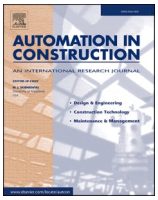
Automation in Construction 125 (2021) 103625

Contents lists available at ScienceDirect

Automation in Construction

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



Vision-based hand signal recognition in construction: A feasibility study

Xin Wang, Zhenhua Zhu *

Department of Civil and Environmental Engineering, University of Wisconsin-Madison, 1415 Engineering Drive, Madison, WI 53706, USA

ARTICLE INFO

Keywords:
Hand signal recognition
Dataset creation
Performance comparison
Feasibility study

ABSTRACT

In construction fields, it is common for workers to rely on hand signals to communicate and express thoughts due to their simple but effective nature. However, the meaning of these hand signals was not always captured precisely. As a result, construction errors and even accidents were produced. This paper presented a feasibility study on investigating whether the hand signals could be captured and interpreted automatically with computer vision technologies. It starts with the literature review of existing hand gesture recognition methods for sign language understanding, human-computer interaction, etc. It is then followed by creating a dataset containing 11 classes of hand signals in construction. The performance of two state-of-the-art 3D convolutional neural networks is measured and compared. The results indicated that a high classification accuracy (93.3%) and a short inference time (0.17 s/gesture) could be achieved, illustrating the feasibility of using computer vision to automate hand signal recognition in construction.

1. Introduction

Maintaining good communication on a construction project site is crucial for keeping the site safe and the project running smoothly and on schedule [1–3]. One of common ways for workers at distance to communicate is through the use of radio transmission. However, it is easily interfered with other senders. Also, both sender and receiver need to be equipped with radio transmission devices kept in the same channel. These limitations impact its wide use for all workers on construction sites.

Hand signals, as another common way to communicate, are used on construction sites due to their simple but effective nature [3–5]. They help workers from different backgrounds and cultures to communicate with each other [6]. Also, consider that construction sites can be exceptionally noisy and words may not be heard clearly. Hand signals aid workers to receive correct directions without the need for complicated way-to-way communication devices [7].

However, hand signals may not always be captured timely or interpreted correctly in the fields. The failure to capture and interpret hand signals easily leads to worker injuries/fatalities, work interruption, and stoppage, etc. For example, it was reported that a concrete buggy fell off a hoist platform and hit the ground 18′ below because a hoist engineer misinterpreted a signal and lowered the platform in advance [8]. Another accident was noted when a crew chief entered an active work area on an All-Terrain Vehicle (ATV). Although he was asked to leave by a foreman using a hand signal, the signal was not captured by the crew chief. As a result, the ATV was hit by a bulldozer and the chief suffered a fractured leg [9]. A concrete truck driver misread the hand signals given by an officer and hit a 27-year-old electrician, who was working on the replacement of traffic lights from the buck of his truck [10]. The incident made the electrician thrown from the bucket and left dangling in the air [10].

These accidents indicate the need to provide a complementary way to help workers capture and interpret hand signals in construction fields. So far, there are many research studies proposed for hand gesture recognition; and the methods in these studies were applied to identify traffic police hand signals, understand sign languages, promote human-computer interactions, etc. [11–13]. They either relied on hand-crafted visual features, such as Histogram of Origented Gradients (HOG) [14] and improved dense trajectories (iDT) [15], or deep neural networks including 2D convolutional neural networks (CNNs) [16,17] and 3D-CNNs [18,19]. The performance of these methods was measured by several datasets, such as NvGesture [20] and EgoGesture [16]. The results illustrated the potential of deep neural networks to achieve hand signal recognition with excellent spatiotemporal learning ability.

Although the performance of existing methods for hand gesture recognition is promising, it is still not clear whether they could be applied in the construction field to capture and interpret hand signals made by construction workers. It is due to the following two major reasons. First, construction scenarios are complex and cluttered with

* Corresponding author.
E-mail addresses: xwang2463@wisc.edu (X. Wang), zzhu286@wisc.edu (Z. Zhu).

https://doi.org/10.1016/j.autcon.2021.103625
Received 6 October 2020; Received in revised form 16 January 2021; Accepted 5 February 2021
Available online 15 February 2021

tools, materials, machines, etc. Also, the hand signal recognition in the construction fields may be impacted by environmental conditions and the existence of other workers. These characteristics have not been fully represented in existing datasets for hand gesture recognition. In addition, the datasets used for testing hand gesture recognition methods contain the video clips of hand gestures that were typically recorded when subjects were sitting or standing still. It is not clear whether existing recognition methods would work when workers are moving and making hand signals simultaneously.

This paper presented a feasibility study on investigating whether hand signals in the construction field could be captured and interpreted automatically with computer vision technologies. First, we provided a literature review of existing hand gesture recognition methods developed for traffic police hand signal identification, sign language understanding, and human-computer interaction. After that, a new dataset containing 11 classes of hand signals for instructing tower crane operations was created under different scenes (e.g. outdoor vs. indoor and single worker vs. multiple workers). To measure the recognition performance with the created dataset, two state-of-the-art 3D-CNNs, namely, ResNeXt-101 and Res3D + ConvLSTM+MobileNet were employed to achieve hand signal recognition. The recognition results demonstrated that a high classification accuracy (93.3%) and a short inference time (0.17 s/gesture) could be achieved, which illustrated the feasibility and potential of computer vision technologies to automate the hand signal recognition in the construction field.

2. Related work

2.1. Vision-based hand gesture recognition

The hand gesture recognition is a hot topic in computer vision and pattern recognition, which plays an increasingly important role in natural human-computer interface [21–23]. Currently, many efforts have been dedicated to hand gesture recognition. They generally relied on motion sensory data [24,25] and videos [16,18]. The motion sensory data usually can be collected by various motion sensors attached on human bodies. These sensors include Micro-electromechanical Systems (MEMS), Inertial Measurement Unit (IMU), electromyography (EMG) devices, etc. The video data contain visual information (e.g., RGB and depth images), which can be conveniently collected by smart phones, RGB-D cameras or stereo cameras.

So far, the video data are widely adopted in the hand gesture recognition field due to their convenient and effective nature. Many vision-based methods have been developed to recognize hand gestures using hand-crafted features or through deep learning. Traditional methods generally relied on hand-crafted features, such as HOG [14], iDT [15] and Mix Features Around Sparse Keypoints (MFSK) [26]. Besides these features, many research studies were focused on deriving novel features to represent the appearance, shape, and motion changes of a gesture [27–29]. For example, Singha and Das [27] proposed an integrated system for recognizing hand gestures, which included three stages: separating skin-colored regions from non-skin colored ones in the preprocessing stage, calculating Eigenvalues for feature extraction and finally recognizing the gestures based on the Eigenvalues using the weighted Euclidean distance. Wang et al. [28] employed a Hidden Markov Model (HMM) based method for the modeling and recognition of hand gestures. The method included the following parts: detecting a palm from a video sequence, recording its center trajectory, extracting the discrete vector features from the trajectory, and classifying the gesture using HMM. Lin et al. [29] developed a statistical method to detect the hand region and derived a new feature descriptor from the hand shape. This feature descriptor was combined with a Gaussian Mixture Model (GMM) to recognize hand gestures. Overall, the key point of these methods is to derive these sophisticated features and then feed them into a classifier to achieve hand gesture recognition.

Recently, the methods using deep learning have become mainstream in hand gesture recognition. Generally, their frameworks can be divided into four types. The first type is to use 2D-CNNs to extract features of single frames [17,30]. Oyedotun and Khashman [30] developed a hand gesture recognition system using three architectures of 2D-CNNs with different hidden layers. Kurmanji and Ghaderi [17] employed famous 2D-CNNs including GoogleNet and AlexNet to identify hand gestures.

The second framework type is to utilize 3D-CNNs to extract features of video clips and then aggregate clip features into video descriptors [18,31]. For instance, Miao et al. [31] proposed a multimodal gesture recognition method using a Res3D network. The extracted spatiotemporal features from the Res3D were combined through canonical correlation analysis and the final recognition was made by a linear SVM classifier. Köpüklü et al. [18] proposed a hierarchical CNN structure to realize the real-time hand signal recognition. The proposed architecture firstly employed a detector which was a lightweight 3D-CNN (ResNet-10) to detect the existence of hand gestures and then utilized deep 3D-CNNs (C3D and ResNeXt-101) to classify the detected gestures.

The third framework type is to combine CNNs with Long Short Term Memory (LSTM) layers to model the temporal evolution of sequences [20,32,33]. For example, Molchanov et al. [20] combined 3D-CNN (C3D) with recurrent layers to perform simultaneous detection and classification of dynamic hand gestures. The recurrent 3D-CNN enabled the gesture classification without requiring explicit pre-segmentation. Cao et al. [32] presented a framework of C3D + LSTM+RSTTM which augmented C3D with a recurrent spatiotemporal transform module. The presented framework could not only capture short-term spatiotemporal features but also model long-term dependencies. Zhang et al. [33] proposed a Res3D + ConvLSTM+MobileNet architecture to recognize hand gestures. In their work, Res3D was used first to learn the local short-term spatiotemporal feature maps. Then, two ConvLSTM layers were stacked to learn the global long-term spatiotemporal feature maps. Finally, parts of MobileNet were employed to learn deeper features based on the learnt two-dimensional spatiotemporal feature maps.

The fourth framework type is to adopt a two-stream CNN architecture where two CNNs are employed to model spatial and temporal information of sequences, separately [34,35]. Wu et al. [34] developed AlexNet network into a two-stream 2D-CNN structure to achieve hand gesture recognition. Huang et al. [35] designed a two-stream 3D-CNN based on C3D where one stream focused on the local, detailed hand gestures while the other stream was designed to extract global hand motions. The contributions of all the methods mentioned above have been summarized in Table 1.

Table 1
Summary of existing vision-based methods for hand gesture recognition.

Categories	Types	Referred Methods	Test Dataset	Reported Accuracy
Feature-based	Hand-crafted features + Classifier	[14,15,26–29]	CAD-60, ChaLearn MMGR, Indian sign language, Patch, etc.	87.1% - 96.2%
Deep learning based	2D-CNNs	[17,30]	VIVA HGD, Cambridge HGD, American sign language, etc.	70.5% -92.8%
	3D-CNNs	[18,31]	EgoGesture, NvGesture, Chalearn IsoGD, etc.	67.7% - 91.9%
	CNNs + LSTM layers	[20,32,33]	EgoGesture, NvGesture, Jester, Chalearn IsoGD, etc.	56.0% -95.1%
	Two-stream CNNs	[34,35]	HandLogin, BodyLogin, German sign language, Chinese sign language, etc.	61.7% - 82.7%

2.2. Datasets for hand gesture recognition

There are several datasets publicly available to evaluate hand gesture recognition performance. They could be classified into two categories, first-person view and second-person view, depending on how the hand gestures in the datasets are captured. In the first-person view dataset, the hand gestures are typically captured by mounting a camera on the forehead of a user, which simulates a natural viewpoint seen through the user's eyes. Examples of the first-person view dataset could be found in the work of Starner et al. [36], Baraldi et al. [37], and Zhang et al. [16]. Starner et al. [36] proposed an egocentric gesture dataset which defined 40 American sign language gestures. Baraldi et al. [37] presented an Interactive Museum dataset containing 7 gesture classes to allow visitors to interactively view the exhibits in a virtual museum environment. Zhang et al. [16] designed an EgoGesture dataset with 83 classes of hand gestures intended to interact with wearable devices.

Most of the datasets were prepared in a second-person view, where a camera is kept a short distance towards a user while recording his/her hand gestures. Several of them were designed for general symbolic hand gestures, such as Cambridge Hand Gesture Dataset [38] and Sheffield KInect Gesture (SKIG) Dataset [39]. Others were created to support human-computer interaction with touchless screens [19,40,41] or in vehicles [14,20], sign language interpretation for English [42] and Italian [43]. The ChaLearn IsoGD Dataset [44] covered various domains including Italian sign language, helicopter and traffic signals, pantomimes and symbolic gestures, and body language. Table 2 summarizes those publicly available datasets in the field of hand gesture recognition.

2.3. Comparison study

To date, many comparison studies have been conducted for those hand gesture recognition methods. Molchanov et al. [20] compared the performances of state-of-the-art methods on the public dataset NvGesture. The comparison results indicated that the recurrent 3D-CNN achieved the best performances in each individual modality (RGB, depth and RGB—D). Zhang et al. [16] evaluated the recognition performances of several representative methods based on hand-crafted features and deep learning technologies. The results showed that the methods based on deep learned features performed better in general. Among those deep learning approaches, the performances of the C3D-based methods were superior to others with accuracy improvement by more than 10%. In the studies of Köpüklü et al. [18] and Benitez-Garcia et al. [19], the results showed that the ResNeXt-101 achieved the best performance. In the work of Kurmanji and Ghaderi [17], the comparison results indicated that the 3D-CNNs could extract the changes in the consecutive frames and tended to be more suitable for the classification of hand gestures in a video sequence; however, they usually needed more time.

Several findings have been noted from existing comparison studies. First, deep learning methods have become mainstream in hand gesture recognition tasks due to the following reasons. The methods based on deep learned features outperformed those based on hand-crafted features in recognition accuracy [16,20]. Also, the way to calculate hand-crafted features is usually computationally intensive and requires a high storage cost. As a result, the methods based on the hand-crafted features are not suitable when a recognition dataset is large-scale [16]. Second, as for those deep learning methods, 3D-CNNs (e.g. C3D, Res3D, ResNeXt-101) illustrated an excellent spatiotemporal learning ability [16,17,20] compared to 2D-CNNs. This is because 2D-CNNs typically process individual video frames directly. They can only characterize the visual appearance but not establish temporal relations between consecutive frames. In addition, the recognition performance of the deep learning methods could be further improved through using a two-stream architecture for CNNs [17] or combing CNNs with LSTM layers [20]. These ways could enhance the ability of the method to capture the temporal information in the recognition process.

Table 2
Datasets in the field of hand gesture recognition.

Datasets	# of Samples	# of Classes	Data source	Usage	View
Egocentric gesture dataset [36]	2500	40	RGB	Sign language	First-person
Interactive Museum [37]	700	7	RGB	Human interaction with exhibits in virtual environment	First-person
EgoGesture [16]	24,161	83	RGB-D	Human interaction with wearable devices	First-person
Cambridge Hand Gesture Dataset [38]	900	9	RGB	General symbolic hand gestures	Second-person
SKIG [39]	1080	10	RGB-D	General symbolic hand gestures	Second-person
ChAirGest [40]	1200	10	RGB-D, IMU	Human interaction with touchless screens	Second-person
CVVR-HAND [14]	886	19	RGB-D	Human interaction with devices in a vehicle	Second-person
NvGesture [20]	1532	25	RGB-D, stereo-IR	Human interaction with devices in a vehicle	Second-person
Jester [41]	148,092	27	RGB	Human interaction with computer	Second-person
IPN Hand [19]	4218	13	RGB	Human interaction with touchless screens	Second-person
MSRGesture3D [42]	336	12	Depth	Sign language	Second-person
ChaLearn MMGR [43]	13,858	20	RGB-D	Sign language	Second-person
ChaLearn IsoGD [44]	47,933	249	RGB-D	Sign language, helicopter and traffic signal, pantomimes and symbolic gestures, body language	Second-person

3. Research gap, objective and scope

As illustrated in the literature review, most of the scenes in existing datasets designed for testing gesture recognition methods are indoor environments. The outdoor environments were considered in few of them (e.g. EgoGesture [16] and ChaLearn IsoGD [44] dataset), where the video clips of hand gestures were typically recorded when subjects were sitting or standing still. They hardly contain the outdoor scenes with large background variations and the inference of moving objects (e.g. people and vehicles). As a result, these test scenarios do not fully represent construction environments, which are more complicated and filled with machines, buildings, workers, etc. Also, construction activities typically occur under different weather conditions (e.g. sunny and cloudy days). Signalmen are always moving and making hand signals simultaneously in the construction field.

So far, it is still not clear how exiting recognition methods perform when being used to capture and interpret hand signals automatically in the construction domain. The main objective of this paper is to fill this gap. It investigates the feasibility of hand signal recognition with computer vision technologies in construction fields. The study includes two

components. First, a new hand gesture dataset for instructing tower crane operations on construction sites was created. Tower crane operations are chosen as the study object due to their important role in construction projects, especially constructing high buildings. Then, two state-of-the-art hand gesture recognition methods were selected and tested with the new dataset. The selection was mainly based on the findings from existing comparison studies of recognition methods. The performance of both methods was further measured in term of classification accuracy and inference time to provide an in-depth analysis on the benefits and limitations of existing vision-based hand gesture recognition in construction scenarios.

4. Feasibility study design

4.1. Scenes design for dataset creation

To capture the characteristics of construction site environments, several factors were considered including weather/environmental conditions, motions in the background, the way to make hand signals, etc. Following these factors, a total of 7 scenes have been designed. Three of them are indoors and the other four are outdoors. The indoor scenes are created as follows. In the first indoor scene, the subject who makes hand signals was requested to sit in a chair under a static but cluttered background. Then, the subject was requested to move when making hand signals as the second scene. In the third indoor scene, the subject was moving and making hand signals, and his or her background was cluttered with other moving persons. The outdoor scenes are classified into two categories: two of them are under sunny conditions and the other two are under cloudy conditions. The subjects in all these four scenes were moving and making hand signals with or without background motions. Table 3 summarizes the characteristics of all the designed scenes mentioned above.

The hand signals made by the subject in each scene are those commonly seen on construction sites. For example, tower cranes are the most frequently shared resources [45,46], which are mainly used for lifting heavy things and transporting them to other places. Hand signals for directing tower crane operations were selected here. According to the American Society of Mechanical Engineering (ASME) [4] and National Commission for the Certificate of Crane Operations (NCCCO) [5], there are 11 classes of hand signals that can be used for signalman to instruct tower crane operations, as indicted in Table 4. In addition, the hand signals in each scene were recorded in two modalities (RGB and depth) under a second-person view, where a camera is kept a short distance towards a subject and the subject is asked to perform hand signals to interact intentionally with the camera.

4.2. Hand gesture recognition & evaluation metrics

Based on the findings from existing hand gesture recognition studies [16,17,20,33], 3D-CNN based networks illustrated an excellent spatio-temporal learning ability. Thus, in this study, the selection of hand gesture recognition methods for testing is focused on state-of-the-art 3D-CNN based networks. Specifically, two networks, i.e., RsNeXt-101 and Res3D + ConvLSTM+MobileNet are considered, since they illustrated higher recognition accuracy than other 3D-CNN networks, such as C3D, ResNet-50, Res3D, and C3D + LSTM+RSTTM [18,19,33]. The robustness of ResNeXt-101 and Res3D + ConvLSTM+MobileNet architectures have been proven by various visual recognition tasks [18,31,33,47] and by non-visual tasks involving speech recognition [48,49] and language processing [50,51].

Table 3
The characteristics of the designed scenes.

No.	Scene	Subject status	Weather conditions	Background conditions
1	Indoor	Sitting on a chair	–	Static
2	Indoor	Moving	–	Static
3	Indoor	Moving	–	Dynamic
4	Outdoor	Moving	Sunny	Static
5	Outdoor	Moving	Sunny	Dynamic
6	Outdoor	Moving	Cloudy	Static
7	Outdoor	Moving	Cloudy	Dynamic

RsNeXt-101 refers to a 101-layer 3D-CNN constructed by repetitive ResNeXt building blocks that aggregate a set of transformations with the same topology. Table 5 shows the specific architecture of ResNeXt-101 used for tests in this study. Res3D + ConvLSTM+MobileNet is an integrated architecture of 3D-CNN (Res3D) and Convolutional LSTM layers. Fig. 1 shows an overview of their architectures. More details of both networks could be found in the work of [33,47,52].

The recognition performance will be evaluated in terms of confusion matrix, gesture classification accuracy and inference time on the test set. The confusion matrix is a specific table layout that allows visualization of the performance of the method. Each row i of the matrix represents the predicted class while each column j means the actual class. The element (i, j) of the table refers to the percentage of the actual class j which is predicted as the predicted class i. The classification accuracy is defined as the percentage of correctly labeled gesture samples by the recognition method. This accuracy information will be further measured under different conditions to represent the robustness of a gesture recognition method. Inference time refers to the processing time of using a trained model to make a prediction. Although the inference time may vary with different computer hardware configurations, it provides an idea of how fast the recognition could be made.

5. Results

5.1. Datasets

To create the dataset, a ZED 2 stereo camera [53] is selected as a recording device. The camera could capture the video clips under the RGB and depth modalities. The maximum resolution of the videos could reach up to 2208 × 1242 pixels at 15 frames per second (fps). When capturing the hand gestures into a video clip, a gesture list with a random selection of 5 gestures is generated first. Then, the subject was asked to continuously perform these gestures in the list at different locations to make sure the gestures appear in different video regions.

A total of 364 RGB-D video clips were collected which are equivalent to more than 426,602 frames in each modality. Among them, there are 1820 gesture samples which are distributed in 7 scenes. Each hand signal category consists of 165 samples on average. The average length of a gesture is 110 frames. The minimum and maximum gesture lengths are 21 and 322 frames separately. The details of the dataset are listed in Table 6. Examples of the collected data were collected from real construction sites, as shown in Fig. 2. As indicated in Fig. 3, the subject may be self-occluded or partially occluded during the collection process. The start and end frame indices of the subject's gesture in each video clip are manually labeled as shown in Fig. 4. The frames are further cropped to smaller ones which only contain the regions of subject. Taking Fig. 4 as an example, the resolution of the original frame is 2208 × 1242 while it becomes 420 × 868 after cropping.

5.2. Recognition performance of the methods

5.2.1. Implementation

The recognition methods have been implemented on an Ubuntu Linux 64-bit operating system with the support of the Pytorch [54] and Tensorflow [55] platforms. Both platforms provide the critical algorithms, functions, and tools required for the methods. The hardware configuration includes an Intel® Core™ i7-4820K CPU (Central Processing Unit) @ 3.70 GHz, a 32 GB memory, and an NVIDIA Titan Xp DDR5X @ 12.0 GB GPU (Graphic Processing Unit).

Table 4
Hand signals for instructing tower crane operations adapted from [4,5].

No.	Hand signal	Examples	No.	Hand signal	Examples
1	Hoist		7	Dog everything	
2	Lower		8	Move slowly	
3	Tower travel		9	Swing right	
4	Trolley travel right		10	Swing left	
5	Trolley travel left		11	Emergency stop	
6	Stop				

Table 5
The architecture of ResNeXt-101.

Layer name	Conv1	Conv2_x	Conv3_x	Conv4_x	Conv5_x	–
Output size	112 × 112	56 × 56	28 × 28	14 × 14	7 × 7	1 × 1
ResNeXt-101	Conv(3 × 7 × 7) stride (1,2,2)	N: 3 F: 128	N: 24 F: 256	N: 36 F: 512	N: 3 F: 1024	Average pooling, fc layer with softmax

Note: F – the number of feature channels and N - the number of blocks.

5.2.2. Training

In order to train and test gesture recognition methods, the dataset is randomly split into the training subset (60%), validation subset (20%) and testing subset (20%). The training subset includes 210 video clips and 1050 gesture samples for the training of the network parameters in the gesture recognition methods. The validation subset includes 70 video clips and 350 gesture samples for providing frequent evaluations of the recognition methods while tuning the hyperparameters of the methods. The test subset includes 84 video clips and 420 gesture samples which were used to test the final recognition performance of the methods. About 17% of test video clips contain the background scenarios which are never seen in the training and validation sets.

The number of parameters for 3D-CNNs is much more than 2D-CNNs, which typically requires more training data to prevent underfitting. Here, the transfer learning strategy is adopted. Both ResNeXt-101 and Res3D + ConvLSTM+MobileNet are pretrained firstly using the Jester dataset [41], which is the largest hand gesture dataset publicly available. However, the Jester dataset does not include any hand signals related to construction. The dataset collected in this study is used to fine-tune both networks to increase the accuracy of recognizing hand signals

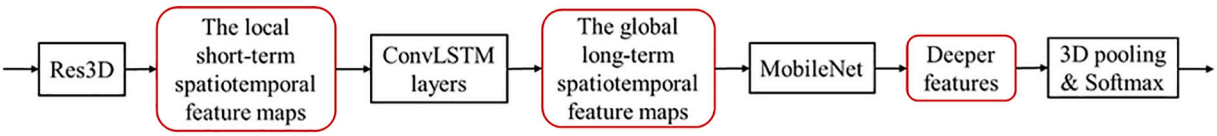


Fig. 1. The architecture of Res3D + ConvLSTM+MobileNet.

Table 6
Dataset configurations.

		Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Scene 6	Scene 7
Duration (s)		2170.7	3909.1	3985.7	4291.1	4797.1	4383.8	4902.7
# of total frames		32,560	58,637	59,785	64,366	71,956	65,757	73,541
Gesture category (# of samples/# of frames)	Hoist	23/1713	22/2427	22/2836	26/2512	23/2916	24/2874	23/3170
	Lower	21/1415	22/2354	22/2741	24/2485	23/2939	24/2617	25/3103
	Tower travel	25/1561	22/2334	23/2706	24/2316	23/2891	24/2277	24/2988
	Trolley travel right	22/1426	23/2649	22/2645	24/2524	23/3057	25/2525	25/3432
	Trolley travel left	24/1880	23/2732	23/2681	26/2747	24/3154	23/2676	25/3169
	Stop	23/1483	23/2451	23/2519	24/2553	26/2916	24/2704	24/3058
	Dog everything	24/1433	23/2460	23/2624	26/2675	25/3314	24/3129	22/2881
	Move slowly	21/1622	23/2519	23/2619	22/2374	25/3248	25/2540	24/3022
	Swing right	23/1373	23/2671	23/2712	24/2811	25/3083	27/2800	24/3584
	Swing left	22/1458	23/2511	23/2845	25/2638	24/2858	25/2847	24/2997
	Emergency stop	22/2018	23/2464	23/2799	25/2851	24/3509	25/2773	25/3501

Fig. 2. The examples of the collected data (top: RGB; bottom: depth).

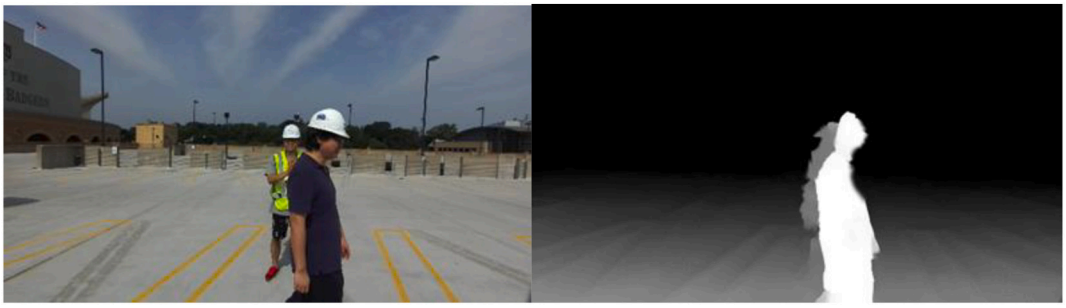


Fig. 3. The subject being self-occluded or partially occluded (left: RGB; right: depth).

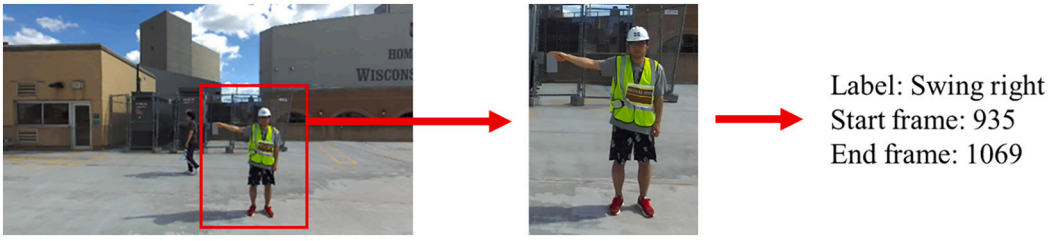


Fig. 4. The manual labeling and cropping process.

in construction and meanwhile shorten the training durations required.

Table 7 summarized the parameters set for the training. The specific training process is conducted as follows. The learning rate and the batch size are initially set as large as possible. When the training loss is steady, the learning rate is reduced with a fixed decay factor. Stochastic gradient descent (SGD) with Nesterov momentum of 0.9, damping factor of 0.9, and weight decay of 0.001 is employed as the optimizer. Moreover, all images of hand gesture samples are randomly cropped with a spatial size of 112 × 112 as the inputs for the data augmentation purpose. Fig. 5 shows the loss reduction along with the training progress. The training loss is higher than the validation loss since the data augmentation process during the training increases the diversity of the training subset and results in more learning difficulties. Taking the depth modality as an example, the training for ResNeXt-101 is completed after 20 epochs and achieves the best validation performance at epoch 18. For Res3D + ConvLSTM+MobileNet, the training is finished within 240 epochs and obtains the highest validation accuracy at epoch 216.

5.2.3. Experimental results

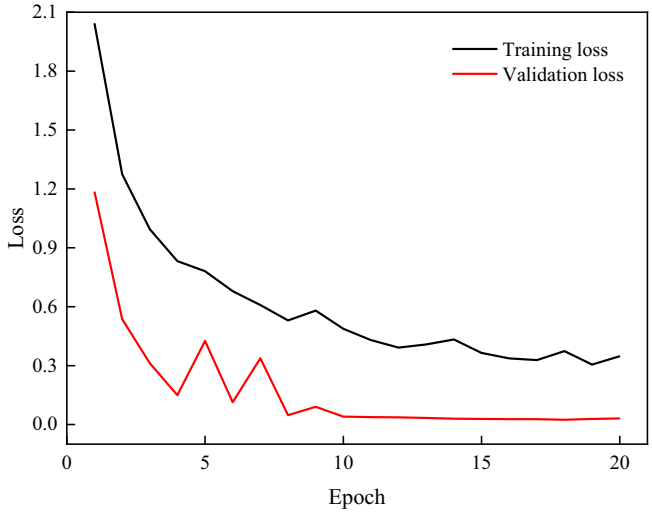
The overall classification accuracy of ResNeXt-101 achieves 93.3% while it is 84.8% for Res3D + ConvLSTM+MobileNet. Tables 8 and 9 presented the confusion matrix of ResNeXt-101 under the RGB and depth modality, separately. Dog everything gesture in RGB modality and lower, swing right, emergency stop gestures in depth modality obtain the highest accuracy (100%). Table 10 indicated the classification accuracy and inference time of the ResNeXt-101 network in different scenes. The network achieves the classification accuracy of 91.9% and the average inference time of 0.26 s/gesture when processing the RGB data. The classification accuracy is 93.3% and the average inference time is 0.17 s/gesture when processing the depth data. In each modality, a higher classification accuracy happened on the recognition of hand gestures in the indoor scenes, and meanwhile, it required less inference time. Take the depth modality for an example. The average classification accuracy for the indoor and outdoor scenes is 96.7% and 91.5%, separately. The average inference time for the indoor and outdoor scenes is 0.13 s/gesture and 0.20 s/gesture, respectively.

Tables 11 and 12 showed the confusion matrix of Res3D + ConvLSTM+MobileNet under the RGB and depth modality, respectively. Dog everything gesture achieves the highest accuracy (91.9%) in RGB modality while the accuracy of emergency stop gesture is the highest (87.5%) in the depth modality. Table 13 presented the classification accuracy and inference time of Res3D + ConvLSTM+MobileNet in both RGB and depth modalities. Specifically, the classification accuracy is 84.5% and the average inference time is 4.82 s/gesture when processing the RGB data. The network achieves the classification accuracy of 84.8% and the average inference time of 4.60 s/gesture when processing the depth data. In each modality, the network achieves a higher classification accuracy in the indoor scenes compared to the outdoor scenes. Taking the depth modality as an example, the classification accuracy keeps 90.0% in the indoor scenes while it is 81.9% in the outdoor scenes. Besides, more inference time is needed for the outdoor scenes under both modalities. For example, the average inference time for the indoor and outdoor scenes is 4.72 s/gesture and 4.89 s/gesture, respectively, in the RGB modality.

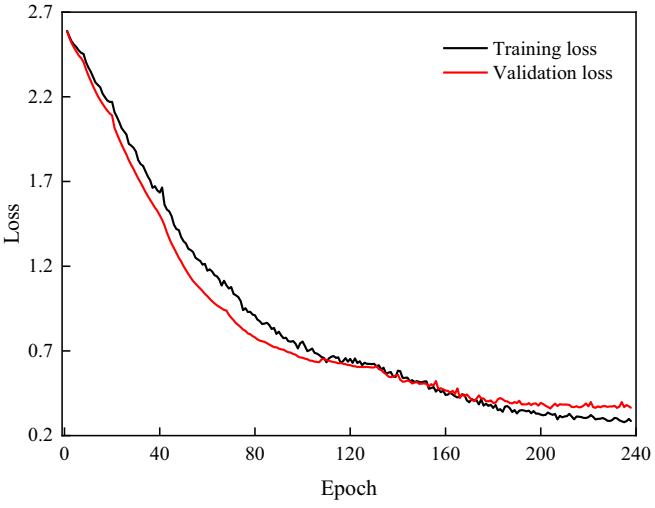
Table 14 compared the inference time of each hand signal category under RGB and depth modalities. As for ResNeXt-101, the average inference time when processing the RGB and depth data is 0.26 s/gesture and 0.17 s/gesture, separately. Compared to ResNeXt-101, it required more inference time for Res3D + ConvLSTM+MobileNet. The average inference time in RGB and depth modality is 4.82 s/gesture and 4.60 s/gesture, respectively. Among all the hand signals, an emergency stop is the gesture which needed the most inference time to be recognized. Take the RGB modality as an example. The inference time of emergency stop for ResNeXt-101 and Res3D + ConvLSTM+MobileNet is 0.43 s/gesture and 6.48 s/gesture, separately.

Table 7
Network parameters.

Network	Learning rate	Step size	Batch size	Length of input video frames
ResNeXt-101	0.01	15	20	32
Res3D + ConvLSTM+MobileNet	0.001	10	20	32



(a) ResNeXt-101



(b) Res3D+ConvLSTM+MobileNet

Fig. 5. The loss reduction along with the training progress for depth modality.

6. Discussion

It was noted by comparing Tables 10 and 13 that the recognition performance of ResNeXt-101 was superior to Res3D + ConvLSTM+MobileNet under both RGB and depth modalities. The higher classification accuracy of ResNeXt-101 indicates that ResNeXt-101 is more feasible to model the spatiotemporal learning tasks in the dataset created for construction gestures. In the meantime, ResNeXt-101 required much less inference time due to its different strategies adopted to form a deep learning network architecture. Res3D +

Table 8
The confusion matrix of ResNeXt-101 under the RGB modality (%).

	Actual										
Prediction	Hoist	Lower	Tower travel	Trolley travel right	Trolley travel left	Stop	Dog everything	Move slowly	Swing right	Swing left	Emergency stop
Hoist	90.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Lower	0.0	97.8	0.0	0.0	0.0	7.5	0.0	0.0	0.0	0.0	0.0
Tower travel	0.0	0.0	92.9	0.0	0.0	0.0	0.0	3.2	0.0	0.0	0.0
Trolley travel right	4.7	0.0	0.0	88.1	0.0	5.0	0.0	0.0	0.0	0.0	0.0
Trolley travel left	0.0	0.0	0.0	0.0	97.7	0.0	0.0	0.0	0.0	0.0	0.0
Stop	0.0	0.0	0.0	2.4	0.0	80.0	0.0	0.0	2.5	0.0	0.0
Dog everything	0.0	0.0	3.6	4.8	0.0	0.0	100.0	9.7	0.0	3.4	0.0
Move slowly	0.0	0.0	0.0	2.4	0.0	0.0	0.0	77.4	0.0	0.0	2.5
Swing right	0.0	2.2	0.0	0.0	0.0	2.5	0.0	3.2	92.5	0.0	0.0
Swing left	0.0	0.0	3.6	0.0	2.3	2.5	0.0	6.5	5.0	96.6	2.5
Emergency stop	4.7	0.0	0.0	2.4	0.0	2.5	0.0	0.0	0.0	0.0	95.0

Table 9
The confusion matrix of ResNeXt-101 under the depth modality (%).

	Prediction										
Actual	Hoist	Lower	Tower travel	Trolley travel right	Trolley travel left	Stop	Dog everything	Move slowly	Swing right	Swing left	Emergency stop
Hoist	95.3	0.0	0.0	9.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Lower	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tower travel	0.0	0.0	85.7	0.0	0.0	0.0	2.7	0.0	0.0	0.0	0.0
Trolley travel right	2.3	0.0	0.0	88.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Trolley travel left	0.0	0.0	0.0	0.0	95.5	0.0	0.0	0.0	0.0	0.0	0.0
Stop	0.0	0.0	0.0	0.0	0.0	87.5	0.0	0.0	0.0	0.0	0.0
Dog everything	0.0	0.0	10.7	2.4	0.0	0.0	91.9	3.2	0.0	3.4	0.0
Move slowly	0.0	0.0	0.0	0.0	2.3	0.0	0.0	80.6	0.0	0.0	0.0
Swing right	0.0	0.0	3.6	0.0	0.0	12.5	2.7	12.9	100.0	0.0	0.0
Swing left	2.3	0.0	0.0	0.0	2.3	0.0	2.7	0.0	0.0	96.6	0.0
Emergency stop	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.2	0.0	0.0	100.0

Table 10
The classification accuracy and inference time of ResNeXt-101.

		RGB		Depth	
		Classification accuracy (%)	Inference time (s/gesture)	Classification accuracy (%)	Inference time (s/gesture)
Indoor	Scene 1	94.0	0.11	94.0	0.09
	Scene 2	98.0	0.24	96.0	0.15
	Scene 3	100.0	0.23	100.0	0.15
	Average	97.3	0.19	96.7	0.13
Outdoor	Scene 4	87.7	0.27	89.2	0.17
	Scene 5	81.5	0.32	89.2	0.21
	Scene 6	97.1	0.29	95.7	0.18
	Scene 7	88.6	0.36	91.4	0.23
	Average	88.9	0.31	91.5	0.20
Average		91.9	0.26	93.3	0.17

ConvLSTM+MobileNet combined ConvLSTM layers with Res3D network for performance improvement [33] but it increased the network complexity. On the contrary, ResNeXt-101 refined the original ResNet block by introducing a new dimension called "cardinality" to provide a new way of adjusting the network capacity [47]. It prevents the network from going deeper or wider to increase the recognition performance.

As for the comparison between different modalities, the classification accuracy on RGB data is slightly better than that on depth data in the indoor environment but becomes worse in the outdoor environment. It may be due to the following reasons. Generally, the depth data filters out the background motion, and allows the networks to focus more on the hand motion. However, the formation of the depth image is easy to be impacted by illumination conditions. As shown in Fig. 6, the depth information is not clearly identified in low-light environments of the indoor scenes. Besides, it required more inference time to process RGB data. RGB images generally contain more information for recognition since they include three channels while depth images have only one channel. It is possible to fuse both modalities to further improve recognition accuracy. Taking ResNeXt-101 as an example, the network achieved a higher overall classification accuracy (93.8%) but needed a longer inference time (0.38 s/gesture) in the RGB-D modality, as shown in Table 15.

When analyzing the results in different scenes, it can be found that the classification accuracy in indoor scenes is generally higher than that in outdoor scenes. Besides, the network achieves better performance in cloudy conditions compared to sunny conditions. These indicate the impacts of different illumination conditions. The outdoor environmental lights are diverse, which may affect the RGB and depth data more easily [16]. The strong illumination conditions in the sunny days have negative impacts on the quality of input images. The images with poor illumination quality are easier to cause false recognition of computer vision technologies.

Compared with other hand signals, emergency stop, swing left, swing right, and stop required more inference time. The length of inference time may be related to the input size of images. It usually takes more time for a trained network to predict if the input size of data is large. Fig. 7 exhibited the average input pixel points per image in different hand signal categories. The input images of the signals of emergency stop, swing left, swing right, and stop generally occupy more pixel points because a signalman needed to swing his or her arms when conducting these gestures.

Overall, a high classification accuracy (93.3%) and a short inference time (0.17 s/gesture) could be achieved with ResNeXt-101 network under the depth modality. The high accuracy indicates that computer

Table 11
The confusion matrix of Res3D + ConvLSTM+MobileNet under the RGB modality (%).

	Prediction										
Actual	Hoist	Lower	Tower travel	Trolley travel right	Trolley travel left	Stop	Dog everything	Move slowly	Swing right	Swing left	Emergency stop
Hoist	81.4	2.2	0.0	4.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Lower	0.0	84.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tower travel	0.0	0.0	82.1	0.0	0.0	0.0	5.4	9.7	0.0	0.0	0.0
Trolley travel right	11.6	0.0	0.0	85.7	0.0	7.5	0.0	0.0	0.0	0.0	0.0
Trolley travel left	0.0	6.5	0.0	0.0	84.1	0.0	0.0	0.0	0.0	3.4	0.0
Stop	0.0	2.2	0.0	0.0	0.0	82.5	0.0	0.0	10.0	0.0	7.5
Dog everything	0.0	0.0	10.7	4.8	0.0	0.0	91.9	12.9	0.0	10.3	0.0
Move slowly	0.0	0.0	0.0	4.8	0.0	0.0	2.7	77.4	0.0	0.0	0.0
Swing right	0.0	4.3	0.0	0.0	0.0	0.0	0.0	0.0	85.0	0.0	5.0
Swing left	0.0	0.0	7.1	0.0	11.4	2.5	0.0	0.0	0.0	86.2	0.0
Emergency stop	7.0	0.0	0.0	0.0	4.5	7.5	0.0	0.0	5.0	0.0	87.5

Table 12
The confusion matrix of Res3D + ConvLSTM+MobileNet under the depth modality (%).

	Prediction										
Actual	Hoist	Lower	Tower travel	Trolley travel right	Trolley travel left	Stop	Dog everything	Move slowly	Swing right	Swing left	Emergency stop
Hoist	86.0	0.0	0.0	11.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Lower	0.0	87.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tower travel	0.0	0.0	78.6	0.0	0.0	0.0	8.1	9.7	0.0	0.0	0.0
Trolley travel right	11.6	0.0	0.0	83.3	0.0	10.0	0.0	0.0	0.0	0.0	0.0
Trolley travel left	0.0	8.7	0.0	0.0	86.4	0.0	0.0	0.0	0.0	6.9	0.0
Stop	0.0	0.0	0.0	0.0	2.3	85.0	0.0	0.0	7.5	0.0	7.5
Dog everything	2.3	0.0	14.3	0.0	0.0	0.0	86.5	6.5	0.0	10.3	0.0
Move slowly	0.0	0.0	7.1	0.0	0.0	0.0	5.4	83.9	0.0	0.0	0.0
Swing right	0.0	0.0	0.0	0.0	2.3	0.0	0.0	0.0	82.5	0.0	5.0
Swing left	0.0	4.3	0.0	4.8	9.1	0.0	0.0	0.0	2.5	82.8	0.0
Emergency stop	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	7.5	0.0	87.5

Table 13
The classification accuracy and inference time of Res3D + ConvLSTM+MobileNet.

		RGB		Depth	
		Classification accuracy (%)	Inference time (s/gesture)	Classification accuracy (%)	Inference time (s/gesture)
Indoor	Scene 1	96.0	4.64	90.0	4.56
	Scene 2	92.0	4.77	82.0	4.56
	Scene 3	94.0	4.76	98.0	4.55
	Average	94.0	4.72	90.0	4.56
Outdoor	Scene 4	78.5	4.84	81.5	4.58
	Scene 5	76.9	4.91	83.1	4.65
	Scene 6	82.9	4.89	82.9	4.63
	Scene 7	78.6	4.93	80.0	4.68
	Average	79.3	4.89	81.9	4.64
Average		84.5	4.82	84.8	4.60

Table 14
The inference time of ResNeXt-101 and Res3D + ConvLSTM+MobileNet (s/gesture).

	ResNeXt-101		Res3D + ConvLSTM+MobileNet	
	RGB	Depth	RGB	Depth
Hoist	0.22	0.13	3.75	3.58
Lower	0.19	0.15	6.45	6.17
Tower travel	0.27	0.16	5.62	5.37
Trolley travel right	0.17	0.11	3.39	3.28
Trolley travel left	0.20	0.12	3.71	3.54
Stop	0.25	0.18	4.35	4.18
Dog everything	0.26	0.14	4.31	4.05
Move slowly	0.28	0.18	5.68	5.47
Swing right	0.33	0.19	4.76	4.48
Swing left	0.30	0.18	4.56	4.35
Emergency stop	0.43	0.38	6.48	6.17
Average	0.26	0.17	4.82	4.60

vision technologies are capable of ensuring safety when they are adopted to recognize hand signals on construction sites. The short duration allows the construction equipment to make actions in real time based on the recognized hand signal. The recognition results illustrate the feasibility and potential of employing computer vision technologies to automate the hand signal recognition in construction field.

7. Conclusions and future work

On construction sites, it is common for workers to rely on hand signals to communicate and express thoughts due to their simple but effective nature. However, hand signals may not always be captured timely or interpreted correctly in the fields, which easily leads to construction errors and even accidents. This paper presented a feasibility study on investigating whether the recognition of hand signals could be automated with computer vision technologies in the construction field through creating a new dataset with 11 classes of hand signals in construction and evaluating two state-of-the-art recognition networks, ResNeXt-101 and Res3D + ConvLSTM+MobileNet in terms of confusion matrix, accuracy and inference time. The results indicated a high classification accuracy (e.g. 93.3%) and a short inference time (e.g. 0.17 s/gesture) could be achieved and illustrated the feasibility of employing computer vision technologies to automate the hand signal recognition on construction sites.

Future work will focus on two aspects. First, more construction hand signals will be included into the dataset to make the training and testing

Fig. 6. The examples of the RGB and depth data in scene 2 (left: RGB; right: depth).

Table 15
The recognition performance of ResNeXt-101 in RGB-D modality.

		RGB-D	
		Classification accuracy (%)	Inference time (s/gesture)
Indoor	Scene 1	96.0	0.16
	Scene 2	98.0	0.34
	Scene 3	100.0	0.33
	Average	98.0	0.28
Outdoor	Scene 4	95.4	0.39
	Scene 5	87.7	0.47
	Scene 6	95.7	0.42
	Scene 7	87.1	0.54
	Average	91.5	0.46
Average		93.8	0.38

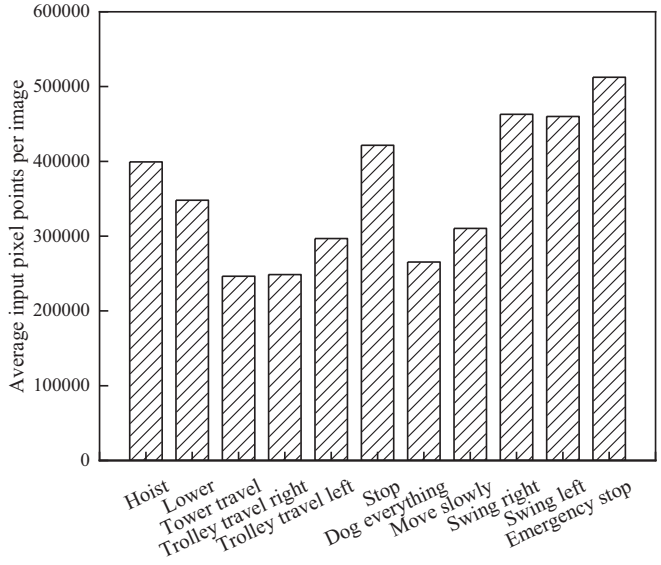


Fig. 7. The average input pixel points per image in different hand signal categories.

of different types of construction hand signal detectors and classifiers. Second, it will investigate what will be an effective way to transmit the signal meanings to the corresponding receivers, after the automatic recognition of hand signals.

Declaration of Competing Interest

None.

Acknowledgement

This paper is based in part upon the work supported by the Wisconsin Alumni Research Foundation (WARF) under Project No. AAD5524 and the M.A. Mortenson Company. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of WARF or Mortenson.

References

[1] The Off-highway Plant and Equipment Research Centre, Hand Signals for When Excavations are Used as Cranes: A Voluntary Code of Practice, Birmingham City University, 2019.
[2] P. Kines, L.P.S. Andersen, S. Spangenberg, K.L. Mikkelsen, J. Dyreborg, D. Zohar, Improving construction site safety through leader-based verbal safety communication, J. Saf. Res. (2010), https://doi.org/10.1016/j.jsr.2010.06.005.
[3] R.L. Neitzel, N.S. Seixas, K.K. Ren, A review of crane safety in the construction industry, Appl. Occup. Environ. Hyg. (2001), https://doi.org/10.1080/10473220127411.
[4] The American Society of Mechanical Engineers, Safety Standard for Cableways, Cranes, Derricks, Hoists, Hooks, Jacks, and Slings, 2012.
[5] National Commission for the Certification of Crane Operators, Signalperson Reference Manual, 2014.
[6] P.D. Bust, A.G.F. Gibb, S. Pink, Managing construction health and safety: migrant workers and communicating safety messages, Saf. Sci. (2008), https://doi.org/10.1016/j.ssci.2007.06.026.
[7] P.E. Hagan, J.F. Montgomery, J.T. O'Reilly, Accident Prevention Manual for Business & Industry: Engineering & Technology, National Safety Council, 2015.
[8] IONAPEX, Safety Talk Report. http://www.ionapex.com/safety-talks/all-topics/mistaken-signals.shtml, 2013.
[9] ENFORM, D8 Bulldozer Contact with Surveyor on ATV. http://www.energysafetycanada.com/files/safety-alerts/SA05-13-ATV-Bulldozer.pdf, 2013.
[10] Reakes, Traffic Signal Worker Thrown From Bucket In Stamford. https://dailyvoice.com/connecticut/stamford/news/traffic-signal-worker-thrown-from-bucket-in-stamford/732557/, 2018.
[11] P. Kumar, H. Gauba, P. Pratim Roy, D. Prosad Dogra, A multimodal framework for sensor based sign language recognition, Neurocomputing (2017), https://doi.org/10.1016/j.neucom.2016.08.132.
[12] T.Y. Pan, C.Y. Chang, W.L. Tsai, M.C. Hu, OrsNet: A hybrid neural network for official sports referee signal recognition, in: MMSports 2018 - Proc. 1st Int. Work. Multimed. Content Anal. Sport. Co-Located with MM 2018, 2018, https://doi.org/10.1145/3265845.3265849.
[13] F. Guo, J. Tang, X. Wang, Gesture recognition of traffic police based on static and dynamic descriptor fusion, Multimed. Tools Appl. (2017), https://doi.org/10.1007/s11042-016-3497-9.
[14] E. Ohn-Bar, M.M. Trivedi, Hand gesture recognition in real time for automotive interfaces: a multimodal vision-based approach and evaluations, IEEE Trans. Intell. Transp. Syst. (2014), https://doi.org/10.1109/TITS.2014.2337331.
[15] H. Wang, D. Oneata, J. Verbeek, C. Schmid, A robust and efficient video representation for action recognition, Int. J. Comput. Vis. (2016), https://doi.org/10.1007/s11263-015-0846-5.
[16] Y. Zhang, C. Cao, J. Cheng, H. Lu, EgoGesture: a new dataset and benchmark for egocentric hand gesture recognition, IEEE Trans. Multimed. (2018), https://doi.org/10.1109/TMM.2018.2808769.
[17] M. Kurmanji, F. Ghaderi, Hand gesture recognition from RGB-D data using 2D and 3D convolutional neural networks: a comparative study, J. AI Data Min. 8 (2020) 177–188.
[18] O. Köpüklü, A. Gunduz, N. Kose, G. Rigoll, Real-time hand gesture detection and classification using convolutional neural networks, in: Proc. - 14th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2019, 2019, https://doi.org/10.1109/FG.2019.8756576.
[19] G. Benitez-Garcia, J. Olivares-Mercado, G. Sanchez-Perez, K. Yanai, I.P.N. Hand, A Video Dataset and Benchmark for Real-Time Continuous Hand Gesture Recognition, 2020.
[20] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, J. Kautz, Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, 2016, https://doi.org/10.1109/CVPR.2016.456.
[21] W. Qi, H. Su, A. Aliverti, A Smartphone-Based Adaptive Recognition and Real-Time Monitoring System for Human Activities, IEEE Trans, Human-Machine Syst, 2020, https://doi.org/10.1109/THMS.2020.2984181.
[22] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, E. De Momi, Deep neural network approach in robot tool dynamics identification for bilateral teleoperation, IEEE Robot. Autom. Lett. (2020), https://doi.org/10.1109/LRA.2020.2974445.
[23] H. Su, Y. Hu, H.R. Karimi, A. Knoll, G. Ferrigno, E. De Momi, Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results, Neural Networks (2020), https://doi.org/10.1016/j.neunet.2020.07.033.

[24] F. Hu, P. He, S. Xu, Y. Li, C. Zhang, FingerTrak: continuous 3D hand pose tracking by deep learning hand silhouettes captured by miniature thermal cameras onwrist, in: Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol, 2020, https://doi.org/10.1145/3397306.
[25] H. Su, S.E. Ovur, X. Zhou, W. Qi, G. Ferrigno, E. De Momi, Depth vision guided hand gesture recognition using electromyographic signals, Adv. Robot. (2020), https://doi.org/10.1080/01691864.2020.1713886.
[26] J. Wan, G. Guo, S.Z. Li, Explore efficient local features from RGB-D data for one-shot learning gesture recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2016), https://doi.org/10.1109/TPAMI.2015.2513479.
[27] J. Singha, K. Das, Recognition of Indian sign language in live video, Int. J. Comput. Appl. (2013), https://doi.org/10.5120/12174-7306.
[28] X. Wang, M. Xia, H. Cai, Y. Gao, C. Cattani, Hidden-Markov-models-based dynamic hand gesture recognition, Math. Probl. Eng. (2012), https://doi.org/10.1155/2012/986134.
[29] L. Lin, Y. Cong, Y. Tang, Hand gesture recognition using RGB-D cues, in: 2012 IEEE Int. Conf. Inf. Autom 2012, ICIA, 2012, https://doi.org/10.1109/ICInfA.2012.6246824.
[30] O.K. Oyedotun, A. Khashman, Deep learning in vision-based static hand gesture recognition, Neural Comput. Appl. (2017), https://doi.org/10.1007/s00521-016-2294-8.
[31] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, X. Cao, Multimodal gesture recognition based on the ResC3D network, in: Proc. - 2017 IEEE Int. Conf. Comput. Vis. Work 2017, ICCVW, 2017, https://doi.org/10.1109/ICCVW.2017.360.
[32] C. Cao, Y. Zhang, Y. Wu, H. Lu, J. Cheng, Egocentric gesture recognition using recurrent 3D convolutional neural networks with spatiotemporal transformer modules, in: Proc. IEEE Int. Conf. Comput. Vis, 2017, https://doi.org/10.1109/ICCV.2017.406.
[33] L. Zhang, L. Mei, S.A.A. Shah, G. Zhu, P. Shen, M. Bennamoun, Attention in convolutional LSTM for gesture recognition, Adv. Neural Inf. Process. Syst. (2018) 1957–1966.
[34] J. Wu, P. Ishwar, J. Konrad, Two-Stream CNNs for gesture-based verification and identification: learning user style, in: IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work, 2016, https://doi.org/10.1109/CVPRW.2016.21.
[35] J. Huang, W. Zhou, Q. Zhang, H. Li, W. Li, Video-based sign language recognition without temporal segmentation, in: 32nd AAAI Conf. Artif. Intell, AAAI, 2018.
[36] T. Starner, J. Weaver, A. Pentland, Real-time american sign language recognition using desk and wearable computer based video, IEEE Trans. Pattern Anal. Mach. Intell. (1998), https://doi.org/10.1109/34.735811.
[37] L. Baraldi, F. Paci, G. Serra, L. Benini, R. Cucchiara, Gesture recognition in ego-centric videos using dense trajectories and hand segmentation, in: IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work, 2014, https://doi.org/10.1109/CVPRW.2014.107.
[38] T.K. Kim, R. Cipolla, Canonical correlation analysis of video volume tensors for action categorization and detection, IEEE Trans. Pattern Anal. Mach. Intell. (2009), https://doi.org/10.1109/TPAMI.2008.167.
[39] L. Liu, L. Shao, Learning discriminative representations from RGB-D video data, in: IJCAI Int. Jt. Conf. Artif. Intell, 2013.
[40] S. Ruffieux, D. Lalanne, E. Mugellini, ChAirGest: A challenge for multimodal mid-air gesture recognition for close HCI, in: ICMI 2013 - Proc. 2013 ACM Int. Conf. Multimodal Interact, 2013, https://doi.org/10.1145/2522848.2532590.
[41] J. Materzynska, G. Berger, I. Bax, R. Memisevic, The jester dataset: A large-scale video dataset of human gestures, in: Proc. - 2019 Int. Conf. Comput. Vis. Work 2019, ICCVW, 2019, https://doi.org/10.1109/ICCVW.2019.00349.
[42] A. Kurakin, Z. Zhang, Z. Liu, A real time system for dynamic hand gesture recognition with a depth sensor, in: Eur. Signal Process. Conf, 2012.
[43] S. Escalera, X. Baró, J. Gonzàlez, M.A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H.J. Escalante, J. Shotton, I. Guyon, Chalearn looking at people challenge 2014: Dataset and results, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2015, https://doi.org/10.1007/978-3-319-16178-5_32.
[44] J. Wan, S.Z. Li, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, ChaLearn looking at people RGB-D isolated and continuous datasets for gesture recognition, in: IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work, 2016, https://doi.org/10.1109/CVPRW.2016.100.
[45] M. Al-Hussein, M. Athar Niaz, H. Yu, H. Kim, Integrating 3D visualization and simulation for tower crane operations on construction sites, Autom. Constr. (2006), https://doi.org/10.1016/j.autcon.2005.07.007.
[46] J. Yang, P.A. Vela, J. Teizer, Z.K. Shi, Vision-based crane tracking for understanding construction activity, in: Congr. Comput. Civ. Eng. Proc, 2011, https://doi.org/10.1061/41182(416)32.
[47] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR, 2017, p. 2017, https://doi.org/10.1109/CVPR.2017.634.
[48] C.F. Chen, Q. Fan, N. Mallinar, T. Sercu, R. Feris, Big-little net: An efficient multi-scale feature representation for visual and speech recognition, ArXiv, abs/1807.03848 (2018).
[49] G. Paraskevopoulos, S. Parthasarathy, A. Khare, S. Sundaram, Multiresolution and multimodal speech recognition with transformers, ArXiv (2020). arXiv:2004.14840.
[50] Q. Wang, Y. Huang, W. Jia, X. He, M. Blumenstein, S. Lyu, Y. Lu, FACLSTM: ConvLSTM with focused attention for scene text recognition, Sci. China Inf. Sci. (2020), https://doi.org/10.1007/s11432-019-2713-1.
[51] J.T. Lu, S. Pedemonte, B. Bizzo, S. Doyle, K.P. Andriole, M.H. Michalski, R. Gilberto Gonzalez, S.R. Pomerantz, Deep spine: Automated lumbar vertebral segmentation, disc-level designation, and spinal stenosis grading using deep learning, ArXiv (2018). arXiv:1807.10215.
[52] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[53] Stereolabs, ZED 2-AI Stereo Camera. https://www.stereolabs.com/zed-2/, 2019.
[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: an imperative style, high-performance deep learning library, ArXiv (2019) 8024–8035.
[55] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A system for large-scale machine learning, in: Proc. 12th USENIX Symp. Oper. Syst. Des. Implementation 2016, OSDI, 2016, pp. 265–283.