IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 17, 2019, accepted May 8, 2019, date of publication May 13, 2019, date of current version May 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2916461

Intelligent Localization of Transformer Internal Degradations Combining Deep Convolutional Neural Networks and Image Segmentation

JIAJUN DUAN, YIGANG HE, (Member IEEE), BOLUN DU, RUAA M. RASHAD GHANDOUR, WENJIE WU, AND HUI ZHANG

School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China

Corresponding author: Yigang He (yghe1221@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51577046, in part by the State Key Program of National Natural Science Foundation of China under Grant 51637004, in part by the National Key Research and Development Plan "Important Scientific Instruments and Equipment Development" under Grant 2016YFF0102200, and in part by the Equipment Research Project in Advance under Grant 41402040301.

ABSTRACT Industrial 4.0 placed higher demands on the field of intelligent equipment monitoring. The transformer is one of the critical power devices, its intelligent monitoring and fault positioning require in-depth studies. In this study, an efficient fault localization method for transformer internal thermal faults was proposed by introducing different deep convolutional neural networks (CNNs) and image segmentation. First, the transformer monitoring images of temperature and velocity fields in fault conditions were simulated using the lattice Boltzmann method (LBM), and the images were also used to highlight features information. In practice, transformer degradation does not frequently occur, so that the fault samples for deep learning are insufficient. To solve this problem, a transfer learning method was employed. Subsequently, fault locations were defined as classification labels, and different CNN's were used to classify the labels to achieve the fault localization results. Next, image segmentation was performed to extract the features of fault areas and simplify the data volumes. Likewise, the CNN's were employed to perform the fault localization again. Afterward, since the monitoring sensors were not located everywhere in a transformer in practical applications, information of partial monitoring areas where the monitoring sensors located was trained following a similar procedure. After image segmentation, the average fault localization accuracy using the information obtained by sensors decreased from 97.95% to 94.42%, while the data volume was reduced to nearly 1% of the original one. Besides, the average calculation time per iteration decreased by 8.816%, while the loss value was reduced by 37.68%. Finally, the Friedman hypothesis test and Nemenyi post hoc test were performed to compare the evaluation indicators of different networks, and the performance of GoogLeNet in this case was considered the best.

INDEX TERMS Condition monitoring, fault diagnostics, convolutional neural networks (CNNs), image segmentation, lattice Boltzmann method (LBM), level set, transformers, Friedman test.

I. INTRODUCTION

In the era of Industrial 4.0, more intelligent fault identification and localization methods need in-depth exploration. Transformer are critical for power systems [1]. According to surveys by IEEE and CIGRE, the annual failure rate of oil-immersed transformers reaches 0.625%, and it increases by 1-2% for large power transformers (above 300kv) [2]. Discussing how to monitor and diagnose transformer faults can help ensure the safety and reliability of the energy system.

The analysis of transformer temperature during monitoring process has long been a hot topic [3]. Several computation and analysis methods have been commonly used, e.g. equivalent thermal circuit model [4], Computational Fluid Dynamics (CFD) [5], and Finite Element Method (FEM) [6], to simulate transformer conditions. Wang *et al.* [7] employed Lattice Boltzmann Method (LBM) to solve the transformer's heat transfer problem. Recently, a promising LBM-based

The associate editor coordinating the review of this manuscript and approving it for publication was José Valente De Oliveira.

numerical simulating method has been developed for LBM's simplicity, computational efficiency, and high scalability in parallel processing [8]. Using LBM, simulation images of oil temperature and velocity distribution of the transformer can be taken. Because LBM is fast, it can better satisfy the requirements of real-time intelligent monitoring systems.

In the field of monitoring internal faults of transformers, there are a vast variety of methods [9], [10], e.g. Dissolved Gas Analysis (DGA) [11], Partial Discharge (PD) [12], Frequency Response Analysis (FRA) [13], Infrared Thermography (IRT) [14]. Fault monitoring through temperature information is not affected by electromagnetic interference. Besides, the sensors can record the changes of information in the monitoring areas. However, existing studies rarely considered the situations when faults occur, and few studies have been conducted on fault identification or localization. The conventional temperature monitoring methods are based on mathematical model and focus on hotspot [15], [16]. By reasonable deductions, it is likely to detect internal faults of the devices based on external monitored information [17]. This method is effective, whereas it requires specific simulation and monitoring systems for different power devices. When impurities (e.g. moisture, machinery components, and metals) appear in transformer oil, the method's insulation performance will be reduced, and the previously derived monitoring standards will be no longer applicable. With the continuous development of artificial intelligence, the deep features of temperature field image data can be extracted.

The continuous advancement of equipment manufacturing technologies and fault monitoring methods may lead to insufficient fault samples. Fault diagnosis accuracies are limited by the lack of fault samples, which is obvious when using deep Convolutional Neural Networks (CNNs). Transfer learning can transfer the knowledge in the *B* field to the *A* field so that it can be adopted to compensate the insufficient fault samples with the simulation data. In addition, less training data, using pre-trained CNNs, can still achieve a relatively high accuracy [18]. Through transfer learning, appropriate transition between different samples can be achieved [19]. Wang *et al.* [20] made improvements in a CNN by learning from the corresponding images published online to detect vehicle monitoring data. Kolar *et al.* [21] used a distributed training method and transfer learning in the field of architecture. They, by training synthetic images, achieved higher accuracies of the validation process using real construction photos. Reference [22] applied deep learning approach to IRT monitoring videos and then determined the condition of the machine automatically. The required data set for training different CNNs can be supplemented with LBM simulation results.

The monitoring images used for deep neural network training process should be processed following feature extraction procedures using image processing methods. Because these images may cover redundant information, and the excessive variation details of temperature may occupy too much data capacity, [23] and [24] studied feature extraction methods of temperature images for rotating machinery. Lee *et al.* [25], by modifying the neural network architecture, enhanced the temperature image quality. Existing fault feature extraction algorithm can correspond to a certain device. They require manual participations during analysis process, which is difficult to perform during intelligent detection. Reference [26] studied a Level Set Method (LSM) for image segmentation to achieve automatic segmentations, which is faster and no longer limited by initial contour settings. The processed monitoring images highlight fault features. Moreover, since CNNs' are capable of distinguishing latent features automatically [27], it can be used to achieve intelligent fault diagnosis and localization [28]. Moreover, different CNNs have been used in a range of monitoring fields [29]–[32], most of which are focused on how to extract degradation features of equipment. Intelligent fault localization methods are more reliable compared with conventional ones since they are not required to calculate the specific features, e.g. hotspot and the relevant threshold for each transformer.

The major contributions of this study are summarized as follows:

1) This study proposed a transformer fault localization method based on deep learning (different CNNs) and image processing algorithms (LBM-LSM). There is hardly any reference about the fault localization problems of transformer internal faults at present.

2) Transfer learning method was used to solve the problems of insufficient monitoring data. LBM simulation images with different fault labels were obtained and then used as the primary source of data to train different CNNs. Besides, the structures of pre-trained networks were modified to fit the diagnosing situations in this study. This method could reduce the amount of data set.

3) The practical monitoring environment limits the size of data, and the monitoring areas usually cannot cover the whole transformer. Given these problems, this paper used LBM-LSM image segmentation method to reduce the data volumes and extract information of partial areas.

4) This study compared fault localization results of different CNNs and conventional classification method. The overall evaluation indices were proposed to measure the effectiveness of each CNNs.

The rest of this paper is organized as follows:

In Section II the whole framework of the proposed method is presented, and procedures of LBM simulation, the image processing algorithm and the fault localization method through CNNs are introduced. In Section III, experiment processes are illustrated. In Section IV, further experiments considering the constrained monitoring areas are presented. In Section V, calculation results are analyzed. In Section VI, a conclusion of this study is drawn.

II. METHODS AND PRE-PROCESSING MATERIALS

The proposed transformer fault localization method combined LBM computation and feature extraction techniques, using image segmentation and deep CNNs. Fig. 1 shows

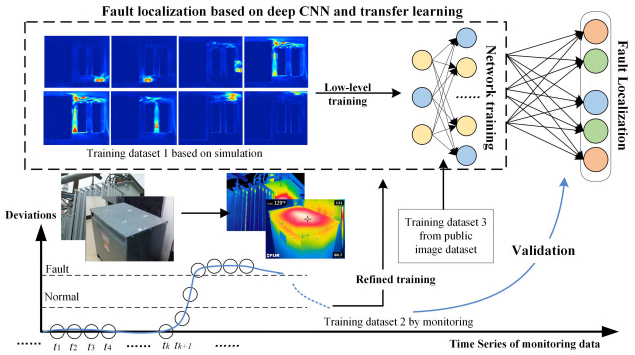

FIGURE 1. Structure of intelligent monitoring method for transformer faults using different CNNs.

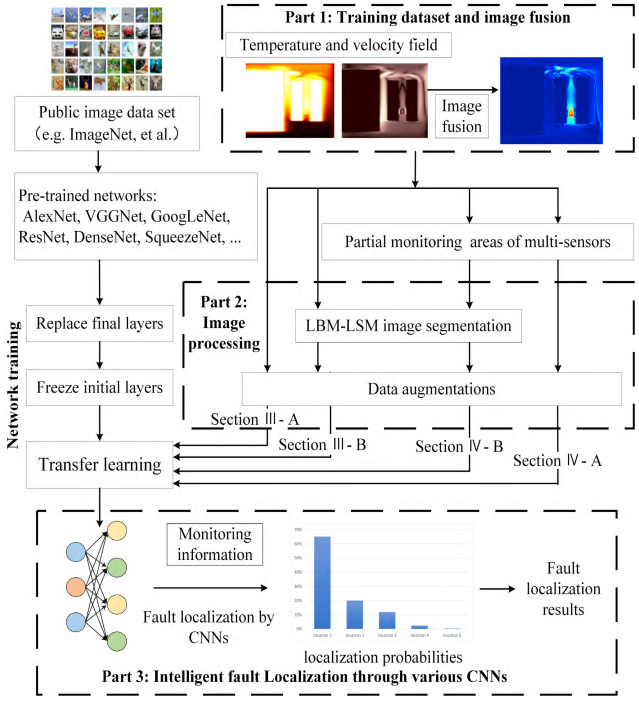

FIGURE 2. Framework of the proposed fault localization method and the structure of this paper.

the overall structure of the proposed transformer monitoring method. Given the shortage of fault data samples in practical applications, transfer learning was adopted. During the monitoring process, almost all data were in normal condition and could be easily identified through iterations in the time series. Once degradation occurred, the monitored information would change. Using the network trained beforehand, preliminary fault localization results could be obtained. Flow chart of the proposed fault localization method and the structure of this paper are shown in Fig. 2, which is split into 3 parts. The four arrows pointing at the third part in Fig. 2 represent one section from III-A to IV-B respectively.

A. LBM SIMULATION METHOD FOR TRANSFORMER MONITORING

When running oil-immersed transformers, the core as a heat source caused natural convection or natural circulation. The temperature of the transformer near the heat source rose, whereas the density decreased. Besides, when the heated oil was close to the heat sink, its heat could be dissipated naturally. Accordingly, the oil temperature had a close relation to the distribution of the velocity field. Besides, the failures (e.g. moisture, mechanical or metallic impurities generated in the oil) had certain effects on the conditions of the oil. By modifying the heat dissipation boundary conditions, the faults could be simulated.

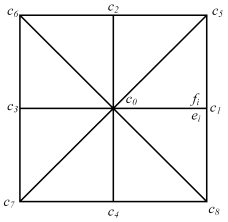

FIGURE 3. D2Q9 structure for LBM.

Using LBM simulation method, temperature distribution field of transformers could be instantly obtained, thereby promoting the acquisition of massive data for deep CNNs training. Temperature and velocity distribution were closely correlated, and their combination can improve the accuracy of fault localization. In this study, temperature and velocity fields were based on standard D2Q9 LBM model, as shown in Fig. 3. This model has been commonly used in solving fluid flow problems for its more vectors than D2Q4 and D2Q5 models [33]. Equilibrium distribution function for velocity field is expressed as [34]:

$$f_i^{eq} = \omega_i \rho \left[1 + \frac{3(c_i \cdot u)}{c^2} + \frac{9(c_i \cdot u)^2}{2c^4} - \frac{3u^2}{2c^2} \right] \tag{1}$$

where f_i denotes the particle distribution functions; c_i is discrete particle velocity vector; $c = \Delta x/\Delta t$, Δx is lattice space; Δt is the lattice time step size;u and ρ are the velocity and density used to calculate the equilibrium distribution function f_i^{eq} respectively. Moreover, $\omega_i$is the weights of the equilibrium distribution functions in the Boltzmann model. He and Luo [35] considered the moments of equilibrium distribution functions as Guass-type integrals, and they derived the weights ω_i:

$$\omega_{\mathrm{i}} = \begin{cases} 4/9, & i = 0 \\ 1/9, & i = 1, 2, 3, 4 \\ 1/36, & i = 5, 6, 7, 8 \end{cases} \tag{2}$$

The discrete particle velocity c_i is expressed as:

$$c_i = \begin{cases} (0, 0) & i = 0 \\ (\cos\theta_i, \sin\theta_i)c, \\ \theta_i = \dfrac{(i-1)}{2}\pi & i = 1, 2, 3, 4 \\ \sqrt{2}(\cos\theta_i, \sin\theta_i)c, \\ \theta_i = \dfrac{(i-5)}{2}\pi + \dfrac{\pi}{4} & i = 5, 6, 7, 8 \end{cases} \tag{3}$$

The macroscopic viscosity ν is calculated by:

$$\nu = \left(\tau - \frac{1}{2}\right) c_s^2 \Delta t \tag{4}$$

where c_s denotes the lattice velocity of sound in corresponding medias, which is equal to $c/\sqrt{3}$[3].

Given that the fluid was driven by buoyancy, an external force should be considered. According to Boussinesq Approximation, the external force is expressed as:

$$F = \rho g \beta \Delta T \tag{5}$$

where ρ denotes the density of transformer oil; g is the gravitational vector; ΔT is the temperature difference between hot and cold boundaries; β is the thermal expansion coefficient. Subsequently, discrete Lattice Boltzmann Equation (LBE) is written as:

$$f_i(\mathrm{x} + c_i \Delta t, t + \Delta t) - f_i(\mathrm{x}, t) = -\frac{f_i(\mathrm{x}, t) - f_i^{eq}(\mathrm{x}, t)}{\tau_f} + \Delta t F_i \tag{6}$$

where: $F_i = \frac{\omega_i}{c_s^2} F \cdot c_i$.

Similar to the velocity field, LBE of temperature field can be expressed as

$$g_i(\mathrm{x} + \mathrm{c}_i \Delta t, t + \Delta t) - g_i(\mathrm{x}, t) = -\frac{g_i(\mathrm{x}, t) - g_i^{eq}(\mathrm{x}, t)}{\tau_g} \tag{7}$$

Equilibrium distribution functions for temperature field are written as

$$g_i^{eq} = \omega_i \rho T \left[1 + \frac{3(c_i u)}{c^2} + \frac{9(c_i u)^2}{2c^4} - \frac{3u^2}{2c^2}\right] \tag{8}$$

After mesoscopic computation, the mesoscopic variables should be reverted to macroscopic variables. According to the model, the macroscopic density ρ, macroscopic speed ρ_u and the internal energy ρ_T can be respectively calculated by:

$$\rho = \sum_{i=0}^{8} f_i = \sum_{i=0}^{8} f_i^{eq} \tag{9}$$

$$\rho_u = \sum_{i=0}^{8} c_i f_i = \sum_{i=0}^{8} c_i f_i^{eq} \tag{10}$$

$$\rho_T = \sum_{i=0}^{8} g_i = \sum_{i=0}^{8} g_i^{eq} \tag{11}$$

Moreover, the winding loss P_T was affected by temperature, which is expressed as:

$$P_T = P_0[1 + k_t(T - T_0)] \tag{12}$$

where P_0 denotes the winding loss at the temperature T_0; $k_t$is the temperature coefficient of the conductor; T is the real-time temperature. The heat flux density q_i of the heat source is determined by the loss of windings and iron core, which is expressed as:

$$q_i = \frac{P}{A_i} \tag{13}$$

where q_i denotes the heat flux density in a meter square of heating surface; P is the total loss; A_i is the heat transfer area.

The heat transfer coefficient h of the transformer tank is calculated by:

$$h = \frac{Nu \cdot \lambda}{H} \tag{14}$$

where Nu denotes the Nusselt number; λ is the thermal conductivity of the medium; H is the characteristic length.

TABLE 1. Simulation algorithm of transformer monitoring images.

Algorithm 1: Transformer temperature and velocity field simulation
Step 1) Initialization
Calculate parameters and initialize boundary conditions;
Generate abnormal area Ω;
Step 2) Evolution:
for i=1, …, N **do**
Step 2.1) Velocity field: Calculate the equilibrium distribution function (1);
Set bounce boundaries or symmetrical boundaries according to geometric size;
Set space Ω as bounce boundaries and of high temperature;
Obtain macroscopic speed ρ_u;
Update current velocity vector u.
Step 2.2) Temperature field: Similar to step 2.1, calculate equilibrium function g_i^{eq} ; Update temperature parameter t.
Step 2.3) Updating and visualization: Update images and display;
end for
Output:
Transformer velocity field (Fig_{VM}) and temperature field (Fig_{TM}).

LBM simulation was performed to obtain different images of specific fault locations. The calculation procedure is presented in Table 1. Input variables for transformer LBM simulation cover physical parameters of transformer oil, e.g. dynamic viscosity (μ), heat capacity at constant pressure (Cp), density (ρ) and thermal conductivity (k_t); parameters of heat sources include thermal conductivities and relevant losses in iron core, primary windings and secondary windings. Furthermore, it also includes geometric parameters of transformer.

B. LBM-LSM IMAGE SEGMENTATION METHOD

1) ENERGY EQUATION

Fuzzy c-means (FCM) clustering algorithm can split an image into c different classes. Assuming that the image contains N pixels, and the value of each pixel is $X = \{x_i\}$, where $i = 1, 2, \ldots, N$. The aim of the calculation process is to optimize the following clustering criteria [26]:

$$J(U, V, X) = \sum_{k=1}^{c} \sum_{i=1}^{N} u_{ki}^{p} \|x_i - v_k\|^2$$
$$s.t. \sum_{k=1}^{c} u_{ki} = 1, \quad 0 \le u_{ki} \le 1, \ \forall k, i \tag{15}$$

where U denotes a partition matrix, where u_{ki} represents the membership that the i-th element belongs to the k-th class. Likewise, V is a centroid vector, where v_k means the centroid of the k-th class. The parameter $p$is a fuzzy index, representing the weighting exponent of each fuzzy membership. The norm operator $\|\cdot\|$ indicates the standard Euclidean distance.

The minimum of objective function J was obtained when high degree memberships were assigned to the pixels with intensities close to the centroid of its particular class, while low membership values were assigned to the pixels with intensities far from the centroid.

If the monitoring images were recorded as $Y = \{y_i\}_{i=1}^N$, the initial image domain selected at the beginning of the image processing was considered bias field image $B = \{\beta_i\}_{i=1}^N$, and equation (15) can be written as a continuous form:

$$J(U, V, B, Y) = \sum_{k=1}^{c} \int_{\Omega_k} U_{ki}^p(x,y) \|Y(x,y) - B(x,y) - v_k\|^2 dxdy \quad (16)$$

Monitoring images should be classified into several types. Considering the convenience when the monitored information was being transmitted, the normal part of monitoring images was marked as black, and the abnormal part was white, *i.e.* $c = 2$. Thus, the optimization function in (16) should be divided into two parts: Ω_1 and Ω_2, and the level set function is expressed as follows:

$$\begin{aligned} &J(U, V, B, Y) \\ &= \int_{\Omega} U_1^p(x,y) \|Y(x,y) - B(x,y) - v_1\|^2 H(\phi) dxdy \\ &\quad + \int_{\Omega} U_2^p(x,y) \|Y(x,y) - B(x,y) - v_2\|^2 (1 - H(\phi)) dxdy \\ &\quad s.t.\ U_1(x,y) + U_2(x,y) = 1,\ 0 \le U_k(x,y) \le 1,\ \forall x, y \end{aligned} \quad (17)$$

where ϕ denotes the signed distance function; $H(\ )$ is the Heaviside function. Level set function J was used as a data link in energy function, shown in (18):

$$E(U, V, B, Y, \phi) = J(U, V, B, Y, \phi) + \nu \int_{\Omega} |\nabla H(\phi)| dxdy \quad (18)$$

where ν is a constant and $\nu > 0$. The latter integral term denotes the length of a given curve which is represented implicitly as the zero level of ϕ.

2) LEVEL SET EQUATION

LBM can be used to solve the parabolic diffusion equation recovered by Chapman–Enskog expansion with the external force, including [26]:

$$\frac{\partial \rho}{\partial t} = \gamma \nabla \cdot \nabla \rho + F \quad (19)$$

The classical energy function optimization method aims to find a stable solution of gradient descent for fixed U, V and B: $\frac{\partial \phi}{\partial t} = -\frac{\partial E}{\partial \phi}$. Substituting it to equation (19), the following level set equation (LSE) could be obtained:

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \delta(\phi) \left(U_1^p(x,y) \|Y(x,y) - B(x,y) - v_1\|^2 \right. \\ &\quad \left. - U_2^p(x,y) \|Y(x,y) - B(x,y) - v_2\|^2\right) \\ &\quad + \nu \delta(\phi) \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right) \\ &\quad s.t.\ U_1(x,y) + U_2(x,y) = 1,\ \ \forall x, y \\ &\quad 0 \le U_k(x,y) \le 1,\ \ \forall k, x, y \end{aligned} \quad (20)$$

To solve the optimization equation, the first derivative of corresponding variables (e.g. u_{ki}, v_k and β_i) should be calculated and set to zero. Accordingly, the following supplementary conditions are available:

$$U_k^*(x,y) = \frac{1}{\sum_{l=1}^{c} \left(\frac{\|Y(x,y) - B(x,y) - v_k\|}{\|Y(x,y) - B(x,y) - v_l\|}\right)^{\frac{2}{(p-1)}}} \quad (21)$$

$$v_k^* = \frac{\int_{\Omega} U_k^p(x,y) (Y(x,y) - B(x,y)) dxdy}{\int_{\Omega} U_k^p(x,y) dxdy} \quad (22)$$

$$B^*(x,y) = Y(x,y) - \frac{\sum_{k=1}^{c} U_k^p(x,y) v_k}{\sum_{k=1}^{c} U_k^p(x,y)} \quad (23)$$

Finally, LSE was added to LBM (19). The ρ in equation (19) was replaced by the signed distance function Φ. After the replacement, it was suggested that (20) is only a variation formula of (19). Thus, the external force term could be obtained by comparison:

$$\begin{aligned} F = \lambda \Big(&U_1^p(x,y) \|Y(x,y) - B(x,y) - v_1\|^2 \\ &- U_2^p(x,y) \|Y(x,y) - B(x,y) - v_2\|^2\Big) \end{aligned} \quad (24)$$

where λ denotes a positive parameter and the parameter $\rho = 2$.

This section presents an image processing method. First, monitoring images should be fused with the relevant information of normal conditions, so that the fault features were extracted. Second, the image segmentation algorithm based on LBM-LSM was used to obtain the contours of fault locations. In the meantime, fault areas were visually displayed on monitoring images. The algorithmic description of image processing is given in Table 2.

Corresponding processing results are shown in Fig. 4. The rows represent segmentation results, and the columns represent different fault locations. When an initial contour was drawn outside the monitoring area, e.g. on the transformer core shown in Fig. 4 (a1), (b1) and (e1), it might always exist. However, this part of the segmentation results did not affect the monitored areas. The last line was the boundaries of the fault areas obtained by LBM-LSM calculation. Fig. 4 shows

TABLE 2. Image fusion and LBM-LSM image segmentation.

Algorithm 2: Image Fusion and Feature Extraction Algorithm Based on LBM-LSM

Input:
Images of Transformer oil velocity field (Fig_{VM}) and temperature field (Fig_{TM});
Step 1) Initialization
Crop the image to match a uniform size;
Step 2) Image Fusion:
Apply image fusion process by (26).
Step 4) Fault image segmentation:
Initialize distant function Φ, set Φ'=Φ-min(Φ);
Initialize centroid values v_1, v_2 and bias field image B;
Calculate $U_1^p(x,y)$, $U_2^p(x,y)$, v_1, v_2, B, external force F by (21)-(24);
Compute discrete LBM equation f_i by (6); Accumulate it;
Update distance value
Output:
fault state information matrix (Fig_D) after fusion processing.

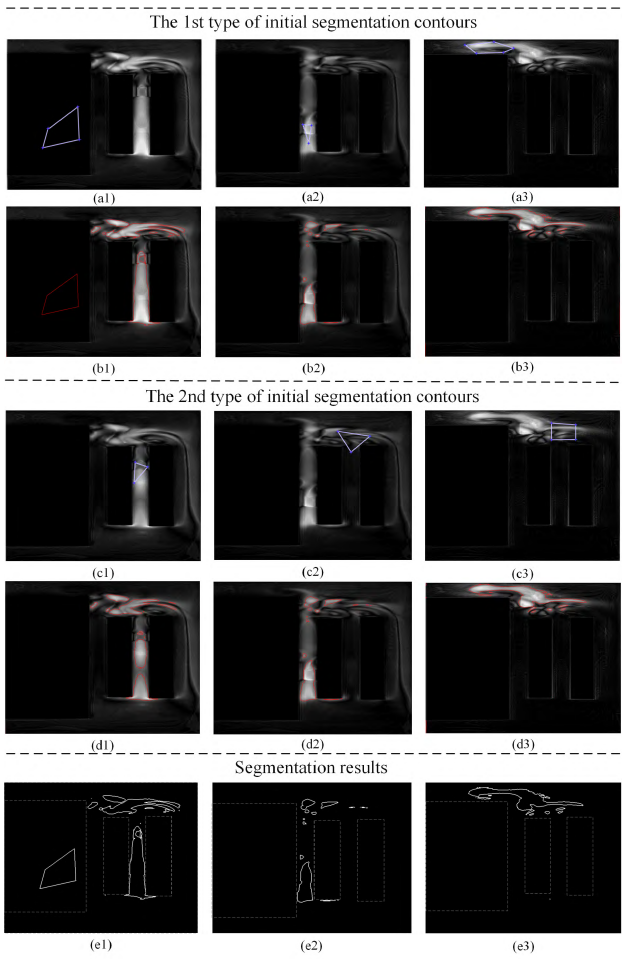

FIGURE 4. Samples of image segmentation results for transformer fault areas.

that, segmentation results were absolutely identical wherever initial contour was.

C. DEEP LEARNING FAULT LOCALIZATION METHOD

Because CNNs have abilities for deep feature extraction, adaptive fault diagnosis can be achieved.

TABLE 3. Dataset of transformer degradation images for training.

Fault location	Original image number	Gray-scale	Rotate and mirror	crop	Adding noises	total for training (80%)
Right	25	25	125	75	25	220
Left	25	25	125	75	25	220
Middle	25	25	125	75	25	220
Upper-Left	15	15	75	45	15	132
Upper-Right	15	15	75	45	15	132
Bottom-Left	15	15	75	45	15	132
Bottom-Right	15	15	75	45	15	132
Normal	50	50	250	150	50	440
Total	185	185	925	555	185	1628

1) DATA SET

Fault diagnosis process required a corresponding data set for training and evaluation. As shown in Fig. 5, fault locations were simulated randomly and classified into different labels. The image was easy to classify after feature extraction process. Intelligent localization algorithms based on deep CNNs were employed to classify fault locations, thereby achieving the adaptive positioning. In total, 240 images were used as original images to be classified into different tags, including different locations of transformer oil degradations. 7 types of fault locations and a normal condition were studied here. Fig. 5 shows the relevant positions of the seven fault location tags: top left, top right, left, center, right, bottom left, and bottom right so that the fault locations of the transformer can be separated into different labels. The rules for the division of transformer areas can be adjusted according to specific applications. The image data set is shown in Fig. 5, including colorful or gray-scale processed ones. Furthermore, more image augmentation methods are to be present in the next section.

2) AUGMENTATION

Training process of CNNs require a considerable amount of monitoring samples. Though the simulation data set and the transfer learning method can make up the shortage of monitoring images that should be obtained under fault conditions, it is necessary to further expand the dataset and to avoid overfitting during training process. After obtaining the processed images by image feature fusion, the data set should be augmented. Using augmentation methods: randomly rotating 90°, 180° or 270°, mirroring, adding noise, etc., other monitoring image versions were obtained. In total, the transformer monitoring image dataset contained 2035 images, 1628 of which (80%) were for training and 407 (20%) for testing. The monitoring image data set of the transformer applied for training different CNNs are listed in Table 3. Besides, to standardize the image size, it is necessary to maintain the consistency of image feature extraction process and meet the deep CNNs requirements.

D. BRIEF INTRODUCTION TO CONVOLUTIONAL NEURAL NETWORKS

AlexNet [36] achieved a top-5 error rate of 15.3% in the 2012 ILSVR (ImageNet Large Scale Visual Recognition

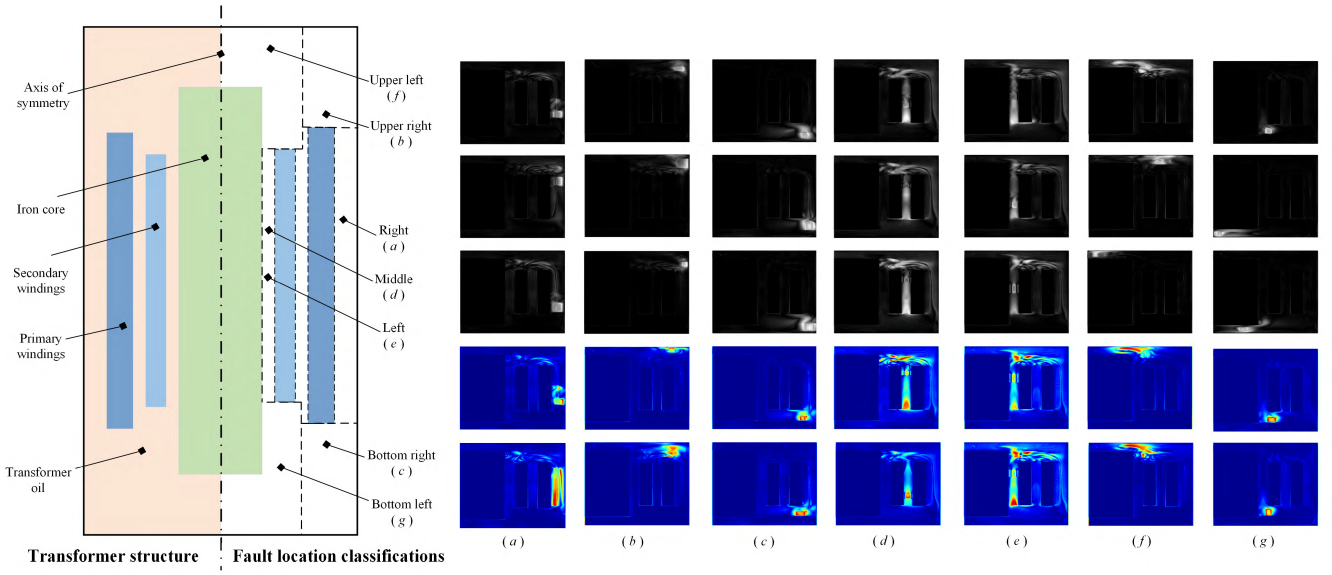

FIGURE 5. Structure and fault locations of the transformer. The pictures on the right side are examples of fault data corresponding to different labels. Their fault locations are: (a) the right, (b) the upper right, (c) the bottom right, (d) the middle, (e) the left, (f) the upper left, and (g) the bottom left areas of the transformer.

Competition), much higher than that of the second one (26.2%). Since then, growing number of complex and better performance convolutional neural networks emerged.

VGGNet [37] emphasizes the depth of the network. It rigorously used a 3×3 filter (stride=1, pad=1) and a 2×2 maximum pooling layer (stride=2). Thus, a large effective receptive field was achieved by the combination of 2 small filters.

GoogLeNet [38] proposed the concept of ''Inception Module'', breaking the tradition of connecting CNNs layer by layer. Each module includes multiple parallel convolutional layers with a size of 1×1, 3×3, 5×5 and a max pooling layer for the extraction of different features simultaneously. Besides, the 1×1 convolutional layer was used to reduce the dimensions before the 3×3 and 5×5 convolutional layers, increasing the depth of the network and reducing the network parameters.

ResNet [39] proposed a residual network which uses the difference between output and input ($H(x)$-x) for optimization training. By introducing ''shortcut'' module and identity connections, ResNet reduced the problem of gradient disappearance in deep neural networks.

Inspired by the ''identity connections'', DenseNet [40] has a dense connection structure, which means each layer obtains the outputs of all the layers before it. This structure further improved the problem of gradient disappearance. DenseNet reduced the amount of computation by promoting feature propagation process and facilitating feature reuse.

Light networks (e.g. SqueezeNet [41]) significantly reduced the amounts of parameters (1/50 of AlexNet) while maintaining a relatively high accuracy. It proposed a new network architecture ''Fire Module''. 1×1 convolution was used to compress the dimension of feature maps and reduce the weight parameters.

TABLE 4. Basic parameters of the diagnosing networks used in this study.

Network	Depth	Size (Mb)	Parameters (Millions)
GoogLeNet	22	27	7
VGG16	16	515	138
VGG19	19	535	144
SqueezeNet	18	4.6	1.24
AlexNet	8	227	61
DenseNet	201	77	20
ResNet18	18	44	11.7
ResNet50	50	96	25.6
ResNet101	101	167	44.6

Moreover, R-CNN (Rigion-CNN) models (e.g. R-CNN, fast R-CNN, Faster R-CNN, and Mask R-CNN [42]) gradually appeared to achieve both image detection and target classification. The deep CNNs used for fault localization in this study and their fundamental parameters are listed in Table 4. Paper [43] gives a detailed comparison analysis of these networks.

III. SIMULATION PROCESS

This section falls into two parts. First, fault simulation image and image fusion algorithm were demonstrated, and fault localization was performed using different pre-trained deep CNNs. Second, fault areas of transformer monitoring images were outlined by LBM-LSM image segmentation method. A further classification and localization process using deep CNNs were demonstrated. The data set used for fault localization are listed in Table 3. After image augmentation process, the number of pictures used for training the CNNs was 1628, and another 407 pictures were for validation process.

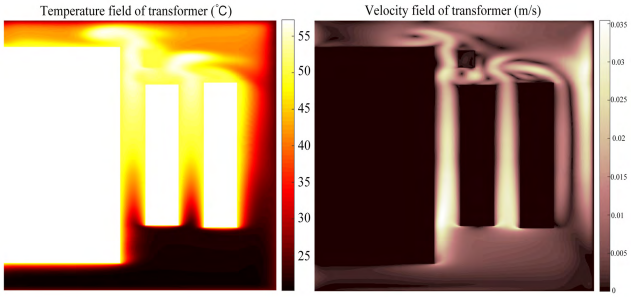

FIGURE 6. This is the simulation results of transformer while degradation appears at the upper left area.

A. FAULT LOCALIZATION BASED ON TRANSFORMER TEMPERATURE AND VELOCITY FIELD

1) IMAGE FUSION AND FEATURE EXTRACTION PROCESS

Implementing the program in Table 1 until meeting the convergence condition in (25), one group of fault simulation results are shown in Fig. 6. The left image above shows the temperature field of transformer and the right demonstrates the velocity field. The convergence criteria during fault simulation could be defined by computing the temperature increment during each simulation step, and the computation would continue until it became smaller than a pre-defined fixed value ε, as shown in the following Equation (25):

$$\sum_{x} \frac{|T(x,t) - T(x,t-1)|}{|T(x,t)|} < \varepsilon = 10^{-7} \tag{25}$$

where x denotes any coordinate within the transformer.

To facilitate the computation process and fault diagnosis abilities, the monitoring information and normal state datasets were merged, so that the overall monitoring deviation images could be obtained. The image fusion process for one result sample is illustrated in Fig. 7.

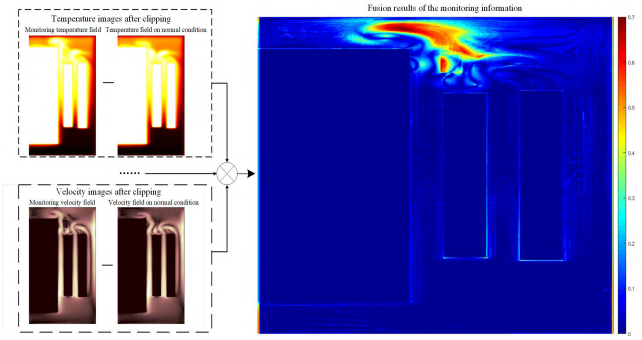

FIGURE 7. An example of Image fusion method.

Denoting the monitoring image of transformer temperature field by Fig_{TM}, and the transformer temperature field image during normal operating condition by Fig_{TN}, respectively. Their deviation value is expressed as $Fig_{TD} = Fig_{TM}\text{-}Fig_{TN}$; Likewise, the monitoring image of transformer velocity field is symbolized by Fig_{VM}, and the transformer velocity field image during normal operating condition is Fig_{VN}, therefore, their deviation value is expressed as $Fig_{VD} = Fig_{VM}\text{-}Fig_{VN}$.

The overall deviation image Fig_D can be defined via Equation (26), which is essentially a matrix of the corresponding pixel values:

$$Fig_D = \frac{1}{2}\left(\frac{Fig_{TD}}{\max(Fig_{TD})} + \frac{Fig_{VD}}{\max(Fig_{VD})}\right) \tag{26}$$

where the temperature and velocity deviation values are divided by their maximum values: max (Fig_{TD}), and (Fig_{VD}) respectively, which presents the normalization procedure. In such a way, different monitoring data were proportionally merged.

2) FAULT LOCALIZATION BY DEEP LEARNING

In this section, we applied different pre-trained deep CNNs to train the overall deviation images and to achieve fault identification and localization automatically. These networks have been trained on more than 1 million images via ImageNet database. Therefore, a small number of training images can be used to quickly transfer the learned functions to new tasks. Besides, the first 10 layers of the networks were frozen during the training process so as to further reduce the parameters need to be fine-tuned.

Replace the last layers with learnable weights for AlexNet, VGGNet, GoogLeNet, ResNet, DenseNet, and SqueezeNet and use them as the training networks respectively for intelligent fault localization diagnosis. In most networks, this layer is a fully connected layer. Replace this fully connected layer with a new fully connected layer with the number of outputs equal to the number of categories in diagnosing data set (8, in this study); For some networks (e.g. SqueezeNet), the last learnable layer is a 1-by-1 convolutional layer. In this case, replace the convolutional layer with a new convolutional layer with the number of filters equal to the number of diagnosing categories.

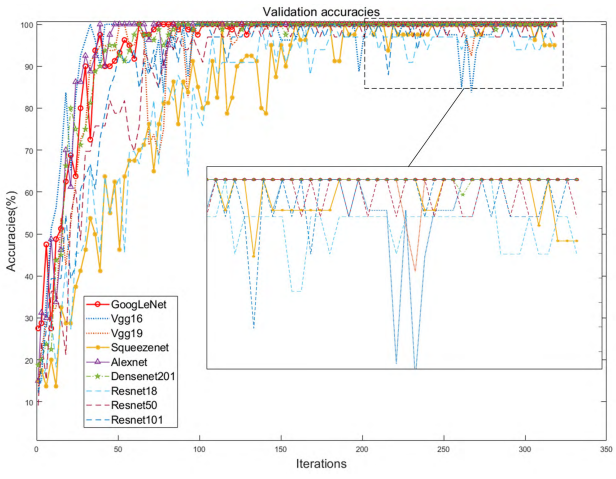

FIGURE 8. Validation results of transformer fault locations based on various CNNs.

20% of the deviation images were randomly select as the testing data set. The validation process of fault diagnosing methods based on different CNNs are shown in Fig. 8.

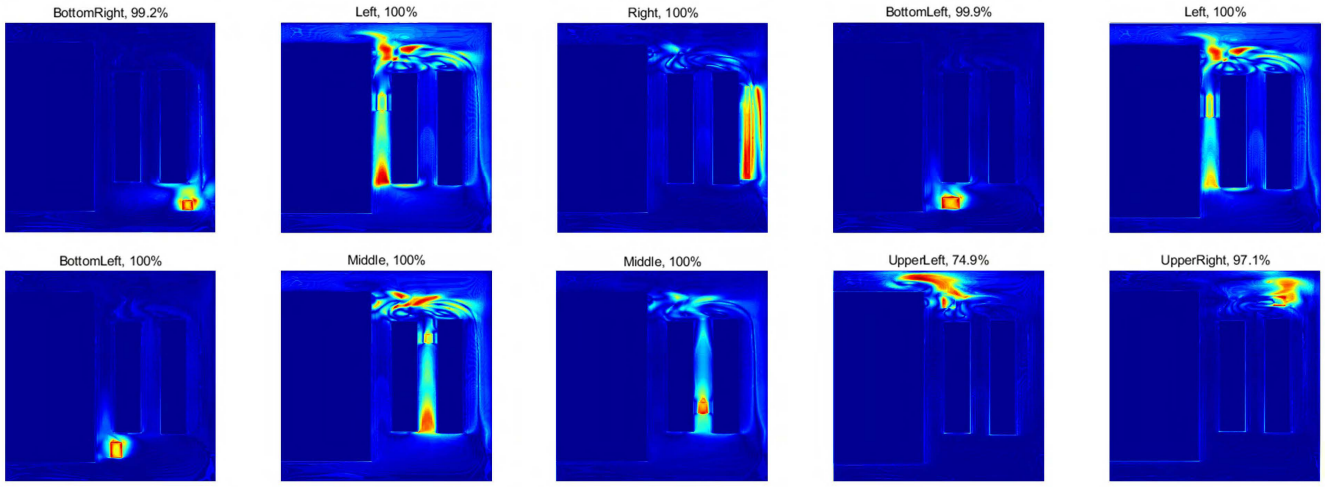

FIGURE 9. 10 Randomly selected samples of Fault localization results. The title of each picture displays the localization category with the highest probability.

TABLE 5. Parameter settings of the networks and computer configuration information.

Hyper parameters of the CNNs		Hardware/Software parameters	
Name	Parameters	Name	Parameters
Solver type	SGD	Memory	8Gb
Momentum	0.9	Processor	Intel Core i5-7400 @ 3.00GHz
Initial learning rate	0.001		
Learning rate drop factor	0.2	Graphics	NVIDIA GeForce GT 730
Batch size	60	Operating system	Windows 10 64 bits
Epochs	10		

TABLE 6. Validation results based on different CNNs.

Name	Accuracy (%)	Loss	Time (s/iteration)	Iterations [a]
GoogLeNet	100	0.0002358	5.415	60
VGG16	100	2.275e-06	23.72	39
VGG19	100	5.335e-06	29.33	96
SqueeZenet	96.25	0.338024	2.275	162
AlexNet	100	1.729e-06	2.02	42
DenseNet	100	0.0029065	27.68	81
ResNet18	97.8	0.057208	3.18	135
ResNet50	100	0.0315973	8.97	117
ResNet101	100	0.0293898	16	84
Average values:	99.339	0.051	13.177	90.667

[a]Iterations indicate the number of iterations when verification accuracies reach the maximum value for the first time.

And in Table 5, hyper parameters were summarized in the left side, and the hardware or software parameters are listed in the right side.

To visually display the fault types and the corresponding probabilities, original dataset (without image augmentation) was used to verify the localization results. The localization results using GoogLeNet are shown in Fig. 9. And the fault diagnosing results of other networks were similar to this network. 10 monitoring images were randomly selected from the verification data set. For each overall deviation image, the label with the highest probability among its fault locations and the corresponding probability value are displayed, as shown in Fig. 9. This Figure suggests that the randomly selected samples could be correctly classified as the accurate locations.

Different neural networks were trained on the fault sample data set and verified using the verification data set. Besides, the verification results: the accuracies, the loss value, the calculation time required for each iteration, and the number of iterations when verification accuracy reaches the maximum value for the first time, are listed in Table 6.

B. FAULT LOCALIZATION AFTER IMAGE SEGMENTATIONS

The internal structure of a transformer is very complex. There are redundant changes of temperature and velocity field distributions, which is inconvenient for data storage and transmitting process. Data reduction and compression processing are necessary. In this section, image edge information is extracted using LSM-LBM method before different CNNs training processes.

The image segmentation method based on LBM-LSM was employed to process the overall deviation images obtained by (26), so that the contours of fault areas were outlined. Several examples of this method are shown in Fig. 4. Deteriorate areas outlined by edge segmentation method were set to 1, which were illustrated by white lines, and the rest areas were set to 0. Assuming that the size of monitoring area comprises $m \times n$ pixels, whose value varies from 0 to 255, and a color image consists of three-dimensional data. After segmentation process, each pixel was 0 or 1, and the dimension of data was reduced from three dimensions to two. Besides, the outline of abnormal areas only took up a small fraction of the whole image. For example, the outline in Fig. 4 (d1) contained 779 pixels, and the size of this image was 224×224. Thus, the number of pixels for original image reduced from $m \times n \times 3 = 224 \times 224 \times 3 = 150528$ to 779. During the data storage or transmitting process, each pixel of the original

TABLE 7. Validation results after the image segmentation.

Name	Accuracy (%)	Loss	Time (s/iteration)	Iterations [a]
GoogLeNet	99.8	0.0002514	3.58	69
VGG16	99.5	0.000156	18.99	52
VGG19	99.5	0.0006877	28.72	54
SqueezeNet	96.1	0.0140718	1.68	225
AlexNet	99.5	0.0085508	1.57	66
DenseNet	99.8	0.0014125	25.35	62
ResNet18	98.6	0.0075096	3.55	144
ResNet50	99	0.0139129	9.41	112
ResNet101	98.9	0.0803395	17.01	135
Average values:	98.967	0.014	12.207	102.111

[a]Iterations indicate the number of iterations when verification accuracies reach the maximum value for the first time.

image required 8-bit (1B) binary data. After outlining fault areas, only the data whose value is 1 should occupy a data space. Also, according to the image size, 8-bit (1B) coordinates were necessary for x and y axis, respectively. Therefore, the total amount of binary data required for original image was $150528 \times 1\text{B} = 147\text{kB}$, and $779 \times 2\text{B} = 1.52\text{kB}$ after outlining respectively, which was approximately only 1% of the original amount.

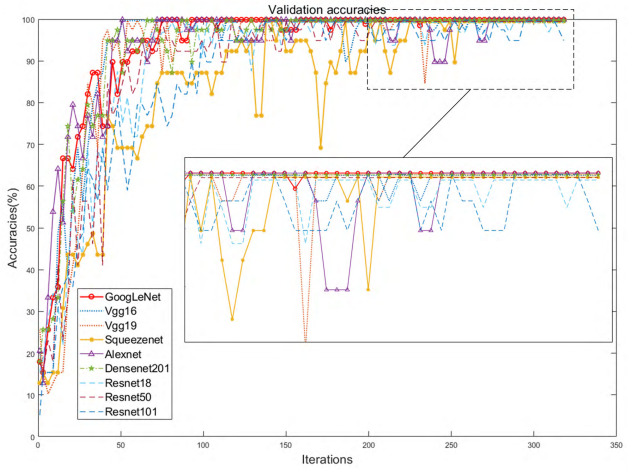

FIGURE 10. Validation results after image segmentations based on various CNNs.

The overall deviation images after segmentation were served as input for different deep CNNs, and the validation process of fault localizations are shown in Fig. 10. After extracting the outlines of fault areas by image segmentation algorithm, data volume can be significantly reduced. Likewise, the verification results are shown in Table 7.

IV. FAULT LOCALIZATION BY MONITORING INFORMATION

A. BEFORE IMAGE SEGMENTATIONS

Considerable monitoring devices and plenty of data are needed if it is necessary to monitor images of the entire transformer. However, in practical applications, monitoring images are usually obtained through only a few sensors,

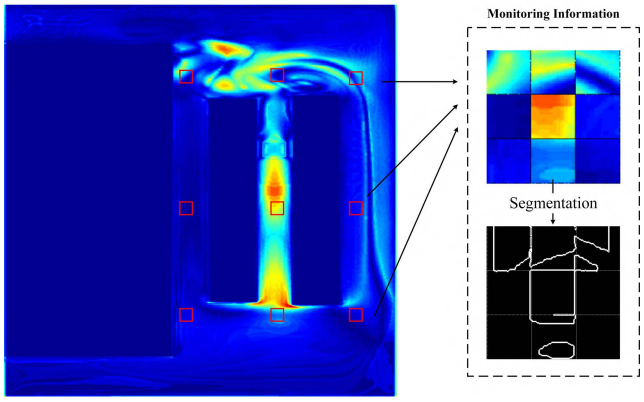

FIGURE 11. Sensors placed on several parts near the transformer windings obtain the monitoring information. And the right side shows segmentation results.

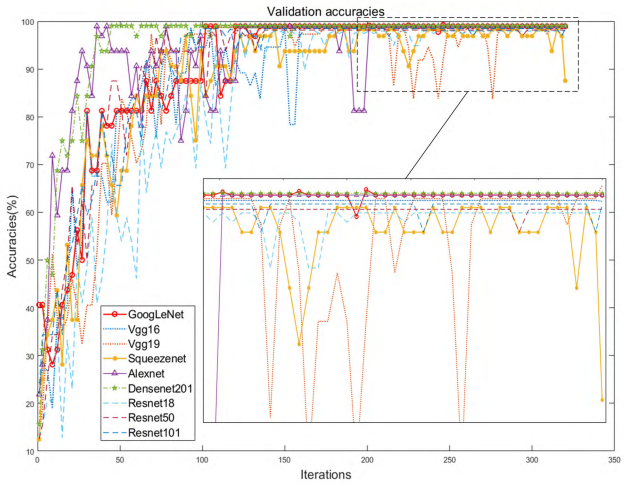

FIGURE 12. Validation results of the 9 sensors' information based on various CNNs.

which are located in partial areas of the whole transformer. In this section, 9 small blocks near the windings were considered information of different detecting sensors. Positions of the monitoring areas are shown in the left picture of Fig. 11. Subsequently, these areas were put together as shown in the right side of the figure. To facilitate calculation and comparative analysis, data set in this section were obtained through the data set listed in Table 3. They were trained under different convolutional neural networks in Table 8 to locate fault areas similar to section III. There were 1628 images for training, and 407 for validation.

Different CNNs were employed to classify fault locations with the information obtained by monitoring sensors. The validation process are shown in Fig. 12. Besides the validation results: the accuracies, the loss value, the calculation time per iteration, and the number of iterations when verification accuracy reaches the maximum value for the first time, are listed in Table 8.

Furthermore, 10 monitoring images were randomly selected from verification data set when using GoogLeNet as the localization network, as shown in Fig. 13. Their most

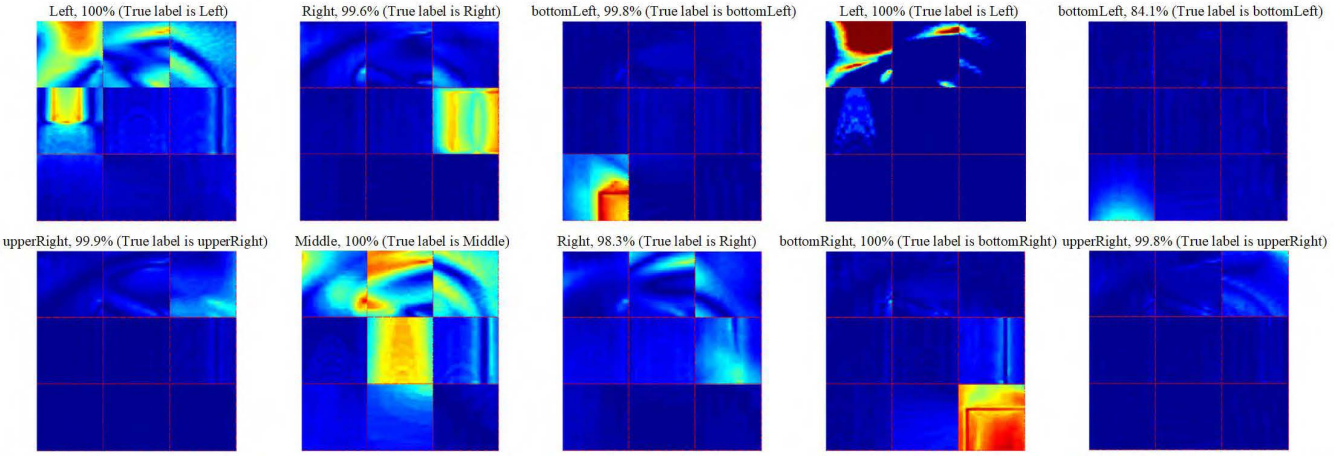

FIGURE 13. 10 Randomly selected samples of fault localization results based on several monitoring sensors (9 sensors). The title of each picture displays the localization category with the highest probability.

TABLE 8. Validation results based on monitoring sensor information before the image segmentation.

Name	Accuracy (%)	Loss	Time (s/iteration)	Iterations [a]
GoogLeNet	99.3	0.0070199	3.48	120
VGG16	99	0.1612007	18.81	129
VGG19	97.6	0.0931014	22.5	96
SqueezeNet	96.6	0.2803907	1.57	180
AlexNet	97.9	0.0008381	1.49	120
DenseNet	99.1	0.0130996	18.18	48
ResNet18	97.9	0.0429953	3.293	85
ResNet50	98.4	0.0091334	9.093	105
ResNet101	98.6	0.0163253	16.32	96
Average values:	98.267	0.069	10.526	108.778

[a]Iterations indicate the number of iterations when verification accuracies reach the maximum value for the first time.

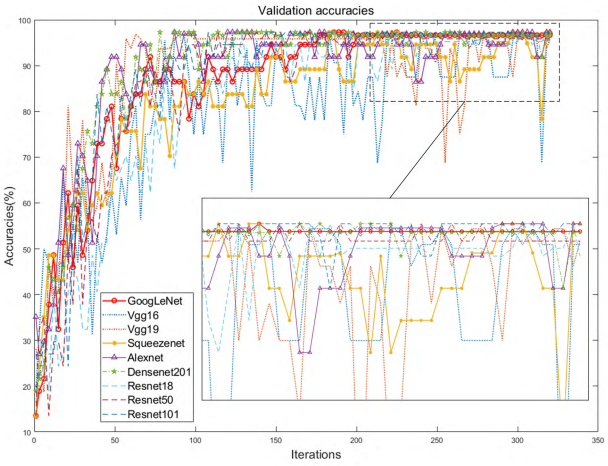

FIGURE 14. Various CNNs' validation results of the 9 sensors' information after image segmentations.

likely positions are illustrated respectively. Because the real labels of monitoring images could not be observed directly in this situation, they are represented in parentheses.

B. AFTER IMAGE SEGMENTATIONS

Similar to section III-B, edges of monitoring areas acquired by several sensors were extracted by LBM-LSM image segmentation method. And the same training procedures were applied for fault localization. Validation process through different CNNs in this situation is shown in Fig. 14. And the fault localization results are listed in Table 9. To visually display the fault localization results, 10 randomly selected validation samples are shown in Fig. 15 likewise.

Comparing the results of Table 6 and Table 7, or Table 8 and Table 9, fault localization accuracies in the latter situation were almost lower than those of the former. However, the iteration time of the latter was shorter than that of the former, and the amount of data was significantly reduced, which would promote data storage and transmitting process.

In addition, the conventional Support Vector Machine (SVM) method was employed here to conduct the fault diagnosing situations. Its average fault localization accuracy was 81.86%, much lower than that of the CNNs. It can be seen that the fault localization results through SVM model is

TABLE 9. Validation results based on monitoring sensor information after the image segmentation.

Name	Accuracy (%)	Loss	Time (s/iteration)	Iterations [a]
GoogLeNet	97.2	0.0215996	2.85	108
VGG16	93.5	0.0084823	23.72	150
VGG19	95.1	0.0012583	19.02	168
SqueezeNet	95	0.1702075	1.31	286
AlexNet	97.5	0.0485637	1.28	96
DenseNet201	96.9	0.0148976	14.73	114
ResNet18	95.2	0.0339539	2.69	246
ResNet50	96.1	0.0212163	7.972	132
ResNet101	96.9	0.0652922	12.81	153
Average values:	95.933	0.043	9.598	161.444

[a]Iterations indicate the number of iterations when verification accuracies reach the maximum value for the first time.

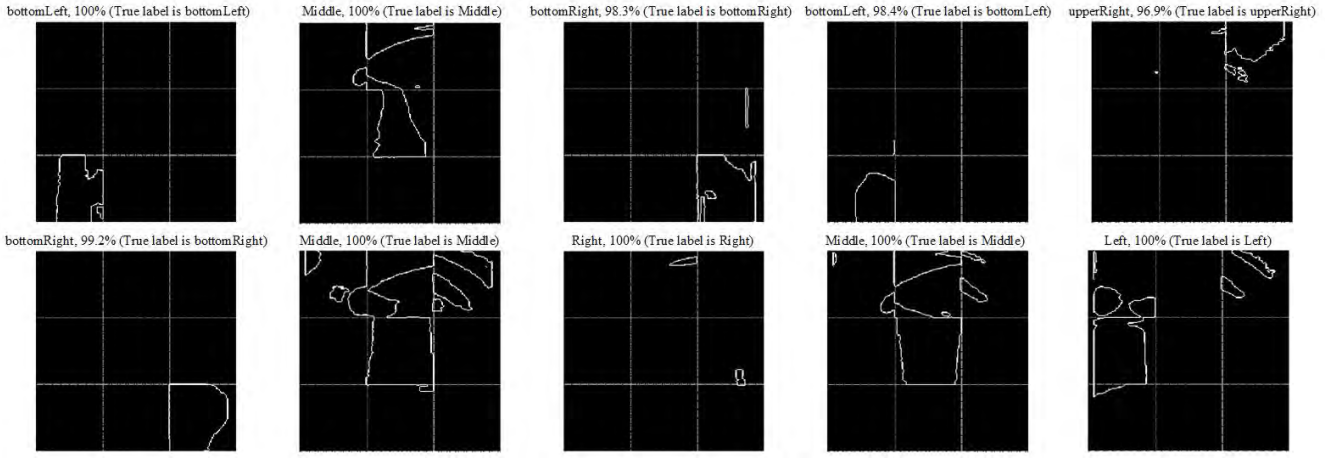

FIGURE 15. 10 Randomly selected samples of fault localization results based on several monitoring sensors (9 sensors). And the fault areas were outlined by LBM-LSM image segmentation method. The title of each picture displays the localization category with the highest probability.

relatively poor in these situations, because it is difficult for shallow networks to extract deep features.

V. ANALYSIS

A. COMPREHENSIVE EVALUATION INDICATORS/INDICES

To analyze which convolutional neural network is the most suitable one for the fault diagnosis situations in this study, validation results in Table 6 to Table 9 were divided by the values of the first line respectively, which is the normalization process. Subsequently, the loss value, the iteration time, and the iterations of the same network were summed, and the results are listed in Table 10. These indicators were all expected to be as small as possible. Therefore, the evaluation indicators of each network were added together to obtain their comprehensive evaluation indicators respectively, shown in the last column of Table 10. This Table suggests that the evaluation indicator of GoogLeNet proved it the best network (the smallest). Besides, the indicators of AlexNet and DenseNet also showed good performance.

TABLE 10. Evaluation indicators of the fault localization results based on different CNNs.

Name	Sum of the normalized indicators				Evaluation indicators
	table VI	table VII	table VIII	table IX	
GoogLeNet	3.000	3.000	3.000	3.000	12.000
VGG16	5.040	6.679	29.444	9.608	50.771
VGG19	7.039	11.541	20.528	7.375	46.483
SqueezeNet	1436.374	59.712	41.893	10.042	1548.021
AlexNet	1.080	35.412	1.548	3.269	41.309
DenseNet	18.786	13.599	7.490	6.537	46.411
ResNet18	245.405	32.954	7.779	3.980	290.118
ResNet50	137.582	59.601	4.789	4.565	206.537
ResNet101	128.970	326.319	7.815	8.428	471.533

In addition, indicators of fault localization results (including the loss value, the iteration time and the number of iterations) only reflect part of the diagnosing effects. The overall accuracies and the stabilities of the diagnosis process should also be considered. The average verification accuracy of the last 50% iterations (the last 160 iterations) is denoted as P_{iter}, and the standard deviation of the last 50% iterations is denoted as S_N^2. Furthermore, the number of iterations used to calculate was *iter*, while the *iter* in this study was 160. the evaluation index was defined as the average value minus the standard deviation, as shown in Equation (27).

$$P_{iter} - S_N^2 = P_{iter} - \sqrt{\frac{1}{iter}\sum_{i=1}^{iter}\left(P_{iter_i} - P_{iter}\right)^2} \quad (27)$$

This index is expected to be as large as possible. The average verification accuracies and standard deviations for different deep CNNs in Fig. 8, Fig. 10, Fig. 12 and Fig. 14 are listed in table 11. According to the last column in table 11, the comprehensive verification effect of DenseNet is the best and the most stable one, then followed by GoogLeNet.

According to normalized evaluation indicators presented in Table 10, the top three networks (and their indices) were GoogLeNet (12), AlexNet (41.306) and DenseNet (46.411). And likewise, according to evaluation indices in Table 10, the top three networks (and their indices) were DenseNet (394.1), GoogLeNet (393.4), ResNet50 (387.9). In addition, considering the basic network parameters in Table 4, parameters of GoogLeNet are less than the others' and its size of network is smaller. Therefore, GoogLeNet is more suitable for the fault localization situations in this study. To judge the pros and cons of the model more rigorously, we conducted the Friedman test and then the Nemenyi post hoc test. The experimental results are to be given in the next section.

B. MODEL SELECTION

We evaluated nine networks using the results of the 5 -fold cross-validation, Friedman test, and Nemenyi post hoc test. A flow chart of the evaluation process is shown in Fig. 16.

The Friedman test is one of the statistical hypothesis tests used to compare multiple systems and solve multi-classification problems [45]. Thus, it is applicable to the case of this study.

TABLE 11. Overall evaluation indices of the fault localization processes based on various CNNs.

Name	Average (%)				Standard deviation				Evaluation indices
	table VI	table VII	table VIII	table IX	table VI	table VII	table VIII	table IX	
GoogLeNet	99.9998	99.7601	98.9402	96.3314	0.0014	0.3939	0.1718	1.0767	393.3878
VGG16	98.6557	99.1157	98.6491	89.4046	3.4069	1.7452	0.0067	7.6021	373.0641
VGG19	99.7877	99.2872	96.6474	93.1787	1.1321	2.1007	3.9916	5.6262	376.0505
SqueezeNet	98.4906	95.9151	96.5926	92.2659	2.2302	5.8379	2.2102	3.9775	369.0083
AlexNet	99.9998	98.8111	97.8704	95.0294	0.0014	2.6063	4.0915	2.6336	382.3778
DenseNet	99.9764	99.7000	99.0500	96.4434	0.1701	0.0000	0.0000	0.8757	394.1241
ResNet18	96.1121	98.4907	97.3133	94.5006	2.3084	1.5811	1.5355	2.8710	378.1208
ResNet50	99.0852	98.9041	98.1264	95.9088	1.3911	1.3816	0.1719	1.2078	387.8720
ResNet101	98.6850	97.8938	98.3625	96.7255	2.1636	2.0381	0.3608	1.0196	386.0846
Average values:	98.9769	98.6531	97.9502	94.4209	1.4228	1.9650	1.3933	2.9878	382.2322

TABLE 12. The five-fold cross-validation results.

Data Set	Networks	Validation results (the rank values were listed in the parentheses)					Average ranks (AR)
		Loss	Time (s/iteration)	Iterations [a]	Average [b]	Standard deviation	
D1	GoogLeNet	0.021600 (5)	2.85 (4)	108 (2)	96.3314 (3)	1.0767 (3)	3.4
	VGG16	0.008482 (2)	23.72 (9)	150 (5)	89.4046 (9)	7.6021 (9)	6.8
	VGG19	0.001258 (1)	19.02 (8)	168 (7)	93.1787 (7)	5.6262 (8)	6.2
	SqueezeNet	0.170208 (9)	1.31 (2)	286 (9)	92.2659 (8)	3.9775 (7)	7
	AlexNet	0.048564 (7)	1.28 (1)	96 (1)	95.0294 (5)	2.6336 (5)	3.8
	DenseNet	0.014898 (3)	14.73 (7)	114 (3)	96.4434 (2)	0.8757 (1)	3.2
	ResNet18	0.033954 (6)	2.69 (3)	246 (8)	94.5006 (6)	2.8710 (6)	5.8
	ResNet50	0.021216 (4)	7.972 (5)	132 (4)	95.9088 (4)	1.2078 (4)	4.2
	ResNet101	0.065292 (8)	12.81 (6)	153 (6)	96.7255 (1)	1.0196 (2)	4.6
D2	GoogLeNet	0.000685 (1)	2.73 (4)	96 (2)	99.0319 (3)	1.9262 (3)	2.6
	VGG16	0.004062 (2)	11.94 (6)	108 (4)	99.1932 (2)	3.5836 (6)	4
	VGG19	0.049244 (7)	14.00 (8)	62 (1)	98.0637 (6)	4.1976 (7)	5.8
	SqueezeNet	0.044134 (6)	1.34 (2)	216 (8)	94.4736 (8)	6.3243 (9)	6.6
	AlexNet	0.014140 (3)	1.30 (1)	105 (3)	99.1932 (1)	1.6846 (2)	2
	DenseNet	0.132542 (9)	15.99 (9)	190 (7)	96.9342 (7)	1.6642 (1)	6.6
	ResNet18	0.080057 (8)	2.24 (3)	252 (9)	93.4248 (9)	4.7212 (8)	7.4
	ResNet50	0.041779 (5)	7.29 (5)	121 (5)	98.3461 (5)	2.2855 (5)	5
	ResNet101	0.022752 (4)	12.25 (7)	122 (6)	98.7092 (4)	2.2080 (4)	5
D3	GoogLeNet	0.037661 (4)	2.11 (4)	99 (2)	98.3413 (3)	2.2537 (5)	3.6
	VGG16	0.010183 (2)	12.57 (9)	119 (4)	97.5078 (4)	4.0449 (7)	5.2
	VGG19	0.002557 (1)	12.01 (8)	98 (1)	93.8788 (7)	4.1883 (8)	5
	SqueezeNet	0.059435 (7)	1.08 (2)	217 (8)	91.5426 (8)	4.5040 (9)	6.8
	AlexNet	0.019974 (3)	1.05 (1)	171 (7)	97.3951 (5)	1.8829 (3)	3.8
	DenseNet	0.052580 (6)	11.98 (7)	103 (3)	94.9006 (6)	1.3305 (1)	4.6
	ResNet18	0.064405 (9)	1.85 (3)	260 (9)	90.2921 (9)	3.8368 (6)	7.2
	ResNet50	0.063651 (8)	5.68 (5)	144 (5)	99.4203 (2)	1.7942 (2)	4.4
	ResNet101	0.040848 (5)	9.64 (6)	144 (6)	99.4611 (1)	2.0028 (4)	4.4
D4	GoogLeNet	0.019598 (3)	1.07 (1)	106 (3)	96.7879 (3)	1.9223 (3)	2.6
	VGG16	0.064305 (9)	1.97 (3)	257 (9)	96.6814 (4)	3.8666 (6)	6.2
	VGG19	0.061750 (8)	5.60 (5)	142 (5)	94.6048 (8)	1.7169 (2)	5.6
	SqueezeNet	0.043255 (5)	9.62 (6)	156 (6)	95.8349 (7)	2.0023 (4)	5.6
	AlexNet	0.002696 (1)	1.10 (2)	97 (1)	96.4886 (5)	4.1707 (8)	3.4
	DenseNet	0.036270 (4)	2.19 (4)	178 (7)	96.9012 (2)	2.2048 (5)	4.4
	ResNet18	0.051202 (6)	12.35 (8)	104 (2)	97.5871 (1)	1.2723 (1)	3.6
	ResNet50	0.011053 (2)	12.02 (7)	113 (4)	96.0193 (6)	3.9817 (7)	5.2
	ResNet101	0.059952 (7)	12.68 (9)	208 (8)	90.992 (9)	4.6486 (9)	8.4
D5	GoogLeNet	0.004320 (2)	2.73 (4)	95 (2)	99.7176 (1)	1.1565 (1)	2
	VGG16	0.014955 (4)	11.94 (6)	88 (1)	98.1848 (6)	3.5065 (9)	5.2
	VGG19	0.035305 (6)	14.00 (8)	98 (3)	98.4671 (4)	2.9981 (7)	5.6
	SqueezeNet	0.003060 (1)	1.34 (2)	145 (6)	98.3864 (5)	2.5172 (5)	3.8
	AlexNet	0.004476 (3)	1.30 (1)	142 (5)	99.0319 (3)	1.3774 (3)	3
	DenseNet	0.017079 (5)	15.99 (9)	98 (4)	99.3142 (2)	1.2654 (2)	4.4
	ResNet18	0.061625 (8)	2.24 (3)	226 (9)	96.2888 (9)	3.1957 (8)	7.4
	ResNet50	0.095801 (9)	7.29 (5)	147 (7)	96.7326 (8)	1.4421 (4)	6.6
	ResNet101	0.037297 (7)	12.25 (7)	150 (8)	98.0234 (7)	2.8626 (6)	7

[a]Iterations indicate the number of iterations when verification accuracies reach the maximum value for the first time.
[b]The rank values of the 'average' are ranked from large to small.

1) FIVE-FOLD CROSS-VALIDATION

Because neural networks are highly sensitive to the initial weight parameters, diagnosis results might be inconsistent. Validation accuracies would fluctuate around the normal states. The verification accuracies of the nine algorithms selected in this study all achieved above 90%, and the fluctuation intervals were partially coincident. Therefore, the error of one test set is not sufficient to approximate the

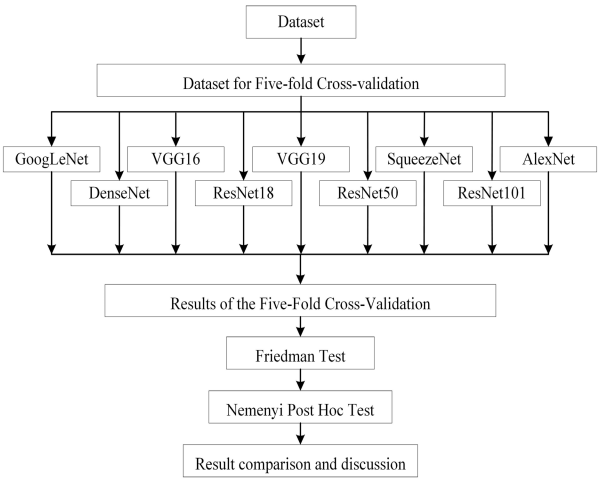

FIGURE 16. A flow chart of the evaluation process.

generalization error. This study divided the data set D into five disjoint subsets, namely: $D = D_1 \cup D_2 \cup \ldots \cup D_5, D_i \cap D_j = \emptyset (i \neq j)$. During each validation process, four subsets were used as training data and the other were used as validation data. The application scenario in Section IV-B was taken as an example to evaluate the networks. The testing results are listed in Table 7, where D_1 denotes the testing results of Section IV-B. The testing results were sorted according to the effects from good to bad, and the corresponding ranks are listed in parentheses.

2) FRIEDMAN STATISTICAL HYPOTHESIS TEST AND NEMENYI SUBSEQUENT TESTING

First of all, the assumption was listed as follows:

H0: In this study, the 9 algorithms are not significant different;

H1: In this study, the 9 algorithms are significantly different;

The Friedman test result obtained using SPSS reached a conclusion that the 9 diagnosis methods in this study have significant differences, because the significance level was lower than 0.05, and the test result was judged to accept hypothesis H1.

Next, The Nemenyi test is applied to quantify the significant differences between each two models. First, we calculate the average rank(AR) of each network: GoogLeNet 2.92, VGG16 5.48, VGG19 5.64, SqueezeNet 5.96, AlexNet 3.12, DenseNet 4.64, ResNet18 6.28, ResNet50 5.08, ResNet101 5.88. Then Nemenyi post hoc test was applied and Critical Difference (CD) value was calculated via Equation (28) (for $\alpha = 0.05$, the Critical Value $q_{0.05}$ for 9 classifiers is 3.102) [44].

$$CD = q_\alpha \sqrt{\frac{K\,(K+1)}{12N}} \tag{28}$$

where K denotes the total number of networks, and N is the total number of datasets used in the study.

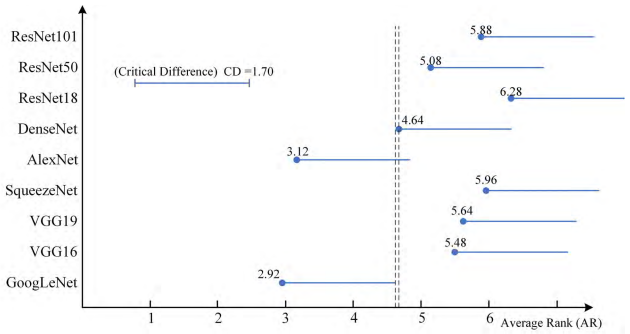

FIGURE 17. Average Rank for the results of various CNNs.

There were 9 models to be compared and 5 data sets for Statistical hypothesis test. The evaluation indicators of the diagnosing effect include not only the accuracy rate, but also five evaluation attributes for each data set. The critical range domain CD=1.70 was calculated by equation (28). Fig. 17 shows the performance ranks of the networks, along with Nemenyi's critical difference (CD) tail. The length of each tail was equal to the critical range. The x-axis shows the average order value of all data sets and their attributes, and the y-axis represents the name of the networks to be detected. Two vertical dashed lines were inserted to clearly identify the end of the best performing classifier's tail and the start of the next significantly different classifier.

As shown in Fig. 17, the performance of the GoogLeNet is the best as its AR value is 2.92. Also, AlexNet has good performance, which didn't show significant differences with GoogLeNet. DenseNet has a slow training speed, but its accuracy and stability during the diagnosing process are excellent. The evaluation indicators focused in this study was not beneficial to this network. In addition, according to Fig. 17, other classification techniques were worse (VGGnet, SqueezeNet, and ResNet) in the case of this study.

VI. CONCLUSION

This study proposed a fault localization method for transformer thermal faults by deep CNNs and image processing. Considering the shortage of fault data samples, transfer learning was adopted, and transformer LBM simulation results were served as a low-layer dataset. Image segmentation method was applied to extract fault features and reduced the data volume significantly.

First, distributions of the transformer temperature and velocity field were obtained through LBM simulation, and the corresponding image fusions were performed. Second, the fault areas of these images were outlined using level set method, which highlighted fault features and reduced the size of data. Fault locations were equivalent to different labels, and different CNNs were trained to classify fault locations. Finally, monitoring information of several sensors placed around the transformer windings were trained using different CNNs. Furthermore, the fault localization results before and after the image segmentation were analyzed.

The proposed intelligent fault localization method could locate internal faults of transformers effectively. The average localization accuracies of different CNNs (the last 50% iterations) before and after the image segmentation slightly decreased from 98.9769% to 98.6531%, as shown in the last row of Table XI. In the meantime, according to Table VI and VII, the average calculation time per iteration of all the CNNs dropped by 7.3615%, from 13.177 sec to 12.207 sec. The loss value reduced by 72.549%, from 0.051 to 0.014. Besides, the data capacities reduced to approximately 1% of the counterparts in original data set. Likewise, the average fault localization accuracies of all the CNNs (the last 50% iterations) before and after image segmentation process reached 97.9502% and 94.4209%, respectively, when using the data set of partial areas. The fault diagnosing accuracies of in all cases was much higher than the SVM's (81.86%). Furthermore, the average calculation time per iteration based on different CNNs was presented in Table 8 and Table 9, and it dropped by 8.816%, from 10.526 to 9.598 sec per iteration. In addition, the loss value reduced by 37.6812%, from 0.069 to 0.043. Finally, the evaluation method was proposed, and Friedman statistical hypothesis test was performed to verify the applicabilities of different network models in this case.

REFERENCES

[1] M. Bagheri, A. Zollanvari, and S. Nezhivenko, "Transformer fault condition prognosis using vibration signals over cloud environment," *IEEE Access.*, vol. 6, pp. 9862–9874, 2018.

[2] G. Rigatos and P. Siano, "Power transformers' condition monitoring using neural modeling and the local statistical approach to fault diagnosis," *Int. J. Elect. Power Energy Syst.*, vol. 80, pp. 150–159, Sep. 2016.

[3] J. R. da Silva and J. P. A. Bastos, "Online evaluation of power transformer temperatures using magnetic and thermodynamics numerical modeling," *IEEE Trans. Magn.*, vol. 53, no. 6, Jun. 2017, Art. no. 8106104.

[4] D. Susa and M. Lehtonen, "Dynamic thermal modeling of power transformers: Further development—Part II," *IEEE Trans. Power Del.*, vol. 21, no. 4, pp. 1961–1970, Oct. 2016.

[5] S. Tenbohlen, N. Schmidt, C. Breuer, S. Khandan, and R. Lebreton, "Investigation of thermal behavior of an oil-directed cooled transformer winding," *IEEE Trans. Power Del.*, vol. 33, no. 3, pp. 1091–1098, Jun. 2018.

[6] L. Deng *et al.*, "Modeling and analysis of parasitic capacitance of secondary winding in high-frequency high-voltage transformer using finite-element method," *IEEE Trans. Appl. Supercond.*, vol. 28, no. 3, Apr. 2018, Art. no. 5500105.

[7] P.-Y. Wang, J.-M. Liu, Z.-H. Liu, and Y.-J. Chen, "Experiment and simulation of natural convection heat transfer of transformer oil under electric field," *Int. J. Heat Mass Transf.*, vol. 115, pp. 441–452, Dec. 2017.

[8] M. Lahonian and A. A. Golneshan, "Numerical study of temperature distribution in a spherical tissue in magnetic fluid hyperthermia using lattice Boltzmann method," *IEEE Trans. Nanobiosci.*, vol. 10, no. 4, pp. 262–268, Dec. 2011.

[9] M. M. Islam, G. Lee, and S. N. Hettiwatte, "A review of condition monitoring techniques and diagnostic tests for lifetime estimation of power transformers," *Elect. Eng.*, vol. 100, no. 2, pp. 581–605, Jun. 2018.

[10] A. J. Christina, M. A. Salam, Q. M. Rahman, F. Wen, S. P. Ang, and W. Voon, "Causes of transformer failures and diagnostic methods—A review," *Renew. Sustain. Energy Rev.*, vol. 82, no. 1, pp. 1442–1456, Feb. 2018.

[11] J. Faiz and M. Soleimani, "Dissolved gas analysis evaluation in electric power transformers using conventional methods a review," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 24, no. 2, pp. 1239–1248, Apr. 2017.

[12] W. Min, H. Cao, J. Cao, H.-L. Nguyen, J. B. Gomes, and S. P. Krishnaswamy, "An overview of state-of-the-art partial discharge analysis techniques for condition monitoring," *IEEE Elect. Insul. Mag.*, vol. 31, no. 6, pp. 22–35, Dec. 2015.

[13] S. Hashemnia, "Charactrisation of power transformer frequency response signature using finite element analysis," Ph.D dissertation, Dept. Elect. Eng., Univ. Curtin, Perth, WA, Australia, 2014.

[14] T. Mariprasath and V. Kirubakaran, "A real time study on condition monitoring of distribution transformer using thermal imager," *Infr. Phys. Technol.*, vol. 90, pp. 78–86, May 2018.

[15] X. Zhang, Z. Wang, and Q. Liu, "Interpretation of hot spot factor for transformers in OD cooling modes," *IEEE Trans. Power Del.*, vol. 33, no. 3, pp. 1071–1080, Jun. 2017.

[16] A. Y. Arabul and I. Senol, "Development of a hot-spot temperature calculation method for the loss of life estimation of an ONAN distribution transformer," *Elect. Eng.*, vol. 100, no. 3, pp. 1651–1659, Sep. 2017.

[17] T. Touret, C. Changenet, F. Ville, M. Lalmi, and S. Becquerelle, "On the use of temperature for online condition monitoring of geared systems—A review," *Mech. Syst. Signal Process.*, vol. 101, pp. 197–210, Feb. 2018.

[18] J. Wang, C. Luo, H. Huang, H. Zhao, and S. Wang, "Transferring pre-trained deep CNNs for remote scene classification with general features learned from linear PCA network," *Remote Sens.*, vol. 9, pp. 1–255, Mar. 2017.

[19] Y. Xue and P. Beauseroy, "Transfer learning for one class SVM adaptation to limited data distribution change," *Pattern Recognit. Lett.*, vol. 100, pp. 117–123, Dec. 2017.

[20] J. Wang, H. Zheng, Y. Huang, and X. Ding, "Vehicle type recognition in surveillance images from labeled Web-nature data using deep transfer learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2913–2922, Sep. 2018.

[21] Z. Kolar, H. Chen, and X. Luo, "Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images," *Automat. Construct.*, vol. 89, pp. 58–70, May 2018.

[22] O. Janssens, R. Van de Walle, M. Loccufier, and S. Van Hoecke, "Deep learning for infrared thermal image based machine health monitoring," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 151–159, Jul. 2017.

[23] L. Duan, M. Yao, J. Wang, T. Bai, and L. Zhang, "Segmented infrared image analysis for rotating machinery fault diagnosis," *Infr. Phys. Technol.*, vol. 77, pp. 267–276, Jul. 2016.

[24] V. T. Tran, B.-S. Yang, F. Gu, and A. Ball, "Thermal image enhancement using bi-dimensional empirical mode decomposition in combination with relevance vector machine for rotating machinery fault diagnosis," *Mech. Syst. Signal Process.*, vol. 38, no. 2, pp. 601–614, Jul. 2013.

[25] K. Lee, J. Lee, J. Lee, S. Hwang, and S. Lee, "Brightness-based convolutional neural network for thermal image enhancement," *IEEE Access*, vol. 5, pp. 26867–26879, Nov. 2017.

[26] S. Balla-Arabé, X. Gao, and B. Wang, "A fast and robust level set method for image segmentation using fuzzy clustering and lattice Boltzmann method," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 910–920, Jun. 2013.

[27] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access*, vol. 6, pp. 30370–30377, 2018.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[29] Y. Qi, C. Shen, D. Wang, J. Shi, X. Jiang, and Z. Zhu, "Stacked sparse autoencoder-based deep network for fault diagnosis of rotating machinery," *IEEE Access*, vol. 5, pp. 15066–15079, Jul. 2017.

[30] J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 185–195, Jan. 2017.

[31] L. Zhao and X. Wang, "A deep feature optimization fusion method for extracting bearing degradation features," *IEEE Access*, vol. 6, pp. 19640–19653, 2018.

[32] W. Zhang *et al.*, "LSTM-based analysis of industrial IoT equipment," *IEEE Access*, vol. 6, pp. 23551–23560, Apr. 2018.

[33] A. A. Mohamad, "The Boltzmann equation," in *Lattice Boltzmann Method: Fundamentals and Engineering Applications With Computer Codes*. London, U.K.: Springer, 2011, ch. 2, sec. 3, pp. 19–22.

[34] S. Chen and G. D. Doolen, "Lattice Boltzmann method for fluid flows," *Annu. Rev. Fluid Mech.*, vol. 30, pp. 329–364, Jan. 1998.

[35] X. He and L. S. Luo, "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 56, pp. 6811–6817, Dec. 1997.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[37] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[38] C. Szegedy. (Sep. 2015). "Going deeper with convolutions." [Online]. Available: https://arxiv.org/abs/1409.4842
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2018, pp. 4700–4708.
[41] L. N. Forrest *et al.*, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Nov. 2016, pp. 1–13.
[42] K. He, G. Gkioxari, P. Dollár, and R. Girshick. (2017). "Mask R-CNN." [Online]. Available: https://arxiv.org/abs/1703.06870
[43] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64270–64277, Oct. 2018.
[44] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Math Statist., Princeton Univ., Princeton, NJ, USA, 1963.
[45] M. Mohammadi, W. Hofman, and Y.-H. Tan, "A comparative study of ontology matching systems via inferential statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 615–628, Apr. 2019.

JIAJUN DUAN received the B.S. degree from Chongqing University, Chongqing, China, in 2016. He is currently pursuing the Ph.D. degree in electrical engineering with Wuhan University, Wuhan, China. His current research interests include intelligent monitoring systems for transformers and deep learning.

YIGANG HE received the M.Sc. degree in electrical engineering from Hunan University, Changsha, China, in 1992, and the Ph.D. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1996. In 1990, he joined the College of Electrical and Information Engineering, Hunan University, and was promoted to Associate Professor and Professor, in 1996 and 1999, respectively. From 2006 to 2011, he was the Director of the Institute of Testing Technology for Circuits and Systems, Hunan University. He was a Senior Visiting Scholar with the University of Hertfordshire, Hatfield, U.K., in 2002. In 2011, he joined the Hefei University of Technology, Hefei, China. In 2017, he joined Wuhan University, Wuhan, China, where he is currently the Vice Dean of the School of Electrical Engineering. He has authored over 200 journal and conference papers in the aforementioned areas and several chapters in edited books. His current research interests include the areas of circuit theory and its applications, testing and fault diagnosis of analog and mixed-signal circuits, electrical signal detection, smart grid, radio frequency identification technology, and intelligent signal processing.

Dr. He has been serving on the Technical Program Committee of a number of international conferences. He was a recipient of a number of national and international awards, prizes, and honors.

BOLUN DU received the B.S. degree from the College of Electrical and Information Engineering, Hunan University, Changsha, China, in 2016. He is currently pursuing the Ph.D. degree with Wuhan University, Wuhan, China. His research interest includes power electronic circuit fault diagnosis.

RUAA M. RASHAD GHANDOUR received the B.S. degree in electrical engineering from the Sudan University of Science and Technology, Khartoum, Sudan. She is currently pursuing the Ph.D. degree with Wuhan University, Wuhan, China. She served as an Electrical Engineer with Sudanese Electricity and Distribution Company. Her main research interests include power transformer fault diagnosis and prediction.

WENJIE WU received the B.S. degree from Wuhan University, Wuhan, China, in 2017, where she is currently pursuing the master's degree in electrical engineering. Her research interest includes transformer fault diagnosis and prognosis.

HUI ZHANG received the M.Sc. degree in electrical engineering from the Wuhan University of Hydropower and Electrical Engineering, Wuhan, in 1997, and the Dr.-Ing. degree in electrical engineering from the Technical University of Karlsruhe, Germany (now Karlsruhe Institute of Technology, KIT), in 2004.

She is currently a Professor with the School of Electrical Engineering, Wuhan University. Her research interests include many fields, such as high-performance optimal pulse width modulation techniques for high power converters and power active filtering, power quality, smart grid, and so on.