Qi Tian
State Key Laboratory of
Coastal and Offshore Engineering,
Dalian University of Technology,
Dalian 116024, China;
Department of Industrial and
Systems Engineering, Rutgers,
The State University of New Jersey,
Piscataway, NJ 08854
e-mail: qt32@scarletmail.rutgers.edu

Shenghan Guo
Department of Industrial and
Systems Engineering, Rutgers,
The State University of New Jersey,
Piscataway, NJ 08854
e-mail: sg888@scarletmail.rutgers.edu

Erika Melder
Department of Computer Science,
University of Maryland-College Park,
College Park, MD 20742
e-mail: erikavmelder@gmx.com;
evmelder@umd.edu

Linkan Bian
Mem. ASME,
Department of Industrial and
Systems Engineering,
Mississippi State University,
Mississippi State, MS, 39762
e-mail: lb1425@msstate.edu

Weihong "Grace" Guo[1]
Mem. ASME
Department of Industrial and
Systems Engineering, Rutgers,
The State University of New Jersey,
Piscataway, NJ 08854
e-mail: wg152@rutgers.edu

Deep Learning-Based Data Fusion Method for In Situ Porosity Detection in Laser-Based Additive Manufacturing

Laser-based additive manufacturing (LBAM) provides unrivalled design freedom with the ability to manufacture complicated parts for a wide range of engineering applications. Melt pool is one of the most important signatures in LBAM and is indicative of process anomalies and part defects. High-speed thermal images of the melt pool captured during LBAM make it possible for in situ melt pool monitoring and porosity prediction. This paper aims to broaden current knowledge of the underlying relationship between process and porosity in LBAM and provide new possibilities for efficient and accurate porosity prediction. We present a deep learning-based data fusion method to predict porosity in LBAM parts by leveraging the measured melt pool thermal history and two newly created deep learning neural networks. A PyroNet, based on Convolutional Neural Networks, is developed to correlate in-process pyrometry images with layer-wise porosity; an IRNet, based on Long-term Recurrent Convolutional Networks, is developed to correlate sequential thermal images from an infrared camera with layer-wise porosity. Predictions from PyroNet and IRNet are fused at the decision-level to obtain a more accurate prediction of layer-wise porosity. The model fidelity is validated with LBAM Ti–6Al–4V thin-wall structure. This is the first work that manages to fuse pyrometer data and infrared camera data for metal additive manufacturing (AM). The case study results based on benchmark datasets show that our method can achieve high accuracy with relatively high efficiency, demonstrating the applicability of the method for in situ porosity detection in LBAM.

[DOI: 10.1115/1.4048957]

Keywords: laser-based additive manufacturing, in situ porosity detection, deep learning, data fusion, inspection and quality control, sensing, monitoring and diagnostics

1 Introduction

Laser-based additive manufacturing (LBAM) produces metal parts from bottom to up through layer-wise cladding, which provides unprecedented possibilities to produce complicated parts with multiple functions for a wide range of engineering applications [1]. Powder bed fusion (PBF) and direct energy deposition (DED) are two main methods in LBAM. PBF fuses the powder preplaced layer by layer selectively using the laser [2]. In DED, the material is infused into a melt pool which is used to fill in the cross-section of the part and produce the part layer by layer. This process creates large thermal gradients, leading to residual stress and plastic deformation [3], which may affect the surface integrity, microstructure, or mechanical properties of the products.

Since the shape and size of the melt pool are very important to determining the microstructure of additive manufacturing (AM) parts [4], melt pool data have been treated as one of the most important signatures for monitoring and predicting defects [5]. Some researchers develop finite element models to simulate the melt pool for parameter selection in the process planning stage [6–9], while others capture melt pool thermal images during manufacturing and correlate the thermal images with defects for in situ process monitoring [10–15].

On the physics-based simulation of the melt pool, finite element model (FEM) has been widely used for modeling the melt pool geometry and thermal distribution. For example, Romano et al. established an FEM for the AM process to simulate the melt pool thermal distribution and geometry in powder bed and compared these characteristics between different powder materials [6]. Song et al. presented a combination of FEM and experimental measurements to analyze the impact of scanning velocity or laser power on the melt pool size during the selective laser melting (SLM) process [7]. Zhuang et al. developed an FEM to model the change of melt pool dimensions and temperature field of Ti–6Al–4V powder during SLM [8]. Tian et al. established an FEM for fluid flow and heat transfer with different parameters to study the impact of different parameters on the dilution ratio of the Ni-based alloy [9]. All these methods, however, are limited to melt pool simulation, which cannot be used for in situ process monitoring or defect prediction due to the randomness in the actual manufacturing process but beyond the capability of FEMs.

In recent years, the developments in LBAM and sensors enable researchers to capture melt pool thermal behavior during the manufacturing process. High-speed thermal images of the melt pool

[1] Corresponding author.

Manuscript received June 4, 2020; final manuscript received October 17, 2020; published online December 17, 2020. Assoc. Editor: Qiang Huang.

captured during LBAM make it possible for in situ melt pool monitoring and anomaly detection. Khanzedeh et al. used tensor decomposition to extract features from thermal image streams to monitor metal-based AM [10]. Mahmoudi et al. used the melt pool thermal images as the input signal to detect layer-wise defects in PBF processes [11]. Seifi et al. built a novel model to study the relationship between thermal images and defects [12].

A major anomaly or defect with LBAM is porosity. The lack of knowledge about the underlying process–porosity relationship, along with the pressing need for efficient and accurate porosity prediction, hampers the wide adoption of LBAM parts. This need has motivated a few recent studies on porosity detection with in situ pyrometry data of the melt pool. Khanzadeh et al. proposed a self-organizing maps-based porosity prediction method considering the thermal distribution and shape of the melt pool during LBAM [13]. Khanzadeh et al. used different supervised machine learning methods to predict porosity based on the features extracted by the Functional Principal Component Analysis from the melt pool images [14]. Khanzadeh et al. analyzed the thermophysical dynamics to predict porosity during DED using thermal images of the melt pool [15]. Scime et al. used the melt pool image obtained by a high-speed camera for in situ detection of keyholing porosity and balling instabilities [16]. Mitchell et al. proposed a Gaussian filter-based method to predict porosity using melt pool pyrometer images [17].

The bulk work of the existing image-based porosity prediction methods, as reviewed earlier, focuses on extracting features from images using statistical or machine learning methods. Deep learning consists of neural networks with multiple hierarchical layers [18], thus is superior over traditional machine learning methods in finding hidden structures within large datasets and making more accurate predictions. Compared with traditional machine learning methods, which have proven to be inadequate at handling massive data efficiently, deep learning has achieved superior performance in image recognition, gait recognition, etc. [19–21], making it suitable to be used for multi-sensor image data.

Deep learning has been investigated for surface defect inspection [22], machinery fault diagnosis [23], defect prediction and residual life estimation [24], and production scheduling [25]; however, its potential in real-time monitoring of LBAM remains to be tapped. A few recent studies explored deep learning for anomaly detection in metal AM using acoustic signals [26,27] and infrared images [28,29]. Scime et al. used a powder bed camera as the sensor and presented a Convolutional Neural Network (CNN) to detect and predict the powder bed defects autonomously [28].

Typical deep learning architectures include CNN, Recurrent Neural Network (RNN), Auto Encoder, and so on. CNN is designed to transfer image data to output variables. Since the sensor data can be represented in 2D images, CNN can be adopted to effectively handle the scale and position in variant structures in image data. RNN is designed to work with sequence prediction problems. The Long-term Recurrent Convolutional Networks (LRCN) [30] have become an important part of RNNs these years due to their superiority in dealing with long-range dependencies.

Our work aims to broaden current knowledge of the underlying process–porosity causal relationship in LBAM and provide new possibilities for efficient and accurate porosity prediction. We present a deep learning-based data fusion method to predict porosity in as-LBAM parts by leveraging the measured melt pool thermal history and deep learning. PyroNet, which is a CNN-based model, is established to correlate pyrometer images with layer-wise porosity; IRNet, which is an RNN-based model, is established to correlate infrared (IR) camera images with porosity. Predictions from PyroNet and IRNet will be fused at the decision-level to obtain a more accurate prediction of layer-wise porosity. We would like to highlight that this is the first work that manages to fuse pyrometer data and IR camera data for metal AM. The pyrometer captures local heat transfer around the melt pool, whereas the stationary IR camera allows to characterize the global, between-layer thermal dynamics. While pyrometer data have been investigated in several recent studies [13–15], IR camera data have never been brought into the picture. Smart data fusion of these two sources of data will allow to significantly enhance the prediction accuracy of internal porosity.

The remainder of the paper is arranged as follows. Section 2 explains the pyrometer and IR data, including the data collection procedure, pre-processing, porosity labeling, and augmenting the imbalanced dataset. Section 3 presents the proposed decision-level data fusion framework for porosity prediction, including a PyroNet for pyrometer images and an IRNet for IR image sequences, and a fusion method to combine the results from PyroNet and IRNet. The results are analyzed in Sec. 4. Finally, Sec. 5 concludes the paper.

2 Data Characterization and Processing

In this section, the data source and pre-processing procedures are first introduced in Sec. 2.1. Section 2.2 describes how we assign porosity labels to input data. The data augmentation procedure to handle the unbalanced data issue is explained in Sec. 2.3.

2.1 Data Collection and Pre-processing. The thermal data of the melt pool in LBAM refer to Marshall et al. [31]. Specifically, a LENS™ 750 system is applied to produce a Ti–6Al–4V thin-wall structure. The system is equipped with two sensors for in-process data collection. A pyrometer (Stratonics, Inc.) measures the melt pool temperature. An infrared (IR) camera (Sierra-Olympic Technologies, Inc. Viento320) detects infrared energy during manufacturing and converts it to produce a thermal image of the melt pool. The setup of the IR sensor and pyrometer inside the LENS™ chamber is presented in Fig. 1.

It can be seen from Fig. 1 that the pyrometer is mounted above the build plate, outside the chamber. In this setup, the pyrometer can monitor the temperature of the melt pool in a vertical direction. The data of the pyrometer are output to comma separated value (CSV) files, each of which contains a 752 × 480 (width × height) matrix of temperature values. As for the IR camera, it is oriented at around 62.7 deg with respect to the middle of the build plate. The IR camera monitors the thermal changes of the sample edgewise, whose data are also output to CSV files, each of which contains a 320 × 240 (width × height) matrix of temperature values.

Due to the different scanning rates of the two sensors, 1564 pyrometer images and 6256 IR images are obtained during LBAM. This indicates that approximately every four IR images correspond to one pyrometer image. Among the 6256 IR images, some are eliminated because they do not contain any useful melt pool information. This is because these images are collected between two layers when the deposition head moves back to its original position to start printing the next layer.

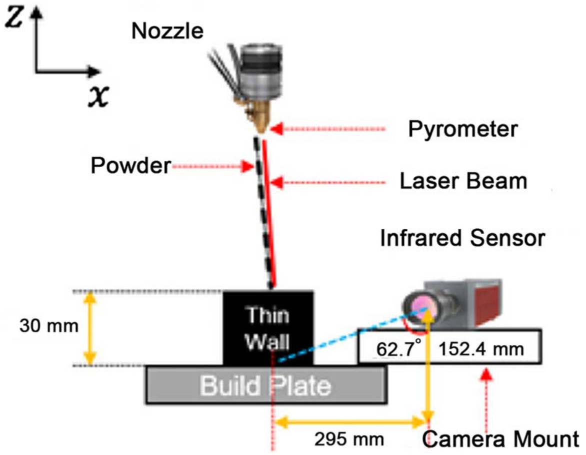

Fig. 1 Setup of the infrared camera and the pyrometer [31]

The raw images are large relative to the melt pool, which brings the need to remove some of the backgrounds and focus on the melt pool. To do so, we select a submatrix from the original data matrix, which is used to generate an Red Green Blue (RGB) image in MATLAB. Specifically, the submatrix of each pyrometer image is in rows 380–580 and columns 120–320, while the one of each IR image is in rows 125–185 and columns 50–190, both of which are focusing on the high-temperature area of the melt pool. The *jet* colormap is selected to provide the best color contrast for the melt pool temperature.

Pyrometer data have strong spatial correlations but relatively weak temporal correlations. The prediction power for porosity in these data mainly comes from their spatial correlations rather than from the neglectable temporal correlations [10,14]. Hence, this study mainly explores the spatial patterns of pyrometer data for porosity prediction. IR data, on the other hand, have much stronger

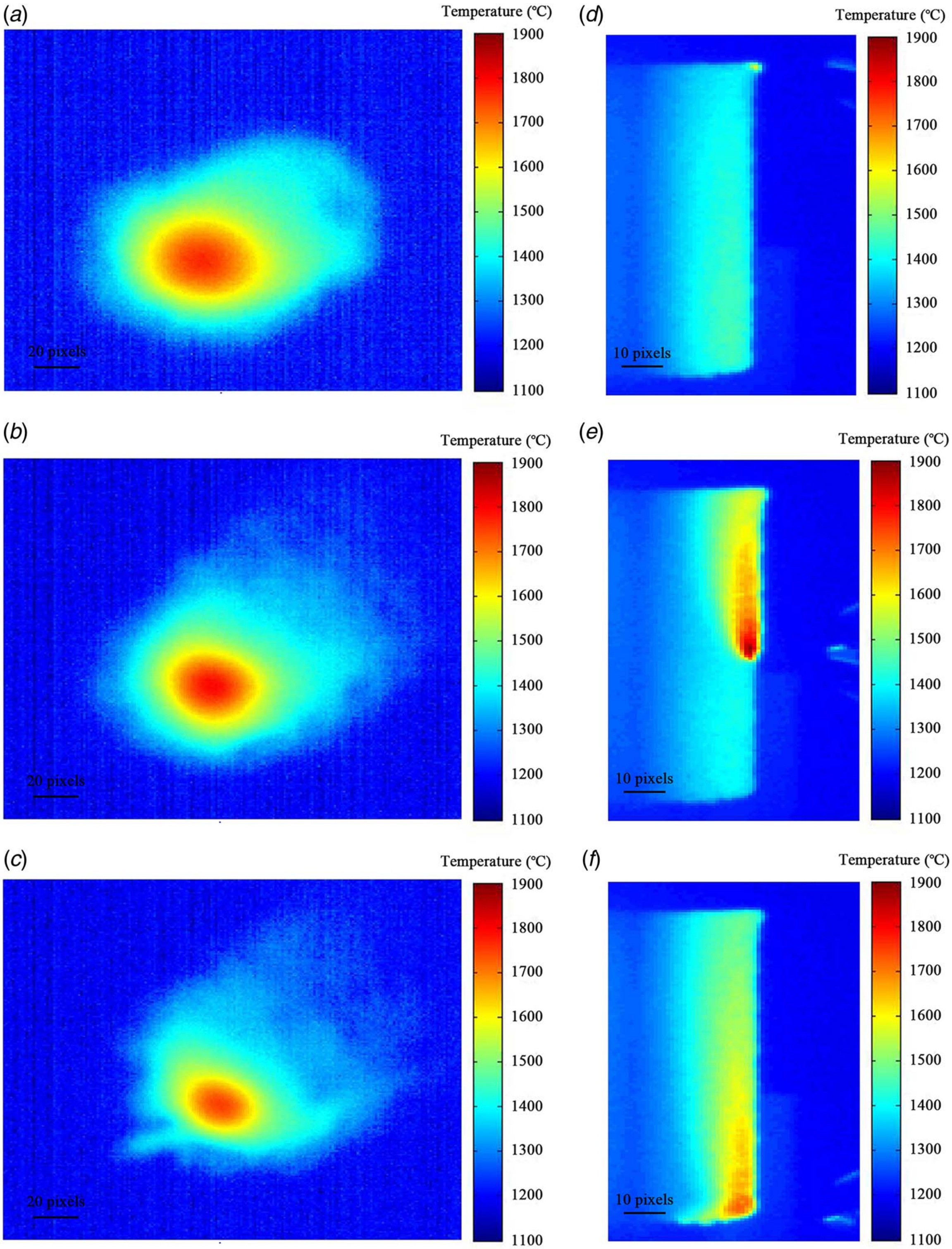

Fig. 2 First, middle, and last scans on layer 29 for (*a*)–(*c*) pyrometer data and (*d*)–(*f*) infrared camera data

temporal dependence. This is because the scanning direction of the IR camera is about 45 deg to the melt pool, making IR images very different from each other in the same layer, as shown in Fig. 2. The temporal correlations in IR images are the major source of prediction power for porosity, thus cannot be neglected. Therefore, it is reasonable to treat IR images as image sequences rather than individual images.

An image sequence contains several continuous-time images, just like the frames in a video. Having a long sequence can better preserve the long-term temporal correlations but poses challenges to the computational efforts of deep learning models. Our preliminary analysis of comparing sequences of 2, 3, 4, or 5 IR images has shown that having three continuous-time IR images as a sequence can provide the best balance between model accuracy and computational efficiency. Therefore, we will transfer every three IR images to an IR sequence.

2.2 Porosity Labeling. Supervised deep learning requires that each melt pool image must have a porosity label. In this paper, the porosity labeling process is conducted by 3D computed tomography (CT) (refer to Khanzadeh et al. [14]), giving the size and shape of the pores for each sample. For those having a pore with diameter larger than 0.05 mm, they are treated as "bad" samples with porosity, otherwise as "good" samples with neglectable porosity. In this way, pyrometer images are labeled [14]. As for the way to label IR images, they are determined following the mapping between pyrometer images and IR images. The porosity label of each IR sequence is then determined according to the majority labels of the images in that sequence. Each IR sequence is then mapped to a pyrometer image according to the timestamp of data collection.

After the above processing and labeling, we obtain 840 pyrometer images, including 774 "good" and 66 "bad" ones. A pyrometer image with the "good" label and one with the "bad" label are compared in Fig. 3. It can be seen that the "good" image has no obvious porosity while the "bad" image does. We also obtain 840 IR sequences, with the same labels as the corresponding pyrometer images.

2.3 Cross-validation With Data Augmentation. The size of pyrometer/IR data set is rather small, which may potentially cause overfitting issue. To avoid overfitting [32], 6-fold cross-validation (CV) was used to partition the pre-processed data into six parts. The random partitioning in CV adopted stratified sampling, i.e., 1/6 portion of the "good" samples and 1/6 portion of the "bad" samples, is randomly drawn from the original data (without replacement) to form a fold. Thus, in each fold, we have 129 "good" samples and 11 "bad" ones. Fivefolds are taken to form the training set, and one fold is preserved as the test set. There are total 645 "good" samples and 55 "bad" ones in the training set, and 129 "good" samples and 11 "bad" ones in the test set.

Data augmentation is a necessary processing step before the actual model training. This is because the training data size, 700, are small for building a deep learning model. Meanwhile, the imbalance between "good" and "bad" samples, i.e., 645 versus 55, is likely to compromise the learning outcome and prediction accuracy—the model trained on such data would not fully learn the population of "bad" samples and tend to make false-negative predictions. To resolve these concerns, we augment the "bad" samples in training data using bootstrapping. Bootstrapping has been used in training data/feature augmentation to improve the learning outcomes of machine learning/deep learning models [33]. Bootstrapping is used to augment the "bad" samples to 645, i.e., the same size as the "good" samples. In this way, we have a balance between the "good" and "bad" samples in the training set. Note that the bootstrapping is only used for the training set.

During model training, 1/6 of the training set is separated as the training-phase validation data, which consists of 1/6 "good" samples and 1/6 "bad" ones. The 5/6 training part is used to train the deep learning models, and the 1/6 validation part is to validate the models with suitable hyper-parameters. In summary, there are 538 "good" and 538 "bad" samples in the training part of the training set, 107 "good" and 107 "bad" samples in the validation part of the training set, 129 "good" and 11 "bad" samples in the test set. By partitioning the pyrometer data in this manner, the corresponding IR data sequences automatically formed the training/testing set in the same way.

3 Deep Learning-Based Data Fusion Method

In this section, we will first introduce the decision-level data fusion framework for porosity detection in Sec. 3.1, followed by a CNN model for pyrometer images and an LRCN model for IR data in Secs. 3.2 and 3.3, respectively.

3.1 Decision-Level Data Fusion. The proposed decision-level data fusion framework is illustrated in Fig. 4. After generating pyrometer images and IR sequences from the raw data, 3D CT is used to identify porosity; porosity labels are then assigned to both pyrometer images and IR sequences. Next, the labeled pyrometer images are fed into a CNN model (VGG16), and the labeled IR sequences are fed into an LRCN model to train the supervised deep learning models. Note that the pyrometer data and IR data are used separately; the two models are trained separately.

After model training, the well-trained VGG16 model and LRCN model are used to predict the porosity condition for the test set pyrometer images and IR sequences. The predicted probability of the ith pyrometer image being "good" is denoted as $\hat{p}_{pyro}(i)$; the predicted probability of the ith IR sequence being "good" is $\hat{p}_{IR}(i)$.

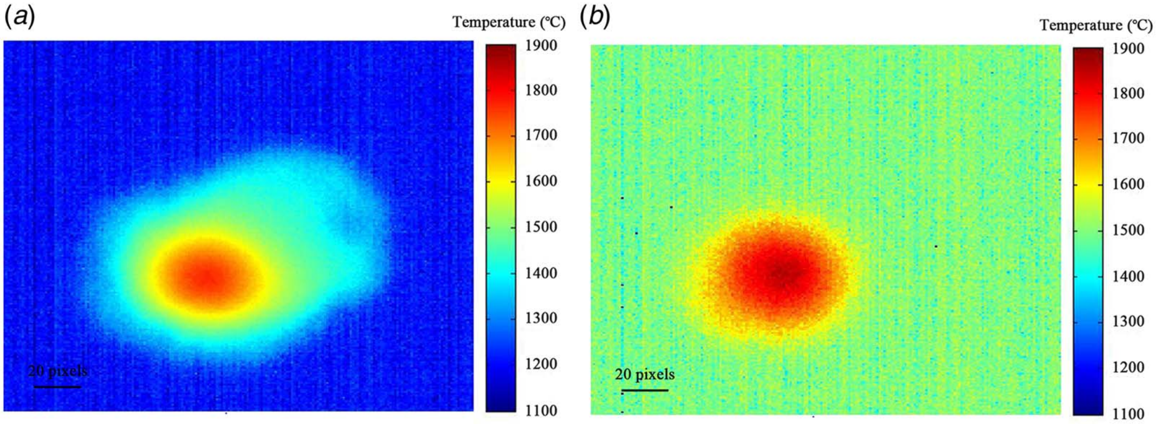

Fig. 3 (*a*) A pyrometer image with "good" label and (*b*) a pyrometer image with "bad" label

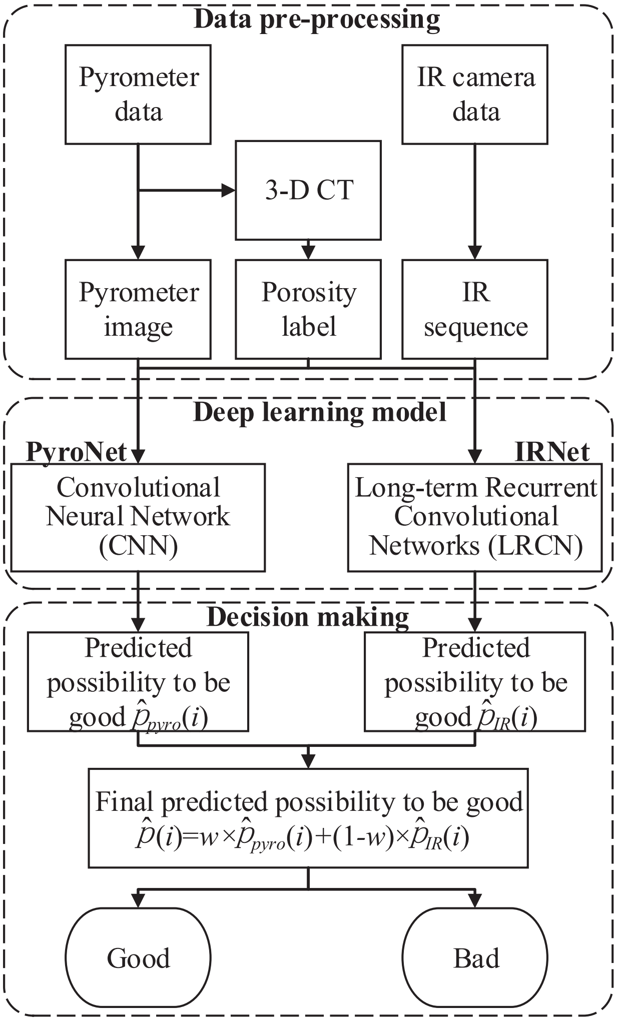
Data pre-processing
Pyrometer data
IR camera data
3-D CT
Pyrometer image
Porosity label
IR sequence
Deep learning model
PyroNet
IRNet
Convolutional Neural Network (CNN)
Long-term Recurrent Convolutional Networks (LRCN)
Decision making
Predicted possibility to be good $\hat{p}_{pyro}(i)$
Predicted possibility to be good $\hat{p}_{IR}(i)$
Final predicted possibility to be good
$\hat{p}(i)=w\times\hat{p}_{pyro}(i)+(1\text{-}w)\times\hat{p}_{IR}(i)$
Good
Bad

Fig. 4 Decision-level data fusion framework

Therefore, each melt pool receives two predictions, which are then fused according to a weighted average of the two predicted probabilities. The final predicted probability of the *i*th sample to be "good," $\hat{p}(i)$, can be calculated by

$$\hat{p}(i) = w \cdot \hat{p}_{pyro}(i) + (1 - w) \cdot \hat{p}_{IR}(i) \tag{1}$$

where $w \in [0, 1]$ is the weighting factor of pyrometer data. If $\hat{p}(i)$ is larger than 0.5, the *i*th sample will be predicted as a "good" sample with neglectable porosity; otherwise, it will be detected as a "bad" sample and that there is porosity.

3.2 Convolutional Neural Network Model for Pyrometer Data—PyroNet. We develop a CNN model, called PyroNet, for the pyrometer data. The PyroNet adopts the classical VGG16 [34] structure for the following advantages. First, VGG16 adopts consecutive 3×3 convolution cores instead of the lager convolution nucleus (such as 11×11, 5×5) in traditional deep learning structure. The smaller convolution kernels can increase the depth of the network and make the network able to learn more complex patterns. Second, the number of parameters in the VGG16 network is small, which can help reduce the computational time. The modified VGG16 architecture for PyroNet is shown in Fig. 5.

The input of PyroNet is RGB pyrometer images with a resolution of 224×224 pixels (width × height) denoted as $\boldsymbol{H}_i$, where i is the index of the pyrometer image and $i = 1, 2, \ldots$. The VGG16 structure mainly contains convolutional layers, max-pooling layers, flatten layer, drop-out layers, and fully connected (FC) layers. The convolutional layers are used to extract features from the input images and the previous convolutional layer, and they have weights that need to be trained. As for the max-pooling layers, they are used to reduce the number of parameters and computation by down-sampling the representation. Flatten layer, just as its name implies, is used to flatten the data matrix into vectors which can be dealt with FC layers. Drop-out layers are used to delete some parameters to avoid overfitting. Finally, the FC layers are mainly used to take the results of the convolution/pooling process and use them to classify the image into a label, thus finish the prediction. Details about the hyper-parameters in our PyroNet are shown in Fig. 6.

In PyroNet, the pyrometer images $\boldsymbol{H}_i$ will go through five blocks that are used to extract features (shown as Conv1_x, Conv2_x, Conv3_x, Conv4_x, and Conv5_x in Fig. 6). Each of these blocks is composed of several convolutional layers and end up with a max-pooling layer. Specifically, the kernel size of the convolution layers is always 3, which means that at a convolutional layer, a $3\times3\times P$ kernel, $\boldsymbol{h}$, is performed on an $M\times N\times P$ input map, $\boldsymbol{x} = (x)_{mnp} \in \mathbb{R}^{M\times N\times P}$, to generate an output map, $\boldsymbol{x}' = (x')_{mnq} \in \mathbb{R}^{M\times N\times Q}$, of Q channels

$$x'_{mnq} = (\boldsymbol{h} * \boldsymbol{x})_{mnq} = \sum_{i_3=-Q/2}^{Q/2} \sum_{i_2=-1}^{1} \sum_{i_1=-1}^{1} h_{i_1 i_2 i_3} \cdot x_{m-i_1,\, n-i_2,\, q-i_3} \tag{2}$$

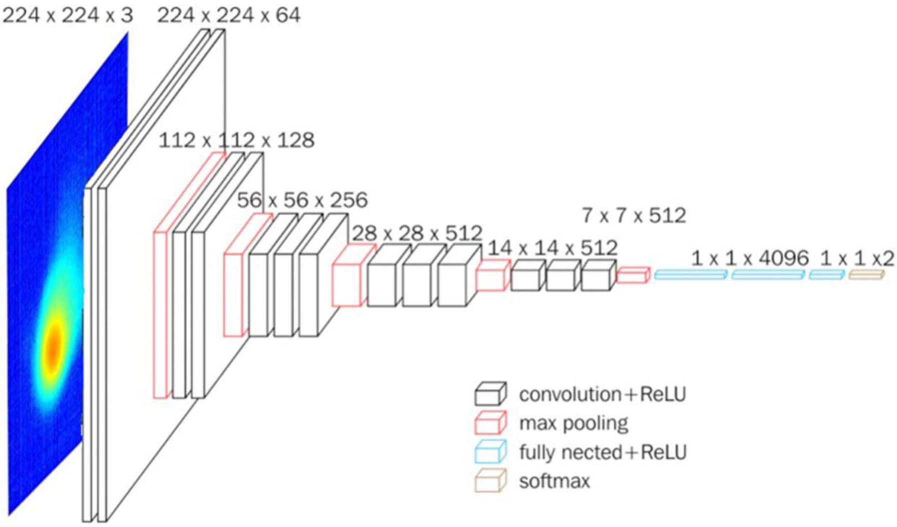

Fig. 5 VGG16 architecture for the PyroNet, adapted from Ref. [35]

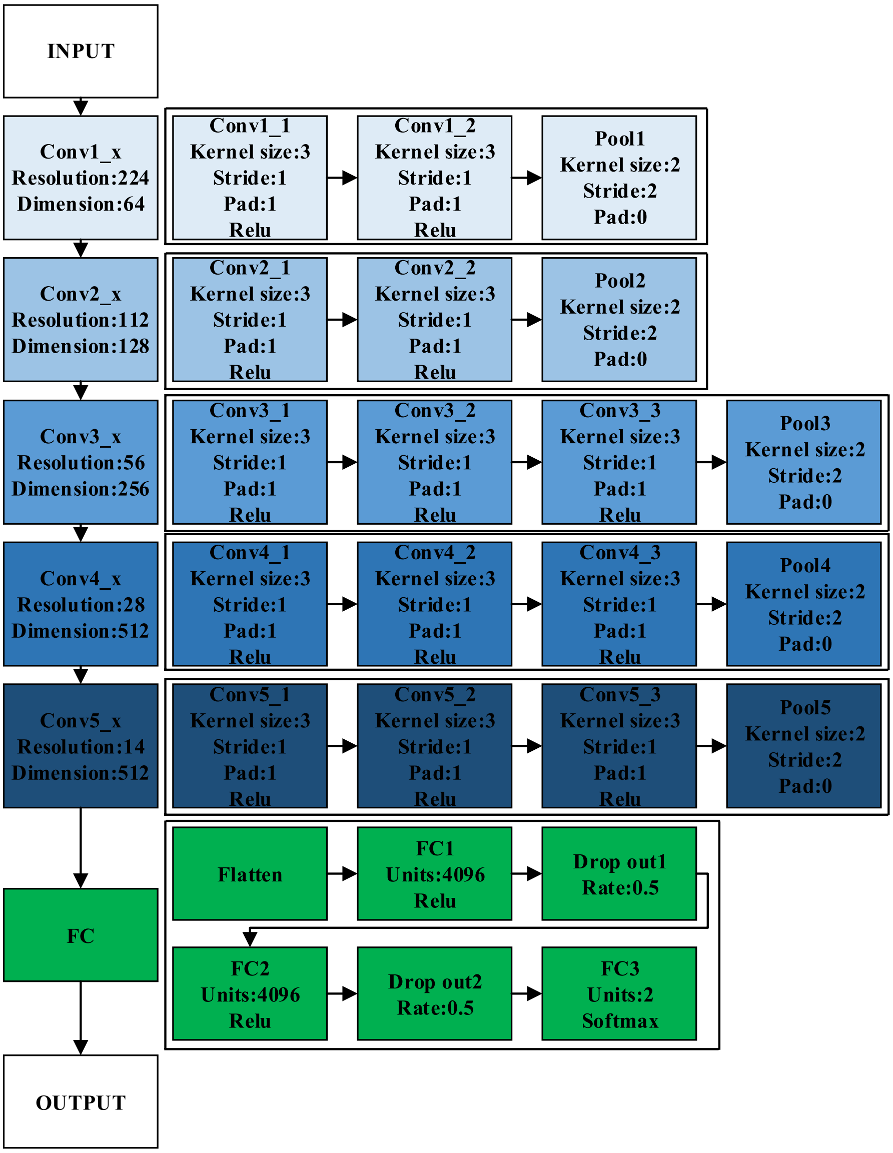

Fig. 6 Hyper-parameters in the PyroNet

where * is the convolutional operator. The activation function of these convolution layers is ReLU [36] (i.e., rectified linear unit), which is used to remove the negative values, and follows a quite simple function $\boldsymbol{y} = (y)_{mnq} \in \mathbb{R}^{M\times N\times Q}$, $y_{mnq} = \max(0,\ x'_{mnq})$. The stride of 1 and pad of 1 are used for all the convolutional layers. The max-pooling layer with a kernel size of 2 then reduces the revolution of $\boldsymbol{y}$ and improves the robustness of learned features. In max-pooling, the $R\times C\times Q$ output map, $\boldsymbol{z} = (z)_{rcq} \in \mathbb{R}^{R\times C\times Q}$, is achieved by computing the maximum values over non-overlapping regions of the input map $\boldsymbol{y}$ with a 2×2 square filter

$$z_{rcq} = \max_{m,\, n\in\{1,2\}} (y_{(2r+m)(2c+n)q}) \tag{3}$$

$q = 1, 2, \ldots, Q$. An illustration of the first max-pooling layer in PyroNet is shown in Fig. 7. Through each block (Conv1_*x*, Conv2_*x*, …, Conv5_*x*), the resolution of input becomes half of its original resolution (224, 112, 56, 28, and 14) due to max-pooling layers, while the dimension doubles until 512 (64, 128, 256, 512, and 512) as the number of filters for convolutional layers increase.

After passing the 5th block (Conv5_*x*), the features extracted for $\boldsymbol{H}_i$ form $z_i^{(5)} = [(z^{(5)})_{rcq}]_i \in \mathbb{R}^{7\times 7\times 512}$, which are flattened into a vector, $z_i^{(5)} \in \mathbb{R}^{1\times 25088}$, and then fed to two FC layers (equipped with ReLU activation function) with 4096 channels each

$$f_{ij}^{(1)} = \max\left(0,\ b^{(1)} + \sum_{k=1}^{25088} z_{ik}^{(5)} \cdot w_{jk}^{(1)}\right), \quad j = 1,\ 2, \ldots,\ 4096 \tag{4}$$

$$f_{ij}^{(2)} = \max\left(0,\ b^{(2)} + \sum_{k=1}^{4096} f_{ik}^{(1)} \cdot w_{jk}^{(2)}\right), \quad j = 1,\ 2, \ldots,\ 4096 \tag{5}$$

where $w_{jk}^{(l)}$ and $b^{(l)}$ are the weight and bias of FC layers, $l = 1, 2$. A 1D feature vector $\boldsymbol{f}_i^{(2)} \in \mathbb{R}^{1\times 4096}$ is thus obtained. To avoid overfitting, a drop-out layer is added after each of these two FC layers, whose rate is 0.5.

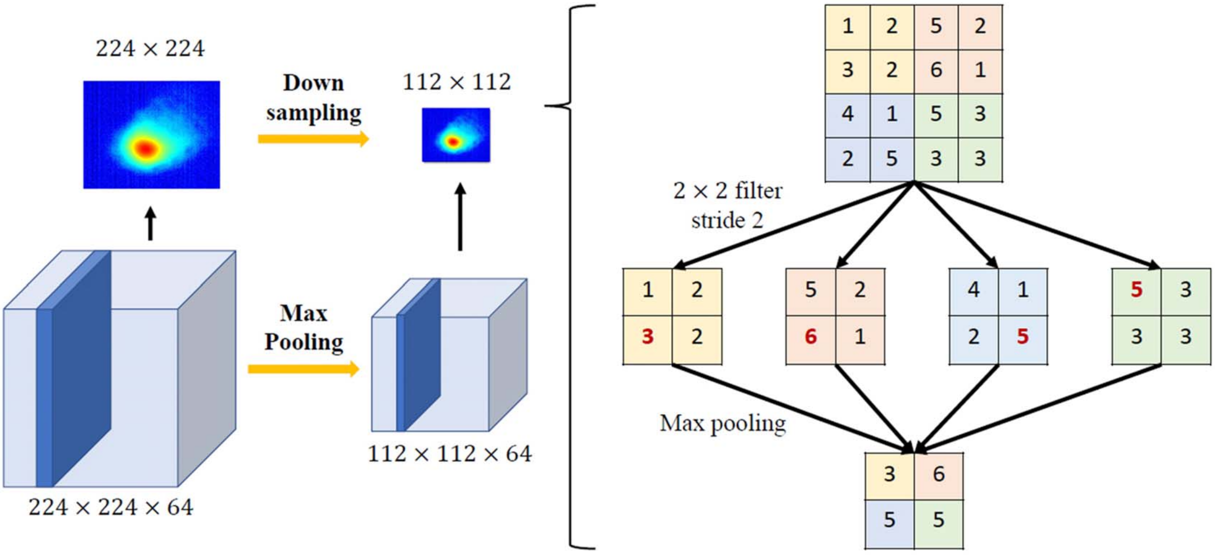

Fig. 7 Max-pooling layer illustration

After that, another FC layer which uses a two-way softmax function as the activation is to establish the relationship between extracted features $\boldsymbol{f}_i^{(2)}$ and binary porosity label (0 for "good" and 1 for "bad") in PyroNet. The probability of a sample to be "good," $\hat{p}_{pyro}(i)$, can be calculated by

$$\hat{p}_{pyro}(i) = \frac{e^{f_{i0}}}{e^{f_{i1}} + e^{f_{i0}}},$$
$$f_{ij} = b_j + \sum_{k=1}^{4096} f_{ik}^{(2)} \cdot w_{jk}, \ j = 0, \ 1 \tag{6}$$

where $\hat{p}_{pyro}(i)$ is the predicted probability of a pyrometer image showing neglectable porosity (diameter of <0.05 mm, predicted as "good"). w_{jk} and b_j are the weight and bias of the final FC layers. The predicted label of the pyrometer image will be "good" if $\hat{p}_{pyro}(i) > 0.5$.

We use the Keras package to build up the PyroNet. The pyrometer images are resized to 224 × 224 pixels (width × height) and then fed into the PyroNet. To train the PyroNet, *categorical cross-entropy* is used as the loss function

$$CE = -p_i \log \hat{p}_{pyro}(i) - (1 - p_i) \log (1 - \hat{p}_{pyro}(i)) \tag{7}$$

where p_i is the ground truth. The stochastic gradient descent (SGD) method is selected as the optimizer. We use classification accuracy as the performance metric in model training. A well-trained PyroNet will be used to predict the probability for the pyrometer images in the test set to be "good," $\hat{p}_{pyro}(i)$, and then used for the data fusion framework to calculate the final predicted probability of a sample to be "good," $\hat{p}(i)$.

3.3 LRCN Model for Infrared Image Data—IRNet.

We develop an IRNet for the IR image sequences. Since our IR data are several individual sequences where each sequence contains three time-continuous IR images, IRNet needs to model the spatial information of each IR image as well as the temporal information among the three images in the same sequence. To model sequential images, IRNet incorporates the idea of the LRCN model [30], which combines a CNN model to deal with image data and an RNN model (e.g., Long Short-Term Memory (LSTM) model) to handle temporal information.

A commonly used structure in the LRCN model is known as the FC-LSTM structure. It adopts several convolutional layers to extract features of images that have been added with temporal information through time-distributed layers first. After that, a flatten layer, a combination of FC layers, and drop-out layers will convert the parameters into a vector and decrease the number of parameters. Finally, an LSTM layer will model the temporal information and another FC layer is used to predict the classification label. This FC-LSTM structure is illustrated in Fig. 8.

The major disadvantage of FC-LSTM is the way to handle spatiotemporal data. Since it uses FC layers to finish transitions between input and state, as well as the ones between state and state, it is unable to encode spatial information [35]. Also, the LSTM layer can only deal with one-dimensional vectors; thus, the information has to be flattened to fit the input of the LSTM layer.

To cope with this disadvantage of FC-LSTM, our IRNet employs the ConvLSTM2D layer in the Keras package to build up the LRCN model, as shown in Fig. 9. The ConvLSTM2D layer is similar to an LSTM layer, while both of the recurrent transformations and input transformations are convolutional. That is to say, it can effectively deal with spatial information and temporal information simultaneously. The input of IRNet is IR sequences, each of which contains three resized 224 × 224 pixels (width × height) RGB IR images.

The input of IRNet is denoted as $\tilde{\boldsymbol{H}}_i$, where i is the index of the IR sequence and $i = 1, 2, \ldots$. Each of the sequences contains three

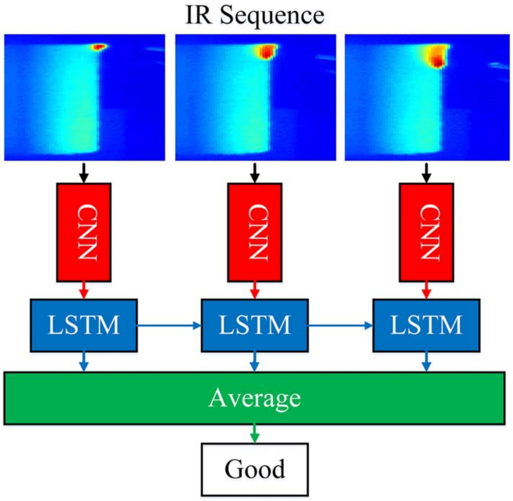

Fig. 8 FC-LSTM structure

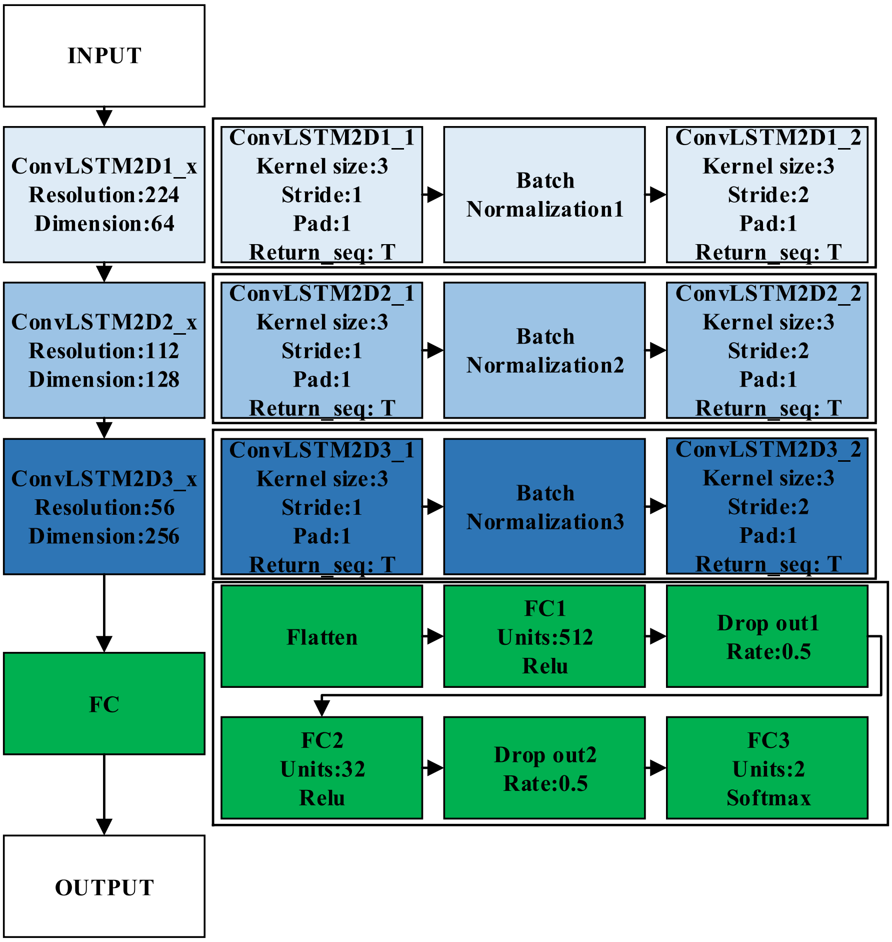

Fig. 9 Hyper-parameters in the IRNet

resized 224 × 224 pixels (width × height) RGB IR images, denoted as χ_1, χ_2, and χ_3 for each sequence. The input sequences will go through three ConvLSTM2D blocks. Each ConvLSTM2D block contains two ConvLSTM2D layers [37] with the same kernel size (set as 3), pad (set as 1), and return sequence (set as True to feedback all the information in each input sequence), whose equations are shown as follows:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \chi_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i)\\
f_t &= \sigma(W_{xf} * \chi_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f)\\
C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * \chi_t + W_{hc} * H_{t-1} + b_c)\\
o_t &= \sigma(W_{xo} * \chi_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o)\\
H_t &= o_t \circ \tanh(C_t)
\end{aligned}
\qquad (8)
$$

where χ_t is each image in the input sequence i, $t = 1, 2, 3$. i_t, f_t, o_t are input gate, forgot gate, output gate of the tth image in the sequence, respectively. C_t is the cell output, and H_t is the hidden state. σ is the logistic sigmoid function, "*" denotes the convolution operator, and "∘" means the Hadamard product. The weight tensor subscripts have the obvious meaning; for example, W_{hi} is the hidden-input gate tensor and W_{xo} is the input–output gate tensor, etc. Bias is also considered, which are denoted as b with certain subscripts.

Through these blocks, the resolution of the images in input sequences is halved twice, from 224 to 56, while the dimension is doubled twice, from 64 to 256, since we set a larger and larger filtering parameter for each block. The stride in the second ConvLSTM2D layers in three blocks is set as 2, which is different from the one in the first ConvLSTM2D layer in that block. By doing so, these ConvLSTM2D are equivalent to one ConvLSTM2D layer with stride 1 and another pooling layer. This arrangement decreases the resolution of the data while offering better performance than using the pooling layer. Also, between two ConvLSTM2D layers in each part, a Batch Normalization layer is used to transfer the mean of activation of the previous layer close to 0 and its standard deviation near 1. Other parameters of the ConvLSTM2D layers use their default values, such as the "tanh" activation, "hard_sigmoid" recurrent activation, and so on.

The rest of IRNet is quite similar to PyroNet: after going through the ConvLSTM2D blocks, the data will be flattened, and a combination of FC layers and drop-out layers will decrease the number of parameters and predict the possibility of IR sequences to be "good" or "bad" by another FC layer using two-way softmax as the activation function. *Categorical cross-entropy* is used as the loss function, SGD with Nesterov [38] momentum is used as the optimizer, and accuracy is used as the performance metric to train the IRNet. Note that the learning rate in SGD is adjusted according to the accuracy of the validation part during model training to avoid local optimum. A well-trained IRNet is then used to predict the probability to be "good" for the IR sequences in the test set, $\hat{p}_{IR}(i)$, and finally used for the data fusion framework to calculate $\hat{p}(i)$ according to Eq. (1).

4 Results and Analysis

Since 6-fold CV is performed to help prevent overfitting, we first present the detailed results and analysis for one of the folds (called Fold 1) in Secs. 4.1 to 4.3. Specifically, we will analyze the performance of PyroNet and compare them with existing studies in Sec. 4.1. The performance of IRNet will be presented in Sec. 4.2. The fused results will be given in Sec. 4.3. Finally, the performance of other folds (Fold 2 to Fold 6) will be listed in Sec. 4.4.

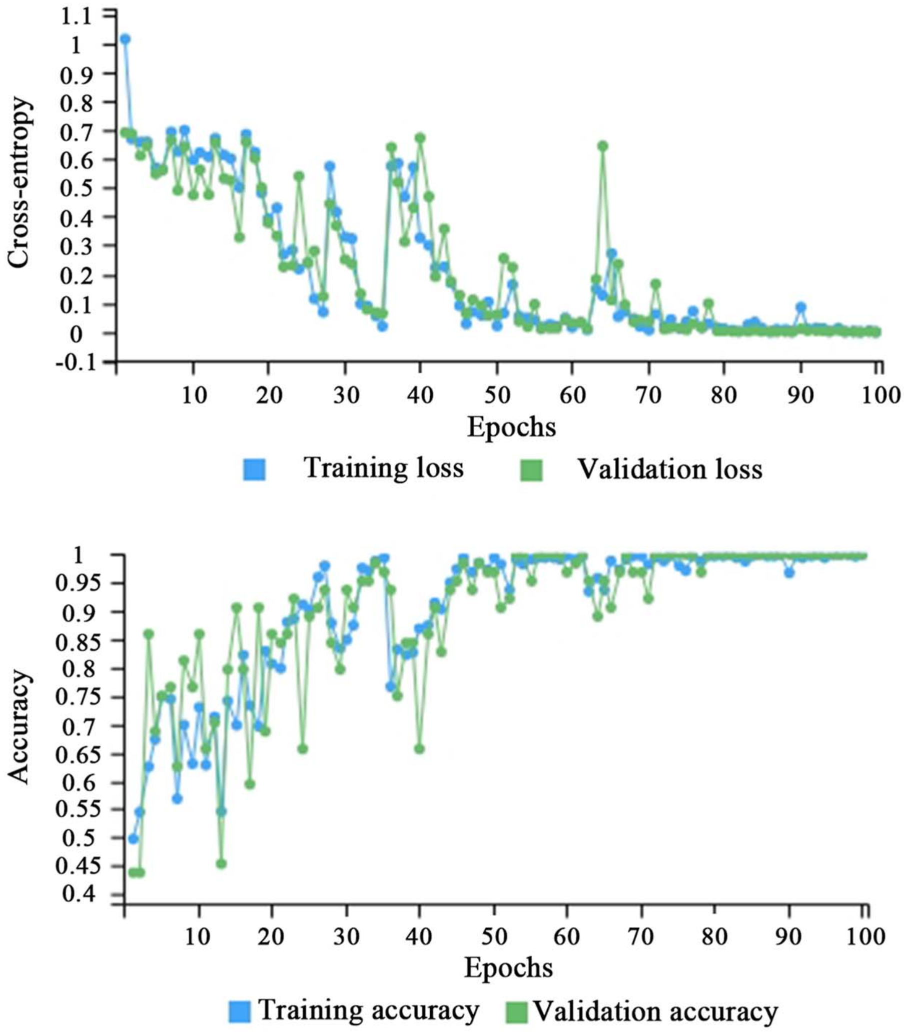

Fig. 10 Loss and accuracy during the training process of the PyroNet

4.1 PyroNet Results.

The PyroNet is trained by a training set, including 538 "good" and 538 "bad" pyrometer images to train the model and 107 "good" and 107 "bad" pyrometer images to validate the model. The batch size is 96, and the epoch number is 100. The loss and accuracy during the training process of PyroNet are presented in Fig. 10.

It can be seen from Fig. 10 that as the epoch number grows, both the training loss and validation loss decrease to nearly 0, while both the training accuracy and validation accuracy increase to almost 1, indicating that our PyroNet is well trained by the pyrometer dataset.

After training the PyroNet, we can use it to predict the porosity label for the test set (including 129 "good" and 11 "bad" pyrometer images). The confusion matrix of PyroNet is presented in Table 1.

In this study, we treat "bad" as positive and "good" as negative. Thus, True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) are 10, 0, 129, 1, respectively. Hence, performance indicators such as accuracy, recall, and precision can be determined by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100\% \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \tag{11}$$

Table 1 Confusion matrix of the PyroNet

	Predicted class	
Actual class	Bad	Good
Bad	10	1
Good	0	129

Table 2 Performance comparison with existing methods

Method	Accuracy (%)	Recall (%)	Precision (%)	False-positive rate (%)
T^2 chart with tensor decomposition by Khanzadeh et al. [10]	90.74	95.24	61.86	10.0
Q chart with tensor decomposition by Khanzadeh et al. [10]	88.19	25.40	80.00	1.08
K-Nearest Neighbor (KNN) by Khanzadeh et al. [14]	NA	98.44	NA	0.19
PyroNet	99.29	90.90	100	0

Note: NA in the table means unavailable.

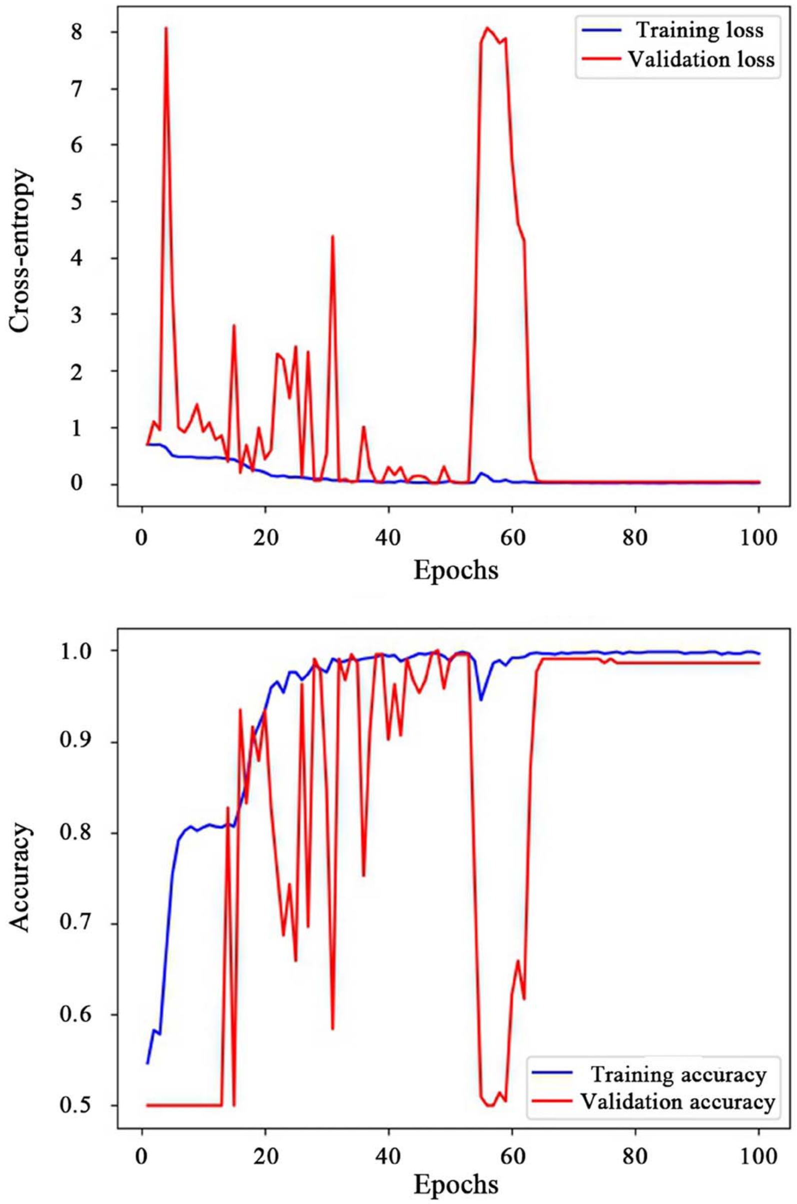

Fig. 11 Loss and accuracy during the training process of the IRNet

Table 3 Confusion matrix of the IRNet

	Predicted class	
Actual class	Bad	Good
Bad	5	6
Good	0	129

The performance indicators are compared with those from existing research that used the same data source. The comparison results are presented in Table 2.

Comparison results in Table 2 show that the proposed PyroNet yields higher accuracy, higher precision, and a lower false-positive rate than existing methods, suggesting the superiority of deep learning methods for porosity detection with melt pool images. We also notice that the recall value of our proposed method is not the best but still acceptable. The difference in the recall performance is partly due to the different number of true "bad" samples in the

Table 4 Estimated probability that the melt pool is "good"

Image index	PyroNet prediction $\hat{p}_{pyro}(i)$	IRNet prediction $\hat{p}_{IR}(i)$	Fused prediction $\hat{p}(i)$	True label
1	0.01	0	0.006	"bad"
3	0.02	0	0.012	"bad"
4	0.74	0.04	0.46	"bad"
5	0.02	0.09	0.048	"bad"
6	1	0.97	0.988	"good"
7	1	0.99	0.996	"good"
8	0.99	0.87	0.942	"good"
11	0.99	1	0.994	"good"
21	0	1	0.4	"bad"
41	0.08	1	0.448	"bad"
47	0	1	0.4	"bad"
62	0	1	0.4	"bad"
81	1	0.85	0.94	"good"
91	0	1	0.4	"bad"
122	0.01	1	0.406	"bad"

Table 5 Confusion matrix after data fusion

	Predicted class	
Actual class	Bad	Good
Bad	11	0
Good	0	129

test set. Khanzadeh et al. [10] had 63 true "bad" samples in the test set, while we have only 11. Thus, the small number of "bad" samples in our test set causes a relatively significant impact on our recall result. It is noticed that the data pre-processing in Khanzadeh et al. [10] differed from ours, but the comparison remains valid and fair since identical raw dataset from the same AM process underlaid the comparison.

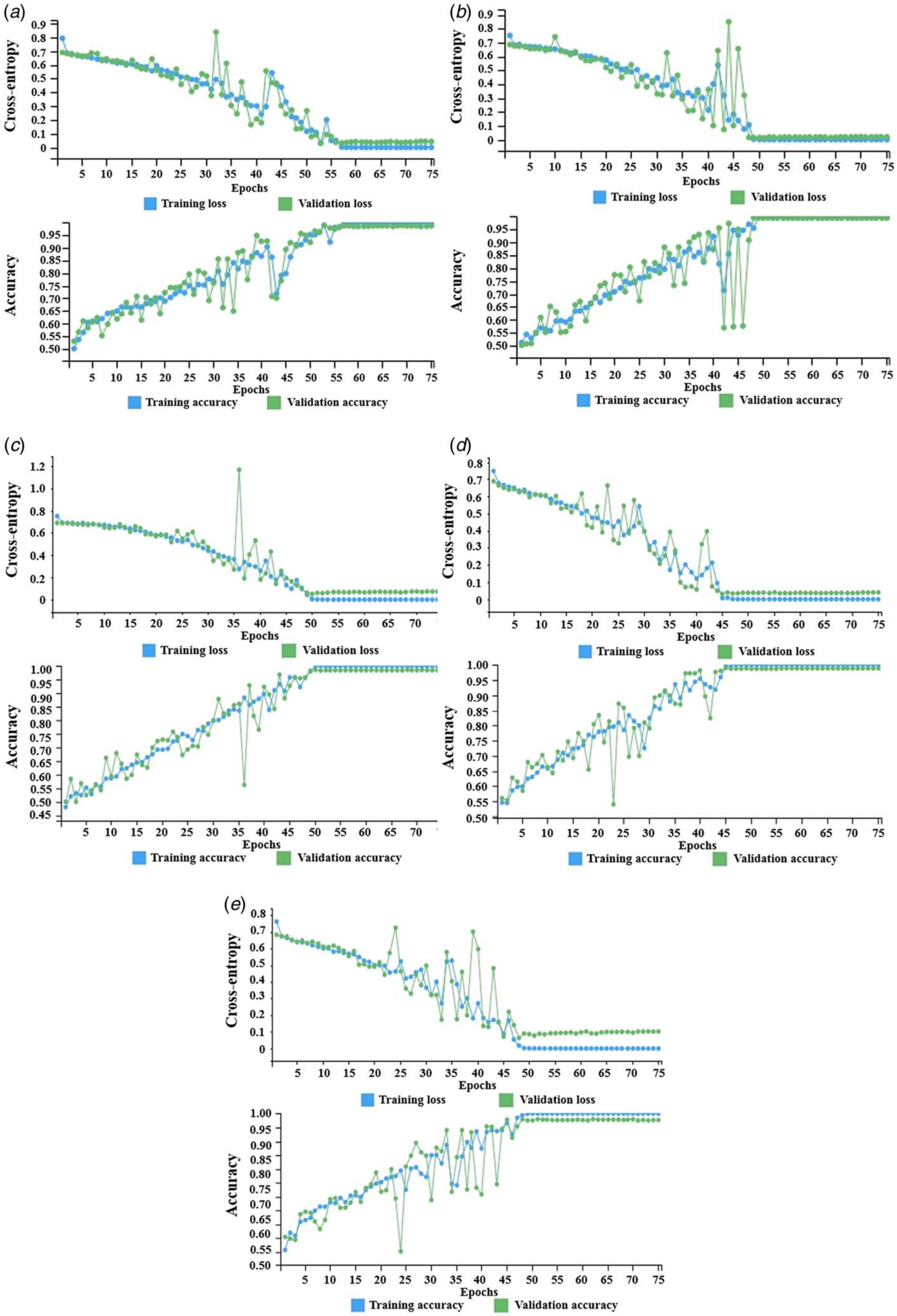

Fig. 12 Loss and accuracy during the training process of the PyroNet for Fold 2 to Fold 6: (*a*) fold 2, (*b*) fold 3, (*c*) fold 4, (*d*) fold 5, and (*e*) Fold 6

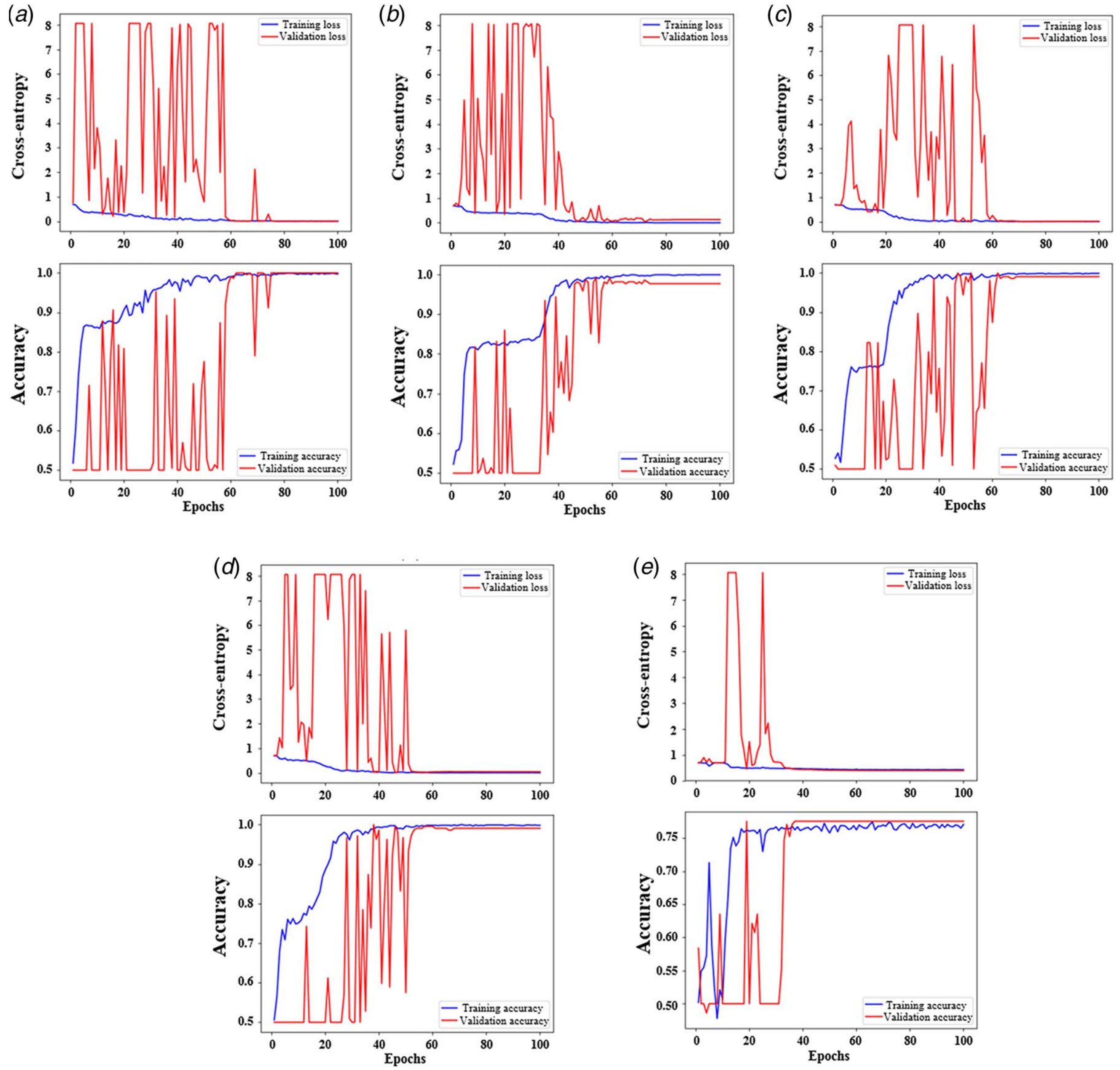

Fig. 13 Loss and accuracy during the training process of the IRNet for Fold 2 to Fold 6: (*a*) fold 2, (*b*) fold 3, (*c*) fold 4, (*d*) fold 5, and (*e*) fold 6

4.2 IRNet Results. The IRNet is trained by a training set, including 1076 IR sequences with half "good" and half "bad" to train the model, and 214 IR sequences with half "good" and half "bad" to validate the model. The batch size is 96, and the epoch number is 100. The loss and accuracy during the training process of IRNet are presented in Fig. 11.

From Fig. 11, it is clear that after some fluctuations, the training process finally converges, which means that the IRNet is well trained. Next, the well-trained IRNet is used to predict porosity label for IR sequences in the test set with 129 "good" and 11 "bad." The confusion matrix of IRNet is presented in Table 3.

From Table 3, it can be seen that the overall performance of the IRNet is not as good as the PyroNet. Thus, we assign a relatively higher weight to PyroNet results, believing the PyroNet to be more effective than the IRNet. A factor of $w = 0.6$ is suggested for the decision-level data fusion for porosity prediction.

4.3 Data Fusion Results and Analysis. By setting the weighting factor as 0.6, we combine the PyroNet predicted probability of a sample being "good" $\hat{p}_{pyro}$ and the IRNet predicted probability $\hat{p}_{IR}$ to calculate the $\hat{p}$ value for each sample in the test set. Most samples have a fused probability of 0 or 1, indicating high confidence in the prediction. The results of selected samples that have $\hat{p}$ between 0 and 1 are shown in Table 4.

From Table 4, it can be seen that the PyroNet prediction and IRNet prediction are not consistent for samples 4, 21, 41, 47, 62, 91, and 122. For most of them, such as samples 21, 41, 47, 62, 91, and 122, PyroNet prediction is accurate compared with the true label. However, for sample 4, it is misclassified by the PyroNet, whose true label is "bad" while the predicted probability to be "good" is $\hat{p}_{pyro}(4) = 0.74$. On the other hand, the IRNet of the 4th sample gives a very low predicted probability to be "good," $\hat{p}_{IR}(4) = 0.04$. It gives the fused predicted probability $\hat{p}(4) = 0.46$, which is less than 0.5, and so, the 4th sample will be predicted as "bad." The predicted labels of other samples after

Table 6 Performance of the PyroNet for all folds

Fold no.	TP	FP	TN	FN	Accuracy (%)
1	10	0	129	1	99.29
2	10	1	128	1	98.57
3	10	0	129	1	99.29
4	10	0	129	1	99.29
5	11	2	127	0	98.57
6	11	5	124	0	96.43

Table 7 Performance of the IRNet for all folds

Fold no.	TP	FP	TN	FN	Accuracy (%)
1	5	0	129	6	95.71
2	1	3	126	10	90.71
3	1	3	126	10	90.71
4	11	13	116	0	90.71
5	8	2	127	3	96.43
6	11	14	115	0	90.00

Table 8 Performance after data fusion for all folds

Fold no.	TP	FP	TN	FN	Accuracy (%)
1	11	0	129	0	100
2	10	1	129	0	99.29
3	10	0	129	1	99.29
4	10	0	129	1	99.29
5	11	1	128	1	99.29
6	11	5	124	0	96.43

data fusion are the same as the ones from PyroNet. By considering the information in IR data, we can achieve perfect results whose confusion matrix is shown in Table 5.

Moreover, both PyroNet and IRNet can predict the porosity condition of the 140 samples in the test set within 30 s. This indicates that the total time of detecting a sample in the LBAM process takes less than half a second (around 0.43 s, to be exact), which proves our proposed method is fast enough for in situ monitoring during the LBAM process.

4.4 Performance of Fold 2 to Fold 6. Similar to Fold 1, we also train the PyroNet and IRNet by using pyrometer data and IR data in the training set of Fold 2 to Fold 6, respectively. The training performance (i.e., training/validation loss and training/validation accuracy) of PyroNet and IRNet for Fold 2 to Fold 6 is shown in Figs. 12 and 13. From these figures, it is clear that the deep learning models were well trained. The training/validation accuracy converged to nearly 1 in all folds for pyrometer data and Folds 2, 3, 4, 5 for IR data; The training/validation loss converged to around 0 in all folds for pyrometer data and Folds 2, 3, 4, 5 for IR data.

Next, we use the trained PyroNet and IRNet on pyrometer data and IR data in test sets of different folds to obtain $\hat{p}_{pyro}$ and $\hat{p}_{IR}$, respectively. The performance including the confusion matrix and accuracy of PyroNet and IRNet for all folds is shown in Tables 6 and 7. From Table 6, it can be seen that the performance of PyroNet in all folds is quite good, and all the accuracy values of them are larger than 96%. Among them, Fold 6 gives relatively poor performance. Comparing with Table 6, the IRNet results in Table 7 are not as good as the ones of PyroNet but acceptable. The accuracy values of all folds are not less than 90%. In agreement with PyroNet, Fold 6 shows the worst performance. The good performance of Tables 6 and 7 proves that both of the PyroNet and IRNet are trained properly and can be used for the data fusion framework.

Finally, a factor of $w = 0.6$, which is the same value as we use in Sec. 4.2, is set as the weight of PyroNet while doing decision-level data fusion to get $\hat{p}$. The final porosity prediction results can be obtained, whose performance is shown in Table 8. From Table 8, we can see that our proposed data fusion framework shows close performance for all folds. Among them, Fold 1 shows the best performance, while Fold 6 shows the worst.

Furthermore, a comparison between Tables 6 and 8 suggests that considering potential features in IR data and incorporating IRNet results can improve the porosity prediction performance for Folds 1, 2, and 5, while the ones of other folds (Folds 3, 4, and 6) will not become worse, which shows the effectiveness of our proposed method.

5 Conclusion and Future Work

In this study, a deep learning-based decision-level data fusion method is proposed for in situ porosity detection during the LBAM process. The proposed method correlates melt pool thermal behavior, captured by pyrometer and infrared camera, with porosity. Specifically, a convolutional neural network called PyroNet is developed based on VGG16 to correlate pyrometer images with porosity; an IRNet is developed based on an LRCN to correlate IR image sequences with porosity. A data fusion framework is proposed to combine the predictions from PyroNet and IRNet to predict porosity.

To our knowledge, this is the first work that manages to fuse pyrometer data and IR camera data for metal AM. This study proves that although IR data were not used in previous studies, it has some implicit associations with porosity, thus can be used together with pyrometer data to help increase porosity prediction accuracy. We also prove that even though both of the pyrometer data and IR data are useful for porosity prediction, considering different collecting methods and data features, it would be better to set a higher weight for pyrometer data comparing with IR data since we have more confidence for pyrometer data. Moreover, despite a relatively slow training process of CNN model and LRCN model, once the deep learning model is well trained by training data, it can be used to predict porosity for new samples with high efficiency, making our method able to be used for in situ melt pool monitoring and detection of internal porosity during LBAM.

The main limitation of this study lies in the empirical selection of the weighting factor w and probability threshold p. In the future, we may consider designing optimization algorithms to select the optimal w and p automatically. Another interesting topic for future work is to incorporate data from additional sensors, LBAM process parameters, and physics information into our proposed data fusion framework to increase the accuracy and efficiency of porosity prediction. Besides, how to propose a suitable method to study the potential temporal patterns of pyrometer data may be an interesting topic in future studies.

Acknowledgment

The authors would like to thank the editor and anonymous referees for their insights, comments, and support. This work was partially supported by the Rutgers University Big Data Pilot Initiative Grant Award.

References

[1] Thompson, S. M., Bian, L., Shamsaei, N., and Yadollahi, A., 2015, "An Overview of Direct Laser Deposition for Additive Manufacturing; Part I: Transport Phenomena, Modeling and Diagnostics," Addit. Manuf., **8**, pp. 36–62.

[2] Yan, Z., Liu, W., Tang, Z., Liu, X., Zhang, N., Li, M., and Zhang, H., 2018, "Review on Thermal Analysis in Laser-Based Additive Manufacturing," Opt. Laser Technol., **106**, pp. 427–441.

[3] Heigel, J. C., Michaleris, P., and Reutzel, E. W., 2015, "Thermo-Mechanical Model Development and Validation of Directed Energy Deposition Additive Manufacturing of Ti–6al–4v," Addit. Manuf., **5**, pp. 9–19.

[4] Guo, Q., Zhao, C., Qu, M., Xiong, L., Escano, L. I., Mohammad, S., Hojjatzadeh, H., Parab, N. D., Fezzaa, K, Everhart, W., Sun, T., and Chen, L., 2019, "In-Situ Characterization and Quantification of Melt Pool Variation Under Constant Input Energy Density in Laser Powder Bed Fusion Additive Manufacturing Process," Addit. Manuf., **28**, pp. 600–609.

[5] Jafari-Marandi, R., Khanzadeh, M., Tian, W., Smith, B., and Bian, L., 2019, "From In-situ Monitoring Toward High-Throughput Process Control: Cost-Driven Decision-Making Framework for Laser-Based Additive Manufacturing," J. Manuf. Syst., **51**, pp. 29–41.

[6] Romano, J., Ladani, L., and Sadowski, M., 2015, "Thermal Modeling of Laser Based Additive Manufacturing Processes Within Common Materials," Procedia Manuf., **1**, pp. 238–250.

[7] Song, J., Wu, W. H., He, B. B., Ni, X. Q., Long, Q. L., Lu, L., Wang, T., Zhu, G. L., and Zhang, L., 2018, "Effect of Processing Parameters on the Size of Molten Pool in Gh3536 Alloy During Selective Laser Melting," IOP Conference Series: Materials Science and Engineering, Nanchang, China, May 25–27, p. 012090.

[8] Zhuang, J.-R., Lee, Y.-T., Hsieh, W.-H., and Yang, A.-S., 2018, "Determination of Melt Pool Dimensions Using Doe-Fem and Rsm With Process Window During Slm of Ti6al4v Powder," Opt. Laser Technol., **103**, pp. 59–76.

[9] Tian, H., Chen, X., Yan, Z., Zhi, X., Yang, Q., and Yuan, Z., 2019, "Finite-Element Simulation of Melt Pool Geometry and Dilution Ratio During Laser Cladding," Appl. Phys. A, **125**(7), p. 485.

[10] Khanzadeh, M., Tian, W., Yadollahi, A., Doude, H. R., Tschopp, M. A., and Bian, L., 2018, "Dual Process Monitoring of Metal-Based Additive Manufacturing Using Tensor Decomposition of Thermal Image Streams," Addit. Manuf., **23**, pp. 443–456.

[11] Mahmoudi, M., Ezzat, A. A., and Elwany, A., 2019, "Layerwise Anomaly Detection in Laser Powder-Bed Fusion Metal Additive Manufacturing," ASME J. Manuf. Sci. Eng., **141**(3), p. 031002.

[12] Seifi, S. H., Tian, W., Doude, H., Tschopp, M. A., and Bian, L., 2019, "Layer-Wise Modeling and Anomaly Detection for Laser-Based Additive Manufacturing," ASME J. Manuf. Sci. Eng., **141**(8), p. 081013.

[13] Khanzadeh, M., Chowdhury, S., Bian, L., and Tschopp, M. A., 2017, "A Methodology for Predicting Porosity From Thermal Imaging of Melt Pools in Additive Manufacturing Thin Wall Sections," ASME 2017 12th International Manufacturing Science and Engineering Conference Collocated With the JSME/ASME 2017 6th International Conference on Materials and Processing, Los Angeles, CA, June 4–8, p. V002T01A044.

[14] Khanzadeh, M., Chowdhury, S., Marufuzzaman, M., Tschopp, M. A., and Bian, L., 2018, "Porosity Prediction: Supervised-Learning of Thermal History for Direct Laser Deposition," J. Manuf. Syst., **47**, pp. 69–82.

[15] Khanzadeh, M., Chowdhury, S., Tschopp, M. A., Doude, H. R., Marufuzzaman, M., and Bian, L., 2019, "In-Situ Monitoring of Melt Pool Images for Porosity Prediction in Directed Energy Deposition Processes," IISE Trans., **51**(5), pp. 437–455.

[16] Scime, L., and Beuth, J., 2019, "Using Machine Learning to Identify in-Situ Melt Pool Signatures Indicative of Flaw Formation in a Laser Powder bed Fusion Additive Manufacturing Process," Addit. Manuf., **25**, pp. 151–165.

[17] Mitchell, J. A., Ivanoff, T. A., Dagel, D., Madison, J. D., and Jared, B., 2020, "Linking Pyrometry to Porosity in Additively Manufactured Metals," Addit. Manuf., **31**, p. 100946.

[18] Dey, N., Ashour, A. S., and Borra, S., 2017, *Classification in BioApps: Automation of Decision Making*, Springer, New York.

[19] LeCun, Y., Bengio, Y., and Hinton, G., 2015, "Deep Learning," Nature, **521**(7553), pp. 436–444.

[20] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B., 2012, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," IEEE Signal Process. Mag., **29**(6), pp. 82–97.

[21] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2017, "Imagenet Classification With Deep Convolutional Neural Networks," Commun. ACM, **60**(6), pp. 84–90.

[22] Weimer, D., Scholz-Reiter, B., and Shpitalni, M., 2016, "Design of Deep Convolutional Neural Network Architectures for Automated Feature Extraction in Industrial Inspection," CIRP Ann., **65**(1), pp. 417–420.

[23] Wang, P., Yan, R., and Gao, R. X., 2017, "Virtualization and Deep Recognition for System Fault Classification," J. Manuf. Syst., **44**(2), pp. 310–316.

[24] Wang, P., Gao, R. X., and Yan, R., 2017, "A Deep Learning-Based Approach to Material Removal Rate Prediction in Polishing," CIRP Ann., **66**(1), pp. 429–432.

[25] Wang, J., Ma, Y., Zhang, L., Gao, R. X., and Wu, D., 2018, "Deep Learning for Smart Manufacturing: Methods and Applications," J. Manuf. Syst., **48**(C), pp. 144–156.

[26] Williams, J., Dryburgh, P., Clare, A., Rao, P., and Samal, A., 2018, "Defect Detection and Monitoring in Metal Additive Manufactured Parts Through Deep Learning of Spatially Resolved Acoustic Spectroscopy Signals," Smart Sust. Manuf. Sys., **2**(1), pp. 204–226.

[27] Ye, D., Hong, G. S., Zhang, Y., Zhu, K., and Fuh, J. Y. H., 2018, "Defect Detection in Selective Laser Melting Technology by Acoustic Signals With Deep Belief Networks," Int. J. Adv. Manuf. Technol., **96**(5–8), pp. 2791–2801.

[28] Scime, L., and Beuth, J., 2018, "A Multi-scale Convolutional Neural Network for Autonomous Anomaly Detection and Classification in a Laser Powder Bed Fusion Additive Manufacturing Process," Addit. Manuf., **24**, pp. 273–286.

[29] Steed, C. A., Halsey, W., Dehoff, R., Yoder, S. L., Paquit, V., and Powers, S., 2017, "Falcon: Visual Analysis of Large, Irregularly Sampled, and Multivariate Time Series Data in Additive Manufacturing," Comput. Graph., **63**, pp. 50–64.

[30] Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T., 2015, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, June 7–12, pp. 2625–2634.

[31] Marshall, G. J., Thompson, S. M., and Shamsaei, N., 2016, "Data Indicating Temperature Response of Ti–6Al–4v Thin-Walled Structure During Its Additive Manufacture Via Laser Engineered Net Shaping," Data Brief, **7**, pp. 697–703.

[32] Faber, N. M., and Rajkó, R., 2007, "How to Avoid Over-Fitting in Multivariate Calibration—The Conventional Validation Approach and an Alternative," Anal. Chim. Acta, **595**(1–2), pp. 98–106.

[33] Rubinstein, R. Y., and Kroese, D. P., 2016, *Simulation and the Monte Carlo Method*, Vol. 10, John Wiley & Sons, Hoboken, NJ.

[34] Simonyan, K., and Zisserman, A., 2014, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv*:1409.15560.

[35] Sugata, T. L. I., and Yang, C. K., 2017, "Leaf App: Leaf Recognition With Deep Convolutional Neural Networks," Materials Science and Engineering Conference Series, Bali, Indonesia, Aug. 24–25, p. 012004.

[36] Romanuke, V. V., 2017, "Appropriate Number and Allocation of Relus in Convolutional Neural Networks," Наукові вісті Національного технічного університету України Київський політехнічний інститут, (1), pp. 69–78.

[37] Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c., 2015, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, Vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Inc., Red Hook, NY, pp. 802–810.

[38] Nesterov, Y., 1983, "A Method of Solving a Convex Programming Problem With Convergence Rate $O(1/k^2)$," Sov. Math. Doklady, **27**(2), pp. 372–376.