Smart Structures and Systems, *Vol. 30, No. 2 (2022) 145-157*
https://doi.org/10.12989/sss.2022.30.2.145

Structural live load surveys by deep learning

Yang Li [1a] and Jun Chen*[1,2]

[1] *College of Civil Engineering, Tongji University, Shanghai 200092, China*
[2] *State Key Laboratory of Disaster Reduction in Civil Engineering, Tongji University, Shanghai 200092, China*

(*Received February 22, 2021, Revised May 1, 2022, Accepted May 6, 2022*)

Abstract. The design of safe and economical structures depends on the reliable live load from load survey. Live load surveys are traditionally conducted by randomly selecting rooms and weighing each item on-site, a method that has problems of low efficiency, high cost, and long cycle time. This paper proposes a deep learning-based method combined with Internet big data to perform live load surveys. The proposed survey method utilizes multi-source heterogeneous data, such as images, voice, and product identification, to obtain the live load without weighing each item through object detection, web crawler, and speech recognition. The indoor objects and face detection models are first developed based on fine-tuning the YOLOv3 algorithm to detect target objects and obtain the number of people in a room, respectively. Each detection model is evaluated using the independent testing set. Then web crawler frameworks with keyword and image retrieval are established to extract the weight information of detected objects from Internet big data. The live load in a room is derived by combining the weight and number of items and people. To verify the feasibility of the proposed survey method, a live load survey is carried out for a meeting room. The results show that, compared with the traditional method of sampling and weighing, the proposed method could perform efficient and convenient live load surveys and represents a new load research paradigm.

Keywords: big data; deep learning; live load survey; web crawler; YOLOv3

1. Introduction

Engineering structures are constructed to meet human life, production, transportation, and aesthetic needs. The reliability of load value is not only the key to design a proposed structure but also the basis for performance evaluation of an existing structure. Therefore, load modeling is critical to ensure the reliability of engineering structures, and a large amount of measured data is essential (Kaimal *et al*. 1972, Kumar 2002a, Wang and Li 2012). Unlike seismic and wind loads, live load data for building structures cannot be automatically collected over time by electronic monitoring equipment. At present, most data required for modeling of floor live load are obtained by researchers conducting on-site surveys. This traditional survey method is inefficient, costly, and difficult to carry out on a large scale, but it has become the conventional method in an era of limited analog data. Taking the load code for building structures in China as an example, it has been modified and improved nine times, and the research methods are primarily based on sampling surveys and statistical regression. In the history of load code development, only two large-scale nationwide load surveys and statistical analyses have been carried out. The first load survey was conducted before the revision of load code for industrial and civil buildings, and 606 residential rooms and 258 office rooms were studied for floor live load. From 1977 to 1981, a second load survey was carried out, based on studies of structural reliability and load combination. In this survey, the floor live load of residential and office buildings was the main concern, and 556 residential rooms and 2201 office rooms were evaluated (Jin and Zhao 2012). Since then, no load surveys have been conducted, and the database of the load code (GB50009-2012) consists of data that resulted from the second load survey.

In the past few decades, scholars have conducted investigations on floor live load, and the research paradigms are similar to the load code of China. Andam (1986) carried out a load survey in Accra having a total area of 27,818 m^2 and obtained the design load at a given fractile. Choi (1990) conducted a load survey of 11 office buildings in Sydney and calibrated the live load model used to calculate a lifetime maximum total load. Based on the Sydney survey data, Choi (1991) then proposed a refined temporary live load model in which all parameters are considered to be area dependent. To formulate a national load code, Asantey and Andam (1996) surveyed the live load of factories and warehouses in Ghana, covering areas of 12,565 m^2 and 13,803 m^2, respectively. Ruiz and Sampayo-Trujillo (1997) conducted a live load survey of office buildings in Mexico City having a total area of 14,890 m^2 and proposed a new load-reducing rule for the Mexican code. Kumar carried out a live load survey of eight office buildings in Kanpur having a total area of 11,720 m^2. Based on the measured data, a probabilistic live load model was proposed, and the sustained load, extraordinary load, and lifetime maximum total load obtained from the model were compared with survey results from the UK, the USA, and Australia (Kumar

*Corresponding author, Professor,
E-mail: cejchen@tongji.edu.cn
[a] Ph.D. Student, E-mail: LIYang95822@163.com

ISSN: 1738-1584 (Print), 1738-1991 (Online)

2002a, b). Ge *et al*. (2008) conducted a live load survey on residential buildings in the Central Plains region and obtained 880 valid samples. The probability distribution functions and statistical parameters of the maximum of the sustained live load and the temporary live load were obtained. Wu *et al*. (2012) conducted a live load survey of residential buildings in Xi'an and Baotou having an area of 7,900 m^2, and the results showed that the probability characteristics of equivalent uniformly distributed live loads of residential buildings in China have changed greatly.

In traditional live load modeling, load surveys are limited by labor costs, time costs, and the difficulty of weighing large items on-site. The survey area or measured sample size is relatively small compared to wide-ranging building structures. Consequently, the conspicuousness of data statistics and the quality of the load model are heavily constrained by the representativeness of the sample. In addition, socio-economic developments have changed the form and function of current residential and office buildings, these changes have widened the gap between the data on which load modeling is based and the current load status. Due to the fact that load surveys can only be carried out in stages, the dynamics of socio-economic change makes it difficult for the collected data to reflect indoor items in time. Therefore, the reliability of load statistics is the weakest component of structural reliability analysis. Research of building live load is urgently needed to find solutions to critical issues, such as sample size, labor cost, site weighing, and timeliness of data. Obviously, a change of research paradigm is required.

Recently, with rapid advances in computer hardware and continuous data accumulation, deep learning (DL) algorithms have been successfully applied to computer vision (Ren *et al*. 2016, He *et al*. 2016), natural language processing (Bahdanau *et al*. 2014, Oord *et al*. 2016), speech recognition (Hinton *et al*. 2012, Graves *et al*. 2013), and other fields. DL brings new ideas and opportunities to solve problems in civil engineering. Convolutional neural networks (CNN) have demonstrated outstanding image processing capabilities. Cha *et al*. (2017) trained a CNN to assess concrete cracks and achieved approximately 98% accuracy. Park *et al*. (2018) proposed a single model combining the residual fitting mechanism in the wavelet transform and CNN network for time-series data classification, and the single model presented a superior performance than feature-based models. Xu *et al*. (2019) proposed a modified fusion CNN architecture that considered the multilevel and multi-scale features of the input images to conduct crack identification inside steel box girders. Duan *et al*. (2019) presented a CNN-based approach for structural damage identification on hangers in a tied-arch bridge structure using the spatial and spectral information of the acceleration responses on the deck. Ni *et al*. (2019) proposed a crack delineation network using convolutional feature fusion and pixel-level classification for structural damage detection and segmentation. Luo *et al*. (2019) introduced a faster region-based CNN for autonomous detection of potholes from images and achieved a high average precision over 93%. Tang *et al*. (2019) proposed a novel method to detect data anomalies based on CNN to imitate human vision and decision making. Recurrent neural networks (RNN) are suitable for processing time-series data. Guo *et al*. (2019) utilized Kohonen neural network and long short-term memory (LSTM) neural network to manage the massive data required to evaluate the health status of bridges. Xie *et al*. (2019) proposed an LSTM method to solve the problem of the multiple factors on landslide displacements in the Laowuji landslide. At present, generative adversarial networks (GAN) are less frequently used in civil engineering. Xiong and Chen (2019) presented a new model that combined the conditional GAN and Wasserstein GAN with gradients penalty to generate loads from individuals walking, jumping, and bouncing.

Inspired by the successful application of DL-based methods to solve problems in civil engineering, this paper reports the development of a novel DL-based framework for structural live load surveys. The core contribution of this framework is to conduct live load surveys without weighing objects directly, allowing load statistics work to be conducted efficiently. Such a method is needed to promote the study of live loads, which has been stagnant for a long time. To construct the proposed DL-based framework, indoor objects and face detection models based on the YOLOv3 (Redmon and Farhadi 2018) algorithm are developed to eliminate the influence of messy backgrounds and count the number of crowds, respectively. Web crawlers are then developed to collect target weights from network big data automatically using detection results. To the best knowledge of the authors, such a DL-based approach to conducting live load surveys is not currently found in the literature. This paper provides insights to readers who are interested in designing DL-based live load survey models.

The remainder of this paper is organized as follows: Section 2 introduces the overall architecture of a DL-based live load survey. Section 3 explains the implementation in detail. Section 4 interprets the training and test results and discusses a case study. Section 5 concludes the paper.

2. Overall architecture of DL-based live load survey

2.1 Rationale for the new framework

The rationale for the DL-based building live load survey method is as follows: Objects in buildings map to the Internet in a variety of digital forms. These forms can be divided into two types of features: direct and indirect. Direct features are unique identifiers, such as barcodes, QR codes, RF codes, and product identifications; indirect features are non-unique identifiers, such as images, videos, and voices. Object features are used to mine Internet big data resources to obtain the weight information needed for the load survey.

Fig. 1 outlines the implementation of the proposed DL-based live load survey method: 1) Portable smart devices, such as smartphones, are used to comprehensively collect images, audios, video, text, and other information about objects in the buildings being surveyed. Such information

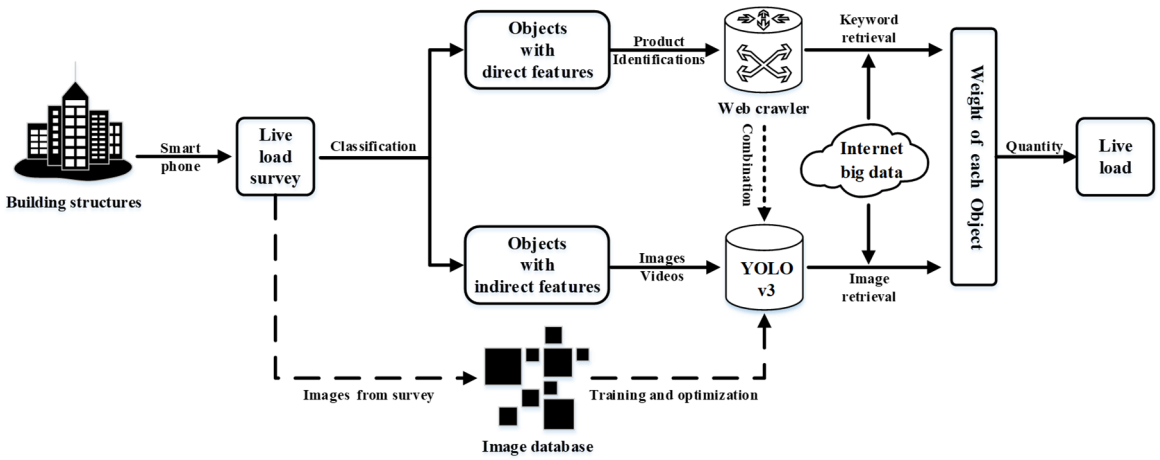


Fig. 1 Workflow of the proposed method

as the building category, room type, area, number of items, is also needed. 2) The collected information is used to classify objects into those having direct or indirect features. 3) Based on various characteristics, objects having direct features are processed in batches using optical character recognition (OCR), two-dimensional code recognition, barcode recognition, and other techniques. Then based on Internet big data, web crawler is combined with keyword retrieval to obtain the weight information of objects. 4) An indoor object detection model based on YOLOv3 is used to detect items having indirect features. After removing the influence of background and unrelated items, images are retrieved from Internet big data, and the weight information of items is obtained using web crawler. To determine the load contributed by people in the building, a face detection model based on YOLOv3 is used to count individuals, and then the total weight of the crowd is obtained using an average weight value.

According to the proposed scheme, an investigator only needs to carry a smartphone to conduct a live load survey, which saves on costs and improves efficiency. At the same time, the images and videos collected during live load surveys can be used to continuously update the dataset to further optimize the DL algorithm. To better understand the proposed method, the implementation of indoor objects and face detection models based on YOLOv3 and weight acquisition through web crawlers are presented, followed by a case study.

2.2 Overall architecture of YOLOv3

YOLOv3 is an improved object detection algorithm based on YOLO9000 (Redmon and Farhadi 2017). This paper presents an indoor object detection model and a face detection model based on YOLOv3 to input the survey image, and then output the position of the target in the image to prepare for subsequent weight acquisition.

As shown in Fig. 2, YOLOv3 is constructed from CBL module, residual module (ResM), UpSampling layer, and Concatenate layer, having a total of 252 layers. The CBL module consists of a convolution (Conv) layer, a batch normalization (BN) (Ioffe and Szegedy 2015) layer, and

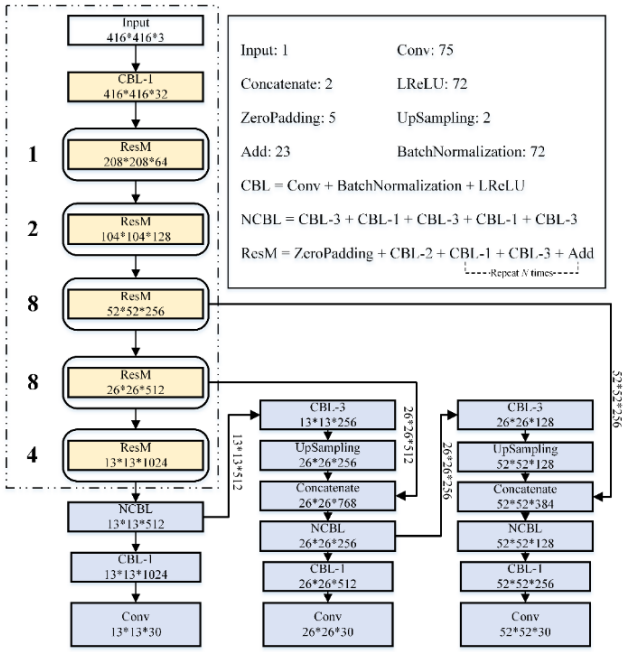


Fig. 2 Architecture of the YOLOv3 network

Table 1 Composition of each CBL module

CBL-1	CBL-2	CBL-3
Conv (3×3/1)	Conv (3×3/2)	Conv (1×1/1)
BN layer	BN layer	BN layer
LReLU layer	LReLU layer	LReLU layer

a leaky rectified linear unit (LReLU) (Maas *et al*. 2013) activation layer. According to the size and stride of the convolution kernel, the CBL module can be divided into three types. Table 1 lists the composition of each type. The ResM draws on the idea of the residual units (He *et al*. 2016) and is responsible for performing feature extraction. ResM consists of a ZeroPadding layer, three types of CBL modules, and an Add layer. The ZeroPadding layer fills zeros at the top and left sides of the input feature map to

facilitate the operation of a convolution kernel with a stride of 2. The Add layer performs element-wise addition between the outputs of the identity mapping and the outputs of the stacked CBL-1 and CBL-3. Noted that the number on the left of ResM in Fig. 2 indicates the times of repetition for the CBL-1, CBL-3, and Add layer in the module.

Several convolutional layers are added on the top of the feature extractor (dotted box) to obtain the first prediction. To combine the advantages of the low-level feature location information and the high-level feature semantic information, the high-level feature maps are upsampled and spliced with the low-level feature maps in the channel dimension using the UpSampling and Concatenate layers. The nearest neighbor interpolation method is applied in the UpSampling layer, which repeats the rows and columns of the data twice. Figs. 3-4 show the operation of the Concatenate layer and UpSampling layer, respectively. In this study, the feature maps from two layers previous of the first prediction are upsampled and merged with earlier feature maps. Several convolutional layers are then added to process this combined feature maps to obtain the second prediction. The same sequence is performed once more to obtain the final prediction. As shown in Fig. 2, YOLOv3 predicts boxes at three scales similar to feature pyramid.

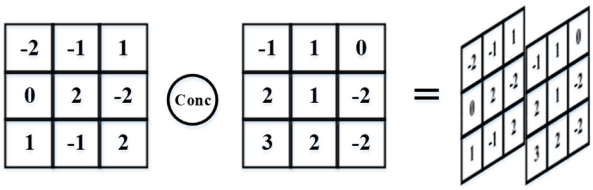


Fig. 3 Concatenate operation

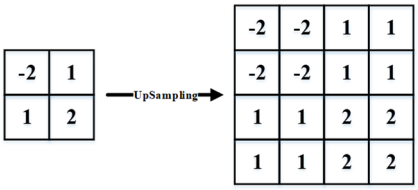


Fig. 4 UpSampling operation

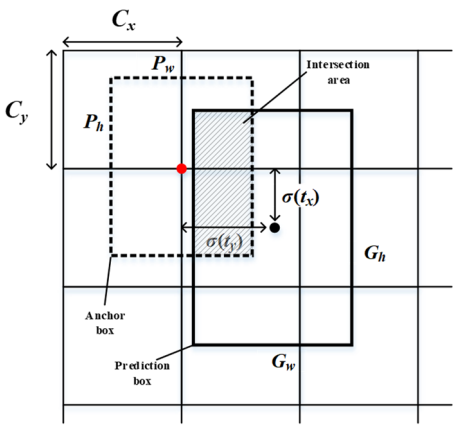


Fig. 5 Anchor box to prediction box process

Each prediction is divided into a different number of grid cells, and each grid cell predicts three bounding boxes based on anchor boxes. Each bounding box contains coordinate offsets, object confidence, and category probabilities. The coordinate offsets are transformed into the location and size under the cell diagram to calculate the loss. Fig. 5 illustrates the conversion process. The dotted rectangle is an anchor box, and the solid rectangle is a prediction box. The conversion process can be expressed as

$$G_x = \sigma(t_x) + C_x, \quad G_y = \sigma(t_y) + C_y \tag{1}$$

$$G_w = P_w \cdot e^{t_w}, \quad G_h = P_h \cdot e^{t_h} \tag{2}$$

where t_x, t_y, t_w, t_h are the four coordinate offsets; C_x, C_y are offsets of the grid cell, where the center of the indoor object or human face is located, from the top left corner of training image; the center coordinates (G_x, G_y) of the predicted box are obtained using sigmoid function $\sigma(.)$; the width G_w and height G_h of the predicted box are regressed from the anchor box with width P_w and height P_h.

During training, YOLOv3 optimizes a multi-part loss function, which consists of localization loss, confidence loss, and classification loss. The localization loss is calculated by the sum-squared error, and the confidence loss and classification loss are calculated by the binary cross-entropy function. Only the predictions of the responsible anchor which has the highest intersection over union (IoU) with the ground-truth box are used for calculating localization and classification loss. Fig. 5 shows an illustration of IoU, which is equal to the intersection area divided by the union area of the anchor box and prediction box. For confidence loss, the predictions of the responsible anchor and anchors that overlap ground truth by less than a certain threshold value are considered.

This study builds a YOLOv3 model on the platform of Keras for indoor object detection. The output layers of the model are determined according to the confidence, coordinates, and category of indoor objects. The training process takes the indoor object images as input to optimize weight parameters and minimizes the multi-part loss. After training, the model can detect the location of each target in a survey image. Each indoor object is then cropped separately according to its coordinates to reduce background interference. The cropped image is used by a web crawler to search the object's weight. A similar procedure is taken to establish the face detection model whose output is the number of persons in an image. The training details of the two models will be explained later.

2.3 Web crawler

In this era of big data, data have become essential in industrial and academic research. Web crawler is an efficient information collection tool that can quickly and accurately collect various data (Kumar *et al*. 2017). In this study, weight information mining programs have been built based on Scrapy (Wang and Guo 2012), which is an asynchronous processing and extensible crawler framework. Fig. 6 shows the overall architecture of Scrapy (Wang and Guo 2012) used for the data collection process.

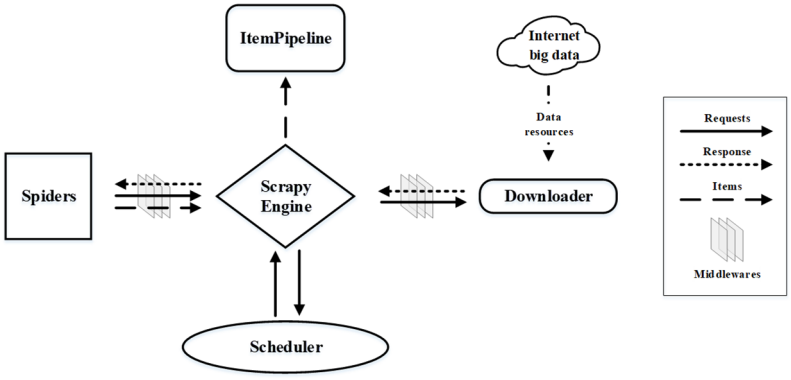


Fig. 6 Architecture of scrapy

Data sources to crawl for weight information are determined, and the initial requests are built in the Spiders component. The Engine receives initial requests and schedules them in the Scheduler to generate the next requests. The Scheduler returns the next requests to the Engine, and the Engine sends the requests to the Downloader through the Middlewares. The Downloader downloads data from Internet big data to generate responses and the Engine receives the responses through the Middlewares. The Spider receives the responses and returns the items (e.g., weight information, material, images) and new requests to the Engine through the Middlewares. The Engine sends items to the ItemPipeline for cleaning and requests to the Scheduler. The Engine asks for the next requests from the Scheduler to begin the next iteration. The above steps are repeated until there are no requests in the Scheduler.

The data source is determined according to the object features obtained earlier. Specific crawling steps to obtain weight information for objects with different features is presented later.

3. Implementation details

Currently, no universal indoor objects dataset exists. A dataset of indoor objects in a meeting room was collected for this study, and an existing benchmark face dataset was also used to explain how to perform the DL-based live load survey method.

3.1 Image acquisition and labeling

An iPhone XS Max with a resolution of 4032 × 3024 pixels was used to obtain 3517 raw images of sustained live load in a meeting room. These images contain five classes of indoor objects, namely two types of chairs (Fig. 7(1)-(2)), a type of conference table (Fig. 7(3)), a solid wood desk (Fig. 7(4)), and a water dispenser (Fig. 7(5)). Images were taken at multiple scales and from different perspectives. A single image can contain one or more object types. Note that no restrictions were imposed on distance or illumination uniformity. FDDB (Jain and Learned-miller 2010), a dataset of facial images with accurate annotations, was used to calculate temporary crowd loads. FDDB includes 2845 images with a total of 5171 faces that present a wide range of difficulties, such as low resolution, occlusions, and different poses.

Ground truth annotations are necessary for supervised learning procedures and were implemented by two procedures. For raw images of sustained live load, LabelImg was used to assign rectangular ground truth bounding boxes to corresponding objects. For FDDB images, because the facial annotations provided are ellipsoid parameters, a script based on Python was built to convert the ellipsoid annotations to rectangular annotations. Fig. 7 shows representative raw images and labeling of ground truth regions.

3.2 Data augmentation

Deep neural networks typically need a large amount of training data to achieve the desired effectiveness. In situations where the data volume is limited, data augmentation can increase the data volume, improve the robustness of the model, and avoid overfitting. In this study,

Fig. 7 Representative raw images with annotations

(a) Raw image (b) Flip horizontal (c) Zoom in

(d) Zoom out (e) Shift horizontal (f) Shift vertical

Fig. 8 Data augmentation applied to one raw image

horizontal flipping, zooming in or out, and shifting were used to augment the data. Each image was randomly scaled, shifted, and flipped to increase the diversity and quantity of data before being forward propagated. These operations make the model more robust to changes in object size and position. Note that the ground truth annotations should change with the augmentation approach. Fig. 8 shows the scaling, shifting, and flipping operations performed to one raw image.

3.3 Hyper-parameters settings

In neural networks, there are artificially adjusted hyper-parameters in addition to the learnable weight parameters. These hyper-parameters affect the time and memory costs, the quality of the recovered model, and the ability of the model to infer the correct results. In this study, the settings of hyper-parameters mainly refer to recommended values in the classic literature.

The dimensions and number of convolutional kernels, depth of the network, and strides were determined as described in Section 2.2. The small constant ε added to the mini-batch variance for numerical stability in BN is set to 1e-3. The exponential decay rates for moment estimates ρ1, ρ2 and small constant η in Adam optimization algorithm are set to 0.9, 0.999, and 1e-8, respectively (Kingma and Ba 2015). The weight decay to prevent overfitting is set to 5e-4 (Redmon and Farhadi 2017). The learning rate directly controls the magnitude of the network gradient update during training. In this study, the learning rate α is initially set to 0.001, and then adaptively adjusted according to the optimization status of each parameter (Kingma and Ba 2015). In addition, the loss of the model on the validation set during the training process is monitored as follows: If the loss of the model on the validation set in three epochs is less than 0.0001, then the learning rate is reduced to 0.1 times the previous value. If the loss of the model on the validation set does not decrease over ten consecutive epochs, then the training is terminated to prevent overfitting and ensure the effectiveness of the training. The size of the mini-batch is set to 64 to provide a more accurate estimate of the gradient and take advantage of multicore architectures. As for the weight parameters initialization, the parameters of the feature extractor are initialized from the pre-trained weight parameters of the COCO dataset (Lin *et al*. 2014), and the remaining weight parameters are initialized from He-Uniform initialization.

3.4 Data crawling

The crawling to collect weight data was divided into two procedures according to the object features, and the database used to store data was MongoDB.

For objects with direct features, object keywords were extracted based on load survey data. Based on these keywords, all data of matching objects on the corresponding website, including attribute information and images were crawled, and the crawled data were then mined to extract the target weight information. The crawling process is shown in Fig. 9(a). Steps 1-14 were repeated until there were no requests in the Engine. For objects with indirect features, the target objects in the load survey acquisition images were detected and cropped using the indoor objects detection model. The cropped target images were automatically uploaded to the e-commerce database based on the Selenium library for image matching, and the details of the retrieved items were downloaded and the data cleaned to obtain weight information of each target object. The data crawling process is shown in Fig. 9(b).

4. Results and discussions

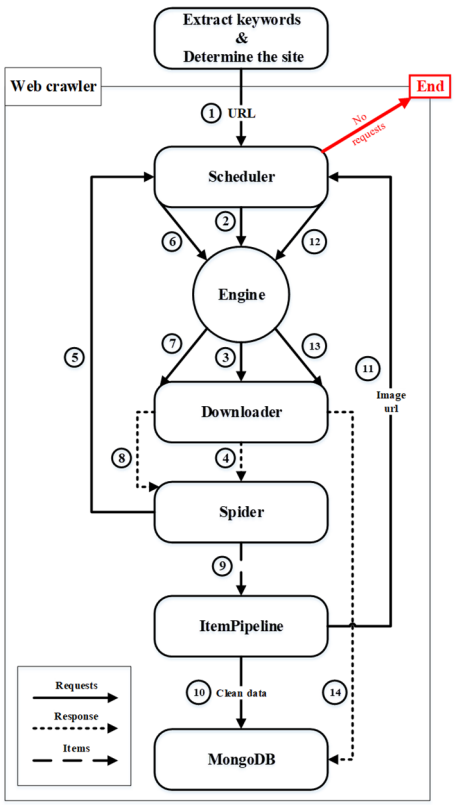
In this paper, 3517 raw images containing 5 classes of indoor objects and 2845 raw images containing 5171 faces are used to train indoor objects and face detection models, respectively. The training, validation, and test sets of both models are independent and randomly sampled from the corresponding image set with proportions of 80%, 10%, and 10%. Data augmentation is performed for each image before forward propagation.

4.1 Anchor box clustering

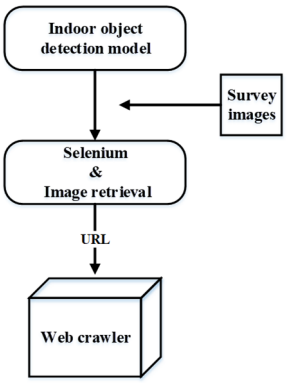
Each scale feature map corresponds to three anchor boxes. In this work, KMeans clustering is conducted on the ground truth boxes of the training set to find anchor boxes. The distance metric based on IoU (Redmon and Farhadi 2017) is defined as

$$D(\boldsymbol{b}, \boldsymbol{p}) = 1 - \mathrm{IoU}(\boldsymbol{b}, \boldsymbol{p}) \tag{3}$$

where D is a distance metric between ground truth boxes and anchor boxes, $\boldsymbol{b}$ represents all ground truth boxes in the training set, and $\boldsymbol{p}$ represents all anchor boxes. Note that the absolute positions of the ground truth and anchor boxes are not considered during the clustering process; only the IoU is considered. The clustering results of the indoor objects and FDDB training sets are shown in Fig. 10. The horizontal and vertical coordinates represent the width and height of the ground truth box. The red pentagrams represent the cluster centers, and points of the same color belong to the same cluster. The data points of the indoor object dataset

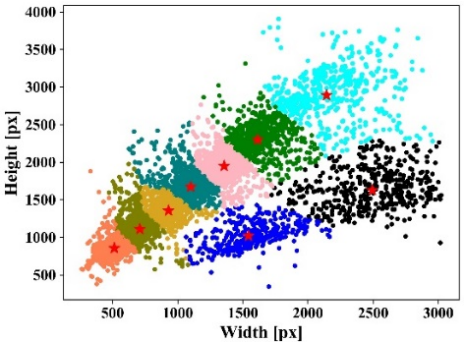


(a) Keyword-based web crawler

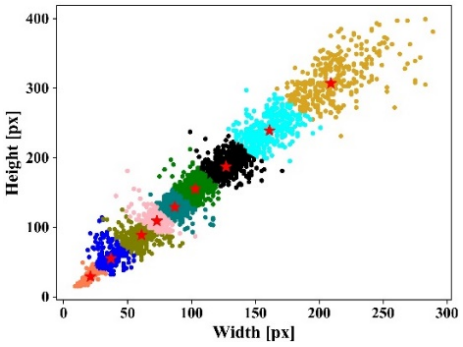


(b) Image-based web crawler

Fig. 9 Flow chart of data crawling process

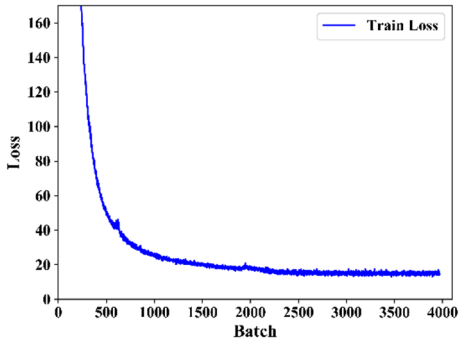


(a) Indoor objects training set

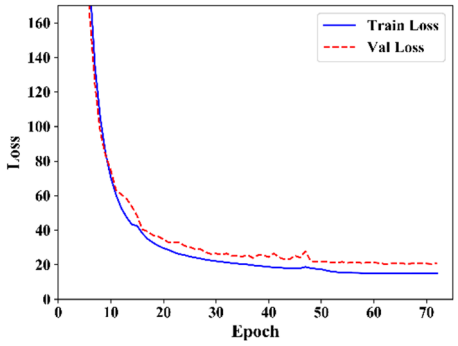


(b) FDDB training set

Fig. 10 Anchor box clustering results

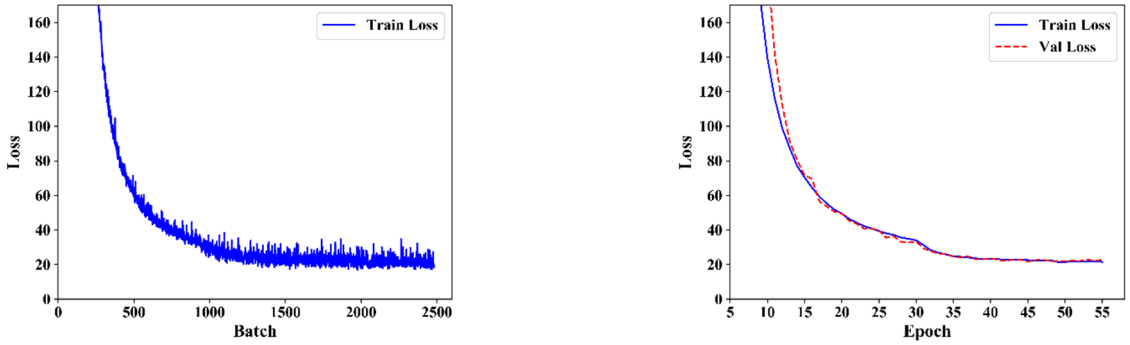


(a) training set



(b) training and validation sets

Fig. 11 The loss values in the indoor objects detection model



(a) training set (b) Training and validation sets

Fig. 12 The loss values in the face detection model

Table 2 Anchor box size in KMeans clustering results

Feature map	Anchor boxes [px]			
	Indoor objects training set		FDDB training set	
	Width	Height	Width	Height
13 × 13	2493	1629	209	309
	2143	2893	161	241
	1614	2300	129	191
26 × 26	1543	1021	105	159
	1357	1950	89	133
	1100	1674	75	111
52 × 52	929	1357	61	91
	709	1110	37	55
	514	857	21	29

are more scattered, which means that the aspect ratio of the objects in the data set is more variable.

Table 2 shows the anchor box sizes in the KMeans clustering results. The average IoU of nine anchor boxes and ground truth boxes in the two training sets are 84.59% and 85.55%, respectively. Based on these anchor boxes, the training of the indoor objects and face detection models was conducted.

4.2 Training results

The training process of both models was as follows: Pre-training weights based on the COCO dataset and He-Uniform initialization were used to initialize the network parameters. The input size of the images was set to 416 × 416 pixels, and data augmentation was performed. The training was divided into two stages. The parameters of the last 67 layers were optimized with a batch size of 64 in the first stage, and the parameters of the entire network were fine-tuned with a batch size of 32 in the second stage. All experiments were performed with the hardware of GeForce RTX 2080Ti graphics processing unit (GPU) and Intel Xeon Silver 4110 central processor unit (CPU), and the software of Ubuntu 16.04 operating system with Keras 2.2.4 as the programming platform.

It took 11.5 hours (72 epochs and 3970 batches) and 3.5 hours (55 epochs and 2480 batches), respectively, to train the indoor objects and face detection model on the hardware described above. Figs. 11-12 show the changes in the loss for the indoor objects and face detection models, respectively, during training. Fig. 11(a) and Fig. 12(a) show that with an increase in the number of batches, the loss gradually decreases and stabilizes for each model. However, there is a phenomenon of local oscillation, which can be mitigated by increasing the batch size. Fig. 11(b) and Fig. 12(b) show that with an increase of epochs, the training set loss and the validation set loss have essentially identical downward trends, and the difference between the two is negligible once they stabilize. This indicates that the models have effectively learned according to the training samples, and the parameters of the network have been optimized.

Using the indoor objects detection model as an example, the activation feature maps and convolution kernels are visualized to access the inner learned features of the CNNs. Fig. 13 illustrates the activation feature maps based on a raw image and convolution kernels for a few layers. Shallow convolution kernels focus on the edge and texture features, so the edge contour of the object is prominent in the shallow activation feature maps. As the layer depth increases, the visual patterns of the convolution kernels become more complicated, and the activation feature maps become more abstract. Eventually, the model pays more attention to the location information of objects, which shows that the network has effectively learned to detect objects.

4.3 Test results and evaluation metrics

Test images were input into the trained network to evaluate the performance of the indoor objects and face detection models. Figs. 14(a)-(d) shows the identification results of the indoor objects detection model. The model correctly outputs the class of the object, along with an accurate location. The model is robust for images with insufficient lighting (d), object truncation (a)-(c), different angles (a), and small objects (c). Figs. 14(e)-(f) shows the identification results of the face detection model. The model can accurately detect faces on people in different poses and attire. However, when a major portion of an object is truncated, or a face is severely occluded, the models fail to

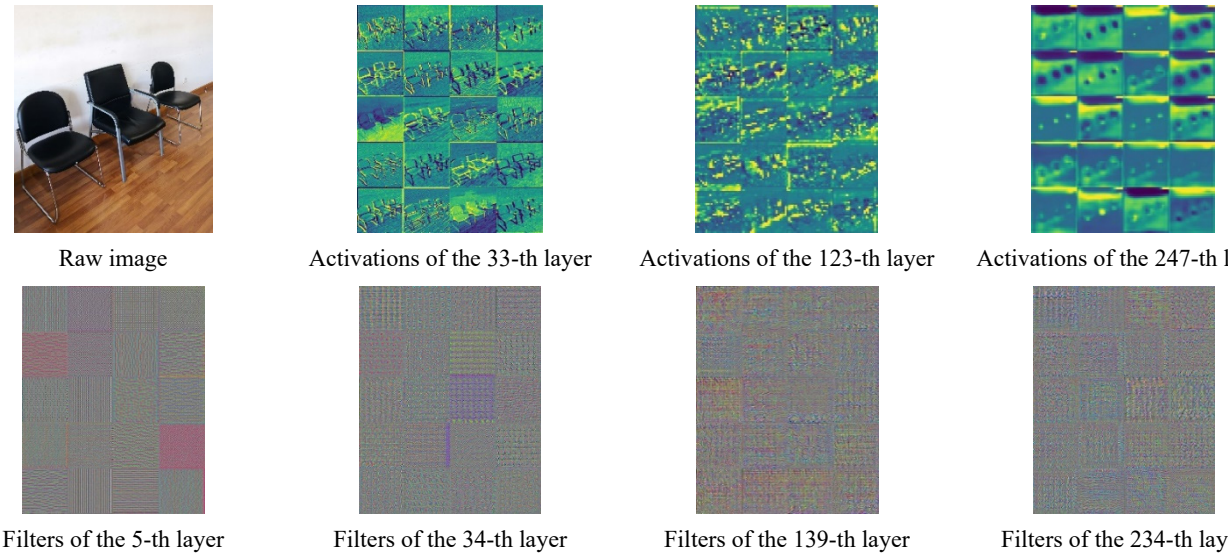


Fig. 13 Visualization of activation feature maps and convolution filters of the indicated layers

Fig. 14 Detection results of indoor objects and face detection models

detect.

In addition, the average precision (AP) and mean average precision (mAP) metrics are calculated to evaluate the models. Precision is the percentage p of correct positive predictions in all detection results. Recall r is the percentage of true positives detected among all relevant ground truths. The precision-recall (P-R) curve indicates the performance of an object detector. AP is defined as the area under the precision-recall curve. In this study, AP is computed using all data points (Everingham *et al*. 2010).

$$AP = \sum_{r=0}^{1} (r_{n+1} - r_n) \cdot \max_{\bar{r} \geq r_{n+1}} p(\bar{r}) \tag{4}$$

where $p(\bar{r})$ represents the measured precision at recall $\bar{r}$. The mAP is calculated by taking the mean AP over all classes. Fig. 15 shows the P-R curves at Igou = 0.75 for all classes. Along with maintaining a high accuracy rate, various classes have a good recall rate. For the face detection model, the ability to find all faces in the images needs to be considered, which means the recall rate is important. The maximum recall rate is 86.77%, which shows that the performance of the face detection model is good. Fig. 16 shows the AP and mAP values at Igou = 0.75 of the indoor objects detection model. Overall, the AP values for all classes reach 75%, and the mAP value is 85.70%, which indicates that the performance of the indoor objects detection model is acceptable.

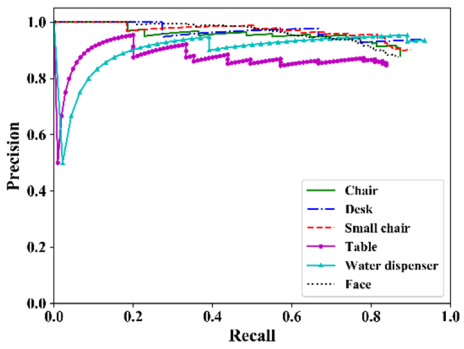


Fig. 15 P-R curves for the indicated classes

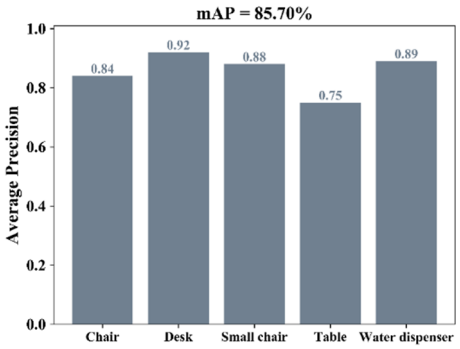


Fig. 16 AP and mAP for indoor objects detection model

(a) No crowd

(b) With crowd

Fig. 17 Meeting room images

4.4 A case study

With the detection models and the web crawler framework established, the live load of rooms can be surveyed. Taking a meeting room shown in Fig. 17 as an example, the survey is carried out by an investigator with a smartphone (iPhone XS Max). The length and width of the room measured with the smartphone are 6.2 m and 6.3 m, respectively. For the water dispenser that has direct features, a picture of the product identification behind it is taken. For furniture with indirect features, several clear pictures containing the target objects are taken. When taking the picture, mutual occlusions of objects should be avoided. The number of various objects is recorded by voice, and the content of the voice is 'The investigation begins! There is a desk, 6 tables, 13 chairs, 14 small chairs, and a water dispenser in the room.' (The original sound was in Chinese). Fig. 18 shows the waveform diagram of the recorded voice. For the crowd, a picture that includes all faces is taken. At this point, the load survey is complete and it only takes 2-3 minutes.

The collected information is processed using different technologies. The water dispenser's product identity is first obtained by OCR. It is then used by the crawler as keywords to parse three web page-level requests to obtain weight information automatically from the manufacture's official website. For furniture, the indoor objects detection model is used to isolate the target object from the background and other objects. The crawler is then used to perform image retrieval on some e-commerce websites, and four level requests are parsed. For the crowd, the face detection model is used to obtain the number of people in

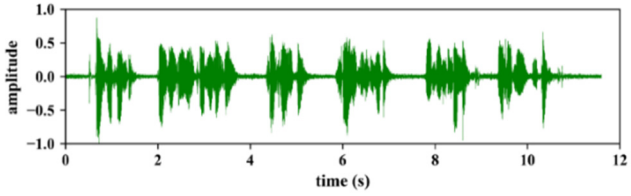


Fig. 18 Waveform diagram of recorded voice data

the crowd and then multiplied by an average weight value to calculate the load of the crowd.

There are 16 data values for the water dispenser. After keyword filtering, only one value meets the requirements, and that weight is 5.1 kg. As for furniture, the web crawling returns different weight values for objects with similar appearances. After removing some obvious irrational values (e.g., 0.1 kg for a chair), the average value of the remaining results is taken as the final weight. There are 91 data values for four types of furniture, and a few matching results are shown in Fig. 19. Fig. 20 shows the distribution of the data, and the horizontal lines represent the average weight of different furniture. The average weights of a chair, small chair, table, and desk are 9.2, 4.8, 50.0, and 37.1 kg, respectively. The numbers for each furniture type are obtained by speech recognition. The numbers for chairs, small chairs, tables, and desks are 13, 14, 6, and 1, respectively. The number of people in the crowd was correctly counted by the face detection model, as shown in Fig. 21. To calculate the total weight, the recommended average adult weight (60 kg) in the Chinese load code is used in this study.

Detection image

Images cropped by the detection model

Matching results using cropped images

Fig. 19 Representative matching results for furniture

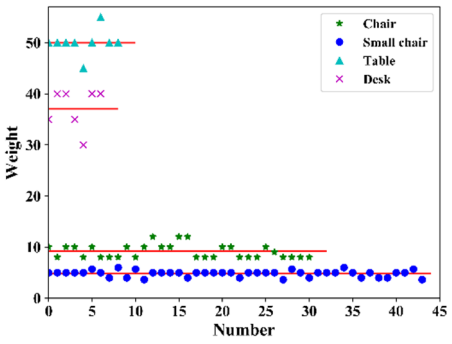


Fig. 20 Distribution of furniture data

Fig. 21 Detection results of crowd

Table 3 Comparison between the DL-based method and actual weight results

Items	Technologies	Number	DL-based method		Actual weight	
			Single (kg)	Total (kg)	Single (kg)	Total (kg)
Table	Object detection Speech recognition Web crawling	6	50.0	300.0	45.9	275.4
Desk		1	37.1	37.1	35.7	35.7
Small chair		14	4.8	67.2	5.0	70.0
Chair		13	9.2	119.6	10.4	135.2
Crowd	Face detection	10	60.0	600.0	71.8	717.5
Water dispenser	OCR	1	5.1	5.1	5.0	5.0
Area	Smartphone	-	39.1 m^2		41.0 m^2	
Total sustained live load (kN)			5.3		5.2	
Total temporary live load (kN)			6.0		7.2	
Sustained live load per unit area (N/m^2)			135.5		126.8	

Table 3 shows a comparison between the DL-based method and actual weight results. For the sustained live load, the DL-based method gives reasonable results in an error range between 3.9% (desk) to 11.5% (chair). The total weight detected is 5.3 kN that is very close to the real value of 5.2 kN. For the crowd load, the actual average weight of the crowd in the meeting room is 72 kg, which is larger than the value (60 kg) used in the load surveys of the last century. It is acceptable since living standards have been continuously improved. The final obtained sustained live load per area is 6.9% higher than the actual result. Considering the degree to which time and labor costs are decreased relative to the traditional method, the survey results of the DL-based method are acceptable.

5. Conclusions

5.1 Concluding remarks

The modeling of the live load is crucial for structural reliability analysis but is also probably the weakest part owing to many problems of the traditional survey method, such as labor-intensive, low efficiency, high cost, limited samples, and difficulties in weighing items on-site. This study suggests a novel structure live load survey method that takes full advantage of the new-developed deep learning and immense Internet data source.

The proposed method utilizes multi-source heterogeneous data such as images, voice, and product identities to obtain the weight of indoor items without weighing them. Various technologies are used to obtain the weight automatically, according to the item characteristics. For items belonging to sustained live load, object detection, web crawler, optical character recognition, and speech recognition technologies are used to obtain the weight in combination with Internet big data. For the crowd that belongs to the temporary live load, a face detection model is used to count the number of persons by which the total crowd weight can be estimated. A case study on a real meeting room shows that the weight obtained by the DL-method is very close to the real value. Moreover, one investigator with a normal smartphone can conduct a quick on-site survey at a low cost, providing all the required information for live load calculation.

5.2 Limitations and further work

Although it is convenient and easy to apply, the proposed DL-method still has several limitations that need further improvement:

- The new survey method actually assumes that the weight information of an object is somewhere on the Internet that can be located by various labels of the object. In this regard, new advanced web-searching technique better than web crawler, especially techniques based on natural language processing, is always needed to further improve the accuracy and success rate of web searching.
- Since different objects with different weights may have similar appearances, an advanced object detection method that can involve more image features of an object is also needed to further reduce the detection error. Moreover, developing a database containing weight information of common objects, as many as possible, might be another way to tackle this issue.
- Currently, we avoid mutual occlusions of objects in an image by asking the investigator to take clear images of each object. Since only five classes of indoor objects were involved in the case study here, it was not difficult to follow these rules. However, for complex indoor scenes having more classes of objects, a new detection model for multiple object detection and with greater generalization ability is necessary to availably solve the problems. Consequently, the on-site investigation efficiency and applicability of the method can be further improved.

Acknowledgments

This research project was financially supported by National Natural Science Foundation of China (Grant No. 52178151); State Key Laboratory for Disaster Reduction of Civil Engineering (Grant No. SLDRCE19-B-22); and Shanghai TCM Chronic Disease Prevention and Health Service Innovation Center (Grant No. ZYJKFW201811009).

References

Andam, K.A. (1986), "Floor live loads for office buildings", *Build. Environ.*, **21**(3-4), 211-219. https://doi.org/10.1016/0360-1323(86)90032-6

Asantey, S.B.A. and Andam, K.A. (1996), "Factory and warehouse live load survey", *Build. Environ.*, **31**(2), 167-178. https://doi.org/10.1016/0360-1323(95)00035-6

Bahdanau, D., Cho, K. and Bengio, Y. (2014), "Neural machine translation by jointly learning to align and translate", *Proceedings of International Conference on Learning Representations*, San Diego, CA, USA, May.

Cha, Y.-J., Choi, W. and Büyüköztürk, O. (2017), "Deep learning-based crack damage detection using convolutional neural networks", *Comput.-Aid. Civil Infrastruct. Eng.*, **32**(5), 361-378. https://doi.org/10.1111/mice.12263

Choi, E.C.C. (1990), "Live load for office buildings: effect of occupancy and code comparison", *J. Struct. Eng.*, **116**(11), 3162-3174. https://doi.org/10.1061/(ASCE)0733-9445(1990)116:11(3162)

Choi, E.C.C. (1991), "Extraordinary live load in office buildings", *J. Struct. Eng.*, **117**(11), 3216-3227. https://doi.org/10.1061/(ASCE)0733-9445(1991)117:11(3216)

Duan, Y., Chen, Q., Zhang, H., Yun, C., Wu, S. and Zhu, Q. (2019), "CNN-based damage identification method of tied-arch bridge using spatial-spectral information", *Smart Struct. Syst., Int. J.*, **23**(5), 507-520. https://doi.org/10.12989/sss.2019.23.5.507

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A. (2010), "The pascal visual object classes (voc) challenge", *Int. J. Comput. Vision*, **88**(2), 303-338. https://doi.org/10.1007/s11263-009-0275-4

Ge, S.J., Chen, H., Sun, Z.S. and Li, J.B. (2008), "Survey and statistic of floor live load of residential building in central plains region", *Build Struct.*, (07), 125-128. https://doi.org/10.19701/j.jzjg.2008.07.038

Graves, A., Mohamed, A. and Hinton, G. (2013), "Speech recognition with deep recurrent neural networks", *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, May.

Guo, A., Jiang, A., Lin, J. and Li, X. (2019), "Data mining algorithms for bridge health monitoring: kohonen clustering and lstm prediction approaches", *J. Supercomput.*, **76**(2), 932-947. https://doi.org/10.1007/s11227-019-03045-8

He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, July.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P. and Kingsbury, B. (2012),

"Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups", *IEEE Signal Process. Mag.*, **29**(6), 82-97. https://doi.org/10.1109/msp.2012.2205597

Ioffe, S. and Szegedy, C. (2015), "Batch normalization: accelerating deep network training by reducing internal covariate shift", *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, July.

Jain, V. and Learned-Miller, E. (2010), "Fddb: a benchmark for face detection in unconstrained settings", UMass Amherst Technical Report; University of Massachusetts Amherst.

Jin, X. and Zhao, J. (2012), "Development of the design code for building structures in China", *Struct. Eng. Int.*, **22**(2), 195-201. https://doi.org/10.2749/101686612X13291382990886

Kaimal, J.C., Wyngaard, J.C., Izumi, Y. and Coté, O.R. (1972), "Spectral characteristics of surface-layer turbulence", *Q. J. R. Meteorol. Soc.*, **98**(417), 563-589. https://doi.org/10.1002/qj.49709841707

Kingma, D.P. and Ba, J. (2015), "Adam: a method for stochastic optimization", *Proceedings of International Conference on Learning Representations 2015*, San Diego, CA, USA, May.

Kumar, S. (2002a), "Live loads in office buildings: lifetime maximum load", *Build. Environ.*, **37**(1), 91-99. https://doi.org/10.1016/S0360-1323(00)00075-5

Kumar, S. (2002b), "Live loads in office buildings: point-in-time load intensity", *Build. Environ.*, **37**(1), 79-89. https://doi.org/10.1016/S0360-1323(00)00074-3

Kumar, M., Bhatia, R. and Rattan, D. (2017), "A survey of web crawlers for information retrieval", *Wiley Interdiscip. Rev.-Data Mining Knowl. Discov.*, **7**(6), e1218. https://doi.org/10.1002/widm.1218

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P. and Zitnick, C.L. (2014), "Microsoft coco: common objects in context", *Proceedings of European Conference on Computer Vision*, Zurich, Switzerland, September.

Luo, L., Feng, M.Q., Wu, J. and Leung, R.Y. (2019), "Autonomous pothole detection using deep region-based convolutional neural network with cloud computing", *Smart Struct. Syst., Int. J..*, **24**(6), 745-757. https://doi.org/10.12989/sss.2019.24.6.745

Maas, A.L., Hannun, A.Y. and Ng, A.Y. (2013), "Rectifier nonlinearities improve neural network acoustic models", *Proceedings of the 30th Workshop on Deep Learning for Audio, Speech and Language Processing*, Atlanta, GA, USA, June.

Ni, F., Zhang, J. and Chen, Z. (2018), "Pixel-level crack delineation in images with convolutional feature fusion", *Struct. Control Health Monitor.*, **26**(1), e2286. https://doi.org/10.1002/stc.2286

Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K. (2016), "Wavenet: a generative model for raw audio", arXiv preprint arXiv:1609.03499.

Park, S., Jeong, H., Min, H., Lee, H. and Lee, S. (2018), "Wavelet-like convolutional neural network structure for time-series data classification", *Smart Struct. Syst., Int. J.*, **22**(2), 175-183. https://doi.org/10.12989/sss.2018.22.2.175

Redmon, J. and Farhadi, A. (2017), "YOLO9000: better, faster, stronger", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July.

Redmon, J. and Farhadi, A. (2018), "YOLOv3: an incremental improvement", arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R. and Sun, J. (2016), "Faster r-cnn: towards real-time object detection with region proposal networks", *IEEE Trans. Pattern. Anal. Mach. Intell.*, **39**(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

Ruiz, S.E. and Sampayo-Trujillo, A. (1997), "Design live loads for classrooms in United States and Mexico", *J. Struct. Eng.*, **123**(12), 1652-1657. https://doi.org/10.1061/(ASCE)0733-9445(1997)123:12(1652)

Tang, Z., Chen, Z., Bao, Y. and Li, H. (2018), "Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring", *Struct. Control Health Monitor.*, **26**(1), e2296. https://doi.org/10.1002/stc.2296

Wang, J. and Guo, Y. (2012), "Scrapy-based crawling and user-behavior characteristics analysis on taobao", *Proceedings of 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Sanya, China, October.

Wang, D. and Li, J. (2012), "A random physical model of seismic ground motion field on local engineering site", *Sci. China: Technol. Sci.*, **55**(7), 2057-2065. https://doi.org/10.1007/s11431-012-4850-5

Wu, X.Q., Yao, J.T. and Liu, Y.J. (2012), "Statistical analysis of live load on residence floor and analysis of residence floor reliability", *Eng. Mech.*, **29**(3), 90-94.

Xie, P., Zhou, A. and Chai, B. (2019), "The application of long short-term memory (LSTM) method on displacement prediction of multifactor-induced landslides", *IEEE Access*, **7**, 54305-54311. https://doi.org/10.1109/access.2019.2912419

Xiong, J. and Chen, J. (2019), "A generative adversarial network model for simulating various types of human-induced loads", *Int. J. Struct. Stab. Dyn.*, **19**(08), 1950092. https://doi.org/10.1142/s0219455419500925

Xu, Y., Bao, Y., Chen, J., Zuo, W. and Li, H. (2019), "Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images", *Struct. Health Monitor.*, **18**(3), 653-674. https://doi.org/10.1177/1475921718764873

HJ