2}$$

Note that after modifying the loss function in this way, the sum of the resulting weighting factors is still 1 and, therefore, does not scale the learning rate indirectly. For determining a suitable λ_I, we used the two best performing ratios $\{\lambda_O = 0.8, \lambda_P = 0.2\}$ and $\{\lambda_O = 0.9, \lambda_P = 0.1\}$ and performed a grid search for λ_I using values between 0.05 and 0.9.

Fig. 8 summarizes the best results for all considered network architectures and compares them to relevant single-task baselines. For all architectures, we could successfully extend our dual-task system so that it is able to handle person detection as well. As shown in Fig. 8a, multi-task learning greatly improves the performance of DONet for person classification, even catching up with the larger ResNet34. Moreover, for posture classification, DONet is able to compete with the best single-task baselines. For MobileNetV2 0.25 (shown in Fig. 8b), a similar trend is emerged. Both person and posture classification significantly benefit from multi-task learning. The obtained results are of the same quality as the single-task baselines of EfficientNet B0 and even ResNeXt50. However, similar to our dual-task experiments, triple-task learning does only lead to results on par for orientation estimation without any further improvement. For



(a) Results for DONet

(b) Results for both MobileNetV2

Model	Multi Task	Orientation MAE ↓	Posture bAcc ↑	IsPerson F_1 ↑
DONet		5.211*	0.9242*	0.9962*
	✓	5.389*	0.9449*	0.9971*
	✓	**5.625***	**0.9487***	**0.9972***
MobileNetV2 0.25		5.256	0.9201	0.9983
	✓	5.320	0.9470	0.9983
	✓	**5.608**	**0.9509**	**0.9984**
MobileNetV2 1.0		4.601	0.9488	0.9990
	✓	4.589	0.9590	0.9988
	✓	**4.683**	**0.9624**	**0.9988**
ResNet34		5.031	0.9422*	0.9969
EfficientNet B0		4.729	0.9483	0.9986
ResNeXt50		4.728	0.9485*	0.9982

(c) Exact values

Fig. 8: Top-2 triple-task results for DONet and both MobileNetV2. Networks selected for application in our multi-task system are printed in bold. * further indicates that optimizing using classical SGD instead of Adam led to better results.

MobileNetV2 1.0 (shown in Fig. 8b), posture classification could be further improved through multi-task learning.

For application on our mobile robots, we finally selected the networks printed in bold in Fig. 8c. This emphasizes person and posture classification but also leads to a small performance drop in orientation estimation. However, the results for orientation estimation are already sufficient for our application scenario. If a GPU is available on our robot, we stick to MobileNetV2 1.0 for our multi-task system (see Fig. 1), if not, DONet and MobileNetV2 0.25 still allow real-time execution without fully utilizing the CPU (see Fig. 5).

E. Comparison to other Person Perception Approaches

For a final assessment, we compare the performance of the networks we selected for application on all three tasks to reference approaches from the literature. Since, to the best of our knowledge, there is no other multi-task system designed to handle the same tasks, we consider each task distinctly. The results show, that our multi-task system performs on par or even better to other state-of-the-art approaches suitable for mobile applications. Note that the same training set could only be used for orientation estimation, whereas for posture classification and person detection, each approach was trained on its own dataset.

Orientation Estimation On our NICR RGB-D Orientation test set, our system outperforms the point cloud approach of [27] by a margin of at least 5.585° (MAE of 11.21°). Compared to the results in [5], our triple-task DONet is on par with the original DONet while solving three tasks at the same time instead of only one (5.625° vs. 5.44° MAE). Moreover, our triple-task MobileNetV2 0.25 (input size of 224×224, width multiplier of 0.25) performs similar to its counterpart MobileNetV2 0.75 (input size of 96×96, width multiplier of 0.75) [5] with comparable FLOPs (5.608° vs. 5.43° MAE). However, our triple-task MobileNetV2 1.0 outperforms all network architectures examined in [5].

Posture classification For posture classification, we compare our system to the multi-class SVM-based approach proposed in [4]. On the test set of our new NICR Multi-Task Dataset, all triple-task networks outperform the approach in [4] by a large margin (0.9487, 0.9509 and 0.9624 vs. 0.8276 bAcc). This also demonstrates the improvements modern deep neural networks can achieve.

Person Detection We evaluated the person detection performance of our multi-task system including preprocessing on the test set of the supermarket dataset [4] (includes heavy occlusions). This allows a comparison to depth-based [7], [4] and color image-based detectors [9], [10], [14], [12]. The results in Fig. 9 indicate that our system outperforms all depth and classical color image-based detectors. Even the deep learning-based detector YOLOV3 [12] is outperformed. Furthermore, all triple-task networks beat their corresponding single-task counterparts, which confirms that multi-task learning improves generalization capabilities. Through multi-task learning, MobileNetV2 0.25 almost reaches the single-task performance of MobileNetV2 1.0. The skeleton estimator OpenPose [14] performs best in this scenario as it is able to detect even single individual body parts resulting in superior performance for heavy occlusions. However, this approach does not meet our real-time requirements and, therefore, is not applicable in our application scenario.

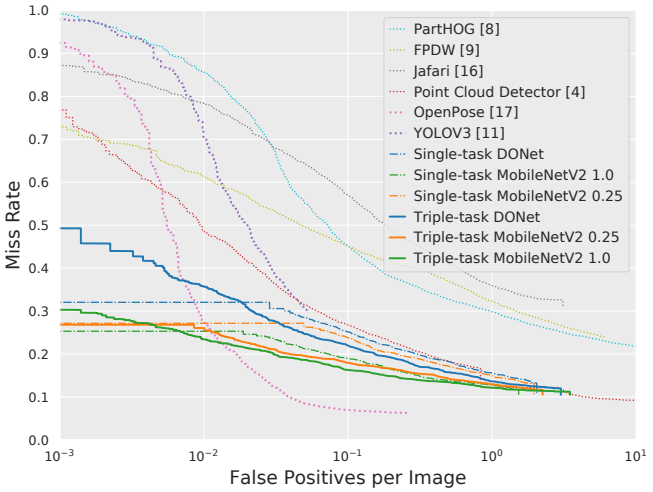


Fig. 9: Results for person detection as DET curves on the test set of the supermarket dataset presented in [4].

VI. Conclusion

We have presented a multi-task system for person detection, body posture classification, and upper body orientation estimation suitable for mobile applications. Due to the combination of preprocessing for detecting regions of interest, the processing of fixed size depth patches, and a lightweight deep neural network, our system runs very energy-efficient and in real time on mobile robots, even on CPU only. We analyzed various network architectures and task weightings and demonstrated that multi-task learning does improve the performance in person detection and posture classification. For both tasks, we recorded a new dataset consisting of more than 235,000 depth patches and combined it with our previously recorded orientation dataset. To support research for depth-based person perception, we make the new dataset as well as our network code and weights available to other researchers[1].

Acknowledgment

The authors would like to thank the Sander family for allowing the recording of negative data in their Edeka supermarket in Ilmenau after closing time.

References

[1] H.-M. Gross, *et al.*, "TOOMAS: Interactie shopping guide robots in everyday use – final implementation and experiences from long-term field trials," in *Proc. of IROS*, 2009, pp. 2005–2012.

[2] H.-M. Gross, *et al.*, "Mobile robot companion for walking training of stroke patients in clinical post-stroke rehabilitation," in *Proc. of ICRA*, 2017, pp. 1028–1035.

[3] H. M. Gross, *et al.*, "Living with a mobile companion robot in your own apartment - final implementation and results of a 20-weeks field study with 20 seniors," in *Proc. of ICRA*. IEEE, 2019, pp. 2253–2259.

[4] B. Lewandowski, *et al.*, "A fast and robust 3d person detector and posture estimator for mobile robotic application," in *Proc. of ICRA*, 2019, pp. 4869–4875.

[5] B. Lewandowski, *et al.*, "Deep orientation: Fast and robust upper body orientation estimation for mobile robotic applications," in *Proc. of IROS*, 2019, pp. 441–448.

[6] S. Ruder, "An overview of multi-task learning in deep neural networks," *CoRR*, vol. abs/1706.05098, 2017.

[7] O. H. Jafari, *et al.*, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," in *Proc. of ICRA*, 2014, pp. 5636–5643.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of CVPR*, 2005, pp. 886–893.

[9] P. F. Felzenszwalb, *et al.*, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.

[10] P. Dollar, *et al.*, "The Fastest Pedestrian Detector in the West," in *British Machine Vision Conference (BMVC)*, 2010, pp. 68.1–68.11.

[11] M. Eisenbach, *et al.*, "Cooperative multi-scale convolutional neural networks for person detection," in *Int. Joint Conf. on Neural Networks (IJCNN)*, 2016, pp. 267–276.

[12] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[13] K. He, *et al.*, "Mask R-CNN," in *IEEE Int. Conf. on Computer Vision*, 2017, pp. 2961–2969.

[14] Z. Cao, *et al.*, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in *Proc. of CVPR*, 2017, pp. 7291–7299.

[15] S. Ren, *et al.*, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 91–99.

[16] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *Proc. of IROS*, 2011, pp. 3838–3843.

[17] M. Munaro, *et al.*, "Tracking people within groups with RGB-D data," in *Proc. of IROS*, 2012, pp. 2101–2107.

[18] S. Neili, *et al.*, "Human posture recognition approach based on convnets and svm classifier," in *Int. Conf. on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2017, pp. 1–6.

[19] B. Cao, *et al.*, "Human posture recognition using skeleton and depth information," in *WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, 2018, pp. 275–280.

[20] A. Vasquez, *et al.*, "Deep detection of people and their mobility aids for a hospital robot," *CoRR*, vol. abs/1708.00674, 2017.

[21] R. Girshick, "Fast r-cnn," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[22] D. Tome, *et al.*, "Lifting from the deep: Convolutional 3d pose estimation from a single image," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 5689–5698.

[23] C. Zimmermann, *et al.*, "3D Human Pose Estimation in RGBD Images for Robotic Task Learning," in *Proc. of ICRA*, 2018, pp. 1986–1992.

[24] T. Schnürer, *et al.*, "Real-time 3d pose estimation from single depth images," in *Int. Joint Conf. on Computer Vision, Imaging and Computer Graphics - Theory and Applications*, 2019, pp. 716–724.

[25] M. Raza, *et al.*, "Appearance based pedestrians' head pose and body orientation estimation using deep learning," *Neurocomputing*, vol. 272, pp. 647–659, 2018.

[26] Y. Kawanishi, *et al.*, "Misclassification tolerable learning for robust pedestrian orientation classification," in *Int. Conf. on Pattern Recognition*, 2016, pp. 486–491.

[27] T. Wengefeld, *et al.*, "Real-time person orientation estimation using colored pointclouds," in *IEEE Europ. Conf. on Mobile Robotics (ECMR)*, 2019.

[28] L. Beyer, *et al.*, "Biternion nets: Continuous head pose regression from discrete training labels," in *German Conf. on Pattern Recognition*, 2015, pp. 157–168.

[29] S. Ruder, *et al.*, "Sluice networks: Learning what to share between loosely related tasks," *arXiv preprint arXiv:1705.08142*, 2017.

[30] I. Misra, *et al.*, "Cross-stitch networks for multi-task learning," in *Proc. of CVPR*, 2016.

[31] Z. Zhang, *et al.*, "Facial landmark detection by deep multi-task learning," in *Europ. Conf. on Computer Vision*, 2014, pp. 94–108.

[32] R. Ranjan, *et al.*, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 41, no. 1, pp. 121–135, Jan 2019.

[33] I. Kokkinos, "Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," *CoRR*, vol. abs/1609.02132, 2016.

[34] R. Caruana, *Multitask Learning*. Boston, MA: Springer US, 1998, pp. 95–133.

[35] A. Kendall, *et al.*, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *CoRR*, vol. abs/1705.07115, 2017.

[36] Z. Chen, *et al.*, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," *CoRR*, vol. abs/1711.02257, 2017.

[37] S. Liu, *et al.*, "End-to-end multi-task learning with attention," in *Proc. of CVPR*, 2019.

[38] S. Müller, *et al.*, "A multi-modal person perception framework for socially interactive mobile service robots," *Sensors*, vol. 20, no. 3, p. 722, 2020.

[39] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Int. Conf. on Machine Learning (ICML)*, 2016, pp. 1050–1059.

[40] M. Sandler, *et al.*, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. of CVPR*, 2018.

[41] K. He, *et al.*, "Identity mappings in deep residual networks," in *Europ. Conf. on Computer Vision*, 2016, pp. 630–645.

[42] S. Xie, *et al.*, "Aggregated residual transformations for deep neural networks," in *Proc. of CVPR*, 2017.

[43] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Int. Conf. on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[44] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Int. Conf. Learn. Represent.*, 2015.

[45] A. Paszke, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of ICLR*, 2015.