# Deep Learning Approach for Estimating Workability of Self-Compacting Concrete from Mixing Image Sequences 

Zhongcong Ding and Xuehui An (D)<br>State Key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing 100084, China<br>Correspondence should be addressed to Xuehui An; anxue@mail.tsinghua.edu.cn

Received 23 May 2018; Accepted 25 October 2018; Published 25 November 2018
Academic Editor: Barbara Liguori
Copyright © 2018 Zhongcong Ding and Xuehui An. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a deep learning approach to better utilize the spatial and temporal information obtained from image sequences of the self-compacting concrete- (SCC-) mixing process to recover SCC characteristics in terms of the predicted slump flow value (SF) and V-funnel flow time (VF). The proposed model integrates features of the convolutional neural network and long short-term memory and is trained to extract features and compute an estimate. The performance of the method is evaluated using the testing set. The results indicate that the proposed method could potentially be used to automatically estimate SCC workability.

## 1. Introduction

Predicting the workability of self-compacting concrete (SCC) during the mixing process is an important research problem in construction engineering. SCC is a highly workable material that can flow to fill gaps in reinforcements, corners of moulds, and voids in rock blocks, without exhibiting any vibration and compaction during the placement process [1-3]. To construct strong and durable structures, the SCC must be of superior workability [4].

Slump flow and V-funnel tests [1,5] are commonly used in estimating SCC workability. Slump flow values (SF) indicate the deformability of SCC, while the V-funnel flow time (VF) indicates the viscosity and segregation resistance ability. However, these tests present a number of problems in practice. One is that the tests must be performed before concrete placement to ensure that the SCC has good workability; this is expensive in terms of labour and time. Another problem is that SCC workability can only be determined immediately before placement. When the concrete is not a qualified SCC, the mixture is wasted and engineers need to change the mix design.

The problems mentioned above can be solved if the SCC workability can be estimated during the mixing process. Chopin proposed that a concrete mixer can be considered as
a large rheometer [6], allowing SCC rheological parameters to be determined by considering fresh concrete as a Bingham material. However, the validity of using a mixer as a reliable rheometer remains an open question. Other methods have been proposed that perform estimation using image processing. This approach is based on the considerations that a number of experienced engineers estimate SCC workability by watching the mixing process. Visual information during mixing is used to deduce the SCC workability characteristics. For example, Daumann and Nirschl used ultramarine blue as a tracer component and determined the homogeneity of mixtures during the concrete-mixing process [7]. Li et al. developed an image processing method to determine the SF and VF values of SCC during the mixing process [8, 9]. However, the methods based on image processing rely heavily on human experience and insight. For example, features such as the tracer component in Daumann's work and boundaries extracted in Li's work had to be manually selected. Furthermore, these features are often related to a specific experimental scene, which limits the applicability of these methods.

To compensate for the drawbacks of the aforementioned methods, the dependency on human experience must be reduced, and methods that dig deep into the information hidden behind the original data must be used. An alternative
approach is to employ automatic feature learning using deep learning- (DL-) based models.

DL deals with the problem of data representation by introducing simpler intermediate representations that can be combined to build complex concepts. Therefore, no specific techniques need to be applied to extract features that represent the image data [10-12]. However, because of its complex structure, DL needs a large volume of data to generate models with high predictive performance and, consequently, has high computational cost.

Deep convolutional neural networks (CNN) are a successful DL model that provides an extremely powerful tool for learning visual representations. The success of Krizhevsky et al. in the ImageNet classification and localization challenges [13] using solutions based on CNN drew attention to the applications of CNN. Subsequently, DL-based methods have been used to significantly improve the state of the art in image classification [14], object detection [15], and semantic segmentation [16].

Compared to CNN, recurrent neural network (RNN) models are "deep in time" and can form implicit compositional representations in the time domain [17]. A significant limitation of simple RNN models that strictly integrate state information over time is known as the "vanishing gradient" effect. This refers to the tendency to backpropagate an error signal through a long-range temporal interval, which becomes increasingly difficult to handle in practice. Long short-term memory (LSTM) units, first proposed in [18], are recurrent modules that enable long-range learning. LSTM units have a hidden state augmented with nonlinear mechanisms that allow state update, reset, and propagation without modification using simple learned gating functions. LSTMs have been recently used widely for sequential labelling [19] and provide significant improvement when ample training data are available. In [17], the authors showed that CNN with RNN units can be generally applied to visual time-series modelling.

Machine learning techniques have been successfully used in several applications in the field of construction engineering, such as automatic crack detection [20, 21], concrete failure surface modelling [22], compressive strength prediction [23-25], durability assessment [26], concrete dam reliability analysis [27], and carbonation prediction [28].

Our objective in this study was to design a method for estimating SCC workability during the mixing process. In this paper, we propose an end-to-end trainable DL architecture that combines CNN and LSTM and helps estimate SCC workability. Instead of using specific feature extractors, we train a neural network to find the relationships between image sequences and SCC workability. Data preparation is performed to make the raw data suitable for training. Using the trained model, the SCC workability characteristics can be predicted automatically. Moreover, the strategy for choosing a proper time resolution is discussed to optimize the performance of the DL method.

The remainder of this paper is organized as follows. In Section 2, we introduce the procedures of data collection, preprocessing, and expanding. In Section 3, we describe the proposed DL model in detail. Training results and discussion
are presented in Section 4, and conclusions are drawn in Section 5.

## 2. Data Preparation

Data preparation involves compiling data in a format that is suitable for use by DL approaches that estimate the workability characteristics. Owing to the nature of concrete experiments, the amount of raw data available for this task were limited and techniques were applied to generate synthetic data. Another issue encountered in this task was that of overfitting, which is quite typical in the DL field. Preprocessing methods were used to avoid this situation. Finally, an operation called data augmentation was considered to further reduce the overfitting problem, without adding new information into the model.
2.1. Data Collection. The proposed model takes image sequences of the SCC mixing process as input. The raw data comprised videos of the SCC mixing process collected from various SCC experiments in the laboratory. Figure 1 shows a typical experimental setup used for recording these videos. Figure 2 shows the utilized single-shaft mixer. The mixer had a capacity of 60 L and a fixed rotating speed of 51 rpm . A portable tripod mounted with a smartphone was placed beside the mixer. The tripod place and shooting angle were different for different experiment batches, so the perspective distortions of the videos were not the same.

During the mixing process, the smart phone was used to record a video through the opening hatch at a frame rate of 30 fps . Each video had a fixed resolution of 1,920 pixels $\times$ 1,080 pixels. After mixing and recording, traditional slump and V-funnel tests were conducted to label the mixing process videos with SF and VF values.

The cement used in all the experiments was 42.5 R Portland cement. Polycarboxylate-based superplasticizer (SP), tap water, and fine and coarse aggregates were used to batch the SCC specimens. The SP, used as a water-reducing agent, had a $20 \%$ solid content. The fine aggregates comprised quartz sands with a maximum particle size of 5 mm , and the coarse aggregates included two types of crushed stones with maximum particle sizes of 10 and 20 mm , respectively. The relative densities of the fine and coarse aggregate were 2.83 and 2.69 , respectively.

Each of the concrete mixes used in this study had a fixed fine aggregate content of $45 \%$ by volume and a fixed 20 : 10 mm coarse aggregate ratio of $1.5: 1$ by weight. The SCC volume for each experiment was 20 L . The single-shaft mixer described previously was used to mix the concrete. All the dry materials were initially mixed for 30 s . Then, water and SP were added, and the materials were wet-mixed for 240 s . The slump test and V-funnel test were conducted after the mixture was poured out. In some experiments, the produced SCC was kept in a container for a period of time, namely, $30 \mathrm{~min}, 60 \mathrm{~min}$, and 90 min . The SCC mixture was then poured into the mixer again for a mixing. This kind of videos was gathered as well. After mixing, the slump test and V-funnel test were conducted again, making sure that the


Figure 1: Schematic diagram of the apparatus: (a) side view; (b) front view.


Figure 2: The single-shaft mixer used in the experiments.
mixture appeared in each video had corresponding SF and VF values precisely.

Data comprising 31 videos of different workability characteristics were collected; the video indexes, mix composition, and workability characteristics of SCC are shown in Table 1. Different water to cement ratio by volume ( $V_{\mathrm{W}} / V_{\mathrm{C}}$ ), SP content of cement by weight (SP\%), and hold time were used to obtain SCC with different workability characteristics. The aggregate dosage was kept constant. The gravel dosage was $902.22 \mathrm{~kg} / \mathrm{m}^{3}$, while the sand dosage was $837.43 \mathrm{~kg} / \mathrm{m}^{3}$.

It is noteworthy that the SCC mixtures corresponding to videos $2,12,15$, and 20 were blocked during V -funnel tests because of large viscosities. The video data contained both spatial and temporal information, which were learned by the DL model. The frames of the videos were extracted as a number of images right after recording, as shown in Figure 3. In this study, the combination of SF and VF values were used as labels for training, which was a vector composed of two elements. When it came to the numerical label, the blocked condition in $V$-funnel tests was set to a default value of 200 s , which was large enough to be distinguished from other cases.
2.2. Preprocessing. Preprocessing methods were applied to each frame, serving two purposes. First, it substantially
reduced the resolution of the input image, thereby decreasing the computational requirements of the neural network. Second, it prevented the overfitting problem from affecting the prediction. The idea of overfitting is introduced in the next subsection. The procedure used for the preprocessing comprised the following steps: (1) conversion of RGB images to grayscale, (2) affine transformation, (3) extraction of the region of interest (ROI), and (4) histogram equalization. Figure 4 shows the entire preprocessing procedure. By taking a frame of video 19 as an example, the procedure is explained in detail next.

Several preprocessing methods can be used to prevent the overfitting phenomenon. The overfitting problem occurs when a model fits the training data too well. In other words, the model learns the details and noise in the training data to such an extent that it negatively impacts the performance of the model on new data. Noise or random fluctuations in the training data are picked up and learnt as features by the model. However, as these concepts do not apply to new data, it negatively affects the model's ability to provide accurate predictions.

For example, the SCC paste might leave some marks on the inner wall of the mixer hatch, which could be recognized as a feature of SCC by mistake, as shown in Figure 5. Overfitting is essentially caused by the small size of dataset and can be overcome by increasing the size of the training dataset.

Considering the limited size of dataset, using some preprocessing methods may also provide an alternative. For example, clipping the wall of mixer from images would prevent the learning algorithm from recognizing the redundant information in this case. Besides the detail mentioned above, colour, illumination, and perspective distortion also comprise noises that should not be learned by the model. Therefore, corresponding approaches were applied to eliminate these as well.

In recognition tasks involving natural images, colour provides extra information, and transforming images to grayscale may hurt performance [29]. However, the images used in this study mainly show the SCC mixing process,

Table 1: Mix composition and workability characteristics of mixing videos.

| Video number | $V_{\text {W }} / V_{\mathrm{C}}$ | SP (\%) | Cement (kg/m ${ }^{3}$ ) | Water (kg/m ${ }^{3}$ ) | Hold time (min) | SF (mm) | VF (s) |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | 1.10 | 0.90 | 527.74 | 183.46 | 0 | 575 | 35.8 |
| 2 | 1.10 | 0.90 | 527.74 | 183.46 | 30 | 355 | Blocked |
| 3 | 1.10 | 1.00 | 527.74 | 183.04 | 0 | 460 | 45.5 |
| 4 | 1.20 | 0.80 | 503.75 | 191.78 | 0 | 480 | 14.5 |
| 5 | 1.20 | 0.80 | 503.75 | 191.78 | 30 | 350 | 24.1 |
| 6 | 1.20 | 0.90 | 503.75 | 191.37 | 0 | 705 | 43.9 |
| 7 | 1.20 | 0.90 | 503.75 | 191.37 | 30 | 565 | 83.0 |
| 8 | 1.20 | 0.90 | 503.75 | 191.37 | 60 | 510 | 14.7 |
| 9 | 1.20 | 1.00 | 503.75 | 190.97 | 0 | 610 | 9.9 |
| 10 | 1.20 | 1.00 | 503.75 | 190.97 | 30 | 585 | 9.2 |
| 11 | 1.20 | 1.00 | 503.75 | 190.97 | 60 | 495 | 56.8 |
| 12 | 1.20 | 1.00 | 503.75 | 190.97 | 90 | 360 | Blocked |
| 13 | 1.30 | 0.90 | 481.85 | 198.60 | 0 | 610 | 18.1 |
| 14 | 1.30 | 0.90 | 481.85 | 198.60 | 30 | 485 | 69.0 |
| 15 | 1.30 | 0.90 | 481.85 | 198.60 | 60 | 340 | Blocked |
| 16 | 1.30 | 1.00 | 481.85 | 198.21 | 0 | 700 | 10.2 |
| 17 | 1.30 | 1.00 | 481.85 | 198.21 | 0 | 690 | 9.9 |
| 18 | 1.30 | 1.00 | 481.85 | 198.21 | 30 | 690 | 52.9 |
| 19 | 1.30 | 1.00 | 481.85 | 198.21 | 60 | 655 | 30.8 |
| 20 | 1.30 | 1.00 | 481.85 | 198.21 | 90 | 440 | Blocked |
| 21 | 1.11 | 0.85 | 525.24 | 184.50 | 0 | 625 | 26.0 |
| 22 | 1.11 | 0.90 | 525.24 | 184.29 | 0 | 675 | 13.0 |
| 23 | 1.13 | 0.85 | 520.31 | 186.12 | 0 | 598 | 16.4 |
| 24 | 1.13 | 0.90 | 520.31 | 185.91 | 0 | 675 | 13.0 |
| 25 | 1.15 | 0.85 | 515.47 | 187.72 | 0 | 600 | 13.0 |
| 26 | 1.15 | 0.88 | 515.47 | 187.61 | 0 | 679 | 31.0 |
| 27 | 1.15 | 0.90 | 515.47 | 187.51 | 0 | 648 | 13.0 |
| 28 | 1.17 | 0.85 | 510.71 | 189.28 | 0 | 680 | 11.6 |
| 29 | 1.17 | 0.90 | 510.71 | 189.08 | 0 | 668 | 9.8 |
| 30 | 1.18 | 0.85 | 508.37 | 190.05 | 0 | 668 | 64.0 |
| 31 | 1.20 | 0.90 | 503.75 | 191.37 | 0 | 710 | 52.0 |



Figure 3: The frames extracted from videos.
which is an industrial scene. In each image, the colour of concrete was influenced by the camera's parameters and illumination condition. Therefore, colour rarely contributed valuable information that helped distinguish workability
characteristics. Besides, RGB images comprised threedimensional inputs of $R, \mathrm{G}$, and B channels to the model, which resulted in an increased computational cost for training. Converting all the images in the sequence to


Figure 4: Flowchart for preprocessing images.


Figure 5: Overfitting example: (a) the mark being learned as the feature of $\mathrm{SF}=340 \mathrm{~mm}$; (b) resulting inaccuracy in prediction of image of the same SF value.
grayscale helped reduce complexity. The conversion was performed using equation (1):

$$
\begin{equation*}
\text { grayscale }=0.299 \times R+0.587 \times G+0.114 \times B \tag{1}
\end{equation*}
$$

The effect of this preprocessing is shown in Figure 6. The conversion was performed without resizing the images.

As shown in Figure 7, the tripod placement and shooting angle differed for different experiment batches, leading to differences in the perspective distortions of the videos. The distortion fluctuations must not be learned by the model as features. To avoid this, affine transformation processing was performed.

The affine transformation technique is typically used to correct geometric distortions or deformations that occur owing to nonideal camera angles. It was appropriate for use in this case as the transformation preserves collinearity and ratios of distances. Transforming and fusing the images to a large, flat coordinate system helped eliminate distortion and enabled easier interactions and calculations that did not require accounting for image distortion.

In this study, the transformation target was set to a rectangle of fixed size 350 pixels $\times 200$ pixels, removing experimenter and environment. The coordinates of the four corners of the mixer hatch were then recognized.


Figure 6: Converting the RGB images to grayscale.


Figure 7: Different tripod placement and shooting angles cause different distortions.

The transformation matrix was computed based on the mapping between the original coordinates and target size. As a result, the grayscale image was finally converted into a subimage containing the mixer inner wall, rotating blades, shaft, and moving SCC. After the transformation, useless visual information was partly eliminated from the processed images. Figure 8 shows an example of the affine transformation.

As noted above, the SCC paste might leave some marks on the inner wall of the mixer hatch, which could be erroneously recognized as a feature of the SCC. The simplest way to avoid this problem involved extracting the ROI without mixer wall from the affined images. This operation comprised three steps.

First, the three reference points used for extraction were identified. Assume that point A represents the bottom left corner of the mixer wall and point $B$ is on the bottom right corner of the mixer wall. Let point $C$ denote the bottom edge of the extracted region, which is defined as a point on the top edge of the rotating shaft.

The ROI was extracted using equation (2):

$$
\left\{\begin{array}{l}
X=x_{\mathrm{A}},  \tag{2}\\
Y=y_{\mathrm{A}}, \\
\text { height }=x_{\mathrm{C}}-x_{\mathrm{A}}, \\
\text { width }=y_{\mathrm{B}}-y_{\mathrm{A}}
\end{array}\right.
$$

Finally, the extracted region was resized to 150 pixels $\times$ 50 pixels. Figure 9 shows an example of the extraction.

Histogram equalization is used to flatten the image histogram. In this process, the image is modelled as a probability density function, and the processing attempts to ensure that the probability that a pixel will take on a particular intensity is the same for all values. This is
especially used in images that have poor contrast. Images that look like they are too dark, washed out, or bright are good candidates for applying histogram equalization. On plotting the histogram, the spread of pixels is limited to a very narrow range. Performing histogram equalization flattens the histogram and provides a better contrast image. Implementing this stretches the dynamic range of the histogram.

In this study, illumination is a noise that should not be learned by the model. Therefore, histogram equalization was applied to each image to eliminate the illumination difference. Figure 10 shows two examples of histogram equalization. The first is related to an image captured in a dark condition, while the latter shows the one captured in a light condition.
2.3. Enlarging Data Amount. As mentioned before, the amount of raw data were limited, and therefore, techniques to supplement this were applied. Figure 11 shows the procedure used for enlarging data, which mainly focused on the image sequences generated from videos.

Each of the 31 videos was first converted into a sequence of images arranged in time series, which included visual information about the entire mixing process, as introduced in Section 2.1. It was observed that the sequence contained several mixing cycles as the blades kept rotating. Owing to engineering practice and existing research [30], the SCC mixture could be regarded as being mixed fully only during the last 60 s of the 240 s mixing process. Therefore, only the image sequence corresponding to the last 60 s was used for training, which was a conservative approach.

To expand the amount of data, two parameters were utilized; these are shown in Figure 12. The first parameter was the segmentation length ( $L$ ), which was used to divide


Figure 8: Affine transformation to the images.


Figure 9: Extracting the ROI.


Figure 10: Continued.


Figure 10: Histogram equalization: (a) dark condition; (b) light condition.


Figure 11: Flowchart for enlarging the amount of training data.


Figure 12: Definition of segmentation length $(L)$ and temporal resolution $(R)$.
the image sequences into subsequences. The second parameter was the downsampling stride, which was defined as the temporal resolution $(R) . R$ represents the precision of a measurement with respect to time.

As mentioned before, the used mixer had a fixed rotating speed of 51 rpm and the video had a fixed frame rate of 30 fps . It was easily determined that each mixing cycle consisted of 35 images using equation (3). Therefore, $L$ was
set as 35 , which ensured that every subsequence exactly contained images of a whole mixing cycle:

$$
\begin{equation*}
N_{\text {frames }}=\frac{\text { frame rate }}{\text { rotating speed }}=\frac{30 \times 60}{50} \approx 35 . \tag{3}
\end{equation*}
$$

$R$ was used to downsample the subsequences to reduce the computation complexity. The case for $R=5$ is first analysed; later, $R=3,7$, and 9 are discussed.

After enlarging the data, the dataset was rearranged into several sequences using SF and VF values as corresponding labels; this is shown in Figure 13. Each sequence consisted of seven images of the mixing process. An example sequence, starting with an image with blade phase angle of 0 , is shown in Figure 14. The phase angle 0 was defined as the phase angle when blade passed by the lower-left corner of the inner wall of the mixer hatch. However, the starting image could be of any angle, which is more practical for training a flexible model.
2.4. Data Augmentation. After data preprocessing and expansion, the raw data were transformed into several image sequences contained 7 processing images of fixed size 150 pixels $\times 50$ pixels. As a result, the amount of data grew remarkably, minimizing the overfitting effect. To further reduce the overfitting problem, the conduction of data augmentation could be taken, which improved the performance of training without adding new information into the model. The operation of data augmentation were typically chosen to be label-preserving, such that they can be trivially used to extend the training set and encourage the system to become invariant to these transformations


Figure 13: The enlarged dataset.


Figure 14: Rotating blade phase angle and the corresponding images.
[31]. In computer vision, state-of-the-art deep learning systems relied on label-preserving image transformations such as scaling, rotation, flipping, and mirroring [32]. However, the augmentation operation did not focus on a single image but the whole sequence in this study. The first image was turned to the end of the sequence in turn, so that the sequence changed without disrupting the temporal order. The operation was based on the fact that the sequences comprised a whole cycle, and it did not matter which image was used for starting the sequence. Figure 15 illustrates the operation, yielding a dataset of seven times the size of the original dataset.

## 3. Model

3.1. Architecture. As illustrated in Figure 16, the proposed neural network comprised convolutional and recurrent parts. The convolutional part extracted spatial features by alternating and stacking convolutional layers and pooling operations. The convolutional layers convolved with raw input data using multiple local kernel filters and generated invariant local features. In the proposed network, the convolutional layers both had four filters with $2 \times 2$ kernel size. The stride length of the convolution was set as 1 , and there was no padding around the input or feature map. The activation function


The dataset was 7 times the size of original dataset

| 2 | 3 | 4 | 5 | 6 | 7 | 1 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 3 | 4 | 5 | 6 | 7 | 1 | 2 |
| 4 | 5 | 6 | 7 | 1 | 2 | 3 |
| 5 | 6 | 7 | 1 | 2 | 3 | 4 |
| 6 | 7 | 1 | 2 | 3 | 4 | 5 |
| 7 | 1 | 2 | 3 | 4 | 5 | 6 |

Figure 15: Data augmentation operation.


Figure 16: Architecture of the proposed model.
was set as rectified linear units (ReLU). The subsequent pooling layers extracted the most significant features with a fixed-length sliding windows over the data by the max pooling operation. The pool size was set to $2 \times 2$. One flatten layer was used to transform the high-dimension feature map to a one-dimensional vector after convolutional and pooling operations. The recurrent part was a LSTM used to learn the sequential feature maps. The input was the flattened vectors which were generated by the convolutional part. The dimension of the LSTM output space was set to five. A "time-distributed" layer was used to connect the convolutional and recurrent parts because the recurrent
layer had an additional time dimension, and therefore, these two parts could not be connected directly. With the time-distributed layer, convolutional parts would then be applied to each of the time steps. After multilayer feature learning, a fully-connected layer converted the features to a two-dimensional vector for this regression task.

The network architecture and the selected parameters of layers were shown in Table 2. The number of parameters (Param \#) of all the layers was listed in the last column in Table 2. The first convolutional layers had 20 trainable parameters, and the second had 68 trainable parameters. The LSTM part had 31,800 trainable parameters. Therefore, there

Table 2: Trainable parameters of the model.

| Layer | Kernel | Output shape | Function | Param \# |
| :--- | :---: | :---: | :---: | :---: |
| Data | - | $7 \times 50 \times 150$ | - | 0 |
| Time distribute (conv 1) | $4 \times 2 \times 2$ | $7 \times 49 \times 149 \times 4$ | 20 |  |
| Time distribute (pool 1) | $2 \times 2$ | $7 \times 23 \times 73 \times 4$ | 0 |  |
| Time distribute (conv 2) | $4 \times 2 \times 2$ | $7 \times 23 \times 73 \times 4$ | - | 68 |
| Time distribute (pool 2) | $2 \times 2$ | $7 \times 11 \times 36 \times 4$ | ReLU | 0 |
| Time distribute (flatten 3) | - | $7 \times 1584$ | ReLU | 0 |
| LSTM 4 | - | 5 | - | 31800 |
| fc 5 | - | 2 | - | 12 |

were 31,900 trainable parameters altogether in the model for further training.

The model was implemented in Python using the Keras package. The CPU used was Intel ${ }^{\circledR}$ Core (TM) i7-4790, RAM was 12.0 GB , and GPU was NVIDIA GeForce GT 730.
3.2. Training. A number of empirical hyperparameters were applied in the training. Cross entropy was employed as the cost function to measure the similarity between the predicted values and target values [33, 34]. Moreover, dropout was used to prevent overfitting of the model. The 31,900 trainable parameters of the model were used in learning for 10 epochs. The sequence samples from 26 out of 31 videos were used as the training set and validation set, while 5 randomly chosen videos were used for testing the performance of the model, as shown in Figure 17. The number of videos that were chosen was $1,2,4,25$, and 27 , and the sequences generated from these videos were completely excluded from use in training. This was enforced to ensure that the testing results were not biased and to definitively determine if the model was effective.
3.3. Learning and Validation Results. The learning and evaluation results are shown in Figure 18. The training accuracy reached 0.9967 at 10 epochs, while the validation accuracy reached 1.0000. It can be further observed from the plot that the accuracy increases in the training and validation sets were in steps. This demonstrated that the training did not encounter the overfitting problem owing to training practice.

## 4. Results and Discussion

4.1. Testing Results. After the training phase, the model was applied to the test sequences for further validation. The predictions of the sequences were computed together to obtain average SF and VF values corresponding to a specific video. Table 3 shows the predicted SF values and relative error (RE) along with the ground truth (GT). From this, it can be concluded that the trained model estimated the SF value effectively.

Table 4 shows the predicted VF values, RE, and GT. RE of predicted VF values were relatively larger compared with that of SF values. The reason was that the GT of VF values were numerically smaller. However, the prediction was also acceptable to some extent. The threshold of VF was 25 s . A SCC mixture with the VF value smaller than

25 s would be seemed as qualified. So videos 4,25 , and 27 would be distinguished as qualified based on the DL model prediction, which was in accordance with the fact. Video 2 would be distinguished as blocked in the V-funnel test because the estimation of the VF value was close to 200 s . 200 s was a default value to represent the blocked condition, so the assessment of video 2 was also in accordance with the fact. However, the assessment of video 1 was mistaken. To sum up, the accuracy of VF assessment was $80 \%$.
4.2. Discussion about Time Resolution. The time resolution $(R)$ was first set to 5 in this study; cases of $R=3,7$, and 9 are discussed in this section. As per the definition of time resolution, the effect of this parameter is reflected in the degree of downsampling. In theory, the larger the $R$, the greater the loss of temporal information.

The $R$ value is also related to the length of the image sequences in the dataset, as shown in Table 5. The number of parameters will not change with $R$ because it is determined by the model architecture. However, with the increase in the $R$ value, the amount of input data fed to the model would decrease. Thus, the larger the $R$, the lesser the computation complexity that must be overcome by the proposed model. Datasets corresponding to different $R$ were generated for further training. After training, the training time consumptions were recorded as an index to assess the performance of different $R$ values, as shown in Figure 19.

The learning results obtained for different $R$ are shown in Figure 20. The training accuracy was more than 0.9940 at 10 epochs for all cases. It could be observed from the plot that the training converged most quickly when $R=3$ and 5 . However, the speed of convergence did not seem to be of importance when compared to other conditions.

Finally, the estimation accuracies of the same testing set were computed, as shown in Figures 21 and 22 and Tables 6 and 7. The coefficient of determination $\left(R^{2}\right)$ were also marked in Figures 21 and 22. It was observed that the estimation results of $R=5$ and 9 were relatively more accurate. By considering all parameters such as computation complexity, converging speed, and accuracy, the choice of $R=5$ was shown to be the most appropriate.

Thus, a framework can be constructed to determine the parameters of downsampling to serve as a strategy for enlarging original data. Figure 23 summarizes this stepwise process. First, the segmentation length is determined based


Figure 17: Samples of 5 randomly chosen videos were used for testing, while others served as the training and validation sets.


Figure 18: Training and validation results.

Table 3: Prediction results of SF values.

| Video number | GT SF (mm) | Prediction (mm) | RE (\%) |
| :--- | :---: | :---: | :---: |
| 1 | 575 | 587 | 2.1 |
| 2 | 355 | 386 | 8.7 |
| 4 | 480 | 487 | 1.7 |
| 25 | 600 | 677 | 12.9 |
| 27 | 648 | 676 | 4.4 |

Table 4: Prediction results of VF values.

| Video number | GT VF (s) | Prediction (s) | RE (\%) |
| :--- | :---: | :---: | :---: |
| 1 | 35.8 | 18.2 | 49.6 |
| 2 | 200.0 | 176.6 | 11.7 |
| 4 | 14.5 | 21.5 | 43.5 |
| 25 | 13.0 | 8.3 | 35.9 |
| 27 | 13.0 | 14.1 | 8.1 |

Table 5: Time resolutions ( $R$ ) and lengths of corresponding downsampling sequences.

| R | Lengths of sequences |
| :--- | :---: |
| 3 | 11 |
| 5 | 7 |
| 7 | 5 |
| 9 | 3 |

on the mixer cycle length. Next, an option list of time resolutions is created. By comparing the training time consumption, converging speed, and accuracy, a choice of time resolution is made, with a series of the training process


Figure 19: Training time consumptions of different $R$.
on the preliminary dataset. Preliminary dataset is a representation of raw data, of which there is a limited amount available. Finally, the parameter pair of segmentation length and time resolution is given, which will be further used to train the whole data.

## 5. Conclusions

In this paper, a method was proposed for estimating SCC workability during the mixing process. A combined model based on CNN and LSTM was utilized to predict the SF and VF values of SCC. The SCC mixing videos were converted into a dataset of image sequences to fit the training needs of the proposed model. The trained DL model achieved good performance and could be taken into consideration for use in automated mixing plants. A framework to determine the data preparation strategy was introduced as well. The strategy mainly focused on the determination of the time resolution of the raw data. The proposed method provides an effective basis that will help develop a smart batching plant in the future.

The data collection approach used was easy to implement as there were no strict requirements that the tripod placement and shooting angle of the smart phone be maintained constant. This enabled collection of data from different experiment batches as long as the mixer volume was the same. The significance of the ease of setting up the experiment would be evident when collecting a high volume of data. Furthermore, the data feed in the proposed model comprised image sequences arranged in time series, which ensured that the potential data in the temporal information were used.


Figure 20: Training results of different $R$.

O Data

- Fit

○ Data

$$
\ldots \quad \text { Fit }
$$


O Data

- Fit
(b)

(d)

Figure 21: Prediction results of SF values with different $R$ : (a) $R=3$; (b) $R=5$; (c) $R=7$; (d) $R=9$.


Figure 22: Prediction results of VF values with different $R$ : (a) $R=3$; (b) $R=5$; (c) $R=7$; (d) $R=9$.

Table 6: Prediction results of SF values with different $R$.

| Video number | GT (mm) | $R=3$ |  | $R=5$ |  | $R=7$ |  | $R=9$ |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Prediction (mm) | RE (\%) | Prediction (mm) | RE (\%) | Prediction (mm) | RE (\%) | Prediction (mm) | RE (\%) |
| 1 | 575 | 509 | 11.5 | 643 | 11.8 | 531 | 7.7 | 515 | 10.4 |
| 2 | 355 | 340 | 3.7 | 314 | 11.3 | 305 | 14.0 | 364 | 2.5 |
| 4 | 480 | 277 | 42.3 | 504 | 5.0 | 535 | 11.3 | 534 | 11.3 |
| 25 | 600 | 599 | 0.1 | 573 | 4.5 | 585 | 2.5 | 596 | 0.6 |
| 27 | 648 | 637 | 1.7 | 589 | 10.0 | 593 | 8.5 | 589 | 10.0 |

In future work, we propose taking into account more practical condition for training. The training and testing images used in this study included preprocessed image
sequences; however, the architecture of the model may also be suitable for other types of data, such as videos captured from a 30 L single-shaft mixer or a twin-shaft mixer. In

Table 7: Prediction results of VF values with different $R$.

| Video number | GT (s) | $R=3$ |  | $R=5$ |  |  |  |  |  |  |  | $R=7$ |  | $R=9$ |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Prediction (s) | RE (\%) | Prediction (s) | RE (\%) | Prediction (s) | RE (\%) | Prediction (s) | RE (\%) |  |  |  |  |  |
| 1 | 35.8 | 25.0 | 30.5 | 18.2 | 49.6 | 20.7 | 42.5 | 36.4 | 1.2 |  |  |  |  |  |
| 2 | 200.0 | 177.5 | 11.2 | 176.6 | 11.7 | 164.3 | 17.9 | 169.7 | 15.2 |  |  |  |  |  |
| 4 | 14.5 | 12.7 | 15.6 | 21.5 | 43.5 | 25.3 | 68.9 | 16.0 | 6.6 |  |  |  |  |  |
| 25 | 13.0 | 0 | 100.0 | 8.3 | 35.9 | 16.3 | 25.2 | 5.36 | 58.9 |  |  |  |  |  |
| 27 | 13.0 | 30.6 | 135.0 | 14.1 | 8.1 | 15.6 | 20.3 | 13.3 | 2.4 |  |  |  |  |  |



Figure 23: Downsampling parameters determination framework.
future research, we plan on collecting a series of video data to explore the possibility of making the proposed model more flexible.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China, the National High Technology Research and Development Program 863, and the National Key Laboratory (nos. 51239006, 2012AA06A112, and 2015-KY01).

## References

[1] H. Okamura and M. Ouchi, "Self-compacting concrete," Journal of Advanced Concrete Technology, vol. 1, no. 1, pp. 5-15, 2003.
[2] X. An, Q. Wu, F. Jin et al., "Rock-filled concrete, the new norm of SCC in hydraulic engineering in China," Cement and Concrete Composites, vol. 54, pp. 89-99, 2014.
[3] L. Coppola, T. Cerulli, and D. Salvioni, "Sustainable development and durability of self-compacting concretes," in Proceedings of 11th International Conference on Fracture 2005 (ICF11), pp. 2226-2241, Turin, Italy, March 2005.
[4] Q. Wu and X. An, "Development of a mix design method for SCC based on the rheological characteristics of paste," Construction and Building Materials, vol. 53, pp. 642-651, 2014.
[5] C. F. Ferraris, L. Browner, C. Ozyildirim, and J. Daczko, "Workability of self-compacting concrete," in Proceedings of International Symposium on High Performance Concrete, pp. 398-407, Orlando, FL, USA, 2000.
[6] D. Chopin, B. Cazacliu, F. De Larrard, and R. Schell, "Monitoring of concrete homogenisation with the power consumption curve," Materials and Structures, vol. 40, pp. 897-907, 2007.
[7] B. Daumann and H. Nirschl, "Assessment of the mixing efficiency of solid mixtures by means of image analysis," Powder Technology, vol. 182, no. 3, pp. 415-423, 2008.
[8] S. Li and X. An, "Method for estimating workability of selfcompacting concrete using mixing process images," Computers and Concrete, vol. 13, no. 6, pp. 781-798, 2014.
[9] Z. Ding and X. An, "A method for real-time moisture estimation based on self-compacting concrete workability detected during the mixing process," Construction and Building Materials, vol. 139, pp. 123-131, 2017.
[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798-1828, 2013.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in Lecture Notes in Computer Science, pp. 346-361, Springer Verlag, Berlin, Germany, 2014.
[12] D. Jia, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248-255, Miami, FL, USA, June 2009.
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2012.
[14] C. Affonso, A. L. D. Rossi, F. H. A. Vieira, and A. C. P. de L. F. de Carvalho, "Deep learning for biological image classification," Expert Systems with Applications, vol. 85, pp. 114-122, 2017.
[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, Columbus, OH, USA, June 2014.
[16] S. Gidaris and N. Komodakis, "Object detection via a multiregion and semantic segmentation-aware U model," in Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1134-1142, Santiago, Chile, December 2015.
[17] J. Donahue, L. A. Hendricks, M. Rohrbach et al., "Long-term recurrent convolutional networks for visual recognition and
description," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 677-691, 2017.
[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 17351780, 1997.
[19] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, 2015, http://arxiv.org/ abs/1506.04214.
[20] A. Zhang, K. C. P. Wang, B. Li et al., "Automated pixellevel pavement crack detection on 3D asphalt surfaces using a deep-learning network," Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 10, pp. 805-819, 2017.
[21] S. Yokoyama and T. Matsumoto, "Development of an automatic detector of cracks in concrete using machine learning," Procedia Engineering, vol. 171, pp. 1250-1255, 2017.
[22] U. Reuter, A. Sultan, and D. S. Reischl, "A comparative study of machine learning approaches for modeling concrete failure surfaces," Advances in Engineering Software, vol. 116, pp. 67-79, 2018.
[23] J. S. Chou, C. F. Tsai, A. D. Pham, and Y. H. Lu, "Machine learning in concrete strength simulations: multi-nation data analytics," Construction and Building Materials, vol. 73, pp. 771-780, 2014.
[24] A. M. Abd and S. M. Abd, "Modelling the strength of lightweight foamed concrete using support vector machine (SVM)," Case Studies in Construction Materials, vol. 6, pp. 8-15, 2017.
[25] Z. M. Yaseen, R. C. Deo, A. Hilal et al., "Predicting compressive strength of lightweight foamed concrete using extreme learning machine model," Advances in Engineering Software, vol. 115, pp. 112-125, 2017.
[26] W. Z. Taffese and E. Sistonen, "Machine learning for durability and service-life assessment of reinforced concrete structures: recent advances and future directions," Automation in Construction, vol. 77, pp. 1-14, 2017.
[27] M. A. Hariri-ardebili and F. Pourkamali-anaraki, "Support vector machine based reliability analysis of concrete dams," Soil Dynamics and Earthquake Engineering, vol. 104, pp. 276-295, 2018.
[28] W. Z. Taffese, E. Sistonen, and J. Puttonen, "CaPrM: carbonation prediction model for reinforced concrete using machine learning methods," Construction and Building Materials, vol. 100, pp. 70-82, 2015.
[29] C. Kanan and G. W. Cottrell, "Color-to-grayscale: does the method matter in image recognition?," PLoS One, vol. 7, no. 1, Article ID e29740, 2012.
[30] B. Cazacliu and N. Roquet, "Concrete mixing kinetics by means of power measurement," Cement and Concrete Research, vol. 39, no. 3, pp. 182-194, 2009.
[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradientbased learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2323, 1998.
[32] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in Proceedings of 16th International Society for Music Information Retrieval Conference, pp. 121-126, Malaga, Spain, 2015, http:// www.ofai.at/~jan.schlueter/pubs/2015_ismir.pdf.
[33] P. Golik, P. Doetsch, and H. Ney, "Cross-entropy vs. Squared error training: a theoretical and experimental comparison," in Proceedings of the 22nd international conference on Machine learning (ICML'05), Bonn, Germany, 2013.
[34] S. Wiesler, J. Li, and J. Xue, "Investigations on hessian-free optimization for cross-entropy training of deep neural networks," in Proceedings of 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 2013.


The Scientific World Journal


