Neural Computing and Applications (2023) 35:24315–24332
https://doi.org/10.1007/s00521-023-09004-y

ORIGINAL ARTICLE

Detection of solidification crack formation in laser beam welding videos of sheet metal using neural networks

Wenjie Huo[1] · Nasim Bakir[2] · Andrey Gumenyuk[2] · Michael Rethmeier[2] · Katinka Wolter[1]

Received: 2 May 2023 / Accepted: 22 August 2023 / Published online: 28 September 2023

Abstract
Laser beam welding has become widely applied in many industrial fields in recent years. Solidification cracks remain one of the most common welding faults that can prevent a safe welded joint. In civil engineering, convolutional neural networks (CNNs) have been successfully used to detect cracks in roads and buildings by analysing images of the constructed objects. These cracks are found in static objects, whereas the generation of a welding crack is a dynamic process. Detecting the formation of cracks as early as possible is greatly important to ensure high welding quality. In this study, two end-to-end models based on long short-term memory and three-dimensional convolutional networks (3D-CNN) are proposed for automatic crack formation detection. To achieve maximum accuracy with minimal computational complexity, we progressively modify the model to find the optimal structure. The controlled tensile weldability test is conducted to generate long videos used for training and testing. The performance of the proposed models is compared with the classical neural network ResNet-18, which has been proven to be a good transfer learning model for crack detection. The results show that our models can detect the start time of crack formation earlier, while ResNet-18 only detects cracks during the propagation stage.

Keywords Deep neural networks · Laser beam welding · Solidification cracks detection · Spatio-temporal features extraction · Welding quality control system

1 Introduction

Laser beam welding is one of the most modern processes in the manufacturing industry for joining metal materials. Solidification cracking belongs to the serious faults in this process. In order to avoid unnecessary costs and ensure high quality, the long-term objective is to establish an intelligent computer-based control system capable of automatically detecting the formation of cracks to replace manual detection which is time-consuming and laborious. Furthermore, real-time crack detection will serve to adjust the welding process in real time through automatic control of the welding parameters. The formation of solidification cracks during welding mainly depends on metallurgical and thermo-mechanical factors. According to established theories, from a thermo-mechanical point of view, the strain and strain rate arising during welding near the solidification front are responsible for the solidification cracks. A solidification crack occurs in the so-called mushy zone, i.e. in the zone immediately behind the weld pool

✉ Wenjie Huo
wenjie.huo@fu-berlin.de

✉ Katinka Wolter
katinka.wolter@fu-berlin.de

Nasim Bakir
nasim.bakir@bam.de

Andrey Gumenyuk
andrey.gumenyuk@bam.de

Michael Rethmeier
michael.rethmeier@bam.de

[1] Mathematics and Computer Science, Free University Berlin, 14195 Berlin, Germany

[2] BAM - Bundesanstalt für Materialforschung und -prüfung, Fachbereich Schweißtechnische Fertigungsverfahren, 12205 Berlin, Germany

where solidification is still incomplete. If the strains exceed the ductility of the material within the mushy zone, solidification cracks will appear. The formation of cracks can be determined by measuring the strain state in the vicinity of the weld pool. Optical flow estimation can be used to calculate displacement and strain fields [1, 2]. Its main disadvantage is the high computational cost, which makes real-time monitoring difficult in practical applications.

In recent years, machine learning methods based on neural networks have been widely researched and applied in industrial fields. Among them, the convolutional neural network (CNN) [3] was first proposed for the recognition of handwritten digits, and then achieved remarkable success for large-scale classification in the ImageNet data set [4]. Nowadays, the CNN is widely employed in the computer vision field due to its good performance, e.g. in image classification, object detection, and action recognition. Crack detection can be regarded as a binary classification problem in image classification. By applying some classic networks, cracks or defects in static images can be successfully detected. However, these networks cannot capture motion information. In the welding process, cracks can occur at any time and the duration of the process is unknown. Locating the initiation and end time of crack formation in untrimmed videos is more complex. If frames are analysed individually, the formation moment of tiny cracks will be hard to identify, which leads to a delay in the alarm. In this situation it is beneficial to consider temporal information.

Action recognition is an extension of image classification. It has received much attention in recent years and has been widely applied in video analysis tasks, such as monitoring abnormal events in surveillance cameras. The commonly applied models can be divided into three types: (1) The two-stream model [5], which contains two CNN networks, one takes the optical flow as input to extract motion information and the other one takes RGB images as input. Each stream is followed by a softmax layer, and they can be fused by averaging or using a SVM. (2) Long short-term memory (LSTM) [6] has achieved outstanding progress in processing a data sequence, and LRCN [7] was proposed to utilize a CNN model to extract spatial features and then impose a LSTM on the result of the CNN to extract temporal features. (3) A three-dimensional convolutional network (3D-CNN) [8] is the third popular method which replaces the 2D convolutional kernel with a 3D kernel that contains an additional time dimension. The optimal configuration of a 3D-CNN was explored by systematic research and named *C3D* [9]. Its successful development makes it possible to learn crack generation features from welding videos. To the best of our knowledge, there is no work on capturing the dynamic crack generation process. However, LSTM [6] and 3D-CNN [8] models have shown promising results in capturing motion features in videos. In this study, we propose two networks based on CNN-LSTM and 3D-CNN for automatic crack detection. A two-stream model has not been chosen because of its low efficiency caused by the optical flow calculation. The contributions of this work are as follows.

- A controlled tensile weldability test (CTW test) is set up to generate digital images containing solidification cracks. With this process 14 high-quality welding videos totaling over 50,000 frames are collected.
- Two different models based on CNN-LSTM and 3D-CNN are developed and compared. To the best of our knowledge, this is the first work to detect temporal boundaries of crack formation during the laser beam welding process.
- The proposed models are evaluated with several previously unseen test sets, and compared with the ResNet-18 model [10] in accuracy and calculation cost. Extensive experiments demonstrate that the boundaries of cracks located by our models are more accurate than those found using ResNet-18.

2 Related work

The rapid development of deep learning makes it a tempting candidate to replace human visual inspections in fault and defect detection. Two fault diagnosis methods are proposed in [11] and [12] to address the problem of scarce faulty samples and a deficit in labelled data. In static crack detection, the trained deep learning model can effectively detect cracks in different structures. A well-known CNN model has successfully found concrete cracks in civil infrastructures. Combined with a sliding window technique, it can detect cracks in images of any resolution, outperforming the traditional Canny and Sobel methods [13]. A fusion framework NB-CNN [14] has been proposed to detect cracks on metallic surfaces in nuclear power plants, while applying Naive Bayes decision to reduce the false positive rate. A shallow CNN architecture optimized on LeNet-5 [3] was proposed in surface concrete crack detection [15]. CrackViT [16] combined the advantages of CNN and transformer networks. It explored different fusion methods for the two models, implementing pixel-level crack extraction. In addition to crack detection, [17] developed and compared four CNNs with different receptive fields and performed the classification of different types of pavement cracks.

The use of transfer learning can solve the challenge of insufficient annotated data. For example, in [18], based on the VGG-16 [19] architecture pre-trained on the ImageNet data set, the influence of the model parameters were investigated using a small data set. In [20], the AlexNet [4] architecture in fully trained transfer learning and classifier modes was trained and compared with six common edge detection schemes. [21] detected two common defects (crack and corrosion) with two pre-trained networks, i.e. VGG-16 and ResNet-18. Moreover, in [22] the performance of 15 state-of-the-art convolutional neural networks which detected cracks was compared in terms of number of parameters, area under the curve (AUC), and inference time.

In the welding area, the application of machine learning in quality prediction and classification has also seen a rapid increase. By learning from digital images collected by high-speed cameras, many CNN-based models have been established to predict different laser welding defects, including porosity, level misalignment [23]; blowout, humping, and undercut [24]; and slag inclusions, cracks, floating holes, and lack of fusing (false friends) [25]. The literature classified welding defects into five categories [26] (conduction welding, stable keyhole, unstable keyhole, blowout, and pores) through X-ray images. The work in [27] extracted features from infrared image sequences and designed an ensemble deep neural network based on CNN and gated recurrent units (GRU), enabling detection of four critical welding defects (sagging, lack of penetration, lack of fusion, and geometric deviations of the weld seam). The authors of [28] designed a high-quality monitoring method for micro-plasma arc welding (MPAW), which extracts the contour of the molten pool region based on the method of Otsu and then applies a support vector machine (SVM) to identify the lack of fusion, humping, and sound weld states.

3 Methodology

A high-level description of our system architecture is shown in Fig. 1. The data generation and pre-processing will be introduced in the next section. This section is about the learning methods which we have applied. Two different machine learning models for analysing the videos of the welding process are introduced in Sect. 3.1. The architecture and hyperparameter details are provided in Sect. 3.2, and the complexity analysis is given in Sect. 3.3.

3.1 Learning temporal features

Before exploring the model architectures, we concentrate on temporal features. Including temporal aspects in the models is the main novelty in this paper.

3.1.1 CNN-LSTM model

Since the formation and propagation of cracks are changing with time, a CNN-based model [29] aims to learn the features of previous information by stacking multiple images together as the input. The drawback of this approach is that the temporal information will be collapsed after each convolution operation. In order to focus on both, the spatial and temporal information, it has been proposed to combine the LSTM with a CNN [7] for the latter's good performance in learning from the sequential data. The architecture of the combined CNN-LSTM is shown in the middle of Fig. 1, in the grey box. It is a simple sequential integration of CNN and LSTM. The input consists of several consecutive frames. First, each frame is sent to the CNN individually to extract visual features, then the feature maps obtained by the CNN are flattened into a one-dimensional vector and fed into the following LSTM in sequence.

Figure 2 shows the internal implementation of the LSTM. The inputs are X_t, the hidden state of the previous timestep is h_{t-1}, and the memory cell state is C_{t-1}. The outputs are h_t and C_t. Both h_t and C_t retain the previous information and affect the present output. In addition, the LSTM incorporates three gates. The forget gate f_t determines how many previous hidden states should be forgotten, the input gate i_t determines how much current input should be updated, and the output gate o_t determines how much of C_t should be transferred to h_t. These modules enable the LSTM to integrate previous and current data. The formulas of the LSTM are given in Eq. (1), where 'x' denotes the input matrix at time t, 'W_i' are weight matrices and 'b' is the offset value, '$\odot$' denotes the matrix multiplication operation, 'σ' denotes the sigmoid function ensuring that the output lies between 0 and 1. It is applied to each gate.

$$\begin{aligned}
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)\\
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)\\
\hat{c}_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \hat{c}_{t-1}\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)\\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{1}$$

3.1.2 3D-CNN model

Our second considered architecture is the 3D-CNN. The 3D-CNN is another network that is widely used in video analysis. Unlike CNN-LSTM, it can extract both appearance and motion features simultaneously. Compared with

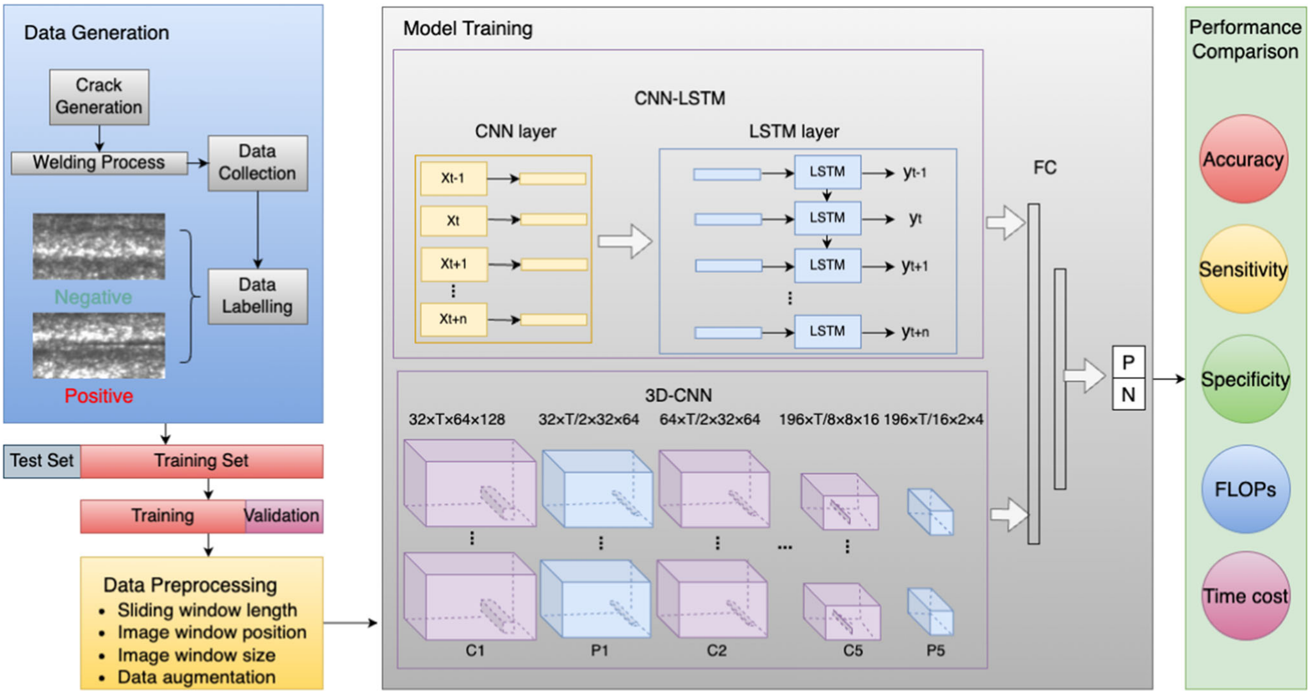


Fig. 1 Overall architecture including the two different models for crack detection

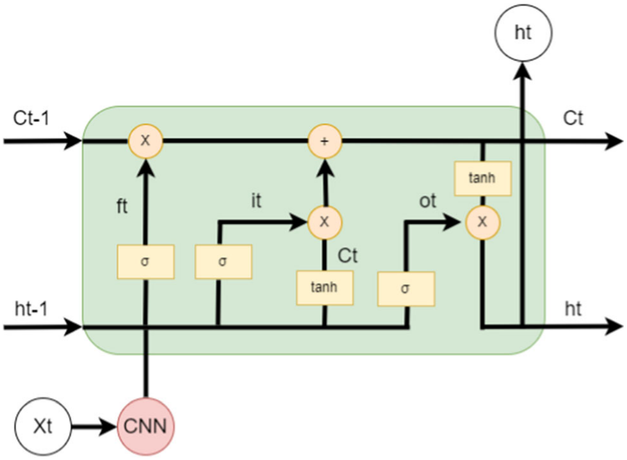


Fig. 2 Internal structure of CNN-LSTM

2D convolution, the 3D-CNN uses an additional dimension in depth. Figure 3 illustrates the difference between multichannel convolution and 3D convolution applied to images. The upper structure is a multichannel convolution, and its input is a multichannel feature map. The feature map of each channel is convolved with a kernel, and the results are added; thus, the output is a single feature map. Multiple such images have been stacked as the input to the CNN model in [29], and each image is treated as a channel. As can be seen from the figure, after the first convolution layer, the temporal information is lost. The structure below is a 3D convolution, and L represents the depth instead of the number of channels. The kernel size of the 3D convolution is $k \times k \times d$, generally $d < L$. Its output feature map is still three-dimensional due to 3D convolution and 3D pooling operations. The kernel size of the 2D convolution is $k \times k$, and the number of channels is C. The total number of parameters is therefore $k \times k \times C$. The 3D convolution has $k \times k \times d \times L$ parameters, many more than the 2D network.

The structure of the 3D-CNN is also shown in Fig. 1. It contains five convolution layers, and each layer is followed by a pooling layer. According to the findings for the C3D model [9], the best kernel size is $3 \times 3 \times 3$, so all filters are of dimension $3 \times 3 \times 3$ with stride $1 \times 1 \times 1$ in the convolution layer. The kernel size in the pooling layer is $2 \times 2 \times 2$ with stride $2 \times 2 \times 2$, except for the first layer which is equipped with $1 \times 2 \times 2$ parameters in order to not merge the temporal information at an early stage.

3.2 Model designs

The feature learning module consists of three main layers: a convolutional, a pooling, and a fully connected layer. The network can be built in various ways by setting different hyperparameters and choosing different sequences of layers. The performance of the model can improve as the model becomes more complex, and the mainstream models (LeNet, AlexNet, VGG, GoogLeNet [30], and ResNet) are designed increasingly deep. However, the time-consuming models are inappropriate as the on-line crack detection system usually requires immediate response. In this

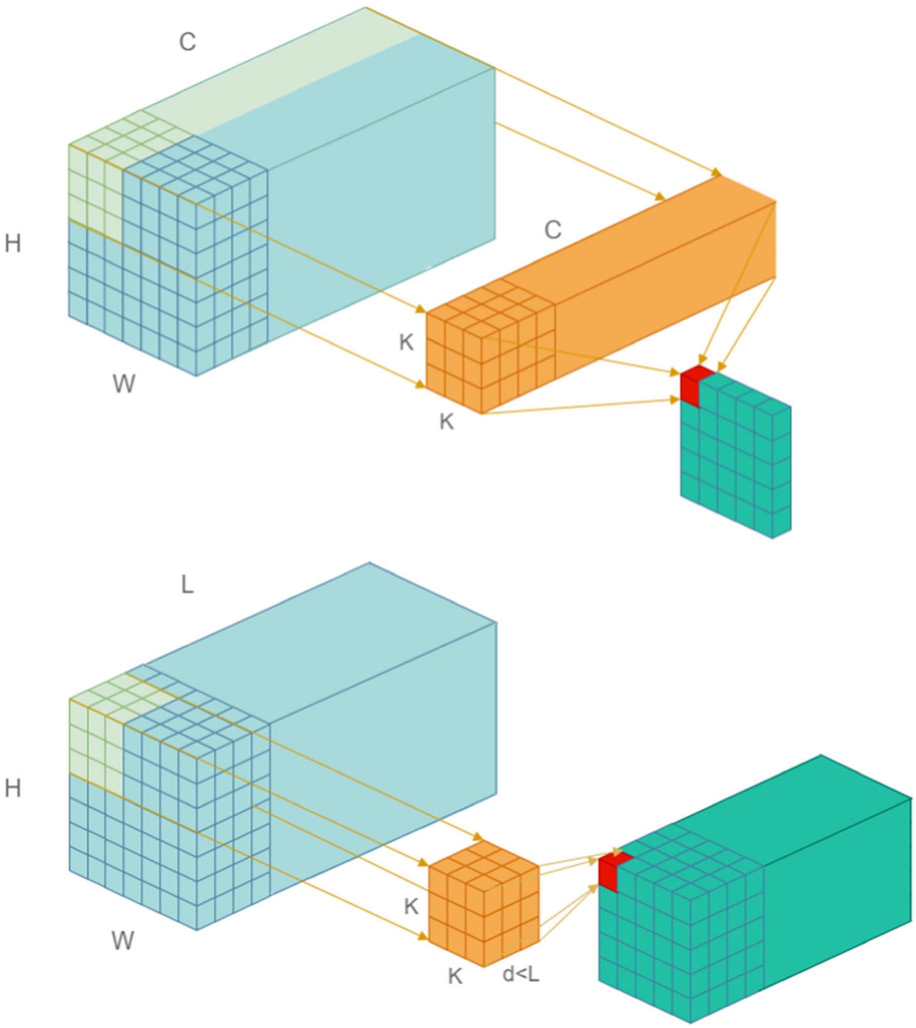


Fig. 3 Comparison of 2D (top) and 3D (bottom) convolution

section, we systematically build networks with different depth (number of layers) and width (numbers of filters) to investigate the improvement in accuracy.

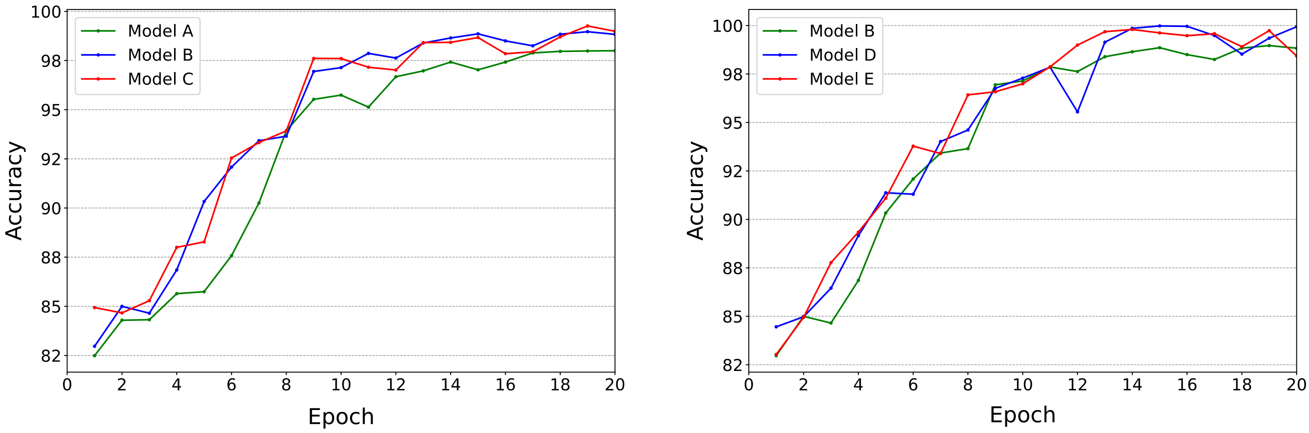
Compared with the images in the database ImageNet, welding crack data contain fewer categories and has simpler features, so we empirically design models with six to eight layers (models A, B, C in Table 1); the pooling operation is not regarded as a layer because it does not contain parameters. In general, with an increase in depth of the CNN, the performance will be better because each layer can focus on different features. Next, we fix the network depth and change the width (models D, E). According to [31], the last column of Table 1 is the theoretical time complexity (relative to model A), calculated by Eq. (2):

$$O\left(\sum_{l=1}^{L} D_k^2 \cdot N_{l-1} \cdot N_l \cdot D_w \cdot D_h\right) \tag{2}$$

where L is the depth of the CNN network. N_{l-1} and N_l are the number of input and output feature maps. D_k is the spatial size of the convolutional kernel and D_w, D_h are the spatial size of the output feature map.

Table 1 Configuration of different models

Model	Depth	Width	Strides	Complexity
A	6	(32, 64, 96, 128)	(1, 1, 1, 2)	1
B	7	(16, 32, 64, 96, 128)	(1, 1, 1, 1, 1)	1.02
C	8	(16, 16, 32, 64, 96, 128)	(1, 1, 1, 1, 1, 1)	1.04
D	7	(16, 32, 64, 128, 196)	(1, 1, 1, 1, 1)	1.75
E	7	(16, 32, 64, 128, 256)	(1, 1, 1, 1, 1)	2.12



(a) Training accuracy with different depths

(b) Training accuracy with different widths

Fig. 4 Effects of network depth and width

Figure 4 shows the training accuracy of the different models. The results illustrate that increasing the depth can improve accuracy, the accuracy of models B and C is higher than that of A. But after reaching a critical value, the accuracy becomes saturated and stagnant, model C seems not significantly more accurate than model B. The effect of width is not as significant as that of depth. Compared to B, the accuracy of models D and E is slightly improved, and their complexity is also higher. It can be seen that for our type of data, overly increasing the depth and width cannot further improve accuracy, but increases the complexity, which is unaffordable and unnecessary. To strike a good balance between cost and accuracy, based on Table 1 and Fig. 4, we ultimately chose model D as the feature extraction network.

In the LSTM layer, there are two hyperparameters that affect accuracy and computational cost, the number of the LSTM layers, and the number of features in the hidden units. The comparison of the best validation accuracy for different configurations is shown in Table 2. The accuracy reaches 95.56% with two LSTM layers and 128 hidden nodes, which cannot be significantly improved with higher configurations.

For 3D-CNN, its structure is consistent with the convolutional layers of CNN-LSTM, but it inflates the model from 2D to 3D. The configurations of CNN-LSTM and 3D-CNN are listed in Table 3. The input size is $T \times 64 \times 128$, which represents the number of input images as well as their height and width. The selection of T is discussed in Sect. 5.2. The size of the feature map will be reduced by half after the pooling layer. When the input data passes through each layer, at L5, the size is reduced to 2×4. Then, the feature maps are flattened to vectors and input into the LSTM module in the CNN-LSTM model or into two fully connected layers in the 3D-CNN model. Finally, the softmax function is applied to scale the probability distribution into the range [0, 1]. The maximum probability is predicted as the final classification. The other hyperparameters are as follows: kernel size $= 3$, number of epochs$= 20$, batch size $= 64$, learning rate $= 1e-4$, dropout $= 0.5$, the adaptive moment estimation (Adam) optimizer is used as well as the cross-entropy loss function, which are standard settings.

Table 2 Maximum validation accuracy for different LSTM configurations

Model	Layers	Hidden units	Parameters	Accuracy (%)
F	2	64	0.7M	92.36
G	2	128	1.3M	95.56
H	2	256	2.7M	95.39
I	3	128	1.5M	95.62

3.3 Complexity analysis

Finally, we will calculate the time complexity of the proposed models. The complexity of the convolutional layer has been given in the previous section. The trainable parameters of LSTM are from Equation. (1), which are $W_{xf}, W_{xi}, W_{xc}, W_{xo}$, all matrices of size $m \times n$, m and n are the dimensions of the hidden state and the input; $W_{hf}, W_{hi}, W_{hc}, W_{ho}$ which are matrices of size $m \times m$ and bias vectors b_f, b_i, b_c, b_o of size $m \times 1$. The total number of parameters in a LSTM block is $W = 4\,m \times (m + n + 1)$[32], and the computational complexity is $O(W) = O(m^2 + mn)$. Therefore, the CNN-LSTM complexity can be calculated as the sum of the complexity of convolutional layers and LSTM layers (Eq. 3).

Table 3 Configuration of the models

Layer	Operator	Channels	Kernel size	Output size	
				CNN-LSTM	3D-CNN
L1	Conv1	16	3	$16 \times 64 \times 128$	$16 \times T \times 64 \times 128$
	Pool1		2	$16 \times 32 \times 64$	$16 \times T \times 32 \times 64$
L2	Conv2	32	3	$32 \times 32 \times 64$	$32 \times T \times 32 \times 64$
	Pool2		2	$32 \times 16 \times 32$	$32 \times T/2 \times 16 \times 32$
L3	Conv3	64	3	$64 \times 16 \times 32$	$64 \times T/2 \times 16 \times 32$
	Pool3		2	$64 \times 8 \times 16$	$64 \times T/4 \times 8 \times 16$
L4	Conv4	128	3	$128 \times 8 \times 16$	$128 \times T/4 \times 8 \times 16$
	Pool4		2	$128 \times 4 \times 8$	$128 \times T/8 \times 4 \times 8$
L5	Conv5	196	3	$196 \times 4 \times 8$	$196 \times T/8 \times 4 \times 8$
	Pool5		2	$196 \times 2 \times 4$	$196 \times T/16 \times 2 \times 4$
L6	FC1	–	–	1×64	1×784
L7	FC2	–	–	1×2	1×2

The 3D-CNN complexity is modified to Eq. (4) as its output feature map size is $D_w \times D_h \times D_d$. It can be seen that the complexity of the CNN-LSTM model increases linearly with input length T, while in the 3D-CNN, when $l = 1$, $N_{l-1} = T$, T only affects the complexity of the first convolutional layer. Compared to CNN-LSTM, its computational cost changes more slowly with the increase in T. In Sec.t 5.3, we will measure and compare their time complexity through the number of floating point operations (FLOPs) according to the model implementation.

$$O\left(T \cdot \left(\sum_{l=1}^{L} D_k^2 \cdot N_{l-1} \cdot N_l \cdot D_w \cdot D_h + \left(m^2 + mn\right)\right)\right) \tag{3}$$

$$O\left(\sum_{l=1}^{L} D_k^3 \cdot N_{l-1} \cdot N_l \cdot D_w \cdot D_h \cdot D_d\right) \tag{4}$$

4 Data generation

In this section, we describe the data collected in laboratory experiments which we have performed in the welding laboratory at BAM.

4.1 Material and welding parameter

The welding experiments were carried out with the TruDisk 16002 disc laser from TRUMPF, with a maximum output power of 16 kW, a wavelength of 1030 nm and a beam parameter product of 8 mm x mrad. The welding experiments were carried out on sheets of austenitic steel grades 1.4301 (AISI 304) and H400 (EN 1.4376) with a thickness of 1.5 mm. The welding parameters applied were 2 kW laser power and a constant welding speed of 1.2 m/min at a focus position of +5 mm. Argon with a flow rate of 20 l/min was used as shielding gas.

4.2 Solidification Crack Generation

The solidification cracks were generated during laser beam welding using the externally restrained hot cracking test. This type of hot cracking tests were developed to force the cracking in the specimen by external stress or stain. In these experiments, the controlled tensile weldability test (CTW test) was employed, in which the weld specimen is subjected to predefined strain and strain rate perpendicular

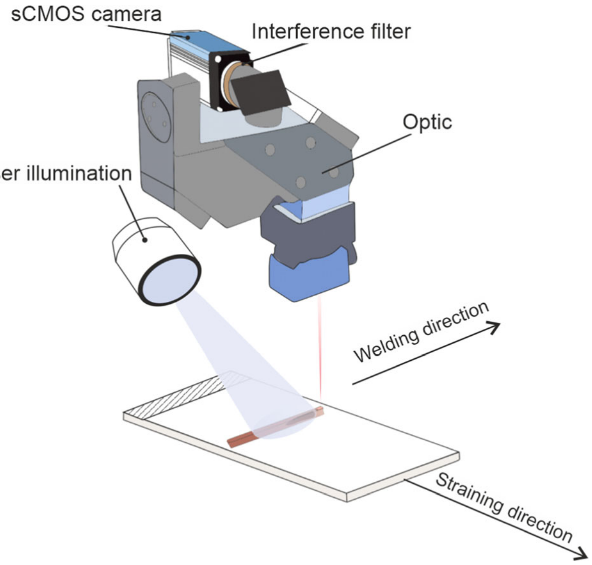


Fig. 5 Schematic representation for the experimental procedure

to the welding direction during welding, as shown schematically in Fig. 5. The externally applied strain contributes to increasing local strains in the hot crack critical region (mushy zone) and generating cracks during welding. During the CTW test, a strain of 5% and 7% was applied for steel grade 304 and 5% for steel grade H400 while welding. Those strain parameters of the CTW test cause cracks of a length between 12 mm and 18 mm in the weld. The strain variation affects only the generated crack length. Figure 6 shows the experimental setup in combination with the CTW test.

4.3 Data Acquisition

The welding process was recorded using an sCMOS camera installed co-axially to the laser path. In order to obtain a valid recording of the welding process and the re-solidified material, an external laser illumination with a wavelength of 808 nm and an interference filter of the same wavelength and with a bandwidth of 20 nm were required, as shown in Fig. 5 schematically. The filter was placed in front of the camera, allowing only the wavelengths from the illumination to pass through and suppressing all other spectral ranges, thus eliminating all optical disturbances during recording. Figure 7 shows two frames before and after the crack initiation. The camera recordings were carried out for 15 cracked welded joints. The recording rate used was 800 fps. The images were stored in tif format with a resolution of 640 × 240 pixels.

The experiments are carried out on two different steel plates applying different strains. Table 4 lists the experimental settings. The data sets with a strain of 5% and strain rate 6 s^{-1} are used as training data. To estimate the models and tune parameters during training, 20% of the frame sequences in the training data are randomly divided into validation set. To verify the performance of the models, the test set is generated by setting the strain differently to 7% or to different strain rates of 4 s^{-1} and 8 s^{-1}. Finally, 14 different sets of welding data with over 50000 frames are obtained.

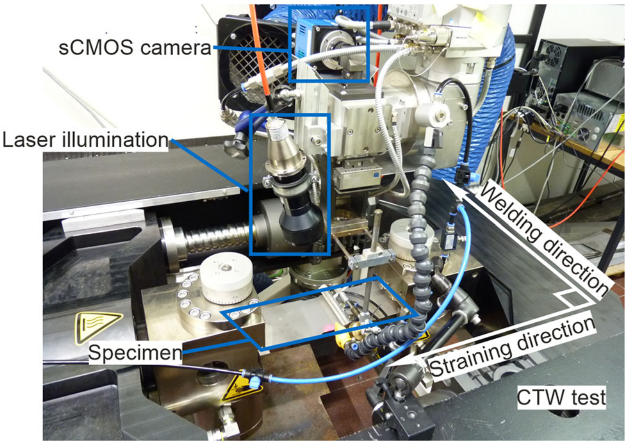


Fig. 6 Experimental setup for laser beam welding in combination with the hot cracking test (CTW test) and the applied coaxial digital camera

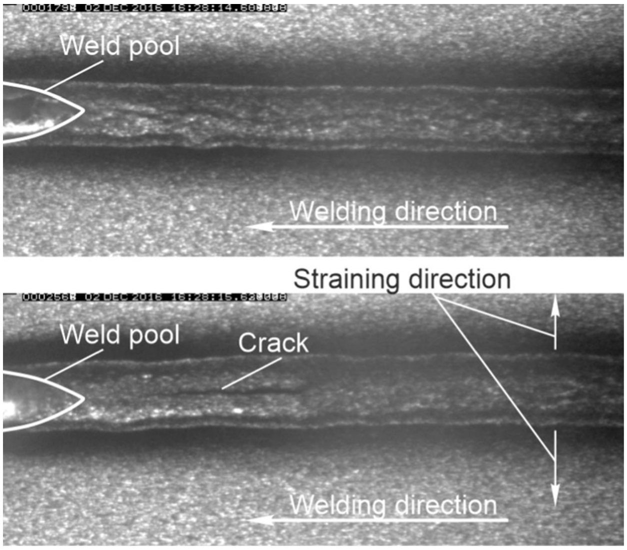


Fig. 7 Two images show the weld seam during welding, (top) before crack formation, (bottom) after crack formation

4.4 Data labelling

The welded specimen were examined by X-ray after welding to determine the location of the crack in the welded specimen. The X-ray images were the ground truth, used to precisely label the area of the image in the videos where the crack started and propagated. The X-ray images allow the precise identification of the frames in which crack formation, propagation, and termination are visible. Figure 8 shows an example of an X-ray image of a cracked weld specimen made of steel grade 304.

4.5 Data pre-processing

The data need to be pre-processed before training. First, a region of interest (ROI) area near the weld pool is selected to reduce the calculation cost. The resolution of 64 × 128 pixels is sufficient, as further increasing the area no longer improves the accuracy, but increases the latency. Figure 9 shows the clipped images at different stages, (a) is the normal state, (b) is the stage when the crack begins to initiate, which is difficult to distinguish from the normal state at this time. (c) is the stage of crack propagation, and (d) is interference data that looks like a crack and may trigger a false alarm.

To expand the training samples, and increase the diversity in the data set, data augmentation methods are

Table 4 Experimental settings

	Material	CTW Strain in %	CTW Strain Rate in s^{-1}	Quantity (Videos/Images)	Name
Training set	AISI 304	5	6	4/14745	AISI304_5_6s
	H400	5	6	2/7215	h400_5_6s
Test set	AISI 304	5	8	2/6600	AISI304_5_8s
		7	6	2/9606	AISI304_7_6s
	H400	5	4	2/6600	h400_5_4s
		5	8	2/6360	h400_5_8s

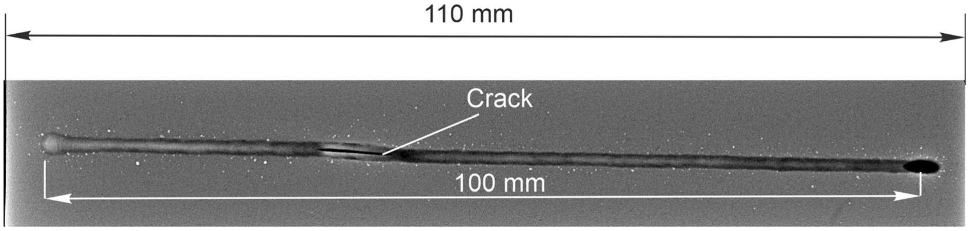


Fig. 8 X-ray image of cracked weld specimen

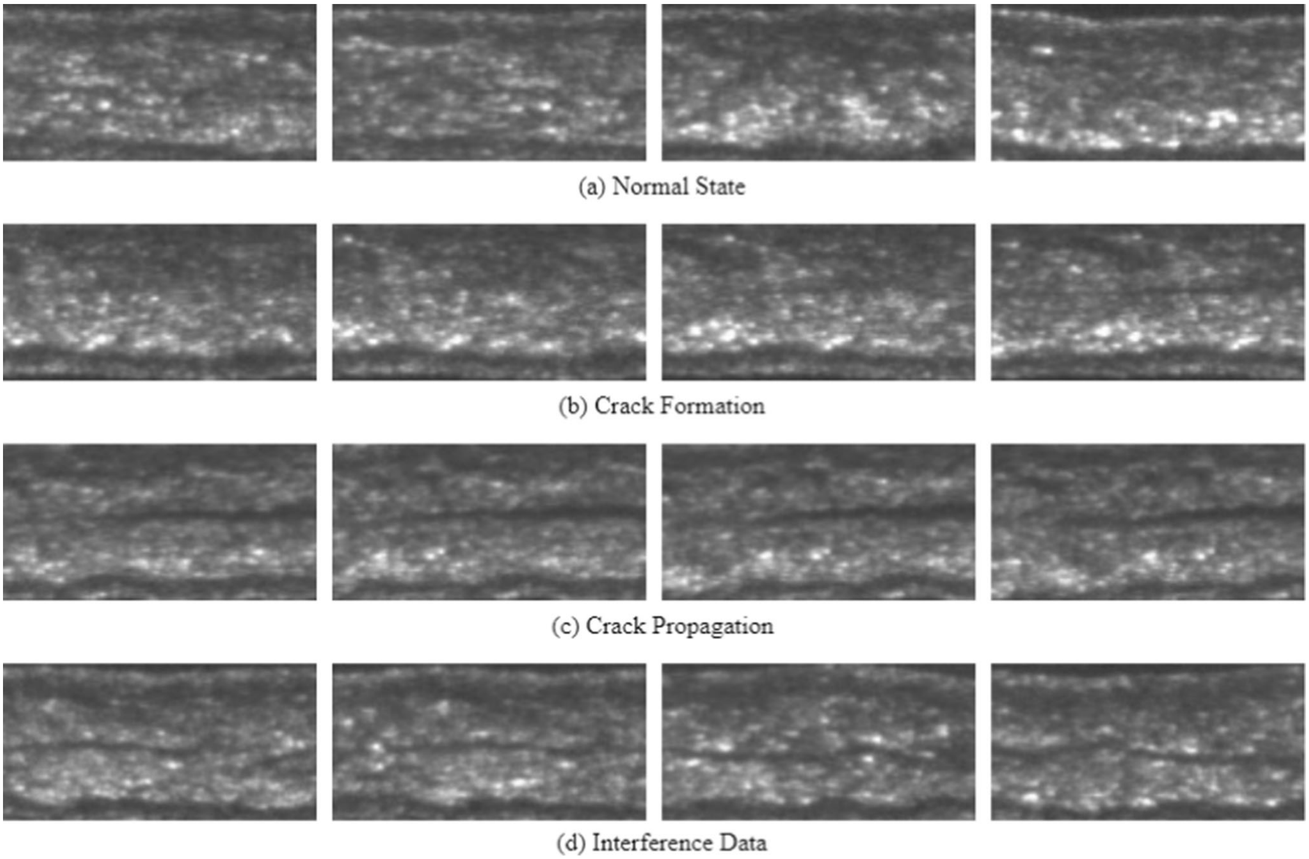


Fig. 9 Frames in different states

utilized. The images will be randomly transformed with a 50% probability in the following ways. First, the cracks in the data set are horizontal with an inclination of no more than ±15°, and all frames in the sequence can be rotated by a certain degree, as shown in (b). (c) simulates brightness variations, and (d) shows the image processed with Gaussian blur.

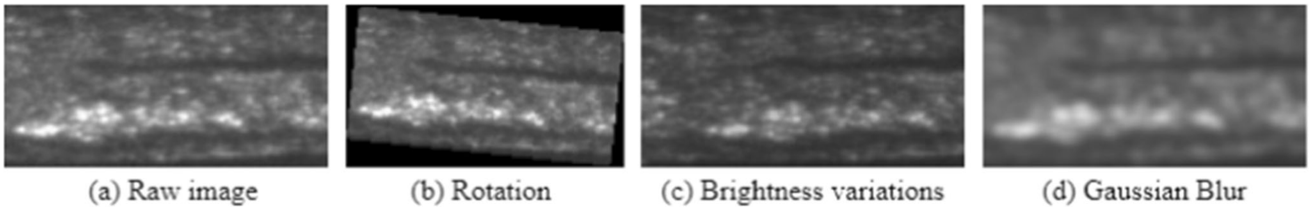

(a) Raw image (b) Rotation (c) Brightness variations (d) Gaussian Blur

Fig. 10 Types of data augmentation

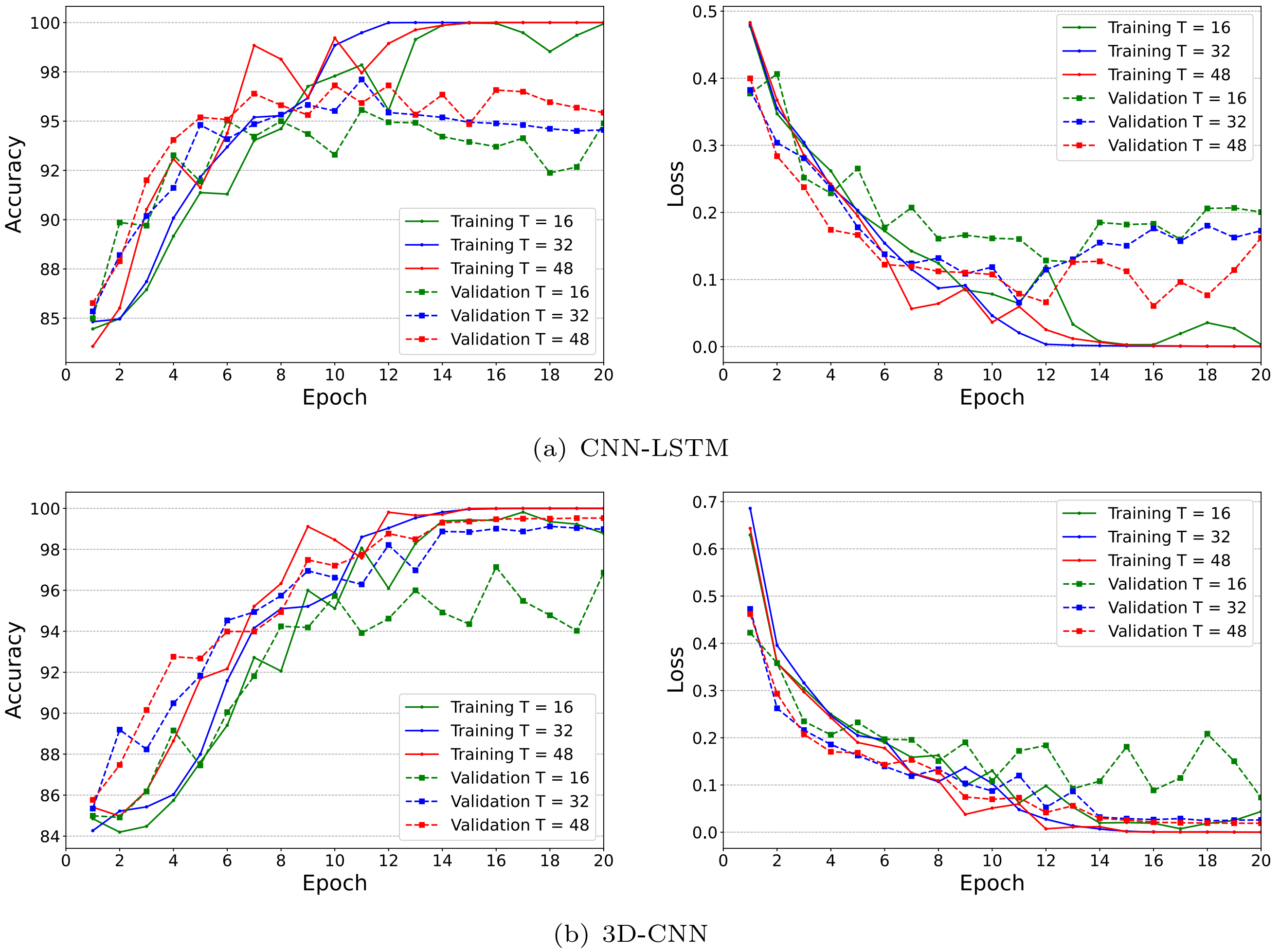
5 Results analysis

In this section, the performance of the two models is evaluated and compared. Both networks are implemented using Python and the Pytorch package [33]. They are trained on the graphical processing unit (GPU) NVIDIA Tesla P100 with 25 GB RAM. The accuracy and loss in the training process is presented first. Then, the accuracy and calculation cost are compared on test sets and, finally, the visualization of the optimal model is given.

5.1 Training results

Figure 11 shows the results of training and validation with respect to accuracy and loss.

To train the temporal model, the input is a video clip composed of consecutive frames; thus, a *T* frame wide window is slid over the videos to construct the input. In order to find the influence of sequence lengths, models with different window size *T* are built. The accuracy and loss in Fig. 11 show that the model converges faster with increasing window size *T* because longer motion and temporal information can be captured. For the CNN-



(a) CNN-LSTM

(b) 3D-CNN

Fig. 11 Comparison of accuracy and loss on training and validation sets

LSTM, when T is 16, the maximum accuracy on the validation set is 95.56% at the 11th epoch. The models with $T = 32$ and 48 can achieve the same accuracy already at the 9th and 7th epochs, respectively. For the 3D-CNN, the maximum accuracy is $97.13\%(T = 16), 99.12\%, (T = 32)$ and $99.52\%(T = 48)$ at the 16th, 18th and 19th epochs, respectively. Overall, as the window size T increases, the 3D-CNN can achieve higher accuracy on the validation set. Also in comparison with the CNN-LSTM, accuracy and loss are both better for the 3D-CNN.

5.2 Evaluation results

Next, we will evaluate the performance of the trained models through the test sets introduced in Table 4.

We note that accuracy is not a robust measure when dealing with unbalanced classes, as we find them in the welding data sets. To improve the model evaluation and better compare the performance of the models, in addition to model accuracy, some application-driven metrics such as sensitivity and specificity are introduced. The metrics can be calculated as defined in Eq. (5). True positives (TP) denotes the cracks correctly predicted, false positives (FP) denotes the normal frames that are mispredicted as showing cracks, and true negatives (TN) and false negatives (FN) denote the normal frames that are correctly and incorrectly predicted. Sensitivity indicates the proportion of true positives in real labelled samples, and specificity indicates that of negative cases. They both denote the ability of the model to correctly identify the positive and negative samples. In our crack formation detection application, early detection of cracks is very important, which some false alarms can be accepted. Therefore, sensitivity is the most important metric.

$$
\begin{aligned}
\text{Accuracy} &= (\text{TP} + \text{TN})/(\text{TN} + \text{FP} + \text{TP} + \text{FN}) \\
\text{Sensitivity} &= \text{TP}/(\text{TP} + \text{FN}) \qquad (5) \\
\text{Specificity} &= \text{TN}/(\text{TN} + \text{FP})
\end{aligned}
$$

As mentioned in Sect. 3.2, the use of very deep and complex networks may result in degraded performance, as welding videos only involve simple features. Therefore, ResNet-18 is used as a model for comparison, and the maximum number of channels is reduced to 196, which improves accuracy and inference time, and is consistent with our proposed networks. Figure 12 shows the overall accuracy, sensitivity, and specificity of ResNet-18 on four test sets. The sensitivity is not very high because the frames at the beginning of crack formation cannot be detected successfully.

Figures 13 and 14 show the results in three aspects of CNN-LSTM and 3D-CNN. It can be seen that compared to ResNet-18, the sensitivity of CNN-LSTM and 3D-CNN is significantly improved. Consistent with the results in the training process, the accuracy gradually improves with the increase in T. On the test set h400_5_4s, although the accuracy of 3D-CNN decreases, the sensitivity is greatly improved to values well above 90%.

The precision–recall curve (PR curve) shows the ability to detect positive samples and is more sensitive to imbalanced data. Precision represents the proportion of true positives in the positive samples predicted by the classifier. It reflects whether the model can detect more positive samples with fewer FP results. Recall is equal to sensitivity (Eq. 6). The PR curve is produced by calculating the precision and recall using a different threshold (prediction probability). The value of the average precision (AP) is equal to the area under the PR curve. The model with higher AP is better because precision and recall are both high. From Fig. 15, it can be seen that when T is set to 16

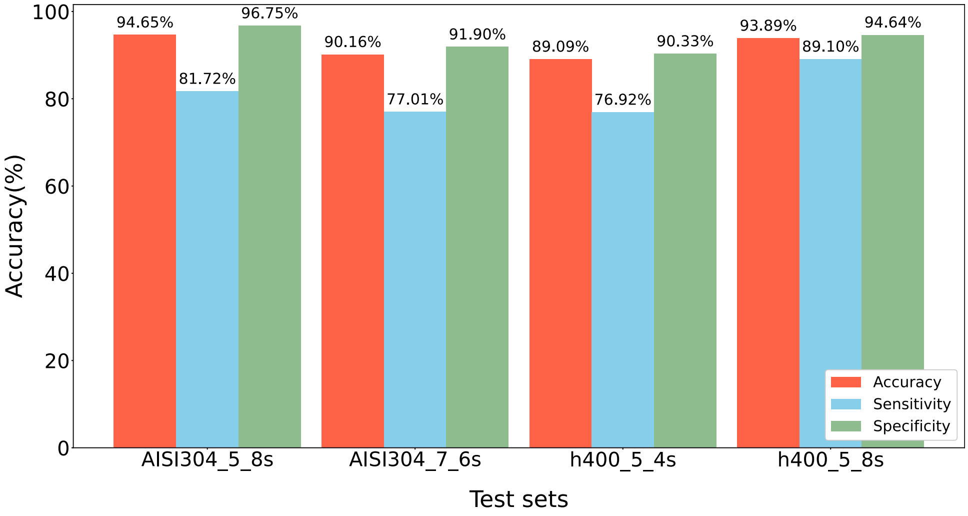
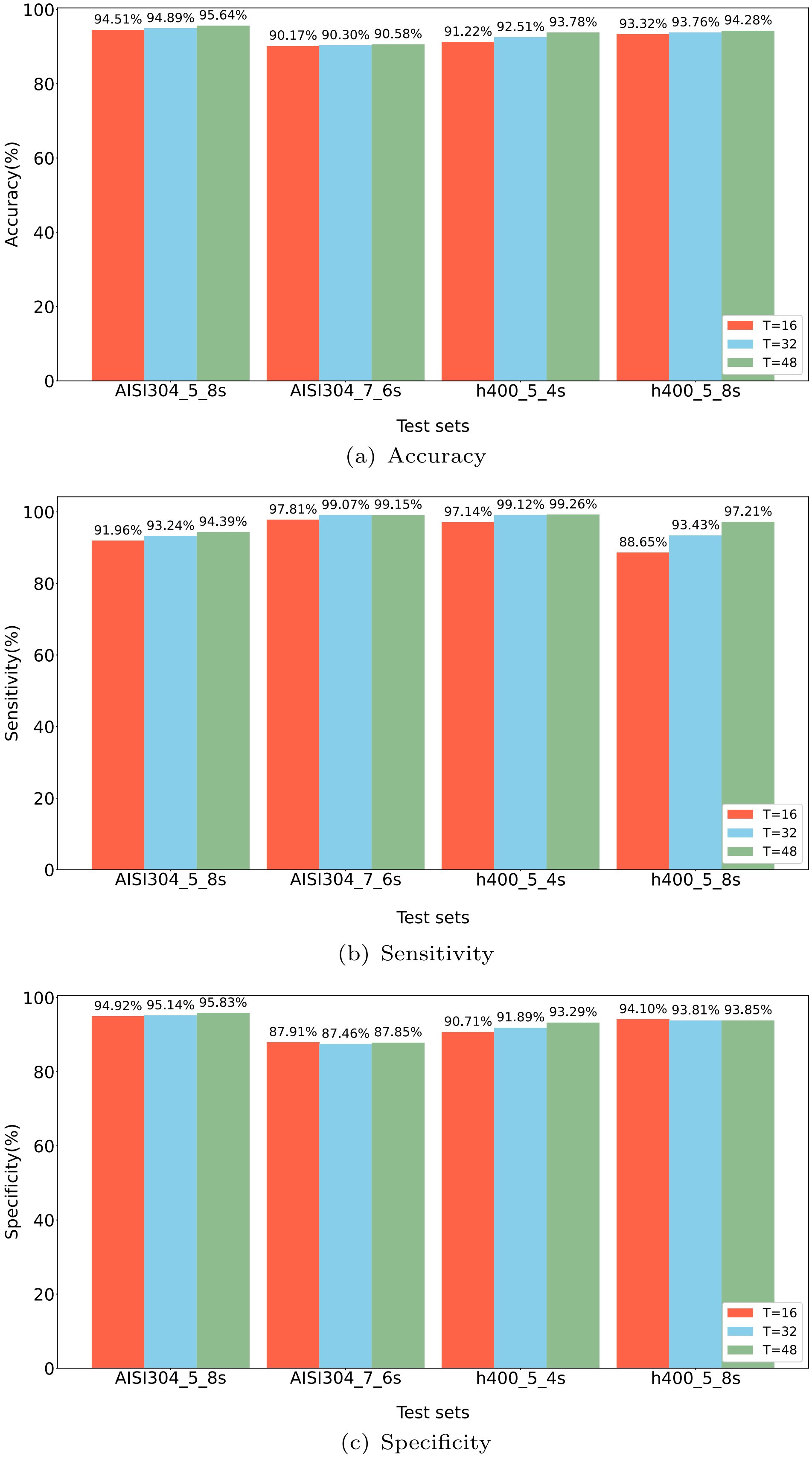


Fig. 12 Results of ResNet-18 on test sets

Fig. 13 Results of CNN-LSTM on test sets

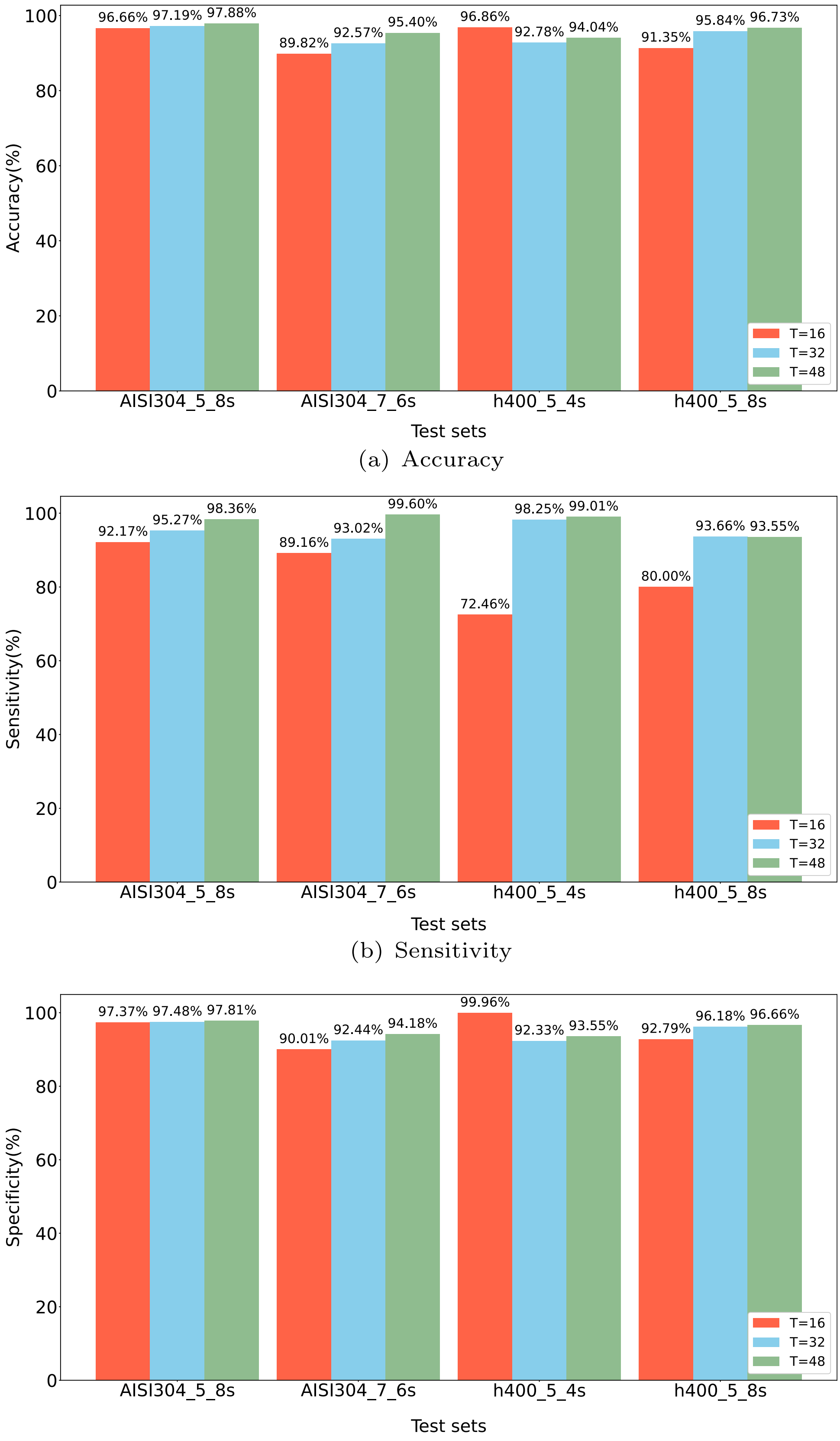


(a) Accuracy

(b) Sensitivity

(c) Specificity

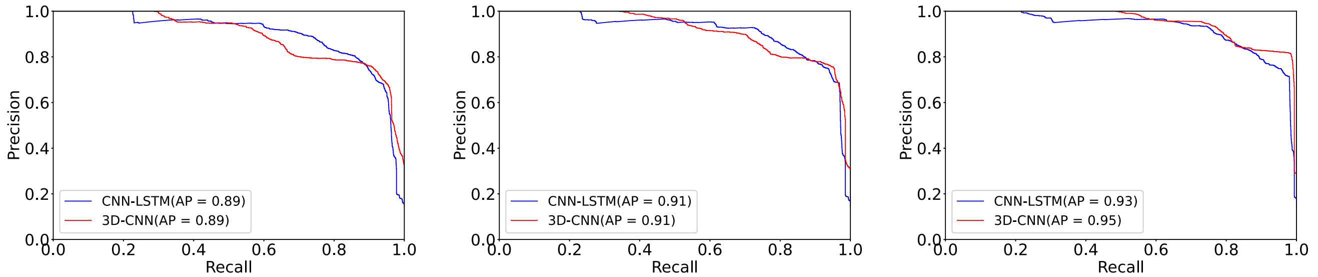
Fig. 14 Results of 3D-CNN on test sets



(a) Accuracy

(b) Sensitivity

(c) Specificity



(a) T=16 (b) T=32 (c) T=48

Fig. 15 Precision–recall curves of different input length

and 32, the APs of the two models are basically equal. When T is 48, the result of 3D-CNN is slightly better than that of CNN-LSTM.

$$\begin{aligned} \text{Precision} &= \text{TP}/(\text{TP}+\text{FP}) \\ \text{Recall} &= \text{TP}/(\text{TP}+\text{FN}) \end{aligned} \tag{6}$$

Figure 16 displays the prediction result of the test set AISI304_5_8s, the ordinate in the figure is the predicted probability, and the number of frames at which cracks appear and end is given. After frame 1005, a crack starts to appear. ResNet-18 issues an alarm at frame 1041, when the crack has formed and propagated. The frames before and after the crack formation are very similar to the start and end of the crack region. Accurately predicting the number of frames is challenging, and raising an alarm before cracks occur is actually better. When the input length is 16, the CNN-LSTM and 3D-CNN models give predictions at frames 1000 and 1001. When the input window length increases to 48, they make predictions 5 and 21 frames in advance as marked in the figure. The frames at the end of a crack until it disappears are not detected. This is because the motion features are not obvious and cannot be distinguished from interference data. In our scenario, it is more important to detect the occurrence moment of the crack formation than its end, and therefore, missing the end of the crack does not define an error for us. Taking more images as input can predict the formation of cracks earlier, but it also requires more memory and processing time. The next section will discuss the computational efficiency of different input lengths.

5.3 Computation cost

In this section, the models are evaluated with respect to their computational cost, which is particularly important for real-time detection. Table 5 lists the processing time of the two algorithms at different sequence lengths on the test set AISI304_5_8s, which contains over 3000 frames, as well as that of ResNet-18 per frame. We find that the processing time of CNN-LSTM and 3D-CNN is very similar. It increases linearly with the sequence length.

The processing time of ResNet-18 is 0.39 ms, which is much faster than the processing time of both our models. It must be mentioned that the processing times are not comparable because ResNet-18 processes only one image at the time, while the temporal models need to process multiple images in each step.

However, we have considered possibilities to speed up the processing time of a larger sequence of frames. It is computationally very inefficient if the sliding window only moves by one frame at a time, because more than 90% of the frames will be processed repeatedly. The displacement between two consecutive frames is very small, thus increasing the distance in a sliding step can improve efficiency while maintaining accuracy. Figure 17 shows the prediction results when T is 48, and the sliding step size is 1, 12, 24, and 36 frames, which means the overlap of the sequences is $98\%, 75\%, 50\%$ and 25%. From the small image in Fig. 17a, one can see that increasing the sliding step size will greatly reduce the inference time. When overlapping only 25%, the detected crack initiation frame is delayed by more than 20 frames. But when the overlapping is 50% or 75%, there is only a few frames difference. In terms of efficiency, a 75% overlap can achieve the same processing time as ResNet-18, while 50% overlap is twice as fast. In summary, when a quick alarm is required so that the welding process can be reset or adjusted promptly, T can be set to 48 and the sliding step is one quarter or half of T, which can meet both accuracy and efficiency requirements.

5.4 Visualization

Since deep learning is treated as a black box, in order to give an interpretation to its output, two visualization methods are applied to the 3D-CNN model, guided backpropagation [34] and gradient-weighted class activation mapping (Grad-CAM) [35]. Figure 18 illustrates the visualization results. Figure 18a shows the original frames input to the network, including the normal, crack

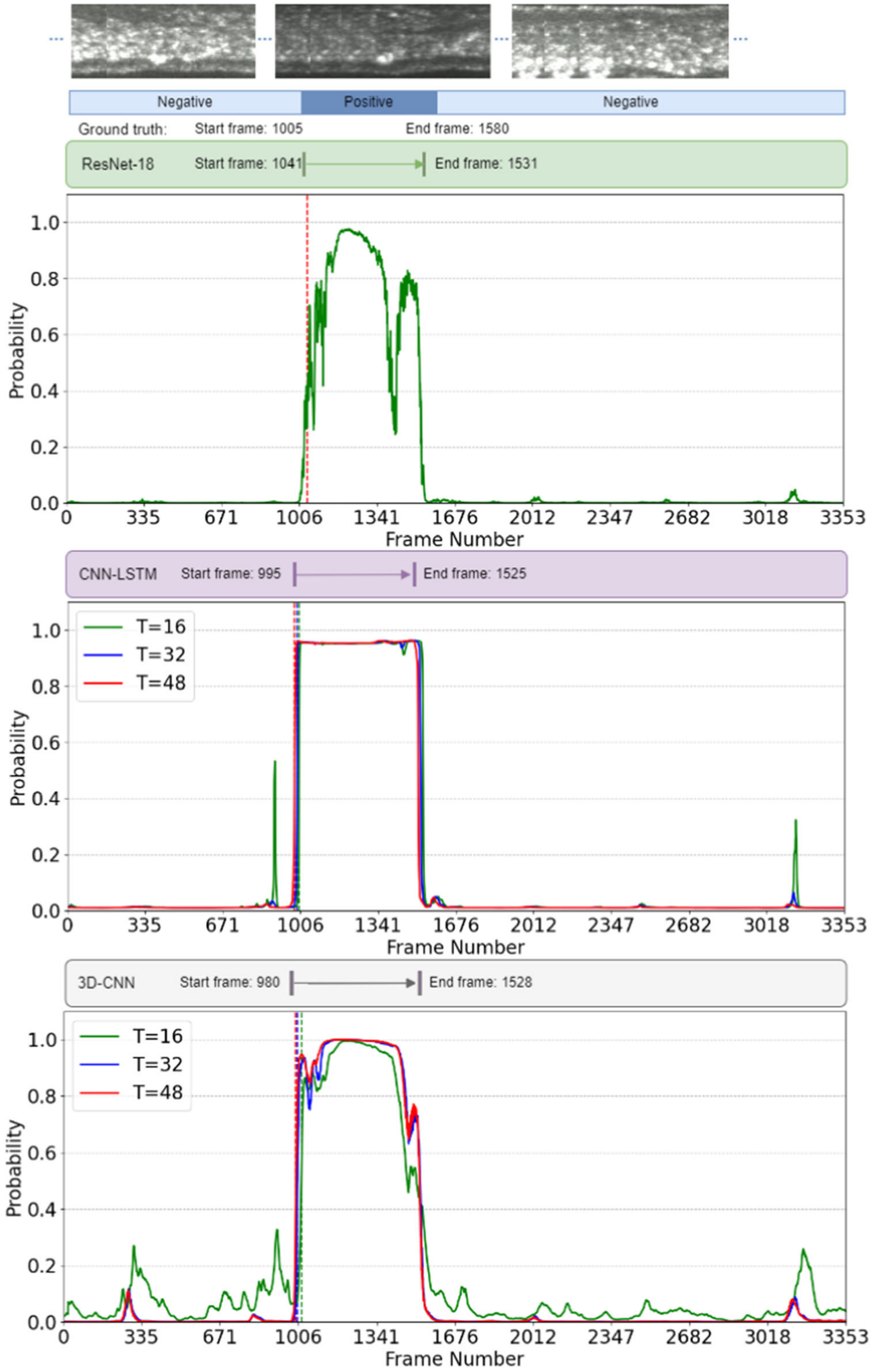


Fig. 16 Prediction results of the three models

formation, propagation, and end stages. Figure 18b shows the reconstructed images obtained by inverting the feature map at the last layer through guided backpropagation. From the figure one can see that a crack is the most discriminative region. Figure 18c shows heat maps generated by Grad-CAM. It reflects the importance of spatial locations calculated by the feature map of the last convolutional layer and the predicted class. The crack regions are highlighted as they are considered important for the final crack prediction.

Table 5 Calculation cost

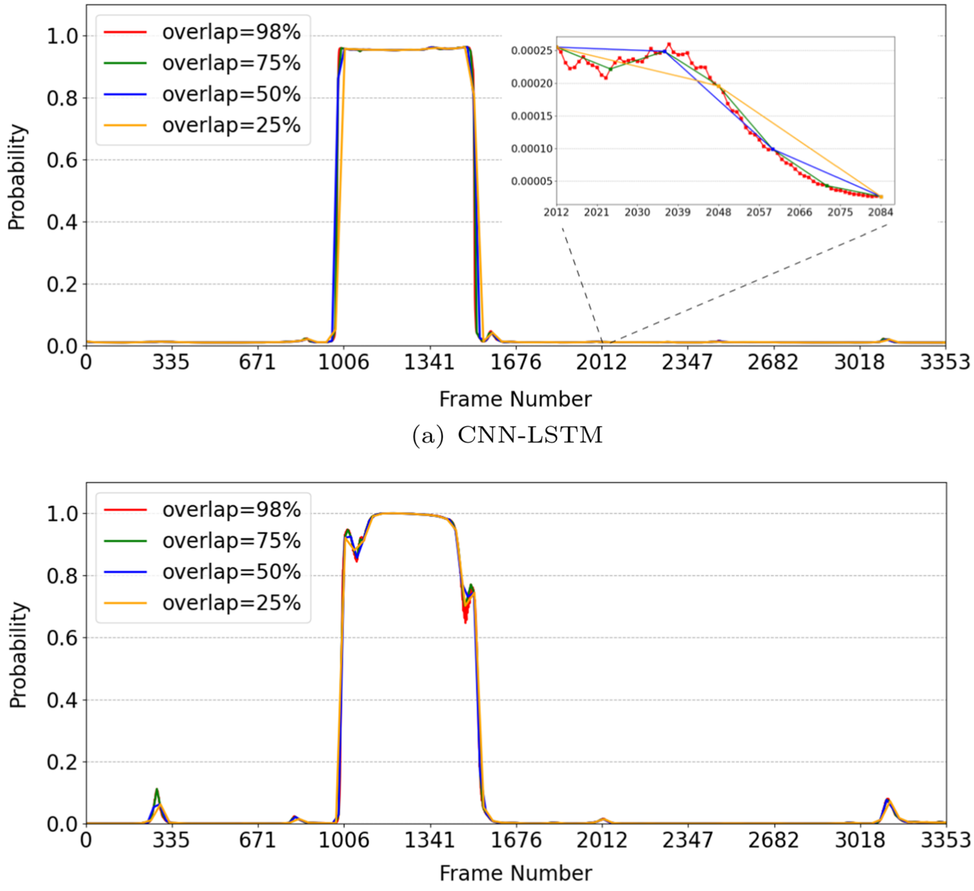
Model	Parameter	Length	FLOPs	Time (ms)
ResNet-18	3.2M	–	0.2G	0.39
CNN-LSTM	1.3M	16	0.6G	2.01
		32	1.2G	4.16
		48	1.8G	5.88
3D-CNN	2.2M	16	0.9G	2.13
		32	1.0G	4.11
		48	1.4G	5.51

6 Conclusions

In this paper, we have developed and studied two machine learning models based on CNN-LSTM and 3D-CNN to predict a common defect in welding processes, the solidification crack. To the best of our knowledge, we are the first to analyse welding videos instead of static images of welded specimen. While the method in this paper is used for laser welding, it is surely also applicable to other welding industrial applications monitored by weld pool videos. The evaluation on high-quality images collected by a high-speed camera shows that both methods can learn temporal features from image sequences and detect the range of crack formation accurately. The model 3D-CNN is slightly better than CNN-LSTM. The results are very promising and encourage us to explore this field further. The accuracy can be improved by stacking more images as input, but this will also linearly increase the processing time. This problem can be solved by increasing the moving step size of the sliding window.

In real-time monitoring systems, time efficiency is very important. In the future, we plan to further optimize the model with respect to its speed while maintaining high accuracy. We will study some lightweight networks, such as depthwise separable convolution in MobileNet [36] and pointwise group convolution in ShuffleNet [37]. On the other hand, a new kind of welding experiments will be conducted to generate a challenging data set completely different from the one used in this paper. The new data set will contain cracks that are inside the material and invisible from the surface, so they can only be located through the strain fields. In order to solve the problem of high computational overhead in the strain calculation, we will

Fig. 17 Prediction results of different overlapping sequences



(a) CNN-LSTM

(b) 3D-CNN

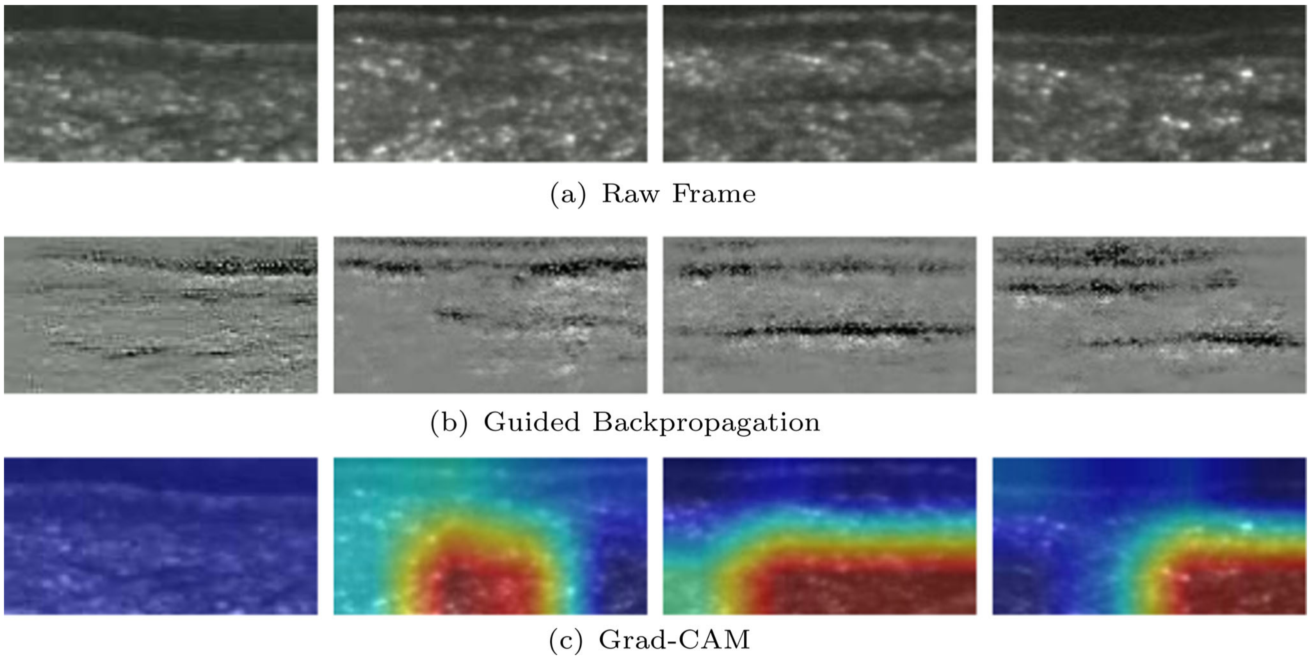


Fig. 18 Different visualization algorithms results

explore predicting strain fields with a machine learning algorithm, and then detecting cracks based on the predicted strain value.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was funded by the China Scholarship Council file Nr. 202008610227 and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project Nr. 434946896 and project Nr. 465316565.

Data availability The data sets generated during and analysed during the current study are available from the corresponding author on reasonable request.

References

1. Bakir N, Gumenyuk A, Pavlov V, Volvenko S, Rethmeier M (2020) In situ determination of the critical straining condition for solidification cracking during laser beam welding. Procedia CIRP 94:666–670
2. Bakir N, Gumenyuk A, Rethmeier M (2018) Investigation of solidification cracking susceptibility during laser beam welding using an in-situ observation technique. Sci Technol Weld Join 23(3):234–240
3. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
4. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
5. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. Adv Neural Inf Process Syst 27
6. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
7. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2625–2634
8. Ji S, Xu W, Yang M, Yu K (2012) 3D convolutional neural networks for human action recognition. IEEE Trans Pattern Anal Mach Intell 35(1):221–231
9. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 4489–4497
10. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International conference on computer vision, pp 4489–4497
11. Tao H, Cheng L, Qiu J, Stojanovic V (2022) Few shot cross equipment fault diagnosis method based on parameter optimization and feature mertic. Meas Sci Technol 33(11):115005

12. Tao H, Qiu J, Chen Y, Stojanovic V, Cheng L (2023) Unsupervised cross-domain rolling bearing fault diagnosis based on time-frequency information fusion. J Franklin Inst 360(2):1454–1477
13. Cha Y-J, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput Aided Civ Infrastruct Eng 32(5):361–378
14. Chen F-C, Jahanshahi MR (2017) NB-CNN: deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion. IEEE Trans Ind Electron 65(5):4392–4400
15. Kim B, Yuvaraj N, Sri Preethaa K, Arun Pandian R (2021) Surface crack detection using deep learning with shallow CNN architecture for enhanced computation. Neural Comput Appl 33:9289–9305
16. Quan J, Ge B, Wang M (2023) CrackViT: a unified CNN-transformer model for pixel-level crack extraction. Neural Comput Appl PP 1–17
17. Li B, Wang KC, Zhang A, Yang E, Wang G (2020) Automatic classification of pavement crack using deep convolutional neural network. Int J Pavement Eng 21(4):457–463
18. Silva WRLd, Lucena DSd (2018) Concrete cracks detection based on deep learning image classification. In: Proceedings, vol. 2. MDPI, p 489
19. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
20. Dorafshan S, Thomas RJ, Maguire M (2018) Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. Constr Build Mater 186:1031–1045
21. Wu R-T, Singla A, Jahanshahi MR, Bertino E, Ko BJ, Verma D (2019) Pruning deep convolutional neural networks for efficient edge computing in condition assessment of infrastructures. Comput Aided Civ Infrastruct Eng 34(9):774–789
22. Rao AS, Nguyen T, Palaniswami M, Ngo T (2021) Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure. Struct Health Monit 20(4):2124–2142
23. Yang Y, Pan L, Ma J, Yang R, Zhu Y, Yang Y, Zhang L (2020) A high-performance deep learning algorithm for the automated optical inspection of laser welding. Appl Sci 10(3):933
24. Zhang Y, You D, Gao X, Zhang N, Gao PP (2019) Welding defects detection based on deep learning with multiple optical sensors during disk laser welding of thick plates. J Manuf Syst 51:87–94
25. Miao R, Shan Z, Zhou Q, Wu Y, Ge L, Zhang J, Hu H (2022) Real-time defect identification of narrow overlap welds and application based on convolutional neural networks. J Manuf Syst 62:800–810
26. Shevchik S, Le-Quang T, Meylan B, Farahani FV, Olbinado MP, Rack A, Masinelli G, Leinenbach C, Wasmer K (2020) Supervised deep learning for real-time quality monitoring of laser welding with x-ray radiographic guidance. Sci Rep 10(1):1–12
27. Knaak C, Eßen J, Kröger M, Schulze F, Abels P, Gillner A (2021) A spatio-temporal ensemble deep learning architecture for real-time defect detection during laser welding on low power embedded computing boards. Sensors 21(12):4205
28. Hong Y, Yang M, Jiang Y, Du D, Chang B (2022) Real-time quality monitoring of ultra-thin sheets edge welding based on micro-vision sensing and SOCIFS-SYM. IEEE Trans Ind Inform 19:5506–5516
29. Hasan M, Choi J, Neumann J, Roy-Chowdhury AK, Davis LS (2016) Learning temporal regularity in video sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 733–742
30. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
31. He K, Sun J (2015) Convolutional neural networks at constrained time cost. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5353–5360
32. Gokmen T, Rasch MJ, Haensch W (2018) Training LSTM networks with resistive cross-point devices. Front Neurosci 12:745
33. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L et al (2019) Pytorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 32
34. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2014) Striving for simplicity: the all convolutional net. arXiv preprint arXiv:1412.6806
35. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
36. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
37. Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 6848–6856

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.