Summer 2020

Human behavior understanding for worker-centered intelligent manufacturing

Wenjin Tao

Department: Mechanical and Aerospace Engineering

HUMAN BEHAVIOR UNDERSTANDING FOR WORKER-CENTERED INTELLIGENT MANUFACTURING

by

WENJIN TAO

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

MECHANICAL ENGINEERING

2020

Approved by

Dr. Ming C. Leu, Advisor
Dr. Zhaozheng Yin
Dr. Ruwen Qin
Dr. Zhihai He
Dr. Frank Liou
Dr. K. Chandrashekhara

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following five articles, formatted in the style used by the Missouri University of Science and Technology:

Paper I, found on pages 8-21, has been published in Manufacturing Letters.

Paper II, found on pages 22-54, has been published in Engineering Applications of Artificial Intelligence.

Paper III, found on pages 55-88, has been submitted to Engineering Applications of Artificial Intelligence.

Paper IV, found on pages 89-119, has been submitted to Information Fusion.

Paper V, found on pages 120-135, has been accepted for publication in Procedia Manufacturing.

ABSTRACT

In a worker-centered intelligent manufacturing system, sensing and understanding of the worker's behavior are the primary tasks, which are essential for automatic performance evaluation & optimization, intelligent training & assistance, and human-robot collaboration. In this study, a worker-centered training & assistant system is proposed for intelligent manufacturing, which is featured with self-awareness and active-guidance. To understand the hand behavior, a method is proposed for complex hand gesture recognition using Convolutional Neural Networks (CNN) with multiview augmentation and inference fusion, from depth images captured by Microsoft Kinect. To sense and understand the worker in a more comprehensive way, a multi-modal approach is proposed for worker activity recognition using Inertial Measurement Unit (IMU) signals obtained from a Myo armband and videos from a visual camera. To automatically learn the importance of different sensors, a novel attention-based approach is proposed to human activity recognition using multiple IMU sensors worn at different body locations. To deploy the developed algorithms to the factory floor, a real-time assembly operation recognition system is proposed with fog computing and transfer learning. The proposed worker-centered training & assistant system has been validated and demonstrated the feasibility and great potential for applying to the manufacturing industry for frontline workers. Our developed approaches have been evaluated: 1) the multi-view approach outperforms the state-of-the-arts on two public benchmark datasets, 2) the multi-modal approach achieves an accuracy of 97% on a worker activity dataset including 6 activities and achieves the best performance on a public dataset, 3) the attention-based method outperforms the state-of-the-art methods on five publicly available datasets, and 4) the developed transfer learning model achieves a real-time recognition accuracy of 95% on a dataset including 10 worker operations.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor, Dr. Ming C. Leu. The success of my research and completion of my dissertation would not have been possible without his insightful suggestions and unwavering support during my Ph.D. study at Missouri University of Science and Technology. His rigorous attitude towards research and teaching will have a significant influence on my future career. It has been a great honor and pleasure for me to have worked with him.

I would also like to extend my sincere gratitude to all my dissertation committee members, Dr. Zhaozheng Yin, Dr. Runwen Qin, Dr. Frank Liou, Dr. Zhihai He and Dr. K. Chandrashekhara. Without their guidance and helpful advice, it would have been impossible for me to complete my dissertation.

I'm extremely grateful to my labmates and friends for their support during my study in Rolla. Also, I would like to thank my colleagues in Industrial AI group at Foxconn Industrial Internet during my CO-OP in Milwaukee.

Last but not least, thanks to my family for being with me.

TABLE OF CONTENTS

Page

PAPER

LIST OF ILLUSTRATIONS

LIST OF TABLES

PAPER V

SECTION

1. INTRODUCTION

1.1. BACKGROUND

Industrial big data has been increasingly accessible and affordable, benefiting from the availability of low-cost sensors and the development of Internet-of-Things (IoT) technologies [12, 21], which builds up the data foundation for advanced manufacturing. A variety of methods and algorithms have been developed to learn valuable information from the data, and to make the manufacturing more intelligent [24]. With the recent fast growing of Artificial Intelligence (AI) technologies, especially deep learning [20] and reinforcement learning [16] methods, AI-boosted manufacturing has been increasingly attractive in both the scientific research and industrial applications.

In an intelligent manufacturing system involving workers, recognition of the worker's activity is one of the primary tasks. It can be used for quantification and evaluation of the worker's performance, as well as to provide onsite instructions with augmented reality. Also, worker activity recognition is crucial for human-robot interaction and collaboration. It is essential for developing human-centered intelligent manufacturing systems. Furthermore, it can be used for knowledge/skill transfer between experienced workers and new workers.

1.2. WORKER-CENTERED SENSING

The first step for human behavior recognition is to sense the human's activity. In this section, different sensing technologies are discussed. Considering their wearablity, they can be grouped as ambient sensing and wearable sensing technologies.

1.2.1. Ambient Sensing. Ambient sensors are deployed in the environment to sense the subject in a passive manner. For example, optic cameras can be used to capture RGB images on human subjects; Depth cameras such as a Microsoft Kinect or Lidar (light detection and ranging) sensors can be applied to sense human objects in the 3D space; Infrared cameras can detect the subject in a dark environment; Pressure sensing mats can be used to capture human's standing states; WiFi signals also have been used for HAR [14]. Ambient sensing can collect a large amount of data without interfering the subject's activity.

1.2.2. Wearable Sensing. Nevertheless, ambient sensors require complex setups and their performance can be affected dramatically by occlusion issues, which are the main challenges in implementing ambient sensing. Also, it becomes more difficult when capturing a subject's outdoor activities. To compensate for these limitations, wearable sensing can be applied. Wearable sensor based activity recognition has captured growing attention nowadays because of the pervasiveness of mobile devices (e.g., smart phones and smart watches), which are embedded with various sensors such as IMU (Inertial Measurement Unit) sensors, heart rate sensors, and ECG (Electrocardiogram) sensors.

1.2.3. Pros and Cons. Vision-based sensors are most widely used for ambient sensing purpose. In the computer vision area, image/video-based human activity recognition with deep learning has been intensively studied in recent years and unprecedented progress has been made [2, 13]. However, vision-based recognition suffers from the occlusion issue, which affects the recognition accuracy. Wearable devices, such as an armband embedded with an Inertial Measurement Unit (IMU), directly sense the movement of human body, which can provide information on the body status. In addition, there are a lot of inexpensive wearable devices in the market, such as Myo armbands [32] and smartphones, which are widely used in activity recognition tasks. Wearable devices are directly attached to the human body and thus do not have the occlusion issue. However, a wearable device can only sense the human body activity locally, and it is challenging to precisely recognize an activity

involving multiple body parts. Although multiple devices can be applied to simultaneously sense the activity globally, it makes the system cumbersome and brings discomfort to the user.

1.3. DATA-DRIVEN WORKER BEHAVIOR UNDERSTANDING

In general, the activity recognition task can be broken down into two subtasks: feature extraction and subsequent multiclass classification. To extract more discriminative features, various methods have been applied to the raw signals in the time or frequency domain, e.g., mean, correlation, and Principal Component Analysis [1, 3, 25, 28]. Different classifiers have been explored on the features for activity recognition, such as the Support Vector Machine [1, 3], Random Forest, K-Nearest Neighbors, Linear Discriminant Analysis [25], and Hidden Markov Model [28]. To effectively learn the most discriminative features, Jiang et al. [15] proposed a method based on Convolutional Neural Networks (CNN). They assembled the raw IMU signals into an activity image, which enabled the CNN model to automatically learn the discriminative features from the activity image for classification.

The critical factor attributed to the success of IMU data-driven activity recognition is to seek an effective representation of the time-series IMU signals. The most widely used approaches fall into two categories: handcrafted feature design and automatic feature learning.

1.3.1. Hand-Crafted Feature Design. It is intuitive to manually pick statistical attributes (e.g., means) or quantity distributions (e.g., magnitude histograms) from IMU signals [10]. For example, Anguita et al. [1] designed as many as 341 features from 3-axis IMU signals while Hammerla et al. [8] preserved the statistical characteristics of IMU data using their empirical cumulative distributions. Xu el al. [35] proposed a multi-level feature learning framework which consists of the signal-based, components-based and sematic-based information for activity recognition. However, handcrafted feature design is mostly

driven by the domain knowledge, prior experience and experimental validation, thus it is possible to neglect some useful features in this manner. In addition, a pre-defined feature extraction mechanism trained on a specific scenario might not work well on other scenarios with different sets of activities to be recognized. That is, those hand-crafted features in the literature might not be transferrable to new application domains, which further makes the feature design time-consuming and labor-costly.

1.3.2. Automatic Feature Learning. The drawbacks of handcrafted features motivate researchers to explore automatic feature learning [15][11]. Deep Convolutional Neural Network (DCNN), as one of the most effective deep learning models, attracts attentions in the mobile sensing domain considering it has achieved the superior performance in other research fields such as computer vision [18] and speech recognition [23]. To improve the accuracy of sensor-based activity recognition, Zeng et al. [37] designed a tri-thread DCNN architecture with the three inputs corresponding to the tri-axis accelerometry data, thus the inputs are one-dimensional time-series signals. To enhance the ability for feature learning, Duffner et al. [6] and Ha et al. [7] took as input the two-dimensional matrix obtained by stacking IMU signals. In order for further accuracy improvement, Ravi et al. [27] combined features learned from the deep model with complementary information from a set of hand-crafted features. In addition, Lane et al. [19] looked into this problem in a practical way and showed the application of deep learning to mobile sensing domain is hardware-efficient and can scale up to a large number of inference classes.

In short, the input to the deep learning network and the architecture of the deep learning model itself are two key factors to the success of automatic feature learning. The input is of great significance because a good representation of the IMU signals can make it easier for automatic learning. In the previous work, IMU signals are directly fed into the DCNN architecture and this simple and raw input may not be a good representation of IMU signals because each value of the raw time-series signals is less meaningful if we do not consider the statisctic property of the whole signals.

In terms of the design of deep learning architecture, the aforementioned simple input restricts the depth of the deep model, limiting the capability to find discriminative features. For instance, the input in [36] is a small 3×30 matrix and there are only two convolutional layers in the architecture. Additionally, the tri-axis accelerometry signals are convolved with one-dimensional kernels in the deep model independently, thus the correlation among different signals is not taken into enough consideration.

1.3.3. Self-Attention Mechanisms. Just like humans can allocate different amount of attention to different aspects when performing a complex task, self-attention mechanisms can model attentions for deep neural networks and have been widely applied in many deep learning tasks [5]. The self-attention mechanism was proposed in [33] for machine translation tasks, in order to distribute different attention over words in a sentence. From then on, attention mechanisms have been increasingly popular in natural language processing (NLP) and computer vision fields, where multiple sources with different importance are involved. For example, Chen et al. [4] used spatial and channel-wise attention for image captioning, and He et al. [9] applied attention in both the spatial and temporal domains for HAR from videos.

1.3.4. Activity Recognition in Manufacturing Fields. Worker activity recognition in the manufacturing area is still an emerging topic and few studies have been made. Stiefmeire et al. [29] utilized ultrasonic and IMU sensors for worker activity recognition in a bicycle maintenance scenario using a Hidden Markov Model classifier. Later they proposed a string-matching based segmentation and classification method using multiple IMU sensors for recognizing worker activity in car manufacturing tasks [30, 31]. Koskimaki et al. [17] used a wrist-worn IMU sensor to capture the arm movement and a K-Nearest Neighbor model to classify five activities for industrial assembly lines. Maekawa et al. [22] proposed an unsupervised measurement method for lead time estimation of factory work using signals from a smartwatch with an IMU sensor. Recently, deep learning methods have been introduced to recognize worker activity in human-robot collaboration studies [26, 34].

1.3.5. Technology Gap. Few attempts have been made for the worker activity recognition in the manufacturing field, and most of them only use single sensing modality, which cannot guarantee robust recognition under various circumstances.

1.4. OBJECTIVES

The overall objective of this dissertation study is to to achieve an effective and efficient understanding of the worker's behavior on the factory floor, which provides the foundation for worker-centered intelligent manufacturing. A few fundamental questions need to be answered:

1. What are the desirable types of sensors to sense the workers in the manufacturing context?
2. How to integrate and fuse the data from multi-modal signals?
3. How to integrate and fuse the data from multiple sensors?

To answer the above mentioned questions, some fundamental research has been performed in this dissertation study as follows:

1. A multi-modal sensing system has been developed.
2. Algorithms for multi-modal signal fusion have been developed.
3. Algorithms for multi-sensor fusion have been developed.

A set of underlying fundamental challenges are: 1) complexity and uncertainty of worker activity, due to the high interclass similarities, high interclass similarities, large intraclass variations, large intersubject variations, and constant occlusions; 2) complexity of multi-source and heterogeneous sensing and modeling; and 3) complexity for human-object interaction understanding.

1.5. ORGANIZATION OF DISSERTATION

In this dissertation, five papers are included.

1. Paper I: A worker-centered training & assistant system is proposed for intelligent manufacturing, which is featured with self-awareness and active-guidance.

2. Paper II: To understand the hand behavior, a method is proposed for complex hand gesture recognition using Convolutional Neural Networks (CNN) with multiview augmentation and inference fusion, from depth images captured by Microsoft Kinect.

3. Paper III: To sense and understand the worker in a more comprehensive way, a multi-modal approach is proposed for worker activity recognition using Inertial Measurement Unit (IMU) signals obtained from a Myo armband and videos from a visual camera, where four different modalities are applied.

4. Paper IV: To learn the importance of different sensors, a novel attention-based approach is proposed to human activity recognition using multiple IMU sensors worn at different body locations.

5. Paper V: A fog computing framework is proposed for assembly operation recognition, which brings computing power close to the data source in order to achieve real-time recognition.

PAPER

I. A SELF-AWARE AND ACTIVE-GUIDING TRAINING & ASSISTANT SYSTEM FOR WORKER-CENTERED INTELLIGENT MANUFACTURING

Wenjin Tao[a], Ze-Hao Lai[a], Ming C. Leu[a], Zhaozheng Yin[b], Ruwen Qin[a]

[a]Missouri University of Science and Technology, Rolla, MO 65409, USA

[b]Stony Brook University, Stony Brook, NY 11794, USA

ABSTRACT

Training and on-site assistance is critical to help workers master required skills, improve worker productivity, and guarantee the product quality. Traditional training methods lack worker-centered considerations that are particularly in need when workers are facing ever-changing demands. In this study, we propose a worker-centered training & assistant system for intelligent manufacturing, which is featured with self-awareness and active-guidance. Multi-modal sensing techniques are applied to perceive each individual worker and a deep learning approach is developed to understand the worker's behavior and intention. Moreover, an object detection algorithm is implemented to identify the parts/tools the worker is interacting with. Then the worker's current state is inferred and used for quantifying and assessing the worker performance, from which the worker's potential guidance demands are analyzed. Furthermore, onsite guidance with multi-modal augmented reality is provided actively and continuously during the operational process. Two case studies are used to demonstrate the feasibility and great potential of our proposed approach and system for applying to the manufacturing industry for frontline workers.

Keywords: Intelligent manufacturing; Deep learning; Augmented reality; Cyber-physical system; Smart manufacturing

1. INTRODUCTION

Cyber-Physical Systems (CPS) have allowed the traditional manufacturing to enter into a new era, which is currently further boosted by Artificial Intelligence (AI) technologies, such as machine learning and deep learning, towards intelligent manufacturing [8, 22]. To meet the fast-growing consumer demands for highly-customized, high-quality products, manufacturers must make their manufacturing systems more flexible and efficient and, meanwhile, ensure that workers in the systems are agile and highly skilled. Workforce training and on-site assistance is essential to help workers learn desired skills, improve worker productivity, reduce the rate of rejects, and guarantee the product quality. Therefore, how to train and assist the workforce flexibly, efficiently and effectively is one of the critical factors contributing to a company's market success. Traditionally, operational instructions are provided in a lecture-based manner or a mentor-based manner. However, these methods have some limitations. For example, the lecture-based training can simultaneously teach lots of workers but is lack of immediate interaction. While it is more interactive and can have real-time communications, the mentor-based training is more costly and inefficient. For further evaluation of the worker's performance and optimization of the operational workflow, a time-motion study is often applied. Nevertheless, it requires a direct and continuous observation of the task and manual analysis for each operational step, which is time-consuming and lack of flexibiliy [5]. To provide the instructional information more interactively and immersively, Augmented Reality (AR) technologies have been widely deployed in industry, and it has been proven to be an excellent interface for presenting multi-media information to workers [4, 11, 12, 17, 21]. However, existing AR methods

often use pre-defined scripts to control how the instructional information is provided and they lack worker-centered considerations that are particularly in need when workers are facing ever-changing demands.

The limitations of existing methods have motivated us to develop a training & assistant system that can effectively improve the workforce outcomes. It is worker-centered, i.e., every element in the system is to assist the worker in achieving the best operational result. To realize worker-centered training, a necessary task is to perceive the worker's states, such as behavior and intention. There exist different kinds of sensors that have been used for this purpose [1, 2, 6, 9, 19]. To recognize worker activities, various methods have been applied [10, 13, 14, 16, 18, 20]. While being aware of a worker's states during the training, necessary instructional information can be introduced to guide the worker's training with AR techniques.

This project aims to develop a self-aware, active-guiding training & assistant system for worker-centered intelligent manufacturing by exploring advanced sensing technologies, AI methods, and AR techniques. Specifically, as shown in Figure 1, we have designed a multi-modal sensing system to sense the worker via different modalities, and have developed efficient and robust deep learning algorithms that analyze sensor data to recognize worker states. This awareness of worker state allows the system to understand the worker both physically and mentally, thus creating a basis for intelligent decision making. Finally, we have created multi-modal AR instructions that are generated according to the training decision made and provided to meet the worker's needs.

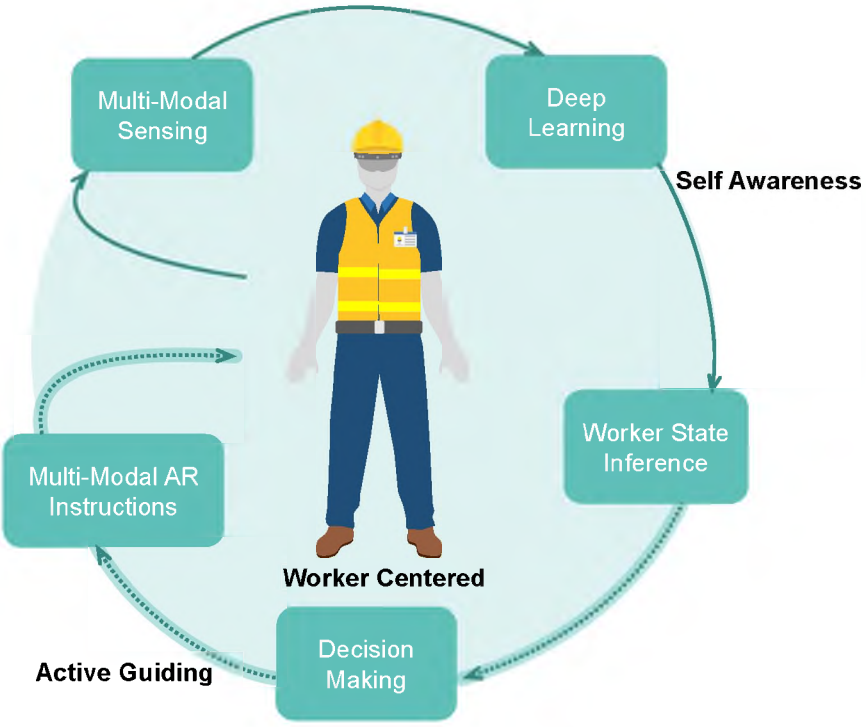


Figure 1. Overview of the proposed worker-centered training & assistant system.

2. WORKER STATE AWARENESS

2.1. MULTI-MODAL SENSING SYSTEM

To comprehensively perceive the worker, we developed a multi-modal sensing system illustrated in Figure 2. The system is composed of both ambient and wearable sensors with different modalities. Each sensor has a unique capability in collecting specific information about the worker. Various ambient sensors were used to capture the worker's activities in the workplace. Optic cameras were used to capture RGB images. Depth cameras such as a Microsoft Kinect or Lidar (light detection and ranging) sensors were applied to obtain data in the 3D space. Infrared cameras can detect the worker in a dark environment. Pressure sensing mats were developed to capture the standing states. Ambient sensing can collect a large amount of data without interfering the worker's activity. Nevertheless, the

complex setup and occlusion issue are main challenges in implementing ambient sensing. To compensate for these limitations, wearable sensing was applied. A smart Eyewear containing cameras was worn to perceive the surroundings from the first-person view of the worker. IMU (Inertial Measurement Unit) sensors were used to capture the movement of the worker body. sEMG (surface Electromyography) sensors were utilized to obtain the muscle activities. ECG (Electrocardiogram) sensors were used to monitor the worker's heart activities. EEG (electroencephalogram) sensors were used to collect electrical events of the human brain. All of the data were synchronized and sent to the local workstation via different transmission protocols.

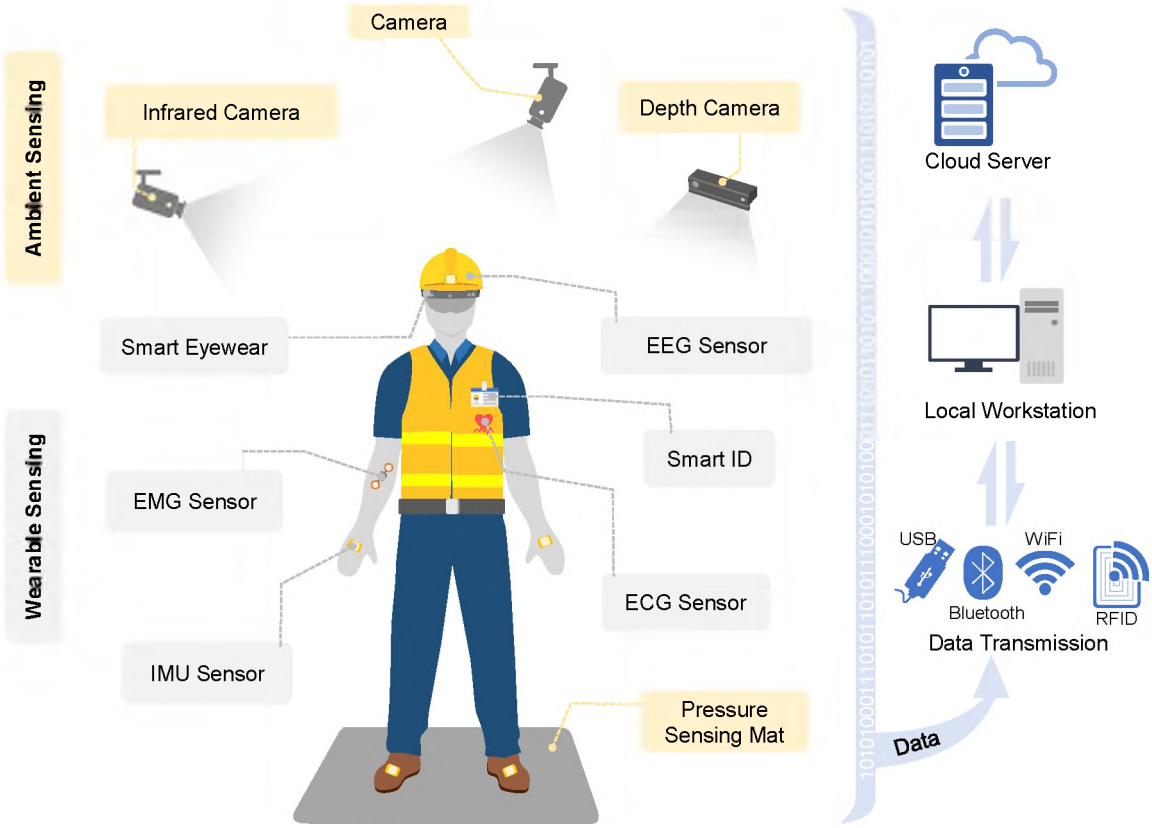


Figure 2. Schematic of the proposed multi-modal sensing system.

2.2. WORKER BEHAVIOR AND INTENTION UNDERSTANDING

With data obtained from the above multi-modal sensing system, we developed deep learning models, such as convolutional neural networks (CNN) and recurrent neural networks (RNN), to understand the worker behavior and intention from both spatial and temporal perspectives (e.g., walking towards a workstation, turning a screwdriver, etc.). The worker intention comprises mental activities related to specific tasks such as having confidence in, or feeling confused for, a specific operation. Specifically, we explored designing different models, including vision-based, IMU-based, sEMG-based, and EEG-based deep learning models, to recognize the worker activity and mental intention. A single sensing modality cannot guarantee robust perception under various circumstances. Therefore, we developed data fusion algorithms to take advantage of multi-modal sensing. Different sensing modalities were fused to augment individual speculations for making the final inference. The optimal fusion method was identified by comparing their overall performance.

2.3. INTERACTING PART/TOOL DETECTION

Most activities of a worker involve worker-object interactions. Detecting objects the worker is interacting with is important for understanding activities of the worker and for providing instructional information to help the worker locate desired objects. In this study, object detection algorithms, such as R-CNN [18], were implemented to recognize the interacting parts or tools in real time (e.g., Figure 3(a)). To establish the dataset for training the algorithms, we designed a data collecting system to take pictures of the objects automatically (see Figure 3(b)). Manually collecting data of some objects from all kinds of scales and viewpoints is difficult or inefficient. Thus, we developed a data synthesis approach to generate data directly from CAD models (see Figure 3(c)). The CAD model of an object was designed from CAD software or 3D scanning data and then imported to

virtual scenarios. The model was rendered with different poses, obtained by setting the camera from various distances and perspectives, to simulate the variations in the physical world. With the synthesizing method, a large amount of data were generated with labels annotated efficiently.

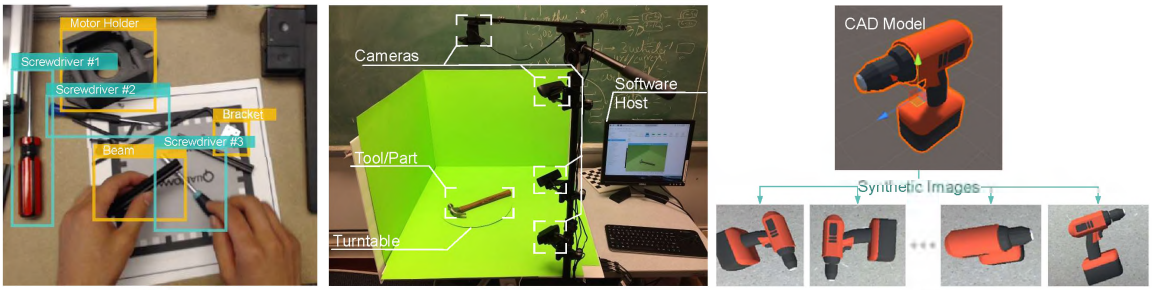


Figure 3. (a) Part/Tool detection results with highlighted bounding boxes; (b) Our developed data collecting system; (c) Image data synthesis rendered from a CAD model.

3. ACTIVE GUIDANCE FOR WORKER

3.1. MULTI-MODAL GUIDANCE WITH AUGMENTED REALITY

Augmented Reality (AR) technologies have been applied to manufacturing training, mainly for simulating costly or dangerous processes beforehand. In this study, we developed an instructional system with multi-modal AR to provide timely, onsite guidance for the worker. The working scenario was captured with a first-person camera to perceive the physical world. The camera pose was estimated in real time to allow the generated virtual information to be superimposed upon the real world with intuitive mapping, which effectively eliminates the discomfort that virtual information may bring to the worker. A monitor-based or glasses-based AR interface was applied to provide graphics that are rendered for the worker and can be overlaid on the physical world. Finger-wearable haptic rings were used to give the worker a realistic feedback of the sense of touch. An auditory

display was included to give the worker a timely sound feedback such as a vocal warning. An assistant laser pointer with two degrees of freedom was designed to help the worker search for the desired tools or parts. All modalities of guidance were integrated to achieve a comprehensive and complementary operational assistance. If the required parts/tools do not appear in the workspace or are not detected, the instructional information still can be provided via the visual or audio interfaces.

3.2. DEMAND ANALYSIS AND GUIDING STRATEGIES

Estimating the worker's potential demands for assistance (i.e., assistant information that can instruct workers to optimize their current operational workflows, e.g., how well the current operation is performed and what the next operation is) and then providing guidance accordingly is crucial to achieving the functionality of active guiding. After the worker's states are perceived, including 1) what the worker is doing, 2) what the worker's mental intention is, and 3) what the desired tools/parts are, all the information is integrated to determine the effective assistance that can meet the worker's demand, such as instructional information to conduct the current step or a reminder warning to fix a previous illegal operation. Furthermore, a guiding strategy was developed in order to provide instructions appropriately. A worker's performance was evaluated in comparison to experienced workers, and a "Demanding Score" is defined to represent the level of demanding for assistant information. Specifically, the time taken of each operational step can be obtained using the deep learning approaches mentioned above. If a particular action takes more time than average, the Demanding Score is increased. Then, if the Demanding Score is higher than a threshold, the needed assistant information will be actively added with the above developed multi-modal guidance system. For example, graphics information will be displayed via the monitor or AR glasses, and the laser pointer will point to the desired tool/part for the next step if the worker is in a confused state. In addition, the training progress of each worker is logged, and it can be retrieved by worker identification

techniques, such as RFID tag or facial recognition. Also, the timing, i.e., how to provide the guidance at the right time, is critical in the training process. It should be timely enough but not disturb the ongoing operation.

4. CASE STUDY

The proposed self-aware and active-guiding training & assistant system has been progressively validated. In this section, two case studies in manufacturing assembly are presented.

4.1. MULTI-MODAL RECOGNITION OF WORKER ACTIVITY

In this case study, we developed a multi-modal approach for worker activity recognition in manufacturing assembly tasks using Inertial Measurement Unit (IMU) signals obtained from a Myo armband and videos from a visual camera (see Figure 4(a)). A worker activity dataset of six assembly activities has been established, as shown in Figure 4(b). These activities are: grabbing tool/part, hammering nail, using power-screwdriver, resting arm, twisting screwdriver, and using wrench. For IMU signals, we designed two modalities in both frequency and spatial domains. For the camera data, two more modalities were included at the video frame and video clip levels. Accordingly, four deep learning networks were built to cope with data from the different modalities. Then, all the individual networks were fused to output the final inference. Various fusion methods were evaluated including the maximum fusion, average fusion and weighted fusion. The developed approach has been evaluated and shown to achieve promising recognition accuracy in experiments, i.e., 97% and 100% in the leave-one-out and half-half experiments, respectively [3, 14, 15].

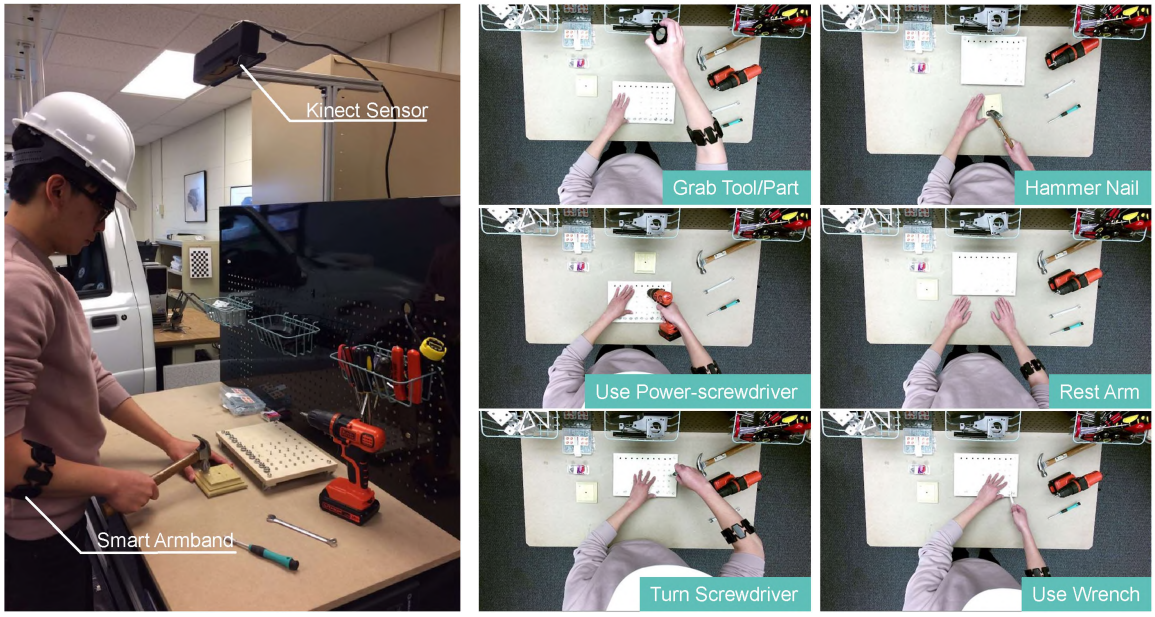


Figure 4. (a) Experimental setup and (b) examples of the six worker activities.

4.2. COMPARISON OF AR AND MANUAL GUIDANCE IN A MECHANICAL ASSEMBLY TRAINING TASK

In this case study, we applied multi-modal AR guidance in a training task, i.e., assembling the spindle subassembly of a desktop carving machine (see Figure 5(a, b)). To assess its effectiveness compared with traditional manual guidance, we recruited 20 subjects without any prior experience on the assembly task. They were divided into two groups and asked to conduct the task with manual and AR instructions, respectively. Then their performances were compared in terms of the completion time and number of errors (see Figure 5(c, d) and Table 1). The AR method has shown superiority over the manual one. This has demonstrated the feasibility and potential of applying the AR method to the industry for the frontline workers [7].

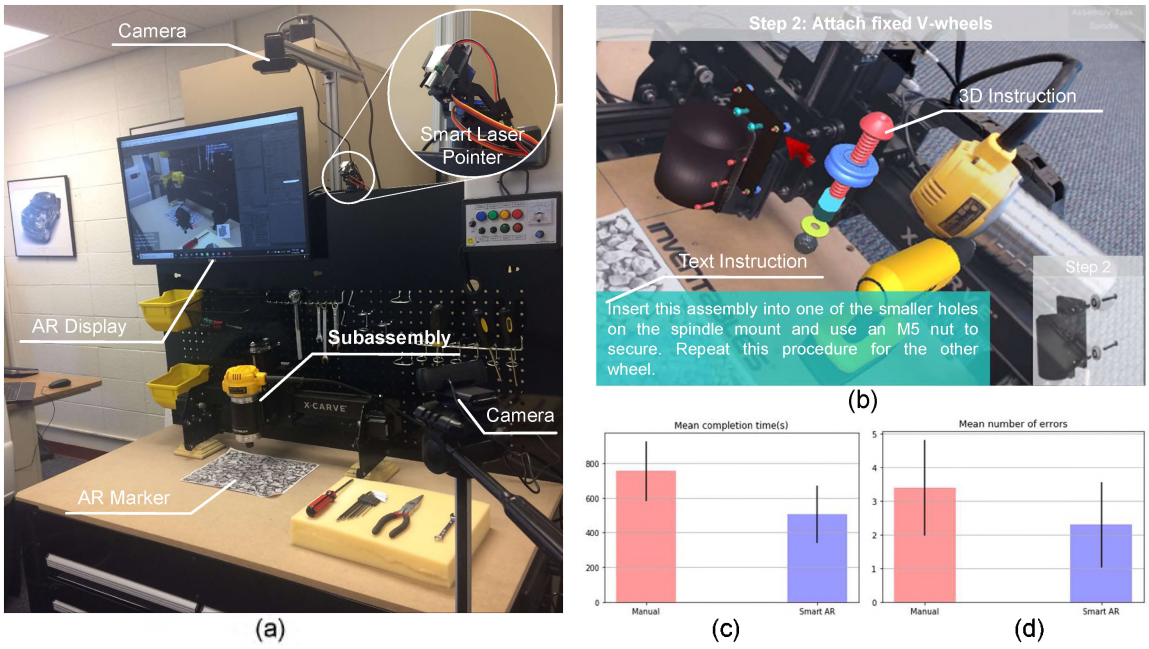


Figure 5. (a) Experimental setup; (b) AR display content; Performance comparison between manual and AR guidance: (c) completion time and (d) number of errors.

Table 1. Error reduction using the multi-modal AR instruction.

Error Type	Reduction (%)
Tool/Part Selection	72.7
Assembly Sequential Order	100
Installation	4.8

5. CONCLUSIONS

In this ongoing research, we have proposed a novel worker-centered training & assistant system for intelligent manufacturing. This system has the self-awareness of the worker's state and can provide active guidance to the worker as needed. Compared to traditional approaches, our proposed system starts with the worker's experience, considers more of the worker's learning effect, and has more interactions with the worker. The worker's state is perceived with multi-modal sensing and deep learning methods, and is used to analyze and determine the potential guiding demands. Then active instructions with

augmented reality are provided to suit the worker's needs. The case studies have shown the feasibility and promise of applying the proposed system for training and assisting frontline workers. Also, our proposed self-aware and active-guiding training & assistant system has constructed a framework for further studies in worker-centered intelligent manufacturing.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Akhavian, R. and Behzadan, A. H., 'Smartphone-based construction workers' activity recognition and classification,' Automation in Construction, 2016, **71**, pp. 198–209.

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

[3] Al-Amin, M., Tao, W., Doell, D., Lingard, R., Yin, Z., Leu, M. C., and Qin, R., 'Action recognition in manufacturing assembly using multimodal sensor fusion,' in '25th International Conference on Production Research Manufacturing Innovation: Cyber Physical Manufacturing, August 9–14, 2019 Chicago, Illinois (USA),' 2019 .

[4] Doshi, A., Smith, R. T., Thomas, B. H., and Bouras, C., 'Use of projector based augmented reality to improve manual spot-welding precision and accuracy for automotive manufacturing,' The International Journal of Advanced Manufacturing Technology, 2017, **89**(5-8), pp. 1279–1293.

[5] Gilbreth, F. B., *Motion study: A method for increasing the efficiency of the workman*, D. Van Nostrand Company, 1911.

[6] Khosrowpour, A., Fedorov, I., Holynski, A., Niebles, J. C., and Golparvar-Fard, M., 'Automated worker activity analysis in indoor environments for direct-work rate improvement from long sequences of rgb-d images,' in 'Construction Research Congress,' Atlanta, GA, 2014 pp. 729–738.

[7] Lai, Z.-H., Tao, W., Leu, M. C., and Yin, Z., 'Smart augmented reality instructional system for mechanical assembly towards worker-centered intelligent manufacturing,' Journal of Manufacturing Systems, 2020, **55**, pp. 69–81.

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

[9] Li, P., Meziane, R., Otis, M. J.-D., Ezzaidi, H., and Cardou, P., 'A smart safety helmet using imu and eeg sensors for worker fatigue detection,' in '2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings,' IEEE, 2014 pp. 55–60.

[10] Liu, Y., Nie, L., Liu, L., and Rosenblum, D. S., 'From action to activity: sensor-based activity recognition,' Neurocomputing, 2016, **181**, pp. 108–115.

[11] Makris, S., Karagiannis, P., Koukas, S., and Matthaiakis, A.-S., 'Augmented reality system for operator support in human–robot collaborative assembly,' CIRP Annals, 2016, **65**(1), pp. 61–64.

[12] Tao, W., Lai, Z.-H., and Leu, M. C., 'Manufacturing assembly simulations in virtual and augmented reality,' 2019.

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

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

[15] Tao, W., Lai, Z.-H., Leu, M. C., and Yin, Z., 'Multi-modal recognition of worker activity for human-centered intelligent manufacturing,' 2019.

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

[17] Uva, A. E., Gattullo, M., Manghisi, V. M., Spagnulo, D., Cascella, G. L., and Fiorentino, M., 'Evaluating the effectiveness of spatial augmented reality in smart manufacturing: a solution for manual working stations,' The International Journal of Advanced Manufacturing Technology, 2018, **94**(1-4), pp. 509–521.

[18] Wang, J., Chen, Y., Hao, S., Peng, X., and Hu, L., 'Deep learning for sensor-based activity recognition: A survey,' Pattern Recognition Letters, 2019, **119**, pp. 3–11.

[19] Ward, J. A., Lukowicz, P., Troster, G., and Starner, T. E., 'Activity recognition of assembly tasks using body-worn microphones and accelerometers,' IEEE transactions on pattern analysis and machine intelligence, 2006, **28**(10), pp. 1553–1567.

[20] Yang, J., Nguyen, M. N., San, P. P., Li, X. L., and Krishnaswamy, S., 'Deep convolutional neural networks on multichannel time series for human activity recognition,' in 'Twenty-Fourth International Joint Conference on Artificial Intelligence,' 2015 .

[21] Yew, A., Ong, S., and Nee, A., 'Towards a griddable distributed manufacturing system with augmented reality interfaces,' Robotics and Computer-Integrated Manufacturing, 2016, **39**, pp. 43–55.

[22] Zhong, R. Y., Xu, X., Klotz, E., and Newman, S. T., 'Intelligent manufacturing in the context of industry 4.0: a review,' Engineering, 2017, **3**(5), pp. 616–630.

II. AMERICAN SIGN LANGUAGE ALPHABET RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS WITH MULTIVIEW AUGMENTATION AND INFERENCE FUSION

Wenjin Tao[a], Ming C. Leu[a], Zhaozheng Yin[b]

[a]Missouri University of Science and Technology, Rolla, MO 65409, USA

[b]Stony Brook University, Stony Brook, NY 11794, USA

ABSTRACT

American Sign Language (ASL) alphabet recognition by computer vision is a challenging task due to the complexity in ASL signs, high interclass similarities, large intraclass variations, and constant occlusions. This paper describes a method for ASL alphabet recognition using Convolutional Neural Networks (CNN) with multiview augmentation and inference fusion, from depth images captured by Microsoft Kinect. Our approach augments the original data by generating more perspective views, which makes the training more effective and reduces the potential overfitting. During the inference step, our approach comprehends information from multiple views for the final prediction to address the confusing cases caused by orientational variations and partial occlusions. On two public benchmark datasets, our method outperforms the state-of-the-arts.

Keywords: Intelligent manufacturing; Deep learning; Augmented reality; Cyber-physical system; Smart manufacturing

1. INTRODUCTION

American Sign Language (ASL) is an important communication way to convey information among the deaf community in North America. Although it is primarily used by people who have hearing or speech difficulties, similar signs also can be used in Natural User

Interface (NUI) systems to realize human-computer/robot interaction by hand gestures. Its automatic recognition using various sensing devices has been studied extensively for decades with significant progress having been made. There are mainly two categories of sensing devices used in those studies: (1) wearable devices, such as a cyber glove embedded with a flex sensor or an Inertial Measurement Unit (IMU) sensor, and a set of trackable markers of a motion capturing system; and (2) non-wearable devices, or markerless vision-based devices, such as a RGB camera or a depth camera. Wearable devices directly sense the hand status like adjacent joints' angles, spatial positions and movements, which can provide fairly precise information of the hand [15, 16]. However, they are still too heavy and uncomfortable for daily use. Markerless vision-based recognition has been increasingly popular recently because it does not need sensors attached to a human and the low-cost vision/depth cameras such as Microsoft Kinect are commercially available. However, it is still challenging to recognize ASL signs because of the complexities of these signs, high interclass similarities, large intraclass variations, and constant finger occlusions.

1.1. RELATED WORK

In this paper, we focus on recognizing the alphabet of American Sign Language (ASL). In general, the ASL alphabet recognition task is formulated as two subtasks: feature extraction and subsequent multiclass classification. Researchers have been using different methods to extract discriminative features and create powerful classifiers.

Pugeault and Bowden [17] applied Gabor filters to extract features from both color and depth images at 4 different scales. Then a multiclass random forest classifier was used to recognize the 24 static ASL alphabet signs. They had 49% recognition rate in the leave-one-out experiment. Half of the signs could not be recognized, showing that Gabor filters cannot capture enough discriminative information for differentiating different signs. In addition, they developed a realtime recognition system which provides an interface for

the user to select the desired sign among ambiguous ones. It is worth mentioning that they publicized their dataset well and this dataset has been the most common benchmark in this research area, as surveyed in the following.

Wang et al. [23] also used color and depth images for recognition. They proposed a Superpixel Earth Mover's Distance (SP-EMD) metric to measure the distance between two signs based on the shape, texture and depth information. Then a template matching technique was utilized for the sign classification. They reported 75.8% recognition rate on the benchmark dataset. Some researchers only focused on either color or depth image. Maqueda et al. [13] deployed a Volumetric Spatiograms of Local Binary Patterns (VS-LBP) descriptor on color videos or images, without using depth images, for extracting spatio-temporal features. By using a Support Vector Machine (SVM) classifier, they had 83.7% leave-one-out accuracy on the benchmark dataset. Nai et al. [14] extracted features from only depth images on randomly positioned line segments and used a random forest for classification, with 81.1% accuracy reported in their paper.

Some studies attempted to exploit the 3D information embedded in the depth images (3D approach). Kuznetsova et al. [10] implemented an Ensemble of Shape Function (ESF) descriptor [24] on the 3D point cloud for feature extraction and a multi-layered random forest for classification. Zhang et al. [26] proposed a Histogram of 3D Facets (H3DF) descriptor to encode the 3D shape information of different hand gestures. Then they used a SVM with a linear kernel for the classification step and got 73.3% in the leave-one-out accuracy. Later, Zhang and Tian [25] combined their H3DF with a dense sampling method and achieved an improved accuracy of 83.8%. Rioux-Maldague and Giguére [19] created a mask from the depth image and applied it on the intensity image to filter out the hand region to form the intensity features. Six binary images were generated using cross-sections of depth images to form depth features, which were then fed into a Deep Belief Network (DBN) and achieved 77% recall and 79% precision on the benchmark dataset. These 3D approaches are promising to achieve better performance than image representations due

to the extra dimension. However, the 3D point cloud obtained from the depth image is sparse at the regions with large gradients and absent at the occluded areas, which affects the overall performance. To fully exploit the 3D benefits from the depth image, some 3D reconstruction methods can be used to recover more valuable information.

Due to the articulated structure of hands, some studies implemented a hand part segmentation step before the gesture recognition (bottom-up approach). Keskin et al. [8] extracted depth comparison features from depth images following the method proposed by Shotton et al. [21] and fed them into a per-pixel random forest classifier. The final predicted label for the whole image is determined by majority voting. They reported their leave-one-out recognition rate as 84.3% on the benchmark dataset. Furthermore, they introduced multi-layered random forests in classifying hand parts to estimate its pose. This classifier is trained using synthetic depth images which have the parts' groundtruth of a hand. To generate more realistic training data for per-pixel hand part classification, a colored latex glove was employed in the research of Dong et al. [5]. They added kinematic constraints on the estimated joint locations to improve the localization accuracy, based on which 13 key angles of the hand skeleton were extracted and fed into a random forest classifier, resulting in 70% recognition rate on the benchmark dataset. One of the major drawbacks for these bottom-up approaches is that the sign recognition performance is highly dependent upon the result of the hand part segmentation, and it is challenging to improve the performance of the hand part segmentation because of the high complexities and constant occlusions.

Recently, deep learning methods such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated their extraordinary performance in various classification and recognition tasks. For example, in the traffic sign classification task and the ImageNet challenge, the CNN systems achieved even better performances than those of humans [7, 20]. Unlike the handcrafted feature extractor, which is designed to capture only specific patterns, the deep learning based feature extractor is automatically trained to capture the most discriminative features using the real data. Ameen and Vadera [2]

introduced a CNN model with both color and depth inputs in the ASL alphabet recognition task. This model has two convolutional layers for each input to extract features from them. Those two sets of features are concatenated into one before being fed into fully connected layers. They reported the accuracy of 80.34% on the benchmark dataset. RNNs have also been utilized for hand gesture recognition tasks and achieved promising results [3].

1.2. PROPOSED METHOD

Depth images contain distance information from the camera plane to the objects in the camera view, where each pixel represents a measured distance. Therefore, it is easier to segment the target object in a depth image than a color image. Thus, this research focuses on recognizing finger spelling signs from depth images as follows:

1. Considering the challenges of the ASL alphabet recognition task, we choose CNN as the basic model to build the classifier because of its powerful learning ability that has been shown.

2. To fully exploit the 3D information provided by depth images, we develop a novel multiview augmentation strategy. It generates more views from different perspectives, in order to augment the perspective variations that cannot be achieved using traditional image augmentation methods.

3. To solve the interclass similarity issues caused by perspective variations and partial occlusions, we first make predictions for all individual views and then fuse information from them for the final prediction.

The remainder of this paper is organized as follows. Our proposed methods of multiview augmentation, CNN model, and inference fusion are detailed in Sections 2, 3 and 4, respectively. The experimental setups and experimental results using the public datasets are described in Sections 5 and 6. Finally, Section 7 gives the conclusions of this research.

2. MULTIVIEW AUGMENTATION

To train a valid CNN classifier with good performance, a large amount of labeled data needs to be fed into it. However, it is always time-consuming and costly to collect enough data with annotated labels. Data augmentation is a common method to solve such an issue, which synthesizes additional data derived from original ones.

Traditionally, data augmentation refers to implementing a series of image transformation techniques on the original images, which consist of rotating, scaling, shifting, flipping, shearing, etc. The image transformation is able to introduce more variations and still keep the recognizable features. However, the basic image transformation cannot introduce realistic variations of different perspectives (e.g., out-of-plane transformations in the real world), which are common for hand gestures because they are highly perspective-dependent.

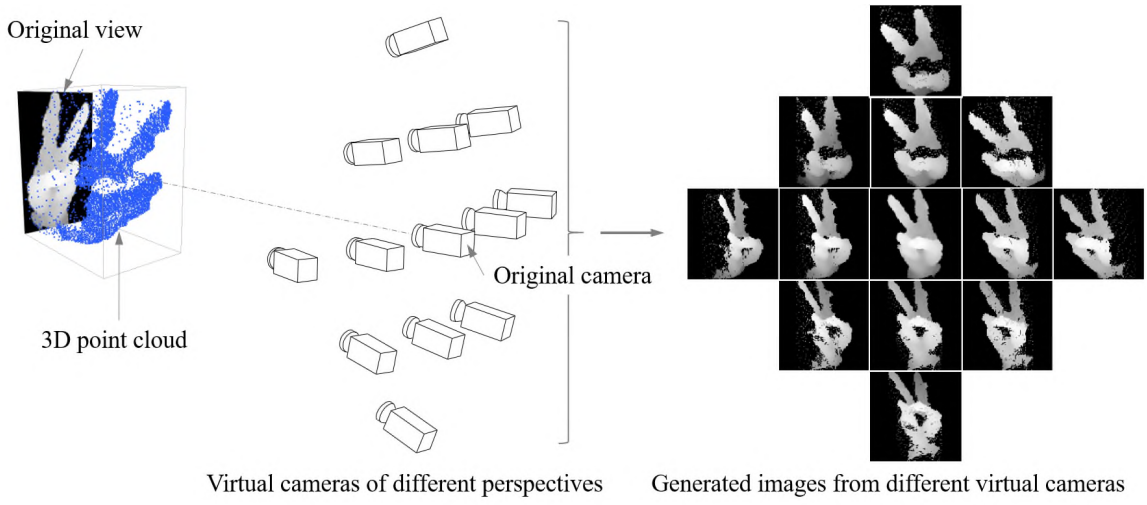


Figure 1. Multiview augmentation strategy.

To synthesize those perspective variations, we propose a multiview augmentation strategy illustrated in Figure 1. A hand gesture is represented as a depth image in its original view, from which a 3D point cloud is obtained. Then additional virtual cameras are set up

and oriented to the point cloud with different perspectives. Finally, a set of additional views are generated from those distributed virtual cameras. The central image on the right hand side in Figure 1 is the original depth image, based on which the other views are generated.

The generation process of a new view is shown in Figure 2. Given a hand depth image I with M pixels (Figure 2(a)), to extract the point cloud $\mathcal{P} = \{p_1, ..., p_m, ..., p_M\}$ from the depth image, each pixel $I(i, j)$ is projected into the 3D space as a point $p_m = (p_m^{(x)}, p_m^{(y)}, p_m^{(z)})$. This projection first translates the origin to the image center and then uses the depth values as the Z values, which is formulated as follows (Figure 2(b)):

$$\begin{aligned} p_m^{(x)} &= j - w/2 \\ p_m^{(y)} &= -i + h/2 \\ p_m^{(z)} &= I(i, j) \end{aligned} \tag{1}$$

where i and j represent the indices of row and column of I, respectively, w and h are the width and height of the depth image, respectively.

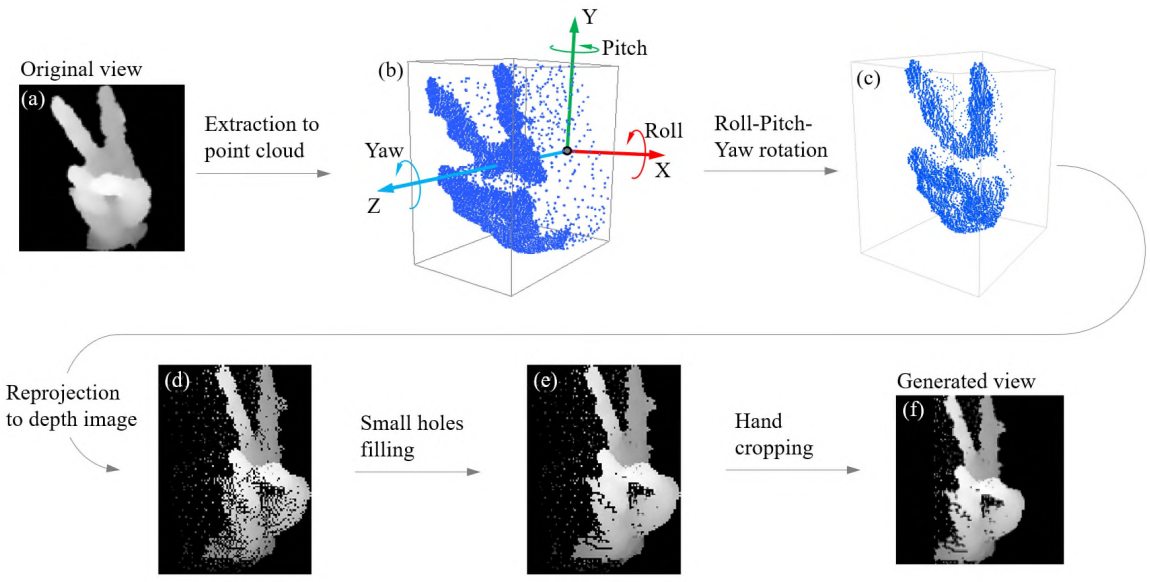


Figure 2. Generation of a new view.

To capture the point cloud from a new perspective, a yaw-pitch-roll rotation transformation on the point cloud around its volume center is implemented. For point p_m in $\mathcal{P}$, its new location after rotation can be calculated by

$$p'_m = R(\alpha, \beta, \gamma) p_m \tag{2}$$

where $R(\alpha, \beta, \gamma)$ is the rotation matrix, α, β and γ represent yaw, pitch and roll angles around z, y and x axes, respectively. It can be further expressed as a multiplication of three orthogonal rotation matrices [11]:

$$\begin{aligned} R(\alpha, \beta, \gamma) &= R_z(\alpha) R_y(\beta) R_x(\gamma) \\ &= \begin{bmatrix} \cos\alpha \cos\beta & r_{12} & r_{13} \\ \sin\alpha \cos\beta & r_{22} & r_{23} \\ -\sin\beta & \cos\beta \sin\gamma & \cos\beta \cos\gamma \end{bmatrix} \\ where & \end{aligned} \tag{3}$$

$$\begin{aligned} r_{12} &= \cos\alpha \sin\beta \sin\gamma - \sin\alpha \cos\gamma \\ r_{13} &= \cos\alpha \sin\beta \cos\gamma + \sin\alpha \sin\gamma \\ r_{22} &= \sin\alpha \sin\beta \sin\gamma + \cos\alpha \cos\gamma \\ r_{23} &= \sin\alpha \sin\beta \cos\gamma - \cos\alpha \sin\gamma \end{aligned}$$

By implementing the yaw-pitch-roll rotation on each point, a new point cloud $\mathcal{P}'$ is generated (Figure 2(c)). Then $\mathcal{P}'$ is reprojected onto a plane to form a new depth image (Figure 2(d)), which is the reverse process of point cloud extraction in Equation 1. After reprojection, the new depth image might have holes because the occluded regions in the original image get exposed in the new one after the yaw-pitch-roll rotation transformation.

Those small holes can be filled by interpolation using their neighboring pixels' values (Figure 2(e)). Then the hand region in the new image is cropped and re-centered by removing its surrounding isolated noises (Figure 2(f)).

3. CNN MODEL

The overall architecture of our CNN model is shown in Figure 5. It is composed of a layered feature extraction module and a classification module.

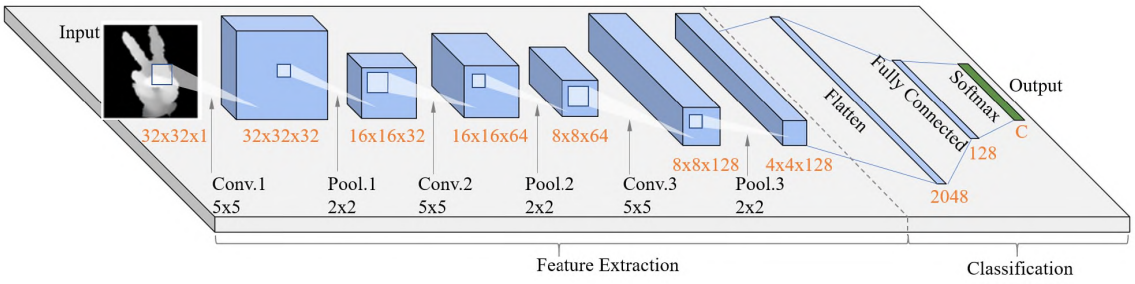


Figure 3. The overall architecture of our CNN model.

In the feature extraction module, suppose there are N depth images $X_n, n \in [1, N]$ after data augmentation, they are scaled to the size 32×32 ($width \times height$) and normalized to the interval $[0, 1]$, and then fed into three 5×5 convolutional layers for feature extraction. Rectified Linear Unit (ReLU) activation function [6] is applied to each convolutional operation. Then each convolutional layer is followed by a 2×2 max pooling layer, which downsamples the previous feature map by a half.

The classification module accepts the $4 \times 4 \times 128$ feature map from the feature extraction module and flattens it as a 2048 feature vector. Then two fully connected layers are used to densify the feature vector to the dimensions of 128 and C sequentially, where C is the number of ASL alphabet sign classes. Then this C-dimensional score vector $S([S_1, ..., S_c, ..., S_C])$ is transformed to output the predicted probabilities with a softmax

function as follows:

$$P(y_n = c|X_n) = \frac{\exp(S_c)}{\sum_{c=1}^{C} \exp(S_c)} \tag{4}$$

where $P(y_n = c|X_n)$ is the predicted probability of being class c for sample X_n.

Dropout has been proved to be a powerful regularization technique used to avoid the overfitting, which randomly drops units from the neural network during training [22]. Therefore, it is implemented after each pooling layer in our CNN model.

The process of training a CNN model involves optimization of the network's parameters w to minimize the cost function for the training dataset X. We select the commonly used regularized cross entropy [6] as the cost function, which is

$$\mathcal{L}(w) = \sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log[P(y_n = c|X_n)] + \lambda l_2(w) \tag{5}$$

where y_{nc} is 0 if the ground truth label of X_n is the cth label, and is 1 otherwise. The l_2 regularization term is appended to the loss function for penalizing large weights, and λ is its coefficient.

4. MULTIVIEW INFERENCE FUSION

Due to the high interclass similarities, some signs are almost the same from certain perspectives. The inference relying on only one view may not be convincing enough. Therefore, we propose a multiview inference fusion strategy in order to augment the speculation of each individual view, as illustrated in Figure 4.

In the inference step, suppose the CNN model described in Section 3 has been trained with the augmented dataset, and we have N_{test} depth images for inference. First of all, each query sample is preprocessed to the right input $X_n^0, n \in [1, N_{test}]$ which has the size of 32×32 and the value range of $[0, 1]$. Then, similar to the multiview augmentation process, a set of new views $\{X_n^1, X_n^2, \cdots, X_n^{N_{APS}}\}$ are generated from the original ones X_n^0, where

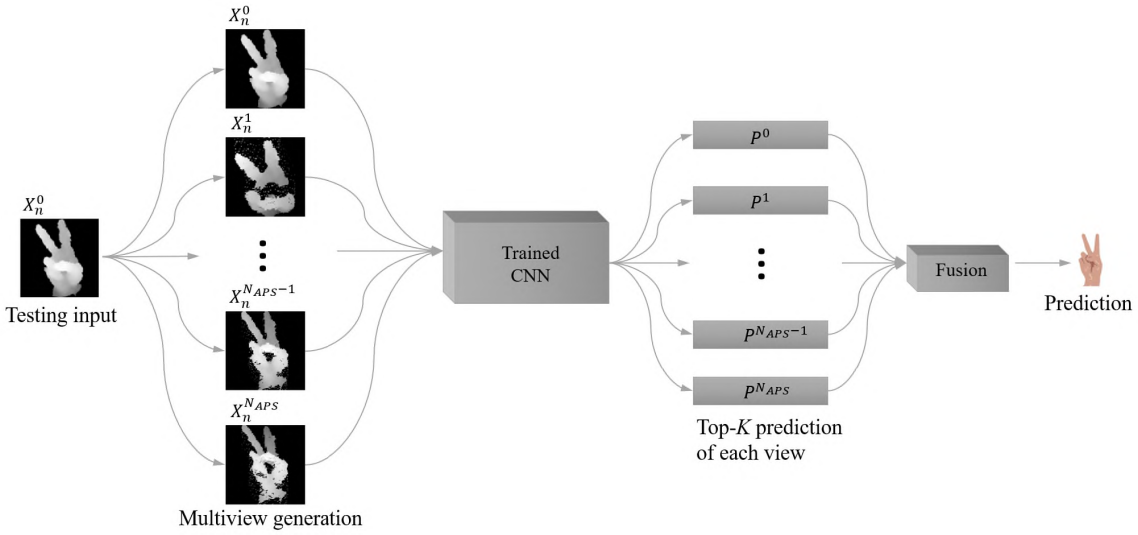


Figure 4. Multiview inference fusion strategy.

N_{APS} is the number of augmentations per sample. After that, there are $(N_{APS} + 1)$ views $\{X_n^0, X_n^1, \cdots, X_n^v, \cdots, X_n^{N_{APS}}\}$ for the original query sample. Each view $X_n^v, v \in [0, N_{APS}]$ is inferred individually by the trained CNN to get the probablity distribution P^v of the top-K predicted classes ($K \in [1, C]$, e.g., $K = 5$). Then they are fed into an inference fusion step for the final prediction.

In the inference fusion step, predictions from all individual views are fused together. We introduce the informativity value I^v to evaluate the prediction confidence at each view v. I^v is calculated with Equation 24, which is modified from the Shannon entropy of a discrete probability distribution to vary in the interval of $[0, 1]$.

$$I^v = \frac{\sum_{k=1}^{K} p_k^v \log p_k^v}{\log K} + 1 \tag{6}$$

where v is the index of views and k is the index of top-K candidates. p_k^v represents the probability of the kth class candidate at the vth view. I^v will be close to 0 if all the top-K candidates have similar probabilities (i.e., $p_k^v \approx 1/K$), and 1 if the probability of top-1 class candidate is about reaching 1 (i.e., $p_1^v \approx 1$).

Then every predicted probability p_k^v at the vth view is weighted by I^v of this view. The final predicted label is chosen as the one that maximizes the $I^v p_k^v$ value:

$$\hat{y}_{fusion} = \max_v \hat{y}_{fusion}^v \tag{7}$$

where

$$\hat{y}_{fusion}^v = \arg\max_k I^v p_k^v \tag{8}$$

5. EXPERIMENTS

5.1. DATASETS

To compare our method with others, we evaluate it on the public ASL alphabet dataset [17]. Some examples of the depth images in this dataset are shown in Figure 5. It has 24 finger spelling signs ('*J*' and '*Z*' are excluded because they involve finger movement) captured by a Kinect with color and depth images recorded. Those signs were performed by 5 different subjects and each of the 24 signs consists of about 500 to 600 samples. As shown in Figure 5, the hand regions were approximately cropped.

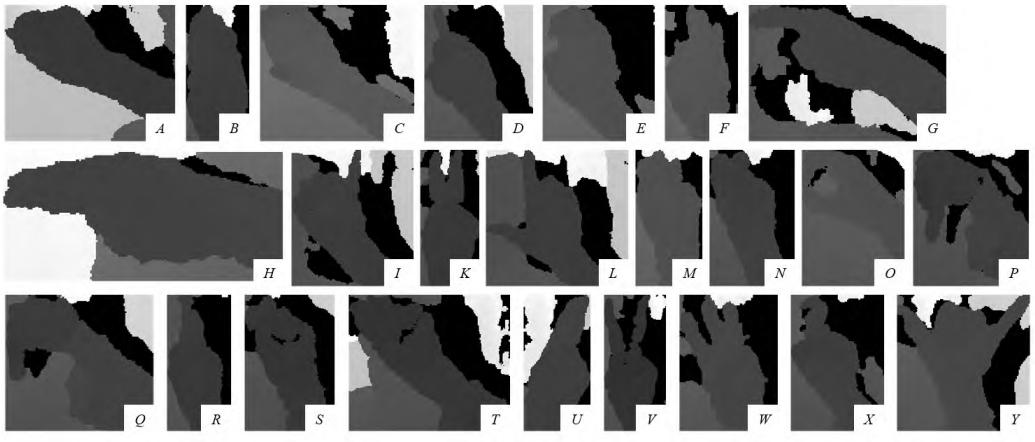


Figure 5. Depth image examples of the 24 signs in the ASL alphabet dataset.

To validate the generalization of our method, the NTU digit dataset [18] is also chosen for experiments. This dataset has 10 signs representing digits from 0 to 9 captured by a Kincet containing color and depth images as well. They are performed by 10 different subjects and each sign has 10 samples. Examples of the depth image of the 10 signs in this dataset are shown in Figure 6. Each image contains background and the hand region is not cropped or annotated.

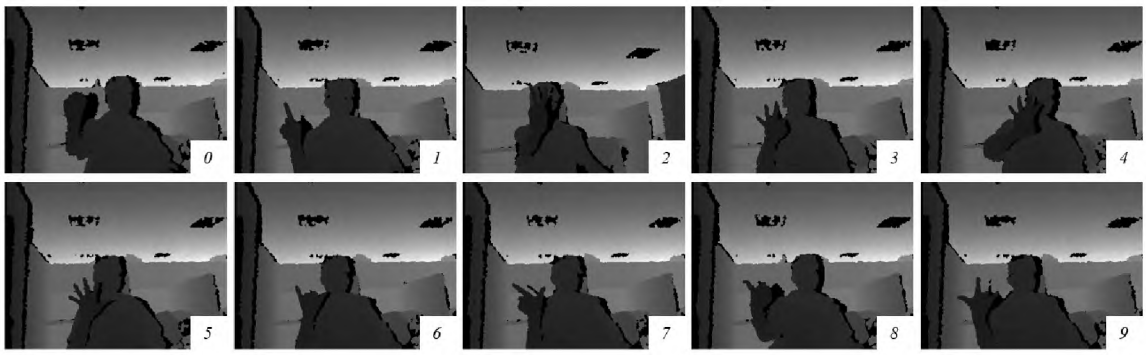


Figure 6. Depth image examples of the 10 signs in the NTU digit dataset.

5.2. PREPROCESSING

The image size of an input sample is a design parameter when building a CNN model and is fixed after the model is created. Thus, it only accepts input samples with the predefined sizes, e.g., our CNN model described in Section 3 needs each input sample to have the uniform size of 32×32 ($width \times height$).

Samples from the first dataset introduced in Section 5.1 have various sizes (as shown in Figure 5). Although each sample in the second dataset shares the same size (as shown in Figure 6), the size is 640×480 and the hand region is only a small part of the entire image. Therefore, a preprocessing procedure is needed to prepare the data for the CNN model. Taking an image from the NTU digit dataset [18] as an example, this process is illustrated

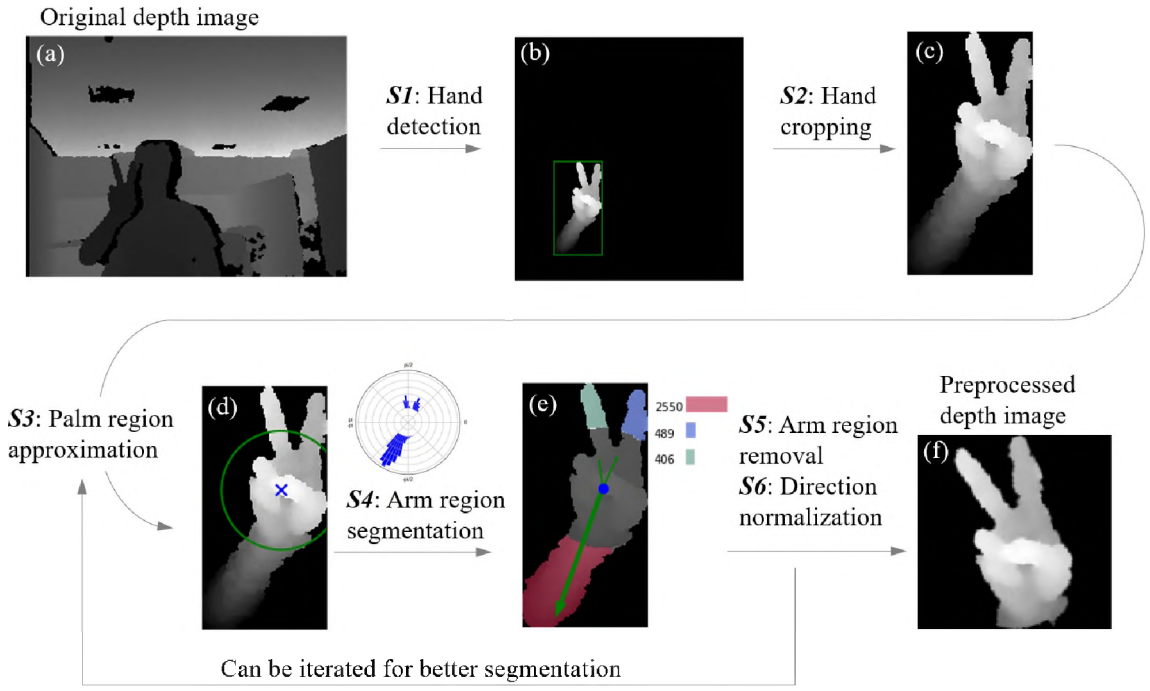


Figure 7. Hand region segmentation.

in Figure 7, where *S1-6* denote the processing steps. Suppose there is a raw depth image D (Figure 7(a)) captured by a depth camera and it is assumed that the hand is the closest object to this camera. First, a band-pass filter is applied to filter out the pixels in the range of $[d_{\min}, d_{\min}+\delta]$, where $d_{\min}$ is the minimum distance value and δ is the threshold distance that should approximately represent the hand occupation along the direction out of the image. After that, the depth image is reversed using the equation $D' = d_{\min}+\delta-D \quad (if\ D \neq 0)$, and then on the new depth image D', hand regions that are nearer to the camera will be brighter, while further regions will be darker. This conversion will let our CNN model focus on the nearer regions which contain more information in distinguishing different signs. There is only one hand and it is the frontmost object in a depth image for both of the datasets. Then the bounding box of the hand is detected (Figure 7(b)) and cropped by using histograms projected onto x and y axes (Figure 7(c)).

The palm center is approximated by calculating the mass center of the depth image and the palm is segmented as a circular region (Figure 7(d)). Then we calculate the polar histogram of the pixels that are outside the palm region, followed by a clustering step. The number of pixels is counted for each cluster. The cluster with the most pixels is segmented as the arm and its direction is taken as the mean of the directions of all its pixels (Figure 7(e)). Finally the arm region is removed, the image is rotated to make the arm direction point down and translated to make the mass center as the image center. Finally, the hand image is reshaped to the size of 32×32 and normalized to the interval of $[0, 1]$ (Figure 7(f)). Note that the processing steps *S3-6* in Figure 7 can be iterated to get a better segmentation result because in some cases the resulted hand still has a large arm area. For example, as shown in Figure 8, the first *S3-6* processing does not remove all the arm region (Figure 8(d)). By implementing the second *S3-6* processing, most of the arm pixels are removed (Figure 8(g)).

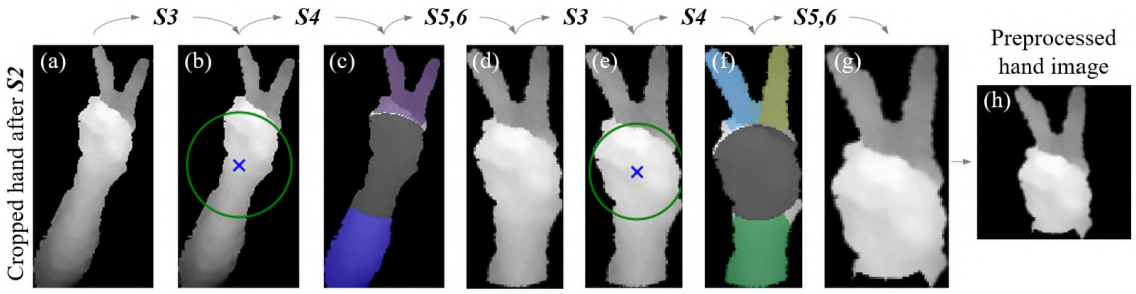


Figure 8. An example of iteration of the processing steps *S3-6* for better segmentation result.

The above preprocessing methods are implemented on the two datasets using tools from the OpenCV library [4], and the resulted samples are shown in Figures 9 and 10, respectively. For the ASL benchmark dataset, the direction normalization step (*S6* in Figure 7) is discarded because some signs (e.g., '*G*' and '*H*') are related to orientations.

The implementation of preprocessing and hand segmentation removes the background and prepares the images to have a centered hand on each with a uniform size 32×32 for the subsequent CNN training process.

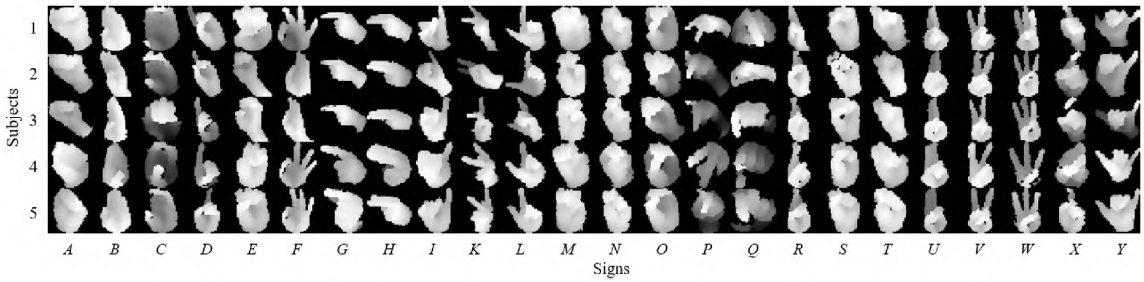


Figure 9. Examples of the 24 signs of each of the five subjects in the preprocessed ASL alphabet dataset.

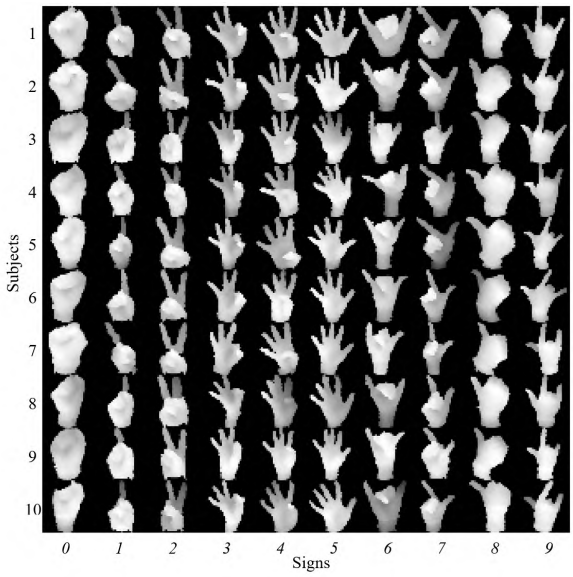


Figure 10. Examples of the 10 signs of each of the ten subjects in the preprocessed NTU digit dataset.

5.3. EVALUATION METRIC

We conduct comparisons with state-of-the-art recognition results on the above two datasets using the same evaluation policies as in [17], which are half-half and leave-one-out policies. For the half-half policy, one half of the dataset is randomly chosen and fed into the CNN model for training, and the other half is reserved for evaluation. For the leave-one-out policy, the samples from $N_{subjects} - 1$ out of $N_{subjects}$ subjects are used for CNN training, and the samples from the left one subject are used for evaluation. We employed a few commonly used metrics to evaluate this multiclass classification performance, which are

1. Accuracy

$$Accuracy = \frac{\sum_{n}^{N_{test}} 1(\hat{y}_n = y_n)}{N_{test}} \tag{9}$$

2. Precision and Recall

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \end{aligned} \tag{10}$$

3. F score

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{11}$$

where 1() in Equation 9 is an indicator function. In Equation 10, True Positive (TP) describes a sample X_n from a certain class y_n that is correctly classified as y_n; False Positive (FP) is defined as a sample X_n from a 'not y_n' class is incorrectly classified as y_n; False Negative (FN) means a sample X_n of the class y_n is misclassified as other 'not y_n' classes. In Equation 11, F score evaluates the overall performance of the of Precision and Recall, which is their harmonic mean in the interval [0,1].

5.4. SOME CNN TRAINING DETAILS

TensorFlow [1] is used in creating the CNN model described in Section 3. For the hyperparameters, we set the batch size, learning rate, dropout rate, regularizer as 256, 0.001, 0.1, and 1e-5, respectively. The Adam optimizer [9] is used in training, and the training is stopped after 100 epochs, which takes approximately 2 hours for a leave-one-out experiment on a workstation with one 12 core Intel Xeon processor, 64GB of RAM and one Nvidia Geforce 1080 Ti graphic card.

6. RESULTS AND DISCUSSION

6.1. EVALUATION OF THE CNN ARCHITECTURE

Due to the high architectural complexity and parametric variation of a CNN model, it is not feasible to evaluate all possible architectures and associated parameters (e.g., number of convolutional layers, kernel size, activation function, pooling method, etc.). In this study, a few representative CNN designs with increasing numbers of layers and different parameters are compared to find the optimal design. As shown in Table 1, eight CNN architectures (listed in columns) are selected and their performance of leave-one-out evaluations is compared. We can see that increasing the depth and the number of filters, from the left (arch-i) to the right (arch-viii), improves the evaluation accuracy. Regarding to the convolutional kernel size, the size of 5 outperforms the size of 3. Therefore, the design of arch-viii is chosen as our baseline CNN architecture (see Figure 5).

Table 1. Comparison of leave-one-out accuracies on the ASL benchmark dataset (without data augmentation) with different CNN architectures (listed in columns).

CNN arch.	i	ii	iii	iv	v	vi	vii	viii
	Input ($32 \times 32 \times 1$)							
	C3-8	C5-8	C3-8	C5-8	C3-16	C5-16	C3-32	C5-32
	Maxpool							
	C3-16	C5-16	C3-16	C5-16	C3-32	C5-32	C3-64	C5-64
	Maxpool							
	FC-1024		C3-32	C5-32	C3-64	C5-64	C3-128	C5-128
			Maxpool					
			FC-512		FC-1024		FC-2048	
	FC-128							
	FC-24							
	Softmax							
Accuracy(%)	81.8	82.7	82.7	84.2	83.0	84.1	84.7	**84.8**

6.2. EVALUATION OF THE MULTIVIEW AUGMENTATION AND INFERENCE FUSION STRATEGIES

To evaluate the proposed multiview augmentation and inference fusion strategies, we compare our methods, including MVA (multiview augmentation) and MVA+IF (multiview augmentation and inference fusion) methods, to JA (jittering augmentation) method [20], which has been proved to be an effective method and is commonly used in image classification tasks.

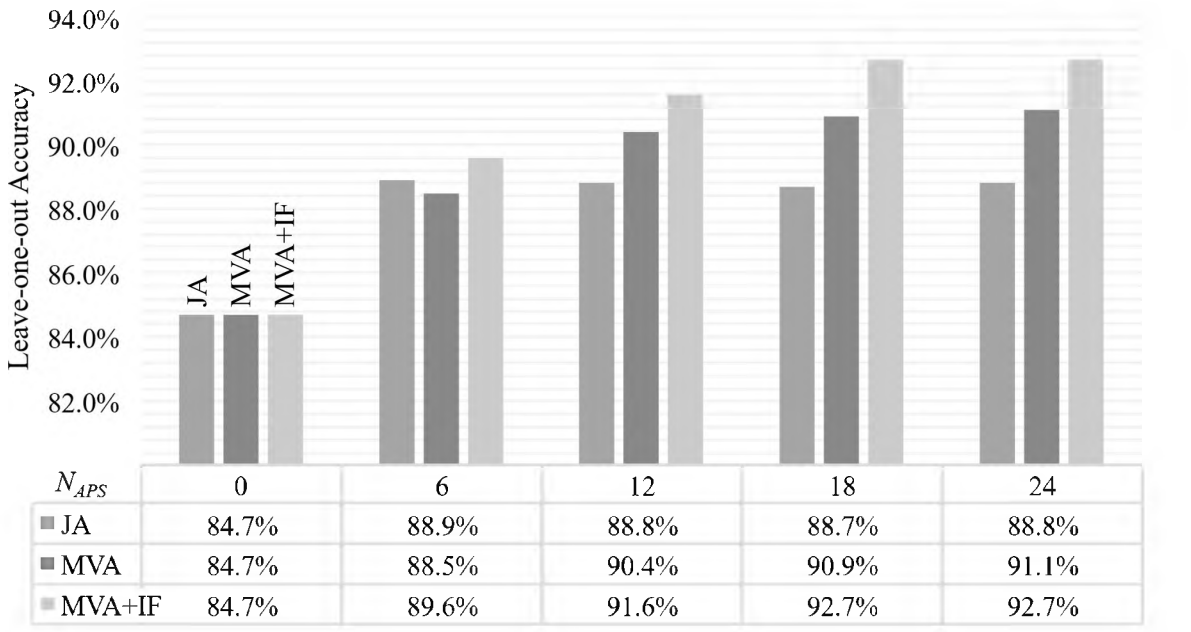
For the MVA and MVA+IF methods, four N_{APS} values 6, 12, 18 and 24 are selected, i.e., new views are generated by implementing yaw-pitch-roll rotation on the extracted point cloud around each axis for $[\pm 10°]$, $[\pm 10°, \pm 20°]$, $[\pm 10°, \pm 20°, \pm 30°]$, and $[\pm 10°, \pm 20°, \pm 30°, \pm 40°]$, yielding 6, 12, 18, and 24 augmented views for each sample, respectively. For the MVA+IF method, multiview augmentations are implemented on both the training data and the testing data. Thus, each testing image has multiple vector outputs before the IF step. Then these vector outputs are fused to generate only one probability distribution. While for the MVA method, we only augment the data in the training phase and do not augment the testing data. Therefore, each testing image has only one vector

output, from which the final prediction can be made. As for the JA method, the same four N_{APS} values are used; 6, 12, 18, and 24 augmented samples are generated by randomly translating in the range of $[-2, +2]$ pixels, scaling in the range of $[0.9, 1.1]$ ratio, and rotating in the range of [-40, +40] degrees.

The comparisons of leave-one-out accuracies on the ASL benchmark dataset are shown in Figure 11 (the half-half accuracies are not considered for comparison purpose because they are about reaching 100%). All the three augmentation methods have obvious accuracy improvements compared with the model without using data augmentation. For the JA method, using the N_{APS} of 6 improves the accuracy from 84.7% to 88.9%, but continuing to increase N_{APS} from 6 to 24 does not further improve the accuracy. The mean accuracy of the four cases ($N_{APS} = 6, 12, 18, 24$) for MVA is about 88.8%. Although the accuracy of MVA method with the N_{APS} of 6 is a little bit lower than that of JA method, the accuracy increases when increasing the N_{APS}, which outperforms JA after $N_{APS} \geq 12$. The highest accuracy of MVA (91.1%) is from the case of $N_{APS} = 24$, which is 2 percentage points higher than JA.

By implementing the multiview inference fusion, more signs are correctly recognized. MVA+IF demonstrates the highest accuracy in all the four cases and for the case of $N_{APS} = 18$ it reaches the best performance of 92.7% accuracy among the three methods. Then increasing the N_{APS} to 24 does not contribute additional improvement.

Overall, the data augmentation techniques, JA and MVA, demonstrate the effectiveness in improving the model performance, because the augmentation process introduces more natural variations to the original dataset to simulate the potential variations in the unseen samples, which pushes the CNN model to learn the most discriminative features and makes the training more robust. Meanwhile, the MVA method outperforms the JA method. It is because the finger spelling signs are highly perspective-dependent, i.e., the appearance of a sign varies significantly from different perspectives, and the MVA method can generate



N_{APS}	0	6	12	18	24
JA	84.7%	88.9%	88.8%	88.7%	88.8%
MVA	84.7%	88.5%	90.4%	90.9%	91.1%
MVA+IF	84.7%	89.6%	91.6%	92.7%	92.7%

Figure 11. Comparison of leave-one-out accuracies on the ASL benchmark dataset using the methods of JA (jittering augmentation), MVA (multiview augmentation) and MVA+IF (multiview augmentation and inference fusion) with different N_{APS} (number of augmentations per sample).

such perspective variations but the JA method can not. Furthermore, the MVA+IF method fuses the predictions of multiple perspectives to make a comprehensive inference, which results in better accuracy than the MVA method.

6.3. IMPACT OF THE NUMBER OF TOP-K CANDIDATES

To find an appropriate number of top-K candidates (the value of K) for the multiview inference fusion step described in Section 4, another set of experiments are conducted on the benchmark dataset. In these experiments, we use different K values, i.e., 3, 5, 7 and 9. Then the leave-one-out evaluation strategy is used and the accuracy evaluated on each of the five subjects is listed in Table 2. We can see that the four 'top-K' cases surpass the

'MVA' case due to the multiview inference fusion step. However, changing the K value from 3 to 9 does not affect the performance much. Therefore, we choose $K = 3$ that can provide enough entries for the fusion process.

Table 2. The leave-one-out accuracy (%) tested on each of the five subjects with different numbers of top-K candidates on the ASL benchmark dataset.

Test subject	1	2	3	4	5
MVA	92.74	86.33	94.34	87.73	88.66
MVA+IF, Top-3	93.56	88.51	94.84	91.55	91.69
MVA+IF, Top-5	93.62	88.52	94.78	91.62	91.66
MVA+IF, Top-7	93.64	88.53	94.79	91.61	91.68
MVA+IF, Top-9	93.63	88.53	94.82	91.63	91.71

6.4. PERFORMANCE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE ASL BENCHMARK DATASET

In this subsection, we compare our results with state-of-the-art performance on the ASL benchmark dataset in terms of accuracy, precision and recall with two evaluation strategies (half-half and leave-one-out). The comparison is summarized in Table 6. The highest accuracies of half-half and leave-one-out strategies in the literature are 100% [26] and 84.3%[8], respectively. For the half-half evaluation, our methods achieve 99.9%, which is almost 100% (there are only about 40 samples misclassified out of 32,831 testing samples). For the leave-one-out evaluation, our CNN model outperforms the state-of-the-art performance even without augmentations. The accuracy is improved by 4% with our implementation of the JA method. By using the MVA method, our model achieves 91% accuracy which is 2% higher than JA. After implementing MVA+IF, the accuracy is improved by another 2 percent. The best accuracy, precision and recall of our results are 92.7%, 93.5% and 92.4%, respectively.

Table 3. Performance (%) comparison on the ASL benchmark dataset.

Method	hh-A	hh-P	hh-R	loo-A	loo-P	loo-R
Pugeault et al. (2011) [17]	-	75	53	49	-	-
Keskin et al. (2012) [8]	97.8	-	-	84.3	-	-
Kuznetsova et al. (2013) [10]	87	-	-	57	-	-
Zhang et al. (2013) [26]	98.9	-	-	73.3	-	-
Rioux-Maldague and Giguére (2014) [19]	-	99	99	-	79	77
Dong et al. (2015) [5]	90	-	-	70	-	-
Maqueda et al. (2015) [13]	97.5	-	-	83.7	-	-
Wang et al. (2015) [23]	-	-	-	75.8	-	-
Zhang and Tian (2015) [25]	100	-	-	83.8	-	-
Ma and Huang (2016) [12]	84	-	-	-	-	-
Ameen and Vadera (2017) [2]	-	-	-	80.3	82	80
Nai et al. (2017) [14]	-	-	-	81.1	-	-
Our CNN	99.7	99.7	99.7	84.7	85.8	84.8
Our CNN+JA	99.9	99.9	99.9	88.9	90.2	89.0
Our CNN+MVA	99.9	99.9	99.9	90.9	91.6	90.8
Our CNN+MVA+IF	**99.9**	**99.9**	**99.9**	**92.7**	**93.5**	**92.4**

Overall, our CNN model outperforms other methods with the multiview augmentation and inference fusion strategies. It is known that the leave-one-out evaluation is a harder task than the half-half evaluation, because in the half-half experiment, all the testing subjects have already been seen by the CNN model during training; but in the leave-one-out experiment, the testing subject has not been seen. Therefore, the leave-one-out performance can demonstrate how well the trained model could be generalized to a new subject. Our model can reach 93% leave-one-out accuracy, which is a significant improvement compared to the previous best benchmark of 84% and is very promising for practical applications.

6.5. PERFORMANCE EVALUATION ON THE NTU DIGIT DATASET

We evaluate the performance of our model on the NTU digit dataset, which also achieves the best accuracies compared to other methods. The comparison is listed in Table 4. The MVA method has 100% and 99.7% for the half-half and leave-one-out accuracies, respectively, which outperforms the results reported in the literatures. The

MVA+IF method further improve the leave-one-out accuracy to 100%, which means that the multiview inference fusion strategy successfully classify the left 0.3% samples that are misclassfied using only MVA.

Table 4. Performance (%) comparison on the NTU digit dataset.

Method	hh-A	loo-A
Ren et al. (2011) [18]	93.9	-
Zhang and Tian (2015) [25]	97.5	99.0
Our CNN+MVA	100	99.7
Our CNN+MVA+IF	**100**	**100**

6.6. FEATURE VISUALIZATION

Although the CNN model demonstrates superior performance on various applications, such as the sign recognition task, it is usually taken as a black box because of its high architectural complexity and tremendous inner parameters, and its hyperparameters are tuned by experience or trial-and-error. To have a better understanding of what the CNN model has learned and what features are extracted by the convolutional operations, we visualize the learned filters and the extracted feature maps of the first convolutional layer since they can be projected into 2 dimensional images, which is shown in Figure 12. The 32 learned filters of the first convolutional layer are presented on the top. It is difficult to make some intuitive explanations on these 5×5 filters, but by reviewing the feature maps obtained from each input using these filters, we can see that some low-level features like edges and curves are extracted. For different signs, the filters are able to identify the discriminative feature elements. For example, the main difference between signs '*M*' and '*N*' is the thumb's position, which is actually learned by the filters (as shown in the feature maps of '*N*', the thumb regions are successfully emphasized compared to the feature maps

of '*M*'). Filters and feature maps of the second and third convolutional layers are not presented here because they involve high dimensional information, thus, cannot be projected to images for visualization purpose.

6.7. FAILURE CASE STUDIES

In this subsection, we discuss the signs that are not correctly classified in the leave-one-out evaluations on the benchmark dataset. The overall mean F scores for the 24 signs are illustrated in Figure 13. The model has great performance (> 95%) on the signs '*B*', '*C*', '*D*', '*F*', '*I*', '*L*', '*O*', '*U*', '*W*', '*X*', and '*Y*'. However, for the signs of '*E*', '*K*', and '*Q*' , the F scores are lower than 85% due to their high subjectwise variations. For example, as shown in Figure 13, different subjects perform the sign of '*K*' in different ways, thus it is difficult for the model to be generalized to the unseen subject in the leave-one-out evaluations.

The confusion matrices and the most confusing sign pairs of the five subjects are shown in Figures 14, 15, 16, 17 and 18, respectively. We can see that, different subjects show different performance on different signs in the leave-one-out evaluations. For the 1st subject (see Figure 14), there are six confusing pairs severely misclassified, which are '*K-G*', '*N-T*', '*R-K*', '*R-U*', '*V-K*', and '*X-G*'. For example, there are 101 '*V*' misclassified as '*K*' because of the high similarity between them (i.e., both have the index and middle fingers pointing up). For the 2nd subject (see Figure 15), the most confusing pairs are '*G-H*', '*V-K*', and '*T-E*'. For the 3rd subject showing the best performance (see Figure 16), most of the signs are successfully classified except the most confusing pairs '*A-T*' and '*K-L*', where there are 99 '*A*' and 109 '*K*' misclassified as '*T*' and '*L*', respectively. The most confusing pairs of the 4th (see Figure 17) and 5th (see Figure 18) subjects are '*G-H*', '*N-M*', '*T-N*', and '*E-S*', '*V-K*', '*Q-P*', respectively.

By reviewing these failure cases, we find that the high similarity between the confusing pairs makes it difficult to distinguish them, and the significant subjectwise difference for the same sign makes it difficult to learn this kind of unseen variations beforehand.

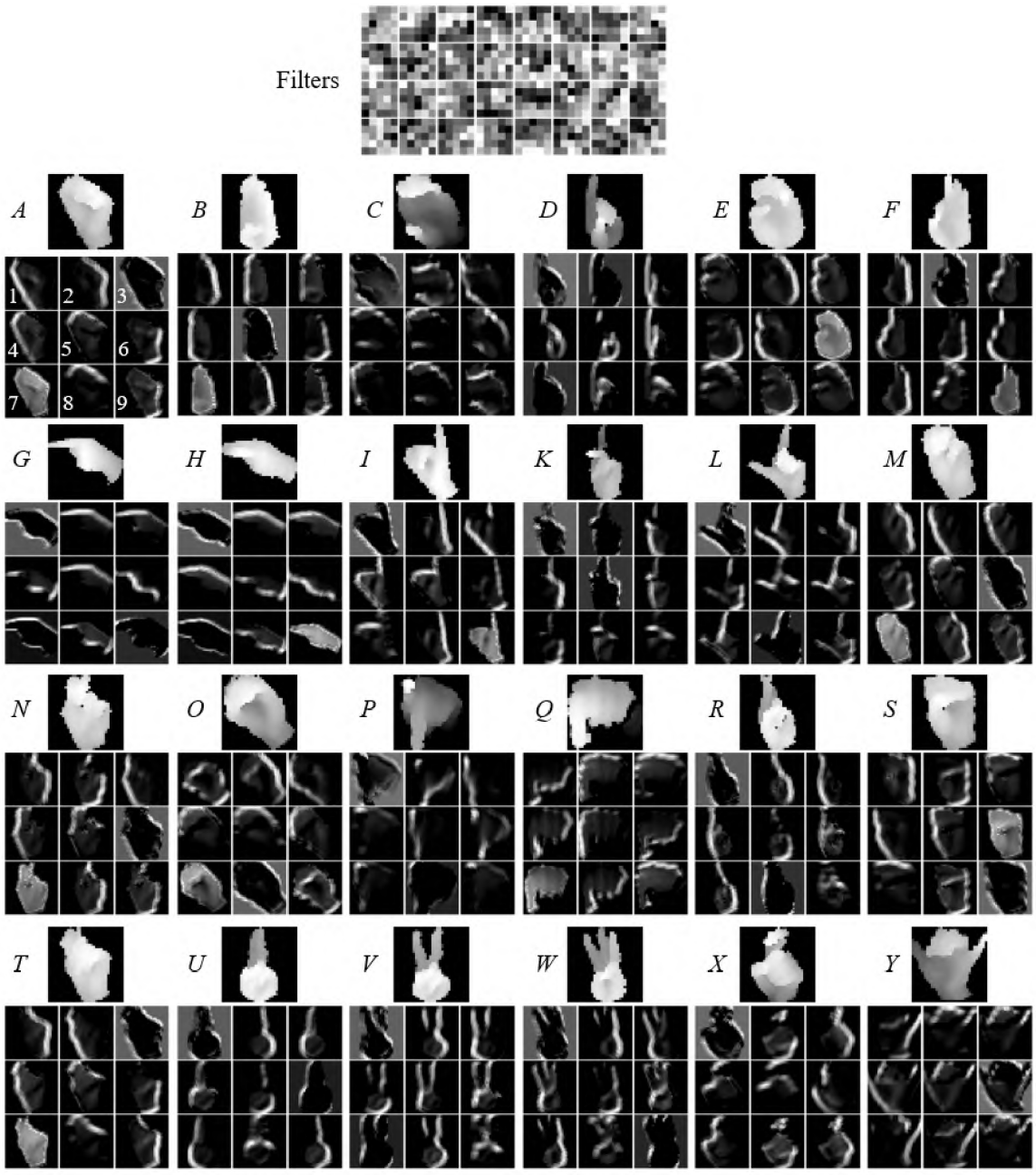


Figure 12. Visualization of the 32 (4 *rows* × 8 *columns*) learned filters (top) of the first convolutional layer, and the top 9 feature maps (the sequence is indexed as shown in *A*'s feature maps) for each of the 24 signs in a trained model on the ASL benchmark dataset.

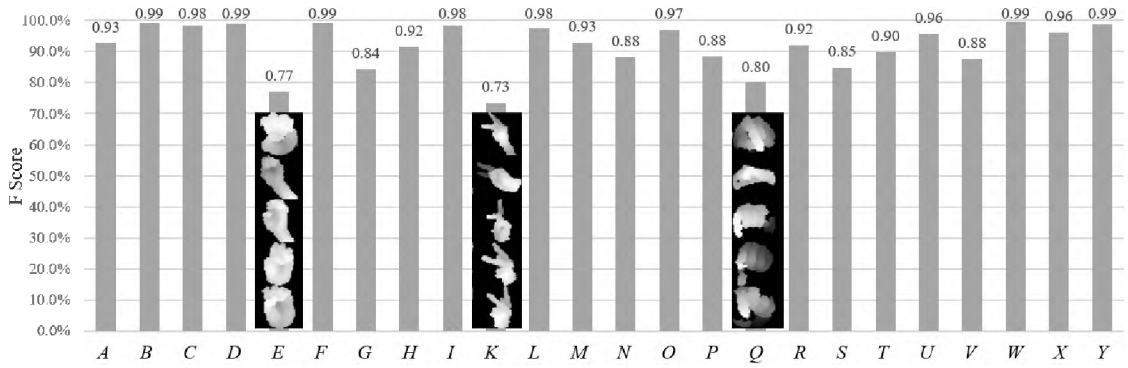


Figure 13. Mean F score of each of the 24 signs in the leave-one-out evaluations on the ASL benchmark dataset.

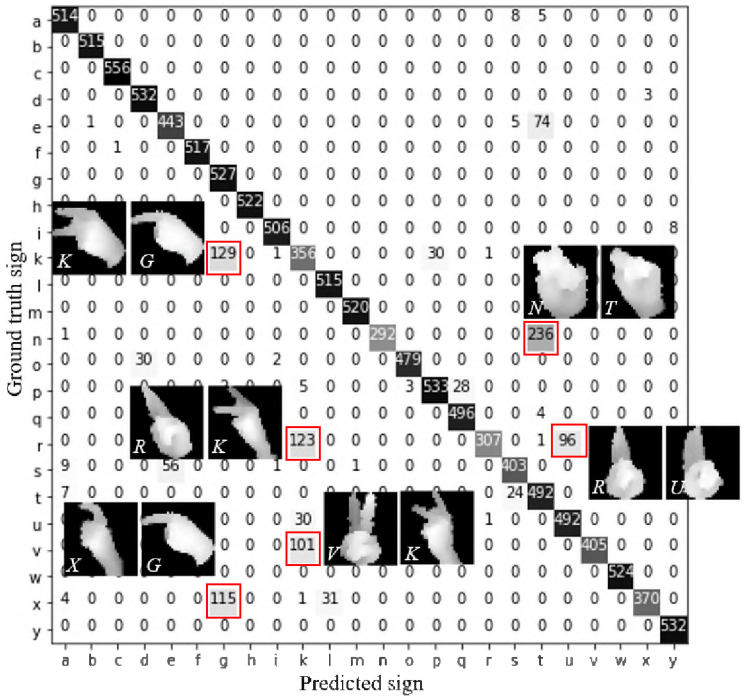


Figure 14. Confusion matrix and the most confusing pairs of the leave-one-out evaluation on the ASL benchmark dataset tested on the 1st subject.

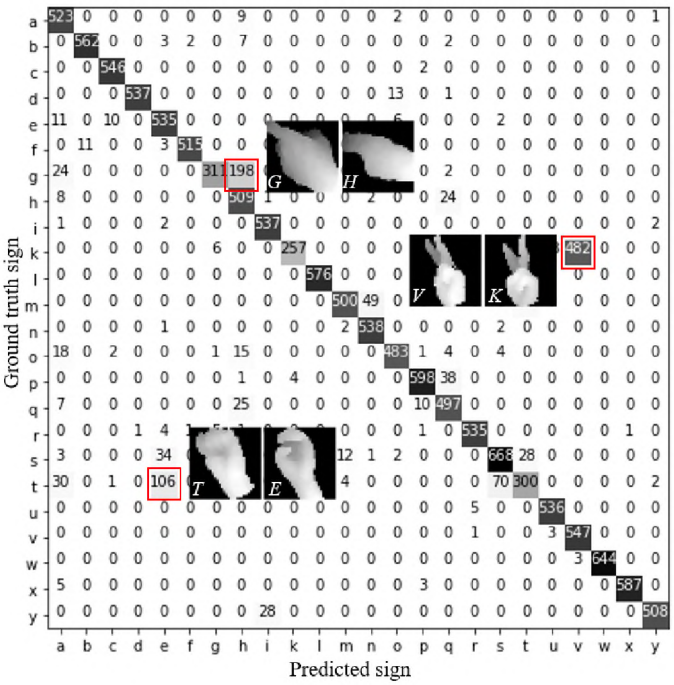


Figure 15. Confusion matrix and the most confusing pairs of the leave-one-out evaluation on the ASL benchmark dataset tested on the 2nd subject.

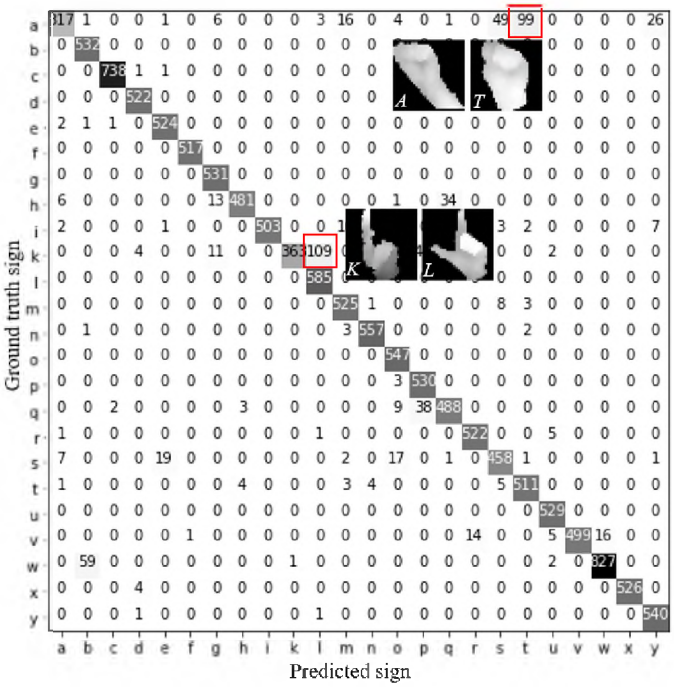


Figure 16. Confusion matrix and the most confusing pairs of the leave-one-out evaluation on the ASL benchmark dataset tested on the 3rd subject.

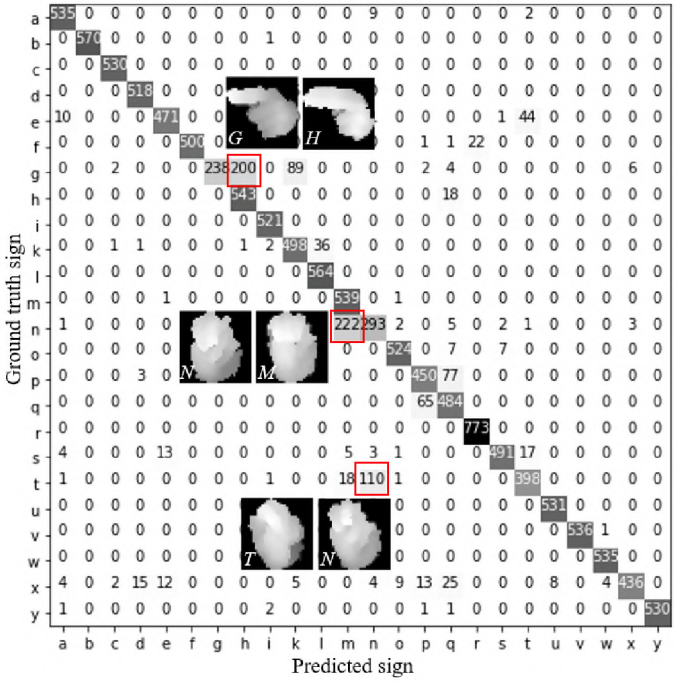


Figure 17. Confusion matrix and the most confusing pairs of the leave-one-out evaluation on the ASL benchmark dataset tested on the 4th subject.

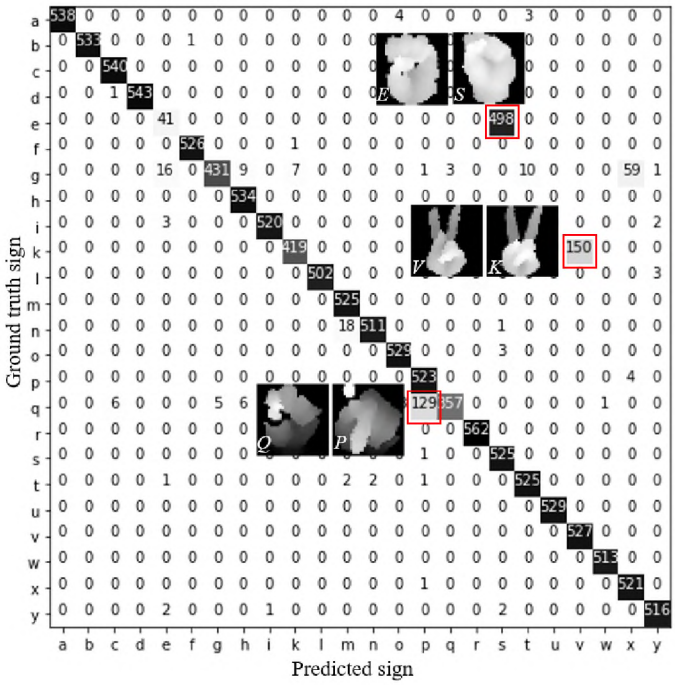


Figure 18. Confusion matrix and the most confusing pairs of the leave-one-out evaluation on the ASL benchmark dataset tested on the 5th subject.

To address these failure cases for further improving the performance, some future work can be explored: (1) more subjects can be considered to include more signing styles for training the model; (2) 3D reconstruction can be implemented to recover more information from the depth image than the current 3D point cloud; (3) the hand skeleton information can be extracted to obtain some skeleton-based features for classification; (4) the RGB images can be included in the model; and (5) the architecture of the CNN model can be explored to improve its performance and efficiency.

7. CONCLUSIONS

In this paper, we propose a novel method of multiview augmentation and inference fusion for ASL alphabet recognition from depth images using a Convolutional Neural Network (CNN). Multiview augmentation first retrieves the 3D information embedded in a depth image, and then generates more data for different perspective views. The result has shown that it outperforms the traditional image augmentation methods because it can simulate realistic perspective variations that the traditional methods cannot. Inference fusion copes with the interclass similarity issues caused by perspective variations and finger occlusions. It comprehends information of all individual views, and then outputs the final prediction, which has been proved to be effective in further improving the model's performance. Our method has been successfully evaluated on two public datasets, the ASL benchmark dataset and the NTU digit dataset. The experimental results have demonstrated that our method makes significant improvement compared to the previous work, achieving recognition accuracies of 100% and 93% in the half-half and the leave-one-out experiments, respectively, on the ASL benchmark dataset, and achieving recognition accuracies of 100% for both the half-half and the leave-one-out experiments on the NTU digit dataset.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., 'TensorFlow: Large-scale machine learning on heterogeneous systems,' 2015, software available from tensorflow.org.

[2] Ameen, S. and Vadera, S., 'A convolutional neural network to classify american sign language fingerspelling from depth and colour images,' Expert Systems, 2017.

[3] Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., and Massaroni, C., 'Exploiting recurrent neural networks and leap motion controller for sign language and semaphoric gesture recognition,' arXiv preprint arXiv:1803.10435, 2018.

[4] Bradski, G., 'The OpenCV Library,' Dr. Dobb's Journal of Software Tools, 2000.

[5] Dong, C., Leu, M. C., and Yin, Z., 'American sign language alphabet recognition using microsoft kinect,' in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops,' 2015 pp. 44–52.

[6] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.

[7] He, K., Zhang, X., Ren, S., and Sun, J., 'Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,' in 'Proceedings of the IEEE international conference on computer vision,' 2015 pp. 1026–1034.

[8] Keskin, C., Kıraç, F., Kara, Y. E., and Akarun, L., 'Hand pose estimation and hand shape classification using multi-layered randomized decision forests,' in 'European Conference on Computer Vision,' Springer, 2012 pp. 852–863.

[9] Kingma, D. and Ba, J., 'Adam: A method for stochastic optimization,' arXiv preprint arXiv:1412.6980, 2014.

[10] Kuznetsova, A., Leal-Taixé, L., and Rosenhahn, B., 'Real-time sign language recognition using a consumer depth camera,' in 'Proceedings of the IEEE International Conference on Computer Vision Workshops,' 2013 pp. 83–90.

[11] LaValle, S. M., *Planning algorithms*, Cambridge university press, 2006.

[12] Ma, L. and Huang, W., 'A static hand gesture recognition method based on the depth information,' in 'Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016 8th International Conference on,' volume 2, IEEE, 2016 pp. 136–139.

[13] Maqueda, A. I., del Blanco, C. R., Jaureguizar, F., and García, N., 'Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns,' Computer Vision and Image Understanding, 2015, **141**, pp. 126–137.

[14] Nai, W., Liu, Y., Rempel, D., and Wang, Y., 'Fast hand posture classification using depth features extracted from random line segments,' Pattern Recognition, 2017, **65**, pp. 1–10.

[15] Oz, C. and Leu, M. C., 'Recognition of finger spelling of american sign language with artificial neural network using position/orientation sensors and data glove,' in 'International Symposium on Neural Networks,' Springer, 2005 pp. 157–164.

[16] Oz, C. and Leu, M. C., 'Linguistic properties based on american sign language isolated word recognition with artificial neural networks using a sensory glove and motion tracker,' Neurocomputing, 2007, **70**(16), pp. 2891–2901.

[17] Pugeault, N. and Bowden, R., 'Spelling it out: Real-time asl fingerspelling recognition,' in 'Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on,' IEEE, 2011 pp. 1114–1119.

[18] Ren, Z., Yuan, J., and Zhang, Z., 'Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera,' in 'Proceedings of the 19th ACM international conference on Multimedia,' ACM, 2011 pp. 1093–1096.

[19] Rioux-Maldague, L. and Giguere, P., 'Sign language fingerspelling classification from depth and color images using a deep belief network,' in 'Computer and Robot Vision (CRV), 2014 Canadian Conference on,' IEEE, 2014 pp. 92–97.

[20] Sermanet, P. and LeCun, Y., 'Traffic sign recognition with multi-scale convolutional networks,' in 'Neural Networks (IJCNN), The 2011 International Joint Conference on,' IEEE, 2011 pp. 2809–2813.

[21] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R., 'Real-time human pose recognition in parts from single depth images,' Communications of the ACM, 2013, **56**(1), pp. 116–124.

[22] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., 'Dropout: a simple way to prevent neural networks from overfitting.' Journal of machine learning research, 2014, **15**(1), pp. 1929–1958.

[23] Wang, C., Liu, Z., and Chan, S.-C., 'Superpixel-based hand gesture recognition with kinect depth camera,' IEEE transactions on multimedia, 2015, **17**(1), pp. 29–39.

[24] Wohlkinger, W. and Vincze, M., 'Ensemble of shape functions for 3d object classification,' in 'Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on,' IEEE, 2011 pp. 2987–2992.

[25] Zhang, C. and Tian, Y., 'Histogram of 3d facets: A depth descriptor for human action and hand gesture recognition,' Computer Vision and Image Understanding, 2015, **139**, pp. 29–39.

[26] Zhang, C., Yang, X., and Tian, Y., 'Histogram of 3d facets: A characteristic descriptor for hand gesture recognition,' in 'Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on,' IEEE, 2013 pp. 1–8.

III. MULTI-MODAL RECOGNITION OF WORKER ACTIVITY FOR HUMAN-CENTERED INTELLIGENT MANUFACTURING

Wenjin Tao[a], Ming C. Leu[a], Zhaozheng Yin[b]

[a]Missouri University of Science and Technology, Rolla, MO 65409, USA

[b]Stony Brook University, Stony Brook, NY 11794, USA

ABSTRACT

In a human-centered intelligent manufacturing system, sensing and understanding of the worker's activity are the primary tasks. In this paper, we propose a multi-modal approach for worker activity recognition using Inertial Measurement Unit (IMU) signals obtained from a Myo armband and videos from a visual camera. Specifically, for the IMU signals, we design two novel feature transform mechanisms, in both frequency and spatial domains, to assemble the captured IMU signals as images, which allow using convolutional neural networks to learn the most discriminative features. Along with the above two modalities, we propose two other modalities for the video data, at the video frame and video clip levels, respectively. Each of the four modalities returns a probability distribution on activity prediction. Then, these probability distributions are fused to output the worker activity classification result. A worker activity dataset of 6 activities is established, which at present contains 6 common activities in assembly tasks, i.e., grab a tool/part, hammer a nail, use a power-screwdriver, rest arms, turn a screwdriver, and use a wrench. The developed multi-modal approach is evaluated on this dataset and achieves recognition accuracies as high as 97% and 100% in the leave-one-out and half-half experiments, respectively.

Keywords: Worker activity recognition; multi-modal fusion ; deep learning; intelligent manufacturing ; human-centered computingg

1. INTRODUCTION

Industrial big data has been increasingly accessible and affordable, benefiting from the availability of low-cost sensors and the development of Internet-of-Things (IoT) technologies [8, 16], which builds up the data foundation for advanced manufacturing. A variety of methods and algorithms have been developed to learn valuable information from the data, and to make the manufacturing more intelligent [19]. With the recent fast growing of Artificial Intelligence (AI) technologies, especially deep learning [15] and reinforcement learning [13] methods, AI boosted manufacturing has been increasingly attractive in both the scientific research and industrial applications.

In an intelligent manufacturing system involving workers, recognition of the worker's activity is one of the primary tasks. It can be used for quantification and evaluation of the worker's performance, as well as to provide onsite instructions with augmented reality. Also, worker activity recognition is crucial for human-robot interaction and collaboration. It is essential for developing human-centered intelligent manufacturing systems.

1.1. RELATED WORK

In the computer vision area, image/video-based human activity recognition using deep learning methods has been intensively studied in recent years and unprecedented progress has been made [3, 9]. However, visual-based recognition suffers from the occlusion issue, which affects the recognition accuracy. Wearable devices, such as an armband embedded with an Inertial Measurement Unit (IMU), directly sense the movement of human body, which can provide information on the body status. In addition, there are a lot of inexpensive wearable devices in the market, such as Myo armbands [32] and smartphones, which are widely used in activity recognition tasks. Wearable devices are directly attached to the human body and thus do not have the occlusion issue. Nevertheless, a wearable device can only sense the human body activity locally, it is challenging to precisely recognize

an activity involving multiple body parts. Although multiple devices can be applied to simultaneously sense the activity globally, it makes the system cumbersome and brings discomfort to the user.

Worker activity recognition in the manufacturing area is still an emerging topic and few studies have been made. Stiefmeire et al. [27] utilized ultrasonic and IMU sensors for worker activity recognition in a bicycle maintenance scenario using a Hidden Markov Model classifier. Later they proposed a string-matching based segmentation and classification method using multiple IMU sensors for recognizing worker activity in car manufacturing tasks [28, 29]. Koskimaki et al. [14] used a wrist-worn IMU sensor to capture the arm movement and a K-Nearest Neighbor model to classify five activities for industrial assembly lines. Maekawa et al. [17] proposed an unsupervised measurement method for lead time estimation of factory work using signals from a smartwatch with an IMU sensor. Recently, deep learning methods have been introduced to recognize worker activity in human-robot collaboration studies [21, 34].

In general, the activity recognition task can be broken down into two subtasks: feature extraction and subsequent multiclass classification. To extract more discriminative features, various methods have been applied to the raw signals in the time or frequency domain, e.g., mean, correlation, and Principal Component Analysis [2, 4, 20, 23]. Different classifiers have been explored on the features for activity recognition, such as the Support Vector Machine [2, 4], Random Forest, K-Nearest Neighbors, Linear Discriminant Analysis [20], and Hidden Markov Model [23]. To effectively learn the most discriminative features, Jiang et al. [11] proposed a method based on Convolutional Neural Networks (CNN). They assembled the raw IMU signals into an activity image, which enabled the CNN model to automatically learn the discriminative features from the activity image for classification.

1.2. PROPOSED METHOD

Few attempts have been made for the worker activity recognition in the manufacturing field, and most of them only use single sensing modality, which cannot guarantee robust recognition under various circumstances. In the present research, to comprehensively perceive the worker, we choose a Myo armband to acquire the Inertial Measurement Unit (IMU) signals and a visual camera to capture the image sequence of the worker's activity. An overview of our method is illustrated in Figure 1. For the IMU signals, we design two novel mechanisms, in both the frequency and spatial domains, to assemble the captured IMU signals as images. The assembled signal representation allows us to use Convolutional Neural Networks to explore the correlation among time-series signals and learn the most discriminative features for worker activity recognition. As for the video data, we propose two modalities, at the frame and video-clip levels, respectively. Overall, we have four modalities in parallel and each of the four modalities can return a probability distribution on the activity recognition. Then these probabilities are fused to output the worker activity classification result. To evaluate the method, a worker activity dataset containing 6 common activities in assembly tasks is established.

The main contributions of our work are as follows:

1. We propose a multi-modal approach for the worker activity recognition in manufacturing, using both wearable devices and visual cameras.

2. To take advantage of the powerful learning ability of CNN on images, we design two novel mechanisms to produce 2D signal representations of the IMU signals from wearable devices, in both the frequency and spatial domains.

3. To synthesize more physical-realistic variations in the training dataset, we propose a kinematics-based data augmentation method for the wearable sensor data. It generates more data by spatial rotation and mirroring, in order to augment variations that cannot be achieved using traditional image augmentation methods.

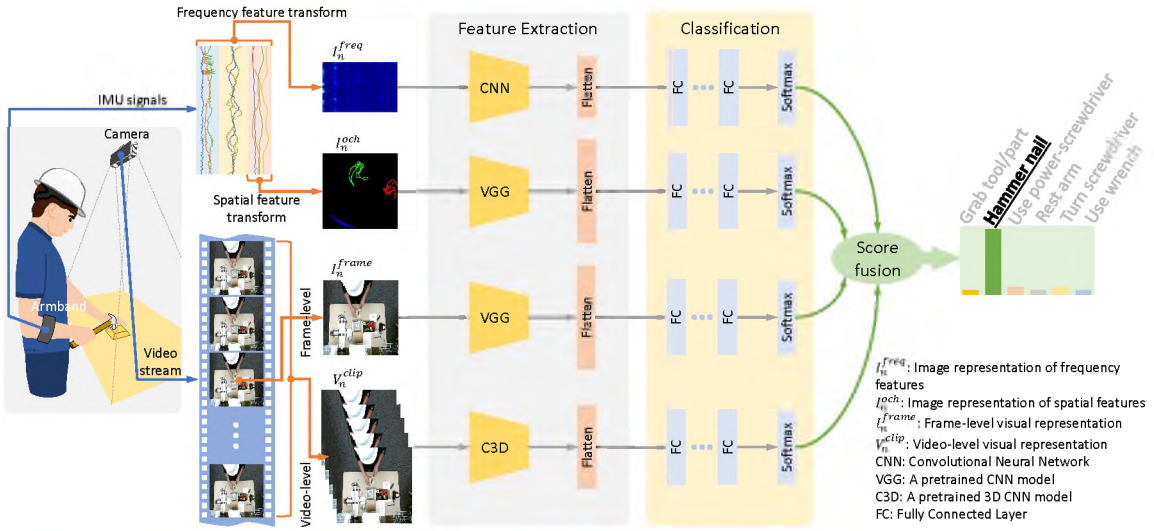


Figure 1. Overview of our multi-modal approach for worker activity recognition.

The remainder of this paper is organized as follows. Section 2 discusses how we build up the worker activity dataset. Section 3 focuses on the novel feature representation and data augmentation. Section 4 describes the details of neural network architectures, training and testing of the multi-modal activity recognition. The experimental setups and results are described in Sections 3 and 6, respectively. Finally, Section 4 provides the conclusions of this research.

2. MULTI-MODAL SENSING AND DATA ACQUISITION

To establish our dataset of worker activity, six activities commonly performed in assembly tasks are chosen, which are: grab a tool/part (GT), hammer a nail (HN), use a power-screwdriver (UP), rest arms (RA), turn a screwdriver (TS), and use a wrench (UW). There are 8 subjects recruited to conduct a set of tasks (listed in Table I) containing the 6 activities.

Table 1. Tasks for collecting worker activity.

No.	Tasks	Activities
1	Grab 30 tools/parts from the 3 containers	GT
2	Hammer 15 nails into the wooden dummy	HN
3	Tighten 20 screws using a power-screwdriver	UP
4	Rest arms for about 60 seconds	RA
5	Tighten 10 nuts using a screwdriver	TS
6	Tighten 10 nuts using a wrench	UW

As demonstrated in Figure 3(a), the subject is asked to stand in front of the workbench, wear a Myo armband on his/her right forearm with a fixed orientation (Figure 3(b)), and perform the tasks on assembly dummies in a natural way. The Myo armband from Thalmic Labs is equipped with IMU sensors for wearable sensor data acquisition. The IMU returns three types of signals (3-channel acceleration, 3-channel angular velocity, and 4-channel orientation) at the sample rate of 50Hz. These 10-channel signals captured on a worker are transmitted via Bluetooth to the workstation in real time.

While collecting wearable sensor data from the Myo armband, an overhung camera is used to record the assembly tasks simultaneously for monitoring the process. Examples of the 6 activities are shown in Figure 3, which are taken from the overhung camera.

3. DATA PREPROCESSING, SIGNAL REPRESENTATION AND DATA AUGMENTATION

Convolution-based deep learning methods need the input data to be formatted as tensors, for example, with a fixed size of $h \times w \times c$ for images or with a fixed size of $h \times w \times c \times l$ for image sequences (video clips) where h, w and c are the height, width and the number of channels of the image, respectively, and l is the image sequence length. Therefore, some preprocessing steps are necessary before the data can be fed into a convolutional neural network. In this section we give a detailed description of the pipeline for data preprocessing

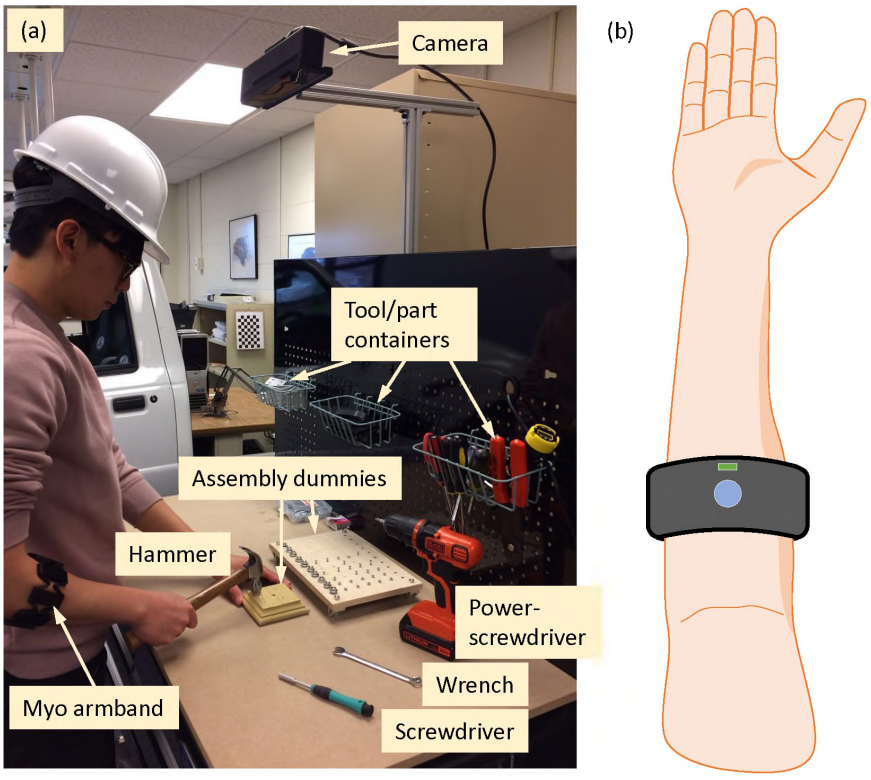


Figure 2. (a) Data collection setup; (b) Wearing orientation of a right-hand.

and the new methods for signal representation. Furthermore, to generate more realistic data, we propose a kinematics-based augmentation method which is also presented in this section.

3.1. DATA SAMPLING

Although the data (i.e., Myo sensor signals and videos) are collected simultaneously for all tasks and each task consists of only one activity, there still might be some unrelated activities inside the data, such as preparing activities before hammering nails. To address it, the recorded videos are manually annotated to locate the time durations (i.e., the starting and ending timestamps), each of which contains only one of the six activities. These durations are used to segment the raw data (Myo sensor signals and videos).

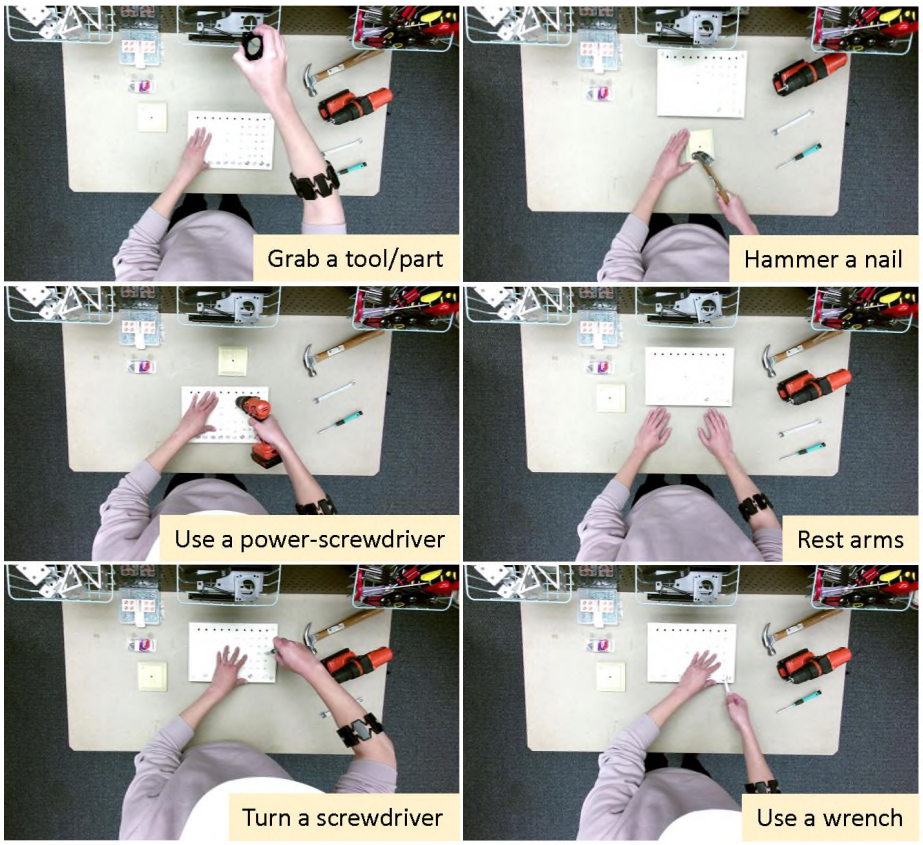


Figure 3. Examples of the 6 activities captured from the overhung camera.

Usually, the duration of an activity instance ranges from a few seconds to more than one minute. Thus, sampling is needed to prepare the data samples for recognition. As depicted in Figure 4, the 10-channel IMU signals and the video recording are synchronized with the timestamps. Then the 50Hz IMU signals are sampled using a temporal sliding window with the width of $T = 64$ timestamps and 75% overlap between two windows. Thus, each IMU sample lasts for about 1.3 seconds, which covers at least one activity pattern. After sampling the IMU signals, the video recordings are sampled according to the time durations of the IMU samples. Then, each video clip has an approximate length of 38 frames.

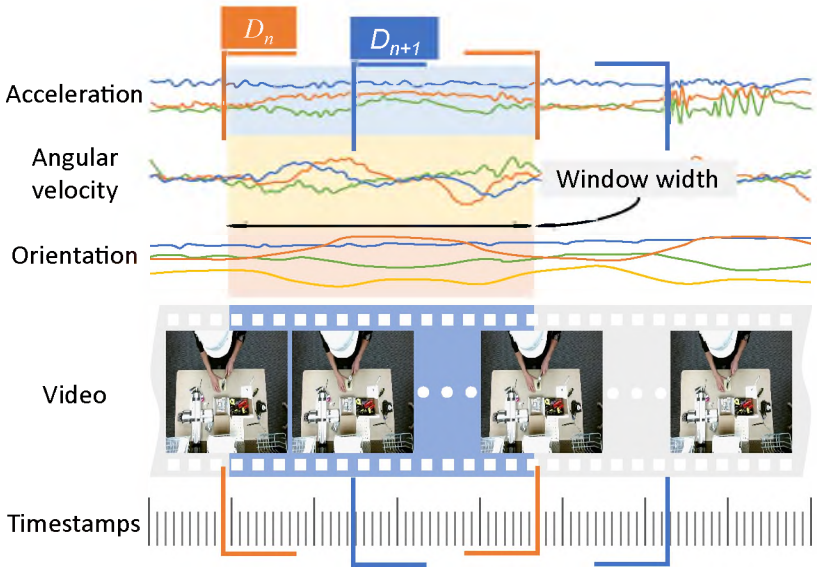


Figure 4. Scheme of the signal sampling method.

After sampling, we denote our dataset as $\mathbb{D} = \{D_1, \cdots, D_n, \cdots, D_N\}$

$$D_n = \{[s_n, v_n], y_n\}, n \in [1, N] \tag{1}$$

where s_n is a sample set of time-series IMU signals, v_n is the corresponding video clip sample, and y_n is the manually labeled ground truth of the activity class. More specifically, s_n a sequence of discrete-time data over T timestamps, $s_n = \{s_{n,1}, \cdots, s_{n,t}, \cdots, s_{n,T}\}$, and each element is elaborated as

$$s_{n,t} = [\underbrace{a^x_{n,t}, a^y_{n,t}, a^z_{n,t}}_{a_{n,t}\text{: acceleration}}, \underbrace{g^x_{n,t}, g^y_{n,t}, g^z_{n,t}}_{g_{n,t}\text{: gyro}}, \underbrace{q^x_{n,t}, q^y_{n,t}, q^z_{n,t}, q^w_{n,t}}_{q_{n,t}\text{: orientation}}], \quad t \in [1, T], \tag{2}$$

where a, g, and q are acceleration, angular velocity, and orientation in quaternion, respectively.

After sampling, the quantitative information of the dataset is listed in Table II. There are 11,211 data samples in total. The eight subjects use different amounts of time to finish each task, therefore they have different numbers of data samples for each activity.

Table 2. Number of data samples for each activity of different subjects.

Subject No.	GT	HN	UP	RA	TS	UW
1	193	140	364	266	222	442
2	302	408	195	56	274	751
3	198	183	171	251	214	567
4	204	172	188	29	82	344
5	187	204	142	43	213	372
6	216	77	179	47	129	301
7	213	196	203	254	231	576
8	200	184	262	145	148	273
Total	1713	1564	1704	1091	1513	3626

3.2. WEARABLE SENSOR SIGNAL REPRESENTATION

To take advantage of the powerful learning ability of CNNs on images, we propose to transfer the time-series IMU sensor signals to the image representation. As shown in Figure 5, the frequency feature transform assembles the sensor signals in a special pattern such that the hidden correlations among different channels of sensor signals are revealed; and the spatial feature transform uncovers the changing history of orientation signals in the spatial domain. Both feature transform mechanisms enable a CNN model to learn the most discriminative features from images, which are not possible in the original time-series sensor signals.

3.2.1. Frequency Feature Transform. Frequency domain analysis is a commonly used technique for signal pattern recognition. Rather than directly applying the frequency transform to time-series signals, we propose a new way to unveil the hidden correlations among sensor signals: 1) The 10-channel signals s_n in an IMU sample are stacked row by row as an image $I_n^{stacked}$ with the size of 10×64 (Figure 5(a)); 2) We expand the 10-row

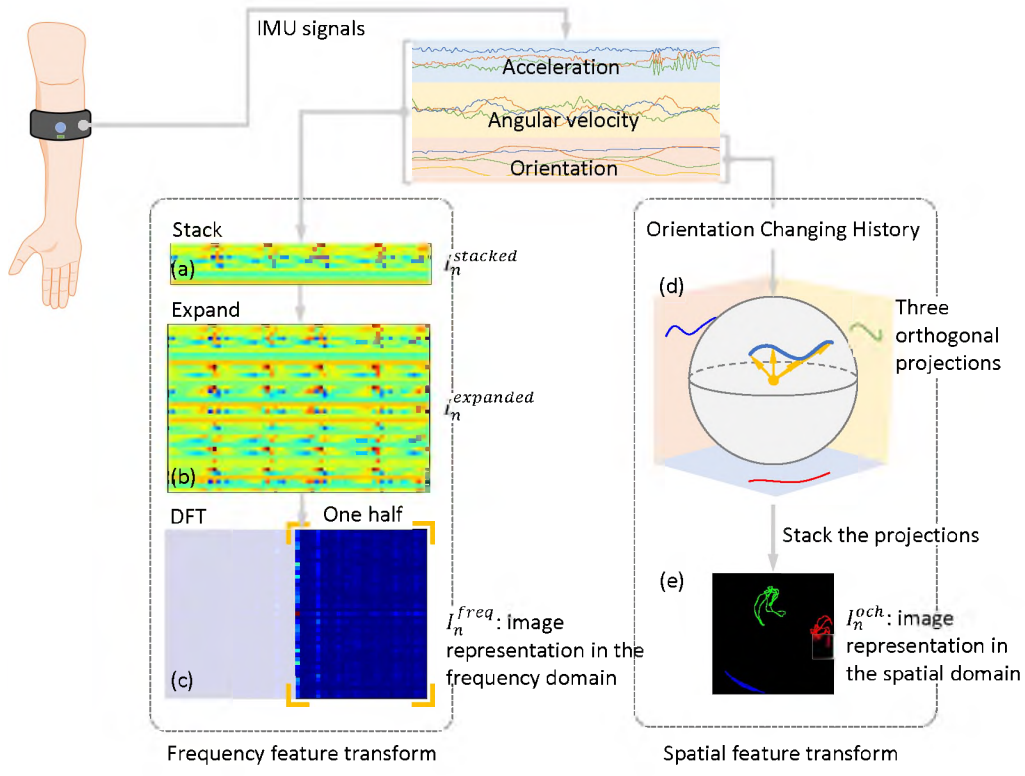


Figure 5. Illustration of the feature transforms for wearable sensor signals.

image with a shuffling algorithm [11] to form $I_n^{expanded}$ (Figure 5(b)) with the size of 42×64. The idea here is to make every pair of 10 channels have the chance to be row-neighbors in the image, then the correlations among different channels can be exposed and be further detected by a CNN model; 3) Two-dimensional (2D) Discrete Fourier Transform (DFT) is applied to $I_n^{expanded}$ to get the representation in the frequency domain to analyze the frequency characteristics. Only its logarithmic magnitude is taken to form the image I_n^{freq} (Figure 5(c)); 4) Due to the conjugate symmetry of Fourier Transforms

$$\begin{cases} I_n^{freq}(u,v) = I_n^{freq}(-u,-v), \\ I_n^{freq}(-u,v) = I_n^{freq}(u,-v), \end{cases} \tag{3}$$

where u and v represent the two directions of an image, we can use only a half to represent the DFT image to remove the redundancy. This will reduce the architectural complexity and the number of training parameters for the CNN model. Here we keep using the notation I_n^{freq} to represent the one-half (the first and fourth quadrants) of DFT image for simplicity.

3.2.2. Spatial Feature Transform. Implementing feature transform in the frequency domain unavoidably abandons the spatial information from the signals, which motivates us to introduce the second mechanism to exploit the spatial information included in the raw signals. Since recovering the spatial trajectory from IMU data is not an easy task, here we develop an *orientation changing history (och)* image to represent the pose-changing information of the subject in the spatial domain

$$I_n^{och} = \mathcal{T}_{och}(q_n) \tag{4}$$

where $\mathcal{T}_{och}$ is the spatial feature transform and I_n^{och} is the resulted image. In the spatial feature transform described below, only the orientation information q_n is considered.

First, a unit vector $\vec{v}_{ref} = [0, 0, 1]$ is rotated by q_n to generate a direction vector $\vec{v}_{n,t}$ by

$$\vec{v}_{n,t} = \vec{q}_{n,t} * \vec{v}_{ref} \tag{5}$$

where $*$ denotes the rotation operation defined as

$$\vec{q} * \vec{v} = [(\vec{q} \otimes [v^x, v^y, v^z, 0]) \otimes \vec{q}^*]_{1:3} \tag{6}$$

where $\otimes$ is the quaternion multiplication, defined as

$$\vec{q}_1 \otimes \vec{q}_2 = \begin{bmatrix} q_1^w q_2^x + q_1^x q_2^w + q_1^y q_2^z - q_1^z q_2^y \\ q_1^w q_2^y + q_1^y q_2^w + q_1^z q_2^x - q_1^x q_2^z \\ q_1^w q_2^z + q_1^z q_2^w + q_1^x q_2^y - q_1^y q_2^x \\ q_1^w q_2^w - q_1^x q_2^x - q_1^y q_2^y - q_1^z q_2^z \end{bmatrix}^T \tag{7}$$

where $\vec{q}_1 = [q_1^x, q_1^y, q_1^z, q_1^w]$, $\vec{q}_2 = [q_2^x, q_2^y, q_2^z, q_2^w]$, and $\vec{q}^*$ is the conjugate of $\vec{q}$:

$$\vec{q}^* = [-q^x, -q^y, -q^z, q^w]. \tag{8}$$

Then, the orientation changing history can be represented by a series of orientation vectors at different time steps.

$$\mathbf{v}_n^{och} = [v_{n,1}, v_{n,2}, \ldots, v_{n,t}], t \in [1, T] \tag{9}$$

which is essentially a set of points on the unit sphere surface.

Secondly, these points are projected onto three orthogonal planes (Figure 5(d)). On each plane, the points are connected with line segments sequentially to form orientation changing curves in an image.

Finally, these three projected images are stacked as a 3-channel image I_n^{och} which is represented in red, green and blue color, respectively (Figure 5(e)). Figure 6 shows some examples of image representations in the frequency and spatial domain, from one subject on six activities, from which we can observe unique patterns of each activity.

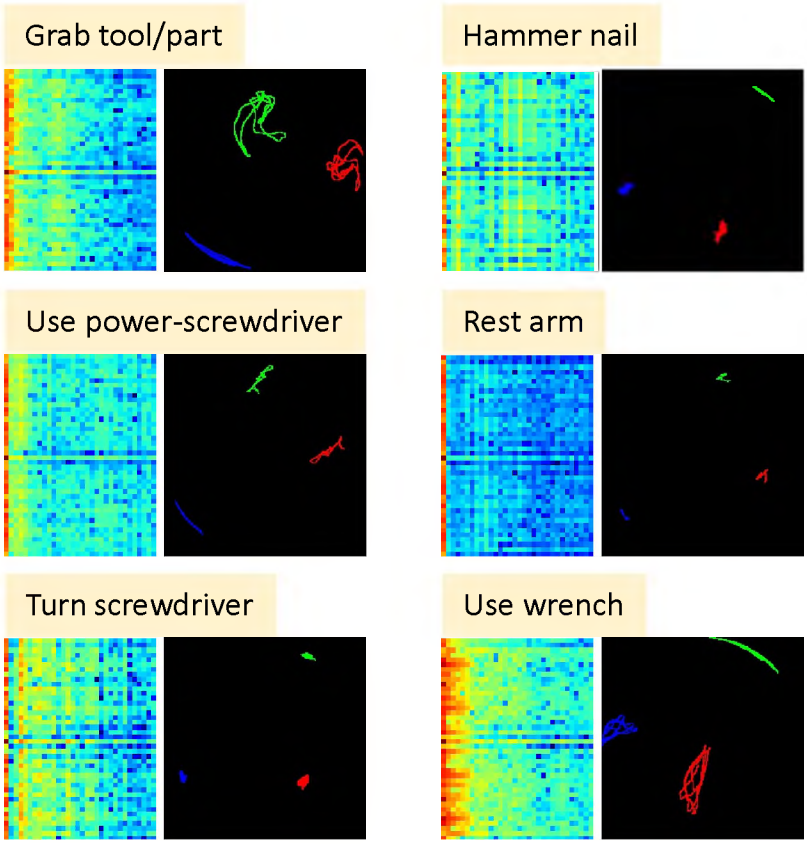


Figure 6. Examples of IMU image representations by the frequency and spatial feature transforms.

3.3. VISUAL SIGNAL REPRESENTATION

Besides the two mechanisms of feature transforms on the IMU sensor signals, since the recorded video contains rich visual contexts of the worker's activity and visual-based activity recognition also has shown promising results [3, 9], we introduce two other mechanisms to represent the video at two levels.

3.3.1. Frame-Level Visual Representation. At the frame level, the middle frame of a video clip is selected as an image representation of an activity, which focuses on the worker's static posture and surrounding environment. The operation is denoted as

$$I_n^{frame} = \mathcal{T}_{frame}(v_n) \tag{10}$$

3.3.2. Video-Level Visual Representation. At the video-clip level, the video clip samples are sampled again to make each video clip have the defined length of frames for a CNN model (to be described in Section 4). The operation is

$$V_n^{clip} = \mathcal{T}_{clip}(v_n) \tag{11}$$

where V_n is the resulted fixed-length video-clip from the operation of $\mathcal{T}_{clip}$.

3.4. KINEMATICS-BASED DATA AUGMENTATION

For deep learning, a large amount of labeled data are needed to train a valid model with a decent performance of generalization. Nevertheless, it is always time-consuming and costly to collect enough data with annotated labels. Data augmentation that synthesizes additional data derived from original ones, is a commonly-used technique to resolve the data shortage problem. Traditionally, image data augmentation refers to implementing a series of image transformation techniques on the original images, which may consist of rotating, scaling, shifting, flipping, shearing, etc., to generate more image data. The image transformation is able to introduce more variations and still keep the recognizable contents, and thus it is applied to generate more data for images and video-clips from the visual signal representation. However, the variations introduced by the basic image transformation is not physically-realistic in our sensor signal context.

To include more reasonable variations in the training dataset, we propose a kinematics-based augmentation method to generate more wearable sensor signal samples, rather than implementing image data augmentation on those images resulted from feature transforms. More specifically, the kinematics-based augmentation refers to creating variations by spatial rotation and mirroring on the four channels of orientation signals.

Suppose we have a four-channel orientation signal represented as a quaternion $\vec{q}_{n,t}$, a new orientation $\hat{\vec{q}}$ can be generated by rotating $\vec{q}_{n,t}$ with

$$\hat{\vec{q}} = \vec{r}_a^{\theta} \otimes \vec{q}_{n,t} \tag{12}$$

where $\vec{r}_a^{\theta}$ represents a rotation quaternion of an angle θ about an axis $\vec{a} = [a^x, a^y, a^z]$. It can be calculated by

$$\vec{r}_a^{\theta} = [a^x \sin(\theta/2), a^y \sin(\theta/2), a^z \sin(\theta/2), \cos(\theta/2)]. \tag{13}$$

Applying mirroring to the original data is to add variations in some situations, for example, the armband is worn in different dominant arms for different subjects. First, the vector $\vec{v}^{mirror}$ mirrored from the current direction vector $\vec{v}_{n,t}$ (Eq. 5) against a certain plane can be calculated with

$$\vec{v}^{mirror} = \vec{v}_{n,t} - 2\vec{v}_{n,t} \cdot \vec{n}^T \cdot \vec{n} \tag{14}$$

where $\vec{n}$ is the normal vector of the given plane.

Then the mirrored quaternion $\bar{\vec{q}} = [\bar{q}^x, \bar{q}^y, \bar{q}^z, \bar{q}^w]$, representing the transition between the two vectors v_{ref} and v_{mirror}, can be obtained by

$$\begin{aligned} [\bar{q}^x, \bar{q}^y, \bar{q}^z] &= \vec{v_{ref}} \times \vec{v^{mirror}} \\ \bar{q}^w &= 1 + \vec{v_{ref}} \cdot \vec{v^{mirror}} \end{aligned} \tag{15}$$

where $\times$ and $\cdot$ are the cross and dot products, respectively.

For the other six channels of linear acceleration and angular velocity, since their measurements are relative to the sensor's coordinate systems, rotation and mirror operation do not affect the values. Some random noises (uniformly distributed in the range of ±5% of the original signals) are added to simulate the possible fluctuations.

4. MULTI-MODAL RECOGNITION

In this section the developed multi-modal approach for worker activity recognition is detailed: four deep learning architectures created for different input modalities are presented; the cost function for training each modality is introduced; and the inference fusion strategies to output the recognition result are described.

4.1. DEEP LEARNING ARCHITECTURES OF FOUR INPUT MODALITIES

After the preprocessing, signal representation generation and data augmentation described in Section 3, there are N[1] data samples $\{X_1, \cdots, X_N\}$, each of which contains four different inputs:

$$X_n = \{I_n^{freq}, I_n^{och}, I_n^{frame}, V_n^{clip}\}, n \in [1, N] \tag{16}$$

where $I_n^{freq}, I_n^{och}, I_n^{frame}$ and V_n^{clip} are the four inputs of frequency feature transform, spatial orientation changing history (och) feature transform, frame-level visual representation and video-level visual representation, respectively.

For the three image inputs, I_n^{freq}, I_n^{och} and I_n^{frame}, 2D convolutional operation [9] is applied to extract features layer by layer. The value at position (x, y) in the jth feature map of the ith layer is computed by

$$v_{ij}^{xy} = g\left(b_{ij} + \sum_k \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} w_{ijk}^{pq} v_{(i-1)k}^{(x+p)(y+q)}\right) \tag{17}$$

[1]Here we use the same notation for simplicity but this N is larger than the one in Eq. 1 due to the data augmentation.

where $g(\cdot)$ denotes a non-linear activation function. b_{ij} is the bias for this feature map, k is the index of the feature maps in layer $(i-1)$, w_{ijk}^{pq} is the value at the position (p,q) of the kernel connected to the kth feature map, and P_i and Q_i are the height and width of the two-dimensional kernel, respectively.

For video-clip input V_n^{clip}, 3D convolutional operation [9] is applied to deal with the additional temporal dimension. The value at position (x,y,z) in the jth feature map of the ith layer is given by

$$v_{ij}^{xyz} = g\left(b_{ij} + \sum_{k} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijk}^{pqr} v_{(i-1)k}^{(x+p)(y+q)(z+r)}\right) \tag{18}$$

where R_i is the size of the 3D kernel along the temporal dimension, w_{ijk}^{pqr} is the (p,q,r)th value of the kernel connected to the kth feature map in the previous layer.

The feature maps obtained from a series of convolutional operations are flattened as a feature vector. To solve the classification problem, the vector is further input to a multi-layer neural network. The value of the jth neuron in the ith fully connected layer, denoted as v_{ij}, is given by

$$v_{ij} = g\left(b_{ij} + \sum_{k=0}^{K_{(i-1)}-1} w_{ijk} v_{(i-1)k}\right), \tag{19}$$

where b_{ij} is the bias term, k indexes the set of neurons in the $(i-1)$th layer connected to the current feature vector, w_{ijk} is the weight value in the ith layer connecting the jth neuron to the kth neuron in the previous layer.

In details, the proposed CNN models for the four input modalities are described as follows:

I_n^{freq}: The architecture of our CNN model for I_n^{freq} is illustrated in Figure 5. It accepts the frequency image as the input, and outputs a probability distribution of the 6 activities. I_n^{freq} has the size of $42 \times 32 \times 1$ (height, width, depth, respectively) and is

normalized to the interval $[0,1]$ before being fed into two 5×5 convolutional layers for feature extraction. Each convolutional layer is down-sampled to a half by implementing a 2×2 max pooling layer. The classification module accepts the $10 \times 8 \times 64$ feature map from the last pooling layer and flattens it as a 5120 feature vector. Then, two fully connected layers are used to densify the feature vector to the dimensions of 128 and C sequentially, where C is the number of worker activity classes. Finally, this C-dimensional score vector $S([S_1, ..., S_c, ..., S_C])$ is transformed to output the predicted probabilities with a softmax function as follows:

$$P(y_n = c|X_n) = \frac{\exp(S_c)}{\sum_{c=1}^{C} \exp(S_c)} \tag{20}$$

where $P(y_n = c|X_n)$ is the predicted probability of being class c for sample X_n.

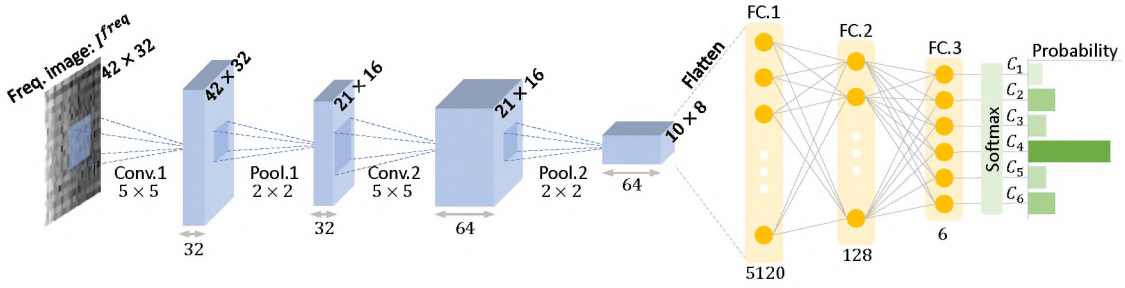


Figure 7. The architecture of our CNN model for I_n^{freq}.

I_n^{och} and I_n^{frame} : For these two input modalities, we use transfer learning to solve the image classification problem instead of building and training CNN models from scratch. To extract image features, we use a VGG network [25] pretrained on the ImageNet dataset [5]. For each image input, the feature vector obtained from the fully connected layer FC7 in the VGG model is used to represent the image, then a new classifier is designed on top of it to output the prediction on activity class.

V_n^{clip}: The video-clip input V_n^{clip} contains spatial-temporal information. We use the C3D model pretrained on the Sports-1M dataset [10, 12, 33]. The C3D network reads sequential frames and outputs a fixed-length feature vector every 16 frames. We extract activation vectors from the fully connected layer FC6-1, which is then connected to a new classifier to predict the worker activity class.

4.2. TRAINING

The process of training a CNN model involves optimization of the network's parameters w to minimize the cost function for the training dataset X. We select the commonly used regularized cross entropy [6] as the cost function, which is

$$\mathcal{L}(w) = \sum_{n=1}^{N} \sum_{c=1}^{C} y_{nc} \log[P(y_n = c|X_n)] + \lambda l_2(w) \tag{21}$$

where y_{nc} is 0 if the ground truth label of X_n is the cth label, and is 1 otherwise. The l_2 regularization term is appended to the loss function for penalizing large weights, and λ is its coefficient. The dropout regularization [26] randomly drops units from the neural network during training, which is commonly used to avoid the overfitting. It is implemented during our training as well.

4.3. INFERENCE FUSION

Just like how human uses five senses to perceive the world, multi-modal approach has the opportunity to integrate all the information and make a comprehensive understanding of the learning problem. Mathematically, each individual model can return a probability distribution on the worker activity prediction, we can design different strategies to fuse the inferences from different models:

4.3.1. Maximum Fusion. This method reports the maximum output within a list of predictions.

$$S_c^{max} = \max_{m \in \{1,2,\ldots,M\}} p_c^m \tag{22}$$

where m is the index of different models and M is the total number of models.

4.3.2. Average Fusion. In this method, we adopt the average to fuse the outputs of different modalities, i.e.,

$$S_c^{avg} = \frac{1}{M} \sum_{m=1}^{M} p_c^m \tag{23}$$

4.3.3. Weighted Fusion. We introduce the informativity value γ^m to evaluate the prediction confidence of each modality m. γ^m is calculated with Eq. 24, which is modified from the Shannon entropy of a discrete probability distribution to vary in the interval of $[0, 1]$.

$$\gamma^m = \frac{\sum_{k=1}^{K} p_k^m \log p_k^m}{\log K} + 1 \tag{24}$$

where m is the index of modalities and k is the index of top-K candidates. p_k^m represents the probability of the kth class candidate at the mth model. I^m will be close to 0 if all the top-K candidates have similar probabilities (i.e., $p_k^m \approx 1/K$), and 1 if the probability of top-1 class candidate is about reaching 1 (i.e., $p_1^m \approx 1$).

Then every predicted probability p_k^m of the mth model is weighted by γ^m of this model and the weighted maximum fusion and the weighted average fusion scores are

$$S_c^{max-w} = \max_{m \in \{1,2,\ldots,M\}} \gamma^m p_c^m \tag{25}$$

$$S_c^{avg-w} = \frac{1}{M} \sum_{m=1}^{M} \gamma^m p_c^m \tag{26}$$

For the above four fusion strategies, the final predicted label is chosen as the one that maximizes the fusion score (e.g., for weighted average fusion, $c^* = \arg\max_c S_c^{avg-w}$).

5. EXPERIMENTS AND EVALUATION METRICS

5.1. IMPLEMENTATION DETAILS

The CNN architectures of the four input modalities described in the previous sections are constructed using TensorFlow [1] and Keras libraries. They are trained individually so that each of them can make its own inference for further decision fusion. The SGD optimizer is used in training, with the momentum of 0.9, the learning rate of 0.001 and the regularizer coefficient of 1e-5. The batch size for each of the four models is 512, 64, 64 and 512, respectively, which is limited by the computation memory. The number of training epochs is 1000 and 100 for the first modality I_n^{freq} and the other modalities, respectively. We use a workstation with one 12-core Intel Xeon processor, 64GB of RAM and two Nvidia Geforce 1080 Ti graphic cards for the training jobs. It takes approximately 30 minutes to train each model for a leave-one-out experiment.

5.2. EVALUATION METRIC

Two evaluation policies are conducted, i.e., half-half and leave-one-out policies. In the half-half evaluation, after randomly shuffling, one half of the dataset is used for training and the other half is kept for testing. In the leave-one-out evaluation, the samples from 7 out of 8 subjects are used for training, and the samples of the left one subject are reserved for testing. We employ several commonly used metrics [6] to evaluate the classification performance, which are listed as follows:

1. Accuracy

$$Accuracy = \frac{\sum_n^N 1(\hat{y}_n = y_n)}{N} \tag{27}$$

2. Precision and Recall

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \end{aligned} \tag{28}$$

3. F_1 score

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{29}$$

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

6. RESULTS AND DISCUSSION

In this section, we first perform evaluations of the data augmentation methods. Then, we compare the performance of different fusion methods. After that, we explore various modalities and their combinations for an ablation study. The performance of our approach on some public dataset is also reported. Then, we conduct visualizations for a better understanding of the CNN model. Finally, future research needs are discussed.

6.1. EVALUATION OF THE DATA AUGMENTATION METHODS

To evaluate the effectiveness of our proposed kinematics-based augmentation (KA) method, we compare it to the jittering augmentation (JA) method [24], which has been proved to be an effective method and is commonly used in CNN-based image classification tasks.

For the KA method, four rotation angles $\{\pm\pi/8, \pm\pi/4\}$ are selected for rotation augmentation, i.e., new samples are generated by implementing rotation on the original signal samples, and two mirroring planes yz-plane and xz-plane are chosen for mirroring augmentation, overall yielding 6 augmented samples for each actual sample. Then the amount of the augmented training dataset is 6 times more than the original one. Note that the augmentation is applied directly on each original signal sample s_n before the feature transforms.

As for the JA method, to have a fair comparison with the KA method, 6 augmented samples are generated by randomly translating in the range of $\pm 10\%$ of the image width/height, scaling in the range of $[0.9, 1.1]$ ratio, and rotating in the range of $[-5, +5]$ degrees. In the JA method, the augmentation is applied to each image I_n^{freq} and I_n^{och} after the feature transforms.

We also evaluate the performance of the JA+KA method, in which the augmented data from the JA and KA methods are integrated. The leave-one-out evaluations of the two modalities I_n^{freq} and I_n^{och} on our activity dataset with the different augmentation methods are shown in Table 3 (the half-half accuracies are not considered for the comparison purpose because they are about reaching 100%). All the three augmentation methods have accuracy improvements compared with the models without using data augmentation. For I_n^{freq}, the JA method improves the accuracy from 88.0% to 88.7%, and the KA method outperforms the JA method, whose accuracy is 90.2%. By combining the JA and KA methods, the accuracy is slightly further improved to 90.5%. For I_n^{och}, the accuracy is improved from 63.6% to 65.0% with the JA method, and is further improved by using the KA method, which is 77.3%, 12 percentage points higher than the JA method. However, the JA+KA method does not further improve the accuracy and its accuracy 75.3% is lower than the KA method.

Table 3. Comparison (%) of accuracy regarding to different data augmentation methods.

Modalities	Data Augmentation Methods			
	None	JA	KA	JA+KA
I_n^{freq}	88.01	88.71	90.18	**90.51**
I_n^{och}	63.59	65.01	**77.27**	75.33

Overall, the data augmentation techniques, JA and KA, demonstrate the effectiveness in improving the model performance, because the augmentation process introduces more variations to the training dataset to simulate the potential variations in the unseen samples, which pushes the deep learning model to learn the most discriminative features and makes the training more robust. Meanwhile, the KA method outperforms the JA method. It is because rather than introducing variations to the image, like what JA method does, KA method directly generates some physically-realistic variations to the original signal sample, which is more effective to augment the dataset to be more comprehensive. Although JA+KA method improves the performance of I_n^{freq} slightly compared with KA method, it does not for I_n^{och}. Because JA+KA method has a larger amount (i.e., 2 times) of training data and I_n^{och} has a more complex architecture than I_n^{freq}, which makes the training less efficient, we choose KA method for both of the modalities in the following study as a compromise between performance and training efficiency.

6.2. EVALUATION OF DIFFERENT FUSION METHODS

Each of the four input modalities generates a vector output before the fusion step. Then, these vector outputs are fused to have only one score vector as the final output. To study the effect of the four different fusion methods: 1) maximum fusion, 2) average fusion, 3) weighted maximum fusion and 4) weighted average fusion, a set of experiments are conducted on our activity dataset.

Table 4. Comparison (%) of different fusion methods for the leave-one-out experiments.

Fusion Methods	Accuracy	Precision	Recall	F Score
Maximum	93.68	92.48	92.50	91.09
Average	**97.17**	**97.04**	**96.82**	**96.81**
Weighted Max.	93.68	92.45	92.49	91.07
Weighted Avg.	96.79	96.38	96.28	96.04

The comparisons of the fusion performance, in terms of accuracy, precision, recall and F score, are listed in Table 4. The average fusion method performs better than the maximum fusion method for all the metric items. The weighted maximum method has the same accuracy as the maximum method but lower precision, recall and F score. The weighted average method has lower performance than the average method. The two weighted methods do not contribute additional improvement as they did in [31]. Therefore, the average method is chosen as the fusion strategy for our following experiments.

6.3. EVALUATION OF DIFFERENT INPUT MODALITIES

A central idea of our approach is that the reasoning based on multiple modalities can significantly improve the inference performance based on single modality. To validate this idea, we perform a comprehensive ablation study where we progressively increase the number of modalities and try different modality combinations. The performance of these cases in terms of accuracy, precision, recall and F score with two evaluation policies (half-half and leave-one-out) is summarized in Table **??**.

To simplify the abbreviation, we use $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$ and $\mathcal{M}_4$ to represent the four input modalities, I_n^{freq}, I_n^{och}, I_n^{frame} and V_n^{clip}, respectively. For the single-modal cases, although $\mathcal{M}_1$ and $\mathcal{M}_2$ are both based on the IMU signals, $\mathcal{M}_2$ shows lower performance because it only uses the 4 orientation channels out of the 10 channels. Also, it demonstrates that the frequency feature transform provides more discriminative features for activity recognition.

$\mathcal{M}_3$ performs better than $\mathcal{M}_4$, which shows that the current pretrained VGG model can extract more discriminative features than the C3D model. Overall, $\mathcal{M}_1$ achieves the highest performance in the single-modal cases, whose metric items are accuracy (90.2%), precision (90.7%), recall (89.5%) and F score (87.6%), respectively.

For the dual-modal cases, all the 6 combinations are evaluated. All the cases have better results compared with their related single-modal cases, e.g., $\mathcal{M}_{\{2,3\}}$ performs better than both $\mathcal{M}_2$ and $\mathcal{M}_3$. $\mathcal{M}_{\{1,3\}}$ has the highest accuracy as their individual modalities are also the highest two for the single-modal cases.

For the triple-modal cases, 4 combinations are tested. The fusion of more modalities further improve the performance than the duel-modal cases. $\mathcal{M}_{\{1,3,4\}}$ has the highest accuracy as their individual modalities are also the highest three for the single-modal cases.

Finally, a quad-modal case $\mathcal{M}_{\{1,2,3,4\}}$ including all the four modalities is experimented, which achieves the highest performance. Therefore, we choose the quad-modal architecture for our model.

Table 5. Overall performance (%) of the half-half (hh) and leave-one-out (loo) experiments.

Methods	Accuracy		Precision		Recall		F Score	
	hh	loo	hh	loo	hh	loo	hh	loo
Previous [30]	97.6	87.4	97.8	89.0	97.5	89.5	97.7	87.6
$\mathcal{M}_1$	99.5	90.2	99.5	90.7	99.6	90.9	99.5	90.3
$\mathcal{M}_2$	93.0	77.3	92.3	77.5	92.5	78.3	92.4	75.0
$\mathcal{M}_3$	100	86.8	100	83.0	100	83.2	100	81.3
$\mathcal{M}_4$	100	80.8	100	79.1	100	77.7	100	74.3
$\mathcal{M}_{\{1,2\}}$	99.6	91.1	99.6	91.5	99.6	92.1	99.6	91.4
$\mathcal{M}_{\{1,3\}}$	100	94.8	100	94.9	100	94.6	100	94.3
$\mathcal{M}_{\{1,4\}}$	100	92.2	100	93.1	100	91.3	100	90.1
$\mathcal{M}_{\{2,3\}}$	100	90.3	100	90.9	100	87.8	100	87.0
$\mathcal{M}_{\{2,4\}}$	100	85.0	100	84.0	100	82.8	100	80.2
$\mathcal{M}_{\{3,4\}}$	100	89.5	100	86.3	100	86.0	100	84.2
$\mathcal{M}_{\{1,2,3\}}$	100	95.3	100	95.4	100	95.4	100	95.2
$\mathcal{M}_{\{1,2,4\}}$	100	93.9	100	93.5	100	94.1	100	93.0
$\mathcal{M}_{\{1,3,4\}}$	100	95.9	100	95.2	100	94.8	100	94.2
$\mathcal{M}_{\{2,3,4\}}$	100	92.6	100	90.4	100	90.7	100	89.5
$\mathcal{M}_{\{1,2,3,4\}}$	**100**	**97.2**	**100**	**97.0**	**100**	**96.8**	**100**	**96.8**

For the half-half experiments, almost all of the testing samples are correctly recognized. It is higher than the leave-one-out experiments. This is because all the testing subjects are seen in the half-half experiment, while the testing subject in the leave-one-out experiment is unseen.

6.4. PERFORMANCE COMPARISON ON THE PUBLIC DATASET

To validate the generalization of our method, a commonly-used public dataset for human activity recognition, PAMAP2 dataset [22], is also chosen for comparison. This dataset has 12 human activities (lying, sitting, standing, walking, running, cycling, Nordic walking, ascending stairs, descending stairs, vacuum cleaning, ironing and rope jumping) captured by three IMU sensors (worn on the wrist, chest and ankle, respectively), and the activities are performed by 9 different subjects. Since the PAMAP2 dataset does not include video recordings, we evaluate the performance of our CNN models of the first two modalities on it. The performance comparison of several existing deep learning models on the PAMAP2 dataset is listed in Table 6. Using the same evaluation protocol, our model achieves the best recognition accuracy, 94.2%, compared with other methods in the literature.

Table 6. Performance (%) comparison of existing deep models on the PAMAP2 activity dataset.

Method	Accuracy
Hammerla et al. (2016) [7]	93.70
Murahari et al. (2018) [18]	87.50
Zeng et al. (2018) [37]	89.96
Xi et al. (2018) [35]	93.50
Xu et al. (2019) [36]	93.50
Our model	**94.16**

6.5. VISUALIZING THE CLASS ACTIVATION MAP OF $\mathcal{M}_3$

Although the CNN model demonstrates superior performance on various applications, such as the image classification task, it is usually taken as a black box because of its high architectural complexity and tremendous network parameters, and its hyperparameters are tuned by prior experiences or trial-and-error. To have a better understanding of which parts of a given image lead a CNN to its final classification decision, we visualize the class activation map (CAM), which consists of producing heatmaps of class activation over input images.

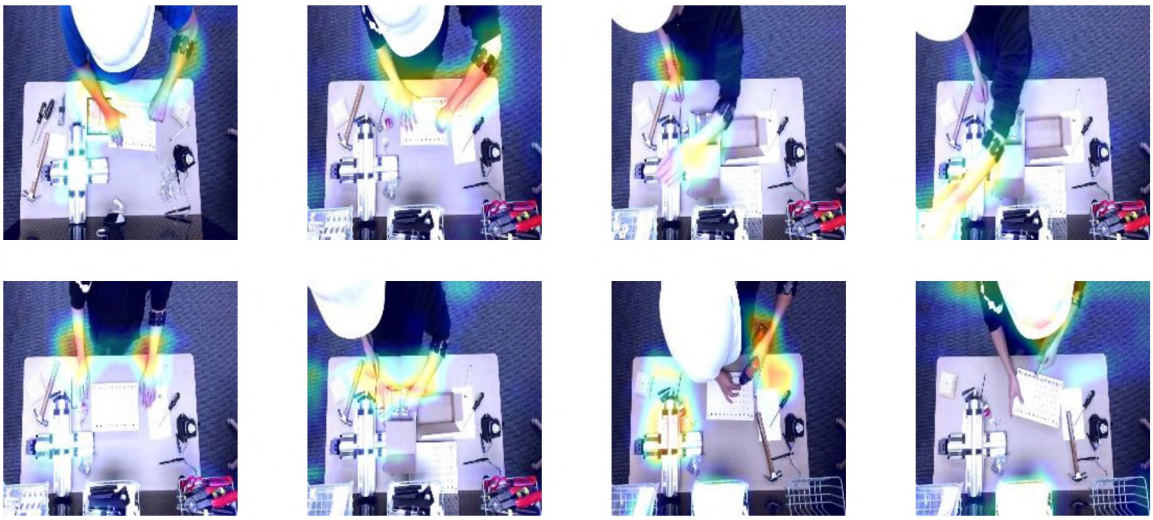


Figure 8. Examples of Class Activation Map (CAM) Visualization.

A class activation heatmap is a 2D grid of scores associated with a specific output class, computed for every location in any input image, indicating how important each location is with respect to the class under consideration. A set of CAM examples are shown in Figure 12, where the generated heapmaps are overlaid onto the input images. We can see that the model is able to focus on the hand and tool regions, where exactly the interaction happens.

6.6. FUTURE RESEARCH NEEDS

At present, we conduct the multi-modal recognition of 6 basic activities. To further push the current approach to the practical application, some directions for future work are considered, such as recruiting more subjects to learn more working styles, optimizing data augmentation techniques to add more variations to the collected data, and exploring different methods of signal preprocessing and feature extraction to fully exploit the recorded signals. In addition, more fusion methods can be explored and every modality can be further improved to reach their optimal performance.

7. CONCLUSION

Worker behavior awareness is crucial towards human-centered intelligent manufacturing. In this paper, we proposed a multi-modal approach for worker activity recognition. Two sensors (wearable device and camera) were adopted to perceive the worker, and four modalities were built to recognize the activity independently. Then, inference fusion was implemented to achieve an optimal understanding of the worker's behavior.

We designed two novel mechanisms to produce image representations of the IMU sensor signals in both the frequency and spatial domains. A kinematics-based data augmentation method was developed to generate more physically-realistic variations in the training dataset. This performs better than the traditional data augmentation method. A worker activity dataset has been established, which currently involves 8 subjects and contains 6 common activities in assembly tasks (i.e., grab a tool/part, hammer a nail, use a power-screwdriver, rest arms, turn a screwdriver and use a wrench). The multi-modal approach is evaluated on the dataset and achieves 100% and 97% recognition accuracy in the half-half and leave-one-out experiments, respectively. Our approach can be further generalized to other sensors, modalities, and working contexts.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., 'TensorFlow: Large-scale machine learning on heterogeneous systems,' 2015, software available from tensorflow.org.

[2] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L., 'A public domain dataset for human activity recognition using smartphones.' in 'ESANN,' 2013 .

[3] Carreira, J. and Zisserman, A., 'Quo vadis, action recognition? a new model and the kinetics dataset,' in 'Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on,' IEEE, 2017 pp. 4724–4733.

[4] Chang, W., Dai, L., Sheng, S., Tan, J. T. C., Zhu, C., and Duan, F., 'A hierarchical hand motions recognition method based on imu and semg sensors,' in 'Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on,' IEEE, 2015 pp. 1024–1029.

[5] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., 'ImageNet: A Large-Scale Hierarchical Image Database,' in 'CVPR09,' 2009 .

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

[7] Hammerla, N. Y., Halloran, S., and Plötz, T., 'Deep, convolutional, and recurrent models for human activity recognition using wearables,' arXiv preprint arXiv:1604.08880, 2016.

[8] Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., and Eschert, T., 'Industrial internet of things and cyber manufacturing systems,' in 'Industrial Internet of Things,' pp. 3–19, Springer, 2017.

[9] Ji, S., Xu, W., Yang, M., and Yu, K., '3d convolutional neural networks for human action recognition,' IEEE transactions on pattern analysis and machine intelligence, 2013, **35**(1), pp. 221–231.

[10] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T., 'Caffe: Convolutional architecture for fast feature embedding,' in 'Proceedings of the 22nd ACM international conference on Multimedia,' ACM, 2014 pp. 675–678.

[11] Jiang, W. and Yin, Z., 'Human activity recognition using wearable sensors by deep convolutional neural networks,' in 'Proceedings of the 23rd ACM international conference on Multimedia,' ACM, 2015 pp. 1307–1310.

[12] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L., 'Large-scale video classification with convolutional neural networks,' in 'Proceedings of the IEEE conference on Computer Vision and Pattern Recognition,' 2014 pp. 1725–1732.

[13] Kober, J., Bagnell, J. A., and Peters, J., 'Reinforcement learning in robotics: A survey,' The International Journal of Robotics Research, 2013, **32**(11), pp. 1238–1274.

[14] Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P., and Roning, J., 'Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines,' in 'Control and Automation, 2009. MED'09. 17th Mediterranean Conference on,' IEEE, 2009 pp. 401–405.

[15] LeCun, Y., Bengio, Y., and Hinton, G., 'Deep learning,' Nature, 2015, **521**(7553), pp. 436–444.

[16] Lee, J., Ardakani, H. D., Yang, S., and Bagheri, B., 'Industrial big data analytics and cyber-physical systems for future maintenance & service innovation,' Procedia CIRP, 2015, **38**, pp. 3–7.

[17] Maekawa, T., Nakai, D., Ohara, K., and Namioka, Y., 'Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory,' in 'Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing,' ACM, 2016 pp. 1088–1099.

[18] Murahari, V. S. and Plötz, T., 'On attention models for human activity recognition,' in 'Proceedings of the 2018 ACM International Symposium on Wearable Computers,' ACM, 2018 pp. 100–103.

[19] Nagorny, K., Lima-Monteiro, P., Barata, J., and Colombo, A. W., 'Big data analysis in smart manufacturing: A review,' International Journal of Communications, Network and System Sciences, 2017, **10**(03), p. 31.

[20] Peterek, T., Penhaker, M., Gajdoš, P., and Dohnálek, P., 'Comparison of classification algorithms for physical activity recognition,' in 'Innovations in Bio-inspired Computing and Applications,' pp. 123–131, Springer, 2014.

[21] Petruck, H. and Mertens, A., 'Using convolutional neural networks for assembly activity recognition in robot assisted manual production,' in 'International Conference on Human-Computer Interaction,' Springer, 2018 pp. 381–397.

[22] Reiss, A. and Stricker, D., 'Introducing a new benchmarked dataset for activity monitoring,' in '2012 16th International Symposium on Wearable Computers,' IEEE, 2012 pp. 108–109.

[23] Ronao, C. A. and Cho, S.-B., 'Human activity recognition using smartphone sensors with two-stage continuous hidden markov models,' in 'Natural Computation (ICNC), 2014 10th International Conference on,' IEEE, 2014 pp. 681–686.

[24] Sermanet, P. and LeCun, Y., 'Traffic sign recognition with multi-scale convolutional networks,' in 'Neural Networks (IJCNN), The 2011 International Joint Conference on,' IEEE, 2011 pp. 2809–2813.

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

[26] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., 'Dropout: a simple way to prevent neural networks from overfitting.' Journal of machine learning research, 2014, **15**(1), pp. 1929–1958.

[27] Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., and Troster, G., 'Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario,' in 'Wearable Computers, 2006 10th IEEE International Symposium on,' IEEE, 2006 pp. 97–104.

[28] Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., and Tröster, G., 'Wearable activity tracking in car manufacturing,' IEEE Pervasive Computing, 2008, **7**(2).

[29] Stiefmeier, T., Roggen, D., and Troster, G., 'Fusion of string-matched templates for continuous activity recognition,' in 'Wearable Computers, 2007 11th IEEE International Symposium on,' IEEE, 2007 pp. 41–44.

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

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

[32] Thalmic Labs Inc., 'Myo armband,' 2017, [Online; accessed 15-November-2017].

[33] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M., 'Learning spatiotemporal features with 3d convolutional networks,' in 'Proceedings of the IEEE international conference on computer vision,' 2015 pp. 4489–4497.

[34] Wang, P., Liu, H., Wang, L., and Gao, R. X., 'Deep learning-based human motion recognition for predictive context-aware human-robot collaboration,' CIRP Annals, 2018.

[35] Xi, R., Li, M., Hou, M., Fu, M., Qu, H., Liu, D., and Haruna, C. R., 'Deep dilation on multimodality time series for human activity recognition,' IEEE Access, 2018, **6**, pp. 53381–53396.

[36] Xu, C., Chai, D., He, J., Zhang, X., and Duan, S., 'Innohar: A deep neural network for complex human activity recognition,' IEEE Access, 2019, **7**, pp. 9893–9902.

[37] Zeng, M., Gao, H., Yu, T., Mengshoel, O. J., Langseth, H., Lane, I., and Liu, X., 'Understanding and improving recurrent networks for human activity recognition by continuous attention,' in 'Proceedings of the 2018 ACM International Symposium on Wearable Computers,' ACM, 2018 pp. 56–63.

IV. ATTENTION-BASED SENSOR FUSION FOR HUMAN ACTIVITY RECOGNITION USING IMU SIGNALS

Wenjin Tao[a], Md Moniruzzaman[b], Ming C. Leu[a], Zhaozheng Yin[b], Ruwen Qin[a]

[a]Missouri University of Science and Technology, Rolla, MO 65409, USA

[b]Stony Brook University, Stony Brook, NY 11794, USA

ABSTRACT

Human Activity Recognition (HAR) using wearable devices such as smart watches embedded with Inertial Measurement Unit (IMU) sensors has various applications relevant to our daily life, such as workout tracking and health monitoring. In this paper, we propose a novel attention-based approach to human activity recognition using multiple IMU sensors worn at different body locations. Firstly, a sensor-wise feature extraction module is designed to extract the most discriminative features from individual sensors with Convolutional Neural Networks (CNNs). Secondly, an attention-based fusion mechanism is developed to learn the importance of sensors at different body locations and to generate an attentive feature representation. Finally, an inter-sensor feature extraction module is applied to learn the inter-sensor correlations, which are connected to a classifier to output the predicted classes of activities. The proposed approach is evaluated using five public datasets and it outperforms state-of-the-art methods on a wide variety of activity categories.

Keywords: Attention Mechanism; Activity Recognition; Neural Networks; Sensor Fusion; Wearable Computing.

1. INTRODUCTION

Human Activity Recognition (HAR) aims to automatically recognize various human activities, such as daily life and sport activities, with algorithms using the input of a series of sensor measurements. It has a wide range of applications, such as human-computer interaction, robot learning, ubiquitous computing, workout tracking, and health monitoring [6, 23, 24, 34]. Although HAR is not a new emerging topic and has been studied for decades, it is still an active area of research now because of remaining challenges, such as the high complexity of human activities, the large variations among different subjects, and the balance between the algorithm complexity and the energy efficiency.

Various sensors have been used for HAR. Considering the wearability, they can be categorized as ambient sensors and wearable sensors. Ambient sensors are deployed in the environment to sense the subject in a passive manner. For example, optic cameras can be used to capture RGB images on human subjects; Depth cameras such as a Microsoft Kinect or Lidar (light detection and ranging) sensors can be applied to sense human objects in the 3D space; Infrared cameras can detect the subject in a dark environment; Pressure sensing mats can be used to capture human's standing states; WiFi signals also have been used for HAR [19]. Ambient sensing can collect a large amount of data without interfering the subject's activity.

Nevertheless, ambient sensors require complex setups and their performance can be affected dramatically by occlusion issues, which are the main challenges in implementing ambient sensing. Also, it becomes more difficult when capturing a subject's outdoor activities. To compensate for these limitations, wearable sensing can be applied. Wearable sensor based activity recognition has captured growing attention nowadays because of the pervasiveness of mobile devices (e.g., smart phones and smart watches), which are embedded with various sensors such as IMU (Inertial Measurement Unit) sensors, heart rate sensors, and ECG (Electrocardiogram) sensors. IMU sensors are the most used for

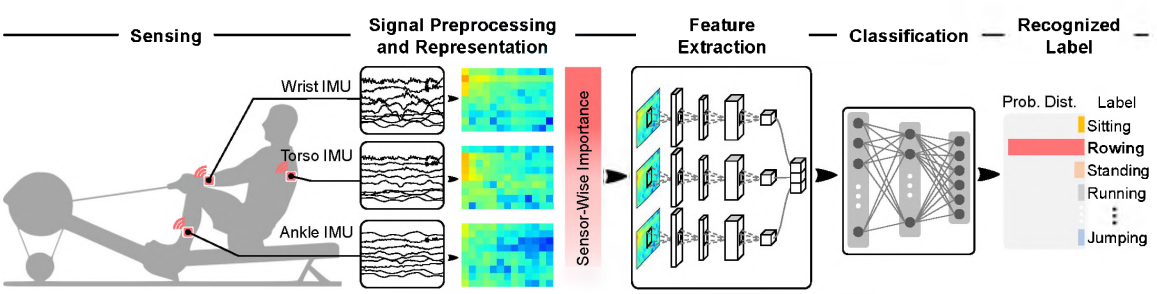


Figure 1. Overview of the human activity recognition pipeline using IMU signals.

HAR as the sensor directly measure the movements of human body. Usually, an IMU has multiple sensors in different modalities, such as an accelerometer, a gyroscope, and a magnetometer, to measure the acceleration, angular rate, and magnetic field, respectively.

In this paper, we focus on accurately recognizing human's physical activities with multiple IMU sensors considering that IMU signals from different locations could augment the perception of human activities. The pipeline of human activity recognition is illustrated in Figure 1. IMU sensors are worn at different body locations to sense the activity, from which a series of signals are captured and preprocessed to have formatted representations. After that, a feature extraction process is implemented to extract high-level features. Then, the extracted features are fed into a classifier to generate a probability distribution of activity classes. Finally, the activity label can be inferred.

1.1. RELATED WORK

The critical factor attributed to the success of IMU-based activity recognition is to seek an effective representation of the time-series IMU signals. The most widely used approaches fall into two categories: handcrafted feature design and automatic feature learning.

1.1.1. Hand-Crafted Feature Design. It is intuitive to manually pick statistical attributes (e.g., means) or quantity distributions (e.g., magnitude histograms) from IMU signals [16]. For example, Anguita et al. [2] designed as many as 341 features from 3-axis

IMU signals while Hammerla et al. [14] preserved the statistical characteristics of IMU data using their empirical cumulative distributions. Xu el al. [39] proposed a multi-level feature learning framework which consists of the signal-based, components-based and sematic-based information for activity recognition. However, handcrafted feature design is mostly driven by the domain knowledge, prior experience and experimental validation, thus it is possible to neglect some useful features in this manner. In addition, a pre-defined feature extraction mechanism trained on a specific scenario might not work well on other scenarios with different sets of activities to be recognized. That is, those hand-crafted features in the literature might not be transferrable to new application domains, which further makes the feature design time-consuming and labor-costly.

1.1.2. Automatic Feature Learning. The drawbacks of handcrafted features motivate researchers to explore automatic feature learning [20][17]. Deep Convolutional Neural Network (DCNN), as one of the most effective deep learning models, attracts attentions in the mobile sensing domain considering it has achieved the superior performance in other research fields such as computer vision [21] and speech recognition [25]. To improve the accuracy of sensor-based activity recognition, Zeng et al. [43] designed a tri-thread DCNN architecture with the three inputs corresponding to the tri-axis accelerometry data, thus the inputs are one-dimensional time-series signals. To enhance the ability for feature learning, Duffner et al. [9] and Ha et al. [12] took as input the two-dimensional matrix obtained by stacking IMU signals. In order for further accuracy improvement, Ravi et al. [30] combined features learned from the deep model with complementary information from a set of hand-crafted features. In addition, Lane et al. [22] looked into this problem in a practical way and showed the application of deep learning to mobile sensing domain is hardware-efficient and can scale up to a large number of inference classes.

In short, the input to the deep network and the architecture of the deep model itself are two key factors to the success of automatic feature learning. The input is of great significance because a good representation of the IMU signals can make it easier for

automatic learning. In the previous work, IMU signals are directly fed into the DCNN architecture and this simple and raw input may not be a good representation of IMU signals because each value of the raw time-series signals is less meaningful if we do not consider the statisctic property of the whole signals.

In terms of the design of deep architecture, the aforementioned simple input restricts the depth of the deep model, limiting the capability to find discriminative features. For instance, the input in [40] is a small 3×30 matrix and there are only two convolutional layers in the architecture. Additionally, the tri-axis accelerometry signals are convolved with one-dimensional kernels in the deep model independently, thus the correlation among different signals is not taken into enough consideration.

1.1.3. Self-Attention Mechanisms. Just like humans can allocate different amount of attention to different aspects when performing a complex task, self-attention mechanisms can model attentions for deep neural networks and have been widely applied in many deep learning tasks [8]. The self-attention mechanism is proposed in [36] for machine translation tasks, in order to distribute different attention over words in a sentence. From then on, attention mechanisms have been increasingly popular in natural language processing (NLP) and computer vision fields, where multiple sources with different importance are involved. For example, Chen et al. [7] uses spatial and channel-wise attention for image captioning, and He et al. [15] applies attention in both the spatial and temporal domains for HAR from videos.

1.2. OUR PROPOSAL

A single IMU sensor[2] collects data only from a specific body location, which may not perform the robust perception under various circumstances, such as when an activity involves multiple body parts or the movements are not captured from the location the IMU

[2]An inertial measurement unit (IMU) can include multiple sensors, such as accelerometers, gyroscopes and magnetometers, here we treat an IMU as an integrated 'sensor' for simplicity.

is worn. Intuitively, multiple IMU sensors have been used to integrate the perception of individual sensors at different body locations for a better understanding of the overall activity.

Traditional methods treat different IMU sensors equally. Few attempts have been made to take the importance of different sensors into consideration when developing HAR algorithms, which cannot provide the correct 'attention' on IMU sensors for different activities. In the present research, to achieve a better understanding of how different sensors contribute to the recognition tasks, we focus on the automatic importance learning for fusing sensors at different body locations.

An overview of our approach is illustrated in Figure 1. IMU signals are captured from multiple sensors worn at different body locations. Firstly, the signals are preprocessed to generate representations in the frequency domain. Secondly, for a sensor at a certain body location, we design a sensor-wise feature extraction module to extract the most discriminative features of signals from each individual sensor. Thirdly, an attention-based fusion mechanism is developed to learn the importance of sensors at different locations and to generate an attentive feature representation. Finally, an inter-sensor feature extraction module is applied to learn the feature relationships among sensors at different locations, which is connected to a classifier to output the predicted classes of activities. To evaluate our method, five publicly available datasets are chosen which contains a wide variety of activity categories, such as daily activities (sitting, standing, vacuum cleaning, etc.), sports activities (cycling, running, playing basketball etc), and car maintenance activities (opening the hood, etc).

The main contributions of this study are as follows:

1. Overall, we propose an attention-based approach for human activity recognition using Inertial Measurement Unit (IMU) signals. Multiple IMU sensors are used to perceive the activities and the importance of each individual sensor is automatically learned to achieve an optimal understanding of the human's activities.

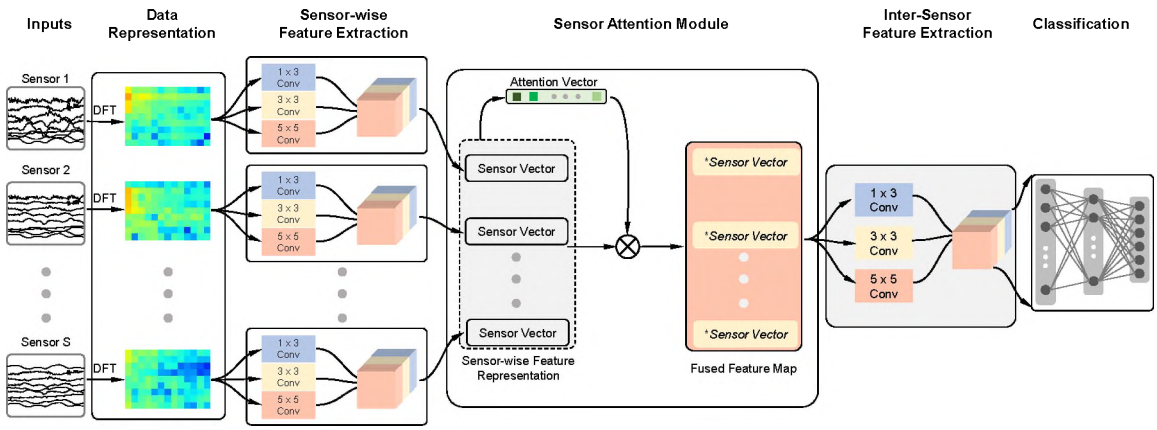


Figure 2. Overview of our attention-based approach for human activity recognition.

2. Regarding to the IMU sensor signal representation, we design a simple yet effective feature transform method to represent the input signals as images in the frequency domain.

3. Regarding to the attention mechanism, we develop a sensor-wise attention module, which enables the network to emphasize features from specific sensors depending on the signals. For fusion purpose, multi-kernel convolutional neural networks are applied to extract the most discriminative sensor-wise and inter-sensor features.

4. Regarding to the experimental validation, our approach outperforms other methods on all of the chosen five public datasets.

The remainder of this paper is organized as follows. Section 2 discusses the details of our proposed approach. Experimental results on five public datasets are described in Section 3, including comparison with the state-of-the-art methods, and the visualization of the results. Finally, Section 4 provides the conclusions of this study.

2. METHODS

In this section, we first present the methods for data preprocessing and representation. Then, each module of our model is explained, including the sensor-wise feature extraction module, sensor attention mechanism, inter-sensor fusion module, and classification module. After that, the training information is detailed.

2.1. SIGNAL PREPROCESSING AND REPRESENTATION

Deep neural networks (DNN) need the input data to be converted as formatted tensors, for example, with a fixed size of $h \times w \times c$ for image inputs where h, w and c are the height, width and the number of channels of the image, respectively. Therefore, some preprocessing steps are necessary before the data can be fed into a DNN. In this section we give a detailed description of the pipeline for data preprocessing and the methods we use for signal representation.

2.1.1. Sampling Procedures. As depicted in Figure 3, the IMU signals from sensors at different body locations are synchronized with the timestamps and denoted as signal sequences. Then, the signal sequences are sampled using a temporal sliding window with the width of T timestamps and Δ_t stride length between two windows. After sampling, we denote our dataset as $\mathbb{D} = \{[D_1, y_1], \cdots, [D_n, y_n], \cdots, [D_N, y_N]\}$ and the nth data is represented as

$$D_n = [d_n^1, d_n^2, \cdots, d_n^s, \cdots, d_n^S], \quad n \in \{1, \cdots, N\} \tag{1}$$

where S is the total number of IMU sensors at different body locations, d_n^s is a sample set of discrete time-series IMU signals from the sth sensor, and y_n is the manually labeled ground truth of the activity class. More specifically, d_n^s a sequence of discrete-time data over T

timestamps, $d_n^s = \{d_{n,1}^s, \cdots, d_{n,t}^s, \cdots, d_{n,T}^s\}$, and each element is elaborated as

$$d_{n,t}^s = [\underbrace{a_{n,t}^x, a_{n,t}^y, a_{n,t}^z}_{a_{n,t}\text{: acceleration}}, \underbrace{g_{n,t}^x, g_{n,t}^y, g_{n,t}^z}_{g_{n,t}\text{: gyro}}, \underbrace{m_{n,t}^x, m_{n,t}^y, m_{n,t}^z}_{m_{n,t}\text{: magnetometer}}, \cdots], \quad t \in \{1, \cdots, T\}, \tag{2}$$

where a, g, and m are sensor readings of linear acceleration, angular velocity, and magnetic field, respectively. In some public datasets, derived information such as gravity-removed linear acceleration and orientation in Euler or quaternion form, is also included.

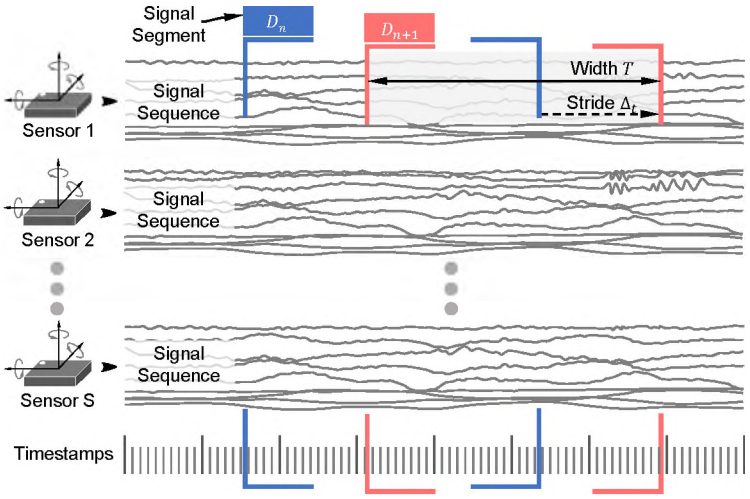


Figure 3. Scheme of the signal sampling method.

2.1.2. Signal Representation. Analyzing signals in the frequency domain is commonly used for signal pattern recognition, because it can extract periodic characteristics which can be more representative than original signals in the time domain. In our study, rather than directly modeling the time-series signals with a DNN, frequency transform is applied as follows: 1) As shown in Figure 4, a signal segment d_n (Figure 4(b), for simple notation, we drop the superscript s that indicates the sth sensor, in the following derivation) is sampled from a signal sequence (Figure 4(a)); 2) A modality-wise normalization is applied to d_n to normalize the signal to the range of $[0, 1]$, generating $\bar{d}_n$ (Figure 4(c)). 3) After normalization, the IMU signal d_n in an IMU segment is represented as an image

I_n with the size of $C \times T$ (Figure 4(d)) where C and T denote the numbers of channels and time frames, respectively, resulting in S image representations for all sensors; 4) One-dimensional Discrete Fourier Transform (DFT) along the time dimension is applied to I_n to get the representation in the frequency domain for analyzing the frequency characteristics. Its logarithmic magnitude is taken to form the image I_n^{DFT}. Due to the conjugate symmetry of Discrete Fourier Transforms

$$I_n^{DFT}(k, c) = I_n^{DFT}(-k, c), \tag{3}$$

where k and c represent the two directions (i.e., frequency and signal channel, respectively) of an image I_n^{DFT}, we can use only a half to represent the DFT image. In the following, we keep using the notation I_n^{DFT} to represent the one-half of DFT image for simplicity (Figure 4(e)).

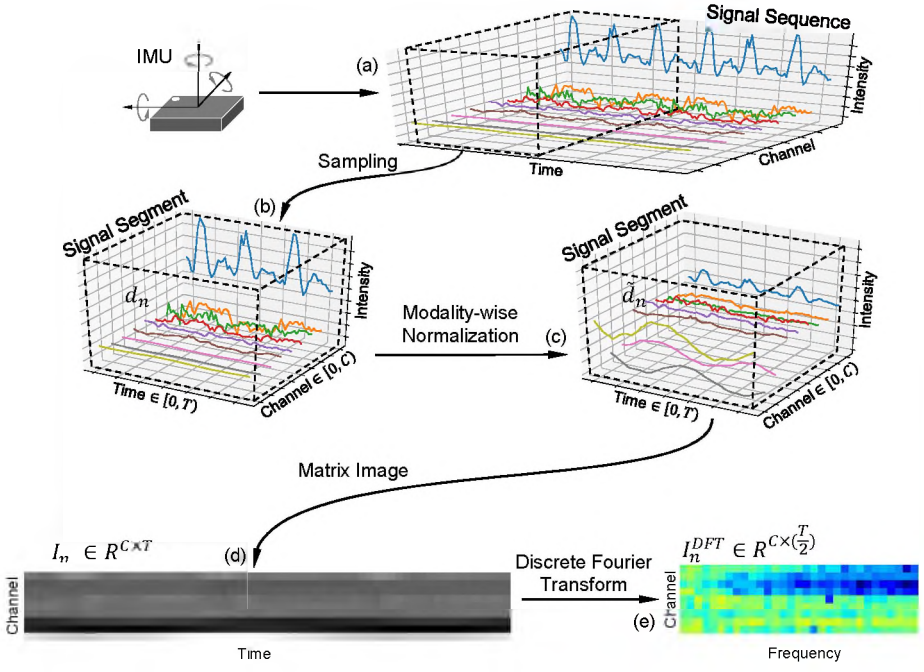


Figure 4. Illustration of the signal representation pipeline for an individual IMU sensor.

Compared with the previous work [20, 35] for signal representation, our method removes the information redundancy, thus reducing the architectural complexity and the number of training parameters for the DNN model.

In total, we have S image representations in the frequency domain for each activity segment. For example, five sensors are included in the Daily dataset [4], i.e., $S = 5$. Figure 5 shows some examples of image representations in the frequency domain, from one subject on 19 activities, from which we can observe the unique patterns of each activity.

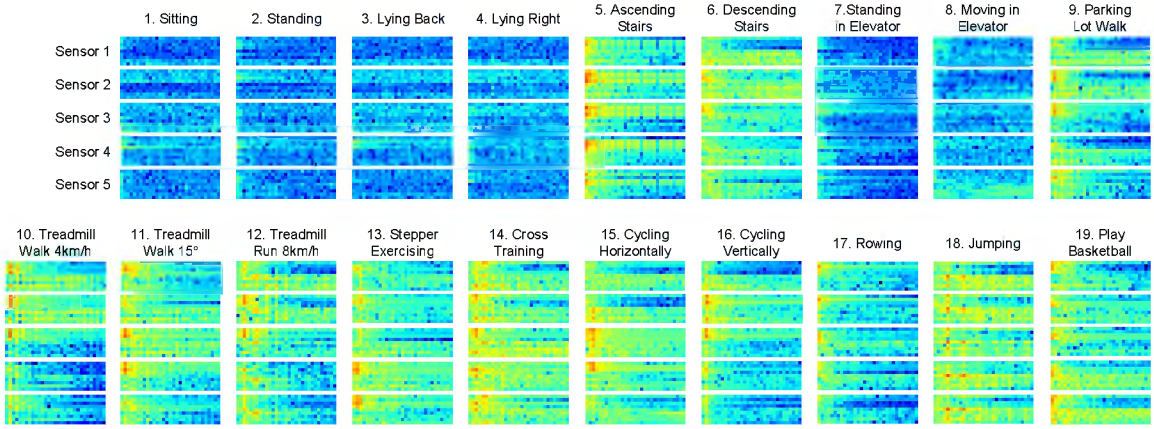


Figure 5. Samples of image representation of different activities from the Daily dataset.

2.2. SENSOR-WISE FEATURE EXTRACTION MODULE

After the above preprocessing step, we have formatted the input ready for DNN. There are N training data samples $\{X_1, \cdots, X_N\}$, each of which contains S sensor inputs:

$$X_n = \{I_n^1, \cdots, I_n^s, \cdots, I_n^S\}, \quad n \in [1, N] \tag{4}$$

For each of the image inputs I_n^s, 2D convolutional operation [10] is applied to extract features layer by layer. The convolutional value using a 2D kernal K at the position (i, j) in the feature map of the lth layer is computed by

$$F^{l}_{i,j} = (F^{l-1} * K)_{i,j} = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} F^{l-1}_{i+p,j+p} K_{p,q} \tag{5}$$

where l is the layer index, $K_{p,q}$ is the value at the position (p,q) of the kernel, and P and Q are the height and width of the two-dimensional kernel K, respectively.

To learn the hidden correlation patterns among multi-channel signals for each individual sensor, we design an intra-sensor feature extraction module. The motivation is to use multiple convolution kernels with various sizes to detect features across different signal channels. As shown in Figure 6, for the input of the sth sensor, 1×3 kernels are used to look at the channel-wise feature, 3×3 kernels are designed to detect the inter-channel features among three channels, and 5×5 kernels are used to discover the inter-channel pattern in a larger perceptive field. In addition, larger size kernels, such as 7×7 and 9×9 can be used to further look into the signals in a larger field.

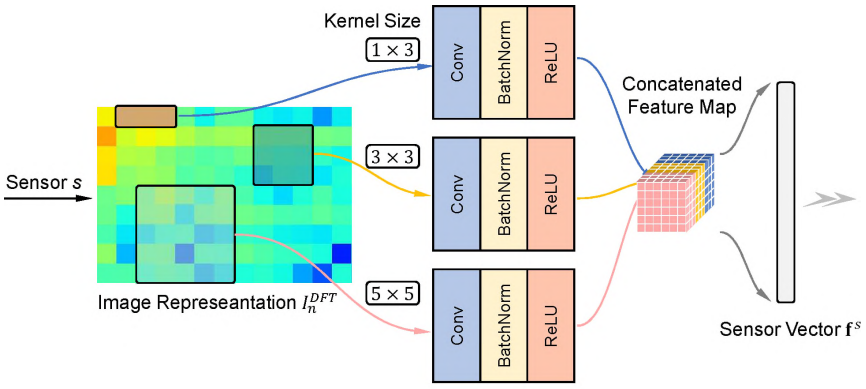


Figure 6. Illustration of the feature extraction module.

After each convolutional layer, a batch normalization layer [18] and an activation layer of ReLU (Rectified Linear Unit) are applied. Then, these extracted feature maps are concatenated to form an information-richer feature set containing features across different signal channels. Finally, the extracted feature maps from each sensor is flattened as a vector representation $\vec{f}^s$, which we call a 'sensor vector' in the following derivations.

2.3. SENSOR ATTENTION MECHANISM

The sensor-wise feature extraction of signals treat every IMU sensor indiscriminately, but sensors at some body locations may be not or less effective to represent a certain activity and discriminate it from others. For example, a sensor worn on the ankle may not be able to effectively perceive the 'rowing' activity. Thus, we propose a sensor attention mechanism to learn more attentions on those discriminative sensors in a signal segment. This sensor attention is a trainable layer inside a DNN, which pools the most discriminative features, as shown in Figure 7.

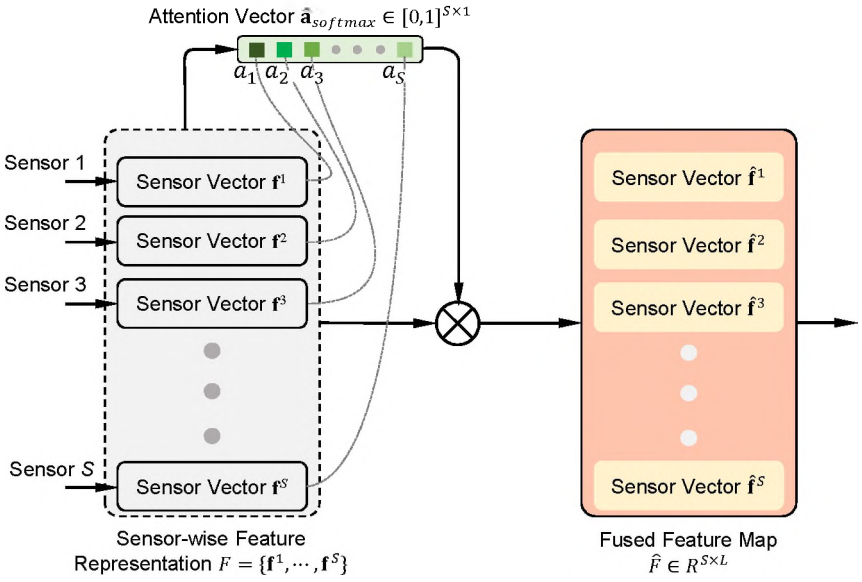


Figure 7. Illustration of the sensor attention mechanism.

Given the sensor-wise feature representation of a signal segment F, our attention module learns an attention score vector, $\vec{a}$, which indicates the feature importance of different sensors within the signal segment:

$$\vec{a} = Fw, \vec{a} \in R^{S\times 1}, \tag{6}$$

where $w \in R^{L\times 1}$ is the weight. Then, the activation vector $\hat{\vec{a}}$ is calculated as

$$\hat{\vec{a}} = tanh(W\vec{a} + \vec{b}), \tag{7}$$

where W is a weight matrix and b is a bias vector.

After the activation process, we have a set of attention score $\hat{\vec{a}}$. Then, the attention score vector is passed through a *softmax* layer:

$$a^s_{softmax} = \frac{\exp(a^s)}{\sum_{s=1}^{S} \exp(a^s)} \tag{8}$$

to get $\hat{\vec{a}}_{softmax} \in [0,1]^{S\times 1}$. Then, the attention-applied feature map $\hat{F}$ of the data segment is computed by

$$\hat{F} = F \odot \hat{\vec{a}}_{softmax}, \quad \hat{F} \in R^{S\times L} \tag{9}$$

where $\odot$ is the element-wise multiplication operator. Here each sensor (each row in $\hat{F}$) has its corresponding attention-applied feature vector $\hat{\vec{f}}$.

Overall, the proposed sensor attention mechanism fuses inputs from multiple sensors into a single representation by assembling the weighted sensor vectors from individual sensors into a 2D feature map, which enables the network to distribute different amount of attention over different sensors.

2.4. INTER-SENSOR FUSION MODULE

As shown in Figure 1, after the attention mechanism is applied, each row of the feature map comes from each individual sensor. The attentive feature map has the size of $S \times L$ ($number\ of\ sensors \times dimension\ of\ each\ sensor\ vector$). To discover the hidden correlations among different sensors. An inter-sensor fusion module is developed. This module essentially follows the same architecture as presented in Section 2.2. By using the 2D convolution, the correlation among sensors can be learned.

2.5. CLASSIFICATION MODULE

As shown in Figure 1, a classification module is designed after the inter-sensor fusion module. First, the feature map obtained from the inter-sensor fusion module are flattened as a feature vector. To solve the classification problem, the vector is further input to a multi-layer neural network. The value of the jth neuron in the ith fully connected layer, denoted as v_{ij}, is given by

$$v_{ij} = g\left(b_{ij} + \sum_{k=0}^{K_{(i-1)}-1} w_{ijk} v_{(i-1)k}\right), \tag{10}$$

where b_{ij} is the bias term, k indexes the set of neurons in the $(i-1)$th layer connected to the current feature vector, w_{ijk} is the weight value in the ith layer connecting the jth neuron to the kth neuron in the previous layer.

The last fully connected layer is used to densify the feature vector to the dimensions of M, where M is the number of activity classes. Then this M-dimensional score vector $\vec{s}([s_1, ..., s_m, ..., s_M])$ is transformed to output the predicted probabilities with a softmax function as follows:

$$P(y_n = m|X_n) = \frac{\exp(s_m)}{\sum_{j=1}^{M} \exp(s_j)} \tag{11}$$

where $P(y_n = m|X_n)$ is the predicted probability of being class m for sample X_n.

2.6. TRAINING

The process of training a DNN model involves optimization of the network's parameters θ to minimize the cost function for the training dataset X. We select the commonly used regularized cross entropy [10] as the cost function for the classifier, which is

$$\mathcal{L}(\theta) = \sum_{n=1}^{N} \sum_{m=1}^{M} y_{nm} \log[P(y_n = m|X_n)] + \lambda l_2(\theta) \tag{12}$$

where y_{nm} is 0 if the ground truth label of X_n is the mth label, and is 1 otherwise. The l_2 regularization term is appended to the loss function for penalizing large weights, and λ is its coefficient.

3. EXPERIMENTS

In this section, we first describe the selected public datasets and evaluation metrics. Then, we perform evaluation of our proposed approach using these datasets, and compare with the state-of-the-arts. After that, we conduct visualizations for a better understanding of the learned attention. Finally, future research needs are discussed.

3.1. DATASETS

As summarized in Table 1, we selected five publicly available datasets for the method validation. These datasets are collected in various contexts by different research groups, including different sensor positions on the human body, different sampling rates, and different numbers of subjects. In addition, the five datasets include activities with different levels of classification difficulties, for example, the relatively more discriminative activities [33] such as walking, sitting, and complex activities in special scenarios such as the manipulative gestures performed in a car maintenance workshop [41]. Figure 8 shows the senor locations on a human body for the five datasets. By leveraging these five different datasets, we are able to test the effectiveness and robustness of our approach.

3.1.1. Daily and Sports Activity Dataset. This dataset is composed by IMU data of 19 daily and sports activities ((1) sitting, (2) standing, (3-4) lying on the back and on the right side, (5-6) ascending and descending stairs, (7) standing in an elevator still, (8) moving around in an elevator, (9) walking in a parking lot, (10-11) walking on a treadmill with a speed of 4 km/h (in flat and 15 deg inclined positions), (12) running on a treadmill with a speed of 8 km/h, (13) exercising on a stepper, (14) exercising on a cross trainer,

(15-16) cycling on an exercise bike in horizontal and vertical positions, (17) rowing, (18) jumping, (19) playing basketball.), captured by five IMU devices (worn on the torso, right arm, left arm, right leg, and left leg, respectively), and the activities are performed by 8 different subjects [4].

Table 1. Information of the five public datasets.

Datasets	#Sensors	Modalities	Number of Channels	Rate (Hz)	Number of Activities	Number of Subjects
Daily [4]	5	A, G, M	9	25	19	8
Skoda [41]	10	A	3	98	10	1
PAMAP2 [31]	3	A, G, M	9	100	12	9
Sensors [33]	5	$A, \bar{A}, G, M$	12	50	7	10
Daphnet [3]	3	A	3	64	2	10

Note: $A, \bar{A}, G, M$ represent the modalities of acceleration, gravity-removed acceleration, angular velocity, and magnetic field, respectively.

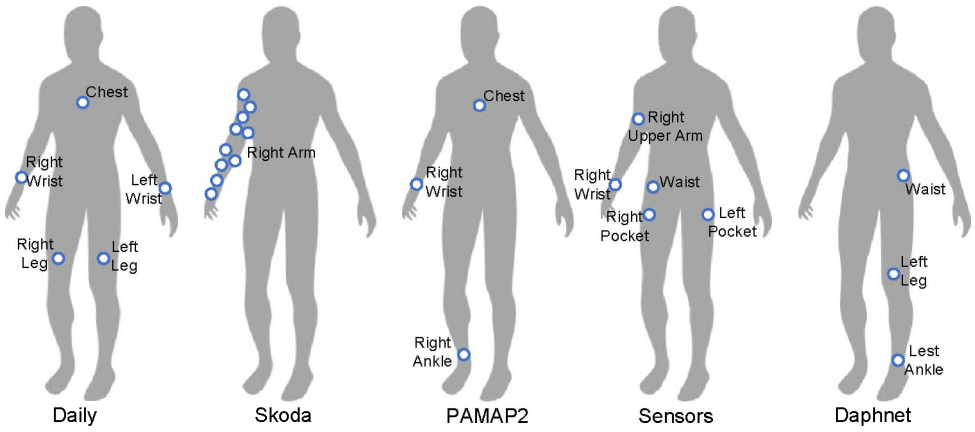


Figure 8. Worn locations of the five datasets.

3.1.2. Skoda Dataset. This dataset contains 10 manipulative activities performed in a car maintenance scenario by a single subject (e.g., the user blocks an opened hood with a stick, and the user grabs the steering wheel and turns it). The dataset has signal recordings from both the left and right arms but they are not synchronized for validation. Therefore, in this study, we focus on signals from 10 sensors worn on the subject's right arm [41].

3.1.3. PAMAP2 Dataset. This dataset has 12 human activities ((1) lying, (2) sitting, (3) standing, (4) walking, (5) running, (6) cycling, (7)Nordic walking, (8) ascending stairs, (9) descending stairs, (10) vacuum cleaning, (11) ironing and rope jumping) captured by three IMU sensors (worn on the wrist, chest and ankle, respectively), and the activities are performed by 9 different subjects [31].

3.1.4. Sensors Activity Dataset. This dataset includes 7 human activities ((1) biking, (2) downstairs, (3) jogging, (4) sitting, (5) standing, (6) upstairs, and (7) walking) captured by five IMU sensors (one in the the right jeans pocket, one in the left jeans pocket, one on the belt position towards the right leg using a belt clip, one on the right upper arm, one on the right wrist), and the activities are performed by 10 different subjects [33].

3.1.5. Daphnet Freezing of Gait Dataset. This dataset contains 3 wearable wireless acceleration sensors at the hip and leg of Parkinson's disease patients that experience freeze of gait (FoG) during walk tasks. This dataset has two classes, FoG and 'no freeze', captured by three sensors (worn at the ankle (shank), on the thigh just above the knee, and on the hip, respectively), and the activities are collected from 10 different patients [3].

3.2. EVALUATION METRICS

Regarding to evaluation metric, the leave-one-out evaluation policy is conducted. In the leave-one-out evaluation, the samples from $N_{subject} - 1$ out of $N_{subject}$ subjects are used for training, and the samples of the left one subject are reserved for testing. We employ several commonly used metrics [10] to evaluate the classification performance:

1. Accuracy

$$Accuracy = \frac{\sum_n^N 1(\hat{y}_n = y_n)}{N} \tag{13}$$

2. Precision and Recall

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \end{aligned} \tag{14}$$

3. F_1 score

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{15}$$

3.3. IMPLEMENTATION DETAILS

The DNN architectures are constructed using TensorFlow and Keras libraries. The SGD optimizer is used in training, with the momentum of 0.9, the learning rate of 0.001 and the regularizer coefficient of 1e-5. We use a workstation with one 12-core Intel Xeon processor, 64GB of RAM and two Nvidia Geforce 1080 Ti graphic cards for the training jobs.

3.4. EVALUATION OF DIFFERENT SIGNAL REPRESENTATION METHODS

To evaluate how the design of signal representation affects the model performance, comparisons have been made among methods using images of (1) raw signals (I^{RS}), (2) Discrete Cosine Transform (I^{DCT}), and (3) Discrete Fourier Transform (I^{DFT}). Table 2 shows the performance of activity recognition with various designs of input images.

The proposed signal representation method I^{DFT} achieves the highest recognition performance. The performance decreases when we use the image of raw signals I^{RS} directly or replace the Discrete Fourier Transform with the Discrete Cosine Transform (I^{DCT}). Therefore, I^{DFT} is selected for the signal representation. Another reason for choosing DFT

Table 2. Performance (%) comparison of different signal representation methods on the Daily dataset.

Methods	Input Size	Accuracy	Precision	Recall	F Score
I^{RS}	$C \times T$	67.57	64.50	67.57	61.78
$I^{RS} \xrightarrow{(DCT)} I^{DCT}$	$C \times T$	90.36	91.85	90.36	89.44
$I^{RS} \xrightarrow{(DFT)} I^{DFT}$	$C \times (T/2)$	90.37	91.86	90.37	89.82

Note: I^{RS}, I^{DCT} and I^{DFT} represent image representations of raw signals, DCT and DFT, respectively. C and L denote the number of signal channels and the number of time frames in a signal segment, respectively.

over DCT is that DFT is symmetric, and only half the image size after remove its symmetric part, which will reduce the complexity of the DNN model and has a better computational efficiency. It saves 50% of the first-layer computation over a DCT.

3.5. EVALUATION OF THE LENGTH OF THE SIGNAL SEGMENT

When sampling the signals (the sampling procedure is discussed in Section 2.1), as shown in Figure 3, there are two parameters to choose, the length of the segment (T) and the stride (Δ_t), which determine how much information the model can digest at each time, and how much shared overlap between two segments, respectively. Here the question is what should be the optimal length and stride for sampling to identify an activity. Table 3 presents the performance comparison of different settings of length and stride evaluated on the validation dataset.

The accuracy decreases when increasing the segment length, because longer length could have multiple repeated patterns in each segment, which makes it harder for the DNN model to learn the most discriminative features. Also, longer segment length leads to less segments, i.e., less training data, which affects the training effect. In terms of stride, short

Table 3. Performance (%) comparison of different settings of segment length and stride on the Daily dataset.

Length	stride	Accuracy	Precision	Recall	F Score
32	8	92.39	93.62	92.39	91.55
32	16	92.37	93.74	92.37	91.91
32	24	90.07	91.31	90.07	89.06
64	16	90.37	91.86	90.37	89.82
64	32	86.63	88.47	86.63	85.24
96	24	89.11	90.87	89.11	88.23
125	–*	85.43	87.83	85.43	84.11

*Since the sequence length of the Daily dataset is 125, the stride value is absent in the last row.

strides can have better performance. This is because the model tends to look into the data more precisely with a shorter stride. Therefore, we choose the parameter setting, $T = 32$ and $\Delta_t = 8$, for the following experiments.

3.6. EVALUATION OF THE EFFECTIVENESS OF THE FUSION MECHANISM

In terms of data fusion, as shown in Figure 1, the information flows are fused at two places: fusion of multi-channel data of a specific sensor in the sensor-wise feature extraction module (Sections 2.2) and fusion of multi-sensor data in the inter-sensor feature extraction module (Section 2.4). The fusion mechanism is realized using convolutional operations with different receptive fields, i.e., 2D kernels of different sizes. When a 2D kernel moves over an area, the hovered information is fused with the summation of point-wise multiplications. Here to validate the effectiveness of the fusion mechanism, we compare it with a method using 1D convolutions which does not include fusion functionalities. The results are listed in Table 4. We can see that, the performance drops dramatically after ignoring the fusion, which demonstrates the the designed fusion mechanism plays a vital role in identifying an activity.

Table 4. Performance (%) evaluation of the effectiveness of the fusion mechanism.

Method	Accuracy	Precision	Recall	F Score
Without Fusion Mechanism*	62.95	63.99	62.95	58.73
With Fusion Mechanism	92.37	93.74	92.37	91.91

* 1D convolutions along each row of the feature maps to ignore the fusion mechanism.

3.7. EVALUATION OF DIFFERENT FUSION METHODS

In this experiment, we compare our attention-based fusion method with two other fusion methods (early fusion and late fusion), whose architectures are presented in Figure 9.

Early fusion fuses information in the input phase. As shown in Figure 9(a), all the S inputs are stacked to generate a single input with the size of $C \times (T/2) \times S$. Then, the integrated input is fed into a DNN model.

Late fusion fuses information in the inference phase. As shown in Figure 9(b), all the S sensor inputs are learned by different DNN models individually. Then, their inferred output probabilities are fused to generate a final output.

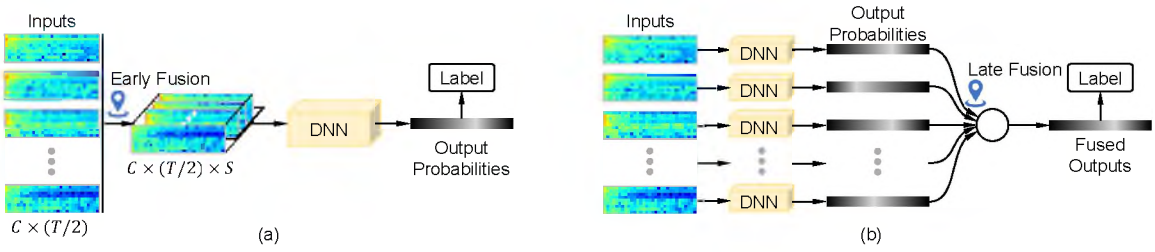


Figure 9. Architectures of different fusion methods.

The performance comparison of different fusion methods is listed in Table 5. For early fusion, the inputs are integrated before feature extraction modules of the DNN model, which lacks individual understanding of signal from each sensor. Later fusion relies on

individual sensor to learn the features and achieves higher performance, but it doesn't have the ability to look into the deep correlations among different sensors as attention fusion does. Overall, the attention fusion achieves the best results.

Table 5. Performance (%) comparison of different fusion methods.

Method	Accuracy	Precision	Recall	F Score
Early Fusion	89.62	90.63	89.62	88.86
Late Fusion	91.57	92.30	91.57	90.43
Attention Fusion	92.37	93.74	92.37	91.91

3.8. COMPARISON WITH THE STATE-OF-THE-ART METHODS

In this subsection, we compare our results with the state-of-the-art performance on the five public datasets. The comparison is summarized in Table 6. We also evaluate our model without the attention mechanism, in which the sensor attention module is removed. Overall, our proposed model achieves higher accuracy than the other methods, which is attributed to two factors: a more effective signal representation method exposing the hidden patterns and an attention-based sensor fusion model extracting the most discriminative features.

Figure 10 shows the normalized confusion matrix of the Daily dataset. We can see that most of the activities are successfully classified. Failures occur in classifying the confusing groups: e.g., (1) sitting, lying on the back, and lying on the right side; (2) standing, standing in the elevator, and moving in the elevator; (3) treadmill walking in flat position and treadmill walking in 15 deg inclined position. By reviewing the failure cases, we find that the high similarity within the confusing groups makes it difficult to distinguish them from others, and the significant subject-wise difference for the same activity makes it difficult to learn this kind of unseen variations beforehand.

Table 6. Performance (%) comparison of existing models on the five public datasets. '–' denotes that the value is not reported in the paper.

Approach	Daily	Skoda	PAMAP2	Sensors	Daphnet
Zhang et al. (2015) [44]	90.60	–	–	–	–
Hammerla et al. (2016) [13]	–	–	93.70	–	76.00
Ordóñez et al. (2016) [29]	–	95.80	–	–	–
Guan et al. (2017) [11]	–	92.40	85.40	–	–
Xi et al. (2018) [37]	–	–	93.50	–	–
Murahari and PIötz (2018) [28]	–	91.30	87.50	–	–
Zeng et al. (2018) [42]	–	93.81	89.96	–	83.73
Cao et al. (2018) [5]	78.48	–	–	–	–
Moya Rueda et al. (2018) [27]	–	–	–	93.68	–
Mohammad et al. (2018) [26]	–	91.20	–	–	–
Shakya et al. (2018) [32]	–	–	–	99.16	–
Xu et al. (2019) [38]	–	–	93.50	–	–
Our model without attention	88.55	94.16	93.14	97.36	89.81
Our model with attention	**92.37**	**95.84**	**94.85**	**99.27**	**91.02**

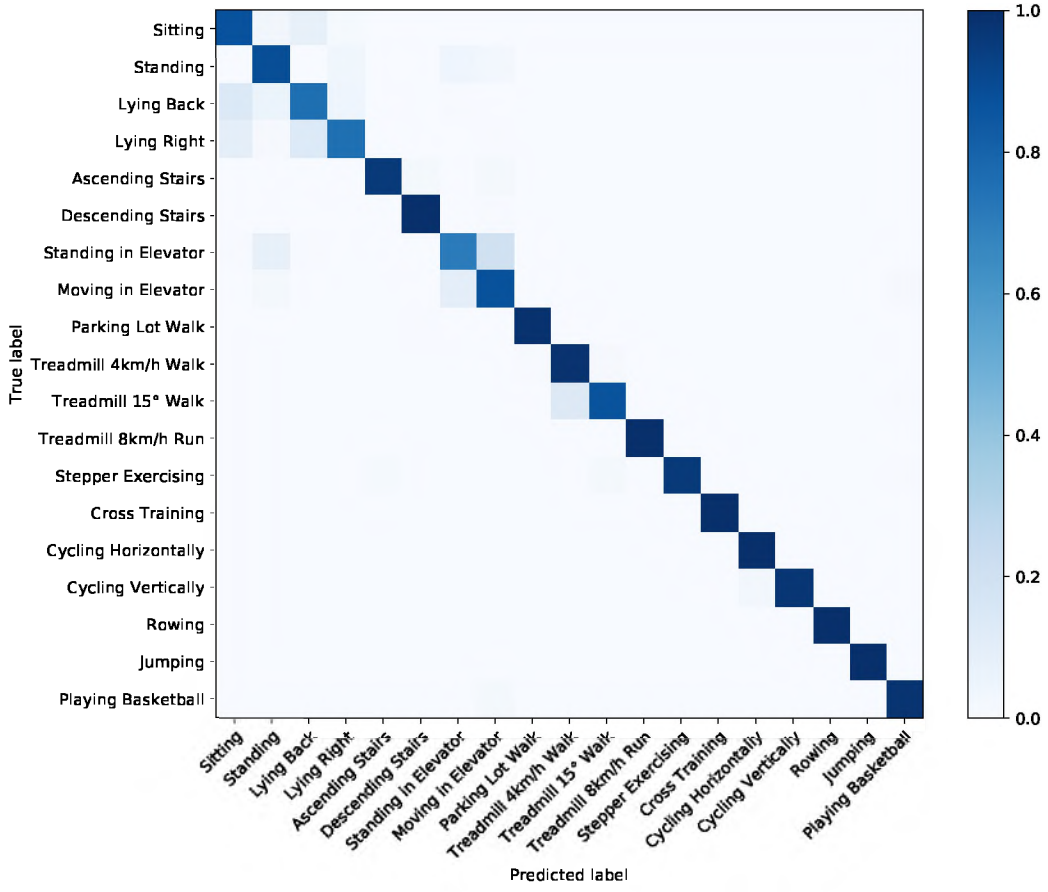


Figure 10. Normalized confusion matrix of the Daily dataset.

3.9. VISUALIZATION OF THE LEARNED SENSOR ATTENTION

In this section, we analyze and visualize the learned attention, i.e., attention weights, of sensors at different body locations. The attention vector $\hat{\vec{a}}_{softmax}$ (Eq. 8) is extracted from a well-trained model and each element of this vector is represented as a heatmap. A few examples of the sensor attention trained on the Daily dataset are shown in Figure 11, where 'hotter' colors represent larger values while 'colder' colors represent smaller ones on the blue-red heatmaps. We can see that different activities shows different attention distributions. For example, the 'rowing' activity has larger attention weights for sensors worn on the arms, because the motion intensities of the arms are larger than other body parts. While for activities such as 'running', 'jumping', and 'playing basketball', the attention is more evenly distributed across different sensors, because these activities involve the whole body. This visualization shows that our model is able to focus on the critical body parts based on their importance when identifying activities.

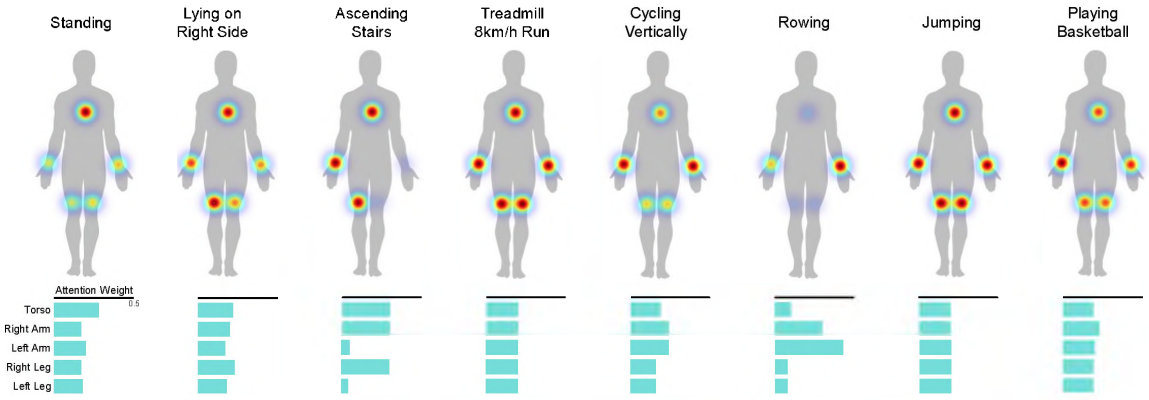


Figure 11. Examples of the importances of sensor at different body locations.

3.10. VISUALIZING THE CLASS ACTIVATION MAP

To have a more intuitive understanding of which regions of an input image are more discriminative to activate our model to its final inference, we visualize the class activation map (CAM), which is a 2D grid of scores associated with a specific output class, computed for every region in an input image, indicating the importance of each region in regard to the class under consideration. A set of CAM examples are shown in Figure 12, where the generated heatmaps are overlaid onto the input images. We can see that the model automatically learns the most discriminative regions in an input image and different activities use different regions (i.e., different signal channels and frequency characteristics) in identifying their categories.

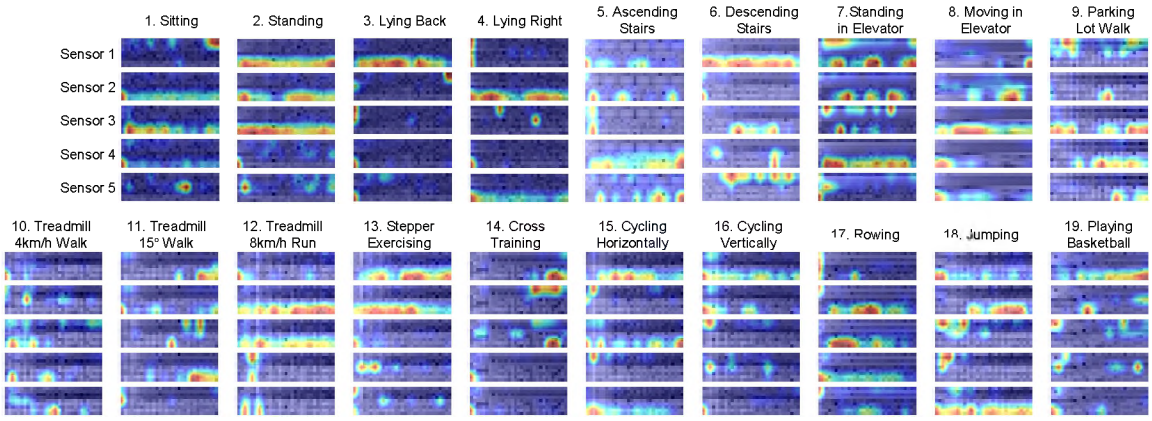


Figure 12. Examples of Class Activation Map (CAM) Visualization.

4. CONCLUSIONS

In this paper, we propose a novel approach of attention-based sensor fusion for Human Activity Recognition (HAR) using Inertial Measurement Unit (IMU) signals obtained from multiple sensors worn at different body locations. For signal representation, a simple yet effective pipeline for feature transform is designed to represent the input signals of

each sensor as images in the frequency domain. Having the formatted images as inputs, a sensor-wise feature extraction module is developed to extract the most discriminative features of signals from individual sensors with Convolutional Neural Networks (CNNs), and to generate a vector representation for each sensor. Then, a sensor attention mechanism is developed to learn the importance of sensors at different body locations and to create an attentive feature representation. After that, an inter-sensor feature extraction module is applied to learn the inter-sensor correlations, which are connected to a classifier to output the predicted classes of activities. This attention-based model is able to learn the importance of sensors at different body locations, yielding a more comprehensive understanding of the human activity. The proposed approach is evaluated on five publicly available datasets and it demonstrates superior performance than the state-of-the-art methods.

ACKNOWLEDGEMENTS

This research work is supported by the National Science Foundation grants CMMI-1646162 and NRI-1830479, and also by the Intelligent Systems Center at Missouri University of Science and Technology.

REFERENCES

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., 'TensorFlow: Large-scale machine learning on heterogeneous systems,' 2015, software available from tensorflow.org.

[2] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L., 'A public domain dataset for human activity recognition using smartphones,' in 'European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN,' 2013 .

[3] Bachlin, M., Plotnik, M., Roggen, D., Maidan, I., Hausdorff, J. M., Giladi, N., and Troster, G., 'Wearable assistant for parkinson's disease patients with the freezing of gait symptom,' IEEE Transactions on Information Technology in Biomedicine, 2010, **14**(2), pp. 436–446.

[4] Barshan, B. and Yüksek, M. C., 'Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units,' The Computer Journal, 2014, **57**(11), pp. 1649–1667.

[5] Cao, J., Li, W., Ma, C., and Tao, Z., 'Optimizing multi-sensor deployment via ensemble pruning for wearable activity recognition,' Information Fusion, 2018, **41**, pp. 68–79.

[6] Casale, P., Pujol, O., and Radeva, P., 'Human activity recognition from accelerometer data using a wearable device,' in 'Pattern Recognition and Image Analysis,' pp. 289–296, 2011.

[7] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., and Chua, T.-S., 'Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning,' in 'Proceedings of the IEEE conference on computer vision and pattern recognition,' 2017 pp. 5659–5667.

[8] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y., 'Attention-based models for speech recognition,' in 'Advances in neural information processing systems,' 2015 pp. 577–585.

[9] Duffner, S., Berlemont, S., Lefebvre, G., and Garcia, C., '3d gesture classification with convolutional neural networks,' in 'International Conference on Acoustics, Speech and Signal Processing,' 2014 pp. 5432–5436.

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

[11] Guan, Y. and Plötz, T., 'Ensembles of deep lstm learners for activity recognition using wearables,' Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2017, **1**(2), p. 11.

[12] Ha, S. and Choi, S., 'Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors,' in '2016 International Joint Conference on Neural Networks (IJCNN),' IEEE, 2016 pp. 381–388.

[13] Hammerla, N. Y., Halloran, S., and Plötz, T., 'Deep, convolutional, and recurrent models for human activity recognition using wearables,' arXiv preprint arXiv:1604.08880, 2016.

[14] Hammerla, N. Y., Kirkham, R., Andras, P., and Ploetz, T., 'On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution,' in 'International Symposium on Wearable Computers,' 2013 pp. 65–68.

[15] He, D., Zhou, Z., Gan, C., Li, F., Liu, X., Li, Y., Wang, L., and Wen, S., 'Stnet: Local and global spatial-temporal modeling for action recognition,' arXiv preprint arXiv:1811.01549, 2018.

[16] Hosein, N. and Ghiasi, S., 'Wearable sensor selection, motion representation and their effect on exercise classification,' in 'International Conference on Connected Health: Applications, Systems and Engineering Technologies,' 2016 pp. 370–379.

[17] Ijjina, E. P. and Mohan, C. K., 'One-shot periodic activity recognition using convolutional neural networks,' in 'International Conference on Machine Learning and Applications,' 2014 pp. 388–391.

[18] Ioffe, S. and Szegedy, C., 'Batch normalization: Accelerating deep network training by reducing internal covariate shift,' arXiv preprint arXiv:1502.03167, 2015.

[19] Jiang, W., Miao, C., Ma, F., Yao, S., Wang, Y., Yuan, Y., Xue, H., Song, C., Ma, X., Koutsonikolas, D., *et al.*, 'Towards environment independent device free human activity recognition,' in 'Proceedings of the 24th Annual International Conference on Mobile Computing and Networking,' ACM, 2018 pp. 289–304.

[20] Jiang, W. and Yin, Z., 'Human activity recognition using wearable sensors by deep convolutional neural networks,' in 'the 23rd Annual ACM Conference on Multimedia Conference,' 2015 pp. 1307–1310.

[21] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 'Imagenet classification with deep convolutional neural networks,' in 'Advances in neural information processing systems,' 2012 pp. 1097–1105.

[22] Lane, N. D. and Georgiev, P., 'Can deep learning revolutionize mobile sensing?' in 'the 16th International Workshop on Mobile Computing Systems and Applications,' 2015 pp. 117–122.

[23] Lara, O. D. and Labrador, M. A., 'A survey on human activity recognition using wearable sensors,' IEEE Communications Surveys & Tutorials, 2013, **15**(3), pp. 1192–1209.

[24] Luo, Z., Hsieh, J.-T., Balachandar, N., Yeung, S., Pusiol, G., Luxenberg, J., Li, G., Li, L.-J., Downing, N. L., Milstein, A., *et al.*, 'Computer vision-based descriptive analytics of seniors' daily activities for long-term health monitoring,' Machine Learning for Healthcare (MLHC), 2018.

[25] Mohamed, A.-r., Yu, D., and Deng, L., 'Investigation of full-sequence training of deep belief networks for speech recognition.' in 'INTERSPEECH,' 2010 pp. 2846–2849.

[26] Mohammad, Y., Matsumoto, K., and Hoashi, K., 'Deep feature learning and selection for activity recognition,' in 'Proceedings of the 33rd Annual ACM Symposium on Applied Computing,' ACM, 2018 pp. 930–939.

[27] Moya Rueda, F., Grzeszick, R., Fink, G., Feldhorst, S., and ten Hompel, M., 'Convolutional neural networks for human activity recognition using body-worn sensors,' in 'Informatics,' volume 5, Multidisciplinary Digital Publishing Institute, 2018 p. 26.

[28] Murahari, V. S. and Plötz, T., 'On attention models for human activity recognition,' in 'Proceedings of the 2018 ACM International Symposium on Wearable Computers,' ACM, 2018 pp. 100–103.

[29] Ordóñez, F. and Roggen, D., 'Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition,' Sensors, 2016, **16**(1), p. 115.

[30] Ravi, D., Wong, C., Lo, B., and Yang, G.-Z., 'A deep learning approach to on-node sensor data analytics for mobile or wearable devices,' IEEE Journal of Biomedical and Health Informatics, 2016.

[31] Reiss, A. and Stricker, D., 'Introducing a new benchmarked dataset for activity monitoring,' in '2012 16th International Symposium on Wearable Computers,' IEEE, 2012 pp. 108–109.

[32] Shakya, S. R., Zhang, C., and Zhou, Z., 'Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data,' International Journal of Machine Learning and Computing, 2018, **8**(6).

[33] Shoaib, M., Bosch, S., Incel, O., Scholten, H., and Havinga, P., 'Fusion of smartphone motion sensors for physical activity recognition,' Sensors, 2014, **14**(6), pp. 10146–10176.

[34] Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., and Havinga, P. J., 'Fusion of smartphone motion sensors for physical activity recognition,' Sensors, 2014, **14**(6), pp. 10146–10176.

[35] Tao, W., Leu, M. C., and Yin, Z., 'Multi-modal recognition of worker activity for human-centered intelligent manufacturing,' tba, 2019, **tba**, p. tba.

[36] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., 'Attention is all you need,' in 'Advances in neural information processing systems,' 2017 pp. 5998–6008.

[37] Xi, R., Li, M., Hou, M., Fu, M., Qu, H., Liu, D., and Haruna, C. R., 'Deep dilation on multimodality time series for human activity recognition,' IEEE Access, 2018, **6**, pp. 53381–53396.

[38] Xu, C., Chai, D., He, J., Zhang, X., and Duan, S., 'Innohar: A deep neural network for complex human activity recognition,' IEEE Access, 2019, **7**, pp. 9893–9902.

[39] Xu, Y., Shen, Z., Zhang, X., Gao, Y., Deng, S., Wang, Y., Fan, Y., Chang, E. I., *et al.*, 'Learning multi-level features for sensor-based human action recognition,' arXiv:1611.07143, 2016, 2016.

[40] Yang, J. B., Nguyen, M. N., San, P. P., Li, X. L., and Krishnaswamy, S., 'Deep convolutional neural networks on multichannel time series for human activity recognition,' in 'the 24th International Joint Conference on Artificial Intelligence,' 2015 pp. 25–31.

[41] Zappi, P., Roggen, D., Farella, E., Tröster, G., and Benini, L., 'Network-level power-performance trade-off in wearable activity recognition: A dynamic sensor selection approach,' ACM Transactions on Embedded Computing Systems (TECS), 2012, **11**(3), p. 68.

[42] Zeng, M., Gao, H., Yu, T., Mengshoel, O. J., Langseth, H., Lane, I., and Liu, X., 'Understanding and improving recurrent networks for human activity recognition by continuous attention,' in 'Proceedings of the 2018 ACM International Symposium on Wearable Computers,' ISWC '18, ACM, New York, NY, USA, ISBN 978-1-4503-5967-2, 2018 pp. 56–63, doi:10.1145/3267242.3267286.

[43] Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., and Zhang, J., 'Convolutional neural networks for human activity recognition using mobile sensors,' in '6th International Conference on Mobile Computing, Applications and Services,' 2014 pp. 197–205.

[44] Zhang, L., Wu, X., and Luo, D., 'Recognizing human activities from raw accelerometer data using deep neural networks,' in '2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA),' IEEE, 2015 pp. 865–870.

V. REAL-TIME ASSEMBLY OPERATION RECOGNITION WITH FOG COMPUTING AND TRANSFER LEARNING FOR HUMAN-CENTERED INTELLIGENT MANUFACTURING

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

[a]Missouri University of Science and Technology, Rolla, MO 65409, USA

[b]Stony Brook University, Stony Brook, NY 11794, USA

ABSTRACT

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

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

1. INTRODUCTION

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

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

In this paper, we aim to develop a real-time application for assembly operation recognition using image frames obtained from a visual camera by leveraging artificial intelligence approaches. To achieve real-time recognition, fog computing technique is introduced, which is an emerging technique that brings computing power close to data sources. It can reduce the latency and cost of delivering data to a remote cloud server [2, 12].

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

2. METHODOLOGY

2.1. THE PROPOSED FOG COMPUTING FRAMEWORK

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

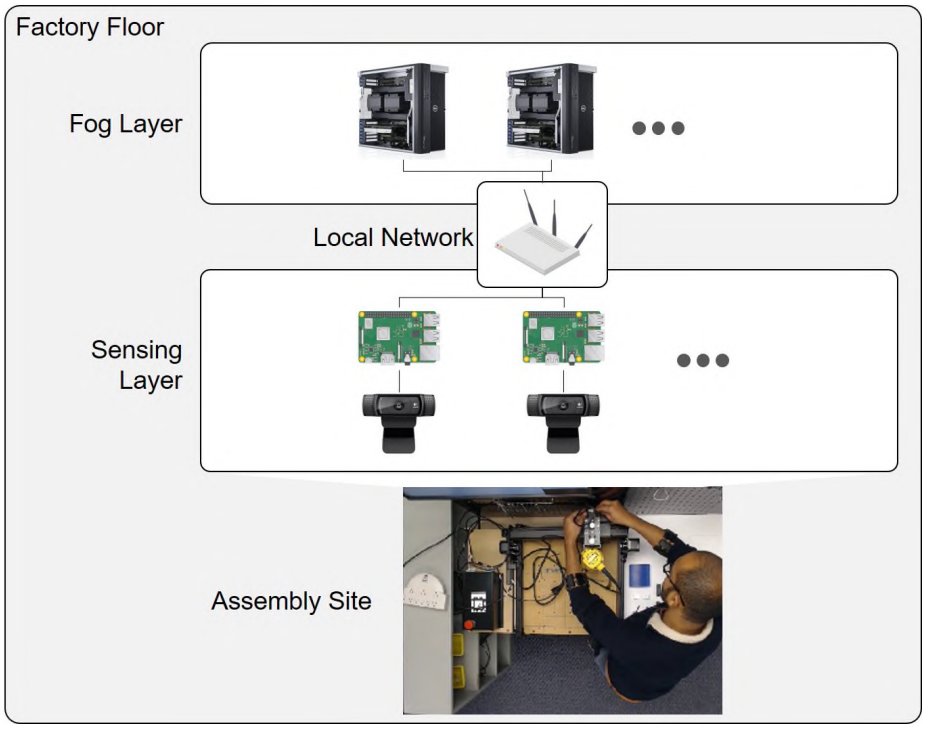


Figure 1. Overview of our fog computing framework.

2.2. ASSEMBLY TASK

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

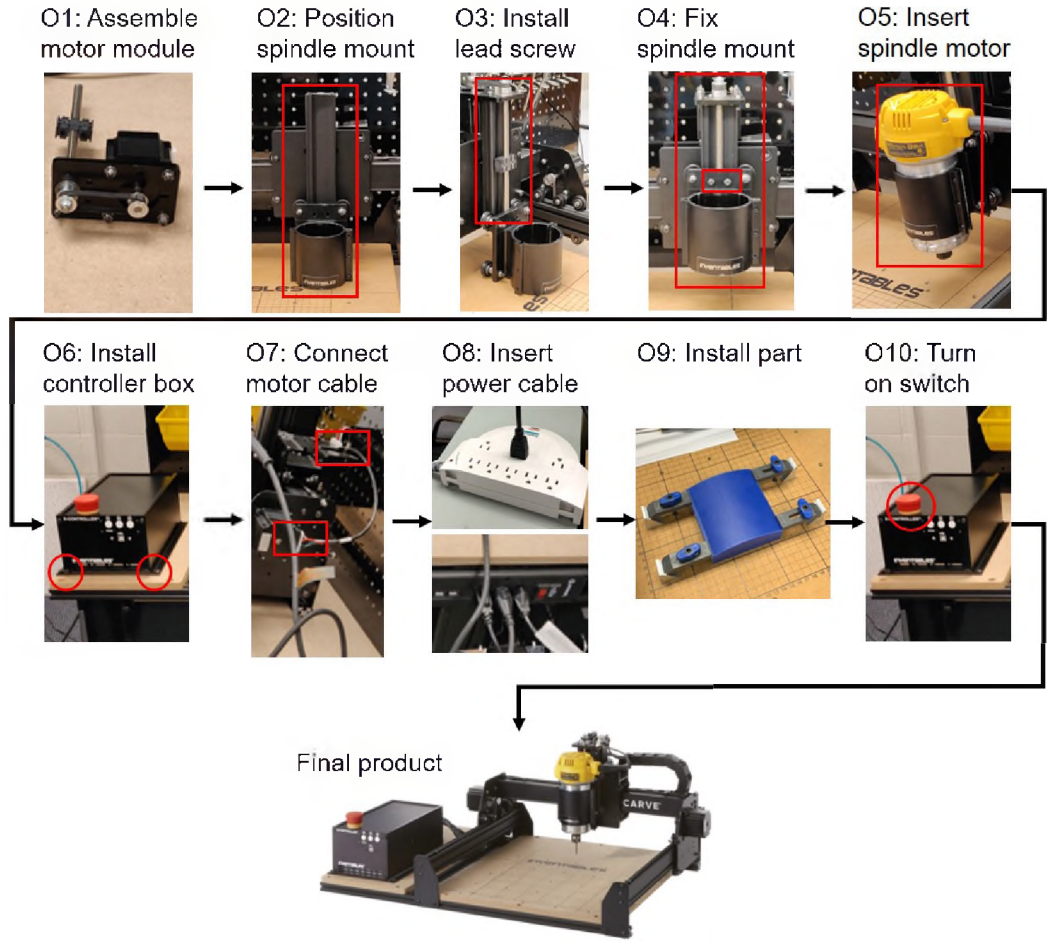


Figure 2. Illustration of the assembly task containing 10 operations from O1 to O10.

2.3. SENSING AND DATA COLLECTION

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

are taken from the top camera. For annotation purposes, each frame of a video has its frame index on the upper-left corner, and its corresponding timestamp is saved separately in another file.

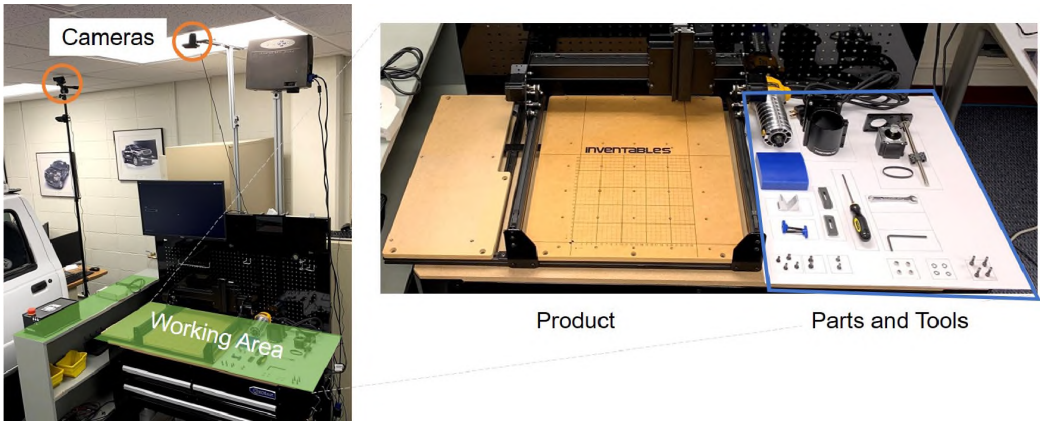


Figure 3. Illustration of the data collection setup.

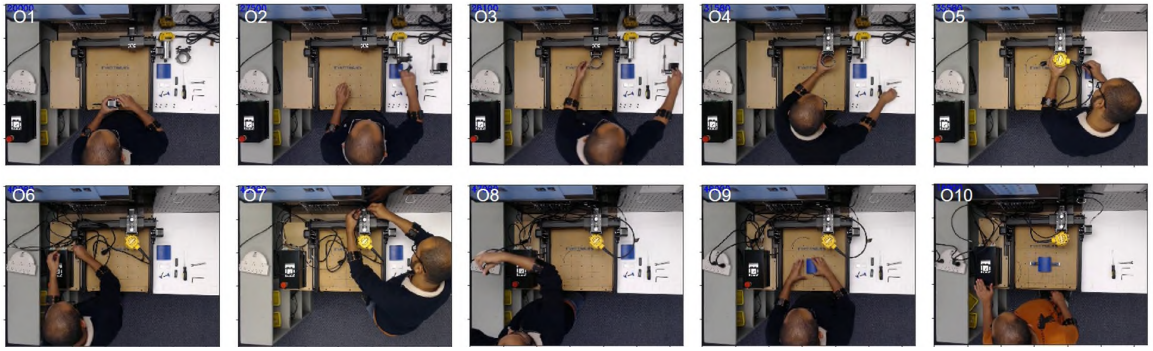


Figure 4. Examples of the 10 assembly operations.

2.4. DATA PREPROCESSING

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

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

2.5. TRANSFER LEARNING AND CUSTOMIZED CLASSIFIER

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

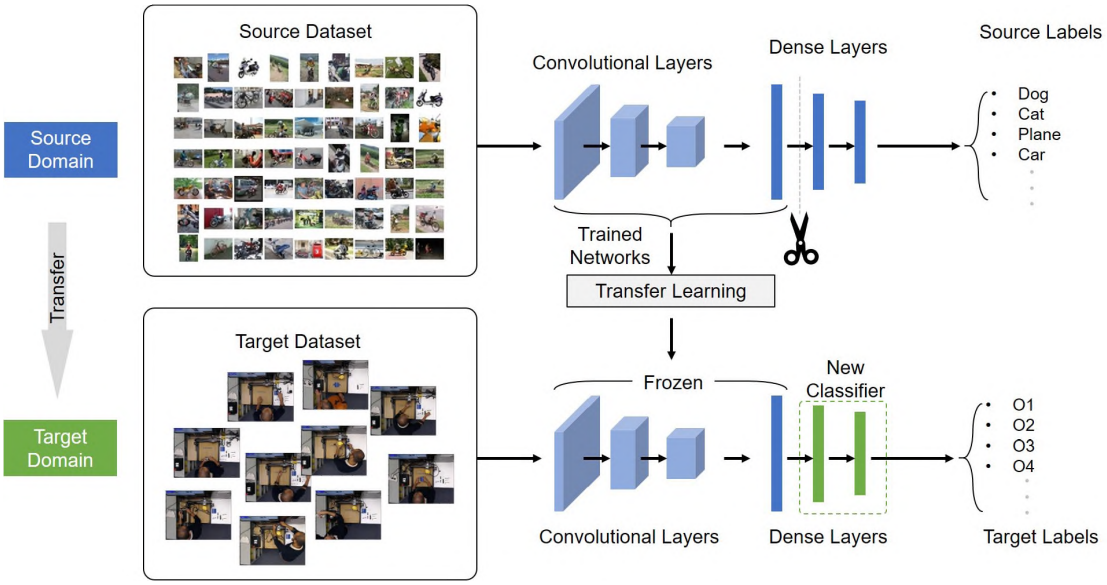


Figure 5. The architecture of our transfer learning model.

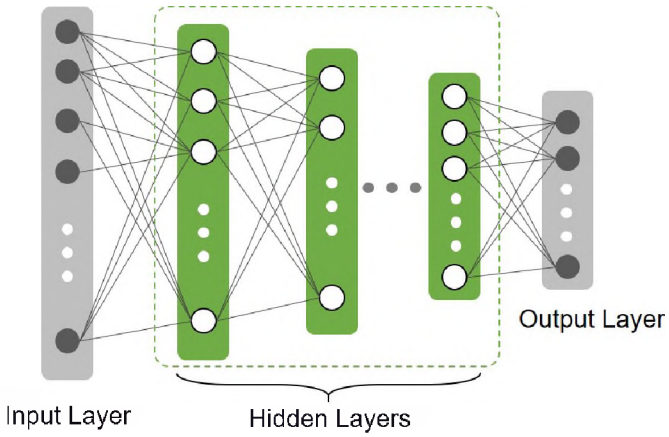


Figure 6. Illustration of the classifier architecture.

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

3. EXPERIMENTAL STUDY

3.1. DATASET ANALYSIS

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

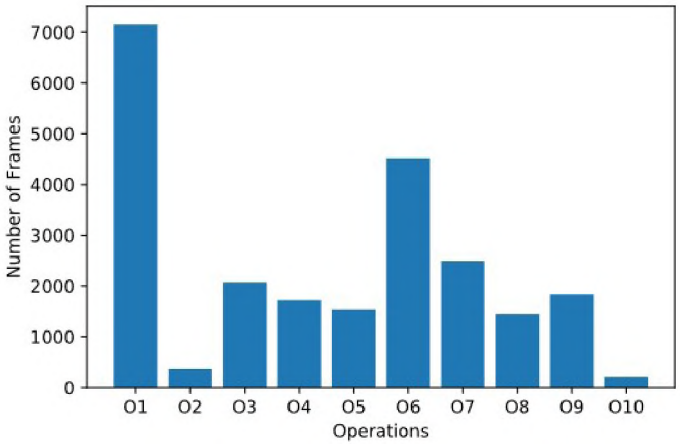


Figure 7. Averaged number of frames for each operation in the dataset.

3.2. EVALUATION METRICS

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

1. Accuracy

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

2. Precision and Recall

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

3. F_1 score

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

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

3.3. IMPLEMENTATION DETAILS

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

3.4. EVALUATION OF DIFFERENT PRE-TRAINED MODELS

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

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

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

3.5. IMPACT OF CLASSIFIER DESIGN

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

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

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

3.6. REAL-TIME RECOGNITION

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

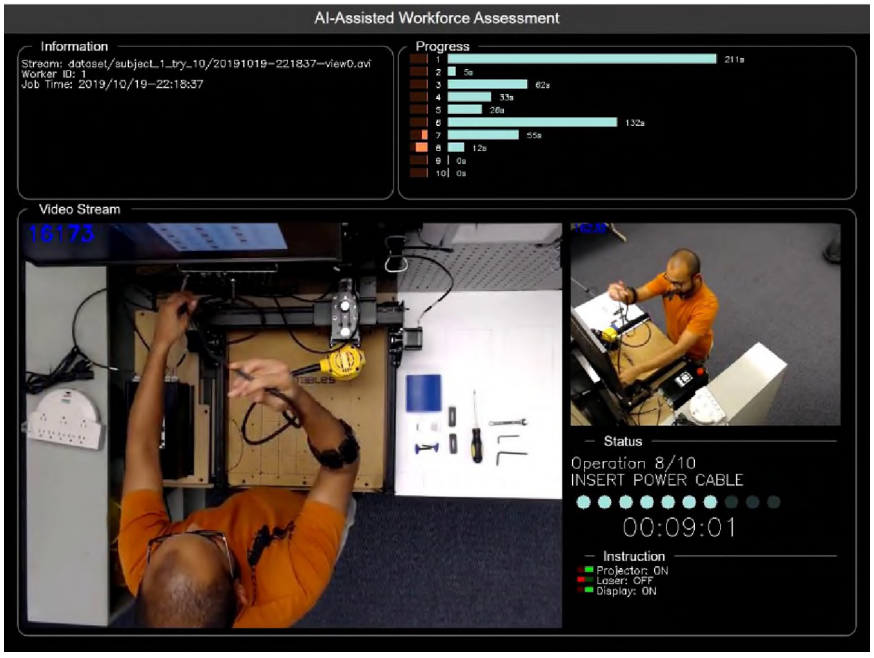


Figure 8. Real-time recognition on the testing subject.

3.7. FAILURE CASES

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

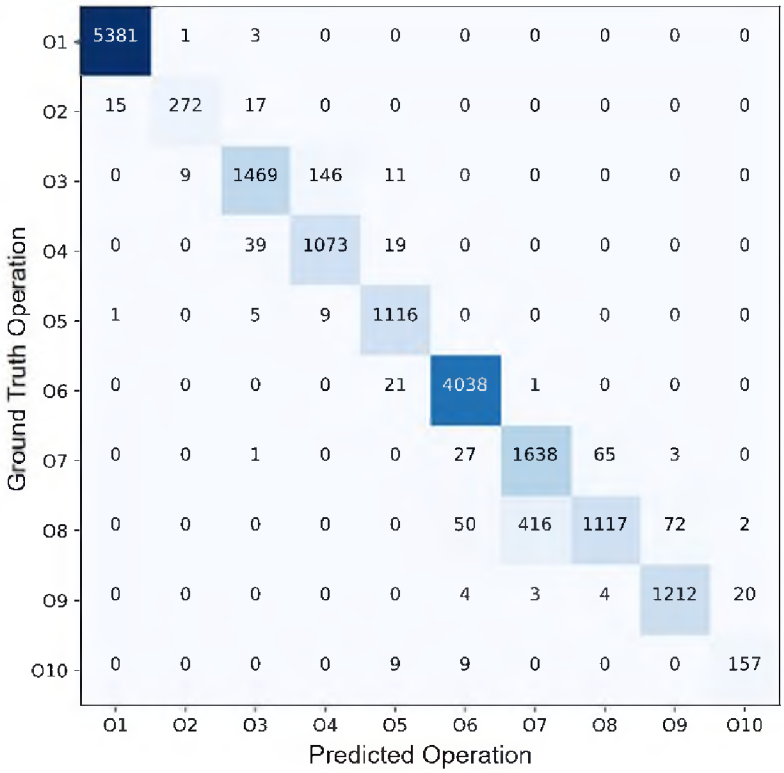


Figure 9. Confusion matrix of the experiment on the testing set.

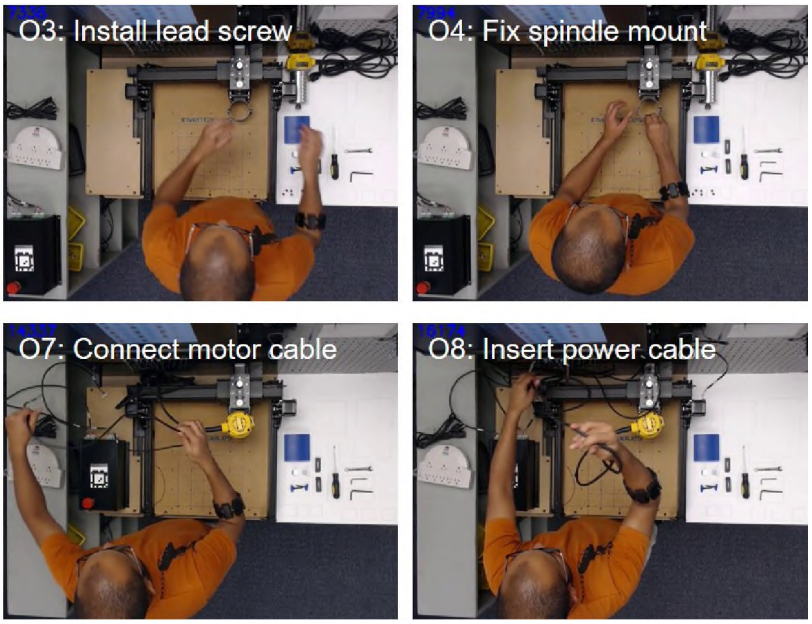


Figure 10. Failure cases from confusing pairs O3-O4 and O7-O8.

4. CONCLUSION AND FUTURE WORK

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

This is an on-going project and some directions for future study are considered, such as recruiting more subjects for data collection to enrich the current dataset, utilizing more cameras to capture the operator's activity from more perspectives, and including more modalities in the current model for information fusion. In addition, instead of using an image-based recognition method, the recording videos can be directly utilized to create a video-based operation recognition model using deep learning methods such as 3D convolutional neural networks.

ACKNOWLEDGEMENTS

This research work is supported by the National Science Foundation grants CMMI-1646162 and NRI-1830479, and also by the Intelligent Systems Center at Missouri University of Science and Technology.

REFERENCES

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

[2] Al-Khafajiy, M., Baker, T., Al-Libawy, H., Waraich, A., Chalmers, C., and Alfandi, O., 'Fog computing framework for internet of things applications,' in '2018 11th International Conference on Developments in eSystems Engineering (DeSE),' IEEE, 2018 pp. 71–77.

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

[4] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.

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

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

[7] Hu, L., Nguyen, N.-T., Tao, W., Leu, M. C., Liu, X. F., Shahriar, M. R., and Al Sunny, S. N., 'Modeling of cloud-based digital twins for smart manufacturing with mt connect,' Procedia Manufacturing, 2018, **26**, pp. 1193–1203.

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

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

[10] LeCun, Y., Bengio, Y., and Hinton, G., 'Deep learning,' Nature, 2015, **521**(7553), pp. 436–444.

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

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

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

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

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

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

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

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

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

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

SECTION

2. SUMMARY AND CONCLUSIONS

This dissertation study focused on developing systems and approaches to achieve an effective and efficient understanding of the worker's behavior on the factory floor, which provided the foundation for worker-centered intelligent manufacturing.

A novel worker-centered training & assistant system for intelligent manufacturing was proposed. This system had the self-awareness of the worker's state and could provide active guidance to the worker as needed. Compared to traditional approaches, the proposed system started with the worker's experience, considers more of the worker's learning effect, and had more interactions with the worker. The worker's state was perceived with multi-modal sensing and deep learning methods, and was used to analyze and determine the potential guiding demands. Then active instructions with augmented reality were provided to suit the worker's needs. The case studies showed the feasibility and promise of applying the proposed system for training and assisting frontline workers. Also, the proposed self-aware and active-guiding training & assistant system constructed a framework for further studies in worker-centered intelligent manufacturing.

A novel method of multiview augmentation and inference fusion for hand gesture recognition from depth images using a Convolutional Neural Network (CNN) was proposed. Multiview augmentation first retrieved the 3D information embedded in a depth image, and then generated more data for different perspective views. The result showed that it outperformed the traditional image augmentation methods because it could simulate realistic perspective variations that the traditional methods could not. Inference fusion coped with the interclass similarity issues caused by perspective variations and finger occlusions. It comprehended information of all individual views, and then outputted the final prediction,

which was proved to be effective in further improving the model's performance. The method was successfully evaluated on two public datasets, the ASL benchmark dataset and the NTU digit dataset. The experimental results demonstrated that the method made significant improvement compared to the previous work, achieving recognition accuracies of 100% and 93% in the half-half and the leave-one-out experiments, respectively, on the ASL benchmark dataset, and achieving recognition accuracies of 100% for both the half-half and the leave-one-out experiments on the NTU digit dataset.

A novel multi-modal approach for worker activity recognition was proposed. Two sensors (wearable device and camera) were adopted to perceive the worker, and four modalities were built to recognize the activity independently. Then, inference fusion was implemented to achieve an optimal understanding of the worker's behavior. Two novel mechanisms were designed to produce image representations of the IMU sensor signals in both the frequency and spatial domains. A kinematics-based data augmentation method was developed to generate more physically-realistic variations in the training dataset. This performed better than the traditional data augmentation method. A worker activity dataset was established, which had 8 subjects and contained 6 common activities in assembly tasks (i.e., grab a tool/part, hammer a nail, use a power-screwdriver, rest arms, turn a screwdriver and use a wrench). The multi-modal approach was evaluated on the dataset and achieved 100% and 97% recognition accuracy in the half-half and leave-one-out experiments, respectively.

A novel approach of attention-based sensor fusion was proposed for Human Activity Recognition (HAR) using Inertial Measurement Unit (IMU) signals obtained from multiple sensors worn at different body locations. For signal representation, a simple yet effective pipeline for feature transform was designed to represent the input signals of each sensor as images in the frequency domain. Having the formatted images as inputs, a sensor-wise feature extraction module was developed to extract the most discriminative features of signals from individual sensors with Convolutional Neural Networks (CNNs), and to generate a vector representation for each sensor. Then, a sensor attention mechanism was developed to

learn the importance of sensors at different body locations and to create an attentive feature representation. After that, an inter-sensor feature extraction module was applied to learn the inter-sensor correlations, which were connected to a classifier to output the predicted classes of activities. This attention-based model was able to learn the importance of sensors at different body locations, yielding a more comprehensive understanding of the human activity. The proposed approach was evaluated on five publicly available datasets and it demonstrated superior performance than the state-of-the-art methods.

A real-time fog computing application was developed for assembly operation recognition in human-centered intelligent manufacturing using image frames obtained from a visual camera. An assembly operation task was formulated and a dataset was established, which contained 10 sequential operations. Transfer learning was utilized and the developed model was evaluated on the dataset and achieved a 95% recognition accuracy.

3. RECOMMENDATIONS FOR FUTURE WORK

Worker behavior understanding on the factory floor remains challenging due to 1) the complexity and uncertainty of worker activity, the complexity of multi-source and heterogeneous sensing and modeling, and the complexity for human-object interaction understanding. To further improve the current dissertation study, this chapter describes some recommendations for future work.

3.1. DATA

Data are crucial for achieving good performance in deep learning tasks. Although there are lots of public datasets of human activity available for different purposes, few attempts have been made for the worker behavior understanding in manufacturing fields. Therefore, efforts can be focused on the manufacturing domain and some comprehensive datasets can be developed in terms of tools, parts, and worker activity in videos.

3.1.1. Developing an Image Dataset for Tool Recognition. Tools in the working scenarios can be treated as "standard" objects because for a specific type of tools, such as the hammers, they all have similar appearances. In this work, a tool dataset covering a wide range of tools available on the market can be created. Firstly, the categorization method of tools can be studied. Then, images belonging to these categories can be collected from the Internet and saved locally under a predefined file structure. After that, annotation information can be added to each image, such as bounding boxes denoting where the tools are. Finally, experiments of tool recognition can be implemented using some benchmark methods.

3.1.2. Developing a Data Synthesis Pipeline for Part Recognition. Compared with "tools", "parts" are less "standard" and more product-specific because their appearances can be totally different for different products. Therefore, it is not possible to collect

image data for them as how we do for collecting the tool dataset. In this work, efforts can be focused on generating the images in a synthetic manner. Firstly, 3D models of the parts can be prepared. Then, a pipeline for the synthetic dataset generation with the help of some open source software can be developed. Finally, experiments of part recognition can be implemented using some benchmark methods.

3.1.3. Developing a Video Dataset for Operational Activity Recognition in the Wild. Understanding human activities in videos is still challenging due to the complexity of activities and the variation randomness of videos. In this work, a video dataset of some common worker activities can be established. Firstly, the categorization method of worker activities can be created. Then, videos belonging to these categories can be collected from the Internet and saved locally under a predefined file structure. After that, annotation information can be added to each video, such as the starting frame and the ending frame of a certain activity. Finally, experiments of activity recognition can be implemented using some benchmark methods.

3.2. DEVELOPMENT OF INTERACTION-AWARE APPROACHES

A deep learning model could achieve good performance by digesting the appearance of a scene, and return correct answers. However, sometime the black box model may not work; even it returns the correct answer, it uses the unrelated information as the cues. The black box model cannot provide a "deep" understanding of the contents, such as where the hands are and what the interactions are. The understanding of a deep model is still superficial. It motivates us to propose a model which can have more understandings, and the research should aim to push the model to learn more meaningful contents and make the deep learning model more explainable, rather than a black box.

To have a better understanding of input image, different modules can be designed, including a hand detection & tracking module, an object detection module, and an interaction module. The first module can detect the hands appeared in the scene and track their

trajectories, which is informative for understand the hand spatial movements. The second module can detect objects, i.e., tools and parts, in the scene. The third module is designed to recognize the relation between hands and objects, such as what object the hand is holding. Then, knowledge from the three modules can be fused to infer the final understanding of the input image, e.g., a hand is tightening a bolt to a part with a wrench.

REFERENCES

[1] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L., 'A public domain dataset for human activity recognition using smartphones.' in 'ESANN,' 2013 .

[2] Carreira, J. and Zisserman, A., 'Quo vadis, action recognition? a new model and the kinetics dataset,' in 'Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on,' IEEE, 2017 pp. 4724–4733.

[3] Chang, W., Dai, L., Sheng, S., Tan, J. T. C., Zhu, C., and Duan, F., 'A hierarchical hand motions recognition method based on imu and semg sensors,' in 'Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on,' IEEE, 2015 pp. 1024–1029.

[4] Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., and Chua, T.-S., 'Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning,' in 'Proceedings of the IEEE conference on computer vision and pattern recognition,' 2017 pp. 5659–5667.

[5] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y., 'Attention-based models for speech recognition,' in 'Advances in neural information processing systems,' 2015 pp. 577–585.

[6] Duffner, S., Berlemont, S., Lefebvre, G., and Garcia, C., '3d gesture classification with convolutional neural networks,' in 'International Conference on Acoustics, Speech and Signal Processing,' 2014 pp. 5432–5436.

[7] Ha, S. and Choi, S., 'Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors,' in '2016 International Joint Conference on Neural Networks (IJCNN),' IEEE, 2016 pp. 381–388.

[8] Hammerla, N. Y., Kirkham, R., Andras, P., and Ploetz, T., 'On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution,' in 'International Symposium on Wearable Computers,' 2013 pp. 65–68.

[9] He, D., Zhou, Z., Gan, C., Li, F., Liu, X., Li, Y., Wang, L., and Wen, S., 'Stnet: Local and global spatial-temporal modeling for action recognition,' arXiv preprint arXiv:1811.01549, 2018.

[10] Hosein, N. and Ghiasi, S., 'Wearable sensor selection, motion representation and their effect on exercise classification,' in 'International Conference on Connected Health: Applications, Systems and Engineering Technologies,' 2016 pp. 370–379.

[11] Ijjina, E. P. and Mohan, C. K., 'One-shot periodic activity recognition using convolutional neural networks,' in 'International Conference on Machine Learning and Applications,' 2014 pp. 388–391.

[12] Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., and Eschert, T., 'Industrial internet of things and cyber manufacturing systems,' in 'Industrial Internet of Things,' pp. 3–19, Springer, 2017.

[13] Ji, S., Xu, W., Yang, M., and Yu, K., '3d convolutional neural networks for human action recognition,' IEEE transactions on pattern analysis and machine intelligence, 2013, **35**(1), pp. 221–231.

[14] Jiang, W., Miao, C., Ma, F., Yao, S., Wang, Y., Yuan, Y., Xue, H., Song, C., Ma, X., Koutsonikolas, D., *et al.*, 'Towards environment independent device free human activity recognition,' in 'Proceedings of the 24th Annual International Conference on Mobile Computing and Networking,' ACM, 2018 pp. 289–304.

[15] Jiang, W. and Yin, Z., 'Human activity recognition using wearable sensors by deep convolutional neural networks,' in 'Proceedings of the 23rd ACM international conference on Multimedia,' ACM, 2015 pp. 1307–1310.

[16] Kober, J., Bagnell, J. A., and Peters, J., 'Reinforcement learning in robotics: A survey,' The International Journal of Robotics Research, 2013, **32**(11), pp. 1238–1274.

[17] Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P., and Roning, J., 'Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines,' in 'Control and Automation, 2009. MED'09. 17th Mediterranean Conference on,' IEEE, 2009 pp. 401–405.

[18] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 'Imagenet classification with deep convolutional neural networks,' in 'Advances in neural information processing systems,' 2012 pp. 1097–1105.

[19] Lane, N. D. and Georgiev, P., 'Can deep learning revolutionize mobile sensing?' in 'the 16th International Workshop on Mobile Computing Systems and Applications,' 2015 pp. 117–122.

[20] LeCun, Y., Bengio, Y., and Hinton, G., 'Deep learning,' Nature, 2015, **521**(7553), pp. 436–444.

[21] Lee, J., Ardakani, H. D., Yang, S., and Bagheri, B., 'Industrial big data analytics and cyber-physical systems for future maintenance & service innovation,' Procedia CIRP, 2015, **38**, pp. 3–7.

[22] Maekawa, T., Nakai, D., Ohara, K., and Namioka, Y., 'Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory,' in 'Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing,' ACM, 2016 pp. 1088–1099.

[23] Mohamed, A.-r., Yu, D., and Deng, L., 'Investigation of full-sequence training of deep belief networks for speech recognition.' in 'INTERSPEECH,' 2010 pp. 2846–2849.

[24] Nagorny, K., Lima-Monteiro, P., Barata, J., and Colombo, A. W., 'Big data analysis in smart manufacturing: A review,' International Journal of Communications, Network and System Sciences, 2017, **10**(03), p. 31.

[25] Peterek, T., Penhaker, M., Gajdoš, P., and Dohnálek, P., 'Comparison of classification algorithms for physical activity recognition,' in 'Innovations in Bio-inspired Computing and Applications,' pp. 123–131, Springer, 2014.

[26] Petruck, H. and Mertens, A., 'Using convolutional neural networks for assembly activity recognition in robot assisted manual production,' in 'International Conference on Human-Computer Interaction,' Springer, 2018 pp. 381–397.

[27] Ravi, D., Wong, C., Lo, B., and Yang, G.-Z., 'A deep learning approach to on-node sensor data analytics for mobile or wearable devices,' IEEE Journal of Biomedical and Health Informatics, 2016.

[28] Ronao, C. A. and Cho, S.-B., 'Human activity recognition using smartphone sensors with two-stage continuous hidden markov models,' in 'Natural Computation (ICNC), 2014 10th International Conference on,' IEEE, 2014 pp. 681–686.

[29] Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., and Troster, G., 'Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario,' in 'Wearable Computers, 2006 10th IEEE International Symposium on,' IEEE, 2006 pp. 97–104.

[30] Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., and Tröster, G., 'Wearable activity tracking in car manufacturing,' IEEE Pervasive Computing, 2008, **7**(2).

[31] Stiefmeier, T., Roggen, D., and Troster, G., 'Fusion of string-matched templates for continuous activity recognition,' in 'Wearable Computers, 2007 11th IEEE International Symposium on,' IEEE, 2007 pp. 41–44.

[32] Thalmic Labs Inc., 'Myo armband,' 2017, [Online; accessed 15-November-2017].

[33] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., 'Attention is all you need,' in 'Advances in neural information processing systems,' 2017 pp. 5998–6008.

[34] Wang, P., Liu, H., Wang, L., and Gao, R. X., 'Deep learning-based human motion recognition for predictive context-aware human-robot collaboration,' CIRP Annals, 2018.

[35] Xu, Y., Shen, Z., Zhang, X., Gao, Y., Deng, S., Wang, Y., Fan, Y., Chang, E. I., *et al.*, 'Learning multi-level features for sensor-based human action recognition,' arXiv:1611.07143, 2016, 2016.

[36] Yang, J. B., Nguyen, M. N., San, P. P., Li, X. L., and Krishnaswamy, S., 'Deep convolutional neural networks on multichannel time series for human activity recognition,' in 'the 24th International Joint Conference on Artificial Intelligence,' 2015 pp. 25–31.

[37] Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., and Zhang, J., 'Convolutional neural networks for human activity recognition using mobile sensors,' in '6th International Conference on Mobile Computing, Applications and Services,' 2014 pp. 197–205.

VITA

Wenjin Tao received his B.S. and M.S. degrees in Mechanical Engineering in 2014 from Beijing Institute of Technology, Beijing, China. In August 2020, he received his Doctor of Philosophy in Mechanical Engineering from Missouri University of Science and Technology, Rolla, Missouri, USA. His research interests lay at the intersection of Advanced Manufacturing, Artificial Intelligence and Robotics, including Cyber-Physical Systems, Industry 4.0, big data analytics, machine learning, human behavior understanding, human-computer/robot interaction, virtual reality/augmented reality, AI-enabled intelligent manufacturing such as data-driven prognostics and predictive maintenance, and design optimization for additive manufacturing. He has authored and co-authored nine journal papers, eight peer-reviewed conference papers, one book, two book chapters, and four patents. He received the Best Paper Award in IISE (Institute of Industrial and Systems Engineers) Annual Conference, 2018.