Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 48 (2020) 926–931

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

48th SME North American Manufacturing Research Conference, NAMRC 48 (Cancelled due to COVID-19)

Real-Time Assembly Operation Recognition with Fog Computing and Transfer Learning for Human-Centered Intelligent Manufacturing

Wenjin Tao[a,*], Md Al-Amin[b], Haodong Chen[a], Ming C. Leu[a], Zhaozheng Yin[c], Ruwen Qin[b]

[a]*Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA*
[b]*Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA*
[c]*Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA*

Abstract

In a human-centered intelligent manufacturing system, every element is to assist the operator in achieving the optimal operational performance. The primary task of developing such a human-centered system is to accurately understand human behavior. In this paper, we propose a fog computing framework for assembly operation recognition, which brings computing power close to the data source in order to achieve real-time recognition. For data collection, the operator's activity is captured using visual cameras from different perspectives. For operation recognition, instead of directly building and training a deep learning model from scratch, which needs a huge amount of data, transfer learning is applied to transfer the learning abilities to our application. A worker assembly operation dataset is established, which at present contains 10 sequential operations in an assembly task of installing a desktop CNC machine. The developed transfer learning model is evaluated on this dataset and achieves a recognition accuracy of 95% in the testing experiments.

Keywords: Intelligent Manufacturing; Smart Manufacturing; Fog Computing; Artificial Intelligence; Operation Recognition.

1. Introduction

Artificial intelligence technologies have been providing more and more possibilities, such as cyber-physical manufacturing [11] and industrial digital twin techniques [7] to traditional manufacturing industries. A human-centered intelligent manufacturing system emphasizes human on the factory floor, i.e., every element in the system is to assist the operator in achieving the optimal operational results [18]. To develop such human-centered systems, the primary task is to accurately understand human behavior. However, recognizing human activity on the factory floor is challenging because it involves some complex behaviors, such as operations in an assembly task, which may contain fine-grained hand movements and is difficult to model and analyze.

A variety of methods have been developed to understand human behavior. Convolutional neural networks (CNN) were used to recognize complex hand gestures with captured images [19, 16]. Hu et al. [8] used sEMG (surface electromyography) sensing signals for hand pose recognition. In the manufacturing area, research work has been performed including the follows. Al-Amin et al. developed a sensor data based worker activity recognition model using depth images for workforce management [1]. Haslgrübler et al. conducted human activity recognition with multi-sensor fusion in harsh environments for industrial assistance systems [5]. Azadi et al. analyzed the feasibility of unsupervised industrial activity recognition based on a frequent micro action [3]. Tao et al. [17, 20] proposed a multimodal approach based on CNN for recognizing 6 worker activities to augment the perception of each individual modality and have a more comprehensive understanding. Recently, deep learning methods have been increasingly popular for various applications [10]. However, it needs a large amount of data to train a deep learning model, which is time-consuming and costly to collect. For a small dataset, transfer learning has been demonstrated to be an effective and efficient approach to transfer learning abilities from pre-trained source models to target models [14].

In this paper, we aim to develop a real-time application for assembly operation recognition using image frames ob-

* Corresponding author. Tel.: +1-573-466-3528; fax: +1-573-341-6512.
E-mail address: w.tao@mst.edu (Wenjin Tao).

2351-9789
10.1016/j.promfg.2020.05.131

tained from a visual camera by leveraging artificial intelligence approaches. To achieve real-time recognition, fog computing technique is introduced, which is an emerging technique that brings computing power close to data sources. It can reduce the latency and cost of delivering data to a remote cloud server [2, 12].

The remainder of this paper is organized as follows. Section 2 explains the proposed methodology, including the framework design, how we define the assembly task, data preparation, and the deep learning approach. The experimental setups and results are described in Sections 3. Finally, Section 4 provides the conclusion and future work.

2. Methodology

2.1. The Proposed Fog Computing Framework

Considering that Internet of Things (IoT) devices do not have enough computing power while cloud solutions are not flexible and may cause latency and privacy issues, we develop a framework of fog computing which runs on a local network on the factory floor. An overview of our framework is illustrated in Figure 1. In the sensing layer, we use multiple cameras to capture the operator's activity at the assembly site. Each camera is connected to a small single-board computer Raspberry Pi, where a video streaming service is served. Thus, image frames captured from each camera is published via a certain network port. In the fog layer, workstations with more computing power are connected to the same local network, through which the streaming images can be accessed. Artificial intelligence computations, such as those for training deep learning models, are implemented in this layer.

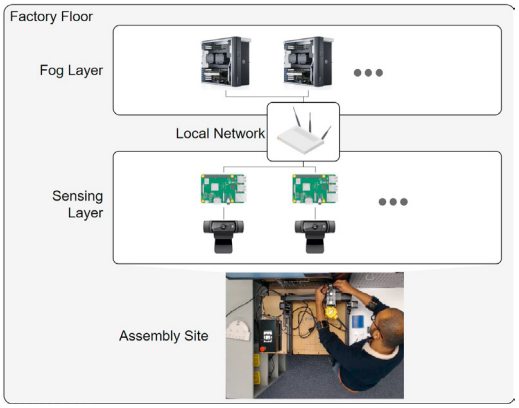

Fig. 1: Overview of our fog computing framework.

2.2. Assembly Task

In this study, we choose a task of assembling a desktop CNC carving machine. The goal of this task is to finish the product assembly with the provided parts, sub-assemblies and tools following installing instructions. This task contains 10 sequential operations, which are: assemble motor module (O1), position spindle mount (O2), install lead screw (O3), fix spindle mount (O4), insert spindle motor (O5), install controller box (O6), connect motor cable (O7), insert power cable (O8), install part (O9), and turn on switch (O10). These 10 operations are illustrated in Figure 2. An image of the final product of the CNC carving machine is shown at the bottom of this figure.

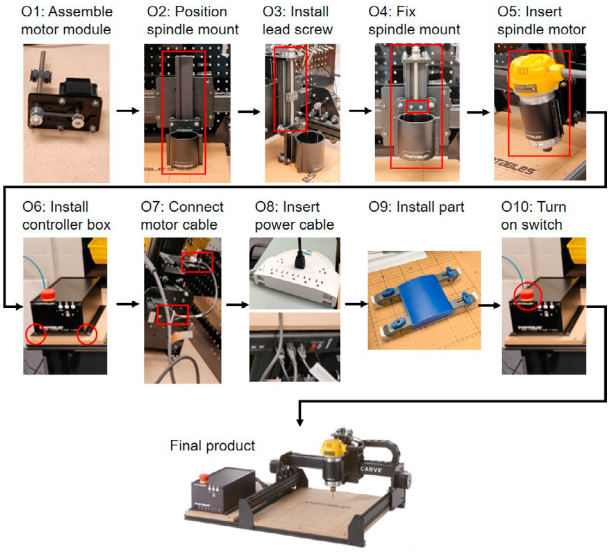

Fig. 2: Illustration of the assembly task containing 10 operations from O1 to O10.

2.3. Sensing and Data Collection

As discussed in Section 2.1, multiple cameras can be used to capture the operator's activity from different perspectives. At present, as shown in Figure 3, two cameras (a top camera and a side camera) of Logitech C920 are used in this system, with an image resolution of 1920 × 1080 and a frame rate of 30 fps. During data collection, the subject is asked to stand in front of the workbench, and perform the tasks with hands in the working area in a natural way. The image data are collected during the operations and the task videos are saved to the disk. Screenshots of the 10 operations are shown in Figure 4, which are taken from the top camera. For annotation purposes, each frame of a video has its frame index on the upper-left corner, and its corresponding timestamp is saved separately in another file.

2.4. Data Preprocessing

In the current study, we choose images captured from the top camera to recognize the operation of the worker because it can cover all the worker activities and the product states. The frames are extracted from the recorded videos. Firstly, a region of interest (ROI) is cropped from an original frame to remove

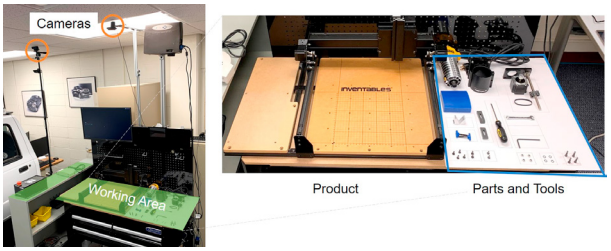

Fig. 3: Illustration of the data collection setup.

the uninformative areas. Since the pre-trained models we use are trained on the ImageNet dataset where each color channel was normalized separately, we implement the same preprocessing transforms as the pre-trained model on our collected data, i.e., normalize the means and standard deviations.

2.5. Transfer Learning and Customized Classifier

Transfer learning can transfer the learned knowledge from a source domain to a target domain, which has been applied in many fields. The general architecture of the transfer learning model is illustrated in Figure 5. Usually, the source dataset contains a large amount of annotated data, with which a deep learning model is trained. For example, a CNN model has a stack of convolutional layers to extract the most discriminative features layer after layer, and a stack of dense layers is used to bridge the extracted features and the source labels. After the source model is trained, a portion of its architecture along with the trained weights is frozen and transferred to a target domain.

For the target model, a new classifier, usually a stack of dense layers, is needed to adapt the source model to the target labels. As shown in Figure 6, the input layer here is essentially the output layer of the transferred model, and the output layer here is set according to the target labels. Then, the hidden layers between them need to be designed in order to have optimal performance.

3. Experimental Study

3.1. Dataset Analysis

To validate the proposed approach, we establish an assembly operation dataset, which has 10 classes of operations as discussed in Section 2.2. The subject is asked to repeat the same assembly task for 10 times. There are 10 videos recorded overall. Since the subject uses a different amount of time to finish each operation, it has a different time duration (number of frames) for each operation. The quantitative information of the dataset is shown in Figure 7. On average, operation O1 takes the longest time to finish while operation O10 takes the shortest time.

3.2. Evaluation Metrics

The dataset is divided into training, validation, and testing sets for experimental evaluation. The 9th repetition is chosen for validation to measure the model's performance during training, using which the hyperparameters are tuned. The last repetition is selected for performance testing to demonstrate how the trained model can generalize on unseen data. We choose several commonly used metrics [4] to evaluate the model performance, which are as follows:

- Accuracy

$$Accuracy = \frac{\sum_i^N 1(\hat{y}_i = y_i)}{N} \tag{1}$$

- Precision and Recall

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \end{aligned} \tag{2}$$

- F_1 score

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

where $1(\cdot)$ is an indicator function in Equation 3. For a certain class y_i, True Positive (TP) is defined as a sample of class y_i that is correctly classified as y_i; True Negative (TN) means a sample from a class other than y_i is correctly classified as 'not y_i'; False Positive (FP) means a sample from a class other than y_i is misclassified as y_i; False Negative (FN) means a sample from the class y_i is misclassified as a 'not y_i' class. F_1 score is the harmonic mean of Precision and Recall, which ranges in the interval [0,1].

3.3. Implementation Details

The transfer learning model described in Section 2.5 is built using the open source machine learning framework PyTorch [13]. During training, we choose a batch size of 64, a learning rate of 0.001, and a dropout rate of 0.5. Transformations such as random rotating, scaling, and cropping are applied to the training set to include more variations in the training phase, which will help the network learn the most discriminative features and generalize to unseen data. A workstation with one 12 core Intel Xeon processor, 64GB of RAM and one Nvidia Geforce 1080 Ti graphic card is used for the network training.

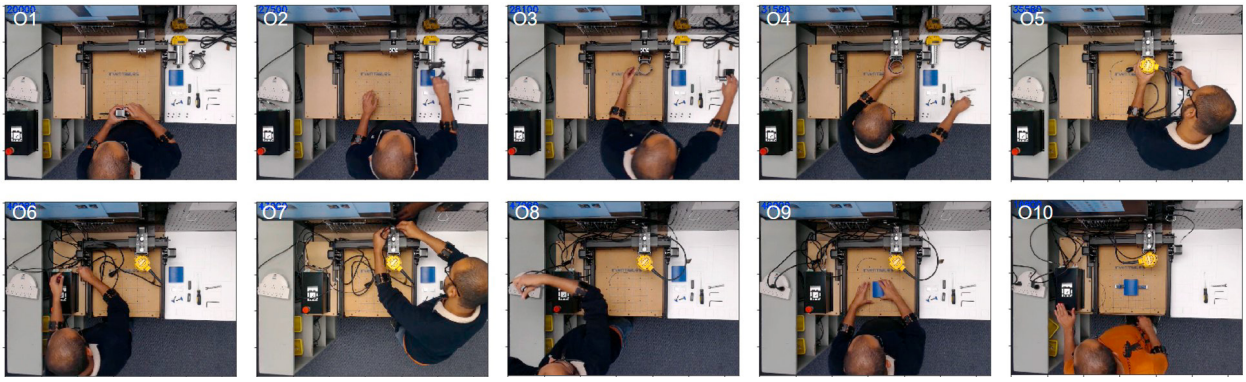

Fig. 4: Examples of the 10 assembly operations.

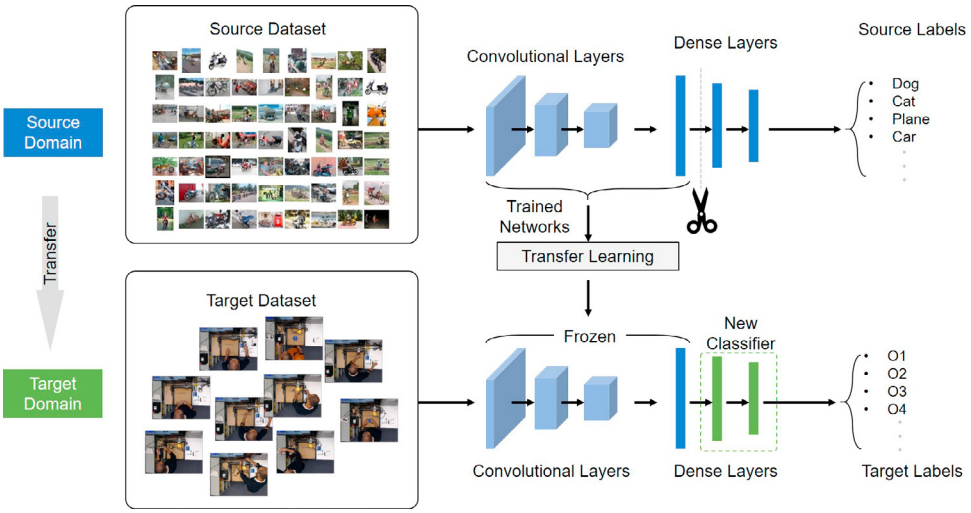

Fig. 5: The architecture of our transfer learning model.

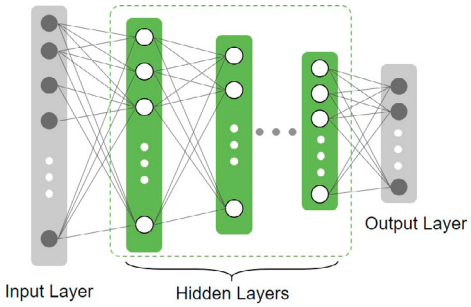

Fig. 6: Illustration of the classifier architecture.

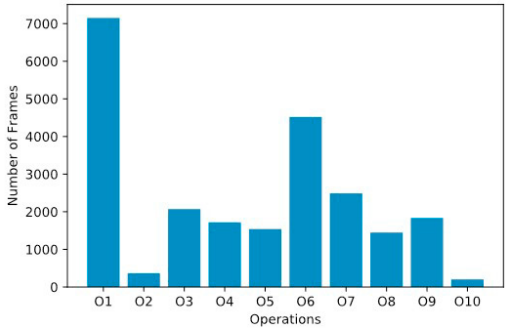

Fig. 7: Averaged number of frames for each operation in the dataset.

3.4. Evaluation of Different Pre-trained Models

There are different pre-trained models with different architectures trained on public datasets, such as ImageNet, for different source tasks. We select three of them, i.e., VGG [15], ResNet [6] and DenseNet [9], in our experiments for comparison. The performance of these three pre-trained models in terms of accuracy, precision, recall and F_1 score is listed in Table 1. Compared with a ResNet model, a VGG model has higher performance for all four evaluation metrics. A DenseNet model has the highest performance among the three, achieving an accuracy of 95%. Therefore, we choose the pre-trained model DenseNet in the following study.

Table 1: Performance (%) comparison of different pre-trained models.

Pre-trained Model	Accuracy	Precision	Recall	F_1 Score
VGG	93.5	92.2	92.0	91.0
ResNet	92.5	90.2	87.6	88.0
DenseNet	94.7	92.8	92.1	92.1

3.5. Impact of Classifier Design

After loading a pre-trained model with partially frozen weights, a new classifier is needed to adapt the source model to the target task. It is infeasible to evaluate all possible classifier designs due to the numerous parameters, such as number of hidden layers between the input and output layers, number of neurons for each hidden layer, and dropout rate during training. To explore the optimal design of hidden layers for the classifier, we compare the performance of four designs using different numbers of layers and neurons: 1). [512 – 256 – 128] (three hidden layers are included and their neuron numbers are 512, 256, and 128, respectively); 2). [512 – 256]; 3). [512]; and 4). [−] (no hidden layer is included, and the input layer is fully connected to the output layer). As shown in Table 2, the four classifier designs are listed and their performances in terms of accuracy, precision, recall and F1 score are compared. It can be seen that, a simpler classifier design, from the top to the bottom, can have better performance and less training time. The 4th design has the highest performance, which reaches 94.7%, 92.8%, 92.1% and 92.1% in accuracy, precision, recall and F_1 score, respectively. Therefore, we choose the 4th design for our customized classifier.

Table 2: Results (%) of different classifier designs.

Hidden Layer	Accuracy	Precision	Recall	F1 Score
[512 – 256 – 128]	92.7	90.9	86.3	87.6
[512 – 256]	93.6	90.2	90.9	89.8
[512]	92.9	92.0	89.4	89.7
[−]	94.7	92.8	92.1	92.1

3.6. Real-Time Recognition

A real-time application of operation recognition is developed to validate the trained model. A screenshot of this application is shown in Figure 8. The video is captured via network transmitting as depicted in Figure 1 or from a saved video file. Inference on each image frame is implemented using the trained model. The prediction of each individual frame is returned and useful information is presented on the interface for users. To make the predictions more stable, a state machine is implemented and a logic for state changing is applied, i.e., if a certain number of consecutive frames are recognized as the next operation, then the current state is updated to the next operation. In addition, the assembly progress can be evaluated quantitatively by accumulating the number of frames for each operation. Such information can be used to provide instructive feedback to the operator in a real-time manner. For example, if a certain operation takes more time to finish than average, instructions of the current operation can be provided to the operator to help improve the working efficiency.

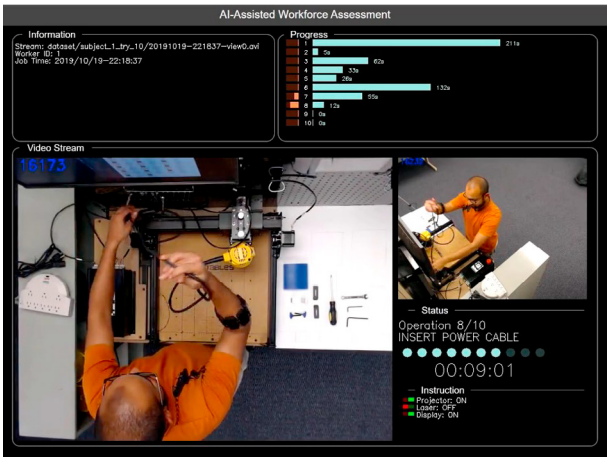

Fig. 8: Real-time recognition on the testing subject.

3.7. Failure Cases

The confusion matrix of the experiment on the testing set is shown in Figure 9. We can see that, most of the frames are along the diagonal and correctly recognized. However, some frames are misclassified and appear as confusing pairs, e.g., O3-O4 and O7-O8. The are 146 frames of O3 misclassified as O4, and 416 frames of O8 misclassified as O7. By reviewing the misclassified frames, as illustrated in Figure 10, we find the reason for the low performance is the high visual similarity shared within each pair makes it confusing and difficult to distinguish between them. Operations O3 and O4 can be very similar because the parts installed at these two steps are adjacent. Operations O7 and O8 share strong similarities because both of them involves cable handling and inserting operation, which makes it challenging for data-driven algorithms to learn the difference.

4. Conclusion and Future Work

In this paper, we develop a real-time fog computing application for assembly operation recognition in human-centered intelligent manufacturing using image frames obtained from a visual camera. An assembly operation task is formulated and a dataset is established, which contains 10 sequential operations. Transfer learning is utilized and the developed model is evaluated on the dataset and achieves a 95% recognition accuracy.

This is an on-going project and some directions for future study are considered, such as recruiting more subjects for data collection to enrich the current dataset, utilizing more cameras to capture the operator's activity from more perspectives, and including more modalities in the current model for information

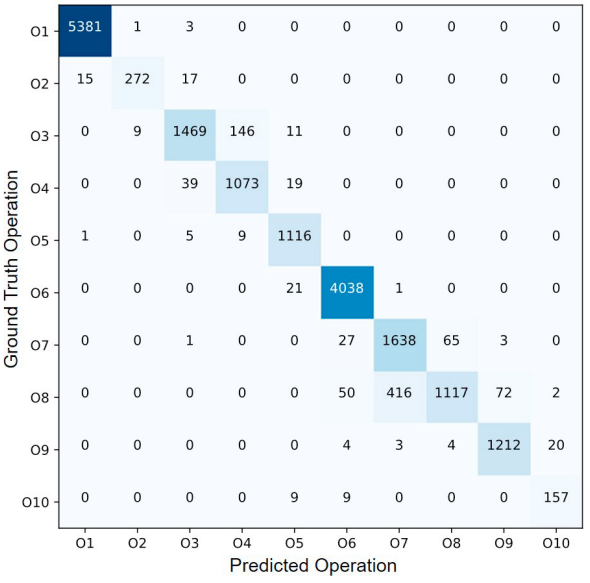

Fig. 9: Confusion matrix of the experiment on the testing set. The values represent the number of frames, e.g., the '146' in the O3 row means there are 146 frames of actual O3 ('install lead screw') incorrectly predicted as O4 ('fix spindle mount').

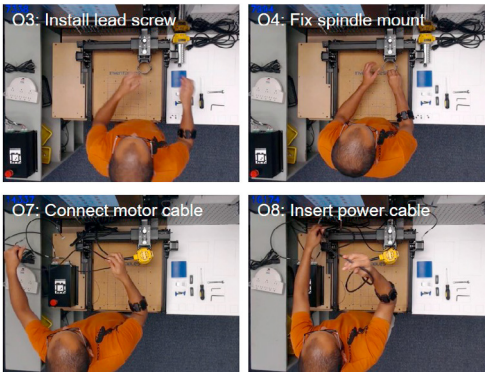

Fig. 10: Failure cases from confusing pairs O3-O4 and O7-O8.

fusion. In addition, instead of using an image-based recognition method, the recording videos can be directly utilized to create a video-based operation recognition model using deep learning methods such as 3D convolutional neural networks.

Acknowledgements

This research work is supported by the National Science Foundation grants CMMI-1646162 and NRI-1830479, and also by the Intelligent Systems Center at Missouri University of Science and Technology. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Al-Amin, M., Qin, R., Tao, W., Leu, M.C., 2018. Sensor data based models for workforce management in smart manufacturing, in: Proceedings of the 2018 Institute of Industrial and Systems Engineers Annual Conference (IISE 2018).

[2] Al-Khafajiy, M., Baker, T., Al-Libawy, H., Waraich, A., Chalmers, C., Alfandi, O., 2018. Fog computing framework for internet of things applications, in: 2018 11th International Conference on Developments in eSystems Engineering (DeSE), IEEE. pp. 71–77.

[3] Azadi, B., Haslgrübler, M., Sopidis, G., Murauer, M., Anzengruber, B., Ferscha, A., 2019. Feasibility analysis of unsupervised industrial activity recognition based on a frequent micro action, in: Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments, ACM. pp. 368–375.

[4] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

[5] Haslgrübler, M., Gollan, B., Ferscha, A., 2016. Towards industrial assistance systems: Experiences of applying multi-sensor fusion in harsh environments, in: Physiological Computing Systems. Springer, pp. 158–179.

[6] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[7] Hu, L., Nguyen, N.T., Tao, W., Leu, M.C., Liu, X.F., Shahriar, M.R., Al Sunny, S.N., 2018. Modeling of cloud-based digital twins for smart manufacturing with mt connect. Procedia Manufacturing 26, 1193–1203.

[8] Hu, Y., Wong, Y., Dai, Q., Kankanhalli, M., Geng, W., Li, X., 2019. semg-based gesture recognition with embedded virtual hand poses and adversarial learning. IEEE Access 7, 104108–104120.

[9] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708.

[10] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

[11] Lee, J., Bagheri, B., Kao, H.A., 2015. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing letters 3, 18–23.

[12] Liu, Y., Fieldsend, J.E., Min, G., 2017. A framework of fog computing: Architecture, challenges, and optimization. IEEE Access 5, 25445–25454.

[13] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch .

[14] Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S., 2014. Cnn features off-the-shelf: an astounding baseline for recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 806–813.

[15] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .

[16] Tao, W., Lai, Z.H., Leu, M.C., Yin, Z., 2018a. American sign language alphabet recognition using leap motion controller, in: Proceedings of the 2018 Institute of Industrial and Systems Engineers Annual Conference (IISE 2018).

[17] Tao, W., Lai, Z.H., Leu, M.C., Yin, Z., 2018b. Worker activity recognition in smart manufacturing using imu and semg signals with convolutional neural networks. Procedia Manufacturing 26, 1159–1166.

[18] Tao, W., Lai, Z.H., Leu, M.C., Yin, Z., Qin, R., 2019a. A self-aware and active-guiding training & assistant system for worker-centered intelligent manufacturing. Manufacturing letters 21, 45–49.

[19] Tao, W., Leu, M.C., Yin, Z., 2018c. American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion. Engineering Applications of Artificial Intelligence 76, 202–213.

[20] Tao, W., Leu, M.C., Yin, Z., 2019b. Multi-modal recognition of worker activity for human-centered intelligent manufacturing. arXiv preprint arXiv:1908.07519 .