https://doi.org/10.2352/ISSN.2470-1173.2021.6.IRIACV-311

Vision-based Machine Learning Worker Assistance System for People with Disabilities on Low-Cost Hardware

Micha Christ; Christian Jauch; Julia Denecke; Saskia J. Wiedenroth; Department Image and Signal Processing, Fraunhofer IPA Stuttgart (Germany)

Abstract

Working in protected workshops places supervisor workers in a work field with concurrent targets. On the one side, the workers with disabilities require a safe space to meet special requirements and on the other side, customers expect comparable time and quality standards than in the normal industry while maintaining cost pressure. We propose a technical solution to support the supervisors with the quality control. We developed a flexible assistance system for people with disabilities working in protected workshops that is based on a Raspberry Pi4 and uses cameras for perception. It is appliable for packaging and picking processes and is supported by additional step by step guidance to reach as many protected workshops as possible. The system tries to support supervisors in quality control and provide information if any action is required to free time for interpersonal matters. An automatic pick-by-light system is included which uses hand recognition. To ensure good speed we used image processing and verified the detections with a machine learning approach for robustness against lighting conditions. In this paper we present the system, which is available open source, itself with its features and the development of the machine learning algorithm.

Introduction

Supervisors in protected workshops have different tasks. Their responsibilities include quality control, work distribution, work preparation and keeping track of customer requirements, such as the deadline of the order and the quantity of products needed for the order. Additionally, they have a protective role and are also responsible for interpersonal issues. Due to this tension, they are heavily burdened. A low cost assistance system can reduce worker stress by helping them with technical tasks such as quality control and work preparation, ultimately leaving them more time for interpersonal issues. Worker assistance systems in general support workers in performing their tasks by receiving and processing information from the environment and providing feedback. In order to guide workers and control processes, and thus reduce error rates and mental stress [3, 4], assistance systems must track the work steps being performed. Systems exist in varying degrees of complexity and differ both in the process of recognizing hands to track work steps and in the type of feedback they provide. Deep learning algorithms have become state of the art in the field of object recognition and can therefore also be applied to the task of hand recognition. However, when it comes to protected workplaces, special requirements must be met. Employees' tolerance thresholds are often low, leading to declining motivation and interpersonal tensions as soon as major changes occur in the workplace. This can be caused by the integration of an assistance system, as it usually involves a change in the workplace and requires compliance with certain conditions, such as wearing a glove or sensor. Moreover, in packaging processes, an order can only be completed if all related components are available in sufficient quantities. However, due to employees' limited cognitive abilities, supervisors cannot rely on employees to refill empty cartons on their own. The situation is exacerbated by time pressure and the quality demands of customers, who need their goods on time regardless of the special circumstances.

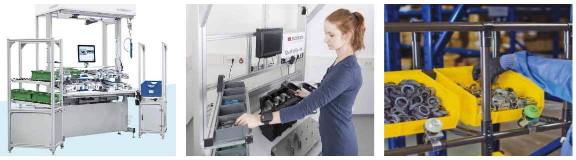
We propose a flexible worker assistance system for packing and picking processes that meets the needs of protected workshops by minimizing the associated changes in the environment known to the worker while providing robust support. We use an RGB camera, image processing tools and an object detector to detect and track the worker's hand to derive a removal of a box component. The worker can continue to work as usual, as there is no interaction or need to wear a glove. Integrated lights provide simple but sufficient feedback. To save time and avoid incomplete orders, the system provides a visual approach to detect box fill levels. This allows the supervisor to be informed of possible empty boxes at an early stage. Our system differs from existing ones in that we use only low-cost hardware. Our focus is to reach and support as many workshops as possible. Therefore, the system is limited to the bare essentials and is easy to replicate. It comes with step-by-step instructions and allows for user-friendly system configuration with a web application.

In this paper, we present our improved system version. We started with very fast detection of hands using edge detection and derived a handle by computing pixel differences of the box images. The increasing computational power of low-cost hardware such as the Raspberry Pi4, advances in lightweight object detectors such as the *Tiny-YOLOv4* [1], and the strong light sensitivity of the hand detection used, as all contours were identified as a possible hand regardless of the trigger, motivated the revision of the original system. In summary, the main contributions of our work are:

- An open-source worker assistance system for packaging processes that allows people with disabilities to continue working as usual, provides user-friendly configuration, detects empty boxes, and tracks the worker's hand with an object detector using Deep Learning, using only low-cost hardware.
- A hand data set that allows the recognition of hands from an egocentric view with and without gloves.

We provide insights into our system flow, introduce hardware and software components and illustrate the development of the train-

[1] https://github.com/AlexeyAB/darknet



(a) ActiveAssist[3] (b) QualityAssist[5] (c) LightGuide[6]

Figure 1: Examples of commercial assistance systems. Complexity decreases from (a) to (c).

ing data set. The performance of different models are compared and exemplary setups demonstrate the accuracy and robustness in operation. The project page [2] provides the software and build instructions.

Related Work

To evaluate our contribution in relation to existing packaging assistance systems and the extensive literature on hand detection algorithms, it is important to to consider several aspects of each approach: what is the level of complexity, how does the system affect the worker, what assumptions are made, and how applicable is the hand detection.

Fig. 1 shows three commercially available worker assistance systems for packaging processes, decreasing in complexity from (a) to (c). *ActiveAssist* [3] includes a vision-based pick-to-light system, hand tracking, in-situ projection, touch screens, and other components. Another complex vision-based system is Der schlaue Klaus[4], which uses object recognition to obtain information. In contrast, *QualityAssist*[5] relies on a wearable transmitter that emits ultrasonic waves. A system mounted at the workplace receives these waves and calculates the position of the worker's hand. A simplified solution is provided by *LightGuide*[6], in which the worker confirms the removal of a component by touching a sensor light. Other work focuses only on the different types of feedback and explores *in-situ* feedback through augmented reality solutions [1, 2]. There is also work that examines the effects of selected assistance systems. [3] evaluates picking assisted by different approaches, such as head-up displays or pick-by-light, in terms of error types, efficiency, and user preferences. The relationship between situation awareness, especially in pick-by-light systems, is investigated in [4]. Considering assistive technology in general for people with cognitive disabilities, a literature review is provided in [5].

However, to our knowledge, none of the existing work focuses on a system design that minimizes the associated changes to the workplace, targets the use of low-cost hardware, and is available as open source while leveraging the state of the art.

Object recognition algorithms have attracted a lot of attention in recent years because they can be used in many different subject areas. One application area is hand detection and localization, which is often extended by gesture recognition and thus can be used for sign language recognition [9] or home automation [10]. [6] points out that hand recognition started with data glove sensors for capturing hand movements and coordinates. A detailed overview of such glove-based systems and their applications is provided by [11]. The drawbacks of these systems, such as unsuitability for the elderly or possible skin irritation, led to the development of image processing-based techniques. Most popular techniques are based on skin color recognition in different color spaces [12] or use colored glove markers [13]. Although there are skin-based models [14] that encompass the entire spectrum of skin color, such color-based methods remain susceptible to skin-like colored objects and varying lighting conditions. Other approaches, summarized in [15], segment the hand from images using depth information from 3D sensors. However, such sensors are quite expensive and the hand should be the object closest to the camera for the algorithm to work properly. Motion-based [16] and appearance-based [17] hand or gesture recognition can be applied when the background is rather static and main movements in the image are caused by hands.

Object detectors based on deep learning aim to detect semantic objects of a certain class in images by applying a feature extraction step and inferring object location and class membership [7]. Various approaches for domain-specific object detectors have emerged over time and can be categorized into two-stage detectors, most representative are the R-CNN variants [18, 19, 20], which first identify the image regions of interest and then features extract and classify the features, and one-stage detectors, such as *YOLOv4* [8] and *SSD* [21], which locate and classify objects without a prior region proposal step. To improve detection speed on low-end devices, several lightweight architectures, such as the *MobileNet* series [22, 23, 24] and lightweight *YOLO* models [26, 29, 25, 27, 28, 30], have already been proposed.

Although even lightweight deep learning detectors require high computational power, they only make few assumptions about the object to be detected while showing promising results in terms of robustness. Therefore, we use *Tiny-YOLOv4* as a hand detector. While literature provides the algorithm, a data set for detecting both hands and hands with gloves is not found.

System Overview

Our proposed system, Fig. 2, consists of three modules that can be installed on one computing unit, but can also be distributed on multiple devices:

- **Sensor application**: Receives information from the workspace via a camera, derives the position of hands, grasps, and empty boxes, and controls visual (LED) and auditory (speaker) feedback.
- **Control app**: Controls the process by receiving messages from the sensor application such as: *"Box1, removal detected; Box2, empty"* and sending commands such as *"Box1, LED to green; Box2, LED to red"*. It displays process information in a web application and sends user-configured settings to the sensor application.
- **Web application**: Allows the supervisor to configure all settings of the current order, e.g. quantity of components, and the sensor application, e.g. box coordinates.

By using low-cost hardware, the system can be replicated with a budget of about 100$: *Raspberry Pi4 B 1gb* (∼ 40$), *camera module v2.1* (∼ 10$), *WS2801 LED-strip* (∼ 15$), passive

[2] https://ambos-3d.ipa.fraunhofer.de

[3] https://www.boschrexroth.com/de/de/produkte/produktgruppen/montagetechnik/news/assistenzsystem-activeassist/index

[4] https://www.optimum-gmbh.de/produkte/der-schlaue-klaus

[5] https://sarissa.de/qualityassist

[6] https://www.turck.de/static/media/downloads/more21651e.pdf

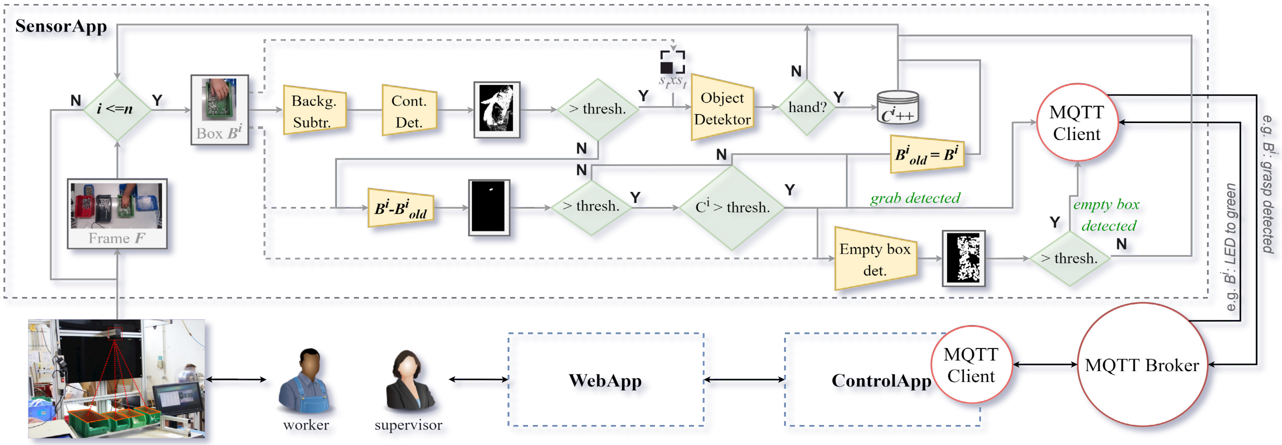


Figure 2: Schematic diagram of the system: Sensor application receives user configured settings from *MQTT* broker. Box images $B^{i,..n}$ are extracted from the current frame F and processed sequentially, by first subtracting static background from the box image and applying contour detection. The box image is stored as B^i_{old} and is only updated if there is no contour in the image. If a contour has been found, a counter C^i is increased if the object detector detects a hand in the box image scaled to size *mxm*. Optionally, the hand detection suspends if C^i is above a threshold. The absence of movement (no contour) leads to the calculation of a pixel-by-pixel difference $d^i = B^i - B^i_{old}$. A removal is derived if d^i and C^i are big enough. An empty box is derived either by counting edges or using saturation-histograms of the box images. *MQTT* messages of detected events are send to the broker and forwarded to the control app which answers with instructions to set the status of the led strip and speaker. The process of setting LEDs and speakers is neglected in this diagram.

cooler ($\sim$ 15\$) and small components like screws, wires, SD-card, and buttons ($\sim$ 20\$). The firmware includes the application which is implemented in *C++* running on Ubuntu 20.04. We use *OpenCV4* and its image processing tools to subtract backgrounds, find contours, blur images and detect edges. Instead of the *YOLO* framework *Darknet*[7], the object detector is instantiated with the *DNN* module of *OpenCV4*, as it provides higher prediction speed on CPUs. The *MQTT* protocol is used for a fast communication between the modules.

Data Set

Creating a data set for a robust hand detection model is a challenging problem due to different illumination settings, hand poses, occlusions, and backgrounds. In packaging processes, the hand is typically placed over a box with components of arbitrary color, shape, and texture, which can have the appearance of fingers or veins under certain conditions. Training a model capable of recognizing hands wearing any type of glove is also not straightforward, as we are not aware of an annotated glove data set based on real data. A large amount of training data would be required to build a robust model with respect to different colored objects and viewing angles. Since our proposed assistance system always places the camera over the worker, we aim to optimize the model only for hands from egocentric views and gloves that are available to us. Table 1 summarizes two different approaches.

In a first approach, we used the data set D^1 consisting only of self-captured images of simulated packaging processes. During acquisition, a pre-trained object detector trained on hand images from the *Open Images Dataset*[8] was used for pre-labeling. Both full resolution images (1280x720) and images cropped to box areas were saved. If images were too blurry or the hand was cropped too much, the data was removed. In total, approximately 5900 of images were captured using six boxes with different components, four gloves (red, green, blue, white), a work area, and the hands of two workers, Fig. 3 (a).

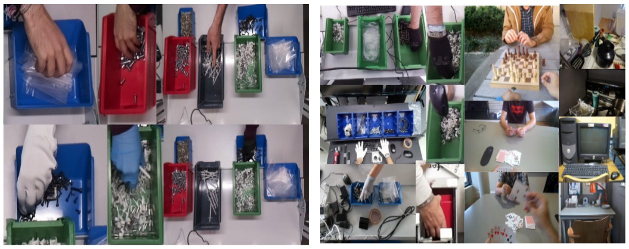

(a) D^1 (b) D^2

Figure 3: Example images: (a) contains self-recorded images with hands and (b) contains self-recorded images with and without hands and images from *EgoHands*, *CMU_KO8* and *Office-Home* data set

In a second approach, a data set D^2 is used. It is based on D^1 and extends it with new self-captured images at different workplaces and other gloves, as well as 2500 images from the *EgoHands* [31] data set. We focused on a balance between full-size images and those with box-cropping, as well as images containing hands with and without gloves. In total, D^2 contains 10,000 images of hands. For better robustness to objects without hands, we added 10,000 images that do not show hands and are from our self-captured images, *CMU_KO8* [32] and the *office-home* [33] data set, fig. 3 (b).

All self-recorded data went through a time-consuming manual labeling process. We removed images where the fingers were no longer visible due to motion blur or where the hand detail was too small. For images that were already labeled, we checked the

[7] https://pjreddie.com/darknet/
[8] https://storage.googleapis.com/openimages/web/index.html

Table 1: Generated data sets and their amount of images: the first five columns refer to self-recorded images. External data sets: Egohands [31],CMU_KO8 [32] and Office-Home [33]

Data set	hands in full res.	hands cropped	hands	hands, gloves	no hands	external data, hands	external data, no hand
D^1	4042	1860	4377	1525	-	-	-
D^2	3614	3886	2500	5000	8618	EgoHands (2500)	CMU_KO8 (1100), Office-Home (282)

annotations. Then, augmentation steps such as horizontal and vertical flips, color channel permutation, and rotations were applied to increase diversity. Thus, the resulting augmented data sets D^1_{aug}, D^2_{aug} contain four times more data compared to the original data sets. In addition, based on D^2_{aug}, we generated a gray-scaled version D^2_{gray} to compare detector performance on three- and single-channel inputs.

Experiments

Given our hand data sets, the goal is to use *Tiny-YOLOv4* to create a robust hand detector. Since the architecture of *Tiny-YOLOv4* does not involve fully linked layers, but only convolutional layers, the size of the input image s_t can vary during model training and s_i during inference (on the RaspberryPi). The only condition is an input divisible by 32. On the one hand, it is hard for the model to extract meaningful features when s_t becomes small. On the other hand, the number of frames per second (FPS) on the RasperryPi must be high enough to detect fast grips, which can be achieved by small s_i. Empirically, 10 FPS has been found to be sufficient, which can be achieved with $s_i \leq 160$ pixels. However, s_t and s_i must not differ too much, otherwise the model would have to detect objects of a magnitude that are not well represented during training. In addition to the development of the hand detector, further experiments were conducted to optimize the detection of empty boxes. Overall, the performance of a saturation-based and an edge-based classification approach is tested on 280 labeled box images.

Hand Detection

Training&Evaluation: The training process is based on the open source framework *Darknet*. To initialize the convolutional layers and thus speed up the training process of the network, the pre-trained weights *darknet53* are used. Shared hyperparameters of all runs are: $batch\_size = 64$, $lr = 0.00261$, $subdivions = 16$ and $max\_batches = 6000$. First, D^1_{aug} was used to train a model with $s_t = 480$. To validate performance on smaller resolutions, we set the input size to 160 and 96 in subsequent iterations. After extending D^1 the process was repeated with D^2_{aug} and D^2_{gray}.

An eight-fold cross validation was used, dividing the data set into one test, one evaluation and six training folds. To determine the most suitable confidence threshold λ_{conf} of a given model based on a data set, common evaluation measures were computed and averaged on the test folds, using a fixed confidence threshold $\lambda_{conf} \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Due to data scarcity and the decision to use all data for final model training after evaluation, we lack a general test data set to compare all model performances. Therefore, the models are used more to compare the quality of the data sets and determine the most appropriate one for our application. Therefore, the focus is on qualitative evaluation, which is achieved by observing and analyzing the robustness and accuracy of the models in exemplary setups. We tested $s_i = 160$ and $s_i = 96$.

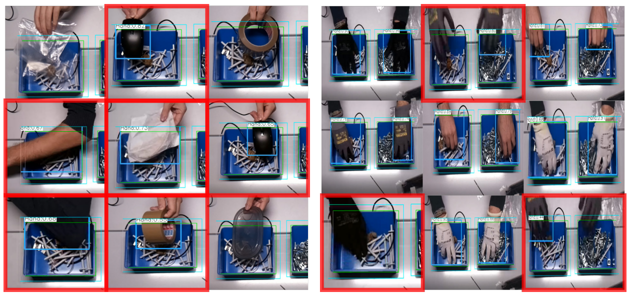
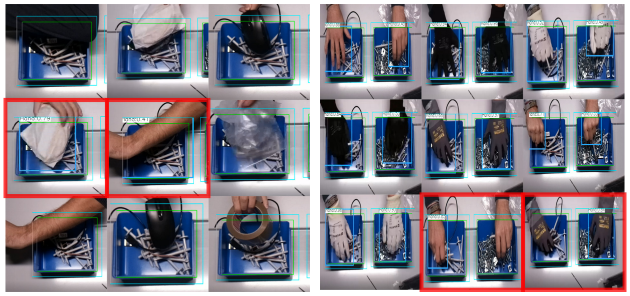

(a) No-hand objects (b) Hands

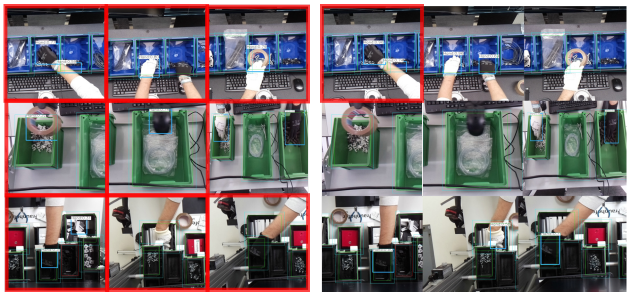
Figure 4: Performance of the model trained on $D^1_{aug,480}$, $s_i = 96$ and $\lambda_{conf} = 0.4$. *FP*s and *FN*s are marked. The model recognizes almost every hand with or without a glove (b), but tends to misidentify objects with skin or glove-like color or shape as hands (a).

Results Looking at the F1 score, the best model based on D^1 achieves a score of 0.98, while using $\lambda_{conf} = 0.3$ and 0.4, respectively, and $s_t = 480$. For D^2, the same λ_{conf} and s_t also yield the best model with an F1 score of 0.93. However, it cannot be concluded from the scores that the first model outperforms the second for the reasons stated above, since both are tested with different retained data from the corresponding data set. In addition, the quantitative results do not necessarily translate to operating performance due to the different input variables s_t and s_i. Qualitatively, both models can recognize hands and gloves with good accuracy, with the model trained on D^2 performing significantly better, see Fig. 4 (b) and Fig. 5 (b). The model trained on D^1 shows significant weakness in correctly classifying objects without a hand, see Fig. 4 (a). Instead, Fig. 5 (a) confirms the effectiveness of extending D^1 to include non-hand objects, as robustness improved significantly. However, both models struggle with not recognizing skin (e.g., arm) or glove-colored objects (e.g., white bag) as hands. Fig. 6 compares robustness and accuracy at different workstations and shows that it is beneficial to include images taken at different workstations, even if the same person uses the same boxes and gloves. Whether $s_i = 96$ or $s_i = 160$ is chosen depends on the objective. Using 96 gives higher FPS ($\sim$ 16), but with $s_i = 160$ the model tends to be more robust and accurate. This is the expected behavior, since with the latter s_i is closer to s_t. We refrain from presenting further results of runs with D^2_{gray} and varying s_t, since they do not lead to a better result at the current stage of the evaluation.



(a) No-hand objects (b) Hands

Figure 5: Performance of the model trained on $D^2_{aug,480}$, $s_i = 96$ and $\lambda_{conf} = 0.4$. *FP*s and *FN*s are marked red. The model detects hands with similar accuracy like in Fig. 4 (b), but is significantly more robust against no-hand objects (a). However, the model still struggles with skin or glove-like colored objects, such as the white bag or human arm.



(a) Model based on $D^1_{aug,480}$ (b) Model based on $D^2_{aug,480}$

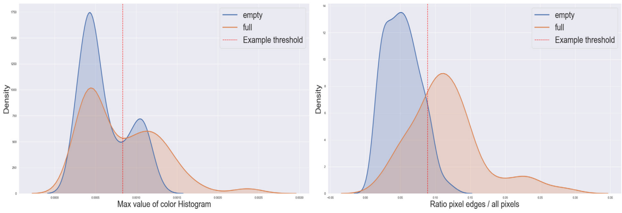
Figure 6: Comparison of the models based on different data sets. s_i is set to 160 and $\lambda_{conf} = 0.3$. The models are tested at different workplaces. *FP*s and *FN*s are marked red. The selected data shows a significant improvement in (b) over *FP*s and *FN*s.

Empty Box Detection

Visual recognition of empty boxes is challenging because there are a variety of boxes and components (screws, foils, plates, nuts, plugs, ...) that differ in size, shape, color, and texture (from rough and many edges to completely smooth and transparent), examples are shown in Fig. 7.

Our first attempt was to derive a box from the saturation histogram by comparing the maximum value to a threshold λ_{hist}. This works well for small components that differ from the box color, but weakens as components get larger and have more uniform colors or boxes are not monochromatic. In addition, light incidence and reflections in empty boxes lead to *false negatives*. We introduce a new method where edges are detected and counted. An empty box is inferred if the number of edges found is less than λ_{edge}. To compare both approaches, 140 images per class were collected and labeled. Both the *False* and *True Positives*, and the *False* and *True Negatives* are considered. For edge detection, OpenCV's *Canny* function is used. Both methods are based on the assumption that the color of the components is significantly different from the color of the box. However, the edge-based approach provides more components (kernels, thresholds, dilation) that can be adjusted to fit the algorithm to the data. The consequences are shown in Fig. 8, as the edge-based detection results in higher *True* and lower *False Positives* and *Negatives*. Nevertheless, the system we designed allows the user to choose between the methods, as the saturation-based approach also has advantages, e.g., the approach tends to perform better on smooth objects such as transparencies.

Figure 7: Small excerpt from the variety of box components.



(a) Saturation-based approach (b) Edge-based approach

Figure 8: Comparison of empty box detection methods: we use 280 images manually annotated with the label empty or full. For each class, we compute the inverse of the most frequent color value (a) and the ratio between pixels belonging to an edge and all image pixels (b) and plot their density. The example thresholds $\lambda_{hist} = 0.0008$ and $\lambda_{edge} = 0.089$ yield 103 *TP*s, 66 *TN*s, 74 *FP*s, and 37 *FN*s for the saturation-based approach and 126 *TP*s, 102 *TN*s, 37 *FP*s, and 14 *FN*s for the edge-based approach.

Conclusion, limitations, and future work

We have presented a worker assistance system for the packaging process, designed for use in protected workshops. The crux of our work is that we avoid the use of expensive hardware and develop the project as open source. This allows for replication at an unbeatable cost and flexible adaptations for the future. Our designed data set allows training an object detector capable of detecting hands with and without gloves. Although the detection is not perfect, we are confident that more data from real-world deployments will further improve the detection. In addition, we have presented two efficient methods to detect an empty box. All this leads to the support of the worker and the supervisor and saves time for interpersonal issues.

The system does not require the detection of individual parts in order to be cost-effective and universally applicable. However, this means that it is not possible to determine how many components the worker has gripped. This can be avoided by using multiple boxes containing the same components. Since the workspace and camera angle are limited, this is also not applicable for many boxes. Robust hand and removal detection also becomes difficult when boxes are stacked rather than side by side. Since our hand detection method is learning-based, it suffers from generalization and novel test data. The more data can be collected during operation in different protected workshops, the more the hand detection can be improved.

Future work will investigate and compare the performance of other lighter object detectors. In addition, it is expected that a more powerful computing unit would result in higher FPS while increasing s_t, which would certainly improve robustness and accuracy. The application should be tested in several protected workshops. Previously unused gloves and boxes needs to be added to the data set, and the model has to be fine-tuned on objects that seem difficult to classify. A cloud-based solution connected to each assistance system can simplify the data collection process, thereby improving the amount and variance of training data.

References

[1] Markus Funk et al., Mobile In-Situ Pick-by-Vision: Order Picking Support using a Projector Helmet, 9th ACM International Conference, pp. 1-4. (2016).

[2] Bernhard Mandl et al., Enhancing workplace learning by augmented reality, Seventh International Conference, pp. 1-2. (2017).

[3] Anhong Guo et al., A comparison of order picking assisted by head-up display (HUD), cart-mounted display (CMD), light, and paper pick list, ACM International Symposium on Wearable Computers, pg. 71-78. (2014).

[4] Christopher Stockinger et al., The Effect of Pick-by-Light-Systems on Situation Awareness in Order Picking Activities", Procedia Manufacturing 45, pp. 96-101. (2020).

[5] Angela L. Sauer et al. Assistive technology effects on the employment outcomes for people with cognitive disabilities: a systematic review, Disability & Rehabilitation: Assistive Technology, pp. 377–391, (2010).

[6] Munir Oudah et al., Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. J. Imaging, 6, 73 (2020).

[7] Licheng Jiao et al., A survey of deep learning-based object detection, IEEE Access 7, 128837-128868 (2019)

[8] Alexey Bochkovskiy et al., YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv preprint arXiv:2004.10934, (2020).

[9] Mohiminul Islam et al., Real time Hand Gesture Recognition using different algorithms based on American Sign Language, icIVPR, pp. 1-6. (2017).

[10] Smit Desai et al., A. Human Computer Interaction through hand gestures for home automation using Microsoft Kinect, International Conference on Communication and Networks, Xi'an, China, 2017, pp. 19–29.

[11] Laura Dipietro et al., A Survey of Glove-Based Systems and Their Applications. Systems, Man, and Cybernetics, Part C: Applications and Reviews, 38, 461 - 482 (2008).

[12] Khamar Basha Shaik et al., Comparative study of skin color detection and segmentation in HSV and YCbCr color space, Procedia Comput. Sci., 57, 41–48 (2015).

[13] Robert Y. Wang et al., Real-time hand-tracking with a color glove, ACM Trans. Graph, 28, 1–8 (2009).

[14] Xenophon Zabulis et al., Vision-based hand gesture recognition for human-computer interaction, The universal access handbook 34, 30, (2009).

[15] Shanxin Yuan et al., Depth-based 3d hand pose estimation: From current achievements to future goals, CVPR, 2018, pp.2636-2645.

[16] Mark Bayazit et al., Real-time Motion-based Gesture Recognition Using the GPU, Proceedings of the MVA, Yokohama, Japan, 2009, pp. 9–12.

[17] V.S. Kulkarni et al., Appearance based recognition of american sign language using gesture segmentation, Int. J. Comput. Sci. Eng, 2, 560–565 (2010).

[18] Ross Girshick, Fast R-CNN, ICCV, 2015, pp. 1440–1448.

[19] Ross Girshick et al., Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, pp. 1137–1149 (2017).

[20] Kaiming He, Mask r-cnn, ICCV, pp. 2980– 2988, (2017).

[21] Wei Liu et al., Ssd: Single shot multibox detector, Computer Vision – ECCV, pp. 21–37 (2016).

[22] Howard, A.G. et al., MobileNets: efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv 2017,arXiv:1704.04861.

[23] Sandler, M. et al., MobileNetV2: Inverted Residuals and Linear Bottlenecks, CVPR, Salt Lake City, UT, USA, 2018, 4510–4520.

[24] Howard A et al., Searching for MobileNetV3, ICCV, Seoul, Korea (South), 2019, pp. 1314-1324.

[25] Zhao, H. et al., Mixed YOLOv3-LITE: A Lightweight Real-Time Object Detection Method. Sensors 2020, 20, 1861.

[26] R. Huang et al., YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers, in Proc. IEEE Int. Conf. Big Data (Big Data), Seattle, WA, USA, 2018, pp. 2503–2510.

[27] Adarsh P et al., YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020: 687- 694

[28] Q. C. Mao et al., Mini-YOLOv3: RealTime Object Detector for Embedded Applications, IEEE Access, 2019, 7:133529–133538

[29] Wong A et al., YOLO Nano: a Highly Compact You Only Look Once Convolutional Neural Network for Object Detection. arXiv 2019, arXiv:1910.01271

[30] Fang W et al., Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments. IEEE Access, 2020, 8:1935-1944.

[31] Sven Bambach et al. Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions, (2015)

[32] E. Hsiao, A. Collet and M. Hebert. Making specific features less discriminative to improve point-based 3D object recognition(2010)

[33] Venkateswara, Hemanth and Eusebio, Jose and Chakraborty, Shayok and Panchanathan, Sethuraman, Deep Hashing Network for Unsupervised Domain Adaptation (2017)

Author Biography

Micha Christ received his BS in media informatics from the Media University Stuttgart in 2019. Since then he has been studying computer science with focus in machine learning to obtain a Master's degree. He works at the department of image and signal processing at Fraunhofer IPA as a research assistance.

Christian Jauch has been a researcher at the Fraunhofer Institute for Manufacturing Engineering and Automation IPA in Stuttgart, Germany since 2015 and studied technical cybernetics at the University of Stuttgart. He works in the department of image and signal processing and is part of the group scene analysis. His work focuses on industrial applications, more specifically on hand pose estimation and detection of hand gestures in manual assembly scenarios.

Dr. Julia Denecke studied computer science at the university of Stuttgart. Since 2007 she has been a research associate at the Fraunhofer Institute for Manufacturing Engineering and Automation in the department of Image Processing and Signal Analysis. 2013 she finished her PhD in the topic of volume data processing. Since 2016 she is the group leader of the scene analysis and focusses on 2D and 3D applications for dynamic detection in scene context.

Saskia J. Wiedenroth has been a researcher at the Fraunhofer Institute for Manufacturing Engineering and Automation IPA in Stuttgart, Germany since 2015 and studied Interactive Media Systems at the HS Augsburg. She works in the department of image and signal processing and is part of the group scene analysis. Her work focuses on human centred software development.