Journal of Manufacturing Systems xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Transferable two-stream convolutional neural network for human action recognition

Qianqian Xiong[a], Jianjing Zhang[a], Peng Wang[b], Dongdong Liu[a,c], Robert X. Gao[a,*]

[a] Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, 44106, USA
[b] Department of Electrical and Computer Engineering and Department of Mechanical Engineering, University of Kentucky, Lexington, KY, 40506, USA
[c] School of Mechanical Electronic and Control Engineering, Beijing Jiaotong University, Beijing, 10004, China

ARTICLE INFO

Keywords:
Human-robot collaboration
Transfer learning
Temporal information

ABSTRACT

Human-Robot Collaboration (HRC), which enables a workspace where human and robot can dynamically and safely collaborate for improved operational efficiency, has been identified as a key element in smart manufacturing. Human action recognition plays a key role in the realization of HRC, as it helps identify current human action and provides the basis for future action prediction and robot planning. While Deep Learning (DL) has demonstrated great potential in advancing human action recognition, effectively leveraging the temporal information of human motions to improve the accuracy and robustness of action recognition has remained as a challenge. Furthermore, it is often difficult to obtain a large volume of data for DL network training and optimization, due to operational constraints in a realistic manufacturing setting. This paper presents an integrated method to address these two challenges, based on the optical flow and convolutional neural network (CNN)-based transfer learning. Specifically, optical flow images, which encode the temporal information of human motion, are extracted and serve as the input to a two-stream CNN structure for simultaneous parsing of spatial-temporal information of human motion. Subsequently, transfer learning is investigated to transfer the feature extraction capability of a pretrained CNN to manufacturing scenarios. Evaluation using engine block assembly confirmed the effectiveness of the developed method.

1. Introduction

Traditionally, robots in manufacturing are pre-programmed to do repetitive tasks, and are strictly separated from human workers due to safety concerns. As the modern manufacturing is transforming into the era of Industry 4.0, which is characterized by ubiquitous sensing, embedded intelligence, and the seamless integration of the cyber and physical worlds to further enhance productivity, efficiency, and agility while maintaining operation safety, robots are increasingly required to achieve a higher level of communication and cooperation with the human workers beyond simple co-existence [1,2].

In recent years, human-robot collaboration (HRC) has emerged as a key component for flexible and intelligent manufacturing [2]. Instead of strict separation between human and robot, HRC allows them to work as a team to collaboratively finish a task in the same workspace and dynamic decision-making beyond pre-programmed instructions [4].

An HRC system consists of four basic elements: perception, recognition, prediction and action [2], as shown in Fig. 1. Perception leverages sensor data to monitor the manufacturing workspace. Various sensing technologies have been developed, such as vision systems [5] and wearable devices [6]. They provide critical information regarding the state of the human worker in the workspace, allowing on-going human actions to be recognized [7]. The sequential patterns embedded within the human actions are then analyzed and serve as the basis for predicting future human actions [8,9]. The predicted future actions answer the question: “what will the worker do next?” and enable the robot to assist the worker in a pro-active, collaborative manner [9]. As the first step after acquiring the sensing data, human action recognition

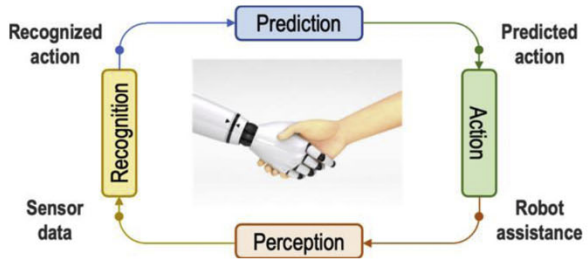


Fig. 1. Fundamental building blocks for realizing HRC.

image, which has been the predominant sensing modality in HRC research.

Human action recognition has traditionally involved two steps: feature extraction and action classification. Feature extraction refers to distilling essential information that is associated with human actions from raw sensing data. For video images, features are commonly extracted based on the distinct pixel intensity variations that encode the information for different image elements, such as curves and shapes. One of the most widely used methods is Scale Invariant Feature Transform (SIFT) [10], in which a series of local feature vectors are generated from captured images to characterize human action. These local features are invariant to transformations such as rotation and shift, and therefore are robust to image variations. Various enhancements to SIFT, such as Speeded Up Robust Features (SURF) and Oriented FAST and rotated BRIEF (ORB), have been developed [11,12]. Alternatively, contour of human poses has been investigated for feature extraction through comparison with pre-constructed models. For example, Belongie et al. introduced a shape context descriptor, which is able to detect shape contours that are similar to the reference models [13]. Skeleton model provides another method for human action characterization, where the information of human pose is reduced to the position and orientation of key body joints [14].

Once human action-related features are obtained, classifiers are deployed for action recognition. Among various classifiers, Hidden Markov Model (HMM) [15] has been widely investigated, which takes into account the transition probability among the atomic movements in human actions and uncertainties in sensing observations. In [16], HMM has been used to analyze the 3D depth information and the developed model is able to characterize both human motion and human-object interactions. Support Vector Machine (SVM) is another commonly used technique for classification. The basic idea of SVM is to find a hyperplane that effectively separates image feature-related to different human actions with the largest margin of separation [17].

One common limitation associated with these traditional techniques is that prior knowledge is required for feature extraction, which can be subjective. Recently, DL has emerged as a new paradigm to overcome these limitations as it is capable of learning features from data automatically in a supervised manner [18]. Successful applications of DL for various manufacturing-related tasks, such as machine fault diagnosis [19] and additive manufacturing process monitoring [20] have been reported, and convolutional neural networks (CNN) has shown to be a serve as input to the CNN.

Despite the progress made in human action recognition, multiple limitations remain. First, the conventional, single-stream CNN structure that only receives one type of input cannot simultaneously parse both spatial-temporal information of human actions. Although this problem can be alleviated by using frame stacks as the network input, it has been shown that this method is inferior to those that use hand-crafted features. The second limitation is that the DL-based method requires a large amount of training data for network weights optimization. However, training data is often difficult to obtain in the manufacturing environment, due to constraints such as continuous production scheduling. Consequently, only a small amount of data is typically accessed.

To tackle these limitations, a transferable two-stream CNN architecture consisting of spatial and temporal streams is proposed in this study. First, the method of optical flow [24] is investigated to extract temporal information from video images of human actions to complement spatial information embedded in the video images. The extracted spatial and temporal information are simultaneously parsed by a two-stream CNN structure for improved accuracy in human action recognition. Second, transfer learning, a technique that allows the transfer of a model learned within a source domain to be applied to a different target domain [25–27], is investigated. Specifically, the two-stream CNN is first pretrained on a large-scale open source dataset which consists of non-manufacturing specific human actions, which allows the network weights to be optimized and the feature extraction capability to be established. Subsequently, the pretrained model is transferred to recognize human actions in the target domain of assembly task where the training samples are limited. Prior studies have shown effectiveness of transfer learning in bearing condition monitoring and fault diagnosis with insufficient faulty data in the target domain [28]. Lastly, t-Distributed Stochastic Neighbor Embedding (t-SNE) [29] is investigated to evaluate the performance of the developed method, by visualizing the separation of the extracted features corresponding to different human actions.

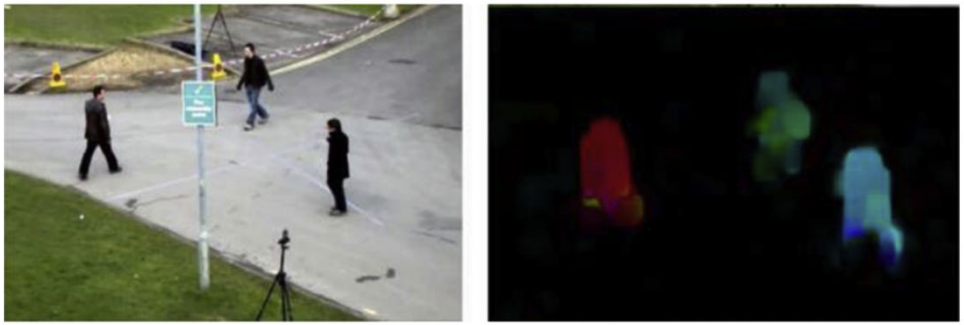
The rest of the paper is organized as follows. Section 2 presents the theoretical foundation for the developed method, whereas Section 3 describes the experimental evaluation and results discussion, using engine assembly as a representative manufacturing scenario. Conclusions are drawn in Section 4.

2. Theoretical foundation

In this section, the theoretical background of the techniques investigated in this research is presented. First, Section 2.1 presents the basics of optical flow, followed by the design of the two-stream CNN structure in Section 2.2. Section 2.3 introduces the mechanism of transfer learning, and its integration with t-SNE for performance evaluation is described in Section 2.4.

2.1. Optical flow

Video images consist of a large amount of information in the form of spatial-temporal pixel intensity variations. In general, it is not



(a) (b)

Fig. 2. Sample still frame (a) and optical flow (b) [24].

The optical flow algorithm calculates the pixel displacement vectors between two consecutive frames that are taken Δt apart. The corresponding pixels in two consecutive frames (before and after the displacement) have the same intensity, and their locations are denoted as (x, y) and $(x + \Delta x, y + \Delta y)$, respectively. Mathematically, this *Brightness Constancy Constraint* (BCC) is expressed as:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{1}$$

Assuming both the time interval Δt and movement Δx, Δy are small, this constraint can be represented by the *Taylor Series* as:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \varepsilon \tag{2}$$

where ε is a small number defined as the remainder of the series. Based on Eqs. (1) and (2), the following equation can be derived:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \tag{3}$$

By denoting $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$, $\frac{\partial I}{\partial t}$ as I_x, I_y, I_t, Eq. (3) is rewritten as:

$$I_x \Delta x + I_y \Delta y = -I_t \Delta t \tag{4}$$

In general, pixels in the immediate neighbourhood of a pixel can be assumed to move at the same velocity. As a result, the 3×3 region around the target pixel can be assumed to have the same displacement between the two consecutive frames. Therefore, by writing Eq. (4) for each pixel in the 3×3 region, the following set of equations can be obtained:

$$\begin{bmatrix} I_x(p1) & I_y(p1) \\ I_x(p2) & I_y(p2) \\ \vdots & \vdots \\ I_x(p9) & I_y(p9) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = -\Delta t \begin{bmatrix} I_t(p1) \\ I_t(p2) \\ \vdots \\ I_t(p) \end{bmatrix} \tag{5}$$

Eq. (5) can be solved using the least square method. By solving Eq. (5) for all 3×3 regions in two consecutive frames, optical flow images can be obtained.

2.2. Two-stream CNN

Two-stream CNN was first proposed by Simonyan et al. [32] in which each stream consists of a series of hierarchically arranged convolutional layers for image feature extraction. Specifically, the feature extraction step is achieved through sequential convolution between the kernels at each layer and the feature maps produced in the preceding layer. For the l^{th} layer with M input feature maps and N kernels, the j^{th} output feature map x_j^l is calculated as:

$$x_j^l = f\left(\sum_{i=1}^{M} x_i^{l-1} * k_{ij}^l + b_j^l\right), \; j = 1, \; ..., N \tag{6}$$

where x_i^{l-1} represents the i^{th} input feature map, k_{ij}^l denotes the j^{th} kernel to convolve with the i^{th} input feature map, b_j^l is the bias term, f denotes a non-linear function, and * denotes the convolution operation.

After the convolution operation, a pooling layer is often implemented as a sub-sampling operation [33]. Max pooling and average pooling are the two most common types of the pooling operation. Max pooling selects the maximum feature value from each local region and discarding the rest, while average pooling computes the mean feature value within each local region. Both methods can reduce the

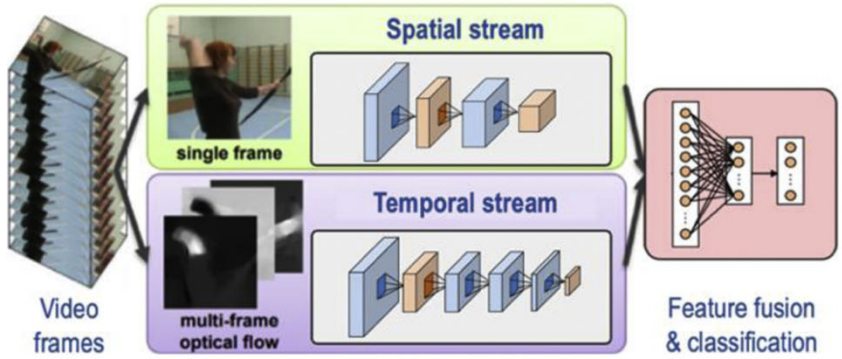


Fig. 4. Structure of two-stream Convolutional Neural Network [32].

dimensionality of the extracted features and thus improving the computational efficiency. Furthermore, they have also shown to reduce the sensitivity of features to small variations, such as pixel intensity change, to improve feature robustness [33]. Mathematically, the output feature maps of the l^{th} layer after pooling can be computed as:

$$x_j^l = f(\beta_j^l down(x_j^{l-1}) + b_j^l),\ j = 1,\ ...,M \tag{7}$$

where $down()$ is the sub-sampling function.

Through sequential operations of convolution and pooling, image features are gradually distilled to reflect the most relevant information to the specific task (e.g., human action recognition) [33].

In the context of human action recognition, the two-stream CNN consists of the spatial stream and temporal stream, as shown in Fig. 4. The spatial stream contains the spatial information from the still video frames, such as the static appearance of the workspace and human pose. The static appearance and human pose can provide useful clue for the action recognition. For example, the specific position of the human body in the workspace may strongly be more associated with certain actions than others, while the specific human pose may indicate the object the worker is handling. The architecture of the spatial stream is essentially a static image classifier, which is pretrained using a static image dataset in the presented research.

The temporal stream consists of a stack of consecutive optical flow frames describing a series of movements during a time period of fixed duration. By observing changes in the movement, temporal information can be extracted to complement the spatial information for more accurate human action recognition. In the presented research, the temporal stream is pretrained using an optical flow dataset processed from human action videos.

To determine relevant network parameters of the two-stream CNN, a parameter grid search is performed. Tables 1 and 2 respectively illustrate the selected combinations of network parameters for each of the two streams, which is obtained by comparing the classification accuracy of the tasks described later in Section 3. The dimensionality of the fused feature is $30 \times 40 \times 64$ (i.e., concatenation of two $30 \times 40 \times 32$ features). The structures of the two streams are illustrated in Figs. 5 and 6, respectively.

Table 2
Structure of the temporal Stream.

Layer	Kernel Size	Stride	Output size
Conv1 (ReLU)	9 × 9(64)	1	120 × 160 × 64
Max Pool	–	2	60 × 80 × 64
Conv2 (ReLU)	5 × 5(64)	1	60 × 80 × 64
Conv3 (ReLU)	3 × 3(64)	1	60 × 80 × 64
Conv4 (ReLU)	3 × 3(32)	1	60 × 80 × 32
Average Pool	–	2	30 × 40 × 32

under the assumption that the collected training data are sufficient to optimize the large amount of network parameters (e.g., weights). However, it is generally difficult in manufacturing settings to acquire sufficient data that contain information on the defects related to the machines or the processes, due to the fact that defect-involved operations, once detected, will be terminated to avoid damage to the machines and products.

Transfer learning refers to the technique that is capable of transferring the learned knowledge from a source domain to a related target domain [26]. If applied properly, it can alleviate the need for collecting a large amount of training data in the target domain and building a new model from scratch [26].

In this research, the transfer of feature extraction by CNN is explored. It is known that the working mechanism of CNN is to first extract low-level image features (such as edge and curve) by convolutional layers close to the input of the network and then assemble these features into high-level patterns in fully-connected layers at the output stage of the network, for purpose of classification. This implies that the initial layers in a CNN have a more generic feature extraction capability that can potentially be generalized across different application domains. In the developed transfer learning framework, the weights of the initial layers in the pretrained CNN are frozen and transferred (i.e., two convolutional layers and two pooling layers in the spatial stream, and four convolutional layers and two pooling layers in the temporal stream, respectively).

To realize human action recognition in the target domain (i.e., manufacturing), the fused features through the transferred layers are

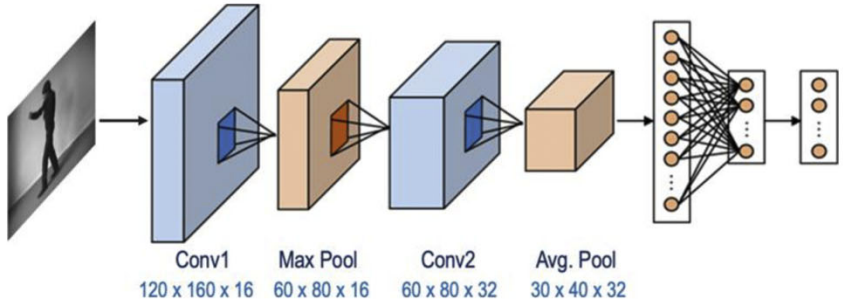


Fig. 5. Illustration of the spatial stream.

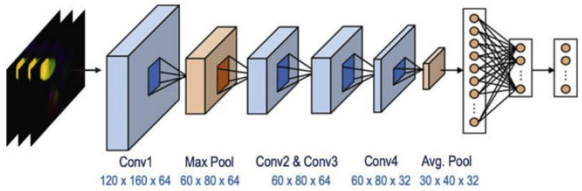


Fig. 6. Illustration of the temporal stream.

to visualize directly. To facilitate performance evaluation of the developed method in terms of the separability among features corresponding to different human actions, the method of t-SNE for visualizing data in a high-dimensional space is investigated [29].

t-SNE is an improved version of Stochastic Neighbor Embedding (SNE). The basic idea of SNE is to find a low-dimensional space representation of the complex data structure that is typically represented in the high-dimensional space for ease of visualization. The technique is based on the pair-wise data similarity between corresponding data points in the low and high dimensional space. To represent the similarities between data points x_i and x_j in a high-dimensional space, the conditional probabilities is computed as:

$$p_{j|i} = \frac{\exp\left(\frac{-\| x_i - x_j \|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(\frac{-\| x_i - x_k \|^2}{2\sigma_i^2}\right)} \tag{8}$$

where σ_i is the variance of the Gaussian distribution centered on x_i. In the low-dimensional space, the counterparts of x_i and x_j, namely y_i and y_j, are randomly assigned initially. The corresponding conditional probabilities can be expressed in a similar manner:

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)} \tag{9}$$

To allow the data points y's in the low-dimensional space to capture the structure of data points x's in the high-dimensional space, the difference between $p_{j|i}$ and $q_{j|i}$ needs to be minimized. The cost function representing the difference between $p_{i|j}$ and $q_{i|j}$ can be expressed as the summation of Kullback-Leibler (KL) divergence over all data points:

$$C = \sum_i KL(P_i \| Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}} \tag{10}$$

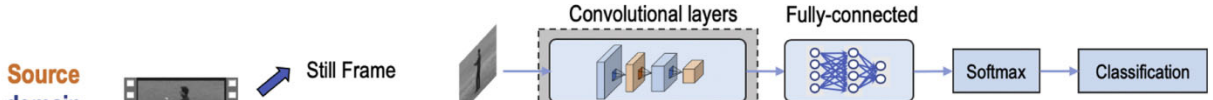
t-SNE utilizes student t-distribution instead of Gaussian distribution when computing the conditional probabilities. The advantage of using t-distribution is that it alleviates issues such as points clustering in SNE and facilitates the optimization of the loss function [29].

3. Experimental evaluation and discussion

The developed method is experimentally evaluated using engine assembly as a manufacturing scenario.

3.1. Experimental setup

The developed transferable two-stream CNN model is comprised of three parts: spatial stream, temporal stream, and classifier. The spatial stream and temporal stream, which consist of convolutional and pooling layers, work as feature extractors. Features extracted by both streams are fused before fed into the classifier, which consists of fully-connected layers and a softmax layer for classification. Specifically, still frames and optical flow images are extracted from open source human action videos to build the pretraining dataset (source domain), which are then utilized to pretrain the spatial and temporal stream of CNN. Then, the convolutional and pooling layers in the pretrained model are transferred to capture features from the assembly dataset (target domain). Finally, the weights in the fully-connected layers are fine-tuned for action recognition in the target domain. The process is illustrated in Fig. 7.



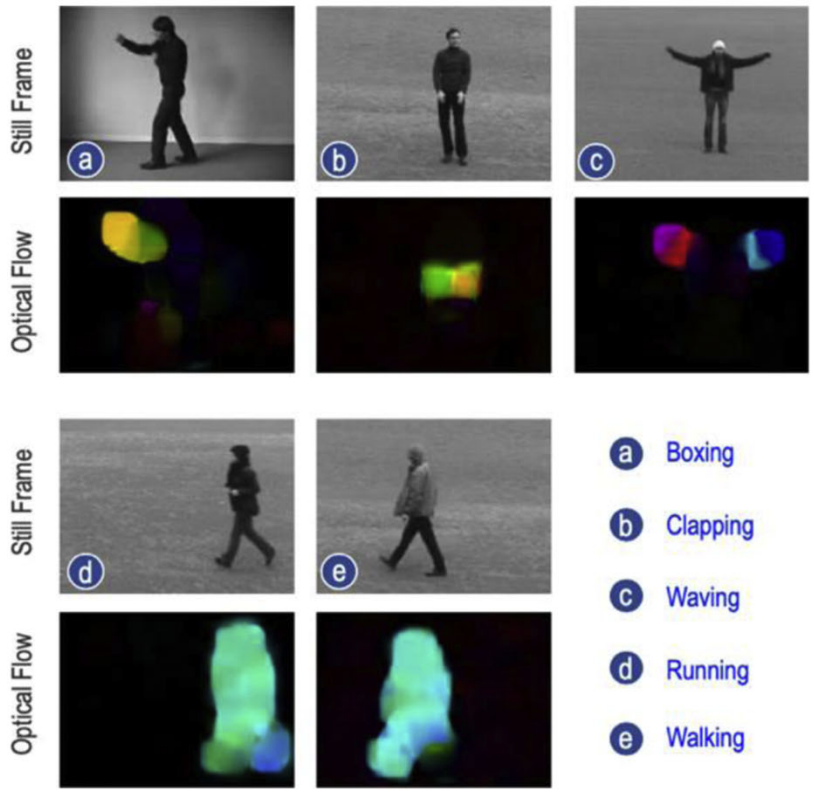
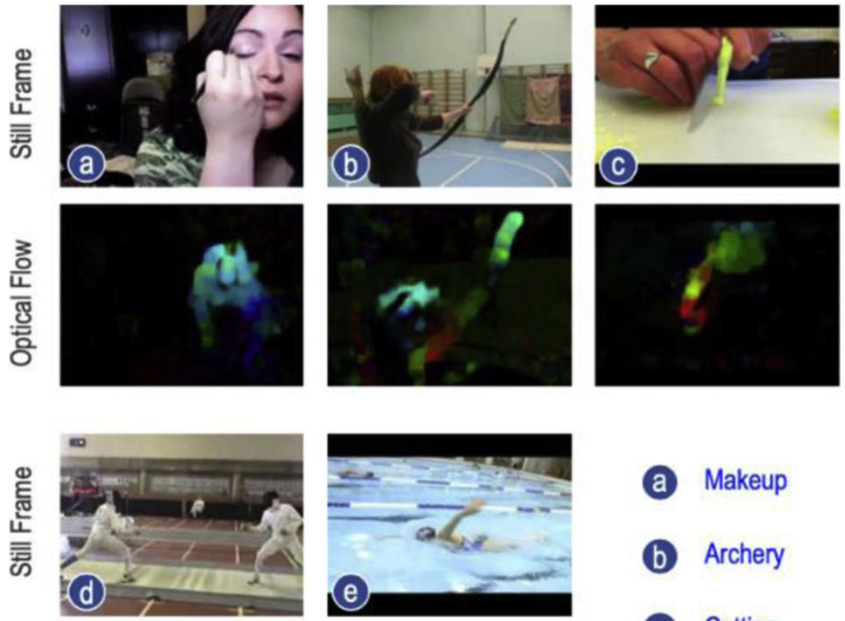


Fig. 8. Pretraining data from KTH [34].



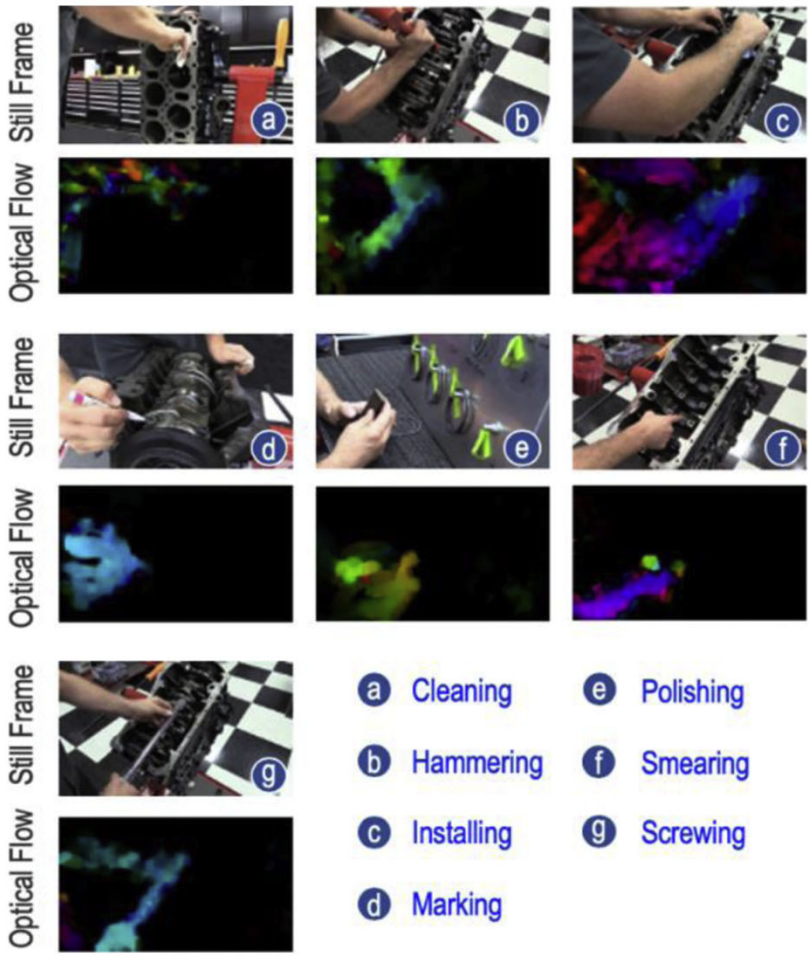


Fig. 10. Image data for engine block assembly [36].

Table 3
Accuracy of pretraining dataset.

	Mean	Std. Dev.
Spatial Stream	83.06 %	4.19×10^{-2}
Temporal Stream	66.37 %	4.48×10^{-2}
Two Stream	88.31 %	3.27×10^{-2}

Table 4
Accuracy of assembly dataset.

	Mean	Std. Dev.
Spatial Stream	99.95 %	0
Temporal Stream	72.88 %	4.48×10^{-2}
Two Stream	100.00 %	0

3.1.2. Pretraining

The objective of pretraining is to develop a model that can recognize human actions by classifying the related images. Two CNNs, one for the spatial stream and another for the temporal stream, have been constructed to classify the images into ten different categories. The spatial stream is pretrained using individual still frames while the temporal stream is pretrained using stacks of optical flow images. Details on the CNN structures of the two streams are illustrated in Figs. 5 and 6 respectively. The pretraining dataset has been randomly split into two sub-datasets for the purpose of training and testing, respectively. A total of 85 % of the samples in the pretraining dataset are used as the training set, and the remaining 15 % are used as the testing set.

To avoid overfitting, which is reflected in the significantly lower performance in network testing than network training, which results from the network parameters (e.g., weights) being over-sensitive to small variations in the training data (e.g., due to noise) and thus fail to

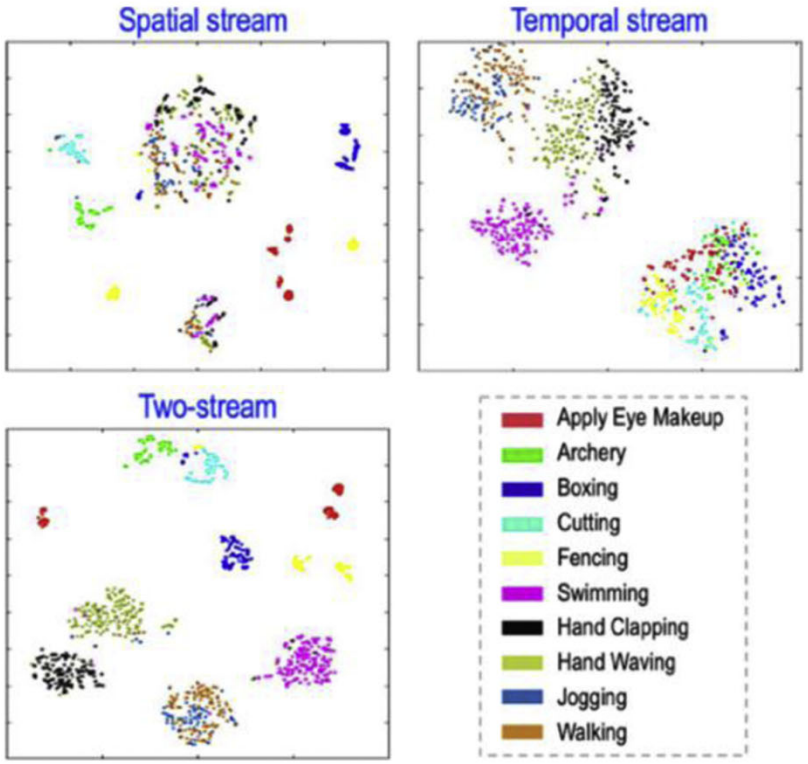


Fig. 11. t-SNE results of pretraining dataset among spatial stream, temporal stream and two-stream network models.

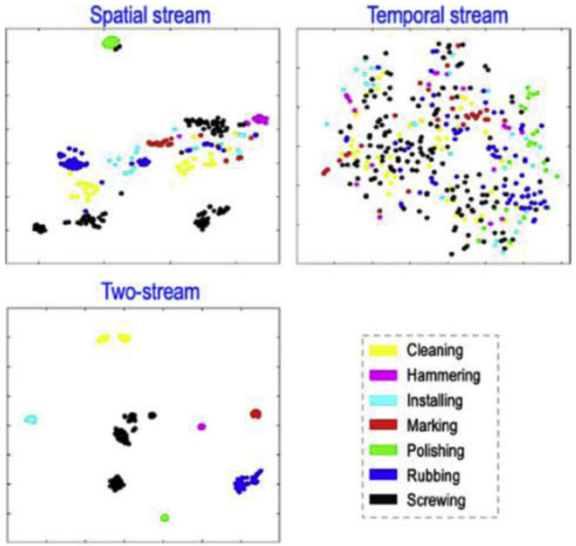


Fig. 12. t-SNE results of assembly dataset among spatial stream, temporal stream and two-stream network models.

accuracy when using the spatial stream alone (83.06 %). It is also considerably higher than the accuracy using only the temporal stream (88.31 % vs. 66.37 %). This confirms the importance of using both the spatial and temporal information for improved action recognition performance. In addition, the standard deviation of two-stream model results (0.0327) is lower than those from the two single-stream models (0.0419 and 0.0448), respectively, suggesting that the two-stream model is more robust to data variations.

In Table 4, it is seen that the mean classification accuracy of the two-stream model has reached 100 %, indicating that the transferred model has effectively captured the action-related image patterns from the assembly dataset, even though the feature extraction capability is obtained from the pretraining dataset, which is not specific to the assembly task. This suggests that the low-level feature extraction mechanism in the CNN is indeed generic and can be effectively generalized among different action recognition tasks. It is also seen that the two-stream model has the best performance as compared to the two single-stream models after transfer, although the spatial stream also achieved good recognition accuracy (99.95 %).

To evaluate the performance of the models beyond classification accuracy, t-SNE is deployed to map the extracted high-dimensional features into a two-dimensional space to visualize the feature separability. The larger the separation, the better the effectiveness of the

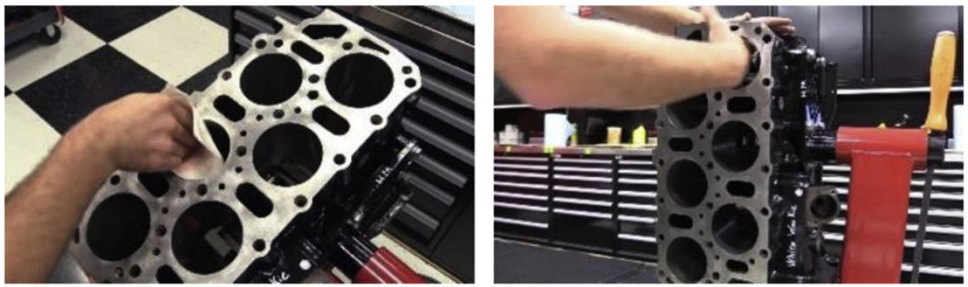
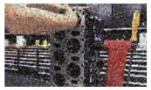


Fig. 13. Cleaning with (a) lying engine block; (b) standing engine block.

Table 5
Accuracy of assembly dataset under different noise densities.

Noise Density	Sample Images (Cleaning)	PSNR (dB)	Mean	Std. Dev.
0		–	100.00 %	0
0.2		15.45	100.00 %	0
0.4		12.41	100.00 %	0
0.6		10.55	99.66 %	4.6×10^{-3}
0.8		9.20	97.12 %	1.4×10^{-2}

temporal stream, as shown by the two clusters with a clearly defined border. This separation is the most obvious in the two-stream model, as the two clusters are completely separated. As another example, the temporal stream model has failed to distinguish “boxing” from “archery” as both actions are finished with the movement of the arm. However, by considering the additional spatial information, “boxing” and “archery” are successfully separated in the two-stream models.

Fig. 12 shows the visualization of features of the assembly dataset. It is seen the performance of the two-stream model is better than either the spatial stream model or the temporal stream model in terms of clearly separating the clusters of different human assembly actions. The separation in the spatial stream model is less obvious. Quantitatively, the mean pair-wise distance between the centroids among different clusters for the two-stream model is about 4 times longer than that of the spatial stream model. Because of the unsatisfactory clustering separation in the temporal stream model, its mean pair-wise centroid distance is not computed.

the room light conditions, noise with different densities is progressively added into the raw images of all the seven assembly actions. As a quantification measure, noise density is used, which refers to the ratio of the number of affected image pixels (by randomly setting their values to either 0 or 255, thereby removing their contribution to image recognition) to the total number of pixels. As an example, a noise density of 0.6 means that 60 % of the pixels are affected by noise.

Digital cameras are generally able to adapt to the dimming light condition by increasing the “brightness level”, which however generates image noise as a trade-off. By progressively adding noise levels, as reflected in the increased noise density values, the effect of varying light conditions can be investigated. Furthermore, the peak signal-to-noise ratio (PSNR) values of the images with different noise densities are calculated. PSNR is expressed as the ratio of the maximum possible value of a signal to the power of distorting noise that affects the quality of its representation [37], and is computed as:

$$PSNR = 10 \times \log_{10}\left(\frac{(2^n - 1)^2}{MSE}\right) \tag{11}$$

where MSE is the deviation of the noise image from the raw image computed as mean square error, and n is determined by the image datatype (e.g., for uint8, n is 8).

Table 5 shows the sample images ("cleaning" operation) with different noise densities and the classification results under the corresponding noise contaminations based on 25 random experiment tests. It is seen that the developed method has been able to correctly identify all seven human assembly actions until the noise density has increased to 0.6 when the mean classification accuracy has dropped to 99.66 %. With a noise density of 0.8, the two-stream CNN has still achieved a classification rate of 97.12 %. These observations indicate that the developed method is robust to the image noise and consequently, the varying light conditions.

4. Conclusion

To improve the accuracy in human action recognition for reliable human-robot collaboration, a hybrid method that integrates optical flow with transferable two-stream CNN has been developed. This method enables the utilization of temporal information embedded in

utilizing spatial and temporal information alone, by up to 27 %.

- Visualization of the extracted features has indicated that the spatial and temporal information complement each other when determining different human actions. Furthermore, despite the similar recognition accuracy achieved by both the spatial CNN and two-stream CNN, the features obtained from the two-stream CNN are four times more separated as compared to the spatial CNN.
- Transfer learning is effective in adapting the feature extraction capability of the network from a source domain to the target domain, as reflected by the 100 % action recognition accuracy in the engine assembly evaluation.
- The developed method has shown to be robust under variations in the assembly configuration and noisy video footage, as reflected in the 97.12 % recognition accuracy under the noise density level of 0.8.

Future work will address the theoretical bound of data transferability and expand the developed method as a trustworthy technique in HRC for broader applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Support by National Science Foundation under award CMMI-1830295 is gratefully appreciated.

References

[1] Liu H, Wang L. Remote human–robot collaboration: a cyber–physical system application for hazard manufacturing environment. J Manuf Syst 2020;54:24–34. https://doi.org/10.1016/j.jmsy.2019.11.001.
[2] Wang L, Gao R, Váncza J, Krüger J, Wang XVV, Makris S, et al. Symbiotic human-robot collaborative assembly. CIRP Ann 2019;68:701–26. https://doi.org/10.1016/j.cirp.2019.05.002.
[3] Wang P, Gao R, Fan Z. Cloud computing for cloud manufacturing: benefits and limitations. J Manuf Sci Eng Trans ASME 2015;137. https://doi.org/10.1115/1.4030209.
[4] Krüger J, Lien TK, Verl A. Cooperation of human and machines in assembly lines. CIRP Ann 2009;58:628–46. https://doi.org/10.1016/j.cirp.2009.09.009.
[5] Pérez L, Rodríguez Í, Rodríguez N, Usamentiaga R, García DF. Robot guidance using machine vision techniques in industrial environments: a comparative review. Sensors (Switzerland) 2016;16. https://doi.org/10.3390/s16030335.
[6] Sundaram S, Kellnhofer P, Li Y, Zhu JY, Torralba A, Matusik W. Learning the signatures of the human grasp using a scalable tactile glove. Nature 2019;569:698–702. https://doi.org/10.1038/s41586-019-1234-z.
[7] Kamel A, Sheng B, Yang P, Li P, Shen R, Feng DD. Deep convolutional neural networks for human action recognition using depth maps and postures. IEEE Trans Syst Man Cybern Syst 2019;49:1806–19. https://doi.org/10.1109/TSMC.2018.2850149.
[8] Zhang J, Liu H, Chang Q, Wang L, Gao R. Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. CIRP Ann 2020;69(1). Accepted.
[9] Maeda G, Ewerton M, Neumann G, Lioutikov R, Peters J. Phase estimation for fast action recognition and trajectory generation in human–robot collaboration. Int J Rob Res 2017. https://doi.org/10.1177/0278364917693927.
[10] Lowe DG. Object recognition from local scale-invariant features. Proc IEEE Int Conf Comput Vis 1999;2:1150–7. https://doi.org/10.1109/iccv.1999.790410.
[11] Bay H, Ess A, Tuytelaars T, Van Gool L. Speeded-up robust features (SURF). Comput Vis Image Underst 2008;110:346–59. https://doi.org/10.1016/j.cviu.2007.09.014.
[12] Rublee E, Rabaud V, Konolige K, Bradski G. ORB: An efficient alternative to SIFT or SURF. Proc IEEE Int Conf Comput Vis 2011:2564–71. https://doi.org/10.1109/ICCV.2011.6126544.
[13] Belongie S, Malik J, Puzicha J. Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell 2002;24:509–22. https://doi.org/10.1109/34.993558.
[14] Han J, Shao L, Xu D, Shotton J. Enhanced computer vision with Microsoft Kinect sensor: a review. IEEE Trans Cybern 2013;43:1318–34. https://doi.org/10.1109/TCYB.2013.2265378.
[15] Liu H, Wang L. Human motion prediction for human-robot collaboration. J Manuf Syst 2017;44:287–94. https://doi.org/10.1016/j.jmsy.2017.04.009.
[16] Wilson AD, Bobick AF. Parametric hidden Markov models for gesture recognition. IEEE Trans Pattern Anal Mach Intell 1999;21:884–900. https://doi.org/10.1109/34.790429.
[17] Sharp M, Ak R, Hedberg T. A survey of the advancing use and development of machine learning in smart manufacturing. J Manuf Syst 2018;48:170–9. https://doi.org/10.1016/j.jmsy.2018.02.004.
[18] Wang J, Ma Y, Zhang L, Gao R, Wu D. Deep learning for smart manufacturing: methods and applications. J Manuf Syst 2018;48:144–56. https://doi.org/10.1016/j.jmsy.2018.01.003.
[19] Wang P, Ananya Yan R, Gao R. Virtualization and deep recognition for system fault classification. J Manuf Syst 2017;44:310–6. https://doi.org/10.1016/j.jmsy.2017.04.012.
[20] Caggiano A, Zhang J, Alfieri V, Caiazzo F, Gao R, Teti R. Machine learning-based image processing for on-line defect recognition in additive manufacturing. CIRP Ann 2019;68:451–4. https://doi.org/10.1016/j.cirp.2019/03.021.
[21] Ijjina EP, Chalavadi KM. Human action recognition in RGB-D videos using motion sequence information and deep learning. Pattern Recognit 2017;72:504–16. https://doi.org/10.1016/j.patcog.2017.07.013.
[22] Wang P, Liu H, Wang L, Gao R. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. CIRP Ann 2018;67:17–20. https://doi.org/10.1016/j.cirp.2018.04.066.
[23] Núñez JC, Cabido R, Pantrigo JJ, Montemayor AS, Vélez JF. Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition. Pattern Recognit 2018;76:80–94. https://doi.org/10.1016/j.patcog.2017.10.033.
[24] Optical flow. 2020https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html.
[25] Torrey L, Shavlik J. Transfer learning. Handb Res Mach Learn Appl Trends Algorithms Methods Tech 2009:242–64. https://doi.org/10.4018/978-1-60566-766-9.ch011.
[26] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2010;22:1345–59. https://doi.org/10.11009/tkde.2009.191.
[27] Yan R, Shen F, Sun C, Chen X. Knowledge transfer for rotary machine fault diagnosis. IEEE Sens J 2019:1–19. https://doi.org/10.1109/jsen/2019.2949057.
[28] Wang P, Gao R. Transfer learning for enhanced machine fault diagnosis in manufacturing. CIRP Ann 2020;69(1). Accepted.
[29] Van Der Maaten L, Hinton G. Visualizing data using t-SNE. J Mach Learn Res 2008;9:2579–625.
[30] Turaga P, Chellappa R, Veeraraghavan A. Advances in video-based human activity analysis: challenges and approaches. Adv Comput 2010;80:237–90. https://doi.org/10.1016/s0065-2458(10)80007-5.
[31] Sánchez J, Salgado A, Monzón N. Computing inverse optical flow. Pattern Recognit Lett 2015;52:32–9. https://doi.org/10.1016/j.patrec.2014.09.009.
[32] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. Adv Neural Inf Proc Syst 2014;1:568–76.
[33] Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012:1097–105.
[34] Recognition of human actions. 2020https://www.nada.kth.se/cvap/actions.
[35] Soomro K, Zamir AR, Shah M. UCF101: a dataset of 101 human actions classes from videos in the wild. 2012. arXiv:1212.0402.
[36] How to assemble an engine block. 2020. cQH0YY https://www.youtube.com/watch?v=zPAEl.
[37] Peak signal-to-Noise ratio as an image quality metric, national instruments. 2020https://www.ni.com/en-us/innovations/white-papers/11/peak-signal-to-noise-ratio-as-an-image-quality-metric.html.