World Electric Vehicle Journal

MDPI

Article

Power Converter Fault Detection Using MLCA–SpikingShuffleNet

Li Wang, Feiyang Zhu *, Fengfan Jiang and Yuwei Yang

School of Electrical Engineering and Automation, Nantong University, Nantong 226019, China; lwee@ntu.edu.cn (L.W.); 2212310009@stmail.ntu.edu.cn (F.J.); 2412310002@stmail.ntu.edu.cn (Y.Y.)
* Correspondence: 2212310035@stmail.ntu.edu.cn

Abstract: With the widespread adoption of electric vehicles, the power converter, as a key component, plays a crucial role. Traditional fault detection methods often face challenges in real-time performance and computational efficiency, making it difficult to meet the demands of electric vehicle power converters for efficient and accurate fault diagnosis. To address this challenge, this paper proposes a novel fault detection model—SpikingShuffleNet. This paper first designs an efficient SpikingShuffle Unit that integrates grouped convolutions and channel shuffle techniques, effectively reducing the model's computational complexity by optimizing feature extraction and channel interaction. Next, by appropriately stacking SpikingShuffle Units and refining the network architecture, a complete lightweight diagnostic network is constructed for real-time fault detection in electric vehicle power converters. Finally, the Mixed Local Channel Attention mechanism is introduced to address the potential limitations in feature representation caused by grouped convolutions, further enhancing fault detection accuracy and robustness by balancing local detail preservation and global feature integration. Experimental results show that SpikingShuffleNet exhibits excellent accuracy and robustness in the fault detection task for power converters, fulfilling the real-time fault diagnosis requirements for low-power embedded devices.

Keywords: power converter; fault detection; depth-wise convolution; channel shuffle; mixed local channel attention; embedded devices; electric vehicle

Academic Editor: Grzegorz Sierpiński

Received: 15 December 2024
Revised: 9 January 2025
Accepted: 10 January 2025
Published: 12 January 2025

Citation: Wang, L.; Zhu, F.; Jiang, F.; Yang, Y. Power Converter Fault Detection Using MLCA–SpikingShuffleNet. *World Electr. Veh. J.* **2025**, *16*, 36. https://doi.org/10.3390/wevj16010036

1. Introduction

Electric vehicles play an indispensable role in achieving global sustainable development goals, significantly reducing greenhouse gas emissions and noise pollution while driving the widespread adoption of renewable energy and fostering technological innovation in transportation infrastructure [1]. However, the efficient operation of electric vehicles relies on the support of key components, among which power converters, as the core elements connecting the battery and the powertrain system, directly impact the energy efficiency and reliability of electric vehicles. The power system of an electric vehicle, as illustrated in Figure 1, includes the alternating current (AC)/direct current (DC) charging interface, high-voltage direct current bus, battery systems comprising both power batteries and auxiliary batteries, powertrain systems such as motors and transmissions, and vehicle control systems including electric air conditioning, power steering, and lighting. Among these core modules, power converters act as vital bridges, facilitating energy transfer between high-voltage batteries and low-voltage systems, as well as between DC and AC systems, thereby ensuring the proper functioning of all subsystems.

However, the technical complexity of electric vehicles, particularly the integration of battery management systems, power systems, and power conversion systems, has significantly increased the reliability and performance requirements of key components [2–4]. As

a bridge between high-voltage batteries and low-voltage systems, the stability of power converters directly affects the overall operational efficiency, performance, and safety of electric vehicles. Over prolonged use, power converters are prone to various faults. Common issues, such as capacitor aging and switch failures, can lead to degraded system performance, compromised voltage stability and energy efficiency, and even equipment damage. Moreover, as a crucial device providing power to the battery management system and vehicle control system, any fault in the converter may result in the failure or malfunctioning of these systems, disrupting battery charge–discharge control and jeopardizing vehicle safety. This risk is especially critical under complex operating conditions such as high-speed driving, rapid acceleration, or emergency braking, where faults in the power converter can lead to transient system instability and heightened safety hazards.

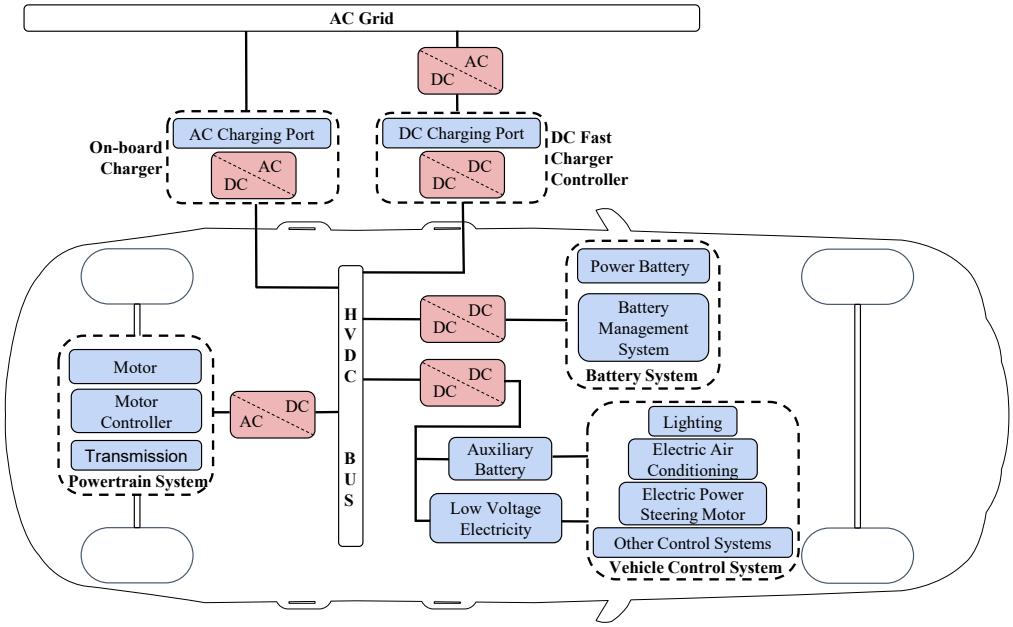

Figure 1. The architecture of an electric vehicle power system.

Therefore, real-time fault diagnosis of power converters is essential. Timely and accurate fault detection not only ensures the stability of electric vehicles under various complex conditions but also enhances driving safety, extends system lifespan, and advances the development of electric vehicle technology.

In recent years, significant progress has been made in the field of fault diagnosis, with the main research approaches being model-based methods and data-driven methods [5]. Model-based methods estimate system states by constructing physical models, providing strong theoretical interpretability. Shao et al. [6] proposed a fault diagnosis method for power systems by integrating intuitionistic fuzzy sets and incidence matrices, which enhances fault tolerance and adapts to topology changes. Peng et al. [7] achieved robust and millivolt-level fault detection and localization for lithium-ion battery packs through full-dimensional statistical analytics and eigenvalue decomposition. Sravani et al. [8] developed a second-order transfer function model based on transient data from computational fluid dynamics, which is used in a Linear-Parameter-Varying observer to generate residuals for fault detection in orifice flowmeters. Zanuso et al. [9] developed a generalized model for interturn short-circuit faults in induction machines, validated experimentally, and enhanced fault detection using a high-frequency injection method. Jiang et al. [10] proposed an adaptive low-rank tensor estimation model for multichannel weak fault detection in rolling bearings, achieving accurate fault characteristic extraction under strong background

noise. However, these methods are highly dependent on the accuracy of the models. Due to the difficulty in modeling complex nonlinear systems, their practical applications are somewhat limited.

With the rise of machine learning and deep learning, data-driven methods have gradually become a research hotspot in fault diagnosis. These methods automatically learn feature representations from large-scale data, demonstrating powerful capabilities in processing complex multidimensional signals [11].

Zhang et al. [12] proposed a battery voltage prediction and fault diagnosis method based on an improved Gradient Boosting Decision Tree and Isolation Forest, achieving high accuracy and strong real-time performance. Zhang and He [13] developed a multiscale spatial–temporal graph neural network, combining multiscale graph operators and gated recurrent units to learn spatial–temporal fault features. Rokocakau et al. [14] proposed a fault diagnosis method for two-level voltage source inverters using shallow neural networks, employing classifiers for fault detection and localization. Alshehri et al. [15] were the first to apply residual neural networks to traffic flow prediction. Their proposed architecture effectively predicts the origin-to-destination flows for the entire network using link flows, thereby reversing the traditional traffic assignment problem. Moussa et al. [16] proposed a transfer learning method based on deep convolutional neural networks to address the dynamic modulus prediction problem. Wu et al. [17] proposed an intelligent fault diagnosis algorithm based on spiking neural networks to address safety and fault detection in high-energy batteries for electric vehicles. Validation results demonstrate the algorithm's rapid and accurate detection capabilities, along with good portability. However, while these algorithms have demonstrated good diagnostic accuracy, the high computational complexity of these neural network models still poses challenges for their deployment in embedded devices.

Despite the advantages of the aforementioned methods, they also have significant limitations. Model-based methods rely on data analysis and cannot perform real-time fault monitoring during device operation. Data-driven methods generally suffer from low computational efficiency, particularly in embedded hardware environments, making them unsuitable for the low-power and high-efficiency requirements of fault diagnosis in electric vehicle power converters. Therefore, addressing these issues in traditional methods and developing a low-power, noninvasive, real-time fault detection approach has become an important research direction in the field of electric vehicle power converter fault detection.

To this end, this study proposes a novel fault detection model named SpikingShuffleNet, designed to address the shortcomings of traditional fault detection methods in terms of real-time performance, computational efficiency, and feature extraction capabilities. The model offers an efficient, accurate, and low-power in situ real-time fault detection solution for electric vehicle power converters. Considering the limitations of existing studies, the key contributions of this paper are highlighted as follows:

1. An efficient SpikingShuffle network module was designed, combining deep convolution and channel shuffle techniques to optimize feature extraction and channel interaction, significantly reducing the model's computational complexity.
2. A complete lightweight diagnostic network was constructed for real-time fault detection in electric vehicle DC–DC power converters by strategically stacking SpikingShuffle modules and optimizing the network architecture.
3. The Mixed Local Channel Attention (MLCA) mechanism was introduced to mitigate the accuracy degradation caused by grouped convolutions and channel shuffle, effectively balancing local detail preservation and global feature integration, resulting in more efficient and precise fault detection performance.

The structure of this paper is organized as follows: Section 2 provides a detailed introduction to the proposed model architecture. Section 3 presents the dataset construction, experimental design, evaluation metrics, and result analysis, validating the effectiveness of the proposed model. Finally, Section 4 summarizes the research contributions.

2. SpikingShuffleNet Enhanced by Mixed Local Channel Attention

This section provides a detailed introduction to the proposed SpikingShuffleNet model, which combines the Leaky Integrate-and-Fire (LIF) information transmission model with the ShuffleNet architecture and incorporates the MLCA mechanism. Specifically, this section covers the following parts: First, the operational mechanism of the LIF information transmission model is explained. Next, the network structure of SpikingShuffleNet and its role in lightweight model design is introduced, with a particular focus on the applications of channel shuffle and depth-wise convolution. Finally, the MLCA mechanism is discussed in terms of how it enhances the network's feature selection capability and learning performance.

2.1. Leaky Integrate-and-Fire Information Transmission Model

In the era of continuous advancements in artificial intelligence, neural networks have emerged as one of the pivotal technologies driving the progress of machine learning and deep learning. Although artificial neural networks have achieved significant success across various domains, they still exhibit limitations when handling dynamic and time-varying data, particularly in addressing complex temporal data processing and enhancing energy efficiency. To tackle these challenges, the spiking neural network (SNN), a next-generation neural network model, was introduced.

The design of SNNs is inspired by biological neural systems, where information is transmitted between neurons through electrical pulses [18]. This approach enables the formation of a system capable of efficiently processing complex spatiotemporal data. The core model of SNNs is the LIF model, which simulates the accumulation process of a neuron's membrane potential and emits a spike signal when the potential reaches a certain threshold, thereby facilitating information transmission. It can be expressed as

$$\begin{aligned} \tau_m \frac{dV(t)}{dt} &= -(V(t) - V_{\text{rest}}) + R_m I(t) \\ \lim_{\alpha \to 0; \alpha > 0} V(t+\alpha) &= V_r, \quad \text{if } V(t) > V_{th} \end{aligned} \tag{1}$$

where $V(t)$ and V_{rest} represent the membrane potential and the resting potential, respectively, $I(t)$ denotes the input current, and R_m represents the leak resistance. V_{th} is the firing threshold, while V_r refers to the postspike reset potential. τ_m represents the membrane time constant that characterizes the dynamic properties of a neuron's membrane potential. Specifically, it reflects the speed at which the membrane potential responds to variations in the input current.

Information is transmitted through spike signals rather than continuous signals, a method that not only closely mimics the operation of biological neural systems but also demonstrates superior performance in processing temporal data. This is particularly advantageous in fault detection of power converters in electric vehicles, where SNNs can effectively identify transient fault spike signals. Compared to the continuous data processing approach of artificial neural networks, SNNs are better at capturing temporal variations with higher precision. Additionally, due to their low-power characteristics, SNNs are particularly well suited for real-time, efficient fault detection in energy-constrained environments, such as electric vehicles.

Based on the LIF model, the information transmission model of SNNs is illustrated in Figure 2. Initially, multiple presynaptic neurons generate input spike signals, which are transmitted to the postsynaptic neurons through distinct synaptic weights. It is assumed that a postsynaptic neuron receives spike signals from n presynaptic neurons, with each presynaptic neuron having a learnable weight denoted as $w_1, w_2, \ldots, w_n$. $X_i(t)$ represents the action potential input from the neuron at position i. During the transmission of information, the postsynaptic neuron receives input spikes from n presynaptic neurons and processes the signal by performing a weighted summation. When the membrane potential exceeds a specified firing threshold, the postsynaptic neuron generates an output spike and passes the signal to subsequent neurons. The neuronal dynamics mechanism of SNNs can be modeled as follows:

$$\begin{aligned} \tau_m \frac{dV(t)}{dt} &= -(V(t) - V_{\text{rest}}) + \sum_{i=1}^{n} w_i X_i(t) \\ S(t) &= \begin{cases} 1, & \text{if } V(t) > V_{th} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

where w_j signifies the weight of the j_{th} input synapse. V_{th} is the firing threshold of the neuron. $S(t)$ represents the output of postneurons at time step t.

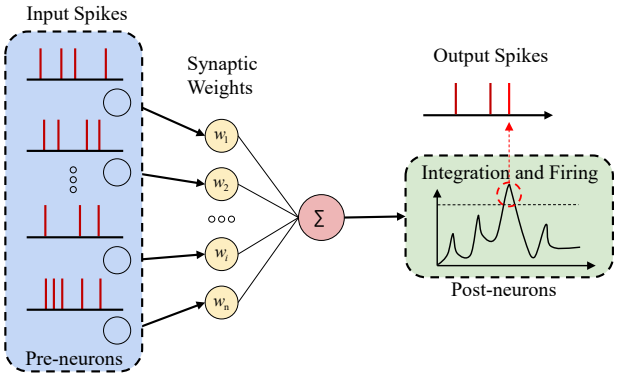

Figure 2. The information transmission model in SNNs.

SNNs transmit information through binary spike signals, with each neuron emitting a spike only when its membrane potential reaches a predefined threshold [19]. This characteristic makes SNNs computationally more efficient, especially when handling spatiotemporal data. Due to the sparsity of spike signals, SNNs can effectively transmit information without significantly increasing the computational load. Furthermore, the low-power nature of SNNs makes them particularly suitable for fault detection in resource-constrained environments, such as electric vehicles, where they can achieve high-efficiency fault detection without substantially increasing energy consumption [20].

Additionally, SNNs exhibit remarkable robustness, maintaining high detection accuracy even in noisy or unstable signal environments. The operating conditions of electric vehicles are often subject to significant variations, including temperature and voltage fluctuations, which can affect the performance of the converter. Through their biologically inspired design, SNNs are able to adapt to different operating environments, ensuring that fault detection remains effective even under unstable or noisy conditions.

By integrating modern neural network architectures, the performance of SNNs in fault detection for electric vehicle power converters can be further enhanced. In summary, with their advantages in temporal data processing, low-power computation, and efficient fault detection, SNNs emerge as an ideal choice for energy-constrained applications such as electric vehicles.

2.2. Optimizing Feature Extraction: Group Convolutions and Channel Shuffle Mechanism

In the design of modern convolutional neural networks, the simplification of overall network architecture is achieved by constructing repetitive modules with identical structures, which not only enhances training efficiency but also increases the depth and complexity of the network. This strategy has been widely adopted in numerous successful network architectures, significantly improving the ability to learn complex features, thereby substantially boosting performance across various application tasks. For instance, networks such as MobileNet [21], EfficientNet [22], and Xception [23] have effectively reduced computational complexity by introducing depth-wise separable convolutions and grouped convolutions, achieving a balance between performance and efficiency while maintaining robust representational power.

However, these network architectures have not sufficiently accounted for the computational burden introduced by 1 × 1 convolutions when the number of channels is excessively large. Although 1 × 1 convolutions offer notable advantages in channel mapping, their computational and memory demands escalate dramatically when the input channels are numerous. To address this challenge, the technique of grouped convolution has been proposed as an effective solution [24,25]. Grouped convolution mitigates this issue by partitioning the input channels into several subsets and performing convolution operations independently on each subset, thereby significantly reducing the computational load. In contrast to traditional convolution layers, which execute fully connected convolution operations for each input–output channel pair, grouped convolutions divide the input channels into multiple groups, and independent convolutions are performed within each group, resulting in a substantial reduction in computational complexity. Specifically, the computational complexity of grouped convolution is calculated as follows:

$$\text{FLOPs}_{\text{Conv}} = C_{\text{in}} \times C_{\text{out}} \times k^2 \tag{3}$$

$$\text{FLOPs}_{\text{GConv}} = g \times \frac{C_{\text{in}}}{g} \times \frac{C_{\text{out}}}{g} \times k^2 = \frac{C_{\text{in}} \times C_{\text{out}} \times k^2}{g} \tag{4}$$

where g represents the number of groups, C_{in} denotes the number of input channels, and C_{out} signifies the number of output channels. k is the size of the convolutional kernel. It is evident that by introducing grouped convolutions, the computational load is significantly reduced, as each convolution within a group only processes a subset of the input channels, resulting in a reduction in computational complexity by a factor of g compared to traditional convolutions. This design effectively optimizes computational efficiency and reduces the demand for computational resources.

Although grouped convolutions offer significant advantages in terms of computational efficiency, they also introduce the issue of hindered information flow when stacked across multiple layers. This is because each grouped convolution operation is performed only on a subset of the input channels, which restricts channel interaction across different groups. Specifically, the output of each group depends solely on the input channels within that group, and there is no direct transmission or interaction of channel information between groups. Figure 3 illustrates the scenario of two stacked grouped convolution layers, where the output of each layer is solely related to its corresponding input channels. This limitation becomes more pronounced when multiple layers are stacked, making it difficult for the network to capture complex relationships between different groups, which in turn affects the model's expressive power and learning capability.

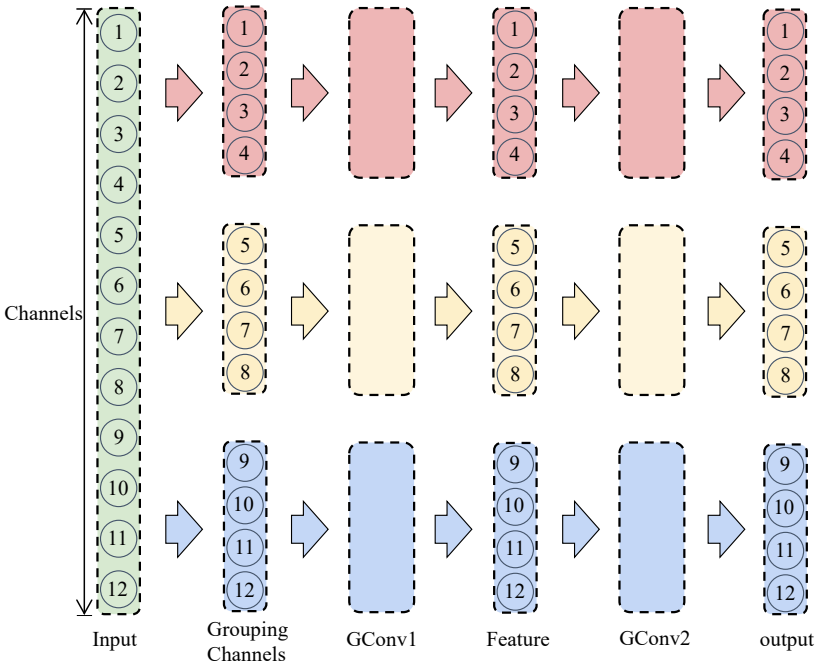

Figure 3. The stacking of group convolutions without channel shuffle.

To address this issue, ShuffleNet introduces the channel shuffle operation, which ensures that feature information can be shared between groups by interleaving data from different channels [26]. As shown in Figure 4, assume the convolutional layer has three groups and the output consists of 3 × 4 channels. The output channels are first rearranged into a matrix of shape (3, 4), representing that the channels from each group are grouped together. Then, a transpose operation is applied to swap the dimensions of the matrix, allowing channels from different groups to interlace with each other. Finally, the transposed matrix is flattened into a one-dimensional array, which is used as the input to the next convolution layer. The channel shuffle operation is differentiable, meaning it can be embedded into the network's architecture and optimized through backpropagation during training. Additionally, this operation adapts to different numbers of groups, ensuring that channel shuffling remains effective regardless of the variation in the number of groups in the convolution layer, thus promoting smooth information flow and enhancing the network's representational capability.

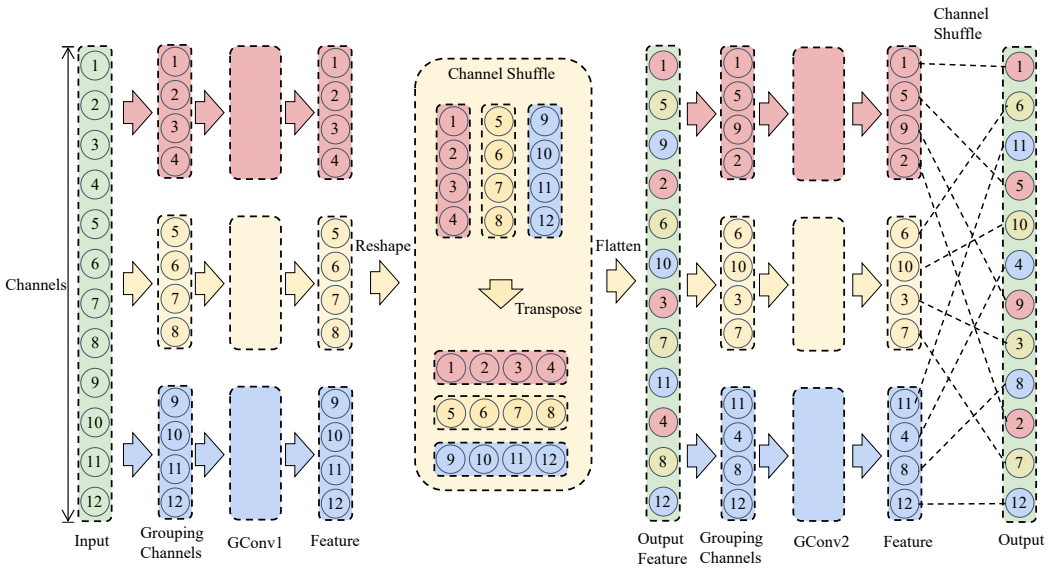

Figure 4. The stacking of group convolutions with channel shuffle.

2.3. SpikingShuffle Unit and Overall Network Architecture

By combining the spike encoding of the LIF model with the lightweight advantages of group convolutions and channel shuffle, we propose a novel network architecture unit, SpikingShuffle Unit, specifically designed for edge deployment, as shown in Figure 5. Figure 5a represents the Standard SpikingShuffle Unit, while Figure 5b represents the Downsampling SpikingShuffle Unit.

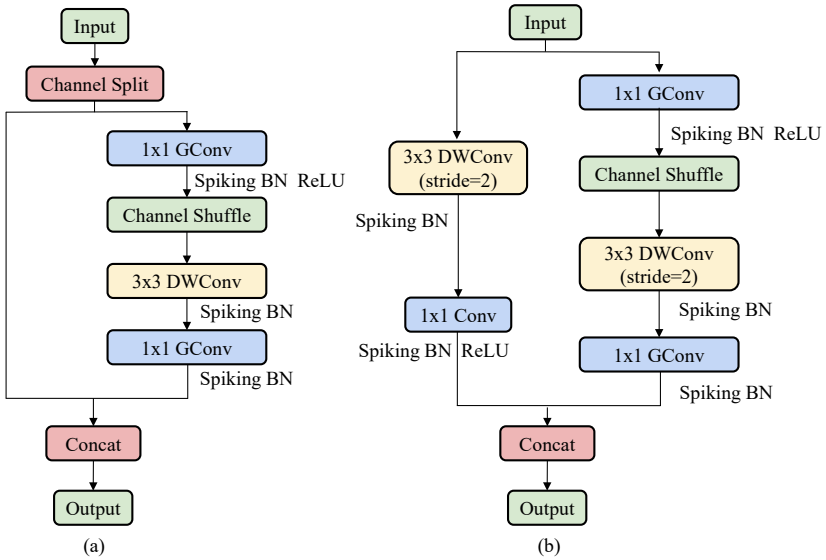

Figure 5. The structure of SpikingShuffle Unit: (**a**) The standard SpikingShuffle unit structure. (**b**) The downsampling SpikingShuffle unit structure.

At the beginning of each network unit, the input feature channels are divided into two branches. In Figure 5a, one half of the feature channels pass directly through the unit and are transmitted to the next unit. This design reflects the concept of feature reuse, similar to the ideas behind DenseNet and CondenseNet. In the other half of the feature channels, 1 × 1 grouped convolutions combined with the channel shuffle operation are first applied to reduce the number of channels, thereby decreasing both computational complexity and parameter count. Next, a 3 × 3 depth-wise separable convolution is performed, followed by another 1 × 1 grouped convolution and channel shuffle operation to restore the channel count of the feature map without incurring additional computational overhead. After the convolution operations, the outputs of the two branches are concatenated, ensuring that the number of channels remains unchanged. In Figure 5b, for spatial downsampling modules, the unit structure is slightly adjusted, and instead of channel splitting, each branch applies a stride = 2 downsampling operation. As a result, after the concatenation of the two branches, the spatial dimensions are halved, but the number of channels is doubled.

The SpikingShuffle Unit is an innovative component designed to enhance the efficiency of deep neural networks, particularly in scenarios where both computational efficiency and the ability to capture complex interdependencies across channels are critical. By integrating the principles of SNNs with ShuffleNet's channel shuffle mechanism, the SpikingShuffle Unit not only improves the representation of information flow but also achieves better learning performance, particularly when dealing with tasks that involve temporal or sequential data.

The overall network architecture draws inspiration from current popular lightweight architectures, such as GhostNET and MobileNet, and has been tailored to meet the real-time requirements of fault detection in electric vehicle power converters by reducing the depth and complexity of the network. Specific architectural details can be found in Table 1.

Table 1. The network structure of MLCA-SSN.

Input	Operator	#exp	#out	MLCA	Stride
$64^2 \times 3$	Conv2d	-	16	-	2
$32^2 \times 16$	SpikingShuffle Unit	48	32	-	2
$16^2 \times 32$	SpikingShuffle Unit	96	32	-	1
$16^2 \times 32$	SpikingShuffle Unit	96	64	1	2
$8^2 \times 64$	SpikingShuffle Unit	128	64	-	1
$8^2 \times 64$	SpikingShuffle Unit	320	128	1	2
$4^2 \times 128$	SpikingShuffle Unit	640	128	-	1
$4^2 \times 128$	SpikingShuffle Unit	640	128	-	1
$4^2 \times 128$	SpikingShuffle Unit	640	128	-	-
$4^2 \times 128$	Spiking Conv2d	-	640	-	1
$4^2 \times 640$	Avgpool	-	-	-	-
$1^2 \times 640$	Spiking Conv2d	-	1280	-	1
$1^2 \times 1280$	Spiking-FC	-	10	-	-

The network adopts a modular design composed of multiple SpikingShuffle Units. In the initial stages, standard convolutional layers are used to extract basic features from the input image. As the network deepens, the SpikingShuffle Units are stacked layer by layer, gradually increasing the number of feature channels and progressively building more complex high-level features, enabling the network to capture intricate patterns within the image.

To effectively control computational complexity in the early stages of the network, small convolutional kernels and fewer channels are employed. This design ensures that the network processes the input data efficiently at the outset, significantly reducing computational overhead. As the network deepens, the number of feature channels increases, enhancing the network's expressive power and enabling it to more accurately identify and distinguish subtle differences within the image.

Each SpikingShuffle Unit introduces grouped convolutions and channel shuffle techniques combined with the spiking transmission model. This approach not only maintains the network's lightweight nature but also improves the flow of information and the efficiency of feature extraction. Grouped convolutions partition the input channels into several small groups, effectively reducing the computational load, while channel shuffling enhances interchannel information transfer across different groups, avoiding bottlenecks in feature extraction.

At the final stage of the network, a global average pooling layer is used to transform the high-dimensional features into a low-dimensional feature vector, which is then processed by a fully connected layer for final classification. This design ensures low computational complexity while enhancing the network's performance and accuracy in complex tasks.

2.4. Lightweight Feature Enhancement Mechanism–Mixed Local Channel Attention

In this study, the MLCA mechanism is selectively integrated into certain SpikingShuffle Units. While the SpikingShuffle Unit design significantly reduces computational complexity through channel shuffling, it may inherently limit the extraction and differentiation of detailed features. To address this limitation, MLCA is introduced to enhance the sensitivity and accuracy of critical feature detection during fault diagnosis.

Conventional channel attention methods, such as SE and ECA, primarily focus on channel relationships while neglecting the spatial information within individual channels. This limitation may lead to the omission of critical features or inaccurate classification when dealing with complex fault patterns. For instance, when the attention weight of a specific channel is low, vital local features within that channel may be overlooked,

potentially causing critical fault information to go undetected. In contrast, MLCA leverages a combination of local and global attention mechanisms to effectively capture intricate local feature details while integrating global context, thereby significantly improving the recognition of complex fault patterns [27].

Specifically, MLCA processes the input feature maps through local average pooling and global average pooling, enabling the capture of diverse features across multiple scales and providing comprehensive informational support. This capability makes MLCA particularly suitable for deployment in SpikingShuffle Unit modules, where the integration of channel and spatial attention mechanisms at both local and global levels effectively enhances model precision.

The principles of MLCA are illustrated in the accompanying Figure 6. The input feature map (C, W, H) first undergoes local average pooling, as shown in Figure 7. By applying average pooling only within local regions, local average pooling preserves more localized feature information, enabling a better focus on regional characteristics while significantly reducing computational cost. Subsequently, the input features are divided into two branches for processing. The first branch primarily captures global information, while the second branch retains local spatial information.

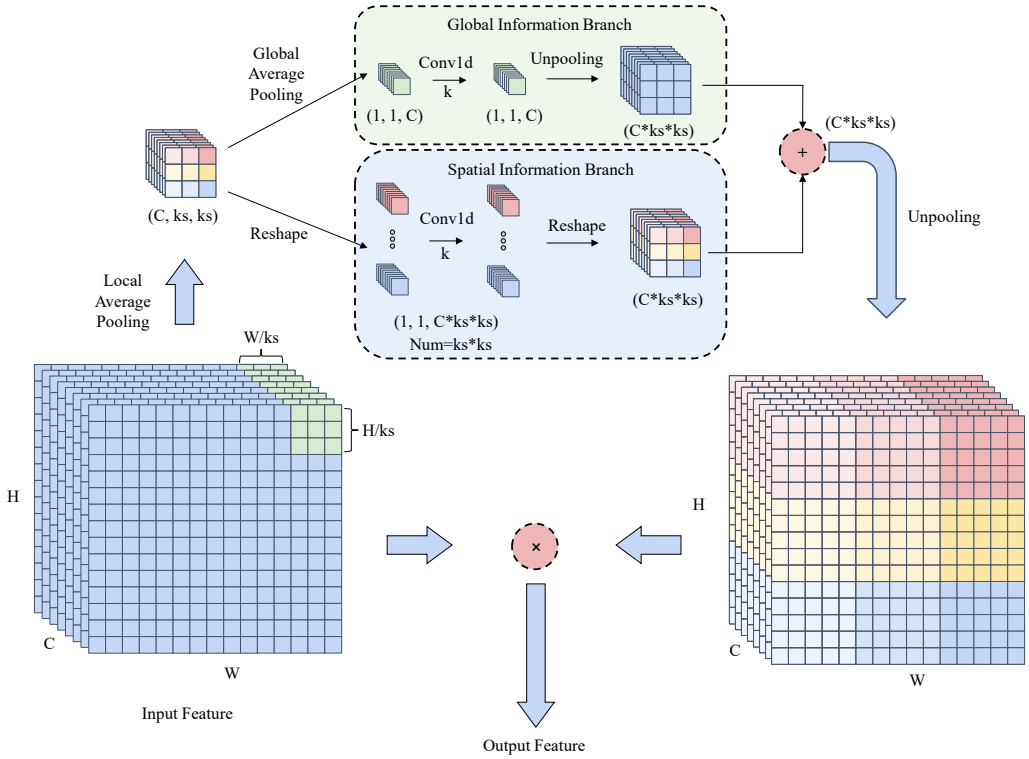

Figure 6. The structure of Mixed Local Channel Attention.

In the first branch, global average pooling is applied to compute the average of all values within each channel of the input feature map, as shown in Figure 7. Global average pooling extracts global features, reducing computational complexity and parameter count, and enhances the model's generalization capability. Following this, a one-dimensional convolution compresses the feature channels, effectively reducing the channel count while maintaining spatial dimensions, thus minimizing model parameters. Finally, unpooling restores the features to their original spatial dimensions, preparing them for subsequent operations. The unpooling process is also illustrated in Figure 7.

The second branch focuses on preserving local spatial information. The feature map resulting from local average pooling is reshaped into multiple one-dimensional data ar-

rays, which are then subjected to one-dimensional convolution for channel compression. Subsequently, unpooling restores the original spatial dimensions, ensuring support for subsequent processing.

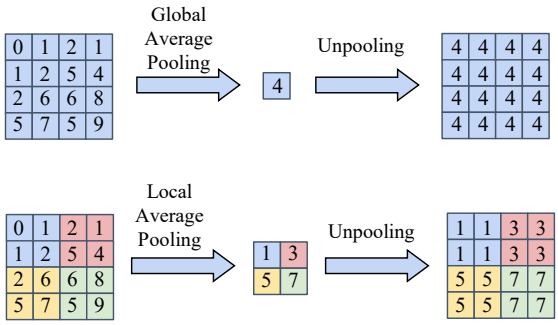

Figure 7. The comparison of global and local average pooling with unpooling operations.

Ultimately, the feature maps generated by the two branches are fused with the original input features. This fusion not only combines global and local information but also retains the spatial characteristics of the original features. Overall, the MLCA module enhances computational efficiency while improving the model's ability to capture critical features. By integrating channel attention and spatial attention mechanisms at both local and global levels, MLCA significantly enhances the network's feature representation and model precision.

In the combined architecture of SNNs and ShuffleNet, MLCA provides an efficient and accurate solution. Its low computational complexity and strong feature extraction capabilities enable outstanding performance in fault detection for electric vehicle converters. In summary, the introduction of MLCA markedly improves detection accuracy and robustness while meeting the demands for real-time performance and computational efficiency, offering robust support for complex fault detection tasks.

3. Experimental Analysis

This section provides a detailed description of the experimental design and implementation process, including dataset construction, evaluation metrics, experimental setup, and analysis of the results. Through ablation studies and comparisons with various lightweight models, the effectiveness of the proposed method in terms of computational efficiency and fault detection accuracy is validated.

3.1. Dataset Construction

The process of dataset construction is as follows: First, data are collected using a Keysight InfiniiVision DSOX3024T oscilloscope, ensuring high measurement accuracy, which is crucial for analyzing subtle changes in converter performance. The experimental circuit selected is a commonly used boost converter, widely applied in power supply modules for various electric vehicle devices, making it highly representative. We extended our analysis to other common DC–DC converter topologies, including boost, buck-boost, Cuk, and super buck converters. It was observed that the output stages of these converters share a similar structure and can be modeled as current source circuits. The primary difference lies in the source of the current delivered to the output capacitor during each switching cycle. Therefore, the conclusions drawn regarding output voltage ripple are applicable across various converter topologies, indicating that the total output voltage ripple reflects the characteristics of internal components and can serve as an indicator for fault diagnosis.

For degradation simulation, aluminum electrolytic capacitors are chosen as the experimental objects, as their performance directly affects the stability and efficiency of the circuit. Over time, these capacitors undergo aging phenomena such as a reduction in capacitance and an increase in equivalent series resistance. These degradation characteristics lead to an increase in output voltage ripple, a reduction in filtering capability, and, consequently, a deterioration in overall system performance. Moreover, the aging characteristics of aluminum electrolytic capacitors are easy to observe and quantify. The changes in capacitance and equivalent series resistance can be precisely measured using testing instruments, providing clear data support for fault diagnosis and degradation modeling. Therefore, studying the aging characteristics of aluminum electrolytic capacitors lays a foundation for further analysis of the impact of device aging on circuit performance as well as the development of efficient fault detection and parameter identification methods.

To simulate the actual aging and performance degradation process of the power converter, voltage ripple data were first collected under normal operating conditions as baseline samples representing the healthy state. Subsequently, to ensure the dataset accurately reflects the real conditions of the DC–DC converter at various aging stages, a thermal aging experiment was conducted based on the Arrhenius thermal aging model. The Arrhenius equation is expressed as follows:

$$k = A \cdot e^{-\frac{E_a}{R \cdot T}} \tag{5}$$

where k represents the reaction rate constant, A is the pre-exponential factor, E_a denotes the activation energy, R is the universal gas constant (8.314 J/mol·K), and T is the temperature. This equation allows the estimation of component degradation time under varying temperature conditions. The primary aging mechanisms of capacitors, such as electrolyte evaporation, electrode corrosion, and insulation material degradation, are typical thermally activated reactions, which the Arrhenius model effectively describes by quantifying how the reaction rate changes with temperature variations. The Arrhenius model establishes a clear relationship between the reaction rate constant (aging rate) and temperature, providing a quantifiable measure of how increased temperature accelerates the aging rate of components. This is particularly crucial for accelerated aging tests conducted under laboratory conditions, as the model allows the estimation of degradation time under elevated temperatures while maintaining the validity of the results for lower-temperature operational conditions. Therefore, it serves as a reliable tool for studying capacitor aging and failure behaviors.

To simulate the actual aging and performance degradation process of power converters, voltage ripple data are first collected under normal operating conditions as baseline samples for healthy status. Then, through thermal chamber aging experiments, electrolytic capacitors with varying degradation levels are obtained, and capacitors with different resistance values are systematically swapped into the circuit to simulate aging and fault scenarios. Voltage ripple data are then collected under these conditions as fault state samples. This method covers a wide range of performance changes in the capacitor, from healthy to degraded states, accurately simulating real-world aging and fault scenarios. The collected data directly reflect the characteristics of different aging stages. This dataset not only includes voltage ripple data from the healthy state as a reference but also includes ripple variation characteristics that manifest with increasing aging. The dataset is comprehensive and representative, providing reliable data support for training and validating fault diagnosis models, thus meeting the needs of the experiment.

The collected raw data undergo preprocessing to enhance quality and usability. Key steps include data cleaning and normalization, continuous wavelet transform, and data augmentation. The continuous wavelet transform can reveal the characteristics of nons-

mooth signals across various time and frequency scales. Unlike traditional signal analysis methods, CWT effectively captures localized features within a signal. Its core principle involves convolving the signal with a series of scaled and shifted wavelet basis functions, resulting in a time–frequency representation of the signal. Data cleaning and normalization remove noise and outliers, while standardization ensures consistency across the dataset.

The aging state of capacitors is characterized by variations in the range of output voltage ripple. Faults are defined as deviations of the actual capacitance value from the nominal value exceeding 20%, and the circuit is categorized based on the degree of deviation in the voltage ripple. Circuits with voltage ripple deviations of less than 2% are considered to be in a healthy state, while deviations between 2% and 20% indicate potential degradation. These latent fault states are further divided into nine health levels. Output signals were collected at 10 ms intervals, and continuous wavelet transform (CWT) was applied to convert the signals into time–frequency domain images, forming the dataset. Each health level generated 1000 time–frequency images, resulting in a dataset of 10,000 images. This fine-grained classification allows for more precise performance monitoring and fault detection. The dataset was conducted using time-based split and stratified sampling strategies. The time-based split simulates real-world data prediction sequences, while stratified sampling ensures that both the training and testing sets maintain a consistent class distribution, improving model generalization and reliability. The dataset is divided into training (60%), validation (20%), and testing (20%) sets to ensure adequate training data and robust evaluation of model generalization during the testing phase.

3.2. Evaluation Metrics

In this study, we employ the following evaluation metrics to comprehensively assess the performance of the model: million floating-point operations per second (MFLOPs), million model parameters (MParams), accuracy, precision, recall, and macro-F1 score [28]. These widely recognized metrics are chosen to evaluate the model's performance from multiple dimensions, allowing for a scientific and objective comparison of different models. Their definitions are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{6}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{F1-score}_i = 2 \times \frac{\text{Recall}_i \times \text{Precision}_i}{\text{Recall}_i + \text{Precision}_i} \tag{9}$$

$$\text{Macro-F1} = \frac{\text{F1-score}_1 + \text{F1-score}_2 + \cdots + \text{F1-score}_n}{n} \tag{10}$$

$$\text{FLOPs} = k_h \times k_w \times \frac{C_{in}}{g} \times C_{out} \tag{11}$$

$$\text{Params} = \left(2 \times k_h \times k_w \times \frac{C_{in}}{g} - 1\right) \times C_{out} \times H_{out} \times W_{out} \tag{12}$$

where TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives in the test images, respectively. C_{in} represents the number of input channels, C_{out} represents the number of output channels, and $k_w \times k_h$ denotes the size of the convolution kernel, with k_h and k_w being the height and width of the kernel, respectively. H_{out} and W_{out} are the height and width of the output feature map.

These evaluation metrics assess the model's performance from various angles, serving the following specific purposes: MFLOPs and MParams evaluate the computational efficiency and resource consumption of the model, helping to determine its suitability for deployment in resource-constrained environments. Accuracy measures the overall correctness of the model's classification, suitable for scenarios where class distribution is balanced. Precision focuses on false positives, making it useful in applications where minimizing the misclassification of negative samples as positive is crucial. Recall emphasizes false negatives, making it suitable for scenarios where maximizing the detection of positive samples and avoiding missed diagnoses is critical. Macro-F1 score combines precision and recall performance, particularly useful for imbalanced class distributions, ensuring balanced performance across different classes. In summary, these metrics provide a comprehensive evaluation of both the computational efficiency and classification ability of the model, aiding in the overall assessment and optimization of its performance across various application scenarios.

Through the combined evaluation of these metrics, this research ensures that both the model's classification performance and its computational efficiency and resource usage are taken into account. This holistic evaluation approach helps identify models that possess both high computational efficiency and the ability to perform high-quality fault detection, ensuring that high accuracy is achieved while meeting resource constraints in practical applications. This clear and comprehensive measurement framework enhances the reliability and applicability of the research conclusions.

3.3. Experimental Setup and Edge Deployment

This section outlines the various parameter settings and hardware components used in the experiments. The model training and testing were performed on a platform equipped with an NVIDIA GeForce RTX 3090 GPU and an Intel Core i9-12900K CPU, ensuring efficient computational performance. The experiments were conducted on a Linux operating system with the development environment based on the PyTorch 2.0-GPU framework. PyTorch was chosen for its robust support for deep learning libraries and efficient GPU acceleration, ensuring both training efficiency and stability on large-scale datasets. The model deployment was carried out on the NVIDIA Jetson Nano, demonstrating the algorithm's potential for real-time in situ detection. During training, the same hyperparameters, including the number of training epochs, batch size, and learning rate, were used to ensure stability and efficiency in the training process. Specifically, the Adaptive Moment Estimation optimizer was employed during the training process with an initial learning rate set to 0.0002, enabling dynamic adjustment for more stable and efficient gradient descent. Furthermore, the model was trained for 50 epochs with a batch size of 32, balancing computational resources and training stability, ensuring effective model convergence while maximizing GPU utilization. Such settings also guarantee the fairness of the comparison, ensuring that the evaluation of different models is based on consistent conditions.

This article selects several well-known and widely used lightweight models, such as MobileNetV3 [29] and ShuffleNet [30]. Among them, some of these models come in different versions and sizes. These models are widely applied in mobile devices and embedded systems due to their efficient computational characteristics and relatively low memory requirements. To ensure comparability of computational complexity across models, we categorized them based on their floating-point operations (FLOPs) into three complexity levels: low-complexity models, medium-complexity models, and high-complexity models. This classification enables a systematic comparison of performance differences among models with varying computational demands under similar computational workloads. Within each complexity level, several representative models were selected to ensure appropriate

comparisons at every level. This grouping strategy not only maintains relative balance in computational resource requirements but also highlights the performance differences of models across varying computational complexities in practical tasks.

Furthermore, to ensure fairness and consistency in the comparison, all models were trained, validated, and tested on the same custom voltage time–frequency image dataset. By using identical datasets and experimental conditions, we eliminated any bias that might arise from dataset variations, ensuring that the comparisons between models were made on a consistent basis, providing a reliable foundation for performance evaluation.

In the in situ detection deployment experiment, the lightweight health assessment algorithm was successfully integrated into the NVIDIA Jetson Nano edge computing platform. The experimental setup, as shown in Figure 8, includes various experimental devices such as an oscilloscope, digital multimeter, LCR meter, electronic load, power supply, NI data acquisition module, NVIDIA Jetson Nano platform, and experimental circuit module. The NVIDIA Jetson Nano, with its compact hardware design and flexible deployment capabilities, enables the efficient integration of the algorithm into real-world applications, allowing for real-time monitoring of the converter's operating status and in situ fault detection. Its lightweight nature not only ensures low power consumption but also meets high-performance requirements, making it suitable for real-time fault detection in various power converters within electric vehicles. During the experiment, devices such as the digital multimeter, LCR meter, and oscilloscope were used for performance monitoring, debugging, and verification of the circuit. This comprehensive experimental platform spans the entire process, from algorithm training and testing to actual deployment, ensuring the reliability of the experimental results and the practical applicability of the algorithm while demonstrating its feasibility and adaptability in industrial scenarios.

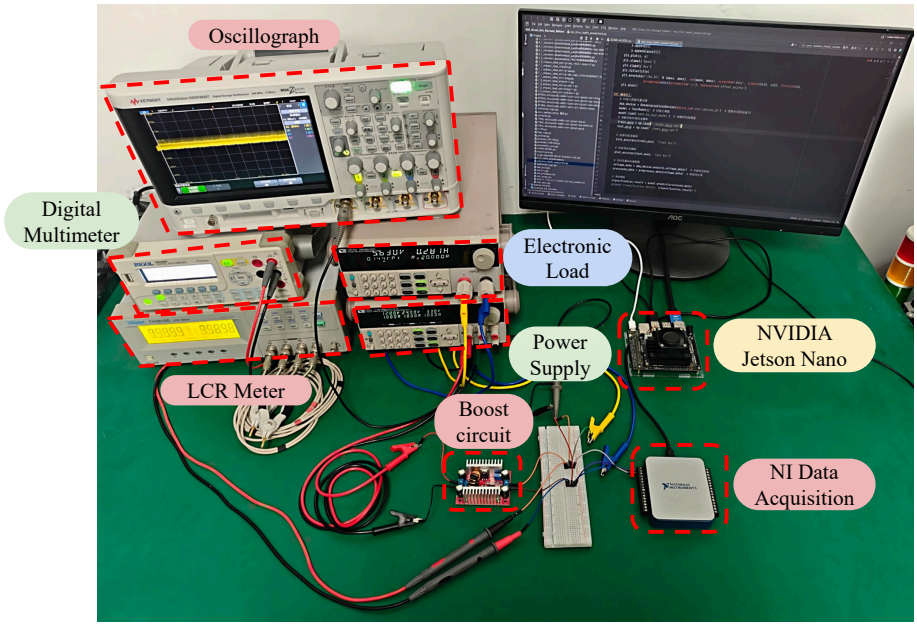

Figure 8. The data acquisition and edge deployment testing platform.

3.4. Ablation Experiments

To thoroughly investigate the impact of different components on model performance, a series of ablation studies were conducted to analyze the effects of the LIF model, channel shuffle, and MLCA modules on the model's performance. The experimental data are shown in Table 2.

Table 2. The impact of different components on model performance.

LIF Model	Channel Shuffle	MLCA	Params (M)	FLOPs (M)	Acc (%)
✗	✗	✗	4.90	15.80	97.12
✓	✗	✗	5.01	14.30	96.86
✗	✓	✗	4.98	14.15	97.02
✗	✗	✓	5.07	16.80	98.86
✓	✓	✓	5.11	15.76	98.61

Firstly, when the LIF model is applied alone, the computational load is significantly reduced, but there is a slight drop in accuracy. This indicates that while the LIF model effectively optimizes computational efficiency, its contribution to accuracy improvement is limited. After introducing channel shuffle, the computational load decreases slightly, and accuracy sees a minor improvement, demonstrating its role in optimizing computational efficiency and enhancing feature flow, although the accuracy gain is relatively small. When MLCA is introduced alone, both the number of parameters and computational load increase significantly, yet the accuracy improvement is particularly pronounced. This shows that MLCA, by focusing more on important features, significantly enhances the model's performance despite the increased computational cost. When all three components are combined, the SpikingShuffle Unit and MLCA work synergistically, enhancing feature extraction efficiency and reducing the reliance on large feature maps, thereby lowering computational demands. Although there is a slight reduction in accuracy compared to using MLCA alone, this combination achieves a favorable balance between accuracy and computational cost while preserving essential feature representation capabilities. Despite the increase in the number of parameters due to the addition of these modules, there is no notable impact on deployment to edge platforms. This validates that the proposed fusion algorithm achieves an excellent balance between accuracy and computational load.

3.5. Model Comparison Experiment

In this study, to evaluate the practical performance of the proposed method within lightweight models, we conducted a comprehensive comparison with several mainstream lightweight networks. The experimental results are presented in Table 3. Through performance evaluation across different complexity levels, it is clear that the proposed method demonstrates outstanding advantages in multiple aspects.

Firstly, in small-scale models with limited computational resources, the proposed method achieves a classification accuracy of 97.51% with a low parameter count and computational load thanks to its efficient network design. Additionally, it outperforms other lightweight models in metrics such as precision, recall, and macro-F1 score. Through structural optimization, this method maintains highly competitive performance in resource-constrained environments, demonstrating its significant potential for applications in edge computing devices.

In medium-scale models, the proposed method further showcases outstanding performance with its efficiency. With a computational demand of only 29.55M FLOPs, it achieves a classification accuracy of 99.76%. In contrast, other models with similar performance, such as MobileNeXt, typically require more than 30M FLOPs to achieve comparable results. This clearly demonstrates that the proposed method maintains high effectiveness and excellent performance even in resource-limited scenarios.

For higher-complexity models, the designed network continues to exhibit superior computational efficiency and performance. With a total computational demand of only 41.88M FLOPs, its classification accuracy approaches 100%. Even under higher-complexity

conditions, the method leverages more efficient resource utilization to achieve exceptional performance.

Table 3. The comparison results among different methods.

Model	Params (M)	FLOPs (M)	Acc (%)	Precision (%)	Recall (%)	Macro-F1 (%)
MobileNetV1 0.5x	1.34	16.28	91.23	91.22	91.24	91.23
ShuffleNetV1 1.0x	1.89	14.97	93.89	93.86	93.88	93.87
ShuffleNetV2 1.0x	2.33	15.84	95.17	95.18	95.16	95.17
MobileNetV2 0.6x	5.18	15.30	97.41	97.36	97.48	97.42
MobileNetV3-L 0.75x	4.00	16.82	97.48	97.38	**97.54**	97.47
MLCA-SSN 1.0x	2.33	15.76	**97.51**	**97.56**	97.46	**97.51**
MobileNetV2 1.0x	3.46	32.55	96.95	96.94	96.96	96.95
FE-Net 1.0x	3.66	32.66	96.45	96.41	96.49	96.45
ShuffleNetV2 1.5x	3.50	32.44	97.25	97.28	97.22	97.25
FBNet-B	4.53	32.01	98.17	98.21	98.13	98.17
ProxylessNAS	4.13	34.72	98.38	98.37	98.39	98.38
MobileNetV3-L 1.0x	5.44	23.76	98.42	98.41	98.43	98.42
MobileNeXt 1.0x	3.91	35.81	99.48	**99.59**	99.35	99.47
MLCA-SSN 1.5x	4.47	29.55	**99.76**	99.58	**99.90**	**99.75**
MobileNetV1 1.0x	4.21	62.39	95.48	95.47	95.49	95.48
MobileViT-XS	2.33	75.95	98.08	98.11	98.05	98.08
FBNet-C	5.49	40.69	98.18	98.19	98.17	98.18
MnasNet-A2	4.81	36.89	98.98	98.96	99.00	98.98
EfficientNet	5.32	42.32	99.50	99.52	99.48	99.50
MnasNet-A3	5.21	43.73	99.51	99.48	99.50	99.49
MobileNeXt+ 1.1x	4.31	45.57	99.34	99.34	99.34	99.34
MLCA-SSN 2.0x	5.91	42.18	**99.98**	**99.96**	**99.99**	**99.97**

Values presented in **bold** denote the highest performance in each metric, while underlined values signify the second highest performance.

Although the proposed network's parameter count and computational load are slightly inferior to a few specially optimized models, this is mainly due to those models' excessive optimization of computational load at the expense of accuracy. In contrast, the introduction of the MLCA mechanism, along with depth-wise convolutions and channel shuffle designs, significantly enhances the network's representational power. These innovative structures improve both performance and generalization while keeping the increase in computational complexity within reasonable bounds. The algorithm achieves a substantial increase in accuracy with minimal computational overhead, striking an excellent balance between accuracy and computational load, which sets it apart from other models in its class.

In conclusion, the proposed network demonstrates significant advantages in lightweight design, excelling in resource efficiency, performance improvement, and applicability. Whether in low-complexity or high-complexity application scenarios, this method achieves outstanding results in key evaluation metrics such as classification accuracy, recall, and macro-level indicators while significantly reducing the demand for computational resources. These results fully validate the efficiency and practicality of the proposed method, highlighting its competitive edge and potential value in real-world deployments.

4. Conclusions

This study proposes an innovative lightweight fault diagnosis method, introducing the SpikingShuffle Unit network module, which combines deep convolution and channel shuffle techniques with spike-based information transmission, effectively reducing the

model's computational complexity. Building upon this, a complete lightweight diagnostic network was constructed by strategically stacking SpikingShuffle modules and optimizing the network architecture. Furthermore, the Mixed Local Channel Attention mechanism was incorporated to address potential information dilution caused by grouped convolutions and channel shuffle. By leveraging the synergy between local and global feature extraction, the Mixed Local Channel Attention mechanism effectively balances local detail preservation with global feature integration, further enhancing the accuracy and robustness of fault detection.

Experimental results demonstrate that the SpikingShuffNet excels in converter fault detection, accurately capturing the impact of component aging on output voltage ripple. This performance highlights its substantial practical application value. The approach presented in this study offers a new technological pathway for real-time fault detection in electric vehicle power converters and provides theoretical support for the development of low-power, high-performance edge computing algorithms. Future work will focus on extending this method to a broader range of power electronic system health monitoring and predictive maintenance scenarios to meet the demands of more complex and dynamic industrial applications.

Author Contributions: Visualization, Y.Y.; writing—original draft, L.W. and F.Z.; writing—review and editing, F.Z. and F.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62103205.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Virmani, N.; Agarwal, V.; Karuppiah, K.; Agarwal, S.; Raut, R.D.; Paul, S.K. Mitigating barriers to adopting electric vehicles in an emerging economy context. *J. Clean. Prod.* **2023**, *414*, 137557. [CrossRef]
2. Acharige, S.S.G.; Haque, M.E.; Arif, M.T.; Hosseinzadeh, N.; Hasan, K.N.; Oo, A.M.T. Review of Electric Vehicle Charging Technologies, Standards, Architectures, and Converter Configurations. *IEEE Access* **2023**, *11*, 41218–41255. [CrossRef]
3. Leal, W.C.; Godinho, M.O.; Bastos, R.F.; de Aguiar, C.R.; Fuzato, G.H.F.; Machado, R.Q. Cascaded Interleaved DC–DC Converter for a Bidirectional Electric Vehicle Charging Station. *IEEE Trans. Ind. Electron.* **2024**, *71*, 3708–3717. [CrossRef]
4. Wouters, H.; Martinez, W. Bidirectional Onboard Chargers for Electric Vehicles: State-of-the-Art and Future Trends. *IEEE Trans. Power Electron.* **2024**, *39*, 693–716. [CrossRef]
5. Zhao, Y.; Geng, L.; Shan, S.; Du, Z.; Hu, X.; Wei, X. Review of sensor fault diagnosis and fault-tolerant control techniques of lithium-ion batteries for electric vehicles. *J. Traffic Transp. Eng. (Engl. Ed.)* **2024**, *11*, 1447–1466. [CrossRef]
6. Shao, N.; Chen, Q.; Dong, Y.; Ding, W.; Wang, L. Power System Fault Diagnosis Method Based on Intuitionistic Fuzzy Sets and Incidence Matrices. *IEEE Trans. Power Deliv.* **2023**, *38*, 3924–3938. [CrossRef]
7. Peng, F.; Jiang, S.; Zhao, Y.; Ren, L. Soft Short-Circuit Fault Diagnosis for Vehicular Battery Packs With Interpretable Full-Dimensional Statistical Analytics. *IEEE Trans. Power Electron.* **2024**, *39*, 11650–11664. [CrossRef]
8. Sravani, V.; Venkata, S.K. Detection of Sensor Faults with or without Disturbance Using Analytical Redundancy Methods: An Application to Orifice Flowmeter. *Sensors* **2023**, *23*, 6633. [CrossRef] [PubMed]
9. Zanuso, G.; Kumar, S.L.S.; Peretti, L. Interturn Fault Detection in Induction Machines Based on High-Frequency Injection. *IEEE Trans. Ind. Electron.* **2023**, *70*, 10639–10647. [CrossRef]
10. Jiang, H.; Wu, Y.; Yuan, J.; Zhao, Q.; Chen, J. Adaptive Low-Rank Tensor Estimation Model Based Multichannel Weak Fault Detection for Bearings. *Sensors* **2024**, *24*, 3762. [CrossRef] [PubMed]
11. Santamaria-Bonfil, G.; Arroyo-Figueroa, G.; Zuniga-Garcia, M.A.; Azcarraga Ramos, C.G.; Bassam, A. Power Transformer Fault Detection: A Comparison of Standard Machine Learning and autoML Approaches. *Energies* **2024**, *17*, 77. [CrossRef]
12. Zhang, Z.; Dong, S.; Li, D.; Liu, P.; Wang, Z. Prediction and Diagnosis of Electric Vehicle Battery Fault Based on Abnormal Voltage: Using Decision Tree Algorithm Theories and Isolated Forest. *Processes* **2024**, *12*, 136. [CrossRef]
13. Zhang, J.; He, X. Compound-Fault Diagnosis of Integrated Energy Systems Based on Graph Embedded Recurrent Neural Networks. *IEEE Trans. Ind. Informat.* **2024**, *20*, 3478–3486. [CrossRef]

14. Rokocakau, S.; Riccio, J.; Tresca, G.; Kumar, R.R.; Cirrincione, G.; Zanchetta, P.; Cirrincione, M. Fault Diagnosis Using Shallow Neural Networks for Voltage Source Inverters in Motor Drives. *IEEE Trans. Ind. Appl.* **2024**, *60*, 7038–7047. [CrossRef]
15. Alshehri, A.; Owais, M.; Gyani, J.; Aljarbou, M.H.; Alsulamy, S. Residual neural networks for origin–destination trip matrix estimation from traffic sensor information. *Sustainability* **2023**, *15*, 9881. [CrossRef]
16. Moussa, G.S.; Owais, M. Pre-trained deep learning for hot-mix asphalt dynamic modulus prediction with laboratory effort reduction. *Constr. Build. Mater.* **2020**, *265*, 120239. [CrossRef]
17. Wu, P.; Tian, E.; Tao, H.; Chen, Y. Data-driven spiking neural networks for intelligent fault detection in vehicle lithium-ion battery systems. *Eng. Appl. Artif. Intell.* **2025**, *141*, 109756. [CrossRef]
18. Zhou, Y.; Zhang, A. Improved Integrate-and-Fire Neuron Models for Inference Acceleration of Spiking Neural Networks. *Appl. Intell.* **2021**, *51*, 2393–2405. [CrossRef]
19. Skocik, M.J.; Long, L.N. On the Capabilities and Computational Costs of Neuron Models. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 1474–1483. [CrossRef] [PubMed]
20. Kim, J.; Kim, H.; Huh, S.; Lee, J.; Choi, K. Deep Neural Networks with Weighted Spikes. *Neurocomputing* **2018**, *311*, 373–386. [CrossRef]
21. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
22. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. *arXiv* **2021**, arXiv:2104.00298.
23. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2017**, arXiv:1610.02357.
24. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Deep Feature Based Rice Leaf Disease Identification Using Support Vector Machine. *Comput. Electron. Agric.* **2020**, *175*, 105527. [CrossRef]
25. Raza, A.; Ayub, H.; Khan, J.A.; Ahmad, I.; S. Salama, A.; Daradkeh, Y.I.; Javeed, D.; Ur Rehman, A.; Hamam, H. A Hybrid Deep Learning-Based Approach for Brain Tumor Classification. *Electronics* **2022**, *11*, 1146. [CrossRef]
26. Chen, Z.; Yang, J.; Chen, L.; Jiao, H. Garbage Classification System Based on Improved ShuffleNet v2. *Resour. Conserv. Recycl.* **2022**, *178*, 106090. [CrossRef]
27. Wan, D.; Lu, R.; Shen, S.; Xu, T.; Lang, X.; Ren, Z. Mixed Local Channel Attention for Object Detection. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106442. [CrossRef]
28. Rainio, O.; Teuho, J.; Klén, R. Evaluation Metrics and Statistical Tests for Machine Learning. *Sci. Rep.* **2024**, *14*, 6086. [CrossRef] [PubMed]
29. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244.
30. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *arXiv* **2017**, arXiv:1707.01083.