Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 00 (2019) 000–000

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), June 24-28, 2019, Limerick, Ireland.

Automatic process modeling with time delay neural network based on low-level data.

Giovanni Menegozzo[a], Diego Dall'Alba[a], Andrea Roberti[a], Paolo Fiorini[a]

[a]*University of Verona, Str. le Grazie 15, Verona, 37134, Italy*

Abstract

Automatic process modelling (APM) is an enabling technology for the development of intelligent manufacturing systems (IMSs). The analysis of obtained models enables the prompt detection of error-prone steps and the design of proper mitigation strategies, in all aspects of the manufacturing process, from parameter optimization to development of customized personnel training. In this work we propose a Time Delay Neural Network (TDNN) applied to low level data for the automatic recognition of different process phases in industrial collaborative tasks. We selected TDNN because they are suited for modelling time dependent processes over long sequences while maintaining computational efficiency. To experimentally evaluate the recognition performance and the generalization capability of the proposed method, we acquired two novel datasets reproducing a typical IMS setting. Datasets (including manually annotated ground-truth labels) are publicly available to enable other methods to be tested on them and they replicate typical Industry 4.0 setting. The first dataset replicates a collaborative robotic environment where a human operator interacts with a robotic manipulator in the execution of a pick and place task. The second set represents a human tele-operated robotic assisted manipulation for assembly applications. The obtained results are superior to other methods available in literature and demonstrate an improved computational performance.

Peer review under the responsibility of the scientific committee of the Flexible Automation and Intelligent Manufacturing 2019

Keywords: Intelligent manufacturing; Industry 4.0; Time delay neural network; TDNN; Collaborative Robot;

Peer review under the responsibility of the scientific committee of the Flexible Automation and Intelligent Manufacturing 2019

1. Introduction

The advent of the Industry 4.0 (I4) has introduced new production forms to allow greater process flexibility without penalizing efficiency and cost. According to [1] one of the nine implementation pillars of I4 is collaborative robots that support human operator in carrying out specific phases of the production process. The collaborative model represents one of the production objectives that need to be met in I4, gaining the best quality and consistency outcome while maximizing flexibility. Obtaining such objectives is very challenging, since on one hand we need to optimize robotic and production parameters while on the other hand we have to obtain the best possible human-operator performance without penalizing ergonomics and user experience. Especially for Small and Medium Enterprises (SME), flexibility is one of the main necessities for survival in current competitive markets [2], but the introduction of collaborative robots encounters difficulties related to management and integration in the industrial environment. From this it follows the lack of empirical literature on modelling collaborative industrial tasks [3]. Automatic process modelling (APM) can facilitate the integration of collaborative robot in SME by providing a more abstract and user friendly interpretation of collaborative systems and an easier integration with decision making and intelligent manufacturing systems (IMS).

The use of low-level data is coherent with I4 guidelines indicating an increasing adoption of cyber-physics systems and give straightforward access to heterogeneous data. The automatic process analysis of collaborative systems enables the prompt detection of error-prone steps and the design of proper mitigation strategies, including all the aspect of the process, from parameters optimization to predictive maintenance [4]. Therefore, APM needs to be fast, robust to different IMS setups and able to perform autonomous identification of target phases. We apply Time Delay Neural Network (TDNN) to low level data and automatically recognize different process phases for APM applications. We selected TDNN because they are suitable for modelling time dependent processes over long sequences while maintaining computational efficiency, since they can represent time relationships between events invariant under time-translation. Thanks to the TDNN ability to learn features from temporal events by analysing their context, TDNNs have been applied successfully to the analysis of heterogeneous data (e.g. low dimensionality kinematic data, images and videos)[5]–[7]. To experimentally evaluate the recognition performance and the generalization capability of the proposed APM method, we acquired two novel datasets reproducing an I4 IMS setting. Datasets (including video recordings, low level sensor data and ground-truth annotations) are publicly available to enable method benchmarking. The first dataset replicates a collaborative robotic environment where a human operator interacts with a robotic manipulator in a pick and place task. The second one represents a tele-operated robotic manipulator for micro assembly applications. The TDNN results are superior to other methods available in literature maintaining a reasonable computational cost. The proposed method and novel datasets are key-components in the development of future IMS with advanced situation awareness capabilities.

2. Method

We will first describe our method based on TDNN then we will present other three supervised algorithm for APM used as benchmarks. Different methods and datasets allow us to compare the capacity of the proposed network with affine standard approaches.

2.1. Time-delay neural network

The proposed TDNN have a pyramidal structure giving them a wider temporal context, i.e. the initial transforms are learnt on narrow ranges values and the deeper layers process the hidden activations from a wider temporal context due to the dilatation on nodes, as shown in Figure 1. Since the convolutions take place in the time axis, stacking layers with and increasing dilatation rate allows the model to learn wider temporal relationships, thus providing a higher feature abstraction. During back-propagation lower layers of the network are updated by a gradient accumulated over all the time steps of the input temporal context. Thus, the lower layers of the network are forced to learn translation invariant feature transforms [8].

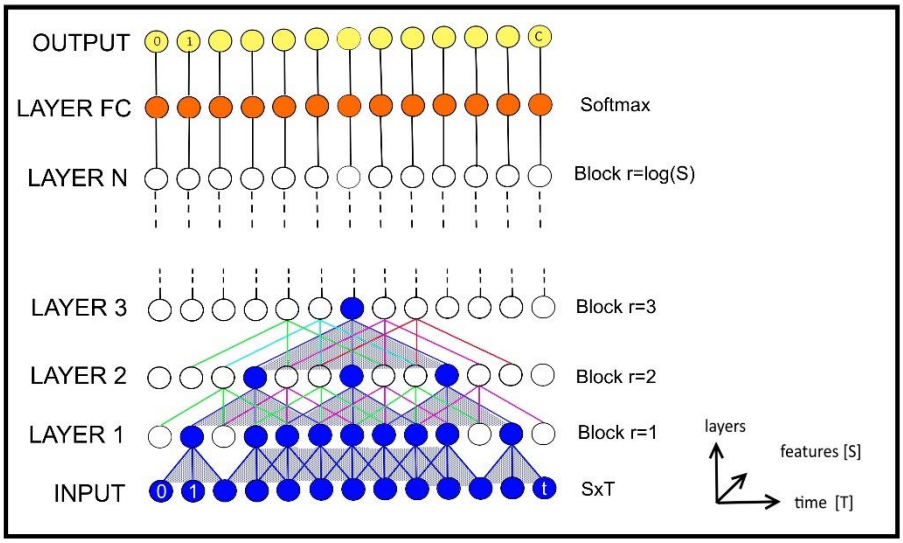

Fig. 1. TDNN network architecture

The inputs to our neural network will be a set of kinematic values available in both new datasets. We grouped all the kinematic values in the **S** vector. Let $\boldsymbol{S_t}$ be the input for time steps t for $1 \leq t \leq T$ where T vary for each data acquisition length. The process phase label for each time step is given by vector $\boldsymbol{Y_t} \in \{0,1\}^{Phase}$ such that the true phase is 1 and all others are 0. We define a block B as a tuple $< A, Re >$ where A is an atrous (or dilated) convolutional and Re is a Rectified Linear Unit activation function (using the notation introduced in [9]) with:

$$A = \sum_k x[r \cdot k] w[k] \quad (1)$$

$$Re = ReLU(x) \quad (2)$$

Where the atrous rate r corresponds to the stride used for sampling the input signal x defined as $x =< S_1, S_2, S_3 \ldots S_T >$, k is the length of the convolutional kernel and w represented its weights. We could observe that standard discrete convolution with kernel dimension k is a special case of atrous convolution with rate $r = 1$. The number of blocks N used in our network depends from the number of features F contained in the input vector S, following the relationship $N = \lfloor \log_2 F \rfloor$. The features contained in the vector S are described in Table I for each dataset considered in this study.

Therefore, our network is composed of the input layer I^{SxT}, N blocks $\sum_{r=1}^{\lfloor \log_2 F \rfloor} B$ with increasing atrous rate and a fully-connected layer with SoftMax activation function to obtain classification results, i.e. Y_t vector.

The proposed methods implementation is based on Keras, with GPU accelerated Tensorflow framework, and it has been tested on mobile workstation equipped with Intel i7 CPU, 8GB RAM and a Nvidia Geforce 1060m video card with 6GB video memory. We used ADAM stochastic optimization algorithm during training with 200 epochs and batch size of 16 [10].

2.2. Other methods

We benchmark the proposed TDNN with other 3 methods based on different approaches in temporal modeling: Long-Short Term Memory (LSTM), Support Vector Machine (SVM) and Random Forest (RF). These methods have been selected since they prove successful in industrial supervised task modeling, as reported in [11].

LSTM architecture belongs to recurrent neural networks. LSTM have been widely used in speech recognition and capture temporal dependencies using parameters that weigh the incidence of past cells on the present state [12]. Back Propagation Through Time (BPTT) recalculate the gradient with respect to the weights for each step and sum

them across timesteps. This leads to longer training times and a higher computational cost compared to feedforward networks [13].

SVMs are often used in industry field for their speed and generalization capacity [14]. Linear SVM performs the calculation of the optimal hyperplane decision boundary, separating one class from the other, based on a training dataset. Linear SVM are parametric and allow online learning with simple algorithms such as stochastic gradient descent. Unlike the Artificial Neural Networks (ANN) they converge to the global minima [15].

RF is an ensemble learning method based on decision tree. RF does not require the creation of a custom architecture depending on data type, but his classifier is able to achieve optimal results on a wide range of data types [16]. In RF, it is assumed that features are already available from some mechanism (e.g. feature engineering). They are fast and excellent for small dataset problem.

3. **Experiments**

We tested our method on our novel datasets, ICRT (Industrial Collaborative Robot-human Task) and VIT-MR (Virtual Industrial Task Master-slave Robot). Both datasets are acquired during the execution of pick and place tasks with different interaction methods. They include synchronized low-level data (including video and robot kinematic variables) together with ground truth annotation of task phases for the evaluation of APM methods.

Although they show differences on the classification objective (i.e. different process labelling) they present comparable kinematic variables. A method able to obtain high accuracy on both datasets implies the ability to generalize the specific dataset abstracting from the execution technology and from the phase considered. To achieve this goal, we used all kinematics data available for the description of the process without any feature selection or pre-processing steps. This choice has been motivated with the goal of demonstrating the capability of the method of working on raw low-level sensory data.

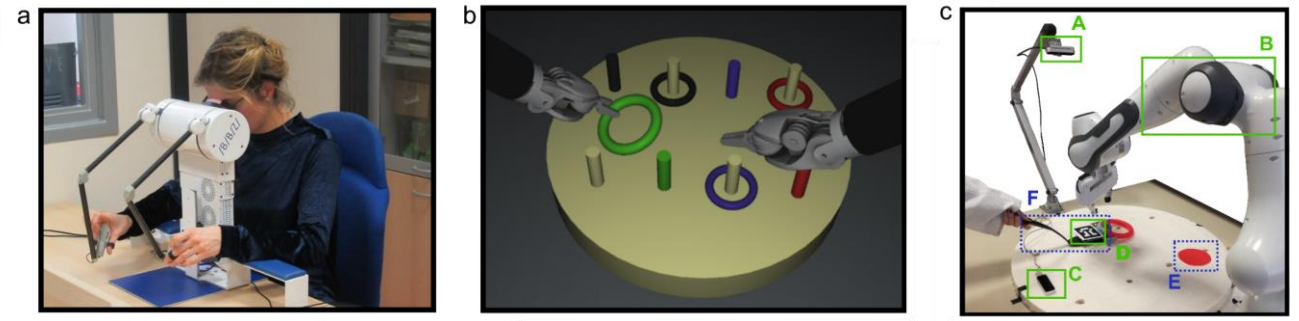

Fig. 2. (a) The hardware training console used by one of the students during data acquisition. (b) Example of the Virtual task considered in the VIT-MR dataset: four colored rings need to be placed in the corresponding peg. (c) ICRT dataset setup. (A) Intel RealSense, (B) Panda robot, (C) Leap Motion, (D) ArUco marker, (E) Drop point for robot, (F) Tool for interacting with ring.

3.1. Virtual Industrial Task Master-slave Robot dataset

VIT-MR (Virtual Industrial Task Master-slave Robot) dataset, was created to replicate all the steps of robotic assisted tele-operated manipulation process typical in high precision small-scale manufacturing. The user controls remote slave manipulators from Leo master console (BBZ srl, Verona, Italy) visible in Figure 2a, a compact hardware device integrating two maters manipulators, high-definition stereo viewer and foot-pedals tray. The console guarantees an immersive user experience, which enable ergonomic slave manipulators controls and enhanced magnified vision. Simulated slave manipulators, visible in Figure 2b provide high-dexterity and movement scaling to ensure a precise and stable components manipulation during assembly process.

We have implemented simulated environment using a research version of Xron (BBZ srl, Verona, Italy), a realistic virtual simulator suitable for high fidelity applications, such as medical training. We have used this experimental setup since it is able to reproduce kinematic variables very similar to a real master-slave robotic setting in the

industrial field. The manipulation task involves the positioning of a set of colored rings in their correct position on a peg board, as shown in Figure 2b. The exercise consists in lifting a ring with one of the robotic tools, passing the ring on the other robotic arm and positioning it in the corresponding pole.
A group of 17 users without significant experience in robotic assisted manipulation has been enrolled in this study. All the users have no more than one-hour previous experience in using similar robotic systems. Each subject had a time slot of one hour, the first half is dedicated to practicing with the specific platform and the second half is dedicated to the recorded trials. Each subject performed from a minimum of ten to a maximum of twenty trials resulting in a total 256 sequences. Multiple users with different levels of experience allows the model to be robust to the difference in movements execution and to better recognize phases transitions.

Table 1. Datasets features description.

VIT-MR	
Index	Description
0	Process phase
1-15	Rotation and translation Right
16	Gripping Angle Right
17-28	Rotation and translation Left
29-31	Cartesian position Left
32	Gripping angle Left
ICRT	
Index	Description
0	Gesture
1-7	ArUco Marker Cartesian position and orientation in quaternions. The reference point of the system is given by the RealSense camera [17]
8-16	Robot joint angles
17	Timestamp
18-25	End-effector cartesian and orientation
26-114	Leap motion hand feature (87) [18]

Table 2. Datasets gesture description.

Index	VIT-MR	ICRT
0	Hand pick the tool with the ring	Colleting the ring
1	Move the ring with the tool at an arbitrary point	Passing the ring from the right arm to the left arm
2	Release the tool	Posing the ring in the correct pole
3	Robot identify and pick the ring	Failing grabbing the ring
4	Let the robot move the ring at drop point	Failing passing the ring from right arm to left arm
5	Robot release the ring	Failing posing the ring in the correct pole

The dataset includes synchronized stereo images, kinematic variables for the two slave manipulators and other system status variables. For each slave manipulator 16 variables are available: Cartesian position, rotation matrix, and tool gripping angle as reported in Table 1. These variables represent all the raw data offered by the system. We excluded images to avoid data requiring preprocessing. The manually annotated phases are six and are defined in Table 2. They include classification on errors occurring during task execution to better describe user-specific performance. The dataset and related detailed documentation are available at gitlab.com/altairLab/VIT-MR.git.

3.2. Industrial Collaborative Robot-human Task dataset

ICRT (Industrial Collaborative Robot-human Task) dataset involves the use of four devices for the interaction between human operator and a collaborative robot. The sensors used are a Leap Motion device (LeapMotion,US), an ArUco marker [19], an Intel RealSense D415(Intel, US) camera and a Panda robot (Franka Emika, Germany). We decided to simulate an environment containing multiple sensors as in a typical I4 context. Thus, we used data extracted directly from machines (robot kinematics) and data obtained from the supervising sensors such as leap motion and ArUco marker. We have selected a ring to avoid complications due to the manipulation of the object

such as the incorrect grasping by the robot. Using a tool allows us to extend the experiment to cases of assembly of dangerous object or not suitable to human contact. The setup is shown in Figure 2c, highlighting the sensors used and the salient points of the task. The experiment is described as follow: in a first phase, a human operator picks the ring through a tool to avoids direct contact with the hand. Then the user moves it to an arbitrary position and release the tool. The second phase involves the scene segmentation and ring recognition by the RealSense camera sensor and afterwards the robot picks the ring and sub-sequentially place it to the drop point, then it will release the ring. When robot ends the task, it returns to a ready position waiting for a new task to be executed. The entire task will be described by the Leap Motion data, to capture hand movements, by the robot joints position and the end-effector pose to capture the motion of the robot and by the ArUco to track the movements of the ring. These values are labelled and detailed in Table 1. This task has been repeated for 40 times by a single human operator. The dataset and related detailed documentation are available at gitlab.com/altairLab/ICRT.git

4. Evaluation and Result

VIT-MR dataset has been evaluated using the LOUO (Leave One User Out) methodology which consists in excluding a user during the training phase and using it during the test phase. This means executing 18 times the training and the test. For ICRT dataset we used leave k-out methodology (with $k = 5$), that consist leaving 5 trial out in training phase and use them as test. This result in 8 split training since the total number of trials was 40. For both datasets, the values shown represent the average values with standard deviation. We adopt for both dataset macro/micro accuracy as metrics to benchmark the different methods, as described in [20]. In our work classes are called process phases being the classification label predicted by the model. Micro average is computed as the average of total correct predictions across all classes and is preferable for classification on unbalanced classes since it aggregate the contributions of all phases to compute the average metric in terms of precision also considering false positive. On the other hand, macro accuracy is computed as the mean of true positive rates for each class. Table 3 shows the results obtained by applying the methods to the two datasets. The data reported were obtained trying to maximize the micro accuracy.

Table 3. Result for Macro and Micro average accuracy for both datasets reported as mean between LOUO results.

Method	**Dataset**					
	VIT-MR			ICRT		
	Micro Average	Macro Average	Time (s)	Micro Average	Macro Average	Time (s)
TDNN	**69.76**	**53.85**	**1042**	**86.95**	**79.04**	**256.7**
Std	± 6.08	± 3.508		± 3.693	± 4.902	
LSTM	60.86	42.35	34455	65.83	51.98	6367
Std	± 8.60	± 3.50		± 13.18	± 15.07	
RFC	62.16	42.7	753.69	86.92	80.32	81.98
Std	± 8.914	± 3.792		± 3.558	± 5.126	
SVM	54.55	32.11	264.82	41.19	6.865	15.31
Std	± 10.42	± 5.254		±1.72	±0.28	

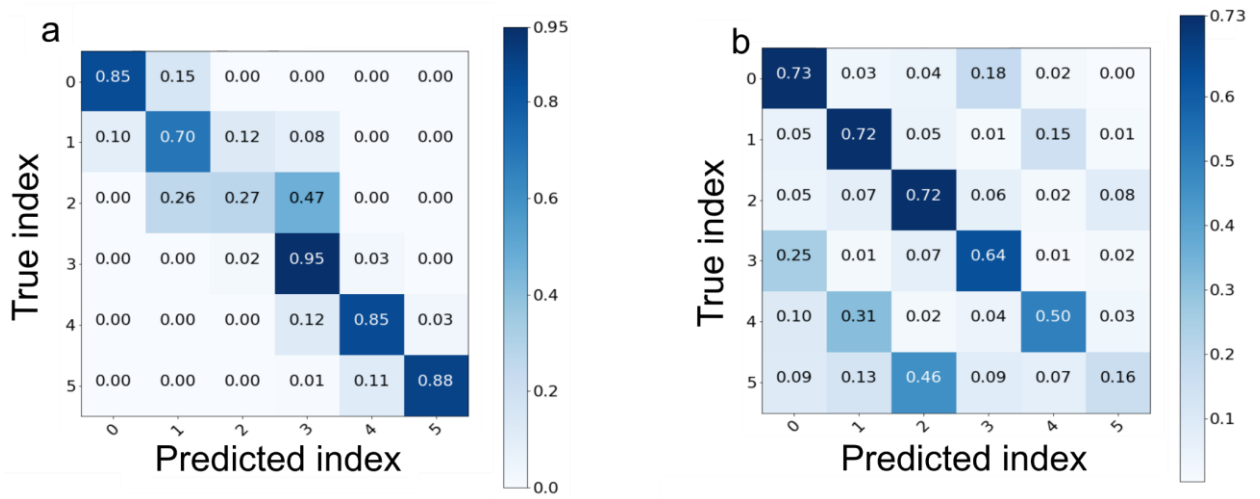

Fig. 3. Normalized confusion matrix for (a) ICRT and (b) VIT-MR datasets. Color indicates the accuracy for each class as represented by the scalar number. Higher color intensity in the diagonal of the matrix correspond to higher accuracy results of the model. Color intensity are normalized with the maximum and minimum values for each dataset. Class indexes are referred in Table 2

5. Discussion and Conclusion

The results in Table 3 show that the TDNN network has excellent ability to adapt to task recognition outperforming the other methods considered. The robustness of the TDNN is demonstrated by the accuracy obtained on both datasets. The substantial difference between micro and macro accuracy for VIT-MR dataset is due to imbalance distribution of phases. This is demonstrated as the difference ranges from a minimum of 16% for TDNN to a maximum of 22% for SVM and thus assuming a significant value for each method presented. This observation does not hold for ICRT dataset, because it provides more balanced phases distribution. Most of the wrong recognitions for the VIT-MR dataset occur between one phase and its corresponding error class. This fact is clearly represented by the normalized confusion matrix showed in Figure 3. Difference of accuracy in the recognition of phases is also given by the type of movement and sensor that more define each phase. The best recognition performances are obtained for robotic movements (Fig 3a last 3 labels) followed by human operator (Fig 3 first 3 labels of both dataset). The proposed method obtains poor recognition performance when used for error recognition, as demonstrate from results in Fig3b last 3 labels of VIT-MR. This suggests that a network trained for APM is not suitable for recognition of errors that need more information, perhaps of a higher level (such as those deriving from the environmental camera). In Table 3 we have also reported the computation times of each method, including the sum of training and prediction time for a single run. This computation time is important to evaluate the ability of different APM method to be applied on novel IMS plants with minimum impact on setup time and thus costs. Although the proposed method obtained the best accuracy performance compared to other methods considered, the absolute values of the results are probably not adequate for real application. This fact could be motivated because the experiment was created to be flexible and generalizable to several processes instead of being accurate. In the current form, the proposed APM method could not be applied for interactive control feedback of an intelligent manufacturing robot. However, we believe that in a collaborative environment an abstract interpretation of the running process phase can still provide significant support both in terms of security and decision-making. We have tested a new type of neural network to model industrial processes with collaborative robots. We have released two new datasets to support objective APM method benchmarking. As future works we aim to extend this work on real industrial case studies and to analyze in detail which kind of sensors the network is most sensitive to. We would also like to train the model at recognizing other process sequences making it independent from the order process executions.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 742671).

References

[1] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, and M. Harnisch, "Industry 4 . 0 : The Future of Productivity and Growth in Manufacturing Industries," 2015.

[2] Z. Bi, S. Member, L. Da Xu, S. Member, C. Wang, and S. Member, "Internet of Things for Enterprise Systems of Modern Manufacturing," vol. 10, no. 2, pp. 1537–1546, 2014.

[3] A. Moeuf, R. Pellerin, S. Lamouri, S. Tamayo-Giraldo, and R. Barbaray, "The industrial management of SMEs in the era of Industry 4.0," *Int. J. Prod. Res.*, vol. 56, no. 3, pp. 1118–1136, 2018.

[4] M. Chalal, X. Boucher, and G. Marques, "Decision support system for servitization of industrial SMEs: a modelling and simulation approach," *J. Decis. Syst.*, vol. 24, no. 4, pp. 355–382, 2015.

[5] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," 2018.

[6] F. Alleau, E. Poisson, C. V. Gaudin, and P. Le Callet, "TDNN with Masked Inputs," 2003.

[7] G. Menegozzo, D. Dall'Alba, C. Zandonà, and P. Fiorini, "Surgical gesture recognition with time delay neural network based on kinematic data," in *2019 International Symposium on Medical Robotics (ISMR)*, 2019, pp. 1–7.

[8] A. WAIBEL, T. HANAZAWA, G. HINTON, K. SHIKANO, and K. J. LANG, "Phoneme Recognition Using Time-Delay Neural Networks," *Readings Speech Recognit.*, pp. 393–404, Jan. 1990.

[9] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," Jun. 2017.

[10] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014.

[11] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data Mining and Analytics in the Process Industry: The Role of Machine Learning," *IEEE Access*, vol. 5, pp. 20590–20616, 2017.

[12] S. Hochreiter and J. Urgen Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[14] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven Soft Sensors in the process industry," *Comput. Chem. Eng.*, vol. 33, no. 4, pp. 795–814, 2009.

[15] Jian-Bo Yang and Chong-Jin Ong, "Determination of Global Minima of Some Common Validation Functions in Support Vector Machine," *IEEE Trans. Neural Networks*, vol. 22, no. 4, pp. 654–659, Apr. 2011.

[16] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?," *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, 2014.

[17] "GitHub - pal-robotics/aruco_ros: Software package and ROS wrappers of the Aruco Augmented Reality marker detector library." [Online]. Available: https://github.com/pal-robotics/aruco_ros. [Accessed: 20-May-2019].

[18] "GitHub - ros-drivers/leap_motion: Leap Motion ROS integration." [Online]. Available: https://github.com/ros-drivers/leap_motion. [Accessed: 14-May-2019].

[19] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[20] N. Ahmidi *et al.*, "A Dataset and Benchmarks for Segmentation and Recognition of Gestures in Robotic Surgery," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 9, pp. 2025–2041, Sep. 2017.