POLITECNICO DI TORINO
Repository ISTITUZIONALE

How to improve human-robot collaborative applications through operation recognition based on human 2D motion

Original
How to improve human-robot collaborative applications through operation recognition based on human 2D motion / Sibona, F., Cen Cheng, P.D., Indri, M.. - ELETTRONICO. - (2022). (IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society Brussels (Belgium) 17-20 October 2022) [10.1109/IECON49645.2022.9969120].

Availability:
This version is available at: 11583/2970839 since: 2022-12-15T13:14:55Z

Publisher:
IEEE

Published
DOI:10.1109/IECON49645.2022.9969120

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

18 November 2026

How to improve human-robot collaborative applications through operation recognition based on human 2D motion

Fiorella Sibona, Pangcheng David Cen Cheng, Marina Indri
Dipartimento di Elettronica e Telecomunicazioni
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
{fiorella.sibona, pangcheng.cencheng, marina.indri}@polito.it

***Abstract*—Human-robot collaborative applications are generally based on some kind of co-working of the human operator and the robot in the execution of a given task. A disruptive change in the collaborative modalities would be given by the capability of the robot to anticipate how it could be of help for the operator. In case of an Autonomous Mobile Robot (AMR), this would imply not only a safe navigation in presence of a human operator, but the automatic adaptation of its motion to the specific operation carried out by the operator. This paper investigates the possibility of achieving operation recognition by monitoring the human motion on a 2D map and classifying his/her path on the map, taken as an image data sample. Deep learning state-of-the-art libraries and architectures are exploited with the aim of making the robotic system aware of the ongoing process. The reported results, relative to a small training dataset, are nonetheless promising.**

***Index Terms*—Human-robot collaboration, AI, Mobile robotics**

I. Motivations and State of the Art

AI-based solutions and digital technologies are evolving quickly in the recent years. The necessity to improve the productivity without penalizing human operators in the manufacturing industry is becoming a challenge. Instead of implementing the Industry 4.0 solutions, where all the tasks are automated while ignoring the human during the process optimization, it is possible to adopt the solutions envisaged in the Industry 5.0 [1], where the factory is human-centered, meaning that the autonomous robots are perceptive and aware about human intentions. Human activity recognition is widely investigated, leveraging sensors for multiple modalities to enable specific applications [2]. In the Industry 5.0 context, the robot is able to actively observe and learn patterns from human workers using machine learning algorithms, so as to predict the human actions and attempt to help. With these features, it is possible to enable mass customization instead of mass production [3].

The increasing demand on customized products requires the combined efforts of intelligent manufacturing systems along with unique human skills, such as creativity, complex reasoning and socio-emotional intelligence [4]. The resulting manufacturing system has high-precision automation and flexible infrastructure able to react to dynamic needs, thanks to the synergy between intelligent machines and humans with flexible problem-solver and decision-maker skills.

A human trajectory tracking algorithm mixing human-oriented Global Nearest Neighbour (GNN) data association and Kalman filter-based human tracking is proposed in [5]. Vision-based approaches to detect humans and objects are widely used; however, they do not provide accurate range information. For this reason, the authors of [5] combined the information of a 2D lidar and a RGB-D-based YOLO (You Only Look Once) system to correct missed information while tracking the human motion.

A dataset containing human motion trajectory and eye gaze data called THÖR (Tracking Human motion in the ÖRebro university) is presented in [6]. The data of humans moving in a room are collected mainly through a motion capture system running at 100 Hz, moreover, the overall dataset is enriched with information from a 3D lidar, eye gaze detectors and a RGB-D camera. The recorded trajectories are available in 2D maps, which are often used for training and motion prediction models of human motion. In [7], human body pose and gaze are analysed to obtain an accurate prediction of the human's intentions. A Recurrent Neural Network (RNN) is used to predict sequences of multiple and variable length actions. Gaze and skeleton dataset is collected using the Optitrack motion capture and Pupil Labs binocular eye gaze tracking systems, while the multiple action sequence comes from a CAD120 RGB-D motion dataset.

A Multiple Predictor System (MPS) for human motion prediction is proposed in [8]. Depending on the context, it automatically switches between three individual classifiers: velocity-based position projection, time series classification and sequence prediction. In order to enhance the robot's ability to adapt its behaviour in environments shared with humans, the MPS is added in the path planning algorithm. In particular, the human's head 2D coordinates are used as features for the predictors [9].

In [10], a human motion prediction algorithm is proposed using a Hidden Markov Model (HMM). The HMM learns a set of movements executed by a human operator in an assembly task and then generates motion transition and observation probability matrices. In this way, it is possible to predict the motion of the human operator and perform assistive motion planning in Human-Robot Collaborative applications. Similarly, HMM is proposed in [11] to recognise human activities based on the principle object affordances, i.e., the relationship

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

between the activity and a particular object/tool.

AlexNet, a Deep Convolutional Neural Network (DCNN) is modified employing transfer learning-enabled algorithm in [12], to enhance the robot's capability to learn human's actions. In particular, human actions can be divided in: (i) generic body motions, e.g., grasping or holding a tool and (ii) specific movements related to a context, e.g., actions performed while using a tool. The training procedure involves two separated deep neural networks, that analyse the human motion and identify the tools associated to the tasks.

A multisensor framework exploiting online transfer learning techniques for human tracking is presented in [13]. The performance for all possible combinations of 3D lidar, 2D lidar and RGB-D cameras are evaluated, and in particular, the solution that combines 2D lidar and RGB-D camera achieved the best results in terms of performance and precision to learn people's movements in the environment. In fact, the sensor's choice may enhance the robot's perception of the human [14], and therefore improve its learning curve about human intentions.

The solution proposed in this paper aims at implementing one of the main capabilities of the data-driven framework introduced in [15], trying to achieve operation learning by monitoring the human collaboration motion on a 2D map. This work main goal is to demonstrate that human behaviour prediction for improving human-robot collaboration applications can achieve satisfying results by classifying the corresponding human path on a map, taken as an image data sample.

The paper is organized as follows: Section II first quickly recalls the data-driven framework this work is part of and the relative problem scenario; it then describes the idea and implementation choices that led to the current solution. In Section III, the solution testing and discussion of results are unfolded. Finally, Section IV draws some conclusions and sketches the future work.

II. Position-based Operation Recognition

This section briefly recalls the definitions and main concepts developed in [15], where a Human-in-the-loop (HITL) data-driven framework has been introduced. The main objective of the outlined framework is to exploit human information to learn a model for operation and task recognition, to make a mobile cobotic platform or manipulator aware of the ongoing process, allowing to enable anticipatory behaviour for improved collaboration along a flexible production line.

A. Problem and Idea

According to the definitions given in [15], this work develops a *global function* of the framework, namely the recognition of the executed operation. Note that we consider an operation as composed of a set of tasks, and each task is brought on – at a "local" level – in collaboration with cobots at specific workstations. Given a collaborative application, the present work aims at allowing the system to learn to recognise the operation – at a "global" level – by collecting information on the human operator motion: at each set of human poses on the map corresponds a specific operation. Note that, learning from data with sufficient generalization capabilities requires data availability. Moreover, being able to straightforwardly identify the relevant information that a human gives back while moving around and predict the often unpredictable human behaviour are both quite complex objectives.

The idea is to emulate how a human being usually perceives its surroundings: the decision making anticipating an action is performed based on an approximated observation of the surrounding environment, in favour of efficiency and resource saving, i.e., when we look around we do not usually catch every single information coming our way before taking a decision. The goal of the presented solution is to demonstrate that the path traversed by a human operator is a sufficient information to identify the performed operation, to possibly anticipate human behaviour in the described restricted context scenario.

B. Solution

In order to record a set of poses occupied on a map by a human operator, the current solution takes advantage of the Sen3Bot meta-sensor implementation [16], [17], a smart AMR whose role is to monitor and safely cooperate with humans. Within the data-driven framework the project is brought towards a collaborative evolution, the Sen3Cobot. Indeed, as specified in [15], the global functions (i) and (ii), i.e., *human operator modeling* and *data collection* are resolved taking into consideration solely the human positions, identified by a computer vision state-of-the-art object detection algorithm.

In the proposed solution, this set of positions is taken track of by plotting it as a path. This way, the time information is not included, as it is not a relevant information for operation recognition. Rather, dropping the time information (i.e., preferring the path data with respect to the trajectory data) allows to extend the operation recognition to different operators, which of course take different total time of completion for each operation. Therefore, the chosen solution takes into consideration that the digital representation of path data, when plotted on 2D map, is simply a matrix. By dropping the time information and given the duality of images as matrices, we translate a spatio-temporal data recognition problem into an image classification problem. This interpretation of data has revealed to be crucial, as it allowed to add to the pool of possible methods to solve the problem a whole range of well-known and well-documented architectures, libraries and tools to implement deep learning models, along with a huge community often providing those tools as open source material.

Note that in the proposed solution the Sen3Cobot stack is improved with the understanding of the executed operation, i.e., implementing the framework global function (iii) – *robotic system awareness*. In fact, the AMR currently implements passive HITL behaviour, as it monitors the area to gather position information from the detected human, and interprets it as an operation to be recognized. However, in the context of the overall data-drive framework, this passive step for operation recognition is fundamental for the decision making before action of the mobile cobot: based on the confidence of the classification, the robot will be given different trajectories to follow. To this end, the robot will need to act according to the probabilities associated to each class of operations. This means the system will iteratively check, while gathering new data, if the guess has changed and send a different reference to the mobile cobot accordingly. With the aim of dealing with the data scarcity problem, the solution takes data augmentation through simple transformations as a first step toward model

improvement. For what concerns data complexity, choosing image datasets rather than video ones allows in some way to have less noisy data, since the image contains only the map and the detected relevant information, i.e., the human operator path.

1) Tools: This subsection briefly introduces the main open source SW tools on which the presented solution is based. First of all, it is worth recalling that the data gathering provided by the Sen3Bot is ROS 1-based (Robot Operating System), featuring a vision module exploiting YOLO, containerized using Docker. Also this additional recognition feature for the Sen3Cobot has been containerized using Docker. Docker [18] allows to build, run and manage Linux Containers. The development and testing of self-contained applications can be done in a lightweight, clean environment. Allowing to leverage GPU within containers, local resources are exploited for running – hundreds of times faster than a regular CPU – Neural Network (NN) based algorithms. This also avoids lag problems derived from using Cloud available GPUs, and security issues.

For what concerns the operation recognition/image classification problem, the `fastai` deep learning library, specifically its second version `fastai v2` [19], has been chosen. This library provides low, mid and high-level APIs to intuitively create deep learning models, either from scratch exploiting the Python libraries it is built on (PyTorch, NumPy, PIL, pandas and others), or allowing to use state-of-the-art architectures and techniques made available following best-practices to get the most out of the available hardware. The authors also made available an interactive book written with Jupyter [20], an open source project providing notebooks. A notebook is an interactive programming environment – whose cells' code can be modified and run straightaway – capable of working with different language backends, kernels, allowing to use several different programming languages. For this first solution implementation, notebooks represented a simple playground for code development and testing. The solution is built upon [21], which provides a docker image with pre-installed `fastai v2` libraries and notebooks, which has been modified to be adapted to solve the considered problem. This enabled containerization of the recognition feature. Given the solution description provided in Section II-B, Figure 1 summarizes the overall structure and tool choices.

Before moving on to the description of the implementation choices, it is fundamental to put in evidence some assumptions, which influenced the development of this preliminary solution. In order to develop a baseline model for our image classification problem, we assume that a single human operator is moving within the monitored area, without additional dynamic obstacles. Moreover, with the aim of starting out with a basic classification problem, the number of operation classes is limited to two. Note that we refer to *classes* of operations since this enables the possibility of fine-tuning pre-trained models using new operations, which may be variants of the main class. In fact, since we defined an operation as a set of tasks performed at workstations (providing the needed machinery/cobot to bring on the necessary task), having main classes of operations with slightly different variants is a plausible situation. Furthermore, the number of operations in a shop-floor is indeed usually limited to the available equipment and setup for the manufacturing of a certain product.

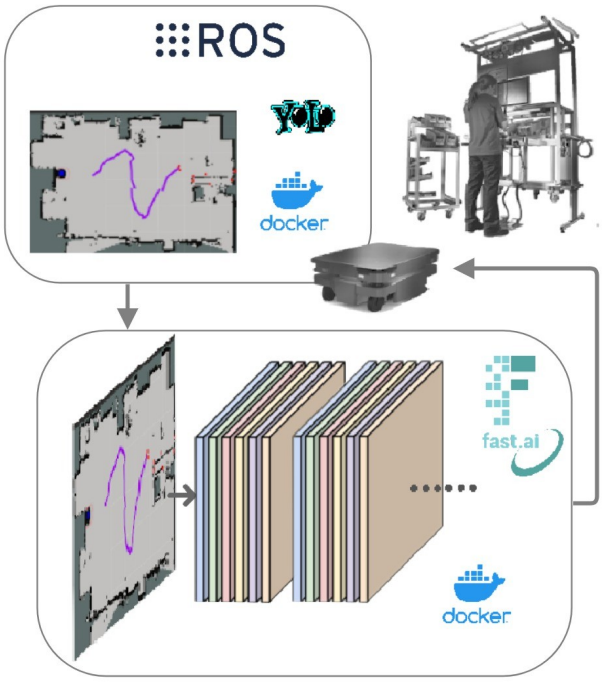


Fig. 1: The proposed solution aims at recognizing – through image classification – the operation executed by the human operator, by tracking the human position on a 2D map. This information will then be passed as input to the *robotic system control* global function.

2) Implementation: This subsection illustrates the solution development and implementation details.

Positions collection In the Sen3Bot stack, the positions associated with the detected human operator are published as virtual obstacles in the navigation local costmap of the AMR: this allowed to enable safe overcoming of the human obstacle. Within this work context, the published ROS topic provides a source of position messages to be plotted graphically on the RViz visualization tool.

Path plotting Filtering of messages was performed, since all points falling within the detected human obstacle bounding box are published as virtual obstacles: laser data may correspond to points behind the human obstacle surroundings (see Figure 2). Those points have been filtered out and, among the points covering the bounding box width, only the nearest one has been maintained at each sampling instant.

Each human 2D position is then collected by a ROS node that pushes it in a type `Path` ROS message – a standard message type in the navigation stack – which is then published on an ad-hoc ROS topic.

As can be seen in Figure 2, the resulting path is not very smooth. Nevertheless, since it tracks a human motion, this can be considered an expected output, due to the non-smooth human motion and detection noise. Also, the resulting path is inevitably dependent on the detection and messages publication (ROS node spin) frequencies.

Data collection Even though simulation has been taken into

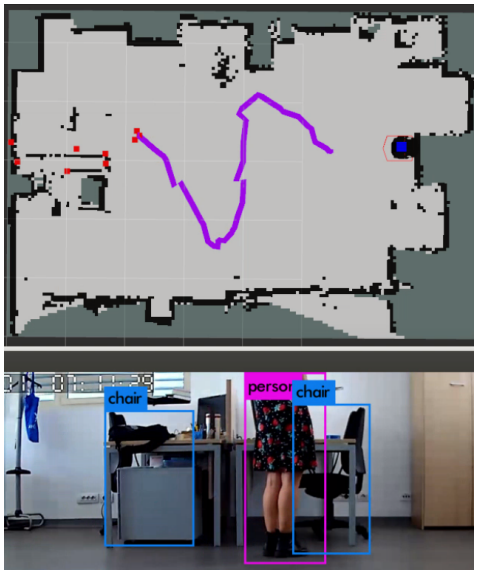


Fig. 2: After data filtering, the detected human path is published, and its content plotted on the 2D map by passing the type `Path` topic to an RViz Display.

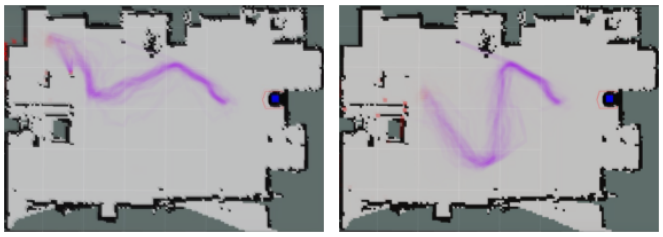
consideration, a collection of real data has been preferred. This is because training a learner on purely synthetic data will unavoidably affect its capabilities of performing classification on real data samples, to the extent that it might not be able to recognize any operation at all. Nevertheless, real data collection combined with data augmentation can improve overfitting issues.

With the purpose of speeding up the data collection, the operation executions have been video recorded and periodic screenshots of the Rviz map visualization have been generated.

Dataset creation Disjoint subsets of such collected samples have been used for training, validation, and testing of the algorithm, respectively. The samples have been gathered from two different human operators' motions, so as to improve generalization capabilities of the learning algorithm. Data samples have been saved in the dataset `/train` and `/valid` folders. Note that training and validation sets include only images representing completed operations. This is because we want the architecture to fit its parameters based on representative samples of each class of operations. On the other hand, to test the model capabilities to recognize an operation from the very beginning of its execution, testing is performed on samples of ongoing operation executions. This allows to observe the model capability to improve its confidence as the operation/path goes towards completion.

To give a compact representation of the collected dataset content, Figure 3 shows the mean values for each training set for each operation. The mean of all the image tensors corresponding to a certain class of operations is obtained by taking the mean along dimension 0 of the stacked rank-4 tensor. Notice that the manipulated tensor is rank 4, since the processed data are RGB images.

Architecture Since the operation recognition problem (interpreted as a plotted path recognition) is now an image classification problem, the well-known "go to" NN architectures for this kind of learning problems are Convolutional Neural Network (CNN) models. Specifically, a Deep ResNet (Residual Network) architecture has been used. ResNets address the degradation of training accuracy (vanishing gradient) problem, i.e., the degradation of training accuracy, associated with network depth [22]. To start out with a moderately deep network, a `ResNet18` has been selected as learning architecture. For what concerns the optimization step, we choose cross-entropy as our loss function, as it is the most common loss function used for binary classification problems. This function will be minimized by the stochastic gradient descent procedure during weight stepping. The learning rate has been set to 0.002, as suggested by `lr_find()`, a `fastai` function that plots the loss against learning rate values and outputs a suggested value, corresponding to the point where the gradient is the steepest.



(a) Class `A` operations mean value.(b) Class `B` operations mean value.

Fig. 3: Mean values images for each of the considered classes of operations, computed among the samples in each corresponding collected dataset.

Data augmentation A total of 300 image samples per class of operations have been collected. Despite being aware of the problem of overfitting due to data scarcity, the choice have been dictated by the purpose of testing the preliminary solution recognition capabilities keeping the number of labelled images low, for the sake of achieving a low complexity setup. Nevertheless, to improve generalization capabilities and avoid overfitting, data augmentation methods have been employed.

As a pre-processing step, all samples within the dataset have been resized to reduce their dimension, as – in our case – size reduction does not seem to affect the model performance. Then, a set of transformations have been selected, namely small rotations and warping, and lighting editing. This set of transformations are defined along with their relative application probabilities, i.e., the probability with which the transformation will be applied to random batch elements during training. The batch size have been set to 64 and the training algorithm will take care of shuffling in a random way across the training data set when choosing candidates for each mini-batch.

Moreover as a callback for every tweak of the training loop, we chose to apply the MixUp method [23]. MixUp generates new data during the learning procedure through convex combination of random pairs of images and

associated labels. A random sample of generated batch elements can be seen in Figure 4.

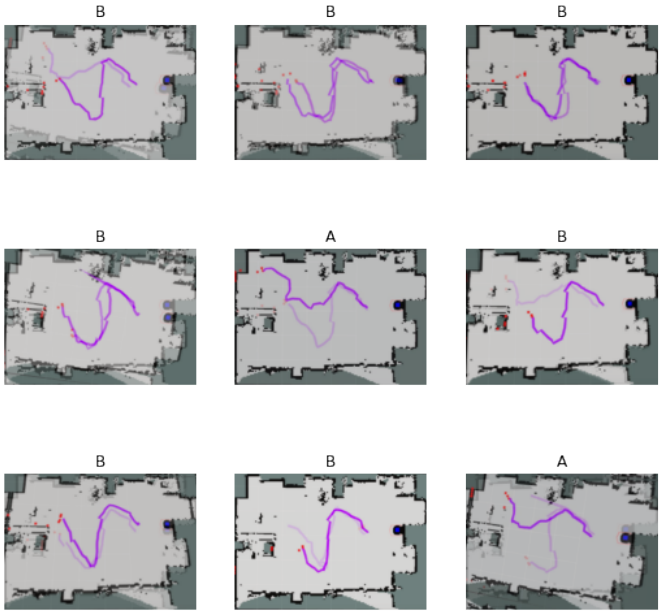


Fig. 4: Sample of batch elements generated by MixUp algorithm. As can be seen, shuffled samples are also affected by the randomly applied transformations for data augmentation.

Model Once the described hyperparameters have been definitively set, a one-cycle training policy have been performed [24], which is a commonly used method for training `fastai` models from scratch, i.e., without transfer learning. As expected, most times the accuracy saturated to 1 during the first couple of epochs, suggesting overfitting issues. Nevertheless, the main aim of the developed work is to test the obtained model on images representing the sequence of sub-paths corresponding to an operation. Therefore, the number of epochs for training has been limited to 1. An accuracy of about 0.93 has been achieved, with a training time of 4 s, running on a PC equipped with a 4GB GDDR6 NVIDIA GeForce GTX 1650 GPU. The trained model has been saved as a baseline model for the considered problem. Figure 5 shows a subset of the top losses peaked during training.

III. Testing

It is worth recalling that, in order to demonstrate the feasibility of the proposed solution for operation recognition problems to improve collaborative applications, the obtained baseline model should be able to distinguish different classes of operations. Additionally, it should ideally improve its guess confidence as it is provided with a sequences of images representing the progression of an operation execution. In fact, within the data-drive framework, according to the output of the proposed solution, a specific reference trajectory will be fed to the mobile robot control system. In particular, a reference trajectory should be generated, resulting from the weighted combination of candidate reference trajectories, were the weights are proportional to the associated operation class probabilities.

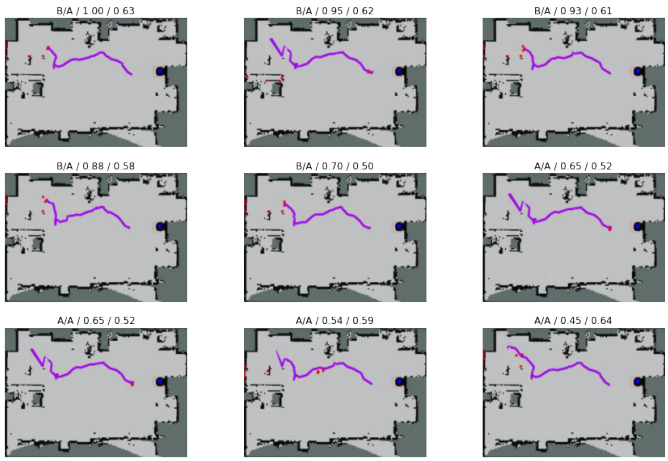


Fig. 5: Subset of samples that generated top losses. The title of each image shows: Predicted class / Actual class / Loss / Probability of actual class.

The baseline model has been first feed with progressive screenshots from a class `A` operation execution. Then, the same has been performed for a class `B` operation. Figure 6 reports the obtained testing results for both the class `A` operation and the class `B` operation testing samples.

As can be seen, for both sequences the model initially struggles to correctly recognise the ongoing operation. This is expected, since the first part of each representative path is identical for both classes of operations, as the first visited workstation is the same. Nevertheless, in the case of operation `A` recognition the classification results start with a couple of switching predictions, 74% `A` at step 1, passing through 59% `B` at step 6, and reaching 67% `A` at step 7, which lead to a prediction above 99% from step 8. The recognition of operation `B` generated similar results, starting from high probabilities associated to the wrong class (95% probability for `A` at step 1) and progressively switching to increasing probabilities for class `B`, eventually reaching above 95%, starting from step 8.

IV. Conclusions and Next Steps

This paper proposes a 2D human motion-based solution to perform operation recognition in the context of human-robot collaborative applications. The solution exploits deep learning state-of-the-art libraries and architectures to obtain a model able to recognize the operation associated to the motion of the monitored human operator. To mainly demonstrate the solution feasibility with the tackled problem, a small dataset have been prepared for training purposes.

The obtained results showed signs of overfitting issues, as expected due to data scarcity for training, but performed well in generalization capabilities when tested on images containing sub-path of the complete operation to be recognised. However, the results are promising, and the possibility to tune the context assumptions, the hyperparameters and the datasets dimension, leaves room for significant improvements.

The next steps include improving the overall model generalization capabilities by exploring hyperparameters choices, trying to get rid of overfitting through data collection and,

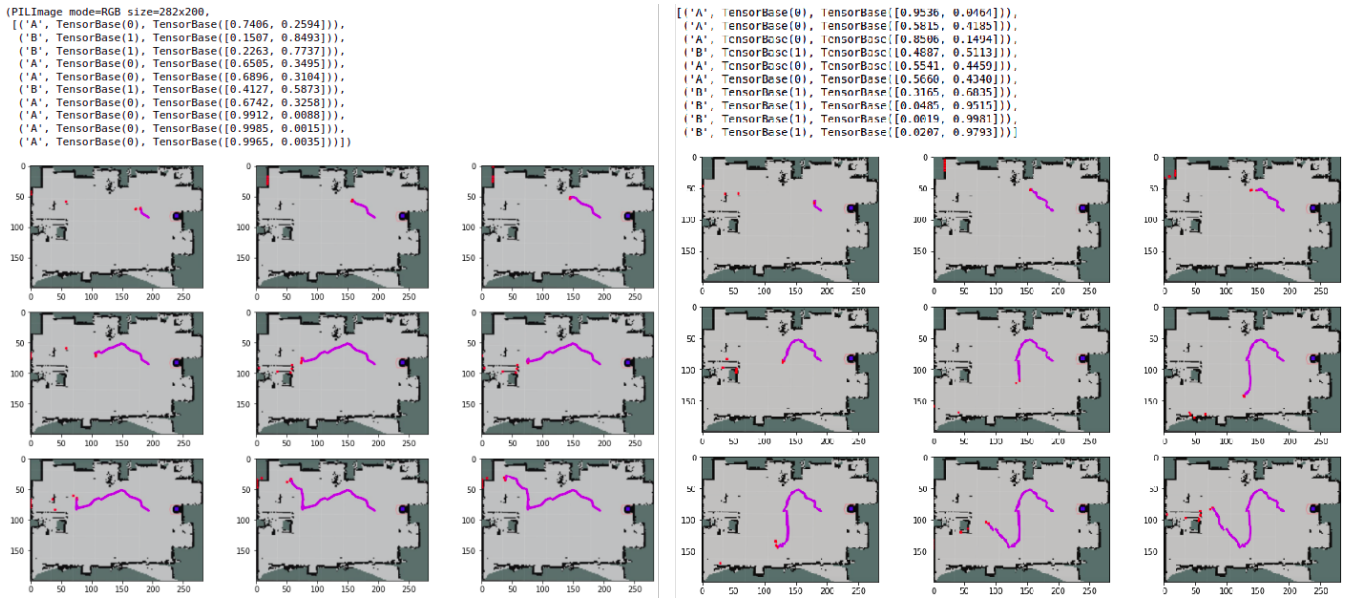


Fig. 6: On the left: prediction labels and probabilities, along with the set of testing sample images, corresponding to an operation `A` execution. On the right: prediction labels and probabilities, along with the input set of testing sample images, corresponding to an operation `B` execution.

mainly, exploiting advanced augmentation techniques. Moreover, the classification may be extended to a wider set of operation classes, so as to provide a more complete solution whose associated model can be then fine-tuned on small class variants.

REFERENCES

[1] S. Nahavandi, "Industry 5.0—a human-centric solution," *Sustainability*, vol. 11, no. 16, p. 4371, 2019.

[2] S. K. Yadav, K. Tiwari, H. M. Pandey, and S. A. Akbar, "A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions," *Knowledge-Based Systems*, vol. 223, p. 106970, 2021.

[3] P. K. R. Maddikunta, Q.-V. Pham, B. Prabadevi, N. Deepa, K. Dev, T. R. Gadekallu, R. Ruby, and M. Liyanage, "Industry 5.0: A survey on enabling technologies and potential applications," *Journal of Industrial Information Integration*, vol. 26, p. 100257, 2022.

[4] Y. Lu, H. Zheng, S. Chand, W. Xia, Z. Liu, X. Xu, L. Wang, Z. Qin, and J. Bao, "Outlook on human-centric manufacturing towards Industry 5.0," *Journal of Manufacturing Systems*, vol. 62, pp. 612–627, 2022.

[5] H. Bozorgi, X. T. Truong, H. M. La, and T. D. Ngo, "2D laser and 3D camera data integration and filtering for human trajectory tracking," in *2021 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2021, pp. 634–639.

[6] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "Thör: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 676–682, 2020.

[7] P. Schydlo, M. Rakovic, L. Jamone, and J. Santos-Victor, "Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5909–5914.

[8] P. A. Lasota and J. A. Shah, "A multiple-predictor approach to human motion prediction," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2300–2307.

[9] V. V. Unhelkar, P. A. Lasota, Q. Tyroller, R.-D. Buhai, L. Marceau, B. Deml, and J. A. Shah, "Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2394–2401, 2018.

[10] H. Liu and L. Wang, "Human motion prediction for human-robot collaboration," *Journal of Manufacturing Systems*, vol. 44, pp. 287–294, 2017.

[11] M. Cramer, J. Cramer, K. Kellens, and E. Demeester, "Towards robust intention estimation based on object affordance enabling natural human-robot collaboration in assembly tasks," *Procedia CIRP*, vol. 78, pp. 255–260, 2018.

[12] P. Wang, H. Liu, L. Wang, and R. X. Gao, "Deep learning-based human motion recognition for predictive context-aware human-robot collaboration," *CIRP annals*, vol. 67, no. 1, pp. 17–20, 2018.

[13] Z. Yan, L. Sun, T. Duckctr, and N. Bellotto, "Multisensor online transfer learning for 3d lidar-based human detection with a mobile robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7635–7640.

[14] A. Bonci, P. D. Cen Cheng, M. Indri, G. Nabissi, and F. Sibona, "Human-robot perception in industrial environments: A survey," *Sensors*, vol. 21, no. 5, p. 1571, 2021.

[15] F. Sibona and M. Indri, "Data-driven framework to improve collaborative human-robot flexible manufacturing applications," in *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2021, pp. 1–6.

[16] M. Indri, F. Sibona, and P. D. Cen Cheng, "Sensor data fusion for smart AMRs in human-shared industrial workspaces," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1. IEEE, 2019, pp. 738–743.

[17] M. Indri, F. Sibona, and P. D. Cen Cheng, "Sen3Bot Net: A meta-sensors network to enable smart factories implementation," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2020, pp. 719–726.

[18] D. Merkel, "Docker: lightweight Linux containers for consistent development and deployment," *Linux journal*, vol. 2014, no. 239, p. 2, 2014.

[19] J. Howard and S. Gugger, "Fastai: a layered API for deep learning," *Information*, vol. 11, no. 2, p. 108, 2020.

[20] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and Jupyter development team, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, 2016, pp. 87–90.

[21] A. Tabb, "Docker fastai repository," Available online: https://github.com/amy-tabb/fastai-docker-example (accessed May 2022).

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[23] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[24] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. International Society for Optics and Photonics, 2019, p. 1100612.