



Skill transfer support model based on deep learning

Kung-Jeng Wang¹ · Diwanda Ageng Rizqi¹ · Hong-Phuc Nguyen²

Received: 6 July 2019 / Accepted: 14 June 2020 / Published online: 27 June 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

The paradigm shift toward Industry 4.0 is not solely completed by enabling smart machines in a factory but also by facilitating human capability. Refinement of work processes and introduction of new training approaches are necessary to support efficient human skill development. This study proposes a new skill transfer support model in a manufacturing scenario. The proposed model develops two types of deep learning as the backbone: a convolutional neural network (CNN) for action recognition and a faster region-based CNN (R-CNN) for object detection. A case study using toy assembly is conducted utilizing two cameras with different angles to evaluate the performance of the proposed model. The accuracy for CNN and faster R-CNN for the target job reached 94.5% and 99%, respectively. A junior operator can be guided by the proposed model given that flexible assembly tasks have been constructed on the basis of a skill representation. In terms of theoretical contribution, this study integrated two deep learning models that can simultaneously recognize the action and detect the object. The present study facilitates skill transfer in manufacturing systems by adapting or learning new skills for junior operators.

Keywords Deep learning · Convolutional neural network · Faster region-based convolutional neural network · Human–machine interaction · Skill transfer

Introduction

Industry 4.0 presents new types of interactions between human and machine. This interaction alters the industrial workforce and significantly improves the nature of work to accommodate increasing variability and flexibility of production (Shin et al. 2017; Oztemel and Gursev 2020). One of the key issues in Industry 4.0 is enabling human-centricity capability, which leads to Operator 4.0 (Romero et al. 2016). Operator 4.0 is characterized using automated systems to diminish the physical and mental stress of humans. In addition, this initiative plays a significant role in enabling humans to exploit and advance their creativity and

innovativeness and improving job skills without sacrificing production objectives.

However, a successful paradigm shift toward Operator 4.0 is not only achieved by proposing new technologies and/or smart machines. Manufacturing companies must increase human productivity by enabling technologies, but this situation also triggers the shifting on hiring patterns and motivates high-skill, high-profit jobs (Jardim-Goncalves et al. 2016; Kiassat and Safaei 2019). By contrast, the availability of highly skilled workers cannot keep up with the human resource market. A study showed that 82% of CEOs and manufacturing executives investigated in the United States revealed that a lack of skilled manpower affects their performance of serving customers (Hill 2017). Junior operators must be timely and efficiently trained.

The lack of skilled workers prompts managers to refine their work processes and introduce new training/skill transfer approaches. Such training approach should be efficient, flexible, and self-organized by machine learning (Liu et al. 2017). In addition, the possession of transferable skills provides flexibility and mobility (Lim et al. 2018). The three types of human–machine relation (Duan et al. 2012) are as follows: relieving human operators by automated devices (physical replacement), improving work performance of

✉ Kung-Jeng Wang
kjwang@mail.ntust.edu.tw
Diwanda Ageng Rizqi
diwanda.ageng@gmail.com
Hong-Phuc Nguyen
nguyenhongphuc@ctu.edu.vn

¹ Department of Industrial Management, National Taiwan University of Science and Technology, Taipei 108, Taiwan

² Department of Industrial Management, Can Tho University, Can Tho City 900000, Viet Nam

human operators by machine support (physical support), and providing task information to advance the cognition process of human operators (informational support and skill transfer).

In comparison with traditional machine learning techniques, deep learning has a network structure that involves multiple hidden layers to extract the embedded features in data and building abstract concepts in a hierarchy procedure (LeCun et al. 2015). A recent report showed that deep learning outperformed human experts while conducting recognition or strategy-related tasks (Sun et al. 2014). In the domain of image recognition, deep learning provides a new approach for increasing the recognition accuracy of human motions. However, human actions and work objects should be efficiently recognized to facilitate skill transfer. Convolutional neural network (CNN) architecture successfully outperforms other deep learning models for most image recognition, classification, and detection tasks (Rawat and Wang 2017).

This study aims to develop a skill transfer support model of tasks in a manufacturing scenario. This skill transfer support model uses the following two types of deep learning as the backbone: CNN for action recognition and faster region-based CNN (RCNN) for object detection. In this model, a human operator is guided while performing tasks based on a skill representation.

The remainder of this article is organized as follows. “Literature review” section reviews previous studies related to human–machine collaboration and deep learning. “Methods” provides the framework and method. “Experiment and discussion” shows the experiment result. “Conclusion” section concludes and discusses future research.

Literature review

Operator 4.0 and human–machine collaboration

Human-centricity concept motivates the development toward Operator 4.0 (Frank et al. 2019), which is aided by cyber-physical system (Ruppert et al. 2018). In the framework of Operator 4.0, workers collaborate and are empowered by physical and digital systems to produce complex tasks (Peruzzini et al. 2020).

Smart machines facilitate human empowerment of their abilities in the following three aspects: extending cognitive strengths, assisting in complex jobs, and embodying human skills to extend physical capabilities (Wilson and Daugherty 2018). For example, human–robot interaction (HRI) focused on physical, cognitive, and social interaction between people and robots to broaden and advance human capabilities and skills (Vasconez et al. 2019). Such study focuses on designing, recognizing, and evaluating the cooperation between humans and robots in communicating and/or sharing in a

physical space for job purposes. In industrial applications, collaborative robotic delivers several advantages, such as relieving from dangerous material handling, heavy tool handling, and high-precision tasks (Villani et al. 2018).

Over the past few decades, HRI has become a growing research area (Landi et al. 2018) in construction, healthcare and assistive robotics, aerospace, edutainment and entertainment, home service, and military and industrial applications (Levratti et al. 2016; Adamides et al. 2017; van Dael et al. 2017; Liu and Wang 2018; Vasconez et al. 2019). In the field of production, new methods and strategies in HRI for fast, affordable, and flexible automation have been constantly identified and developed (Koch et al. 2017; Backhaus and Reinhart 2017). An efficient HRI system can recognize the intention of human workers and provide assistance during an assembly operation (Liu and Wang 2017). The integration of collaborative robots is one of the pillars for flexible automation in the Industry 4.0 era (Koch et al. 2017; Wang et al. 2018a, b). Therefore, the initial paradigm for robot usage has shifted during the years, originating from an idea in which robots work with complete autonomy in a separate cell to a scenario where robots and humans simultaneously work and interact.

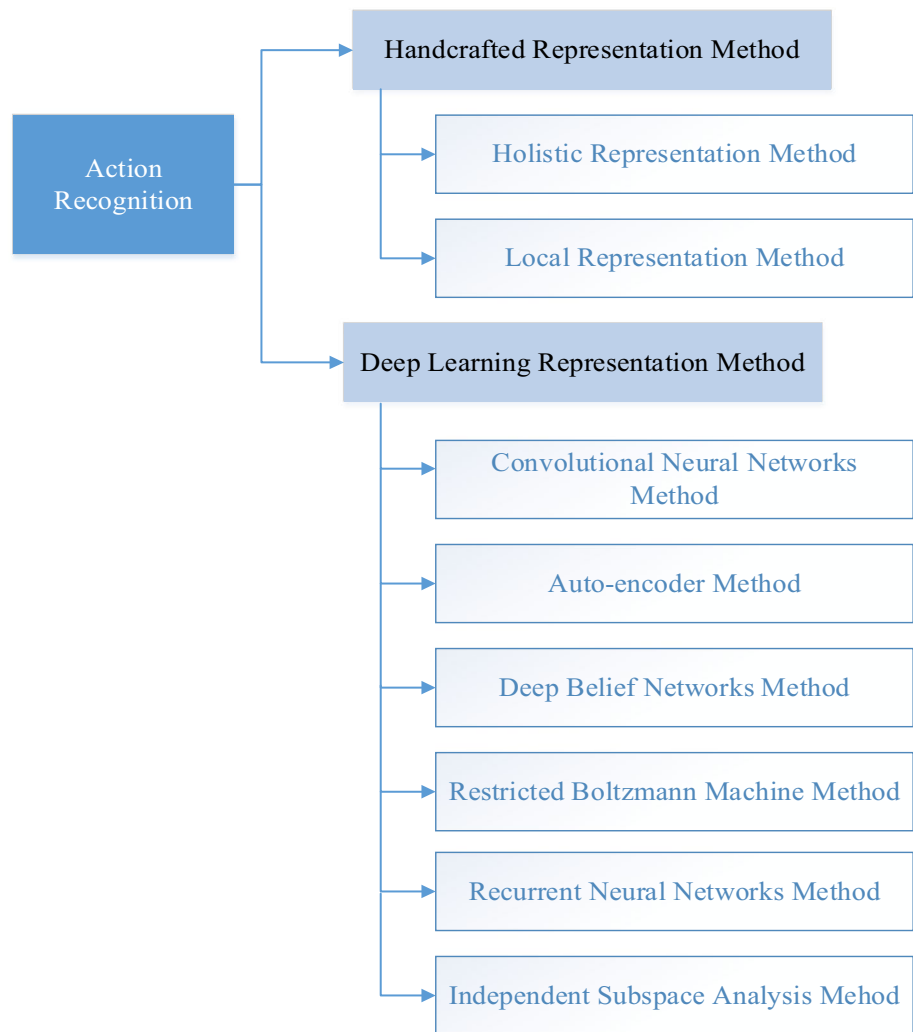
Action recognition

One of the recently investigated aggressive domains in computer vision is human activity recognition. Action recognition is generally implemented in two stages: action representation and classification (Idrees et al. 2017). The core of video action recognition is action representation, which is also denoted as feature extraction. Yao et al. (2019) suggested that an effective action representation should be discriminative, straightforward, and low dimensional. Discriminative refers to the representations of actions from the same class that provides identical information. However, the representations of action from several classes provide different characteristics. Straightforward is the action representation that is easy to compute. In addition, action representation should be low cost in terms of classification and feature saving.

The taxonomy of action recognition is shown in Fig. 1. Action recognition is classified into representation methods based on handcrafted and deep learning. Both methods recognize action classes based on the appearance and motion patterns in videos (Shahroudy et al. 2018).

The study of handcrafted representation method was started by extracting global features, such as silhouette- and optical flow-based features. Subsequently, this method demonstrated a milestone in action recognition field (Yao et al. 2019). Some important works on improved dense trajectories (Peng et al. 2016) include encryption of extracted dense trajectories, trajectory-aligned histograms of oriented

Fig. 1 Taxonomy of action recognition (Yao et al. 2019)



gradients, histogram of optical flow, and motion boundary histograms with the Fisher vector or hybrid super vector (Peng et al. 2016).

By contrast, a deep learning representation method differs from handcrafted one in terms of design (Yao et al. 2019). The handcrafted method manually designates the feature, whereas deep learning representation method can automatically learn the trainable feature from videos. Auto-encoder method enables a neural network to automatically learn a sparse shift-invariant representation of the local $2D + t$ salient information (Baccouche et al. 2012). Deep belief networks learn invariant spatiotemporal features from videos (Chen et al. 2010). The restricted Boltzmann machine catches various human motions based on features of action. Veeriah et al. (2015) applied recurrent neural network, which is known for constructing long short-term memory, to learn and recognize complex dynamics of various actions. Furthermore, an independent subspace analysis method learns invariant and robust spatial features of the normalized video cubes (Pei et al. 2016).

Over the past few years, the CNN-based method is the most researched approach in various fields of computer vision, including action recognition, and has shown a considerable achievement (Ciocca et al. 2018; Yao et al. 2019). CNN works effectively on image processing and understanding task due to the proximity of its layers and its rich available information. Moreover, images can be automatically extracted to produce rich correlated features (Zhang et al. 2018). Ciocca et al. (2018) also stated that features learned by CNN are analyzed to be more powerful and expressive than those of the handcrafted ones. Therefore, CNN is applied in this study to perform action recognition. Further details on the CNN-based methods are discussed in “Convolutional neural network” section.

Object detection

Object detection is a task to estimate the contexts and locations of existing objects in each image. The problem of object detection is determining the location of objects in

a specific image (object localization) and the classification of each object (object classification). Based on this definition, the traditional models for object detection can be split into the following three phases (Zhao et al. 2019): informative region selection, feature extraction, and classification (Table 1). Manually constructing a robust feature descriptor to perfectly characterize all types of objects is challenging due to the variety of appearances, illumination conditions, and background.

The integration of deep neural networks with regions with CNN features (R-CNN) has resulted in a higher gain in this field compared with that of the traditional approach, which uses discriminant local feature descriptors and shallow learnable architectures (Zhao et al. 2019). CNN has deep architecture with the ability to learn more sophisticated features than that of the shallow ones. In addition, the training algorithm facilitates the learning of informative object representations without manually designating the features because of its expressiveness and robustness.

After the introduction of the R-CNN, another improvement model has been recommended (Zhao et al. 2019). The first improvement model is fast R-CNN, which simultaneously binds box regression and classification optimization tasks. In addition, faster R-CNN model is developed to propose an additional sub-network for generation region proposals. The latest developed model is You Only Look Once (YOLO), which achieves object detection by using a fixed-grid regression (Gu et al. 2018). These models not only carry different qualities of detection performance over the primary R-CNN but achieve a real-time and accurate object detection. Further explanation details on faster R-CNN are presented in “Faster regional-convolutional neural network” section.

Convolutional neural network

CNN is a variant of multilayer perceptron inspired from the biological concept, which is a feedforward artificial

neural network (Yao et al. 2019). The architectures of CNN are multistage and trainable, where every stage contains multiple layers (Bhandare et al. 2016), including an input layer, an output layer, and multiple hidden layers. These hidden layers are either convolutional, rectified linear units (ReLU), pooling, or fully connected. The convolutional layer conducts a convolution operation and an additive bias to the input data, initially passing the result via an activation function and then delivering it to the next layer. The convolution operation at location (x, y) in the j th feature map in the i th layer is defined in Eq. (1) as follows:

$$v_{il}^{xy} = \varphi \left(b_{i,j} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{i,j,m}^{p,q} v_{(i-1),m}^{(x+p),(y+q)} \right) \quad (1)$$

where φ is a non-linear activation function, w is the weight matrix, P is the height of the kernel, and Q is the width of the kernel. The ReLU layer applies a non-saturating non-linearity function or loss function (Traore et al. 2018):

$$f(x) = \max(0, x) \quad (2)$$

Non-linear down-sampling forms the pooling layer. Max pooling is the most frequently used pooling function, which takes over the output with the maximum activation among a rectangular neighborhood (Carrio et al. 2017). Finally, after going through the convolutional and pooling layer, the high-level reasoning in the CNN is finalized via fully connected layers, in which each neuron is connected to all activations in the previous layer (Yao et al. 2019).

The classification task is the major function of output layer in CNN architecture. Logistic regression model is commonly used as the output layer for a CNN model. In addition, for multiclass classification task, the logistic regression model is then established as multinomial logistic function, which is mostly termed as softmax function. For j possible classes, a weighting vector W , and a bias b , the probability that vector x is a member of class i

Table 1 The phase of traditional object detection models (revised from Zhao et al. 2019)

Phase	Explanation
Informative region selection	Since different objects may occur in any points of the image and have divergent aspect ratios or sizes, scanning the whole image with a multi-scale sliding window is needed. Even though this strategy is able to discover all possible positions of the objects, yet the weaknesses are also obvious. Because of the large number of candidate windows, this strategy is computationally expensive and also generates too many redundant windows. After all, if only a fixed number of sliding window templates are covered, unsuitable regions may be produced
Feature extraction	To recognize different objects, one has to extract visual features which can present a semantic and robust representation. According to Zhao et al. (2019), there are three types of representative features; which are scale invariant feature transform, HOG, and Haar-Like. They are representative due to the fact that these features can produce representations related with complex cells in human brain
Classification	A classifier is essential to distinguish a target object based on all the existing categories and to represent with more hierarchical, semantic and informative way for visual recognition. Supported vector machine, AdaBoost and deformable part-based model are commonly used for classifying the objects

in softmax function can be defined as follows (Dewa and Afiahayati 2018):

$$P(Y = i|x, W, b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}} \quad (3)$$

CNN has been widely used in some domains of research, including the medical field. For example, CNN is applied to detect and/or classify breast cancer in breast histopathology (Bejnordi et al. 2017). In the manufacturing field, CNN is introduced to classify the defect of circuit board (Iwahori et al. 2018). The CNN is also adapted to classify the existing defect in the electronic circuit board into multiple types of defect based on its shape. Over the past few years, CNN has reached a substantial improvement in image classification and object detection. Some CNN architectures, such as ZFNet (Zeiler and Fergus 2014), VGG (Zhao et al. 2019), GoogLeNet (Szegedy et al. 2015), BN-Inception (Jaderberg et al. 2015), and ResNets (He et al. 2016), have been constructed. These architectures can produce pre-trained models (represented by weights) on large-scale datasets. By contrast, an additional training step (transfer learning) is executed to fine-tune the pre-trained model of the network for learning new dataset with small scale or a new modality (Yao et al. 2019). Image-based action recognition using CNN was also conducted by Qi et al. (2017). They investigated the transfer of CNN from object to action recognition and achieved 82.2% of mAP. The VGG-16 model makes an improvement over AlexNet (Simonyan and Zisserman 2014) and is utilized as the basic model to construct the neural network by using a dataset of people playing musical instruments to evaluate the proposed method.

Inception v2

AlexNet network has been successfully applied to various computer vision tasks, such as object detection, segmentation, human pose estimation, video classification, object tracking, and super-resolution (Szegedy et al. 2016). AlexNet contained 8 layers; the first five were convolutional layers, and the last three were fully connected ones. VGGNet and GoogLeNet resulted in similarly high performance in the ILSVRC classification challenge. The quality of these network architectures is further improved by utilizing deep and wide networks. Both network architectures are widely utilized in many domains, including proposal generation in detection, in which AlexNet cannot compete.

Although VGGNet and GoogLeNet demonstrate high performance, the inception architecture of GoogLeNet is much lower than that of VGGNet or its high performing successors in terms of computational cost. Inception was designed to perform effectively even under limited memory and budget. For example, GoogLeNet engaged only 5 million parameters

while AlexNet used 60 million parameters. Furthermore, utilizing this network in big-data scenarios is feasible due to the computational cost of inception (Szegedy et al. 2016). The layout of Inception v2 network is shown in Table 2.

Faster regional-convolutional neural network

Faster R-CNN comprises two modules (Ren et al. 2017). The first module is a deep fully convolutional network that proposes regions. Instead of using a selective search algorithm on the feature map to identify the region proposal, a separate network is used to predict such region proposals. Meanwhile, the second module is the fast R-CNN detector that works with the proposed region. The region proposal network (RPN) module instructs the fast R-CNN module of the direction.

Ren et al. (2017) stated that an RPN captures an image (of any size) because the input and the outputs are a set of rectangular object proposals, with each set possessing an objectness score. Membership to a set of object classes versus background is measured using objectness score. They attempted to slide a small network over the convolutional feature map, which is the output of the last shared convolutional layer, to generate region proposals. This network then captures an $n \times n$ spatial window of the input convolutional feature map as input. Each sliding window is mapped to a low-dimensional feature. Finally, this low-dimensional feature is supplied into two siblings or fully connected layers, which are a box-regression (reg) and a box-classification (cls) layer.

The RPN and faster R-CNN are trained independently, and their convolutional layers are subsequently modified in different approaches. Rather than learning two split networks, Ren et al. (2017) proposed a technique for sharing convolutional layers between the two networks. A pragmatic

Table 2 Inception v2 network architecture (Szegedy et al. 2016)

Layer (type)	Patch size/stride	Input size
Conv	$3 \times 3/2$	$299 \times 299 \times 3$
Conv	$3 \times 3/1$	$149 \times 149 \times 32$
Conv padded	$3 \times 3/1$	$147 \times 147 \times 32$
Pool	$3 \times 3/2$	$147 \times 147 \times 64$
Conv	$3 \times 3/1$	$73 \times 73 \times 64$
Conv	$3 \times 3/2$	$71 \times 71 \times 80$
Conv	$3 \times 3/1$	$35 \times 35 \times 192$
3 × Inception	5×5 replacement	$35 \times 35 \times 288$
5 × Inception	$n \times n$ factorization	$17 \times 17 \times 768$
2 × Inception	Coarsest grid	$8 \times 8 \times 1280$
Pool	8×8	$8 \times 8 \times 2048$
Linear	Logits	$1 \times 1 \times 2048$
Softmax	Classifier	$1 \times 1 \times 1000$

four-step training algorithm is adopted to learn shared features via alternating optimization. In the first step, the RPN is trained via initialization with an ImageNet pre-trained model and fine-tuned end-to-end for the region proposal task. In the second step, a separate detection network is trained by fast R-CNN by adopting the proposals by RPN. The ImageNet pre-trained model is also used for initialization of the detection network. At this stage, the two networks do not share convolutional layers yet. In the third step, the detector network is used to initialize the RPN training. However, the shared convolutional layers are fixed, and only the layers unique to RPN are fine-tuned. At this stage, the two networks share convolutional layers. Finally, by maintaining shared convolutional layers, the unique layers of fast R-CNN are fine-tuned. As a result, both networks sharing the same convolutional layers comprise a unified network.

Summary

Currently, CNN and faster R-CNN are constructed independently, and combining the two as the backbone of skill transfer support model is possible. Specifically, CNN can be implemented to perform action recognition, while faster R-CNN is implemented to perform object detection. Such a model can be used as a guide for operators to adopt the new skills for assembly operations.

Methods

Research framework

A skill transfer support model framework by CNN and faster R-CNN is proposed in the present study. The framework of this research is presented in Fig. 2. In this study, human expert operations are recorded using two cameras from different angles. The videos are split into images. Each image comprises the motion of the operator and the parts/tools related to the operator's task. The image is then trained using CNN and faster R-CNN. The context of the actions is recognized when action recognition and object detection are performed to assist or identify the intention of the operator. In addition, this study applies a formal skill representation to define alternatives for job sequences. In the skill transfer section, this model aids a junior operator by advising him/her on what should be performed next based on the skill representation.

Skill representation is developed as a precedence diagram to define the standard operation procedure of the operator doing the assembly operations. In some circumstances, there are several operation procedures of producing the same product. Therefore, all the possible sequences are drawn to guide the operator as needed. Furthermore,

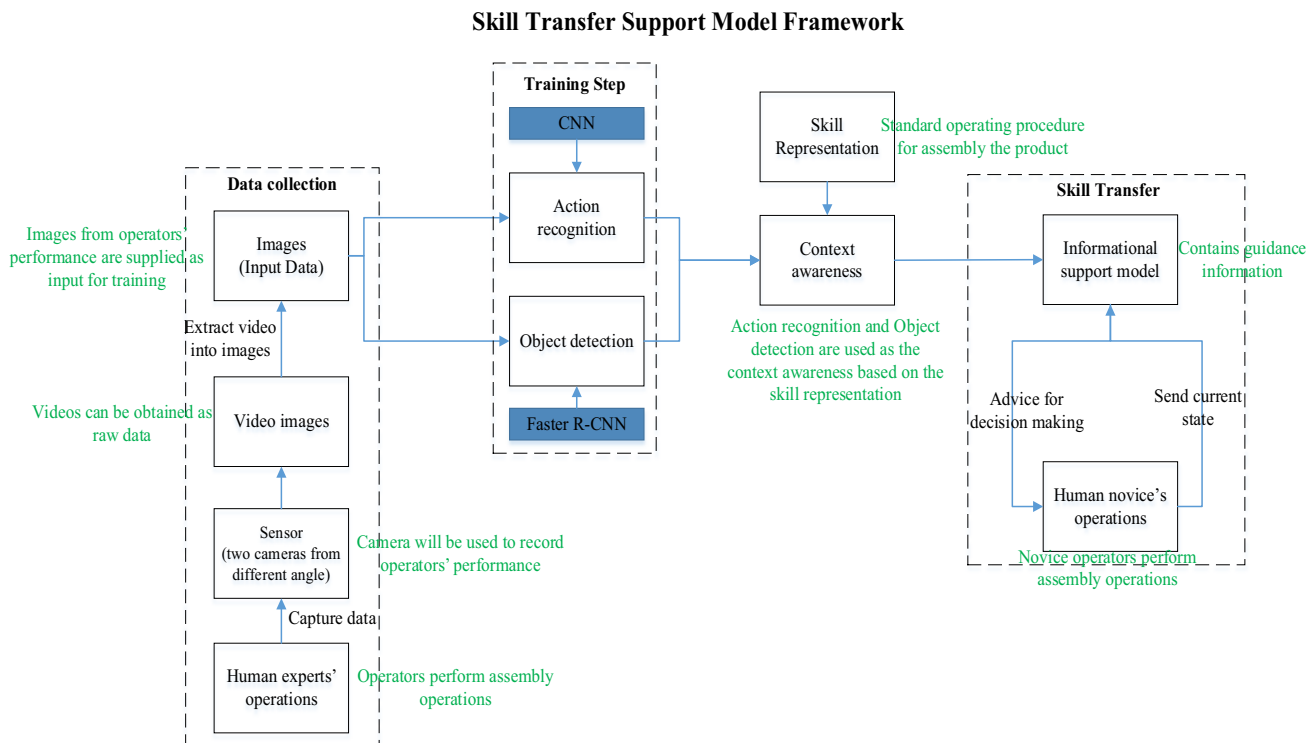


Fig. 2 The proposed framework of a skill transfer support model

based on the possible sequences, the skill transfer model will aid the operator by advising them regarding the sequential operation. This advice is displayed in the form of text information of what should be done next based on their current state of motions, as well as showing them the corresponding tool/object required.

Action recognition based on deep learning approach is implemented in this study. CNN is designed to distinguish between different action classes in an assembly operation. Video action recognition is divided into two tasks: classification and detection. Classification indicates assigning a set of predefined action classes, while detection indicates temporally locating predefined action in a video.

CNN architecture for action recognition

CNN is developed specifically in this study for action recognition. As shown in Figs. 3 and 4, the inputs for this network are images with the size of $100 \times 100 \times 3$ from two sets of cameras in different angles, whereas the output of this network is the action classification of an operator’s task. CNN architecture comprises three convolutional layers, two max pooling layers, and two fully connected layers. Three dropout layers are also used to maintain the capability of the network in demonstrating better generalization performance and less overfitting of the training data.

CNN is constructed from an input layer, an output layer, and multiple hidden layers, where the hidden layers are either convolutional, pooling, or fully connected. The convolutional layer operates a convolution operation and an additive bias to the input data and passes the result initially via an activation function and then to the next layer. The convolution operation at location (x, y) in the j th feature map in the i th layer of this study is defined in Eq. (4), where φ is a non-linear activation function, b is an additive bias, m is the number of layers, w is the weight matrix, and P and Q are the height and width of the kernel, respectively.

$$v_{il}^{xy} = \varphi \left(b_{i,j} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{i,j,m}^{p,q} v_{(i-1),m}^{(x+p),(y+q)} \right) \quad \{i = 1, 4, 7\}. \tag{4}$$

Faster R-CNN architecture for object detection

The proposed faster R-CNN applies a single yet unified network for object detection as shown in Fig. 5. Faster R-CNN has two networks: RPN for generating region proposals and a network using these proposals for object detection. The images with the size of $299 \times 299 \times 3$ from two cameras with different angles are initially provided as an input to a CNN that produces a convolutional feature map. On the contrary, the output of this network is the image with the boundary

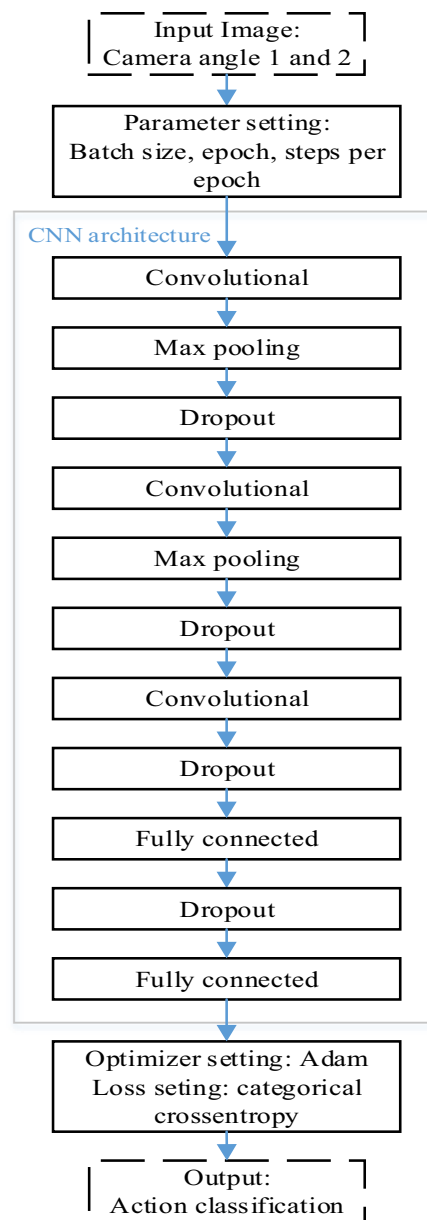


Fig. 3 The present CNN framework

box and classification of the object. The phase of object detection applied in this study has different orders with the phase of a traditional object detection model. Object detection starts with the feature extraction using CNN, followed by RPN, and finalized with classification. The detailed procedure of faster R-CNN is shown in Figs. 6 and 7.

RPN carries the output feature maps from the first CNN as input. A sliding window with $n = 3$ is used in this study, indicating that it slides 3×3 filters over the feature maps to create the region proposals. The detailed procedure of the RPN is illustrated in Fig. 8. The RPN outputs feed into two separate fully connected layers to predict a boundary box and two objectness scores. The objectness measures whether

Fig. 4 The present CNN procedure

Procedure: Convolutional Neural Network

Input: Images from cameras (angle-1&2)

Output: Action classification

Parameters:

Batch size - the size of samples for networks

Steps per epoch - the number of times the training loop to update parameters

Epoch - the number of times to run over the data set extracting batches

BEGIN

Step 1: Categorize the images based on the action

Step 2: Set the training parameters

- Batch size, epoch, and steps per epoch

Step 3: Construct the CNN architecture

- Construct the convolutional, pooling and dropout layer

Step 4: Set the model optimizer and loss function

Step 5: Train the model

- Split the dataset into train (train_x and train_y) and test data set

Step 6: Test the model

- Validate the model using test data set (video)

END

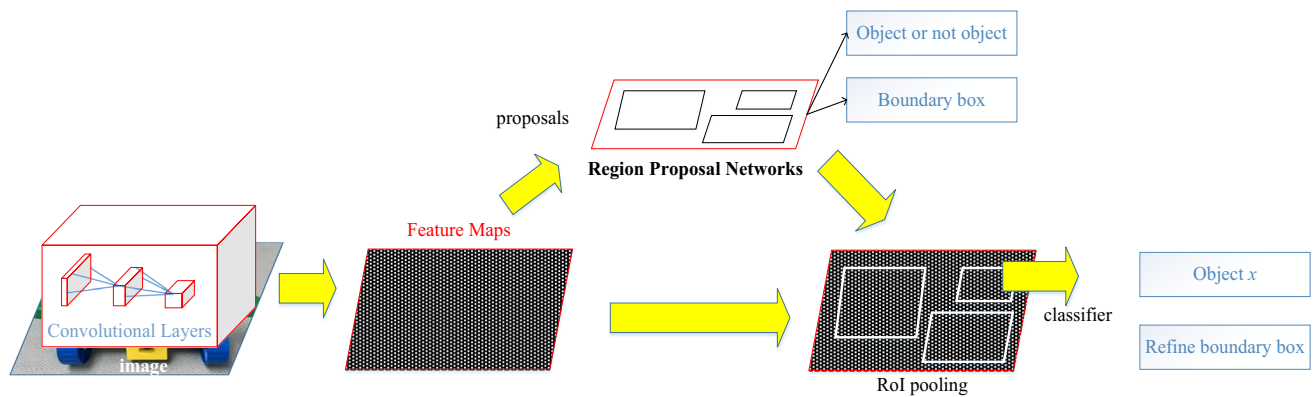


Fig. 5 The proposed faster R-CNN framework (revised from Ren et al. 2017)

the box contains the object while the classifier has two possible classes, namely, object or background. The predicted region proposals are then reshaped using an RoI pooling layer, which is then used to classify the image among the proposed regions and predict the offset values for the bounding boxes.

This study conducted a transfer learning strategy for training the network. Given that the features generated by the preceding layers are more general than those generated later in the process, inception v2 adaptation is determined because the features become specific to the details of the image classes involved in the training dataset. Moreover, this strategy can reduce the problem of network overfitting because the convolutional layers in the network have been trained on a large Microsoft Coco dataset (Lin et al. 2014; Microsoft 2019).

The parameters related to the training process of the proposed CNN and faster R-CNN are shown in Table 3. The initial learning rate was set to a relatively low value for the

transferred convolutional and pooling layers (these layers have been previously trained on the Coco dataset) to train the faster R-CNN network. The overall procedure of the proposed skill transfer support model is shown in Fig. 9.

Skill representation and skill transfer

A job is assumed to be done by one of many task sequences. These options offer the flexibility to the operator to facilitate the production of the desired product in many approaches, which is defined in this research as the skill representation. The sequences is based on a skill representation diagram. The sequences of tasks (A thru J) based on the skill representation diagram are illustrated in Fig. 10.

- A–B–D–G–J
- A–B–E–H–J
- A–C–F–I–J

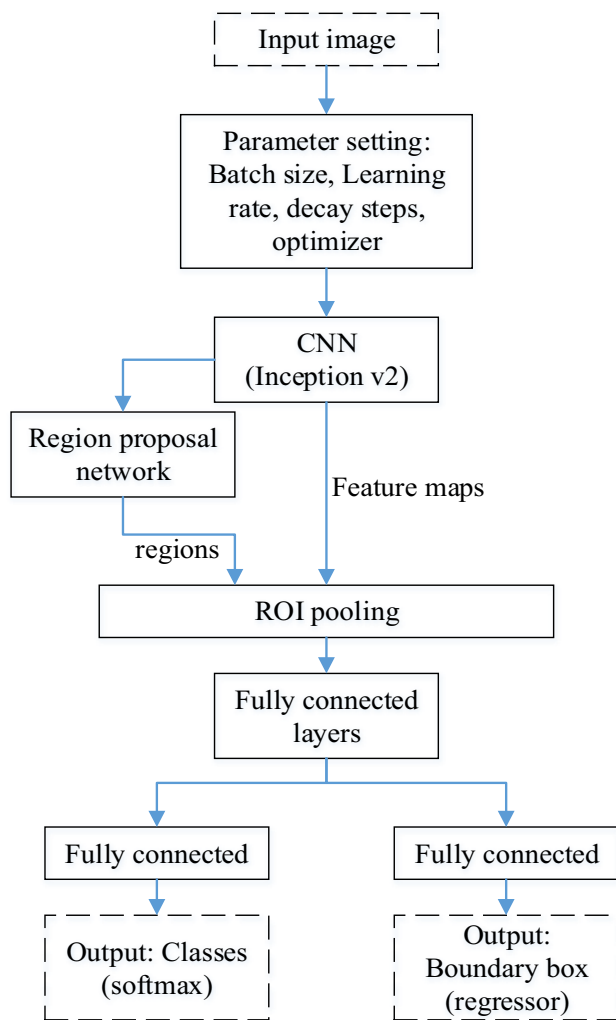


Fig. 6 The present faster R-CNN framework

After independently training the two neuro network models, they are combined as the backbone of the proposed skill transfer support model. The model guides a junior operator while performing a sequence of assembly operations. In addition, the action recognition and object detection model can be run simultaneously, and the proposed model is supplied by guidance based on the skill representation.

Experiment and discussion

Experimental setting

A case study using Lego assembly was conducted to evaluate the performance of the proposed model. In the experiment, four components must be assembled to produce a desired shape, as shown in Fig. 10. In this experiment, action recognition is one of the main issues, as shown in Fig. 12. Several

options of sequence operations can be performed in terms of Lego assembly. The assembly process was recorded by a video camera. The video images were processed to recognize the human activities associated with each video frame and determine the components correlated with the action (Fig. 11).

Illustration on skill representation

Three sequence options can be performed to produce the assembled Lego. These options offer the flexibility to the operator to facilitate the production of the desired Lego in many approaches, which is defined in this research as the skill representation. The sequences based on the skill representation diagram is listed in Fig. 10.

For example, the operator can start with Component 1 and then assemble Component 2 to construct the Lego. Subsequently, this shape can be assembled with Components 3 or 4. If the operator selects to assemble with Component 3, then the final step is combining the shape into Component 4. Otherwise, if the operator selects to assemble with Component 4, then the final step is combining the shape into Component 3. Nine classes of actions and nine classes of objects are available for training the model that fits into the scenario (see in “Appendix”).

Model evaluation

Images covering all motions involved in the assembly process were obtained prior to taking the video from the two cameras to train the CNN and faster R-CNN network for human action recognition and assembly object detection. The output of the CNN model is the classification of the motion of the operator’s task, whereas the output of the faster R-CNN model is the detection of the objects that appear in the video while performing the assembly tasks. The performance of the human operator is slightly different while recording the video, thereby reflecting the variability of human operator in performing the same task. This variability is utilized to avoid overfitting in the training.

This experiment was conducted using two cameras, which work independently and set in different angles, as shown in Fig. 13. The videos were recorded for 10 to 17 s depending on the operators’ speed of performing the operations. The frame width is 540 pixels and the height is 960 pixels. The frame rate of the videos is 30 frames/s. Every operator completed three trials one per sequence of motion to enrich the training dataset.

Among all the images, 80% was used for training the networks and 20% were allotted for testing. The learning and loss curves of nine CNN classes are shown in Fig. 14a, b, respectively. The training accuracy achieved 80% after 10 epochs. In addition, CNN has a good fit because the

Fig. 7 The present faster R-CNN procedure

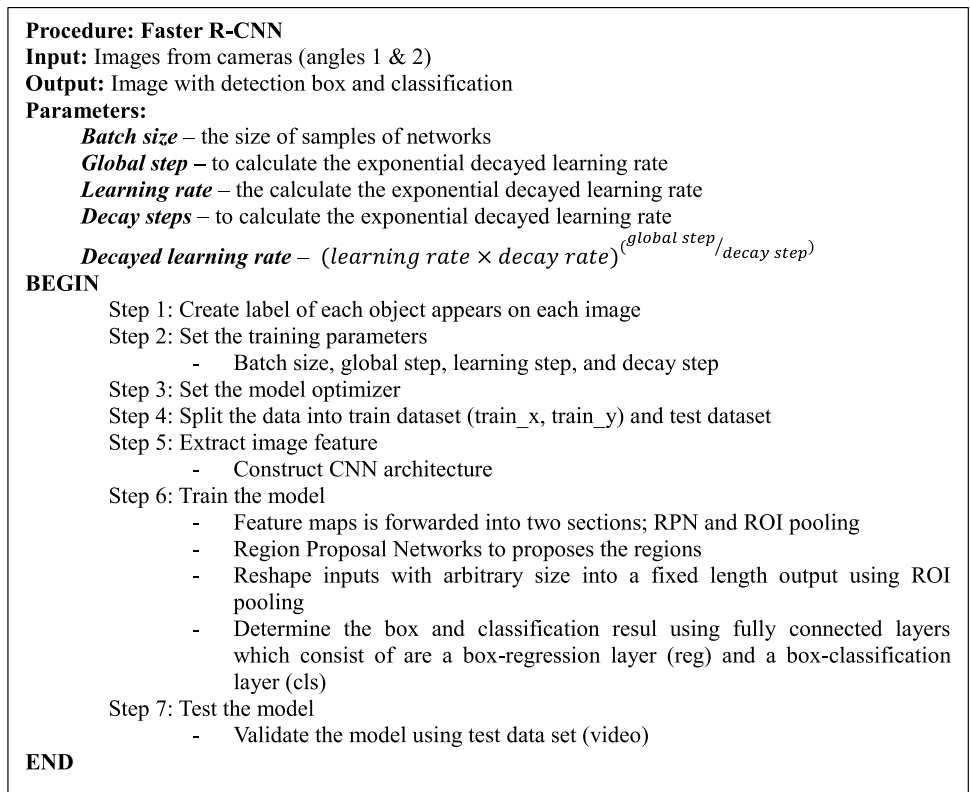


Fig. 8 The proposed region proposal network (RPN) framework (revised from Ren et al. 2017)

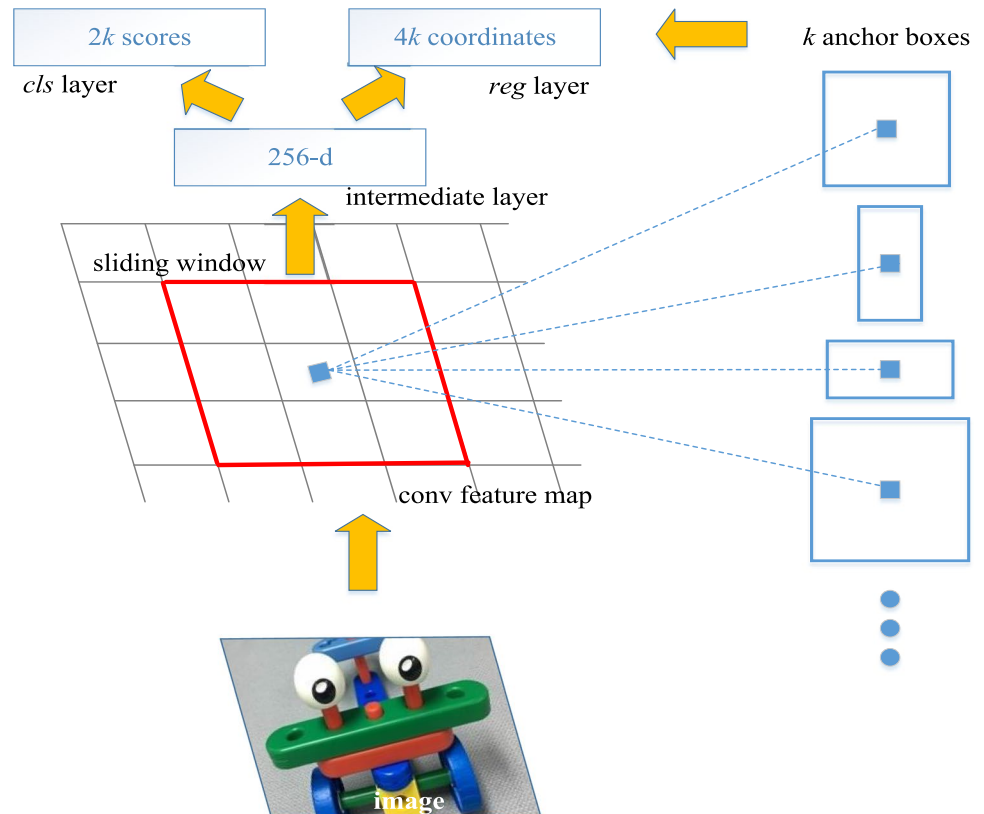


Table 3 Settings of parameter related to CNN and Faster R-CNN

Parameters	Value	
<i>(a) Learning rate of CNN</i>		
Batch size	60	
Steps per epoch	25	
Epoch	25	
Layer (type)	Output shape	Param #
<i>(b) Layer settings of CNN</i>		
conv2d_7 (Conv2D)	(None, 98, 98, 32)	896
max_pooling2d_5 (MaxPooling2D)	(None, 49, 49, 32)	0
dropout_9 (Dropout)	(None, 49, 49, 32)	0
conv2_8 (Conv2D)	(None, 47, 47, 64)	18,496
max_pooling2d_6 (MaxPooling2D)	(None, 23, 23, 64)	0
dropout_10 (Dropout)	(None, 23, 23, 64)	0
conv2d_9 (Conv2D)	(None, 21, 21, 128)	73,856
dropout_11 (Dropout)	(None, 21, 21, 128)	0
flatten_3 (Flatten)	(None, 56448)	0
dense_5 (Dense)	(None, 128)	7,225,472
dropout_12 (Dropout)	(None, 128)	0
dense_6 (Dense)	(None, 9)	1161
Parameters	Value	
<i>(c) Learning rate of faster R-CNN</i>		
Batch size	25	
Initial learning rate	0.0002	
Decay steps	900,000	
Global steps	120,000	
SGD momentum optimizer value	0.90	

Fig. 9 Procedure of the proposed skill transfer support model**Procedure: Proposed skill transfer support model****Input:** Video, CNN model, Faster R-CNN model, Skill representation**Output:** Video with action classification, object detection, and assembly guidance**BEGIN**

Step 1: Load faster R-CNN model for object detection

Step 2: Load CNN model for action recognition

- Define the label of action classes

Step 3: Create function of CNN

- Resize the frame, convert data type and predict the frame

Step 4: Open the tensorflow session for faster R-CNN

Step 5: Load the video and set the frame per second

Step 6:

WHILE video is opened:

expand dimensions

run tensorflow and detect the objects using faster R-CNN model

visualize the boundary box and label

load the CNN function and predict the action

complement text based on skill representation

display video frame

END

plot of training and validation loss decreases to a point of stability with a slight difference. For faster R-CNN, the loss curve for nine classes is shown in Fig. 12c. The

loss values converged at 5,000 steps. This model has good fit because the plot of the training and validation loss decreases to a point of stability with a slight difference.

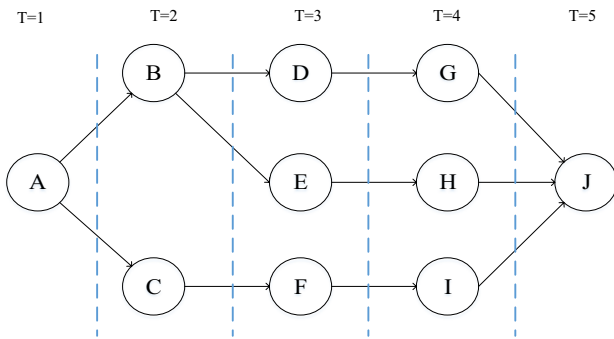


Fig. 10 Skill representation diagram

The training accuracy for CNN and faster R-CNN is 96.16% and 98.46% in nine classes, respectively. Moreover, the F1-score is employed as a performance indicator of the proposed model. The F1-score reflects better the confusion matrix and presents a weighted compromise between precision and recall. F1 score is calculated using Eq. (5). Faster R-CNN implemented for the present model achieves 94.47%.



(a) Example of 1st angle camera

(b) Example of 2nd angle camera

Fig. 13 Two cameras angle setting

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

Once the training was completed, the two networks were used to process the video images. From the available frames, 200 frames were randomly selected and used for testing the

Fig. 11 A case study: components and assembled lego

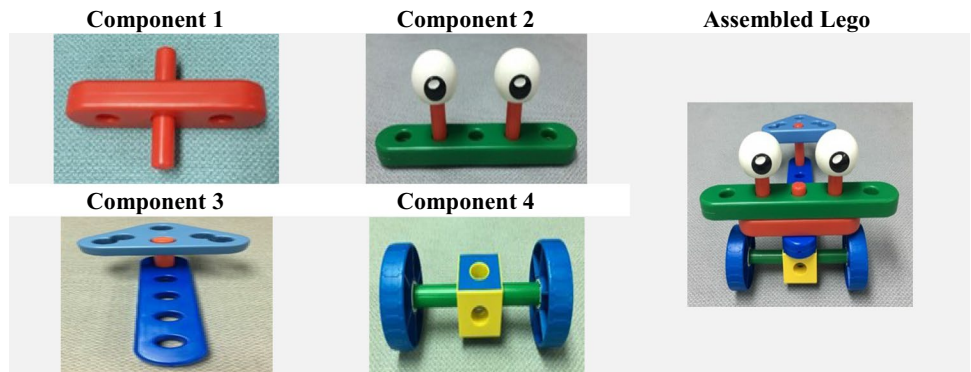


Fig. 12 Examples of human action images



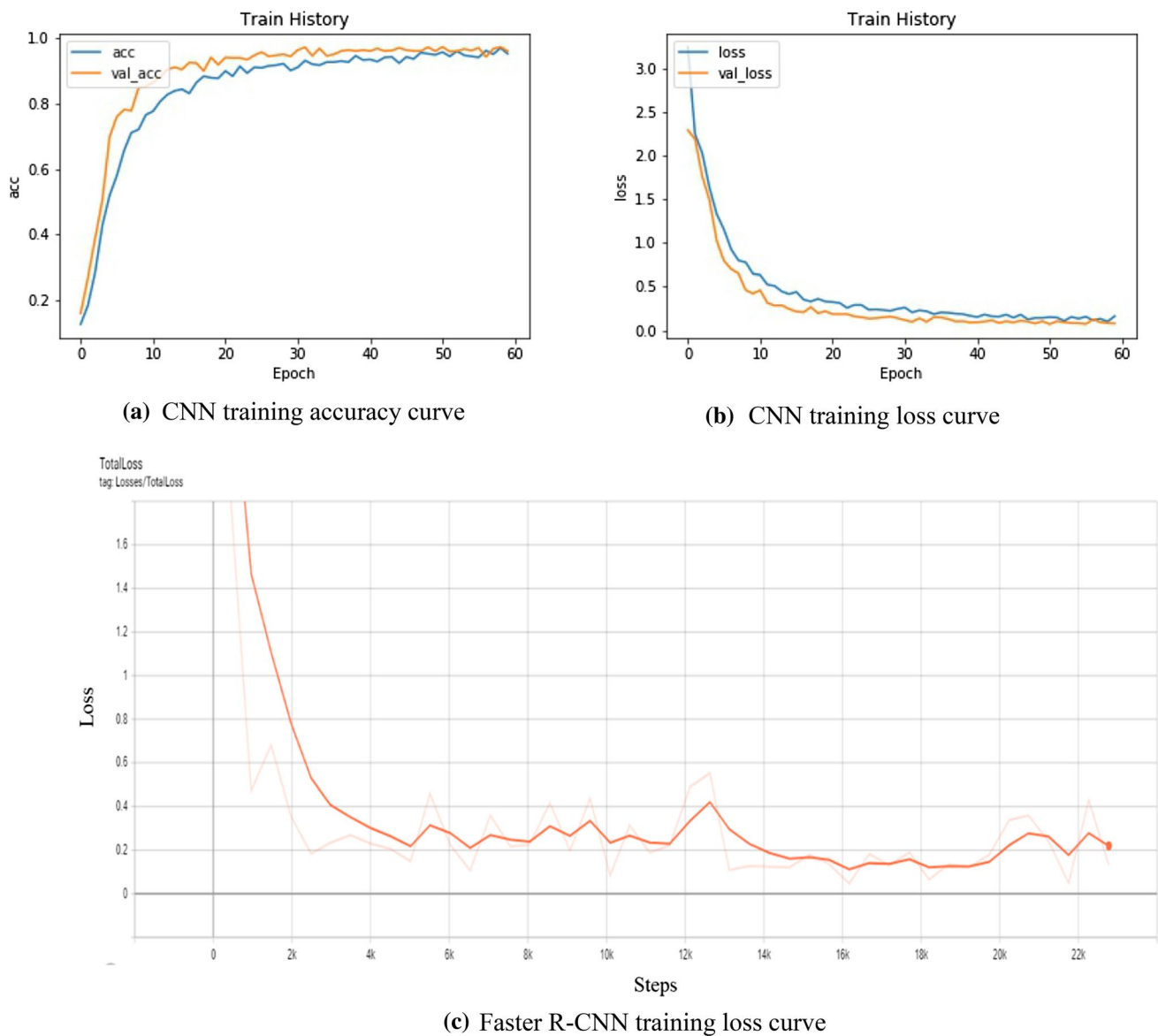


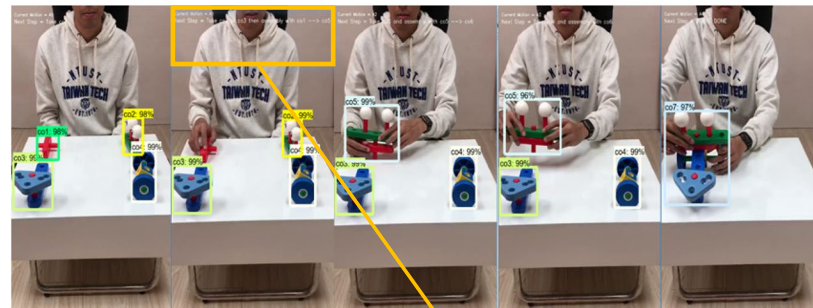
Fig. 14 CNN and faster R-CNN training performances

networks. For each test frame, the action recognition model and object detection were applied to respectively recognize the motion and object associated with the frame. Among the 200 test frames, 11 frames were misclassified in action recognition model, leading to a classification accuracy of 94.5%. Meanwhile, in the object detection model, two objects were misclassified, leading to a classification accuracy of 99%. Most misclassifications occurred during the transitions among human motions, which caused uncertainty in classifying these transition motions into a predefined category.

Illustration on skill transfer support

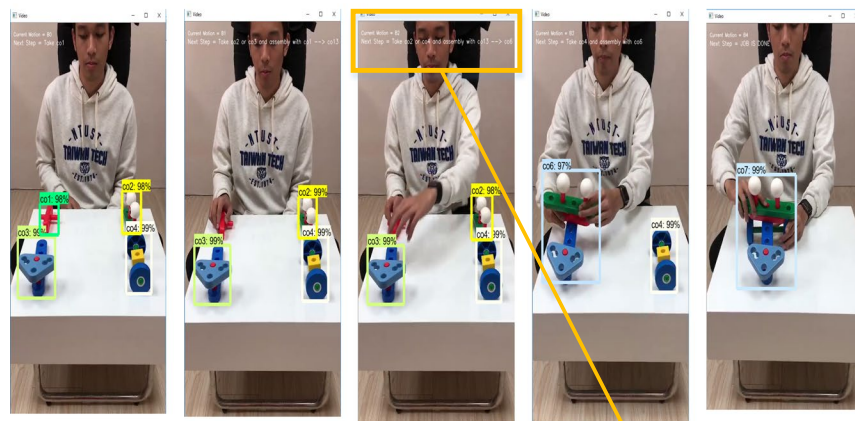
In the proposed scenario, the three sequences are considered options to construct the same finished good. After detecting the motion of the operator, the model guides the operator by giving instructions of their subsequent task, as shown in Fig. 15. This model can sequentially detect the motion of each class. Therefore, the operator is guided successively based on the currently selected operation.

Fig. 15 Skill transfer support model: an illustration



Current Motion = A1
Next Step = Take co2 or co3 then assembly with co1 --> co5

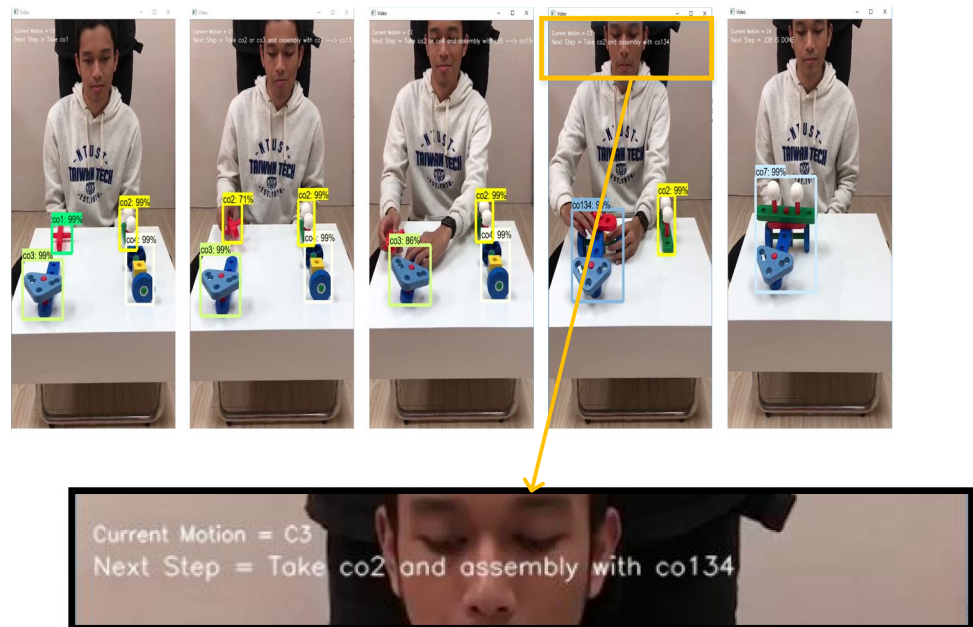
(a) Illustration of 1st sequence



Current Motion = B2
Next Step = Take co2 or co4 and assembly with co13 --> co6

(b) Illustration of 2nd sequence

Fig. 15 (continued)

(c) Illustration of 3rd sequence

Conclusion

This study developed a skill transfer support model for skill transfer of assembly tasks in a manufacturing scenario. This model used two types of deep learning as the backbone: CNN for action recognition and Faster R-CNN for object detection. Inside this model, the human operator is guided by the model based on its skill representation during performance of assembly tasks. The proposed CNN obtained 94.5% accuracy in action recognition. The object detection model achieved 99% accuracy. Faster R-CNN implemented also achieves 94.47%. Subsequently, these models are integrated and run simultaneously to advise the junior operator in terms of the assembly tasks.

In terms of practical contribution, the proposed model enables the following functions:

- To help junior operators in performing complex tasks.
- To guide the operator on the subsequent task on the basis of a skill representation and recommend the tools or part related to a particular task.
- To propose a new training method for new jobs.

In terms of theoretical contribution, this study achieves the following goals:

- To integrate two deep learning models, namely, CNN and faster R-CNN, to offer a new skill-transferring method from senior to junior operators.
- To perform effectively in terms of accuracy and F1-score.
- To simultaneously recognize the action of a worker and detect objects.





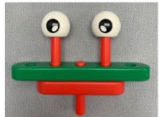
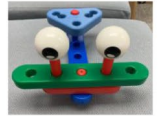



Some challenges in the future study include the following:

- The grasp prediction model can be further studied to empower machine recognition to humans to realize high performance in HRI. Such a model will be informative for robot action planning to assist the operator by handing over parts or tools related to the tasks.
- Additional learning modules, such as single-shot detector or YOLO as the comparison of the currently used module, can be developed and used for object detection.
- Complicated operations that involve assembly and split motions and small parts or tools can be experimented to test the robustness of the proposed model.










Acknowledgements The authors gratefully acknowledge the comments and suggestions of the editor and the anonymous referees. This work is partially supported by Ministry of Science and Technology of the Republic of China (Taiwan) under the Grant No. MOST 107-2221-E-011-101-MY3.

Appendix: Object and action classes

(a) Object

Class	Image
1	
2	
3	
4	
5	
6	
7	
8	
9	

(b) Action

Class	Image	Class	Image
1		6	
2		7	
3		8	
4		9	
5			

References

- Adamides, G., Katsanos, C., Constantinou, I., Christou, G., Xenos, M., Hadzilacos, T., et al. (2017). Design and development of a semi-autonomous agricultural vineyard sprayer: Human–robot interaction aspects. *Journal of Field Robotics*, *34*(8), 1407–1426.
- Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2012). Spatio-temporal convolutional sparse auto-Encoder for sequence classification. In *BMVC* (pp. 1–12).
- Backhaus, J., & Reinhart, G. (2017). Digital description of products, processes and resources for task-oriented programming of assembly systems. *Journal of Intelligent Manufacturing*, *28*(8), 1787–1800.
- Bejnordi, B. E., Zuidhof, G., Balkenhol, M., Hermsen, M., Bult, P., van Ginneken, B., et al. (2017). Context-aware stacked convolutional neural networks for classification of breast carcinomas in whole-slide histopathology images. *Journal of Medical Imaging*, *4*(4), 044504.
- Bhandare, A., Bhide, M., Gokhale, P., & Chandavarkar, R. (2016). Applications of convolutional neural networks. *International Journal of Computer Science and Information Technologies*, *7*(5), 2206–2215.
- Carrio, A., Sampedro, C., Rodriguez-Ramos, A., & Campoy, P. (2017). A review of deep learning methods and applications for unmanned aerial vehicles. *Journal of Sensors*. <https://doi.org/10.1155/2017/3296874>.
- Chen, B., Ting, J., Marlin, B., & Freitas, N. (2010). Deep learning of invariant spatio-temporal features from video. In *Proceedings of the annual conference of on neural information processing systems (NIPS)*.
- Ciocca, G., Napoletano, P., & Schettini, R. (2018). CNN-based features for retrieval and classification of food images. *Computer Vision and Image Understanding*, *176*, 70–77.
- Dewa, C. K., & Afiahayati, (2018). Suitable CNN Weight Initialization and Activation Function for Javanese Vowels Classification. *Procedia Computer Science*, *144*, 124–132.
- Duan, F., Tan, J. T. C., Tong, J. G., Kato, R., & Arai, T. (2012). Application of the assembly skill transfer system in an actual cellular manufacturing system. *IEEE Transactions on Automation Science and Engineering*, *9*(1), 31–41.
- Frank, A. G., Dalenogare, L. S., & Ayala, N. F. (2019). Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*, *210*, 15–26.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354–377.
- 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).
- Hill, T. (2017). *Manufacturing strategy: The strategic management of the manufacturing function*. London: Macmillan International Higher Education.
- Idrees, H., Zamir, A. R., Jiang, Y. G., Gorban, A., Laptev, I., Sukthankar, R., et al. (2017). The THUMOS challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, *155*, 1–23.
- Iwahori, Y., Takada, Y., Shiina, T., Adachi, Y., Bhuyan, M. K., & Kijisirikul, B. (2018). Defect Classification of Electronic Board Using Dense SIFT and CNN. *Procedia Computer Science*, *126*, 1673–1682.
- Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. In *Advances in neural information processing systems* (pp. 2017–2025).
- Jardim-Goncalves, R., Grilo, A., & Popplewell, K. (2016). Novel strategies for global manufacturing systems interoperability. *Journal of Intelligent Manufacturing*, *27*(1), 1–9.
- Kiassat, C., & Safaei, N. (2019). Effect of imprecise skill level on workforce rotation in a dynamic market. *Computers & Industrial Engineering*, *131*, 464–476.
- Koch, P. J., van Amstel, M. K., Dębska, P., Thormann, M. A., Tetzlaff, A. J., Bøgh, S., et al. (2017). A skill-based robot co-worker for industrial maintenance tasks. *Procedia Manufacturing*, *11*, 83–90.
- Landi, C. T., Villani, V., Ferraguti, F., Sabattini, L., Secchi, C., & Fantuzzi, C. (2018). Relieving operators’ workload: Towards affective robotics in industrial scenarios. *Mechatronics*, *54*, 144–154.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.
- Levratti, A., De Vuono, A., Fantuzzi, C., & Secchi, C. (2016, July). TIREBOT: A novel tire workshop assistant robot. In *2016 IEEE international conference on advanced intelligent mechatronics (AIM)* (pp. 733–738). IEEE.
- Lim, C. H., Kim, M. J., Heo, J. Y., & Kim, K. J. (2018). Design of informatics-based services in manufacturing industries: case studies using large vehicle-related databases. *Journal of Intelligent Manufacturing*, *29*(3), 497–508.
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755). Cham: Springer.
- Liu, M., Ma, J., Lin, L., Ge, M., Wang, Q., & Liu, C. (2017). Intelligent assembly system for mechanical products and key technology based on internet of things. *Journal of Intelligent Manufacturing*, *28*(2), 271–299.
- Liu, H., & Wang, L. (2017). Human motion prediction for human–robot collaboration. *Journal of Manufacturing Systems*, *44*, 287–294.
- Liu, H., & Wang, L. (2018). Gesture recognition for human–robot collaboration: A review. *International Journal of Industrial Ergonomics*, *68*, 355–367.
- Microsoft (2019) Microsoft COCO: Common objects in context. <https://arxiv.org/abs/1405.0312>
- Oztemel, E., & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, *31*(1), 127–182.
- Pei, L., Ye, M., Zhao, X., Dou, Y., & Bao, J. (2016). Action recognition by learning temporal slowness invariant features. *The Visual Computer*, *32*(11), 1395–1404.
- Peng, X., Wang, L., Wang, X., & Qiao, Y. (2016). Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, *150*, 109–125.
- Peruzzini, M., Grandi, F., & Pellicciari, M. (2020). Exploring the potential of Operator 4.0 interface and monitoring. *Computers & Industrial Engineering*, *139*, 105600.
- Qi, T., Xu, Y., Quan, Y., Wang, Y., & Ling, H. (2017). Image-based action recognition using hint-enhanced deep neural networks. *Neurocomputing*, *267*, 475–488.
- Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, *29*(9), 2352–2449.
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (pp. 1137–1149). Los Alamitos, CA: IEEE
- Romero, D., Bernus, P., Noran, O., Stahre, J., & Fast-Berglund, Å. (2016, September). The operator 4.0: human cyber-physical systems & adaptive automation towards human-automation symbiosis work systems. In *IFIP international conference on advances in production management systems* (pp. 677–686). Cham: Springer.

- Ruppert, T., Jaskó, S., Holczinger, T., & Abonyi, J. (2018). Enabling technologies for operator 4.0: A survey. *Applied Sciences*, 8(9), 1650.
- Shahroudy, A., Ng, T. T., Gong, Y., & Wang, G. (2018). Deep multimodal feature analysis for action recognition in rgb+d videos. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 40(5), 1045–1058.
- Shin, S. J., Kim, D. B., Shao, G., Brodsky, A., & Lechevalier, D. (2017). Developing a decision support system for improving sustainability performance of manufacturing processes. *Journal of Intelligent Manufacturing*, 28(6), 1421–1440.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
- Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1891–1898).
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).
- Traore, B. B., Kamsu-Foguem, B., & Tangara, F. (2018). Deep convolution neural network for image recognition. *Ecological Informatics*, 48, 257–268.
- van Dael, M., Verboven, P., Dhaene, J., Van Hoorebeke, L., Sijbers, J., & Nicolai, B. (2017). Multisensor X-ray inspection of internal defects in horticultural products. *Postharvest Biology and Technology*, 128, 33–43.
- Vasconez, J. P., Kantor, G. A., & Cheein, F. A. A. (2019). Human–robot interaction in agriculture: A survey and current challenges. *Biosystems Engineering*, 179, 35–48.
- Veeriah, V., Zhuang, N., & Qi, G. J. (2015). Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 4041–4049).
- Villani, V., Pini, F., Leali, F., & Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55, 248–266.
- Wang, P., Liu, H., Wang, L., & Gao, R. X. (2018a). Deep learning-based human motion recognition for predictive context-aware human–robot collaboration. *CIRP Annals*, 67(1), 17–20.
- Wang, K. J., Nguyen, P. H., Xue, J., & Wu, S. Y. (2018b). Technology portfolio adoption considering capacity planning under demand and technology uncertainty. *Journal of Manufacturing Systems*, 47, 1–11.
- Wilson, H. J., & Daugherty, P. R. (2018). *Collaborative intelligence: Humans and AI are joining forces*. Brighton: Harvard Business Review.
- Yao, G., Lei, T., & Zhong, J. (2019). A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters*, 118, 14–22.
- Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Cham: Springer.
- Zhang, J., Shao, K., & Luo, X. (2018). Small sample image recognition using improved Convolutional Neural Network. *Journal of Visual Communication and Image Representation*, 55, 640–647.
- Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.