Automation in Construction 86 (2018) 118–124

Contents lists available at ScienceDirect

Automation in Construction

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

A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory

Lieyun Ding[a,b], Weili Fang[a,b,*], Hanbin Luo[a,b], Peter E.D. Love[c], Botao Zhong[a,b], Xi Ouyang[d]

[a] Dept. of Construction Management, School of Civil Engineering and Mechanics, Huazhong University of Science and Technology, Wuhan, Hubei, China
[b] Hubei Engineering Research Center for Virtual, Safe and Automated Construction, China
[c] Dept. of Civil Engineering, Curtin University, Perth, Western Australia, Australia
[d] School of Electronic Information and communications, Huazhong University of Science and Technology, Wuhan, Hubei, China

ARTICLE INFO

Keywords:
Deep learning
Convolution neural network
Long short-term memory
Unsafe actions
Safety
Video surveillance

ABSTRACT

Computer vision and pattern recognition approaches have been applied to determine unsafe behaviors on construction sites. Such approaches have been reliant on the computation of artificially complex image features that utilize a cumbersome parameter re-adjustment process. The creation of image features that can recognize unsafe actions, however, poses a significant research challenge on construction sites. This due to the prevailing complexity of spatio-temporal features, lighting, and the array of viewpoints that are required to identify an unsafe action. Considering these challenges, a new hybrid deep learning model that integrates a convolution neural network (CNN) and long short-term memory (LSTM) that automatically recognizes workers' unsafe actions is developed. The proposed hybrid deep learning model is used to: (1) identify unsafe actions; (2) collect motion data and site videos; (3) extract the visual features from videos using a CNN model; and (4) sequence the learning features that are enabled by the use of LSTM models. An experiment is used to test the model's ability to detect unsafe actions. The results reveal that the developed hybrid model (CNN + LSTM) is able to accurately detect safe/unsafe actions conducted by workers on-site. The model's accuracy exceeds the current state-of-the-art descriptor-based methods for detecting points of interest on images.

1. Introduction

Ensuring the safety of people is a pervasive and challenging task in construction due the dynamic and complex work conditions that exist on-site [1–3]. Accidents and fatalities during construction have been and remain a worldwide problem. This is despite regulatory reforms, legislation, and efforts by industry associations and extensive research being undertaken to redress this problem [4–7]. According to Heinrich [8], approximately 88% of all accidents that occur during construction materialize as a consequence of unsafe behavior. If unsafe behavior can be reduced or even prevented, then safety performance will naturally improve. According to Fam, et al. [9] unsafe behavior is enacted when an employee does not respect safety rules, standards, procedures, instructions, and specified project criteria. Such actions can adversely influence an employee's performance and/or endanger others within the workplace.

Conventional methods to determine workers' behavior have been predominately based upon observational methods. While such methods may provide useful information, they are time-consuming, labor-intensive and are subjective in nature. Due to these limitations, computer vision technology, which has been used for object recognition [10–12], can be applied to identify workers' unsafe actions on-site [13–18]. Human behavior recognition has been typically based on the use of depth sensors (Kinect™), and collection of motion data from stereo videos that are reconstructed to build a three-dimensional (3D) skeleton model [19–25]. For example, multiple video cameras have been used to monitor the behavior of workers by estimating the positioning of an individual's joints in 3D [20–24]. This method provides a useful way to obtain accurate motion data. But more specifically, it provides the ability to record, model, and analyze the human motions that have resulted from committing an unsafe action. However, monitoring the positioning of workers within a 3D environment may require lengthy computational periods and the depth sensor's line of motion may also be subjected to sensitivities in lighting [26,27].

Against this contextual backdrop, a novel hybrid deep learning model that integrates a convolution neural network (CNN) and long short-term memory (LSTM) to automatically recognize workers' unsafe actions is developed. The hybrid model is used to: (1) identify unsafe actions; (2) collect motion data and site videos; (3) extract visual features from videos using a CNN model; and (4) sequence the learning

* Corresponding author at: Dept. of Construction Management, School of Civil Engineering and Mechanics, Huazhong University of Science and Technology, Wuhan, Hubei, China.
E-mail address: weili_f@hust.edu.cn (W. Fang).

https://doi.org/10.1016/j.autcon.2017.11.002
Received 23 March 2017; Received in revised form 30 October 2017; Accepted 3 November 2017
Available online 14 November 2017
0926-5805/

features that are enabled by the use of LSTM models.

Video cameras are used to collect motion data. Then, a deep learning technique is applied to detect unsafe actions. Deep learning is essentially a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data. The videos of human actions contain spatial and temporal information, and a deep CNN model is trained and learns from multiple frames and spatial features contained within them. The features generated from the CNNs are fed into the LSTM models so that they can learn from peoples' actions over a period of time. The CNN is akin to a feed-forward Neural Network; it has an end-to-end structure with an automatic feature extraction. The LSTM, however, is similar to Recurrent Neural Network (RNN) models, which can enable long-range temporal interval learning to occur. The CNN and LSTM are merged together so that a sequence of feature representations of unsafe acts derived from action videos can be automatically extracted. An experiment is used to demonstrate the effectiveness of the developed deep learning hybrid model.

2. Prior research

The orthodoxy of computer vision has been reliant on extracting handcrafted features from inputs. Hand-crafted feature-based methods usually employ a three-stage procedure, which consists of: (1) extraction; (2) representation; and (3) classification. Image representation that is used to recognize human actions can extract features such as shapes and temporal motions from images. Action recognition features, however, need to contain rich information so that a wide range of them can be identified and analyzed. Techniques that can be used to analyze such features include: (a) classifier tools (e.g. Support Vector Machine (SVM)); (b) temporal state-space models (e.g., Hidden Markov models (HMM); (c) conditional random fields (CRF)); and (d) detection-based methods (e.g., bag-of-words coding). Gong, et al. [27], for example, applied the space–time interest point detector to identify interest points on the images of workers and equipment [28]. Then, using the Histogram of Oriented Gradients (HOG) [29] and Histogram of Optical Flow (HoF) descriptors to determine these interest points. This method, however, is unable to capture a worker's motions that are contained in a video or displayed on an image, which hinders its ability to accurately detect actions.

Research focusing on detecting and recording unsafe actions has tended to be based on the use of depth sensors (Kinect™) or multiple cameras to extract 3D skeleton models of a worker. For example, Han and Lee [20] utilized stereo cameras to collect motion data to construct a 3D skeleton model and used pattern recognition to identify common unsafe actions. Similarly, Liu, et al. [21] used two smartphones as stereo cameras to acquire motion data to extract 3D human skeletons to track people. Alternatively using a depth (RGB-D) sensor, Han, et al. [22] developed a modeling methodology to recognize and classify unsafe behaviors. Yet, the aforementioned methods have been demonstrated in a controlled indoor environment and therefore have not accommodated the nuances of a construction site. In addition, the requirement for lengthy computation periods, low levels of accuracy, the presence of occlusions, and high levels of illumination have stymied their capacity to effectively learn and therefore used in a real-life setting.

The use of CNN + LSTM to examine spatial and temporal information is an area that has received a considerable amount of interest within the field of computer vision [30,31]. For example, an action in a video may span different granularities. Therefore, to better recognize such actions Li et al. [29] modelled each granularity as a single stream by 2D (for frame and motion streams) and 3D (for clip and video streams) using convolutional neural networks (CNNs). In this instance, the CNN is able to learn from spatial and temporal representations. However, to address the issues associated long-term temporal dynamics Li, et al. [30] employed LSTM networks to the frame, motion and clip streams. The use of LSTMs have also been applied to deal with unsegmented videos and improve the training ability of temporal deep learning models to detect activity progression [31].

In consideration of this earlier work, the framework developed in this paper uses an Inception-v3 rather than a Visual Geometry Group (VGG) deep network, which previous studies have tended to use. The Inception-v3 deep neural network, which was developed by Google®, has been demonstrated to achieve 76.88% Top-1 and 93.344% Top-5 accuracy at the ImageNeT Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) [32]. The structure of the Inception-v3 is presented in Szegedy, et al. [33]. There are four convoluting modules in each basic configuration of the Inception-v3. The CNN comprises of a 42 layer deep convolutional network over a 299 × 299 receptive field, containing over 130 layers [33]. It is a complex neutral network, that consists of many convolutional and pooling layers, batch normalization and special modules. The ILSVRC2012 used a subset of ImageNet with approximately 1000 images in each of 1000 categories. In total, 1.2 million training, 50,000 validation, and 150,000 testing images are available. Top-1 is the conventional level of accuracy that is required: the model answer with the highest probability needs to match the expected answer. A level of Top-5 accuracy refers to any of the model's five highest probability answers that correspond to the expected answer.

3. Deep learning

Deep learning methods incorporating CNN have been demonstrated to be an effective method for computer vision and pattern recognition [34,35]. Lecun, et al. [36] developed the LeNet-5 (a CNN model) based on the Mixed National Institute of Standards and Technology (MNIST) dataset to recognize hand-written numbers. Existing limitations in computing power have hindered the potential of CNNs, but they have been successfully applied to small datasets such as the MNIST, and the Canadian Institute for Advanced Research (CIFAR-10). Improvements in hardware have provided an ability to effectively train large CNN networks by stacking multiple convolutional and pooling layers to not only recognize features from static images, but also those from videos [37,38] The CNN is configured using a graphics processing unit (GPU). This is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer, which is intended for output in a display device [39].

Video recognition, however, is a more challenging task when compared to static images due to the difficulties associated with capturing consecutive spatial and temporal information. Xu, et al. [40] proposed a 3D convolution to compute features from both spatial and temporal dimensions. Additionally, Karpathy, et al. [41] increased the speed of training using two stream CNNs (a low-resolution context stream and a high-resolution fovea stream) to implement a largescale video classification. Notably, deep learning significantly outperforms traditional methods in video recognition. For example, it has been demonstrated that two stream CNNs [42] outperform the Interrupt Descriptor Table (iDT) method [43].

Building upon research presented above, the next section of this paper introduces a deep hybrid model that combines CNN and LSTM network to collect motion data from a video camera and learn from representations that are acquired. The CNN model is applied to each frame to capture the spatial features from the videos, while the LSTM network is used to understand the temporal information from the consecutive frames that are produced.

3.1. Action video representation with deep models

Fig. 1 presents the workflow of the proposed action recognition approach. Initially, the deep models are trained to compute feature representations from the action videos, which are structured using a combination of CNNs and LSTM models. Then, the sequences of feature vectors generated from the second LSTM layer are inserted into the

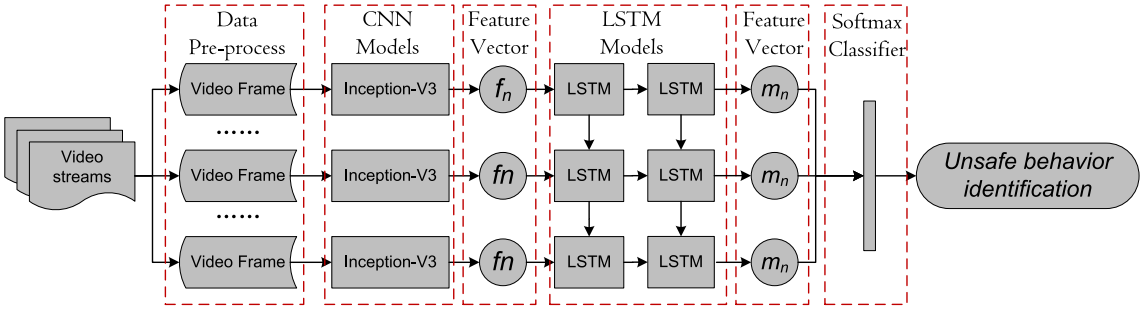

Fig. 1. Method of CNN-LSTM models-based action recognition.

Softmax classifier located in the last layer of the CNN's architecture. The *Softmax* function [44] used in the classification process is expressed as a probabilistic function, as shown in Eq. (1):

$$P(y^{(i)} = n \mid x^{(i)}; W) = \begin{bmatrix} y^{(1)} = 1 \mid x^{(1)}; W \\ y^{(2)} = n \mid x^{(2)}; W \\ \vdots \\ y^{(i)} = n \mid x^{(i)}; W \end{bmatrix} = \frac{1}{\sum_{j=1}^{n} e^{W_j^T x^{(i)}}} \begin{bmatrix} e^{W_1^T x^{(i)}} \\ e^{W_2^T x^{(i)}} \\ \vdots \\ e^{W_n^T x^{(i)}} \end{bmatrix} \quad (1)$$

Where, P stands for the i^{th} training example out of m number of training examples, the j^{th} class out of n number of classes, and weights W; $W_j^T x^{(i)}$ stands for the inputs of the *Softmax* layers.

The procedure to implement the model is described as follows:

Step 1: Each video stream is separated into twenty-five clips. Video frames were randomly selected, as a representation of their time slice. Inputs to the CNN models are video frames, which hold the raw pixel values of an image. The convolutional (CONV) layers compute the output of neurons that are connected to the local regions for each input. As a result, a dot product between their weights and a small region to which they are connected in this input volume is computed. The outputs of the CNN from the fully connected layer will be a 2048-dimension feature vector chosen as a spatial feature.

Step 2: The spatial feature of each frame is obtained following Step 1, and then a video can obtain 25 feature vectors. These feature vectors are fed into the LSTM models, and then sequential temporal features are trained. A sequence feature is generated from the second LSTM model that is produced.

Step 3: The temporal feature vectors become the inputs of the *Softmax* classifier, which enables a probability to be generated. This sequence is averaged over the time steps resulting in a representation.

3.2. Deep CNN models

The CNN has a multi-layer architecture that enables feature extraction to be automatically generated. It also facilitates the classifier to map the extracted feature vector to the final prediction. For each layer, a convolution operation and activation function on the output of the previous layer in the forward propagation phase is employed, which is formalized as:

$$h_{ij}^{k} = f((W^{k*}x)_{ij} + b_k) \quad (2)$$

Where, f is the activation function, b_k is the bias for this feature map, and W^k is the value of the kernel connected to the kth feature map. The CNN network is able to be trained more easily than a multi-layer perceptron (MLP), as the basic cell and its weighting can be shared, thus reducing its complexity. Here the cells act as local filters over the input space and are well-suited to exploit the strong spatially local correlation present in the natural images. For each channel in layer A, weights of the filter are shared to reduce the parameter number. The CNN, is therefore, trainable with a limited number of parameters in the backward propagation phase.

In the Inception-v3 network, modules are comprised of composite layers of five homogeneously-shaped filters, including 1 × 1, 3 × 3, 5 × 5 convolutional kernel sizes, and the output of a 3 × 3 average pooling operations. In the research presented in this paper, an Inception-v3 network with pre-trained parameters based on the *ImageNet* is directly applied to raw action video frames. The *ImageNet* is a large visual database designed for use in visual recognition software. For a detailed description of this database refer to Deng, et al. [32]. The 1 × 1 × 2048 outputs of the last pooling layer are chosen as spatial features. That is, the 2048-dimensional features generated from the last pooling layer in the forward pass of the pre-trained model are used. As a result, the model will not retrain and repeat the Inception network through back propagation. The lack of a need to retrain the model can significantly reduce training time when compared with the use of both CNN-and LSTM models.

3.3. LSTM models

The LSTM neural network is used to process a sequence of a length N input sequence $\{x_1, x_2, \ldots, x_N\}$. RNNs can learn complex temporal dynamics by mapping input sequences to a sequence of hidden states, and to their outputs. However, there exists a vanishing and exploding gradient problem (i.e. difficulty in training the neural network) when vanilla RNNs learn long-term dynamics [45]. LSTMs provide a solution by incorporating memory units that allow the network to learn when to forget previous hidden states and to up-date them when new information is provided. The LSTM has an advanced RNN architecture, which can learn long-range dependencies due to its memory cell. Such models have the ability to control the level of information that flows from a cell.

Fig. 2 illustrates a basic LSTM neuron. Within LSTM models, there exist three gates to control and update the cell's state: (1) inputs; (2) forget; and (3) output. As noted in Fig. 2, the gates are used as a mechanism to determine the information that is able to be received by the cell. The memory cell in each gate consists of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between zero and one, and describes how much of each component can be forwarded to the cell.

For time step t, the cell state can be updated by using the following Equations:

$$\begin{cases} i_t = \delta(W_{xi}x_t + V_{hi}h_{t-1} + b_i) \\ f_t = \delta(W_{xf}x_t + V_{hf}h_{t-1} + b_f) \\ o_t = \delta(W_{xo}x_t + V_{ho}h_{t-1} + b_o) \\ g_t = \tanh(W_{xc}x_t + V_{hc}h_{t-1} + b_c) \\ c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \\ h_t = o_t \otimes \tanh(c_t) \end{cases} \quad (3)$$

Where, δ stand for activate function sigmoid defined as $\delta(x) = (1 + e^{-x})^{-1}$, i_t, f_t, o_t, c_t stand for the outputs of the 'input', 'forget', and 'output' gates and cell at time t, respectively. h_t, b_t, b_f, b_o and b_c stands for offset vector, W_{xi}, W_{xf}, W_{xo}, W_{xc}, V_{hi}, V_{hf}, V_{ho} and V_{hc} stand for the coefficient matrix.

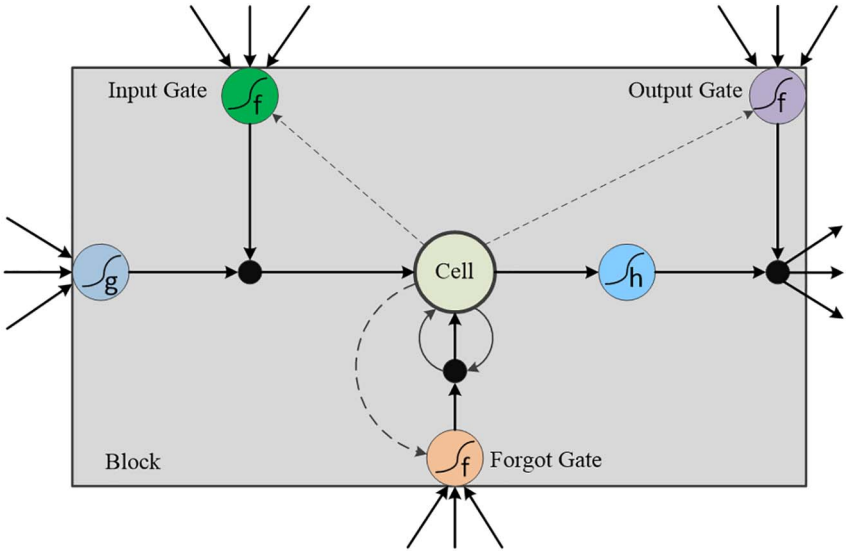

Fig. 2. Basic structure of LSTM unit models.

(1) Normal climbing (2) With an object (3) Backward-facing (4) Reaching far

Fig. 3. Climbing actions for the ladder.
(1) Normal climbing
(2) With an object
(3) Backward-facing
(4) Reaching far

In the proposed hybrid model presented in Fig. 1, a two-layer LSTM network is constructed to learn about the video's temporal dynamics of the features that are generated by the last pooling layer of the Inception-v3 networks. In Fig. 1, $\{f_1, f_2, \ldots, f_n\}$ are the n features that are computed by the Inception-v3 network from n frames in each video. Thus, from an input sequence $\{f_1, f_2, \ldots, f_n\}$, the memory cells in the two LSTM layers will produce a representation sequence $\{m_1, m_2, \ldots, m_n\}$. This sequence is then averaged over the entire time period resulting in representation F, which is expressed as:

$$F = \frac{m_1 + m_1 + \cdots + m_n}{n} \tag{4}$$

This feature vector F feeds into the *Softmax* layer so that the unsafe behavior can be identified for each input derived from the video. W is the parameter vector of the last *Softmax* layer. Notably, $\{f_1, f_2, \ldots, f_n\}$ are the n features that are computed by Inception-v3 networks from n frames in each video. Moreover, F represents the mean pooling feature vector using the weighted features $\{m_1, m_2, \ldots, m_n\}$, which learned from the two LSTM layers and W is the parameter vector of the last logistic regression layer.

4. Evaluating the deep hybrid learning model to detect unsafe behavior

To effectively implement safety control process on-site there is a need to identify the sources of risk that can cause injury and fatalities [46–48]. A number of methods for determining unsafe behaviors have been reported in the extant literature [20–22]. For example, Han and Lee [20] identified a series unsafe behaviors based on a list of Occupational Safety and Health Administration (OSHA) accident statistics juxtaposed with companies' safety performance records. Similarly, unsafe actions have been identified by injury records, accidents report and near-miss reports. Other approaches utilize safety documents relying on individual capabilities (e.g. a safety manager's expertise).

4.1. An experiment: Capturing the motion of unsafe behavior

Falls are one of leading causes of accidents in construction, accounting for 34% fatalities and 24% non-fatalities. Notably, falls from ladders account for 9% of deaths and 6% of injuries [49]. Ladders are an important piece of equipment and are integral for undertaking work at heights. Considering their importance and use in construction, an experiment was designed to examine unsafe behaviors and the effectiveness of the developed hybrid deep learning model to detect such actions.

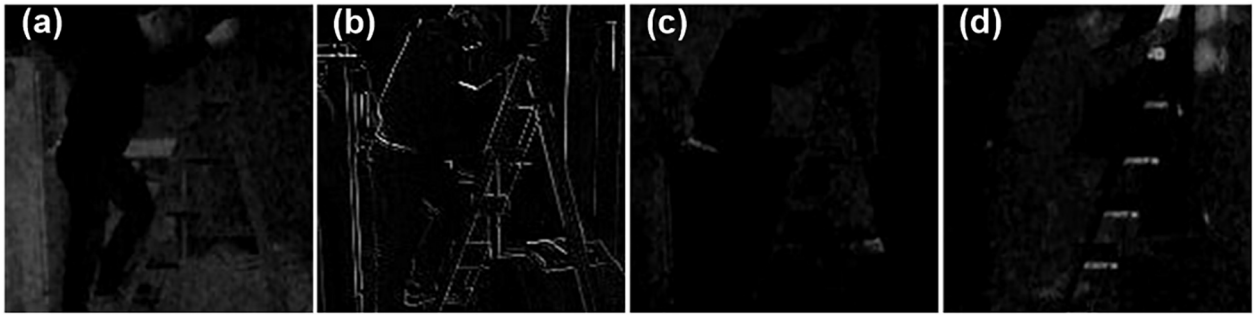

Fig. 4. Examples of feature maps generated by the convolution layer in CNN models.

In the experiment, video recordings of a person climbing and dismounting from a ladder were collected in a laboratory environment using video cameras (Fig. 3). Each video is on average 8 s in length, and has a resolution of 1920 ∗ 1080. In the experiment, a person simulated the actions that a construction worker would typically perform on-site. For each class of actions, 50 samples (i.e., the number of cycles) were collected to reflect the varying deviations in motion of the person for each climbing action. To accommodate the potential for bias, motion samples were recorded and observed from different viewpoints.

5. Implementation details

The deep learning models were trained using Theano [43]. All algorithms were performed on a server equipped with a 2.50 GHz Intel(R) Xeon(R) E5-2680 CPU, a NVIDIA(R) Tesla(TM) K80 GPU and 64G RAM. The simplest and most common method to reduce 'overfitting' (i.e. noise) on image data is to artificially enlarge the dataset. Data augmentation was used to ensure the model's fit to detect and classify unsafe actions. Utilizing the temporal continuity of video, different frames were randomly selected in each video clip, which enabled new training examples to be generated. All the original video frames were resized to be 384 × 384 due to the requirements of the CNN models. Then, each video's datasets were split into twenty-five clips and frames were randomly selected, as a representation of their time slice. Then, the image mean for the input of the CNNs was set to 117 and y relabeled to be in {0, 1} for single task.

Spatial features were extracted via the pre-trained Inception-v3 networks [33]. The random cropping method for input of the CNN network was used, which is a form of data augmentation technique. In the experiment, a random crop of 299 × 299 patches from the 384 × 384 image were used as an input. The output of the last pooling layer of Inception-v3 networks was used to represent spatial features, which have a 2048-dimensional vector.

For each video, the 25 selected frames were processed in parallel using the CNNs to obtain their vectors, which were rearranged in a temporal order to represent the spatial features. These video representations were pre-processed to a zero mean. Next they were used as the input to a LTSM network's next two-layers, with each having 2048 hidden units to extract spatial features with size $k = 2048$. The output of the LSTM layer was then followed by a mean pooling, with the *Softmax* layer being used to obtain the probability score (Fig. 2). The size of the parameter W of the logistic regression was 2048. During the training process, a *RMSProp* [50] (i.e. an optimizer that utilizes the magnitude of recent gradients to normalize them) was used with a batch size setting of 40. To prevent over-fitting, strategies such as a l_2 norm regularization with the coefficient $\lambda = 0.0001$ behind the LSTM layer, were employed when the error rate did not improve.

5.1. Motion recognition

The experimental data comprised of a total of 200 pictures, which was divided into two sets: (1) training; and (2) test. The data sets were also divided into two modes:

1. A total of 160 and 40 videos were randomly selected from the training and test sets, respectively. Both the training and test sets contained the four behaviors, which are identified in Fig. 3. These four behaviors were, in turn, marked with two types of labels (0, 1), for normal ladder and abnormal ladder climbing.
2. Different viewpoints (front, back, left and right) from the training and test sets were obtained. Videos from the front, back and left were used as the training set. The remaining view (right) videos were used as the test set. As a result, the data set was divided four times to ensure that each direction of the video can became a sample test set, while ensuring that varying types of action were captured. These four actions are labeled separately for different tags (0, 1, 2, 3), and represent the actions shown in Fig. 3.

The CNN was applied to learn from the spatial information from the actions identified in the videos. The output of the convolution layer of the CNN models for normal climbing is presented in Fig. 4. Here it can be seen that the original image selected for normal climbing and parts (a), (b) and (c) contain the extracted spatial information.

From the differing viewpoints, the detailed accuracy rate of two/four action types are presented in Fig. 5(a). Two types of labels (0, 1) are used to refer to normal and abnormal ladder climbing. In addition, the labels (0, 1, 2, 3) refer to the climbing actions in Fig. 3. It can be seen from Fig. 5(a) that the 'left' viewpoint was the 'best'. Though, this may have been due to the 'left' having a better access to the view .

The models predictive accuracy from the training process is presented in Fig. 5(b), which tested the performance of the algorithm to recognize a safe/unsafe action. Its accuracy in this experiment was revealed to be 97% (Fig. 5(a)). However, the model's accuracy in recognizing all four types of actions was 92%.

While a high levels of detection and classification accuracy were obtained for the four types of unsafe actions, several were unable to be identified from other viewpoints. Reasons for these lower levels of accuracy are twofold: (1) a lack of data for computer extraction and limited learning features. In addition, the paucity of data can cause 'overfitting', resulting in the model not being able to acquire additional information; and (2) the spatial feature of an image being computed into a 2048-dimension vector by the CNN models may not capture the image's entire data.

5.2. Performance evaluation

The hybrid (CNN + LSTM) deep learning model's performance was compared with four types of feature descriptor: (1) HOG [51]; (2) HOF [52]; (3) Motion Boundary Histograms (MBH) [53]. These methods are dependent on extracting handcrafted features from inputs, which are then fed into a classifier. For each trajectory, several descriptors (i.e., HOG, HOF and MBH) with exactly the same parameters were computed [54]. The final dimensions of the descriptors were 96 for HOG, 108 for HOF and 192 for MBH. Extracting features from static images involves

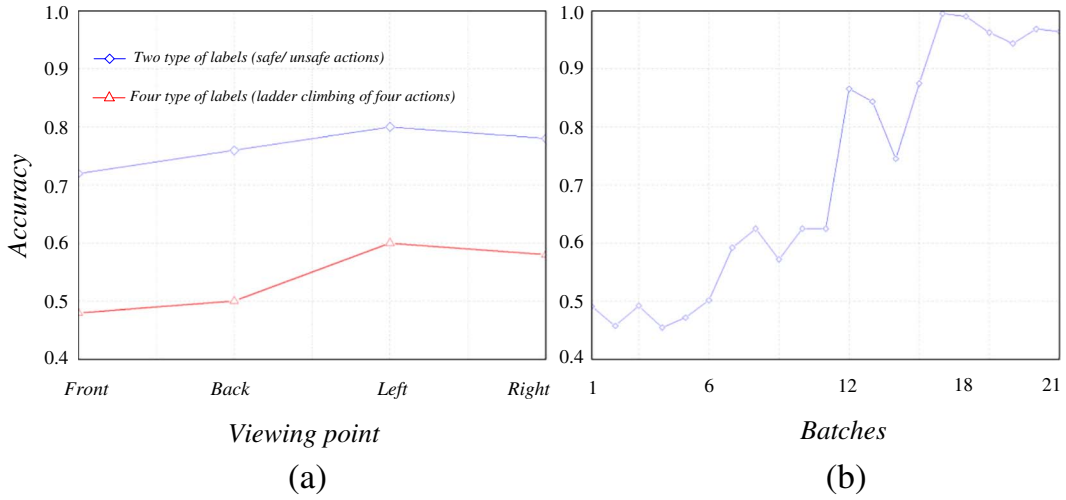

Fig. 5. (a) Accuracy data of four types of worker climbing action, and (b) prediction accuracies from training process with the CNN + LSTM method.

Table 1
Comparison of the hybrid deep learning method and feature descriptors.

Algorithm	HOG	HOF	MBH	HOF + MBH	CNN + LSTM
Two types of labels	82.1%	83.7%	85.5%	88.9%	97%
Four types of labels	74.3%	77.6%	80.8%	86.3%	92%

two steps: (1) a feature detection; and (2) then a local feature description step. Previous research [29] has extended these local features to calculate spatio-temporal points of interest in the recognition of recorded motions. These features provide the pre-processing step that is required prior to applying a classification method. In Table 1, it can be seen that the ability of the developed hybrid (CNN + LSTM) model is more accurate than other methods in automatically extracting and training the feature to learn and capture motions.

In China, for example, video surveillance has been widely applied to continuously monitor safety during construction [48]. This practice is not common in many other countries (e.g., Australia, United Kingdom, and United States) due to issues surrounding privacy and its adoption would also require the permission and cooperation of unions [55]. In addressing the issue of privacy, workers can be provided with an option of not being videoed when they conduct their activities. In previous health and safety studies, for example, it has been demonstrated that the impact of video recording on an observed person's behavior is insignificant [56].

Explicitly, the hybrid (CNN + LSTM) model's ability to accurately detect unsafe behavior provides a high degree of assurance that it can be potentially used in a real-life setting. Close collaboration, however, will be required with site management and their team and subcontractors to engender the model's process of learning from behaviors that are enacted. For example, assistance will be required to identify the precise location of cameras (i.e., viewpoints) and occlusions (e.g., those from workers and equipment) to collate the images needed to ensure the deep model is able to detect and provide an assessment of unsafe actions. While data augmentation was applied to enhance the model's capacity to learn, it is not suitable for real-time monitoring due to the length of time (> a second) required to detect an unsafe action.

In the developed hybrid (CNN + LSTM) model, a larger set and longer training period is required to learn about the features being studied in the videos. However, it was observed that some actions were not able to be recognized. This may have been attributable to the size of the sample used for training and the limited number of unsafe actions that were considered. With larger datasets it is anticipated that the performance of the hybrid (CNN + LSTM) model may further improve and provide more accurate results.

6. Conclusion

The experimental results have demonstrated that the proposed hybrid model (CNN + LSTM) can automatically extract and classify unsafe behaviors (i.e., those associated with climbing a ladder) using conventional video with a high level of accuracy. This method does not only provide the means to recognize human movement, but also produces data for motion analysis. The models accuracy to detect safe/unsafe actions was 97% and 92% respectively, which exceeds the actual capacity of current state-of-the-art methods (e.g., HOG, HOF, MBH) by an average of 10%.

The developed hybrid (CNN + LSTM) model can potentially be applied to automatically detect unsafe actions during construction and thus lead to perfunctory intervention by management, which can result in immediate behavior modification. The video can be used to provide workers with direct visual feedback and therefore enable them to be educated and learn how their activities should be safely performed. Improvements in the motion capture algorithms are, however, required to reduce error detection time. Such advancements to algorithm computation ability will assist to minimize bias and enable unsafe behaviors to be detected more accurately. In addition, there is a need to better understand the context of temporal-spatial information so as to be able to determine the relationships between equipment and workers. Thus, further research needs to focus on recognizing actions that simultaneously accommodate multiple pieces of equipment/workers contained within video frames.

Acknowledgments

This research is supported in part by a major project of The National Social Science Key Fund of China (Grant No.13&ZD175), supported by National Natural Science Foundation of China (Grant No. 7 1732001, No.51678265, No.71301059), supported by “the Fundamental Research Funds for the Central Universities” (Grant No.2017KFYXJJ134).

References

[1] D. Fang, H. Wu, Development of a Safety Culture Interaction model for construction projects, Saf. Sci. 57 (8) (2013) 138–149, http://dx.doi.org/10.1016/j.ssci.2013.02.003.
[2] L. Ding, X.U. Jie, A review of metro construction in China: organization, market, cost, safety and schedule, Frontiers of Engineering Management. 4 (1) (2017), http://dx.doi.org/10.15302/j-fem-2017015.

[3] Y. Zhou, L. Ding, X. Wang, M. Truijens, H. Luo, Applicability of 4D modeling for resource allocation in mega liquefied natural gas plant construction, Autom. Constr. 50 (2015) 50–63, http://dx.doi.org/10.1016/j.autcon.2014.10.016.

[4] Y. Zhou, H. Luo, Y. Yang, Implementation of augmented reality for segment displacement inspection during tunneling construction, Autom. Constr. (2017), http://dx.doi.org/10.1016/j.autcon.2017.02.007.

[5] L.Y. Ding, B.T. Zhong, S. Wu, H.B. Luo, Construction risk knowledge management in BIM using ontology and semantic web technology, Saf. Sci. 87 (2016) 202–213.

[6] Y. Zhou, W. Su, L. Ding, H. Luo, P.E.D. Love, Predicting safety risks in deep foundation pits in subway infrastructure projects: a support vector machine approach, J. Comput. Civ. Eng. 31 (5) (2017).

[7] Y. Zhou, L. Ding, Y. Rao, H. Luo, B. Medjdoub, H. Zhong, Formulating project-level building information modeling evaluation framework from the perspectives of organizations: a review, Autom. Constr. 81 (2017) 44–55.

[8] H.W. Heinrich, D. Petersen, N.R. Roos, Industrial Accident Prevention: A Safety Management Approach, (1980), http://dx.doi.org/10.2307/2518508.

[9] I.M. Fam, H. Nikoomaram, A. Soltanian, Comparative analysis of creative and classic training methods in health, safety and environment (HSE) participation improvement, J. Loss Prev. Process Ind. 25 (2) (2012) 250–253, http://dx.doi.org/10.1016/j.jlp.2011.11.003.

[10] T. Guan, Y. Wang, L. Duan, R. Ji, On-Device mobile landmark recognition using binarized descriptor with multifeature fusion, ACM Trans. Intell. Syst. Technol. 7 (1) (2015) 1–29, http://dx.doi.org/10.1145/2795234.

[11] Y. Zhang, T. Guan, L. Duan, B. Wei, J. Gao, T. Mao, Inertial sensors supported visual descriptors encoding and geometric verification for mobile visual location recognition applications, Signal Process. 112 (C) (2015) 17–26, http://dx.doi.org/10.1016/j.sigpro.2014.08.029.

[12] B. Wei, T. Guan, L. Duan, J. Yu, T. Mao, Wide area localization and tracking on camera phones for mobile augmented reality systems, Multimedia Systems 21 (4) (2015) 381–399, http://dx.doi.org/10.1007/s00530-014-0364-2.

[13] S. Chi, C.H. Caldas, Image-based safety assessment: automated spatial safety risk identification of earthmoving and surface mining activities, J. Constr. Eng. Manag. 138 (3) (2012) 341–351, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000438.

[14] I.P.T. Weerasinghe, J.Y. Ruwanpura, Automated data acquisition system to assess construction worker performance, Construction Research Congress, 2009, pp. 61–70, , http://dx.doi.org/10.1061/41020(339)7.

[15] F. Wang, X. Luo, H. Li, Y. Yu, X. Yang, Motion-based analysis for construction workers using biomechanical methods, Frontiers of Engineering Management. 4 (1) (2017) 84, http://dx.doi.org/10.15302/j-fem-2017004.

[16] J. Gong, C.H. Caldas, Learning and classifying motions of construction workers and equipment using bag of video feature words and Bayesian learning methods, International Workshop on Computing in Civil Engineering, 2015, pp. 274–281, , http://dx.doi.org/10.1061/41182(416)34.

[17] Z. Zhu, I. Brilakis, Concrete column recognition in images and videos, J. Comput. Civ. Eng. 24 (6) (2010) 478–487, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000053.

[18] M.W. Park, A. Makhmalbaf, I. Brilakis, Comparative study of vision tracking methods for tracking of construction site resources, Autom. Constr. 20 (7) (2011) 905–915, http://dx.doi.org/10.1016/j.autcon.2011.03.007.

[19] S.J. Ray, J. Teizer, Real-time construction worker posture analysis for ergonomics training, Adv. Eng. Inform. 26 (2) (2012) 439–455, http://dx.doi.org/10.1016/j.aei.2012.02.011.

[20] S. Han, S. Lee, A vision-based motion capture and recognition framework for behavior-based safety management, Autom. Constr. 35 (2013) 131–141, http://dx.doi.org/10.1016/j.autcon.2013.05.001.

[21] M. Liu, S. Han, S. Lee, Tracking-based 3D human skeleton extraction from stereo video camera toward an on-site safety and ergonomic analysis, Constr. Innov. 16 (3) (2016) 348–367, http://dx.doi.org/10.1108/ci-10-2015-0054.

[22] S. Han, S. Lee, F. Peña-Mora, Comparative study of motion features for similarity-based modeling and classification of unsafe actions in construction, J. Comput. Civ. Eng. 28 (5) (2014), http://dx.doi.org/10.1061/(asce)cp.1943-5487.0000339.

[23] S.U. Han, M. Achar, S.H. Lee, F. Peña-Mora, Empirical assessment of a RGB-D sensor on motion capture and action recognition for construction worker monitoring, Visualization in Engineering. 1 (1) (2013) 6, http://dx.doi.org/10.1186/2213-7459-1-6.

[24] S. Han, S. Lee, F. Peña-Mora, Vision-based detection of unsafe actions of a construction worker: case study of ladder climbing, J. Comput. Civ. Eng. 27 (6) (2013) 635–644, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000279.

[25] J. Seo, S. Han, S. Lee, H. Kim, Computer vision techniques for construction safety and health monitoring, Adv. Eng. Inform. 29 (2) (2015) 239–251, http://dx.doi.org/10.1016/j.aei.2015.02.001.

[26] D. Wang, F. Dai, X. Ning, Risk assessment of work-related musculoskeletal disorders in construction: state-of-the-art review, J. Constr. Eng. Manag. 141 (6) (2015), http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000979.

[27] J. Gong, C.H. Caldas, C. Gordon, Learning and classifying actions of construction workers and equipment using Bag-of-Video-Feature-Words and Bayesian network models, Adv. Eng. Inform. 25 (4) (2011) 771–782, http://dx.doi.org/10.1016/j.aei.2011.06.002.

[28] I. Laptev, T. Lindeberg, On space-time interest points, Int. J. Comput. Vis. 64 (2–3) (2005) 107–123, http://dx.doi.org/10.1007/s11263-005-1838-7.

[29] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2008) 1–8, http://dx.doi.org/10.1109/CVPR.2008.4587756.

[30] Q. Li, Z. Qiu, T. Yao, T. Mei, Y. Rui, J. Luo, Action recognition by learning deep multi-granular spatio-temporal video representation, ACM on International Conference on Multimedia Retrieval, 2016, pp. 159–166, , http://dx.doi.org/10.1145/2911996.2912001.

[31] Y.H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: deep networks for video classification, Computer Vision and Pattern Recognition, 2015, pp. 4694–4702, , http://dx.doi.org/10.1109/cvpr.2015.7299101.

[32] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, F.F. Li, ImageNet: a large-scale hierarchical image database, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2009) 248–255, http://dx.doi.org/10.1109/CVPR.2009.5206848.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, Computer Vision and Pattern Recognition, 2016, pp. 2818–2826, , http://dx.doi.org/10.1109/CVPR.2016.308.

[34] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117, http://dx.doi.org/10.1016/j.neunet.2014.09.003.

[35] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105, , http://dx.doi.org/10.1145/3065386.

[36] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.

[37] X. Zhang, H. Zhang, Y. Zhang, Y. Yang, W. Meng, H. Luan, J. Li, T.S. Chua, Deep fusion of multiple semantic cues for complex event recognition, IEEE Trans. Image Process. 25 (3) (2016) 1033–1046, http://dx.doi.org/10.1109/TIP.2015.2511585.

[38] W. Wang, Y. Yan, S. Winkler, N. Sebe, Category specific dictionary learning for attribute specific feature selection, IEEE Trans. Image Process. 25 (3) (2016) 1465, http://dx.doi.org/10.1109/TIP.2016.2523340.

[39] ger, Jens, Westermann, diger, Linear algebra operators for GPU implementation of numerical algorithms, ACM Trans. Graph. 22 (3) (2003) 908–916, http://dx.doi.org/10.1145/1201775.882363.

[40] W. Xu, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231, http://dx.doi.org/10.1109/TPAMI.2012.59.

[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, F.F. Li, Large-scale video classification with convolutional neural networks, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2014) 1725–1732, http://dx.doi.org/10.1109/TIP.2016.2523340.

[42] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, Adv. Neural Inf. Proces. Syst. 1 (4) (2014) 568–576, http://dx.doi.org/10.1002/14651858.CD001941.

[43] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: A CPU and GPU Math Compiler in Python, (2010), pp. 3–10 http://conference.scipy.org/proceedings/scipy2010/bergstra.html.

[44] Y.J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput. Aided Civ. Inf. Eng. 32 (5) (2017) 361–378, http://dx.doi.org/10.1111/mice.12263.

[45] J. Kolen, S. Kremer, Gradient Flow in Recurrent Nets: The Difficulty of Learning Longterm Dependencies, Wiley-IEEE Press, 2001, pp. 237–243 IBSN: 9780470544037.

[46] R.M. Choudhry, Behavior-based safety on construction sites: a case study, Accid. Anal. Prev. 70 (6) (2014) 14–23, http://dx.doi.org/10.1016/j.aap.2014.03.007.

[47] E.S. Geller, Behavior-based safety and occupational risk management, Behav. Modif. 29 (3) (2005) 539–561, http://dx.doi.org/10.1177/0145445504273287.

[48] S. Guo, L. Ding, H. Luo, X. Jiang, A big-data-based platform of workers' behavior: observations from the field, Accid. Anal. Prev. 93 (2016) 299–309, http://dx.doi.org/10.1016/j.aap.2015.09.024.

[49] U.D.o, Labor, Workplace Injuries and Illnesses - 2011, (2012).

[50] T. Tieleman, G. Hinton, Lecture 6.5-RMSProp, COURSERA: Neural Networks for Machine Learning, University of Toronto, Tech. Rep, 2012.

[51] J. Yang, Z. Shi, Z. Wu, Vision-based action recognition of construction workers using dense trajectories, Adv. Eng. Inform. 30 (3) (2016) 327–336, http://dx.doi.org/10.1016/j.aei.2016.04.009.

[52] H. Wang, C. Schmid, Action recognition with improved trajectories, IEEE International Conference on Computer Vision, 2014, pp. 3551–3558, , http://dx.doi.org/10.1109/ICCV.2013.441.

[53] H. Wang, C. Schmid, Lear-inria submission for the thumos workshop, ICCV Workshop on Action Recognition with a Large Number of Classes, Vol. 2 2013 (p. 8).

[54] H. Wang, A. Kläser, C. Schmid, C.L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vis. 103 (1) (2013) 60–79, http://dx.doi.org/10.1007/s11263-012-0594-8.

[55] W. Knight, New boss on construction sites is a drone, MIT, Technol. Rev. (2015), http://dx.doi.org/10.1109/CVPR.2008.4587756.

[56] P. Carayon, Handbook of Human Factors and Ergonomics in Health Care and Patient Safety, CRC Press, 2016 (1439830347).