IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 September 2022, accepted 12 September 2022, date of publication 15 September 2022,
date of current version 26 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3206948

RESEARCH ARTICLE

Generating Synthetic Depth Image Dataset for Industrial Applications of Hand Localization

ALES VYSOCKY, STEFAN GRUSHKO, TOMAS SPURNY, ROBERT PASTOR, AND TOMAS KOT
Department of Robotics, Faculty of Mechanical Engineering, VSB—Technical University of Ostrava, 70800 Ostrava, Czech Republic

Corresponding author: Ales Vysocky (ales.vysocky@vsb.cz)

This work was supported by the Research Platform focused on Industry 4.0 and Robotics in Ostrava Agglomeration Project through the Operational Program Research, Development and Education, under Project CZ.02.1.01/0.0/0.0/17_049/0008425. This article has been also supported by specific research project SP2022/67 and financed by the state budget of the Czech Republic.

ABSTRACT In this paper, we focus on the problem of applying domain randomization to produce synthetic datasets for training depth image segmentation models for the task of hand localization. We provide new synthetic datasets for industrial environments suitable for various hand tracking applications, as well as ready-to-use pre-trained models. The presented datasets are analyzed to evaluate the characteristics of these datasets that affect the generalizability of the trained models, and recommendations are given for adapting the simulation environment to achieve satisfactory results when creating datasets for specialized applications. Our approach is not limited by the shortcomings of standard analytical methods, such as color, specific gestures, or hand orientation. The models in this paper were trained solely on a synthetic dataset and were never trained on real camera images; nevertheless, we demonstrate that our most diverse datasets allow the models to achieve up to 90% accuracy. The proposed hand localization system is designed for industrial applications where the operator shares the workspace with the robot.

INDEX TERMS Depth camera, hand tracking, hand localisation, image segmentation, synthetic dataset, domain randomization.

I. INTRODUCTION

Safety is a key factor in collaborative robotics. With an increasing tendency of working closer to the robot without protective barriers, reactive systems based on collision detection became insufficient for fluent collaboration. Predictive systems that use cameras, laser scanners, and other sensors to detect the presence of the operator is one of the solution for the future of human-robot cooperation. This paper focuses on robot-assisted assembly conditions, where the operator shares the workplace with the robot and collaborate in a close proximity.

In human-machine interaction, hand gesture recognition and processing is a key topic because they represent a natural way for humans to communicate non-verbally. Using a recognized gesture, we can create a specific command to control the robot; by knowing the position of the hand in the workspace, we can guide the robot to a specific location; finally, information about the presence of the hand in the workspace can be used for safety measures. Camera-based safety systems often use detection and tracking, which are utilized to localize the operator in the workspace and, potentially, adapt the technological process [1], [2], [3] in order to insure the safety of each operator. In our approach we focus on localization of fingers, hands and whole arms in the workplace from the top view camera. The output of the hand recognition system can be used to control robotic applications using various [4], [5], [6] gesture-based interfaces.

In this paper, we focus on the problem of semi-automatic generation of synthetic datasets for training depth-image segmentation models for the task of hand localization. Our work contributes to the topic by providing new synthetic datasets for industrial environments suitable for various hand tracking applications as well as ready-to-use pre-trained models. We also provide a simulation scene that can be customized and optimized for specific applications. We further elaborate and analyze the characteristics of these datasets that affect the resulting generalization ability of the trained models, and

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

provide tips on how to adapt the used simulation environment to achieve satisfactory results when creating datasets for customized applications. The models in this paper were trained exclusively on the synthetic dataset and were never trained with real camera images, nevertheless, we demonstrate that our most diverse dataset allows the models to achieve 90% accuracy.

II. RELATED WORK

A. HAND LOCALIZATION

Over the years, approaches to dealing with the hand localization problem have gradually improved. Initial color-based methods analyze different components in the corresponding color space (RGB, HSV) [7]. This approach usually requires tuning the application for specific conditions and environments, and therefore the results of these approaches are strongly influenced by ambient lighting, background and obstacles.

More sophisticated approaches are usually based on machine learning models trained on a specific dataset. Modifications and extensions to the dataset used allow the created model to be generalized, making it less sensitive to changes in the environment. Approaches with relatively limited generalization capability include methods such as support vector machines [8], [9] and hidden Markov models [10]. With the development and successful application of deep learning in the field of image recognition, recent work in hand detection has mainly focused on convolutional network models, which have enabled to significantly increase the accuracy of image segmentation methods [11]. Encoder-decoder networks and their derivatives have been successful in solving the image segmentation task in a wide range of conditions and environments. U-Net is an example of such a coder-decoder network for image segmentation.

The majority localization methods generally use RGB camera images because these cameras are widely available. Specialized sensors such as Kinect and Leap Motion have been applied in a large number of human body tracking tasks as a source of information for human-machine interaction. However, while the default skeleton tracking in Kinect achieves high accuracy and can be even improved by correction [12], it cannot track the hands without seeing the entire human body. On the other hand, the Leap Motion controller has various drawbacks that make it unsuitable for a relatively large workspace [13]. The use of depth camera can increase the robustness of the application in terms of less emphasis on color and more attention to the shape during detection. This is advantageous in low-light conditions or if the user wears protective gloves of different colors. In the vast majority of work, the actual search of the hand region is performed in a limited area that has been defined by a simple [14] heuristic that allows the camera image to be cropped to contain only the hand-related part of the depth image. Alternatively, some approaches make use of sliding window predictions over the entire image in order to localize the hand.

B. DATASET COLLECTION

The most important step in training a machine learning model is to obtain a dataset sufficiently large to accurately represent the real-world domain in a wide range of circumstances and contexts. A typical approach for obtaining such a dataset is to manually label hundreds of thousands of images to define the ground truth for each image, or to use third-party labeled datasets available on various platforms. While obtaining sample data can be relatively straightforward, the subsequent labeling of the data can take an enormous amount of time, depending on the complexity of the scene and the desired diversity of labels. Melireddi *et al.* have demonstrated the use of coloured gloves [15] to label hand regions in the created dataset image. Another approach to the labelling automation is the assumption that the hand is the closest object to the camera [16] and can be found by applying a color threshold to the image. An alternative approach [17] is to use tracking sensors, or infrared markers [18] fastened to the hand and fingers to automatically generate the labels.

Datasets obtained by collecting images from a real camera have the advantage of being close to the real domain but at the same time - possess problem of limited range of captured conditions, since the environment arrangements which were provided during acquisition is limited and usually cannot cover the full range of scenarios (e.g. changing positions of obstacles in the view, lighting, reflections, shadows).

Synthetic datasets provide an alternative to traditional manual collection and labeling. These datasets are created programmatically by simulating a domain, or by combining real images with a known ground truth into new ones [19], or by combining these methods and overlaying labeled objects over a randomized simulated scene [20]. Each method allows to produce arbitrary large fully-labeled datasets [21]. Prepared simulated environment allows to generate extensive synthetic datasets by adjusting the conditions and and applying augmentations to the generated images [22]. Keskin *et al.* demonstrated synthetic dataset generation based on a fully simulated scene [23]. They used a 3D skinned mesh with a skeleton defining parts of the hand and links of the fingers, which were used for both animating the mesh and creating the ground truth labels. This solution reduces the cost of preparing datasets while increasing data diversity and labeling accuracy.

In general, approaches to generating synthetic datasets can be divided into two main groups depending on their appearances: realistic and randomized datasets. Realistic datasets have an obvious advantage: they are very similar to the real environment, which allows the model to learn important realistic characteristics of the domain. However, the use of synthetic data entails the so-called ''reality gap'', which is the inability to fully reproduce real-world data for numerous reasons, including textures, lighting, and complex domain specifics. All appearances generated by realistic simulations can only cover a user-defined scale of conditions, e.g. daylighting, programmed object position and interactions. Thus,

these generated environments represent only a subset of all the conditions that may occur in reality. Achieving higher photo-realism with high fidelity rendering engines comes at the cost of computational resources and rendering time.

In an attempt to mitigate the ''reality gap'' the opposite approach can be used, in which domain randomization is introduced to simulate a sufficiently large number of variations of all relevant domain features. For camera images, this may include randomization of viewing angles, camera and shader effects, lighting, material textures, color, shape, scale, and relative position of objects, while maintaining ''sensible'' invariants and constraints, forcing the model to learn the most important characteristics describing the objects being sought. This can further simplify the setup and speed up the simulation process, as requirements to simulation accuracy, model quality, and rendering accuracy are reduced, which saves computational resources.

C. HANDS DATASETS

Since datasets represent a crucial factor for successful implementation of hand tracking, a large variety of datasets have been published recently. The EgoHands dataset [24] contains 15K manually annotated RGB images of two people's first-person interactions. The annotations include semantic pixel-level segmentation masks for each hand. A. Bojja *et al.* in their work [25] presented an automatically labeled depth image dataset (HandSeg dataset) containing 150K recordings with random hand gestures in front of a depth camera. The data acquisition method was based on color gloves, which were used to create ground truth annotations using HSV color thresholding.

As an alternative to real camera images, many research groups used fully synthetic images rendered in customized visualizers. The ObMan dataset [26] is an example of such a fully synthetic approach, where RGB-D images were created using realistic 3D models of the human body with one hand holding commonplace objects. During the creation of the dataset, the camera was randomly pointed at the hand holding the object. The generated images varied greatly in pose, background, texture, and lighting, and in total the dataset contains 150k fully annotated depth and color images, with hand keypoints, object and hand segmentation masks.

A similar dataset was presented in Zimmermann *et al.* (RHD, Rendered Hand Pose dataset [27]) where an extensive RGB-D dataset was generated using 3D character models matched with highly parameterizable hand 3D model MANO [28], allowing segmentation masks to be collected for each segment of each finger. In each frame, the selected character model was posed in a random keyframe of any of 39 animated actions, and a new camera location was randomly selected from a spherical vicinity around the selected character hand. However, this limits the possible range in which the hand can appear in the image, so that it only appears in the centre of the image and never at the edges. The generated scenes additionally had randomized backgrounds, global lighting, specular reflections, and directional light sources. An additional policy ensured that the camera was rotated so that the hand was at least partially visible from the selected viewpoint, and the random background image did not contain a person.

Mueller *et al.* [29] presented pose and shape reconstruction of interacting hands, with model trained on synthetic dataset containing depth images samples complemented with RGB-encoded segmentation masks, where the color represented correspondence to vertices of a MANO hand model [28]. The model was additionally trained on a real camera data to help model to generalize, since the generated dataset did not contain any augmentation nor background obstacles.

An extensive review of available depth-based hand datasets is available in [30].

However, each of the discussed publicly-available datasets posses one or several following disadvantages:

- based solely on RGB information;
- assumption that the hand covers the majority of the image area;
- assumption that the hand is the closest object to the camera;
- absence of obstructions around the hand.

In addition, the available datasets assume a different placement of the camera than in our specific industrial storage. These factors served as a motivation for creating own customized dataset generator and subsequently training of the network.

We focused on the depth image dataset because depth capture is less sensitive to light, color, and texture, but rather focuses on shape.

III. METHODOLOGY

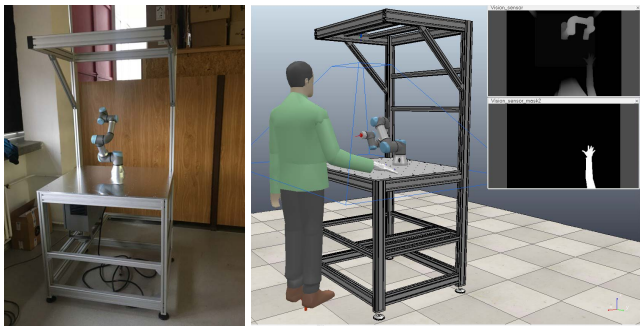
In order to localize the hand in the scene we propose a method based on a convolutional neural network trained on the synthetic dataset. The dataset is generated in the simulated environment which is set according to the testing scenario on real workspace. We compare the effects of different augmentations and simulated scene settings on the resulting accuracy of the trained neural network by evaluating the quality of the segmentation on a testing dataset which comprises of images obtained from a real sensor [31].

A. SYSTEM SETUP

Parameters and general appearance of the simulated scene were set with respect to the presumed use in the industrial application and in close correspondence to our experimental workplace (Figure 1). In the workplace, a single depth camera heading downwards is mounted 1 meter above the work table. The initial experiments were carried out using setup with a camera placed 1 m above the floor when no robot was involved. We utilized RGB-D Intel RealSense D435 camera as a sensor for capturing the depth images.

Specifications of the Intel RealSense D435 are:

- **Outer dimensions:** 90 × 25×25 mm.



(a) (b)

FIGURE 1. Experimental workplace with the robot (a) and simulation (b).

- **Field of view:** over 90° of depth diagonal field of view, from 0.2 m to over 10 m range depending on lighting conditions
- **Streams:** Up to 1280 × 720 active stereo depth resolution, up to 1920 × 1080 RGB resolution

In our experiment, we use a depth stream of 640 × 480, which is processed using a modified Intel RealSense librealsense library. In the pre-processing stage, we use a colorization filter that scales the depth information to an 8-bit value with a minimum range of 0.2 m, which corresponds to the minimum distance for proper camera operation, and a maximum distance of 1 m, which is the distance between the camera and the table. A customized hole-filling filter is used to remove shadows from the depth image caused by the stereoscopic camera technology. To fill in depth information where it is missing, we use a static, unobstructed image of the scene captured at the workstation. Image pre-processing is completed by scaling to a resolution of 320×240, which corresponds to the image size in the generated dataset.

B. DATASET GENERATION

For our application in an industrial environment with a specific camera orientation, we needed to create a custom dataset with features appropriate to the intended working environment. To be able to create large datasets for a specific environment, we developed a simulation-based dataset generator that creates an image of the dataset as well as a ground truth (labelled image). The simulation is implemented in the CoppeliaSim simulation platform (Figure 2).

In the simulation environment, the real camera is simulated by a vision sensor which is set according to the field of view of the real camera. During generation of the dataset, hand models are dragged through the field of view of the camera (Figure 3) - this allows to generate images with the most of the possible positions of the hand in the monitored space. Vision sensor captures depth image of the scene, where pixel values are scaled to 8-bit range and stored as a gray-scale image. In addition to the hand object, we can place to the scene different objects (geometric primitives) or noise. By our definition, random background objects can be both closer and further away from the camera than the hand, because in

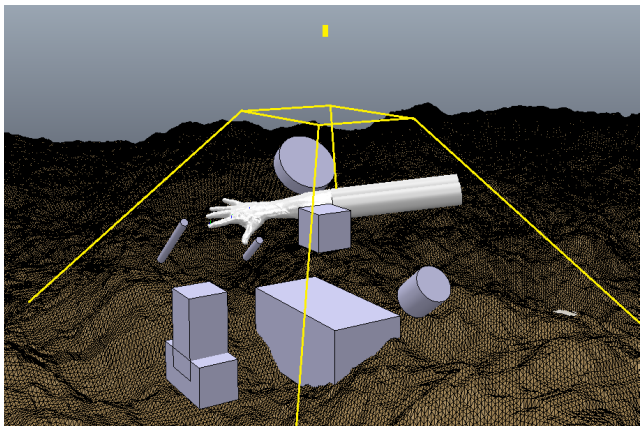


FIGURE 2. Simple dataset generator in CoppeliaSim.

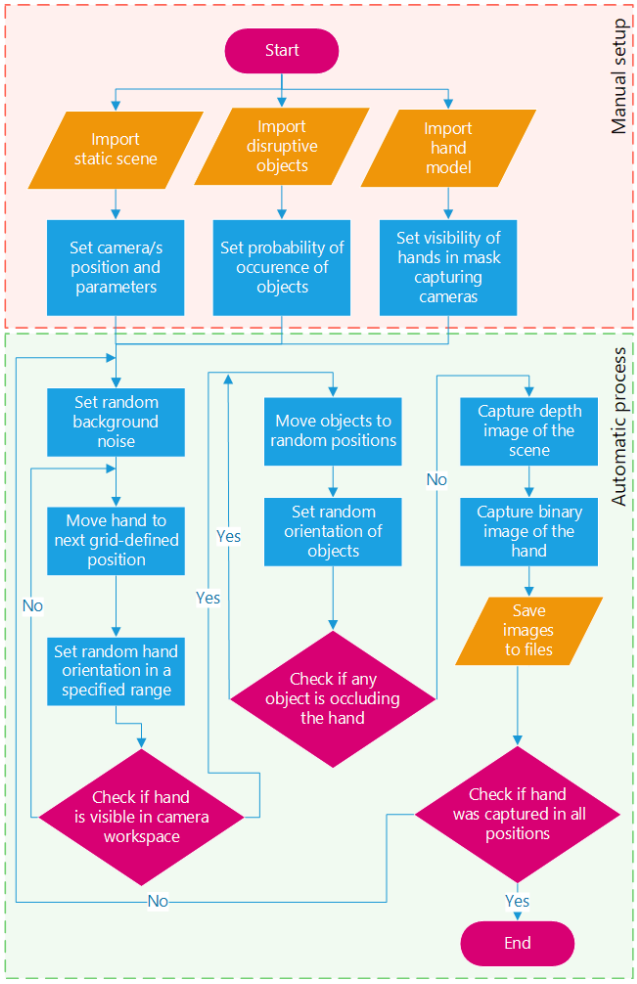


FIGURE 3. Flowchart of dataset generation.

industrial conditions it is not possible to ensure that the hand is always the closest object to the camera. Positions and orientations of these objects are random, but they are governed by the policy, which ensures that the created object never overlaps the hand (considering fingertip and palm centre point) in the camera view. If an overlay is found, the object is

TABLE 1. Dataset conditions.

Dataset	Background	Objects	Post-processing
A	-	-	-
B	Low frequency Perlin noise	-	-
C	Low frequency Perlin noise	Geometric primitives	-
D	Low frequency Perlin noise	Geometric primitives	Blur with kernel size 7 and transparent high frequency Perlin noise

moved to a different position until the requirement is satisfied. By adding random objects at random positions to the scene we attempt to decrease the sensitivity of the system to irrelevant items, which otherwise may be incorrectly detected as fingers.

This experiment was conducted on four datasets with varying level of complexity. Figure 4 shows example images taken from the generated datasets. Datasets differ by added random background noise, blur and obstacles in the scene, see Table 1. The post-processing applied to the images was adjusted according to the real camera image: the added Perlin noise was simulating the actual noise from the real sensor. The images with random obstacles were additionally blurred with a blur filter with kernel size 7. The hand model was iteratively shifted with an increment in each direction to cover the entire field of view of the sensor. The size of the position increment relates to the size of the final dataset: we use the increment of 5 mm which gives us 2346 images, 2 mm for 34166 images and 1 mm for 270106 images. Orientation of the hand is semi-random and defined by the rules which ensure that the hand and fingers are within the field of view of the vision sensor: the fingertip point and the centre point of the palm are ensured to be within the truncated pyramid corresponding to the field of view of the camera. Roll and pitch of the palm are constrained to $\pm 15°$ and $\pm 30°$ respectively. The simulation scene utilizes two 3D meshes of the right hand for dataset generation: an open hand gesture and a gesture - pointing with index finger. In our case we considered it sufficient, because the rules of scene simulation ensured that each hand gesture could be observed from multiple perspectives. Furthermore, our goal was only to perform segmentation the hand pixels from the input image, and we did not aim to classify the gestures (which otherwise would have required us to provide sufficient samples for each possible variation of each gesture in question).

The simulation also includes the second vision sensor, which is set to only capture pixels pertaining to the hand and the arm. This second sensor is placed at the same location as the first sensor and its output image is binarized (see Figure 5) and is considered as ground truth for the image segmentation task. The pixel values of the binary image are represented by 2 classes: hand-related pixels (1) and background-related pixels (0).

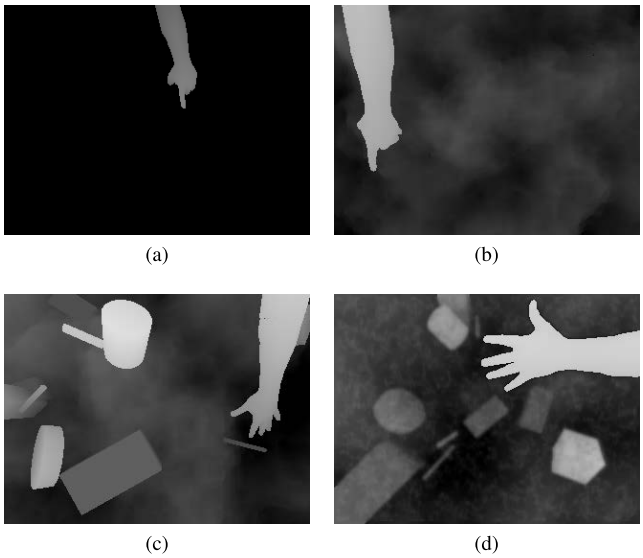


FIGURE 4. Dataset images generated with different setup: (a) Dataset A - only the hand is captured, (b) Dataset B - the hand with background noise, (c) Dataset C - obstacles with random position incorporated to the image, (d) Dataset D - apart from obstacles, blur and additional noise is added during post-processing phase.

FIGURE 5. Dataset image (a) and corresponding binary ground truth (b).

C. NEURAL NETWORK TRAINING

The neural network used in the experiment is implemented using TensorFlow. The architecture is based on U-Net [32] which is a fast convolutional network for accurate image segmentation. It has a contracting part consisting of convolutional layers and max pooling operations (see Figure 6). This part is responsible for capturing the context. The symmetrical expanding part of the network provides precise localisation. Our network contracting part consists of 5 convolutional layers with increasing number of filters which are multiples of 16. Convolutional layers are followed by pooling layers with the size 2×2. Expanding side is a set of upsampling deconvolution blocks.

We trained and validated models using the generated datasets (A, B, C, D, CD) of different sizes (2346, 34166, 270106). The fifth dataset (CD) was generated by randomly combining the C and D datasets.

All models were trained for 8 epochs with an initial learning rate of 0.001 with a total of 32 images per mini-batch. 20% of each corresponding dataset was used for validation. For better sensitivity for both hand sides, we used horizontal

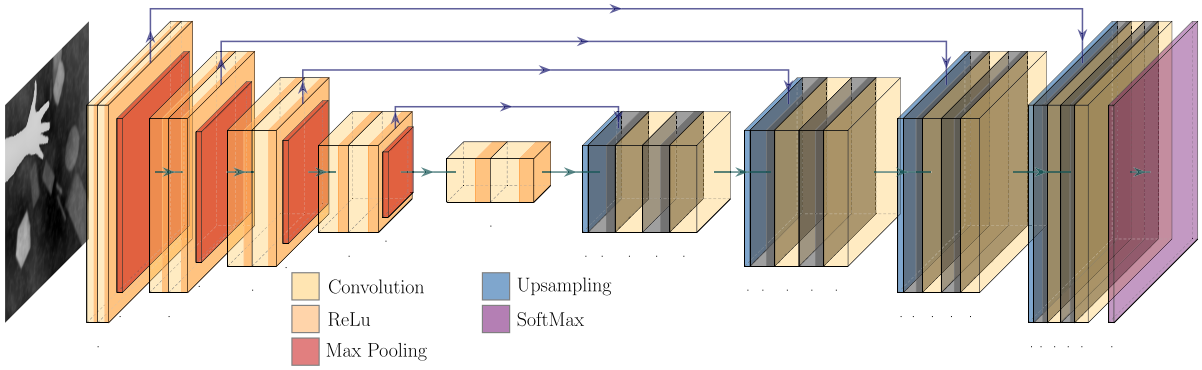


FIGURE 6. U-Net architecture [32] (Created using [33]).

and vertical flip image augmentations during the training stage because the datasets were generated only with the model of the right hand.

Soft dice loss was used as the loss function. This loss function is often used in segmentation tasks to assess the similarity between two samples. For binary case, the loss is based on the ratio of the number of correctly predicted pixels to the total number of pixel of both prediction and the ground truth and calculated based on the equation 1.

$$Loss = 1 - \frac{2 \sum\limits_{pixels} y_{true}\, y_{pred}}{\sum\limits_{pixels} y_{true}^2 + \sum\limits_{pixels} y_{pred}^2} \tag{1}$$

where y_{true} is the ground truth and y_{pred} represents the network output - prediction.

The training was performed on the laptop with 16 GB of RAM, Intel Core i7-6700HQ CPU and with NVIDIA GeForce GTX 1070 (8GB VRAM) graphic card.

Prediction of a single image takes around 1 ms, which we consider a sufficient speed for processing of a video stream of the camera.

Figures 7 and 8 depict the training and validation processes for each 34k dataset. From both training and validation plots, it may seem that the simplest dataset A was able to achieve the best accuracy (in terms of mIoU metric). However, in reality this only means that this dataset is the simplest for the adaptation of the network, because the simpler the dataset, the higher the accuracy the network will achieve not only on the training dataset but also on the validation dataset (because, as mentioned above, the training and validation datasets are composed of subsets of the original dataset generated in the simulation with the same conditions). At the same time the opposite result will be evident for the most demanding datasets, which require the network to learn more important features of the images. Only evaluation of the trained models on the test dataset (which will be the same for all evaluated models and will include the samples corresponding to the real application) can serve as a real accuracy evaluation. This evaluation will be done in the next section.

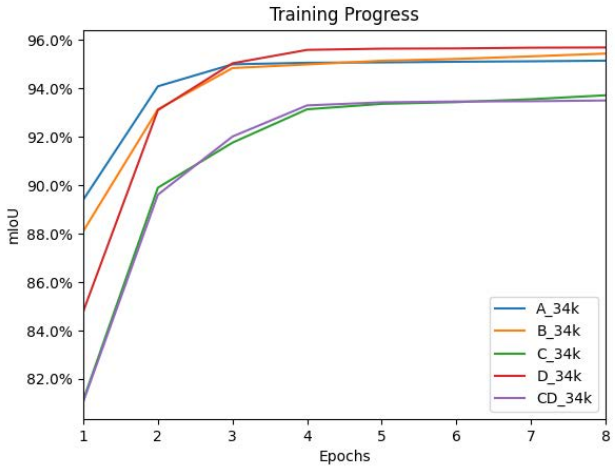


FIGURE 7. Mean intersection over union of different models during training process performed using training dataset.

IV. EVALUATION

In order to compare the impact of datasets, we propose an evaluation dataset captured on a real sensor as a benchmark and use it to measure the quality of the predictions generated by the trained neural network models. This benchmark has a quantitative evaluation using the metric of mean intersection over union (IoU), which is represented by the ratio between the overlap area and union area of the predicted and baseline regions. In addition, we perform a qualitative evaluation of the obtained results, where we examine the predictions and explain the reason for the quantitative result.

Table 2 shows precision and recall values of the benchmark dataset containing 100 manually-labelled camera images. Rows of the table represent the models trained on different datasets, while columns represent different sizes of the dataset. For the benchmark test, we used a cluttered environment with various obstacles that had several characteristics, such as sharp edges, rounded shape, glossy material, transparent material, multiple oblong objects resembling arm and finger. The reference scene was illuminated by both indirect

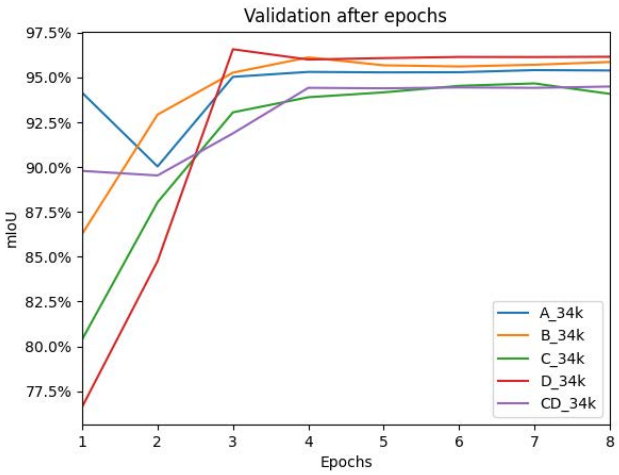


FIGURE 8. Mean intersection over union (mIoU) of different models after validation performed on validation dataset.

TABLE 2. Mean IoU for different models and number of samples.

	IoU			Precision			Recall		
#	2.3k	34k	270k	2.3k	34k	270k	2.3k	34k	270k
A	48%	9%	-	19%	7%	-	99%	100%	-
B	69%	80%	82%	46%	63%	64%	86%	95%	97%
C	68%	88%	90%	78%	89%	96%	55%	88%	89%
D	71%	73%	74%	79%	52%	50%	66%	93%	95%
CD	61%	87%	89%	42%	86%	89%	69%	89%	90%

and direct sunlight, and can generally be considered more complex than we would expect for industrial use.

The data in Table 2 show that predictions based on the model A do not provide sufficient accuracy and the result does not significantly improve with an increasing amount of samples. The best results are obtained with models based on dataset C and the combined dataset C+D. With the increasing dataset size the accuracy of predictions increases. Additional test with the unified datasets C and D (dataset CD) with the total number of 540212 samples did not further improve the accuracy.

The fast convergence observed in Figure 7,8 could have been a result of over-fitting and, indeed, the results of test subset evaluation (see Table 2) showed that the simplest datasets (A, B) did have difficulty generalizing, which apparently caused their over-fitting; however, more complex datasets allowed the networks to successfully generalize and over-fitting did not occur. We also assume that the observed fast convergence is partially due to the fact that the utilized U-Net architecture is rather simple, the inputs are small, and the training domain contains only one target class.

The second evaluation is qualitative, in which we review the predicted images and inspect the meaning of quantitative result along with the influences caused by the differences in datasets. Figure 9 shows the scene as captured by the RGB camera, along with the corresponding depth image and

TABLE 3. Experimental comparison with other datasets.

Dataset	Size	Objects	Labelling method	mIoU	mAP	mAR
Ours (CD)	270k	Yes	Synthetic	89%	89%	90%
RHD [27]	44k	No	Synthetic	68%	65%	59%
HandSeg [15]	150k	No	Markers (gloves)	59%	54%	32%
ObMan [26]	148k	Yes	Synthetic	47%	72%	44%
DenseHands [29]	85k	No	Synthetic	57%	9%	92%

prediction. The figure does not cover the dataset A because it did not provide satisfactory results. All images correspond to predictions of the models trained on 270106 samples. It can be observed that the predictions of model B contain false positives associated with the obstacles. Dataset B did not include obstacles and apart from the hand the network also incorrectly marks large objects in the scene. The results of dataset C show the best agreement with the manually labelled result. However, the predictions are less sensitive to the details such as fingers of the open hand when the hand is more distant from the camera. Results of the predictions generated by model D are the most detail-sensitive, this results in false positive errors which include regions pertaining to the objects adjacent to the hand. Predictions of the CD model show the similar results as the C model, but are more sensitive to details. Generally it can be considered as an advantage, however, it is worth noting that the forth sample contains a large false positive region.

Comparing the results from the Table 2 and Figure 9 it can be observed that when the dataset without obstacles, the number of false positive predictions is high. Random background objects added to the dataset increase the accuracy of the predictions. Additional noise may increase the sensitivity to the shape details, nonetheless it has to be balanced to avoid type I errors.

To compare our dataset with existing work in this area, we adapted several well-known publicly available RGB-D and depth-based hand datasets (see Table 3); their descriptions were presented in section II.C. The following modifications were applied to adapt the labeled inputs of the datasets:

- Single class masks: (HandSeg - merging both hands' masks; DenseHands - binarized dense correspondence was used as hand masks; RHD - all masks except hands were filtered, ObMan - all masks except hands were filtered).
- The 0-1 m depth range was mapped to the 0-255 byte range according to the settings in the test environment. The remaining range was truncated to the 1 m boundary.

The dataset adaptation code is available in the GitHub repository [31]. Our dataset input pipeline automatically adapted all images to 320 × 240 resolution. We then trained the U-Net model using adapted datasets with an 80% / 20% training-validation split and the equivalent training settings. The trained models were evaluated on a set of real camera data representing the expected environment. Because DenseHands, RHD, and HandSeg contained only hand masks, the

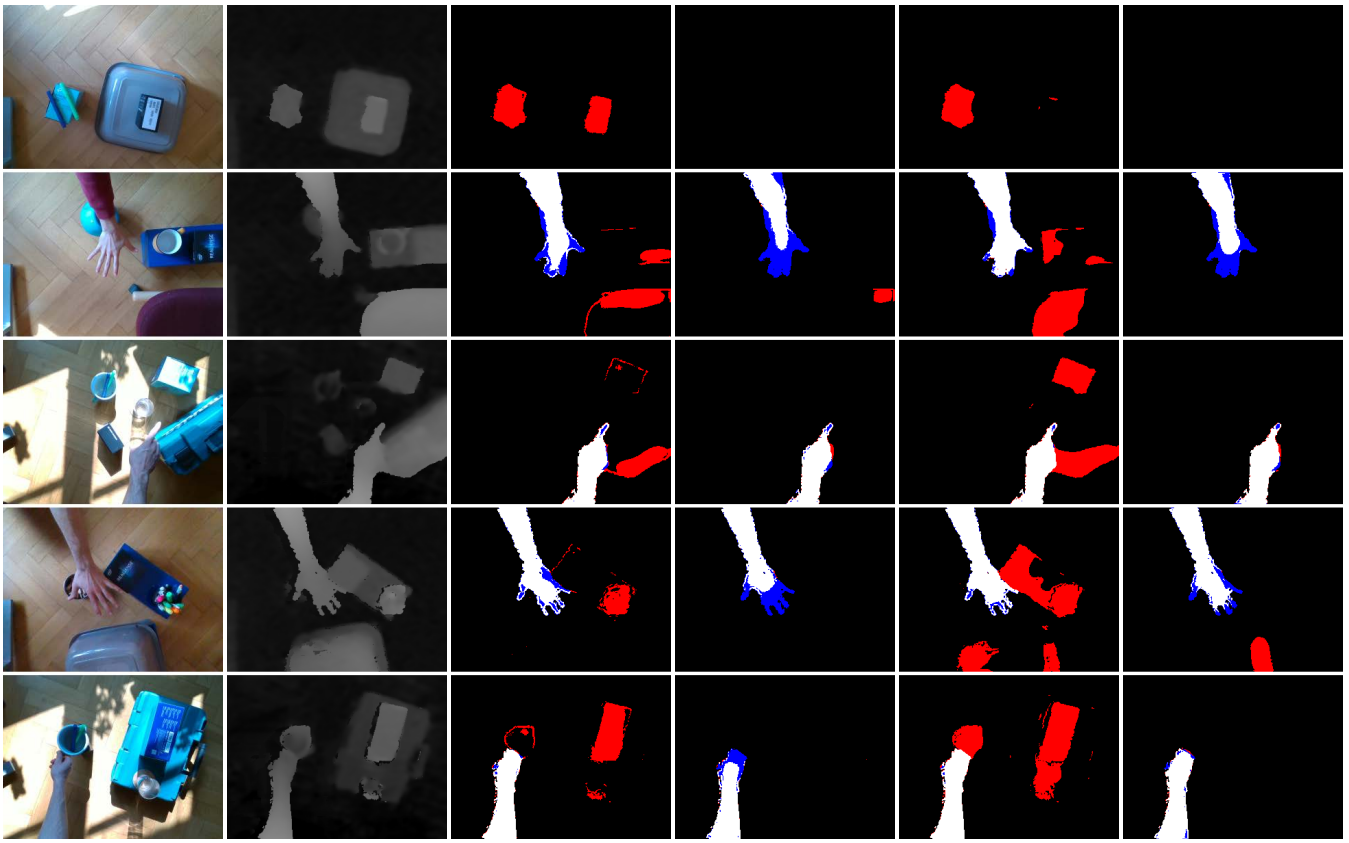


FIGURE 9. Prediction of five sample images using different models: (a) RGB image of the scene, (b) corresponding depth image, (c) prediction of the model B, (d) prediction of the model C, (e) prediction of the model D, (f) prediction of the model CD. White pixels represent the regions where the prediction matches the ground truth (true positives), red pixels represent false positive errors, blue pixels represent false negative errors.

test dataset was prepared with two sets of masks (entire arms and hands only). Although the compared datasets contained different numbers of samples, we assume that with the same number of samples, no large difference would be observed because Table 2 shows that the difference in performance between 34k and 270k is not significant.

As already mentioned, most work on related topics relies on some or all of the following assumptions: the hands are the closest object to the camera, the hand is in the center of the image, and there are no other objects in the camera image besides the hands. The results presented in Table 3 are partially due to these assumptions and the fact that the environment for use varies. For our task these assumptions cannot be guaranteed, therefore when creating our dataset we tried to avoid these shortcomings by making the modelled scene contain random obstacles that force the network to learn the important features corresponding to the hands. In addition, the applied post-processing, which incorporates blur, ensures a higher similarity of the generated images with the real ones. These conditions were necessary for our intended environment of use (industrial workspace), in which obstacles of undefined shapes can be found in the workspace.

The results in the Table 3 correspond to the input data for the trained network. The DenseHands dataset has high similarity to our dataset A, where no objects and noise are present in the scene and the training process tends to overfit. The images of this dataset feature low variability in hand position and orientation. A slight improvement in results can be observed in the Obman dataset, which includes several objects in the surroundings. Yet the position of the hand is mostly in the middle of the image and at approximately the same depth. Better results are shown by the Handseg dataset, which is not synthetic and has a natural representation of the images acquired by the camera. However, the low variability of the dataset features causes the trained model to perform significantly worse than our presented dataset under specified environmental conditions. The high variability of the images in the RHD dataset makes the results better, but the absence of noise that could make the synthetic dataset look similar to the actual camera images limits the quality of the predictions compared to a network trained on our dataset.

V. DISCUSSION

The initial experiment with the camera mounted above the ground with common items serving as obstacles was extended

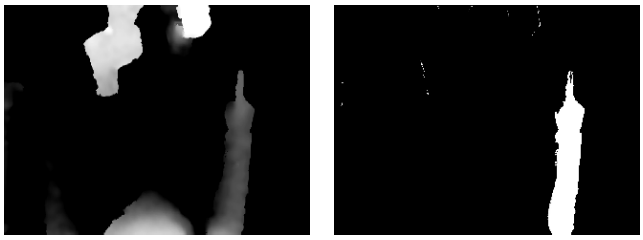


FIGURE 10. Real workplace prediction: (a) Depth camera image, (b) prediction.

to a real-world scenario. We used existing workplace with industrial collaborative robot UR3e to test the trained models. Figure 1a represents the real workplace with the robot, Figure 1b depicts the corresponding simulation model in CoppeliaSim. The improved simulation model utilized a mannequin with a modified hand mesh to capture the workplace images making the image more realistic. The obstacle was represented by the robot which was moved to random positions. An example of the generated dataset image and the prediction is shown in the Figure 1b (the dataset image is in the upper right corner and the ground truth image is located below). A new dataset consisting of 270k images was generated in the above described scene with the same parameters as in the initial experiment above. Figure 10 depicts the depth image from the real camera and the corresponding output of the neural network, where the hand pixels are found and the robot is correctly filtered out.

The proposed system of hand localisation represents the first stage for human-robot cooperation using gestures. The second stage is gesture recognition and localization of the hand key points (such as fingertips). This could be done with another neural network, which would utilize the determined hand location as an input. For the task of hand key points localization we applied an open-source RGB-based solution OpenPose which requires specification of a square area of the image, where the hand is present (region of the interest). The specification of the region of the interest was provided by our system - the result is illustrated in Figure 11. The hand region localized by our model is marked by the blue square and the extension of this region (marked by white square) was used as an input for OpenPose. OpenPose uses only the colour channel to localize the key points of the hand. The implemented system works as an alternative to the default OpenPose hand localization, which requires that at least the torso to be within the camera field of view in order to correctly detect the hand key points.

Generating a representative dataset and avoiding "reality gap" in a simulation often requires highly detailed modeling of the target environment, which is time-consuming and requires extensive manual tuning of the simulated scene, since the generated scenes must represent a wide range of circumstances that may occur in reality. For this reason, we opted for the Doman Randomization technique because it does not require the simulated scene to be an exact representation of the real workspace, and can provide a wide

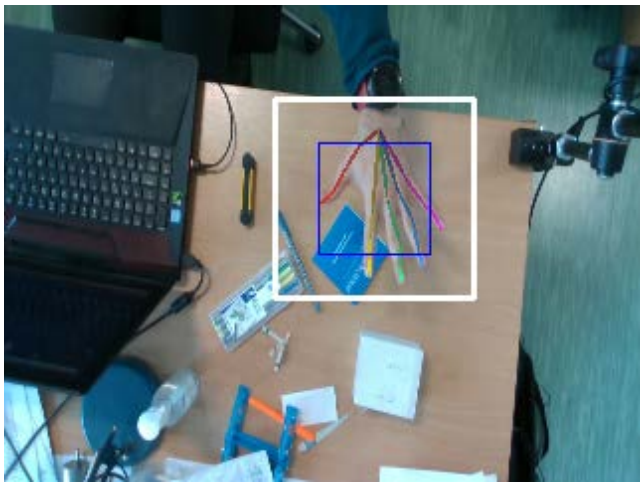


FIGURE 11. Hand key points localization using OpenPose network.

range of conditions that allows the model to generalize. In terms of labor required to prepare and acquire the dataset, the synthetic dataset in our case remains an advantageous option, since the simulation can easily be extended and adapted to any specific workspace. For synthetic datasets, the most time-consuming operation is the preparation of the simulation, the collision rules for the obstacles and selecting augmentations, which, however, need only be set once; after that the process of data set creation is simple and generating an arbitrary number of images takes little time compared to manual arrangement and labelling of images.

The conventional approach of collecting a dataset from images from real cameras requires manual compositing of the workspace to create a sufficiently large and diverse dataset and subsequent manual labeling, which is much more labor-intensive. Repeatedly manually rearranging elements in the workspace to provide enough diversity in the dataset needed to generalize the trained network is tedious, time-consuming, and still cannot come close to the diversity of scenes created with Domain Randomization.

In terms of performance, a sufficiently complex scene with an arbitrary number of added obstacle objects (which represent objects present in the real environment) will not affect the performance of the simulation, since it uses neither complex rendering nor physical simulation. The simulation can be further optimized to achieve even better performance.

VI. CONCLUSION

In this paper, we focused on generating synthetic datasets for training depth image segmentation models for the hand localization task. The use of a domain randomization technique enabled the rapid generation of an arbitrarily large synthetic dataset that included a wide range of samples with features important for accurate hand localization. The evaluations performed on the trained models allowed us to analyze the effects of the complexity of the dataset and the additional post-processing augmentations on the resulting image segmentation accuracy. Moreover, these benchmarks allowed us to identify the version of the dataset with the highest

accuracy of over 90%. We provide new synthetic datasets for industrial environments suitable for various hand tracking applications, as well as ready-to-use pre-trained models and simulation scenes that can be used to create custom datasets.

In the future, we plan to extend the dataset generator to enable a simpler and more user-friendly solution for adapting the simulation to the requirements of the real workspace. The use of a specialized parameterized hand model allows the generation of an arbitrary number of gestures, which is necessary to further classify the gestures. We also plan to investigate the effects of image augmentations applied to the RGB-D synthetic datasets to improve the generalization capabilities of the trained models.

REFERENCES

[1] S. Grushko, A. Vysocký, D. Heczko, and Z. Bobovský, "Intuitive spatial tactile feedback for better awareness about robot trajectory during human–robot collaboration," *Sensors*, vol. 21, no. 17, p. 5748, Aug. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/17/5748

[2] S. Grushko, A. Vysockỳ, P. Oščádal, M. Vocetka, P. Novák, and Z. Bobovskỳ, "Improved mutual understanding for human-robot collaboration: Combining human-aware motion planning with haptic feedback devices for communicating planned trajectory," *Sensors*, vol. 21, no. 11, p. 3673, 2021.

[3] I. El Makrini, G. Mathijssen, S. Verhaegen, T. Verstraten, and B. Vanderborght, "A virtual element-based postural optimization method for improved ergonomics during human-robot collaboration," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1772–1783, Jul. 2022.

[4] X. Zhang, X. Bai, S. Zhang, W. He, P. Wang, Z. Wang, Y. Yan, and Q. Yu, "Real-time 3D video-based MR remote collaboration using gesture cues and virtual replicas," *Int. J. Adv. Manuf. Technol.*, vol. 121, no. 11, pp. 7697–7719, 2022.

[5] B. Gajšek, S. Stradovnik, and A. Hace, "Sustainable move towards flexible, robotic, human-involving workplace," *Sustainability*, vol. 12, no. 16, p. 6590, 2020.

[6] O. Mazhar, B. Navarro, S. Ramdani, R. Passama, and A. Cherubini, "A real-time human-robot interaction framework with robust background invariant hand gesture detection," *Robot. Comput.-Integr. Manuf.*, vol. 60, pp. 34–48, Dec. 2019.

[7] R. M. Gurav and P. K. Kadbe, "Real time finger tracking and contour detection for gesture recognition using OpenCV," in *Proc. Int. Conf. Ind. Instrum. Control (ICIC)*, May 2015, pp. 974–977.

[8] C.-C. Hsieh and D.-H. Liou, "Novel Haar features for real-time hand gesture recognition using SVM," *J. Real-Time Image Process.*, vol. 10, pp. 357–370, Jun. 2012.

[9] L. Fang, G. Wu, W. Kang, Q. Wu, Z. Wang, and D. D. F. Feng, "Feature covariance matrix-based dynamic hand gesture recognition," *Neural Comput. Appl.*, vol. 31, pp. 1–14, Dec. 2019.

[10] M. Hu, F. Shen, and J. Zhao, "Hidden Markov models based dynamic hand gesture recognition with incremental learning method," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2014, pp. 3108–3115.

[11] D.-S. Tran, N.-H. Ho, H.-J. Yang, E.-T. Baek, S.-H. Kim, and G. Lee, "Real-time hand gesture spotting and recognition using RGB-D camera and 3D convolutional neural network," *Appl. Sci.*, vol. 10, no. 2, p. 722, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/2/722

[12] Z. Bobovský, V. Krys, and V. Mostýn, "Kinect v2 infrared images correction," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 1, 2018, Art. no. 1729881418755780, doi: 10.1177/1729881418755780.

[13] A. Vysocký, S. Grushko, P. Oscádal, T. Kot, J. Babjak, R. János, M. Sukop, and Z. Bobovský, "Analysis of precision and stability of hand tracking with leap motion sensor," *Sensors*, vol. 20, no. 15, pp. 1–14, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4088

[14] T. Coogan, G. Awad, J. Han, and A. Sutherland, "Real time hand gesture recognition including hand segmentation and tracking," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, A. Nefian, G. Meenakshisundaram, V. Pascucci, J. Zara, J. Molineros, H. Theisel, and T. Malzbender, Eds. Berlin, Germany: Springer, 2006, pp. 495–504.

[15] A. Bojja, F. Mueller, S. Malireddi, M. Oberweger, V. Lepetit, C. Theobalt, K. Yi, and A. Tagliasacchi, "HandSeg: An automatically labeled dataset for hand segmentation from depth images," in *Proc. 16th Conf. Comput. Robot Vis. (CRV)*, May 2019, pp. 151–158.

[16] A. Memo, L. Minto, and P. Zanuttigh, "Exploiting silhouette descriptors and synthetic data for hand gesture recognition," in *Proc. STAG*, 2015, pp. 15–23.

[17] A. Wetzler, R. Slossberg, and R. Kimmel, "Rule of thumb: Deep derotation for improved fingertip detection," in *Proc. BMVC*, Jul. 2015, pp. 1–12.

[18] G. Hillebrand, M. Bauer, K. Achatz, and G. Klinker, "Inverse kinematic infrared optical finger tracking," in *Proc. 9th Int. Conf. Hum. Comput. (HC)*, Jan. 2006, pp. 6–9.

[19] Y. Toda, F. Okura, J. Ito, S. Okada, T. Kinoshita, H. Tsuji, and D. Saisho, "Training instance segmentation neural network with synthetic datasets for crop seed phenotyping," *Commun. Biol.*, vol. 3, p. 173, Apr. 2020.

[20] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," 2018, *arXiv:1804.06516*.

[21] A. Vysocký, S. Grushko, R. Pastor, and P. Novák, "Simulation environment for neural network dataset generation," in *Proc. Int. Conf. Modeling Simulation Auto. Syst.* Cham, Switzerland: Springer, 2022, pp. 322–332.

[22] J. Malik, A. Elhayek, F. Nunnari, K. Varanasi, K. Tamaddon, A. Heloir, and D. Stricker, "Deephps: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth," in *Proc. Int. Conf. 3D Vis. (DV)*, 2018, pp. 110–119.

[23] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 1228–1234.

[24] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1949–1957.

[25] A. K. Bojja, F. Mueller, S. R. Malireddi, M. Oberweger, V. Lepetit, C. Theobalt, K. Moo Yi, and A. Tagliasacchi, "HandSeg: An automatically labeled dataset for hand segmentation from depth images," in *Proc. 16th Conf. Comput. Robot Vis. (CRV)*, May 2019, pp. 151–158.

[26] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11807–11816.

[27] C. Zimmermann and T. Brox, "Learning to estimate 3D hand pose from single RGB images," 2017, *arXiv:1705.01389*.

[28] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–17, Nov. 2017.

[29] F. Mueller, M. Davis, F. Bernard, O. Sotnychenko, M. Verschoor, M. A. Otaduy, D. Casas, and C. Theobalt, "Real-time pose and shape reconstruction of two interacting hands with a single depth camera," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–13, 2019.

[30] A. Lopes, R. Souza, and H. Pedrini, "A survey on RGB-D datasets," *Comput. Vis. Image Understand.*, vol. 222, Sep. 2022, Art. no. 103489, doi: 10.1016/j.cviu.2022.103489.

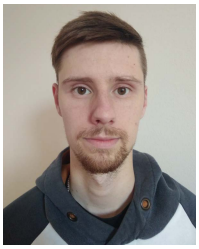
[31] A. Vysocky. *Simulation Dataset Generator*. Accessed: Sep. 6, 2022. [Online]. Available: https://github.com/AlesVysocky/HGR_CNN

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," 2015, *arXiv:1505.04597*.

[33] H. Iqbal. (2022). *Plotneuralnet*. [Online]. Available: https://github.com/HarisIqbal88/PlotNeuralNet

ALES VYSOCKY received the master's and Ph.D. degrees in mechanical engineering with specialization on robotics from the VSB—Technical University of Ostrava. His research interests include human–robot cooperation and interaction in industrial applications focused on safety and conditions for direct cooperation and workspace sharing with the robot.

STEFAN GRUSHKO received the Ph.D. degree in robotics from the VSB—Technical University of Ostrava, Czech Republic. He is currently a Research Associate and a Lecturer with the VSB—Technical University of Ostrava. His Ph.D. research focused on improving mutual awareness during human–robot interaction. His recent research and teaching interests include human–robot collaboration, teleoperation, and upper-limb prosthetic devices.



TOMAS SPURNY received the master's degree in robotics from the Technical University of Ostrava, Czech Republic, in 2021, where he is currently pursuing the Ph.D. degree. His current research interest includes movement prediction of human workers sharing workspace with collaborative robots.

ROBERT PASTOR received the master's and Ph.D. degrees from the VSB—Technical University of Ostrava. He is currently working as a Research Assistant with the VSB—Technical University of Ostrava. His research interests include human machine interfaces, teleoperation, mobile robotics, and evolutionary robotics.

TOMAS KOT received the M.Sc. and Ph.D. degrees in robotics and the Habilitation degree from the Technical University of Ostrava, Czech Republic, in 2004, 2011, and 2020, respectively. He is currently working as a Senior Researcher with the Technical University of Ostrava. His research interests include complex simulations and control of mechatronic systems, visualization, application of virtual and augmented reality in robotics, optimization of layouts of robotized workplaces, algorithms for automatic design of an optimal kinematic structure of a robotic manipulator suitable for a given task, and lately also collision avoidance for collaborative robots sharing workspace with human workers.