Deep Learning-Based Hand Gesture Recognition for Collaborative Robots

Cristina Nuzzi, Simone Pasinetti, Matteo Lancini, Franco Docchio, and Giovanna Sansoni

This paper is a first step towards a smart hand gesture recognition set up for Collaborative Robots using a Faster R-CNN Object Detector to find the accurate position of the hands in RGB images. In this work, a *gesture* is defined as a combination of two hands, where one is an *anchor* and the other codes the command for the robot. Other spatial requirements are used to improve the performances of the model and filter out the incorrect predictions made by the detector. As a first step, we used only four gestures.

We acquired four different small datasets with different characteristics to evaluate the performances in different situations: a first dataset in which the actors are dressed in casual wear; a dataset in which the actors wear skin-like color clothes; a dataset in which the actors wear light-blue gloves; and a dataset in which the camera is placed close to the operator. We tested the performances of the model in two experiments: first by using a test dataset composed of images of actors who were already present in the corresponding training dataset and second, by using a test dataset composed only of images of a chosen operator not present in the corresponding training dataset.

Our experiments show that the best accuracy and F1-Score are achieved by the *Complete* dataset in both cases, and that the performances of the two experiments are comparable. We tested the system in real-time, achieving good performances that can lead to real-time human-robot interaction with a low inference time.

Hand Gestures for Human-Robot Collaboration

Robotic systems are nowadays a fundamental part of the industrial world. Several types of these systems are available on the market, and often new innovative prototypes are designed depending on the required task. The wide use of robotics in the industrial world is justified by the great workload that a machine can bear and by the force and precision required by the specific operation. That is why robotic systems are placed into the so-called *robotic cells*, to keep them separated from human workers and to protect them from any possible damage. These robots are usually programmed to perform their movements at high speed and force, and without the help of information acquired by specific sensors, they are not able to guarantee the safety of the operator in every situation [1]. For these reasons, robots and humans in collaborative environments must work in a safe and efficient way, thus making the identification of effective means of communication a necessity.

To create an effective communication method for collaborative robots, as a first step, it is best to think of a human-human team: the human role in collaboration teams is fundamental, since the human operator has the know-how to perform operative tasks and the ability to identify issues that can arise during the operation, thus intervening to solve them [2]. This opens the door to a second step, focused on reproducing the naturalness of human-human communication in a human-robot team, using both voice and gesture commands [3].

However, natural gestures are not suitable for the industrial environment, where the safety of the operator is placed first and it is mandatory to avoid wrong and/or unnecessary commands.

In a previous work, detailed in [4], we presented a system based on Deep Learning for the recognition of gestures based on the simultaneous presence of the left hand closed as the *anchor gesture* and of the right hand used to specify the type of gesture, the focus being on the robustness of the recognition, even at the expense of gesture naturalness. Suitable datasets of images were collected and a Faster Region Proposal Convolutional Neural Network (or Faster R-CNN) [5] was implemented to detect the gestures. The tests carried out so far have shown that the approach is very promising, with a correct prediction of gestures in 88.50% of the observations. However, we realized that the overall performance could be improved: enhancing the prediction function used for inferencing the gestures (thereafter called Custom Prediction Function, CPF);

The research presented in this paper is a scientific extension of the work presented at *IEEE International Workshop on Metrology for Industry 4.0 and IoT 2018* [4].

1094-6969/19/$25.00©2019IEEE

Fig. 1. Examples of gestures performed by different actors. The START command is performed by both hands closed, the STOP one has the right hand open and the LEFT/RIGHT commands have the right hand pointing left or right accordingly. Note that the images are mirrored by the sensor.

and providing a different strategy in the composition and in the training of the image datasets. In this work we focus on these topics and show the resulting improvements.

Vision Systems and Deep Learning: Related Work

Machine vision for hand gesture recognition plays a central role in this context: it allows the robot to see the environment and/or to focus on its specific task. This means making the robotic system more flexible and automated, no longer a rigid and heavily-limited system. Artificial intelligence research developed rapidly in the last decade, and among the vast number of available algorithms, Neural Networks manifested an increase in usage, also because of the constant hardware improvements. *Deep Learning* in particular, the modern evolution of Neural Networks, has brought new turns in the world of artificial intelligence: through special types of networks called Convolutional Neural Networks (CNN), it is possible to explicitly extract information from visual data [6]. Neural Networks can also be used to determine the position of objects in an image: these algorithms, based on CNNs, are commonly called *Object Detectors*. Among the different detectors developed so far, one in particular has shone for years, the Faster Region Proposal Convolutional Neural Network (or Faster R-CNN) [7]. This algorithm uses a combination of two networks: a CNN for features extraction and a Region Proposal Network to detect and locate Regions of Interest (RoIs) in the image. Thanks to this combination and to the use of anchor boxes to speed up the detection of RoIs, this algorithm has an inference time suitable for real-time applications, achieving state-of-the-art accuracy.

Since object detectors focus only on a portion of the image which contains the object they were trained on, they are suitable to recognize human figures and human body parts, like the hands or the face [8]. The literature shows that object detectors are intensely and competitively used in autonomous driving research, to detect not only other vehicles and street signs but also pedestrians [9]. Hence, it is straightforward to apply them also at the hands to perform a simple recognition based only on visual features.

Not only 2D information is useful for the problem but also depth: in [10] a system implementing both RGB and depth data is shown, which uses the detection of the face to identify the human; then, by applying a skin color-based algorithm for the hands detection, it can safely recognize the hand also when it is overlapping with the face or when other people are in the background. This research has shown that the depth information alone can be used for gesture recognition instead of the combination of the two, pointing out the importance of the former.

The illumination of the scene is an issue for Machine Vision: that is why the research performed in [11] investigates the performance of a single near infrared (NIR) Depth camera instead of the traditional RGB and depth ones. The research showed that the depth map obtained with this method is a high-resolution one compared with those obtained from traditional depth cameras such as Kinects and proposes a novel method for acquiring 3D data based on a "space slicing" technique. As shown in [12], it is also possible to perform a semantic segmentation of the image and to assign to every group of pixels a specific label to separate objects by their semantic meaning, thus creating a different object detector with similar performances.

The Hand Gesture Recognition Procedure

Gestures

Gestures are defined in a very specific way: a gesture must be performed by both hands with the left one closed (*anchor*), and the hands must be sufficiently close to each other, both along the horizontal and the vertical dimensions. If any of these conditions is not met, the gesture is invalid. Some gesture examples are shown in Fig. 1.

We have 4 types of gestures to identify specific commands for a robot:

- **START:** this command is the first to be performed by an operator to start communicating with it. If other gestures are detected before the start command, no command is executed;
- **STOP:** if this command is detected, the robot is ordered to stop completely;
- **LEFT/RIGHT:** these commands in our experiment identify a specific path for the robot to be executed, not a constant directional control.

Overview of the Complete Procedure

As shown in Fig. 2, our complete procedure is composed of two main blocks: the Faster R-CNN Object Detector block

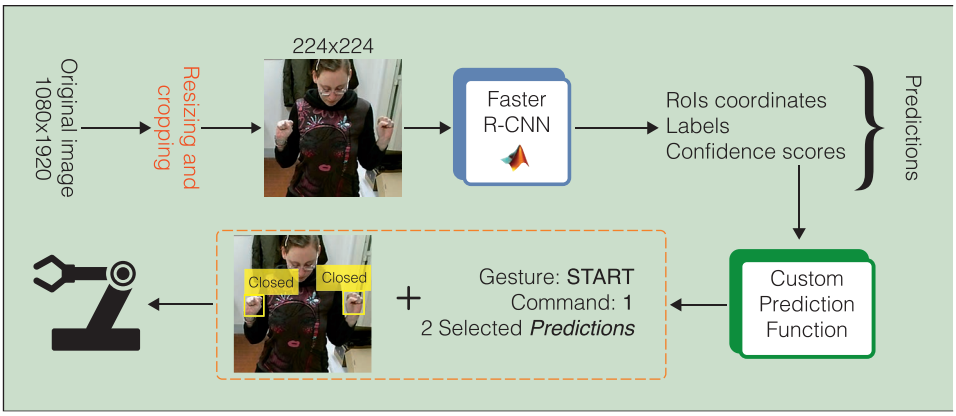

Fig. 2. Complete system overview. Starting from the left, the original image is acquired, cropped and resized to fit the size required by the algorithm and then passed to the Faster R-CNN Object Detector. The output of the detector is a set of predictions composed of three elements: RoIs coordinates, the Class Labels and the Confidence Scores, both relative to the objects in the RoIs. This set of predictions is passed to our Custom Prediction Function that determines the complete gesture and outputs a numeric command for the robot.

and the Custom Prediction Function (CPF) block. The Faster R-CNN detector carries out the detection of single-hand gesture; since it is the same as the one detailed in [4], [5], it is not presented in this paper. For the sake of clarity, it is sufficient to mention that its objective is to provide a set of *predictions* to the CPF block. Each prediction is composed of: the centers of the RoIs that locate the hands in the image (see the yellow boxes in Fig. 2); the Class Label of the hand gesture detected in each RoI; and the Confidence Score of the classification, which indicates how much the algorithm is confident about the prediction.

The CPF block checks if the predictions correspond to allowed gestures. To this aim, for each image, only the predictions with a confidence score greater than 90% are retained; among them, the presence of at least one closed-hand gesture is verified, otherwise the predictions are discarded, and the gesture is classified as NONE.

Residual predictions are good candidates to represent gestures: the challenge is to combine predictions in *pairs* of single-hand gestures, where one is the left-hand anchor and the other is the right hand (closed, open or pointing left or right, depending on the gesture).

Fig. 3 shows the steps carried out to perform this task:

- The centers of the image RoIs are used to compute the Euclidean distance $\mathbf{E_D}$ between each pair of predictions:

Fig. 3. Example of the workflow carried out by the CPF block. (a) RoIs detected by the Faster R-CNN algorithm in the image; (b) corresponding predictions and Euclidean distances among prediction pairs (black segments); (c) filtering out of outlier pairs; (d) filtering out of predictions corresponding to overlapping RoIs (black segments); (e) residual pairs; (f) residual RoIs; (g) lowest Confidence Score gesture; (h) highest Confidence Score gesture.

referring to the image in Fig. 3a, four RoIs are detected and six $\mathbf{E}_D$ values are obtained, each represented by a black segment connecting centers P_1, P_2, P_3 and P_4, as shown in Fig. 3b;

- Two thresholds are defined, denoted by $\boldsymbol{T}_{CD}$ (close distance) and $\boldsymbol{T}_{FD}$ (far distance), which define the lower ($\boldsymbol{T}_{CD}$) and the upper ($\boldsymbol{T}_{FD}$) boundary of a suitably pre-set range of $\mathbf{E}_D$ distances. Only those pairs of predictions with $\mathbf{E}_D$ values within this range are retained: in Fig. 3c the *pair* of predictions P_1-P_3 is eliminated and the residual pairs are represented by green segments;
- If two predictions correspond to RoIs which overlap in the original image, we discard the ones that have the lowest Confidence Score: in the example of Fig. 3c, prediction P_1 and P_3 belong to overlapping RoIs, and the former is eliminated, since its Confidence Score is lower than the one detected for prediction P_3. This step is graphically represented by the black segments in Fig. 3d, corresponding to eliminated predictions *pairs* P_1-P_2 and P_1-P_4;
- The prediction with the lowest column index of the image is considered to check if it corresponds to the *anchor* gesture: in the image of Fig. 3e, the anchor gesture is represented by prediction P_4. Hence, the Class Labels of all the predictions connected to P_4 are considered and, for each *pair*, the corresponding gesture is determined. Referring to the gestures in Fig. 3f, two pairs of predictions are considered, namely P_2-P_4 and P_3-P_4 , and are assigned to the STOP and to the LEFT gestures, shown in Fig. 3g and Fig. 3h, respectively. The pair P_2-P_3 is not a candidate since the anchor gesture is absent;
- The prediction pair with the highest total Confidence Score is selected among the residual candidates. This is the case of the gesture shown in Fig. 3h corresponding to the LEFT gesture.

In this work, the CPF block has been implemented in MATLAB, using conditional matrices as a filter-like approach, to avoid sequential operations in loops and to parallelize the calculations.

The Datasets and the Training Strategy

We used a combination of different datasets containing RGB images acquired by a Kinect v2 camera in different set ups, though any RGB camera can be used for the task. The actors were told to move around the test area while performing the gestures, so in our sets we have gestures performed while front-facing the camera and while being laterally oriented. Our datasets are as follows:

- *Base* Dataset: is composed of 609 images containing gestures taken from 15 different operators. With respect to our previous work, this dataset has been reduced of 140 images all corresponding to NONE gestures;
- *Light colors* Dataset: is composed of 383 images containing gestures taken from five different operators. These are taken with operators wearing clothes similar to their skin color or similar to the background color, to test the performance of the system in problematic conditions. With respect to our previous work, this dataset has been reduced by 17 images all corresponding to NONE gestures;
- *Gloves* Dataset: is composed of 400 images of gestures taken from five different operators. These are taken with the operators wearing light blue gloves, to create some contrast;
- *Zoom* Dataset: is composed of 710 images of gestures taken from seven different operators. These are basic gestures taken with the camera positioned very close to the operator to reduce the disturbances from the background. With respect to its previous version, this dataset has been increased by 380 images; in addition, nine images corresponding to NONE gestures have been eliminated;
- *Complete* Dataset: is composed of the combination of the above-mentioned datasets, to take into account each experimental set up;
- *Errors* Dataset: is composed of 140 images taken from the *Base* dataset, 17 images from the *Light Colors* dataset and nine images taken from the *Zoom* dataset. These images contain purposely NONE gestures represented by unallowed gestures and situations where multiple operators, not performing any gesture, are in the scene.

In this work, the *Complete* training and test datasets have been obtained through the following steps:

- The four datasets were randomly shuffled separately, to ensure a random selection of the data in the following step;
- From each dataset, 80% of the data has been selected for training (*Complete* training dataset) and the remaining 20% from each dataset has been selected for testing (*Complete* test dataset);
- The two datasets obtained so far (training and test) were both randomly shuffled, to ensure that the data in every batch used by the algorithm for training was not all from the same original dataset (*Base*, *Light Colors*, *Gloves* or *Zoom*).

This strategy was adopted to overcome the limitations of the one used in [4], where the *Complete* training and test datasets have been obtained through the following steps:

- The four datasets were combined into one dataset (*Complete* total dataset);
- The *Complete* total dataset was randomly shuffled;
- From the *Complete* total dataset, 80% of the data was selected for training (*Complete* training dataset) and the remaining 20% of the data was selected for testing (*Complete* test dataset).

In [4], the latter strategy turned out to be only partially adequate, the reason being that combining the four datasets first and then performing a random shuffle of the *Complete* total dataset cannot guarantee that the training and test datasets are going to be composed of a proportional number of samples from each original dataset.

Experimental Results

We tested the gesture recognition procedure both to evaluate the time required to train the Faster R-CNN detector (training

Table 1 – Training times		
Dataset	**This work**	**Previous work [4]**
Base	17.42 min	19.15 min
Light Colors	11.08 min	11.95 min
Gloves	11.23 min	12.10 min
Zoom	25.63 min	14.40 min
Complete	71.56 min	59.18 min

time) and to assess the ability of the CPF block to correctly detect the gestures (inference time).

Training Times

We trained the procedure on a laptop equipped with Windows 10, 16 GB of RAM memory, an Intel CPU i7-6700HQ and a GPU NVIDIA GTX 1060 with 6 GB of memory. The results are shown in Table 1, for the five datasets. Thanks to the modified training strategy and the random shuffle with a different seed, the training times obtained in this work are lower with respect to the ones obtained in [4] (reported in the right column of the Table), with the only exceptions of the *Zoom* and of the *Complete* datasets. This is by no means surprising, since the number of examples in the two training datasets increased by 53% and by 11% with respect to the first implementation, respectively.

Performances of the Gesture Recognition Procedure

We used a set of statistical tools to evaluate the performances of the system [13]. These are the following:

- The **Positive Predictive Value** (PPV) represents the ratio of correctly predicted positive observations to the total predicted positive observations;
- The **Sensitivity** (or True Positive Rate, TPR) represents the ratio of correctly predicted positive observations to the total positive observations in the ground truth;
- The **Specificity** (or True Negative Rate, TNR) represents the ratio of correctly predicted negative observations to the total negative observations in the ground truth;
- The **F1-Score** is the harmonic average of TPR and PPV (also called Precision and Recall) and is used to compare different experiments;
- The **Confusion Matrix** reports the number of predicted gestures with respect to the ground truth and gives a quantitative measure of the ability of the system to correctly recognize the gestures;
- The **Recognition Accuracy** (RA) is the number of total correct classifications (positive and negative) over the total number of test examples.

To evaluate the ability of the proposed procedure to generalize well, a second experiment has been carried out on the *Complete* dataset, this time excluding from the training dataset every image of a chosen actor, selected among the others because the number of images picturing him are sufficient for a test dataset without excessive reduction of the training dataset. Hence, the test dataset in this experiment is composed only of images of this chosen actor. The training times of this experiment are the same of the previous one because the numbers of images in the datasets are almost the same between the two experiments.

In Table 2, the evaluation metrics for every test are summarized: the results of the first experiment are reported in the first line, the results of the second one are reported in the second line, and the results obtained in our previous work are reported in the third line.

It is evident that all of the evaluated metrics increased for all of the datasets of the first experiment, the only exceptions

Table 2 – Performances Evaluation						
Dataset		**PPV**	**TPR**	**TNR**	**F1-Score**	**RA**
Base	**This work**	**94.55%**	**91.23%**	**25.00%**	**92.86%**	**86.89%**
		87.01%	**73.64%**	**36.84%**	**79.80%**	**68.22%**
	Previous work	86.40%	92.31%	48.48%	89.26%	82.67%
Light Colors	**This work**	**82.00%**	**63.10%**	**25.00%**	**71.30%**	**57.14%**
		44.23%	**46.00%**	**3.33%**	**45.10%**	**30.00%**
	Previous work	81.63%	60.61%	35.71%	69.57%	56.25%
Gloves	**This work**	**93.55%**	**95.39%**	**73.33%**	**94.66%**	**91.25%**
		86.84%	**94.29%**	**0.00%**	**90.41%**	**82.45%**
	Previous work	84.62%	88.71%	44.44%	86.61%	78.75%
Zoom	**This work**	**99.11%**	**94.10%**	**95.83%**	**96.52%**	**94.37%**
		93.75%	**82.42%**	**92.20%**	**87.72%**	**86.45%**
	Previous work	88.24%	83.33%	40.00%	85.71%	76.56%
Complete	**This work**	**97.53%**	**96.73%**	**83.33%**	**97.13%**	**95.01%**
		94.71%	**95.51%**	**78.41%**	**95.11%**	**92.12%**
	Previous work	91.94%	94.06%	64.79%	92.99%	88.50%

being the TNR values for the *Base* and *Light Colors* datasets and the TPR value for the *Base* dataset. This behavior was expected, because both the *Base* and the *Light Colors* datasets contain a number of true negatives significantly lower with respect to the experimental situation presented in [4].

The results obtained analyzing the *Gloves* dataset of the first experiment deserve a specific comment: this dataset has not been modified with respect to [4], hence the evident improvement of all the metrics can be related to the different implementation of the CPF block.

The results of the second experiment are lower and do not always surpass the ones obtained in the previous work; however, the results obtained for the *Complete* dataset show a notable improvement with respect to the previous work and are comparable with the ones obtained in the first experiment. The differences between the performances of the two experiments are due to the peculiarity of the images examined. In fact, by excluding a different actor from the one reported here the performances are quite different, due to: different hand size; different gesture performing style; and different quality of the images, provided that the excluded actor has a total number of images in the test dataset comparable with the ones in the test dataset of the first experiment.

The normalized confusion matrices of the *Complete* test dataset for both experiments are presented in Fig. 4, where the numbers along the diagonal represent the number of correctly predicted gestures. The images in Fig. 5 show, from left to right, correct predictions of the RIGHT (Fig. 5a), the START (Fig. 5b), the LEFT (Fig. 5c), the STOP (Fig. 5d) and the NONE (Fig. 5e) gestures. There are only a few errors, mainly due to the fact that the operator is acquired laterally with respect to the camera. The images in Fig. 5f and Fig. 5g correspond to this situation: in the former the STOP gesture is incorrectly predicted as a NONE, and in the latter the STOP gesture is incorrectly predicted as LEFT. Residual errors are related to: the presence of the background, which is very cluttered; the low resolution of the images; and incorrect bounding boxes dimensions and/or coordinates detected by the Region Proposal network, which leads to incorrect gesture predictions by the CNN, as shown in Fig. 5h and Fig. 5i. A background removal technique may be applied to reduce the first two effects, and an improved tuning of the parameters of the Region Proposal network may lead to better performances with respect to the third effect.

The developed procedure has been tested also on the *Errors* dataset, to evaluate its ability to detect true negatives. This dataset was made of 166 images: among them only five images represented correct gestures, the others being NONE. The procedure recognized one true positive, 107 true negatives, 55 false positives and three false negatives, resulting in the metrics shown in Table 3. It is worth noting that false positives (55) are observed for 89% of the occurrences in correspondence of images where more than one operator is in the scene, while residual errors are due to incorrect predictions of the Faster R-CNN block (Fig. 5i and Fig. 5j).

Finally, for every experiment we tested the offline inference time, which for every image has been calculated as the

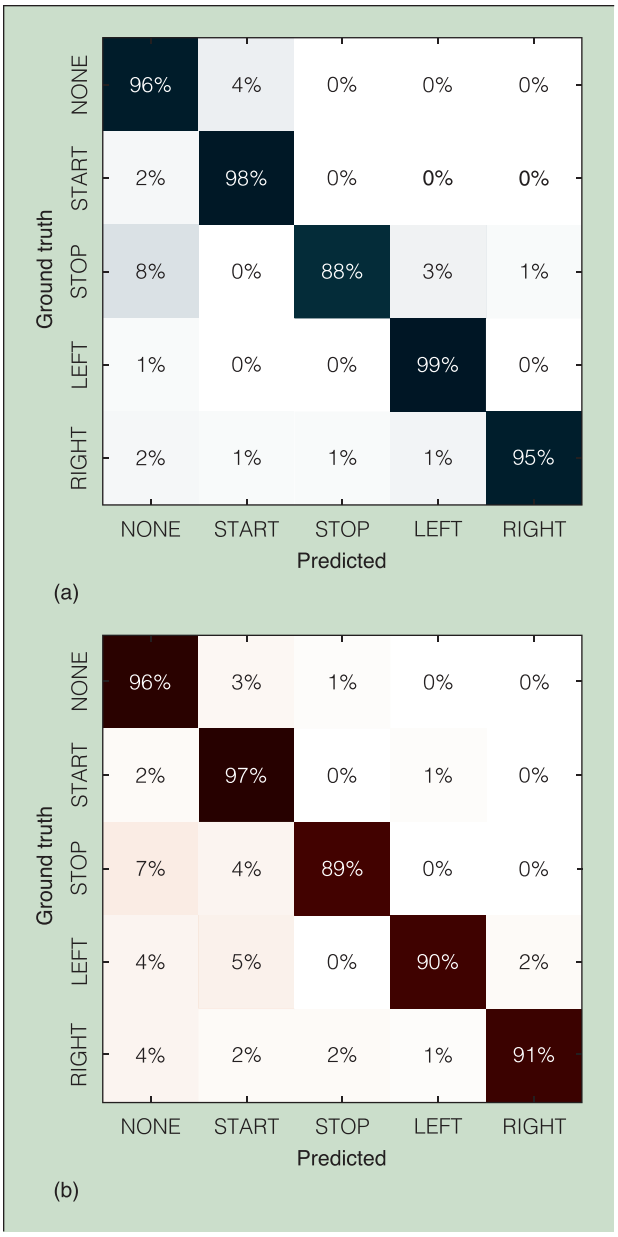

Fig. 4. (a) Normalized confusion matrix of the Complete test dataset obtained in this work. (b) Normalized confusion matrix of the Complete test dataset with a chosen operator excluded from the training dataset and only used for testing.

difference between the time at which the image is opened ($\mathbf{T}_{\mathbf{start}}$) and the time at which the CPF block outputs the predictions on the image ($\mathbf{T}_{\mathbf{end}}$). The average offline inference time is *0.11 s* with standard deviation of *0.01 s*. We also tested the online inference time, which for every image has been calculated as the difference between the time at which the image is acquired by the sensor ($\mathbf{T}_{\mathbf{acq}}$) and the time at which the CPF block outputs the predictions on the visualized image in real-time ($\mathbf{T}_{\mathbf{vis}}$).

To make the two experiments comparable, we performed the online test on the same number of images in the offline test, achieving an online inference time of *0.13 s* with standard deviation of *0.01 s*. This value is a little higher with respect to the offline inference time, due to the time spent for resizing,

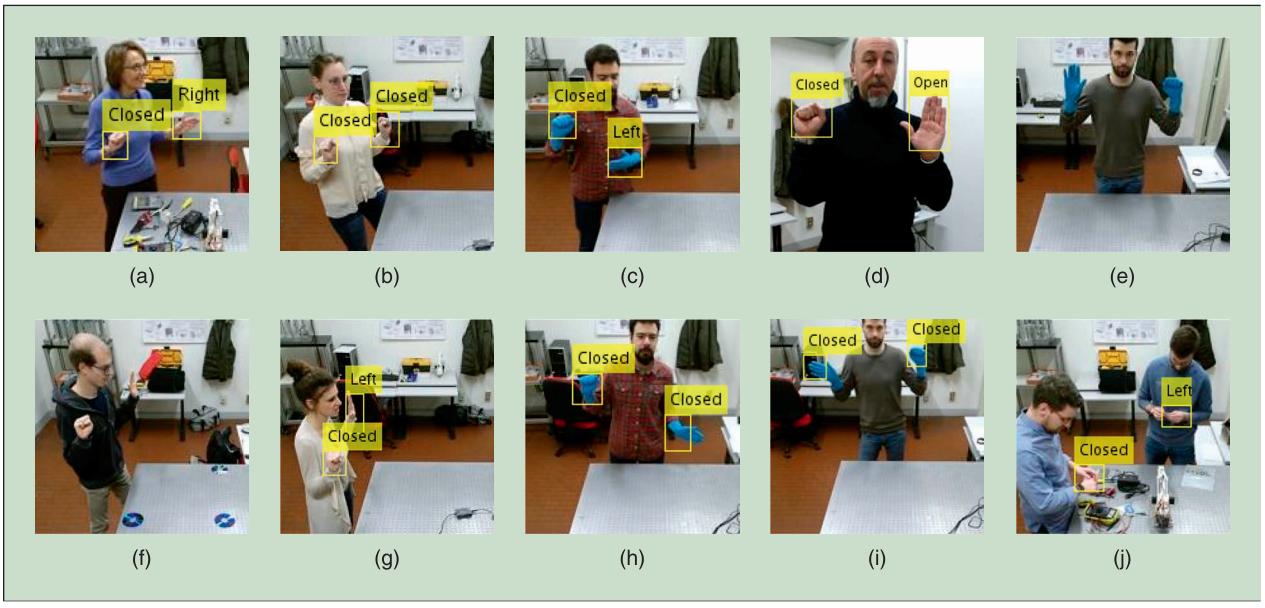

Fig. 5. Images from (a) to (e) show correct gestures taken from the Complete test dataset. In yellow there are the predicted bounding boxes. (f) Incorrect prediction of the STOP gesture; (g) incorrect prediction of the STOP gesture; (h) incorrect prediction of the RIGHT gesture. Images from (i) to (j) show false positives taken from the Errors dataset.

Table 3 – Performances on the Errors Dataset

Datasets	PPV	TPR	TNR	F1-Score	RA
Errors	1.79%	25.00%	66.67%	3.33%	65.06%

cropping and visualizing the acquired images before their processing; however, it is lower than the one measured in the previous version of the procedure (*0.23 s*), and it is well suited for real-time applications.

In both experiments we found that there is not any elaboration difficulty with respect to the different gestures, but the inference time can be higher for those images where the Faster R-CNN detector outputs more than two valid predictions: in this case the CPF intervenes and, according to the adjustments performed, requires more time.

Conclusions and Future Work

The work presented in this paper is a second step towards a more complex and sophisticated set up for collaborative robots. We first addressed the problem by defining simple but robust gestures to be detected by the system, and then we found out the limitations of the Faster R-CNN Object Detector algorithm used, as well as the development platform. Our experiments show that even if we use a relatively small dataset, the performances are quite good, and the inference time is suited for real-time applications, while keeping the training time low: this is due to the vectorized implementation of the procedure in the MATLAB platform as well as the presence of more than one hand in the images, thus doubling the training examples. The best performances of the algorithm are achieved when the operator stands in front of the camera, and while even far positions are usually recognized properly, it is best to stand relatively close to the camera, to guarantee a high-resolution image of the hand to be recognized and exclude the background as much as possible. As for our future work, we first want to expand the gestures by introducing new single-hand gestures and two-hand combinations, which can intuitively link to a robot command. Then we plan to test the new system in a set of experiments to control in real-time an industrial robot performing different tasks.

References

[1] J. Krüger, T. K. Lien and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP Annals*, vol. 58, no. 2, pp. 628-646, 2009.

[2] M. P. Pacaux-Lemoine, D. Trentesaux and G. Z. Rey, "Human-machine cooperation to design Intelligent Manufacturing Systems," in *Proc. IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016.

[3] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: a review," *Int. J. of Industrial Ergonomics*, 2017.

[4] C. Nuzzi, S. Pasinetti, M. Lancini, F. Docchio and G. Sansoni, "Deep learning based machine vision: first steps towards a hand gesture recognition set up for collaborative robots," in *Proc. 2018 IEEE Int. Workshop on Metrology for Industry 4.0 and IoT*, 2018.

[5] "Object detection using faster R-CNN deep learning," Mathworks. [Online]. Available: https://it.mathworks.com/help/vision/examples/object-detection-using-faster-r-cnn-deep-learning.html.

[6] Y. Lecun, Y. Bengio and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 05 2015.

[7] R. Shaoqing, H. Kaiming, G. Ross and S. Jian, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proc. 28th Int. Conf. Neural Information Processing Systems - Volume 1*, MIT Press, pp. 91-99, 2015.

[8] S. C. Hsu, Y. W. Wang and C. L. Huang, "Human object identification for human-robot interaction by using fast R-CNN," in *Proc. 2018 2nd IEEE Int. Conf. Robotic Computing (IRC)*, 2018.

[9] H. Kim, Y. Lee, B. Yim, E. Park and H. Kim, "On-road object detection using deep neural network," in *Proc. 2016 IEEE Int. Conf. Consumer Electronics-Asia (ICCE-Asia)*, 2016.

[10] M. Van den Bergh and L. Van Gool, "Combining RGB and ToF cameras for real-time 3D hand gesture interaction," in *Proc. 2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 66-72, 2011.

[11] D. Ionescu, V. Suse, C. Gadea, B. Solomon, B. Ionescu, S. Islam and M. Cordea, "A single sensor NIR depth camera for gesture control," in *Proc. 2014 IEEE Int. Instrum. Meas. Technology Conf. (I2MTC)*, pp. 1600-1605, 2014.

[12] C. Shuhan, W. Ben, L. Jindong and H. Xuelong, "Semantic image segmentation using region-based object detector," in *Proc. 2017 13th IEEE Int. Conf. Electronic Meas. Instrum. (ICEMI)*, 2017.

[13] K. J. Van Stralen, V. S. Stel, J. B. Reitsma, F. W. Dekker, C. Zoccali and K. J. Jager, "Diagnostic methods I: sensitivity, specificity, and other measures of accuracy," *Kidney Int.*, vol. 75, no. 12, pp. 1257-1263, 2009.

Cristina Nuzzi (c.nuzzi@unibs.it) started her Ph.D. degree work in applied mechanics at the University of Brescia, Italy in 2017 and works in the Laboratory of Vision Systems for Mechatronics under Prof. Giovanna Sansoni. She received the B.S. and M.S. degrees in industrial automation engineering from the University of Brescia in 2015 and 2017, respectively. Her research interests include vision systems, image processing and deep learning solutions for robotics, with particular attention on the metrological characterization of systems.

Simone Pasinetti (simone.pasinetti@unibs.it) has been a Research Fellow for the Laboratory of Vision Systems for Mechatronics (Vis4Mechs) in the Department of Mechanical and Industrial Engineering (DIMI) at University of Brescia, Italy since Jan. 2015. He received the B.S. degree and the M.S. degree (with honors) in industrial automation engineering from University of Brescia, in 2009 and 2011, respectively. He received the Ph.D. degree in applied mechanics from University of Brescia in 2015. During his Ph.D. work, he was in contact with the Institute of Intelligent Systems and Robotics (ISIR), Paris, France, where he carried out research concerning the robotics for rehabilitation and dynamic posture analysis.

Matteo Lancini (matteo.lancini@unibs.it) is currently an Assistant Professor with the Mechanical and Thermal Measurements Laboratory of the University of Brescia, Italy. He received the M.S. degree in mechanical engineering in 2005 and his Ph.D. degree in applied mechanics from the University of Brescia in 2015, with a thesis on measurement systems for robotic rehabilitation. He is a member of the International Society of Biomechanics. His current research interests include measurement systems for biomechanical analysis, in particular for robotic gait and rehabilitation, as well as industrial diagnostics using non-destructive techniques based on vibration measurements.

Franco Docchio (franco.docchio@unibs.it) is presently a Full Professor in Electrical Measurements with the Dipartimento di Elettronica per l'Automazione of the University of Brescia, Italy, where he joined in 1987. He received his M.S. degree in 1976 and worked at the Centro di Elettronica Quantistica of Italy between 1978 and 1987, where he carried out research concerning laser development, laser applications in industry and bio-medicine and laser-tissue interaction. Prof. Docchio, author of more than 250 publications, is a member of the Laboratory of Vision Systems for Mechatronics. He is currently Fellow of the European Optical Society.

Giovanna Sansoni (giovanna.sansoni@unibs.it) is now Full Professor of Electrical Measurements at the Department of Information Engineering and Head of the Laboratory of Vision Systems for Mechatronics (Vis4Mechs) at University of Brescia, Italy, were she joined in 1985. She received her degree in electronic engineering at the Politecnico of Milan, Italy in 1984. Her research interests are in the 3D vision area. Among these are the: implementation of camera and projector calibration for the absolute measurement of shape in active stereo vision systems; development of light coding methods for whole-field optical profilometry; application of optical instrumentation to the acquisition and the reverse engineering of free-form surfaces.