An Approach for adapting a Cobot Workstation to Human Operator within a Deep Learning Camera

Olatz De Miguel Lázaro, Wael M. Mohammed, Borja Ramis Ferrer, Ronal Bejarano and Jose L. Martinez Lastra
FAST-Lab, Faculty of Engineering and Natural Sciences, Tampere University
P. O. Box. 600, FI-33014 Tampere, Finland
{olatz.demiguellazaro, wael.mohammed, borja.ramisferrer, ronal.bejarano, jose.martinezlastra}@tuni.fi

***Abstract*— One of the major objectives of international projects in the field of Industrial Automation is to achieve a proper and safe human-robot collaboration. This will permit the coexistence of both humans and robots at factory shop floors, where each one has a clear role along the industrial processes. It's a matter of fact that machines, including robots, have specific features that determine the kind of operation(s) that they can perform better. Similarly, human operators have a set of skills and knowledge that permits them to accomplish their tasks at work. This article proposes the adaptation of robots to the skills of human operators in order to implement an efficient, safe and comfortable synergy between robots and humans that are working at the same workspace. As a representative case of study, this research work describes an approach for adapting a cobot workstation to human operators within an installed deep learning camera on the cobot. First, the camera is used to recognize the human operator that collaborates with the robot. Then, the corresponding profile is processed and serves as an input to a module in charge of adapting specific features of the robot. In this manner, the robot can adapt e.g., to the speed of operation according to the skills of the worker or deliver parts to be manipulated according to the handedness of the human worker. In addition, the deep learning camera is used for stopping the process at any time that the worked leaves unexpectedly the workstation.**

Keywords*— *deep learning, human robot collaboration, cobots, face recognition, pose detection

I. Introduction

Manufacturing systems, as presented by the ANSI/ISA-95 standard, comprises of the Enterprise Resources Planning (ERP) at level 4, Manufacturing Execution Systems (MES) at level 3 and Factory shop floor at levels 0, 1, and 2. This hierarchical representation considers resources at the factory shop floor, where human operators work. In this regard, the MESA [1] defines the MES as set of 11 functions [2] which can be mapped to other functions from different organizations [3]. This representation devotes a function for human management, titled labor management. Besides, the human operator can be involved in other functions like resource allocation and status function. This reflects the importance of keeping the human in the loop of manufacturing systems. Currently, and as part of keeping the human in the loop, the Human-Machine Collaboration (HMC) is introduced. Moreover, the European Commission under Factories of Future (FoF) program has opened several topics for possible fund to support the Human-Robot Collaboration (HRC) research [4].

The HRC addresses the shared tasks between humans and robots that need to be executed in parallel at the same workstation. This motivates researchers to create methodologies and approaches for modelling and understating this collaboration, which in return, enriches the industrial research and increases the occupational safety in general.

In this context, the objective of this research work is to present an approach that permits collaborative robots (cobots) to adapt to the working environment, and more importantly, to the operator they are sharing tasks with. The presented approach proposes the use of the advances of deep learning algorithm for recognizing the operator. By adapting the operation of a cobot to the specific work characteristics of the human worker, such as speed or height, the safety and comfort of the human is increased, thus creating a more efficient and productive environment. Deep learning provides a fast and powerful tool to implement a face recognition model that recognizes the operator in the workstation and adapts the cobot's operations accordingly.

In addition, the article includes a use case aiming at highlighting a prototype to validate the proposed approach. It is important to highlight that this paper presents a preliminary stage of the research work, hence, some of the architecture may change in the future. Still, the contribution is significant enough to present it and point out the direction of the research.

The rest of the document is structured as follows: Section II presents the related research and state of the art in the field of HRC and deep learning for industrial uses. Section III presents the approach of this research. Section IV provides the implementation of the presented approach. Finally, Section V concludes the paper and provides possible future work.

II. Related Work

A. Human Robot Collaboration for Manufacturing

The human role at the factory shop floor has been evolving due to the evolution of the automation levels in manufacturing systems [5]. Currently, the human is expected to work with dissimilar resources at the factory shop floor, such as robots, which raises the occupational safety measures and requires scheduling [6], [7] . Due to this, research centers and technology providers tend to include the human in the manufacturing loop rather than increasing the automation level which, in some cases, can be complicated and expensive [8]. Thus, the concept of keeping humans in the loop has been introduced in different fields and levels. As an example, Human Robot Collaboration (HRC), which represents the parallelism in the execution of shared tasks, and Human Robot Interaction (HRI), which represents humans and robots sharing tasks and information [9].

The HRC advances in the concept and the need by the market induced robot manufacturers to introduce the collaborative robots (cobots). Some of the most known commercial cobots are: the ABB YuMi [10], Yaksawa MOTOMAN SDA [11] and the KUKA LBR iiwa [12]

. Mainly, cobots permit safe interaction with human operators while working at the same location [13]. However, and due to their safety restriction and physical nature, cobots do not provide high payload and fast manipulation [8]. This restricts the usage of the cobots to tasks where human and robot need to work together at the same time [14].

The deployment of cobots at factory shop floors brought the challenge of adaptability since the operators can be changed during the production cycle. In other words, the cobots need to adapt to the physical skills and/or properties of the operator such as height and to the habits of the operator such as the dominant hand [15]. These challenges derived research works to conduct tests on the involvement of cobots at industrial environments in order to demonstrate their true potential. As an example, Iñaki Maurtua et al. presented in [16] a test where cobots have been tested against several measures like safety, trustworthiness, usability and productivity among others. As a result, the flexibility and adaptability aspect ranked third with almost 55% after safety and usability in first and second. In order to enhance the adaptability, a human recognition model needs to be included to allow the cobot to understand the nature and the profile of the worker [17].

B. *Deep Learning for Industrial Applications*

The Machine Learning (ML) technologies study the possibility of permitting digital computing units to train and evolve in order to support the decision-making process. Historically, the term of ML was introduced by Arthur Samuel in 1959 [18]. At that time, the computers were not prepared to provide the capabilities that are needed for ML implementation. Since then, the research on the ML field has been driven by the evolution in the Computer Science field. In such scope, several techniques and approaches have been applied such as Artificial Neural Networks (ANN), Supported Vector Machines (SVM) and Random Forests (RF). Accordingly, several application fields, like healthcare, urban development, social and industrial, have been benefiting from these approaches [19], [20], [21].

Deep Learning (DL) is considered as a branch of ML where the multi-hidden layers concept of the learning model is applied [22]. In addition, DL can support both the supervised [23] and the unsupervised learning methods [24]. Mainly, DL is applied in applications that require complex and large models, which, in turn, require high computational resources. For instance, the YOLOv3 algorithm is used for objects detection [25] and Hidden Markov Model (HMM) for acoustics recognition, which is applied for voice recognition [26].

DL is an innovative topic in the industrial applications field. Several research works have been conducted in order to provide high-level intellect to production systems. As an example, the usage of DL in predictive maintenance allows the detection of patterns in the collected data that might trigger maintenance issues [27]. Another application appears in the monitoring and performance analysis. This comprises of collecting big data from the factory shop floor and, then, processing it in order to extract patterns that represent the production systems [28]. Accordingly, the extracted patterns can be represented as Key Performance Indicators (KPIs) to support the decision-making at the Enterprise Resource Planning (ERP) level. In addition, the DL is intensively employed for optimizing manufacturing plans targeting e.g., production, resource allocation, logistics, in order to maximize the outcome of these systems with respect to the available resources [29] [30] [31].

Similarly, DL is applied in the HRC in order to provide cognition of the ambient of the collaboration scene. More precisely, and as presented in [32], three models were tested in order to choose better one based on performance. The first model is the Multi-Layer Perceptron (MLP), the second model is Convolution Neural Network (CNN) and the third model is Long Short-Term Memory (LSTM) networks. According to the authors, the selected model was the MLP for body posture recognition and CNN for voice recognition. As a result, the authors claimed a potential usage of the DL in HRC that can be converted to a ready application for industrial cases. Another research, as presented in [33], presents the potential benefits of the usage of ML and DL in industrial applications in general. The authors anticipate that the ML and DL in the HRI can increase the occupational safety.

III. The Approach

The goal of this paper is to present an approach for implementing DL techniques in a HRC system by using a DL camera.

HRC applications require extensive flexibility to vary the programmed tasks in order to collaborate with a human operator in workstations. A great amount of programming, trial and error are necessary for those kinds of operations, which derives in an increase on time and cost of programming. Traditional programming requires the operator to propose a solution and write a precise program that the robot can execute to automate its tasks. For this reason, traditional programming is not enough to meet the needs of HRC systems. With ML systems, input data is collected and the desired target values are defined and the ML model will find a program that fits the data. This allows a more flexible solution to complex problems, especially those that are too complicated for humans to solve [34].

Effective communication between humans and robots is an essential part of HRC. By enabling computer vision in HRC systems, the robots can gather information from their environment and respond accordingly. This allows the human operators involved in the HRC to communicate using gestures or poses [35].

The objective of this implementation is to create a DL model that is able to detect and recognize faces, so that the information can be transmitted to the collaborative robot's controller. The model will also be able to recognize whether the human worker is ready to perform the task or not by detecting the orientation of their head, to identify whether the operator is facing the workstation or not.

The camera used during the implementation is an AWS DeepLens camera [36]. AWS DeepLens is a fully programmable video camera and development platform integrated with the Amazon Web Services Cloud [37]. It allows the developer to run deep learning models locally. The camera captures images and feeds ANN models in order to achieve computer vision.

This device provides real-time processing of scenes, composed of both video feed and sound. The workflow of a DeepLens project is shown in Fig. 1. The camera receives a video stream as an input and produces two output streams: the device stream, which is not processed, and the project stream, which comprises the processed frames of the input video.

The AWS DeepLens is an AWS IoT Greengrass core device, where the AWS Lambda functions run. The Inference Lambda function gathers image frames from the captured video stream and sends them to the ANN model, which has been trained by ML software. The software used to train the model can be Amazon's own model training service, SageMaker, or other ML framework like Apache MXNet, TensorFlow or Caffe. The model runs the convolutional neural network on each frame and sends them back to the Lambda function, where they are passed on in the project stream.

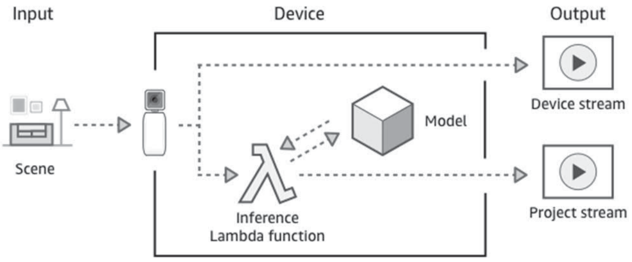

Fig. 1 Basic workflow of an AWS DeepLens project *[37]*

The output of the Inference Lambda function is a JSON payload that is published to an AWS IoT MQTT topic, as seen in Fig. 2. Once the payload is published, it is evaluated by a Lambda function and it can be viewed through the management console.

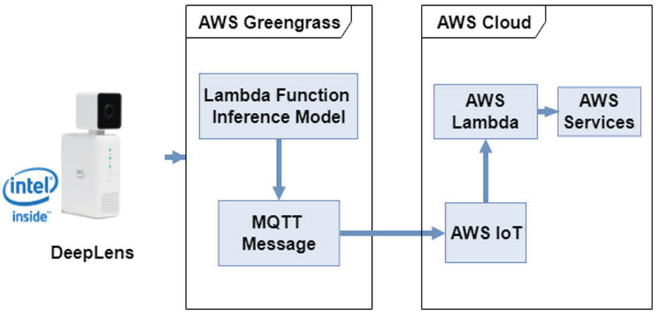

Fig. 2. DeepLens architecture [38]

The web browser is the interface between the developer and the AWS DeepLens device, and it is used to create and deploy the deep learning projects. An AWS DeepLens project is composed of both the deep learning models and the inference Lambda functions.

To create a custom project, a DL model may be created and trained using Amazon SageMaker or another of the supported ML environments. The model is imported into the AWS DeepLens. Then, the Inference Lambda function is created and published in AWS Lambda. Afterwards, the AWS DeepLens project is created and both the model and the function are added to it. Once the project is created, it can be deployed to the DeepLens device. It is important to mention that developing a neural network implies a great amount of time and effort, and requires thousands of images in order to train the model. However, the advantage of the DeepLens device is that it also comes with a series of sample projects, whose models have been pre-trained.

On the other hand, Fig. 3 shows the steps to follow in order to create and deploy the sample projects. All steps are to be done in the web browser. Moreover, the functionalities of the sample projects can be extended so that they perform a specific job. To extend those functionalities, the models can be trained and edited and the Inference Lambda functions can be configured.

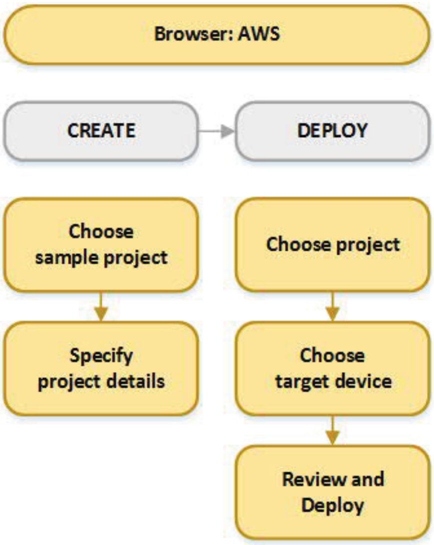

Fig. 3. Steps to create and deploy a sample project [39]

IV. Enhancing UnMi with DeepLearning for HRC

This section provides an explanation on how to adapt the deep learning models discussed in the previous section into an industrial scenario where a collaborative robot (cobot) is in use. DL can be used to enhance said robot's performance and productivity [32] and increase the flexibility of the HRC system. This section will present the industrial scenario in which the camera will be implemented and explain how the modelled deep learning algorithms facilitate the operations of both the robot and the human worker in the workstation.

The AWS DeepLens camera will be implemented in a YuMi IRB 14000 robot [40]. This ABB robot is a dual-arm collaborative robot designed for small parts assembly processes. For this implementation, once the deep learning projects are created and deployed, the AWS DeepLens deep learning camera will be mounted on top of the robot, in resemblance to a human head.

The workstation, depicted in Fig. 5, consists of one YuMi robot standing opposite an operator. The process that takes place in the workstation is a box assembly process. The wooden box is made of six sides that are bolted together. All parts necessary for the assembly are located on a table between the robot and the operator. The robot is responsible for holding the sides of the box while the operator fastens the bolts to attach them.

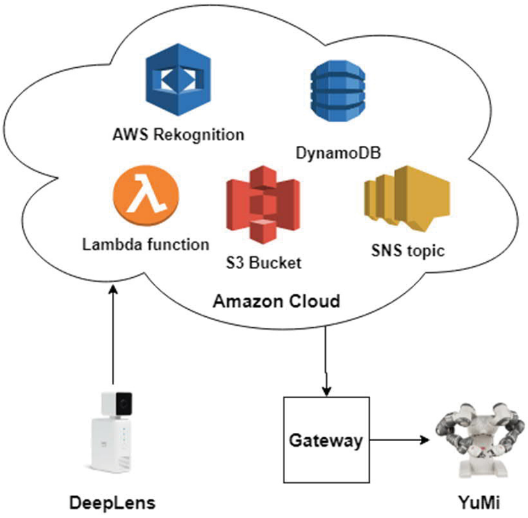

Fig. 4. Overview of the project

The objective of the abovementioned process is to mock a real industrial process. See Fig. 4. However, in a real industrial process, there will probably be more than one operator that works in the same workstation during different shifts. For this reason, the work characteristics of the robot should change to adapt to the operator's work characteristics; those characteristics being, for example, the speed in which they work, their height or whether they are right- or left-handed. Furthermore, HRC systems need to guarantee the safety of the human worker. The collaborative robot should be able to notice when the worker is not ready to work and stop the operation to avoid any accidents.

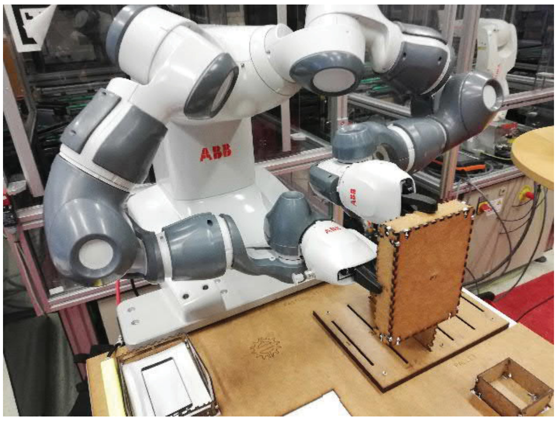

Fig. 5. YuMi robot performing box assembly process

With the video camera, YuMi detects which of the human workers is in the workstation at any moment, and sends a message to the robot's controller to adjust its operations to accommodate to the user's profile. This flexibility allows the workers to work at their own pace, reducing stress and risk of accidents on the workplace. In addition, there are instances in which the worker might be in the workstation but they are not focused on the task. In those moments, the robot should stop moving and wait until the operator is ready to continue their work. To enable the communication between the Amazon Cloud services and the robot, there needs to be a gateway. This gateway listens for incoming messages from the Amazon Cloud and transmits the information back to the robot's controller.

For this application, a pre-trained DL model from one of DeepLens's sample projects is used: the face detection model. This model gathers the video stream as an input and determines when there is a face in the feed. Then, it sends the feed frame containing the face to a Lambda function, along with a percentage of certainty. The model has a Single Shot Detector (SSD) architecture with a ResNet-50 feature extractor, and it was trained in Apache MXNet DL framework.

In the approach, it was mentioned that the main components of a DeepLens project were the DL model and a Lambda function. Aside from those, there are other Amazon Web Services that are used for the implementation:

- S3 Bucket: Cloud storage service in which images can be uploaded and accessed. The bucket will be used to store both the face images obtained from the video feed and the photographs of all operators working with YuMi.
- AWS Rekognition: This is a facial recognition cloud service that is capable of detecting and extracting data from a face in an image and of comparing faces in different images. The operators' photographs in the S3 bucket are uploaded to a Rekognition collection and assigned a unique ID.
- DynamoDB: In a database table the operator's name and their average working speed are stored. Each item will also have a unique ID, equal to the ID of the operator's photograph in the Rekognition collection, which will be used to link the worker's image to their data.
- Simple Notification Service (SNS): Amazon's messaging service. Lambda functions are able to publish messages in a SNS topic. A gateway will receive the information posted by the Lambda function and transmit it to YuMi.

The overview of the DeepLens implementation can be seen in Fig. 6

The deeplens-face-detection Inference Lambda function runs the DL face detection model in the DeepLens device, illustrated as step 1 in Fig. 6, and the model searches for faces in the video feed. When a face is found with a certainty higher than 85% (step 2), the face frame is uploaded to an Amazon S3 bucket, as can be seen in step 3 in Fig. 6. After detecting a face and uploading it to the S3 bucket, the Lambda function waits for 15 seconds before checking the feed again. This way, it allows the rest of the services to be performed without overwhelming the system by uploading too many images to the bucket.

The S3 bucket has two folders: a "DeepLens" folder, in which the faces found in the video feed are stored, and a "Rekognition-Images" folder, where the pictures of all the operators are uploaded. This last folder will be used to send the faces of the operators to the Rekognition collection.

A second Lambda function is triggered when a new image is uploaded to the "DeepLens" folder in the S3 bucket (step 4). In step 5, the "face-analysis-function" Lambda function calls the AWS Rekognition API to compare the face in the new image with the reference faces that are stored in the Rekognition collection. If it finds a match in the collection, the Lambda function will look for the operator's details in the DynamoDB database, as illustrated during step 7, using the unique Rekognition ID assigned to the operator. The Lambda function will extract all the attributes assigned to the operator in the database in JSON format. The AWS Rekognition API will also indicate the orientation of the face by determining its pitch, roll and yaw angles. The Lambda function will check the angles to establish if the operator is looking towards the workstation or not.

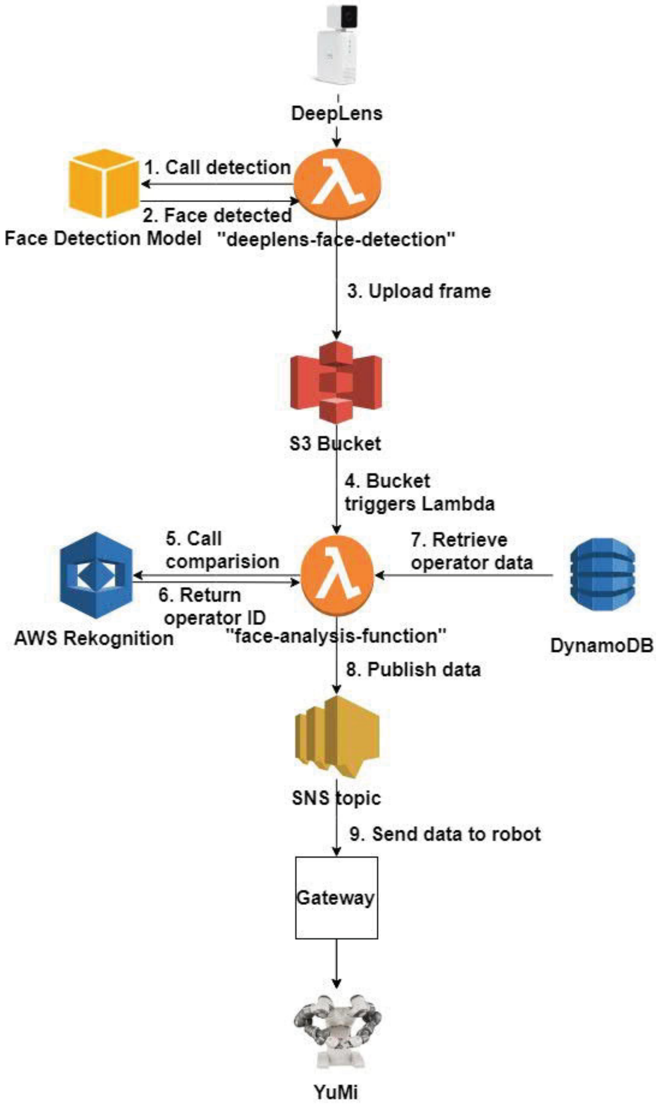

Fig. 6. Workflow of the DeepLens implementation

All information provided by AWS Rekognition and the database will be published in a message to a SNS topic (step 8). The message will contain all the operator's information that is available in the database, along with an indication to stop the movements of the robot if the operator is looking away from the workstation. A gateway will receive the message and send the value of the needed speed to the robot controller (step 9).

The face detection model and the "deeplens-face-detection" inference Lambda function run on the DeepLens device as part of the AWS IoT Greengrass core software in the device. The rest of the services and resources are cloud-based and can be accessed through the AWS Management Console.

With the face recognition project, the collaborative robot is able to adapt to the way of working of each of the human operators whose working details are stored in the worker's database. YuMi's movement speed will change to match the one of the human operators, as well as the height in which the assembly is done, and it will pause when it is indicated. This way, accidents in the workstation are avoided and the efficiency of the tasks is improved. The communication between robot and worker is simpler, since operators will not need to input their own work requirements before beginning the process. The DL model will be the one responsible for those operations.

It should be noted that deep learning can be used for more than facial recognition and pose detection to aid Human-Robot Collaboration. In an industrial environment, deep learning can help the programming of a robot by using imitation learning. In this type of machine learning, human knowledge can be transferred to a machine by means of demonstration. The human will show the robot how to perform a certain task a few times and the robot will learn how to perform it by imitation.

Furthermore, DL techniques can also be used to teach the robot to grip objects. In scenarios where objects with varying shapes are handled, computer vision can be used to analyze the shape and appearance of an object and process how the robot can grab it.

V. Conclusion

This article introduces the use of deep learning techniques within a Human-Robot Collaboration system. This was illustrated in an industrial scenario where a YuMi collaborative robot was enhanced with a deep learning camera. Further work will include testing of the system during the assembly process, to calculate the rate of success of the algorithm and the time that it takes from the detection of the worker until the new operation values are applied on the robot. Aside from the models discussed in the article, the deep learning algorithms might be extended by adding voice commands or hand motions to control the robot; for example, the operator could raise a hand to indicate when the robot should stop its operations.

The use of deep learning techniques to aid collaborative work results in a reduction of time and costs derived from programming. Deep learning enables a better, more natural communication between the robot and the human worker. Combining the best capabilities of both machines and human operators enhances their labor, and productivity and efficiency on their operations is increased. Once the implementation is finished, operators will be able to work comfortably at their own pace in the workstation.

REFERENCES

[1] "MESA International," 18-Jun-2019. [Online]. Available: http://www.mesa.org/en/index.asp. [Accessed: 26-Jun-2019].

[2] W. M. Mohammed *et al.*, "Generic platform for manufacturing execution system functions in knowledge-driven manufacturing systems," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 3, pp. 262–274, Mar. 2018.

[3] S. Iarovyi, W. M. Mohammed, A. Lobov, B. R. Ferrer, and J. L. M. Lastra, "Cyber–Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1142–1154, May 2016.

[4] "Factories of the Future (FoF) - Research & Innovation - Key Enabling Technologies - European Commission." [Online]. Available: http://ec.europa.eu/research/industrial_technologies/factories-of-the-future_en.html. [Accessed: 26-Jun-2019].

[5] Å. Fasth, J. Stahre, and K. Dencker, "Level of automation analysis in manufacturing systems," in *Advances in Human Factors, Ergonomics, and Safety in Manufacturing and Service Industries*, W. Karwowski and G. Salvendy, Eds. CRC Press, 2010, pp. 233–242.

[6] D. M. D'Addona, F. Bracco, A. Bettoni, N. Nishino, E. Carpanzano, and A. A. Bruzzone, "Adaptive automation and human factors in manufacturing: An experimental assessment for a cognitive approach," *CIRP Annals*, vol. 67, no. 1, pp. 455–458, Jan. 2018.

[7] B. R. Ferrer, W. M. Mohammed, A. Lobov, A. M. Galera, and J. L. M. Lastra, "Including human tasks as semantic resources in manufacturing ontology models," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 3466–3473.

[8] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris, "Design Considerations for Safe Human-robot Collaborative Workplaces," *Procedia CIRP*, vol. 37, pp. 248–253, 2015.

[9] F. Ranz, T. Komenda, G. Reisinger, P. Hold, V. Hummel, and W. Sihn, "A Morphology of Human Robot Collaboration Systems for Industrial Assembly," *Procedia CIRP*, vol. 72, pp. 99–104, Jan. 2018.

[10] "ABB's Collaborative Robot -YuMi - Industrial Robots From ABB Robotics." [Online]. Available: https://new.abb.com/products/robotics/industrial-robots/irb-14000-yumi. [Accessed: 26-Jun-2019].

[11] "SDA - Yaskawa Europe GmbH." [Online]. Available: https://www.yaskawa.eu.com/en/products/robotics/motoman-%20robots/seriesdetail/serie/sda/. [Accessed: 26-Jun-2019].

[12] "LBR iiwa," *KUKA AG*. [Online]. Available: https://www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa. [Accessed: 26-Jun-2019].

[13] A. Hussnain, B. R. Ferrer, and J. L. M. Lastra, "An application of Cloud Robotics for enhancing the Flexibility of Robotic Cells at Factory Shop Floors," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 2963–2970.

[14] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, "Collaborative manufacturing with physical human–robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 40, pp. 1–13, Aug. 2016.

[15] R. M. del Toro, M. C. Schmittdiel, R. E. Haber-Guerra, and R. Haber-Haber, "System Identification of the High Performance Drilling Process for Network-Based Control," pp. 827–834, Jan. 2007.

[16] I. Maurtua, A. Ibarguren, J. Kildal, L. Susperregi, and B. Sierra, "Human–robot collaboration in industrial applications: Safety, interaction and trust," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, p. 172988141771601, Jul. 2017.

[17] J. C. Mateus, D. Claeys, V. Limère, J. Cottyn, and E.-H. Aghezzaf, "A structured methodology for the design of a human-robot collaborative assembly workplace," *The International Journal of Advanced Manufacturing Technology*, Feb. 2019.

[18] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," p. 21, 1959.

[19] J. L. Seixas, S. Barbon, and R. G. Mantovani, "Pattern Recognition of Lower Member Skin Ulcers in Medical Images with Machine Learning Algorithms," in *2015 IEEE 28th International Symposium on Computer-Based Medical Systems*, 2015, pp. 50–53.

[20] J. Oh, "Finding Main Streets: Applying Machine Learning to Urban Design Planning," p. 9.

[21] A. Shah, P. Belyaev, B. R. Ferrer, W. M. Mohammed, and J. L. M. Lastra, "Processing mobility traces for activity recognition in smart cities," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 8654–8661.

[22] A. Gajate, R. E. Haber, J. R. Alique, and P. I. Vega, "Transductive-Weighted Neuro-Fuzzy Inference System for Tool Wear Prediction in a Turning Process," in *Hybrid Artificial Intelligence Systems*, 2009, pp. 113–120.

[23] W. Gu, X. Xu, and J. Yang, "Path Following with Supervised Deep Reinforcement Learning," in *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, 2017, pp. 448–452.

[24] Q. Li, J. Zhao, and X. Zhu, "An Unsupervised Learning Algorithm for Intelligent Image Analysis," in *2006 9th International Conference on Control, Automation, Robotics and Vision*, 2006, pp. 1–5.

[25] P. Tumas and A. Serackis, "Automated Image Annotation based on YOLOv3," in *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, 2018, pp. 1–3.

[26] L. Cuiling, "English Speech Recognition Method Based on Hidden Markov Model," in *2016 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, 2016, pp. 94–97.

[27] H. Yan, J. Wan, C. Zhang, S. Tang, Q. Hua, and Z. Wang, "Industrial Big Data Analytics for Prediction of Remaining Useful Life Based on Deep Learning," *IEEE Access*, vol. 6, pp. 17190–17197, 2018.

[28] X. Xu and Q. Hua, "Industrial Big Data Analysis in Smart Factory: Current Status and Research Strategies," *IEEE Access*, vol. 5, pp. 17543–17551, 2017.

[29] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, Jul. 2018.

[30] T. Yang, Y. Hu, M. C. Gursoy, A. Schmeink, and R. Mathar, "Deep Reinforcement Learning based Resource Allocation in Low Latency Edge Computing Networks," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 2018, pp. 1–5.

[31] C. Zhang, L. Xu, X. Li, and H. Wang, "A Method of Fault Diagnosis for Rotary Equipment Based on Deep Learning," in *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, 2018, pp. 958–962.

[32] H. Liu, T. Fang, T. Zhou, Y. Wang, and L. Wang, "Deep Learning-based Multimodal Control Interface for Human-Robot Collaboration," *Procedia CIRP*, vol. 72, pp. 3–8, Jan. 2018.

[33] M. Zamora, E. Caldwell, J. Garcia-Rodriguez, J. Azorin-Lopez, and M. Cazorla, "Machine Learning Improves Human-Robot Interaction in Productive Environments: A Review," in *Advances in Computational Intelligence*, 2017, pp. 283–293.

[34] "Differences between machine learning and software engineering — Futurice." [Online]. Available: https://www.futurice.com/blog/differences-between-machine-learning-and-software-engineering/. [Accessed: 13-Feb-2019].

[35] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, Nov. 2018.

[36] "AWS DeepLens – Deep learning enabled video camera for developers - AWS," *Amazon Web Services, Inc.* [Online]. Available: https://aws.amazon.com/deeplens/. [Accessed: 30-Jan-2019].

[37] "What Is AWS DeepLens? - AWS DeepLens." [Online]. Available: https://docs.aws.amazon.com/deeplens/latest/dg/what-is-deeplens.html. [Accessed: 30-Jan-2019].

[38] J. Chen, "AWS DeepLens & SAP apps integration (HANA & SAC)," *Jeff Chen*, 03-Jan-2019. .

[39] "Creating and Deploying an AWS DeepLens Sample Project - AWS DeepLens." [Online]. Available: https://docs.aws.amazon.com/deeplens/latest/dg/deeplens-create-deploy-sample-project.html. [Accessed: 13-Feb-2019].

[40] "ABB's Dual-Arm Collaborative Robot - Industrial Robots From ABB Robotics." [Online]. Available: https://new.abb.com/products/robotics/industrial-robots/yumi. [Accessed: 30-Jan-2019].