energies

MDPI

Article

Android-Based Real-Time Classification of Electric Fire Short-Circuit Traces Using Lightweight Deep Learning Model

Mohammad Hadi Nazari and Junho Bang *

Department of IT Applied Engineering, Jeonbuk National University, Jeonju 54896, Republic of Korea; hadinazari@jbnu.ac.kr
* Correspondence: jhbang@jbnu.ac.kr

Abstract

This paper presents a lightweight deep learning framework for classifying electric fire short-circuit traces to enhance safety and fault diagnosis in electrical energy systems. Accurate differentiation between primary (PSCT) and secondary short-circuit traces (SSCT) is essential for identifying failure origins, yet conventional manual inspection is time-consuming and subjective. To address these limitations, we systematically evaluate three lightweight convolutional neural network (CNN) architectures MobileNetV2, MobileNetV3, and EfficientNet using transfer learning on a domain-specific image dataset. The models are assessed based on accuracy, loss, precision, recall, and F1-score. Experimental results show that EfficientNet achieves the highest classification accuracy, while MobileNetV3 demonstrates the lowest validation loss and superior generalization stability. Based on a performance–efficiency trade-off analysis, MobileNetV3 is deployed on an Android platform using TensorFlow Lite, enabling real-time, offline, and on-device inference. To the best of our knowledge, this is among the first studies to integrate lightweight CNN-based short-circuit trace classification with real-time mobile deployment for on-site energy system fault analysis. By bridging the gap between deep learning and field deployment, the proposed mobile system ensures low-latency execution and provides a rapid, reliable, and portable solution for improving operational safety in electrical fire investigations.

Keywords: electric fire investigation; energy system fault analysis; MobileNetV3; lightweight CNN; TensorFlow Lite; Android deployment; on-device inference; deep learning

1. Introduction

Deep learning has significantly transformed image-based analysis by enabling automatic feature extraction and high-accuracy classification across a wide range of safety-critical applications. Rather than relying on traditional feature engineering methods that often fall short in complex environments, convolutional neural networks (CNNs) excel at learning hierarchical and discriminative visual representations.

To fully understand the evolution of these vision systems, it is essential to examine the specific methods and benchmarks established in the recent literature. For instance, Wang et al. [1] developed a deep residual network architecture for medical image segmentation, demonstrating high sensitivity to structural deformations. Similarly, Kumar and Priyadarshi [2] implemented an optimized convolutional framework combined with automated denoising filters, which drastically reduced false-positive rates in fine-grained pixel classification. In the domain of network security and structural data processing, Zhang et al. [3] provided a foundation for handling extensive digital data processing

Academic Editor: Maria Founti

Received: 21 May 2026
Revised: 28 June 2026
Accepted: 2 July 2026
Published: 4 July 2026

pipelines, while Chen et al. [4] successfully utilized transfer learning on pre-trained deep models to automate quality inspection, demonstrating that high-level feature maps could be adapted to domain-specific datasets with minimal training overhead.

The underlying capability of these networks to adapt to diverse datasets stems from landmark advancements in visual pattern recognition and large-scale benchmarking. Wagobera et al. [5] demonstrated how machine learning software can optimize targeted classification tasks by mapping fine-grained input distributions. To train such networks effectively without overfitting, Deng et al. [6] introduced the ImageNet database, establishing a massive hierarchical visual framework that revolutionized how deep architectures learn generalized edge and texture features. Leveraging this dataset, Krizhevsky et al. [7] developed AlexNet, a pioneering deep CNN that utilized GPU acceleration and dropout regularization to achieve a breakthrough reduction in top 5 classification error rates. Building upon these principles, Zhao et al. [8] provided a comprehensive review of architectural optimizations in computer vision, emphasizing how deeper, residual paths stabilize gradient flow. Additionally, Li [9] explored the direct translation of these complex image recognition models into practical industrial frameworks, while Redmon and Farhadi [10] advanced real-time feature extraction by introducing the single-stage YOLOv3 framework, prioritizing execution speed without sacrificing structural localization. While these foundational studies underscore the exceptional capability of CNNs in capturing subtle structural variations, their methodologies rely on heavy, multi-layered architectures designed for high-performance computing clusters, making them highly impractical for immediate field deployment.

This methodological limitation becomes particularly evident when transitioning from general industrial inspection to the highly specialized field of electrical fire investigations. In this context, accurately identifying the origin of ignition remains a critical yet challenging task. A key aspect of forensic analysis involves distinguishing between primary short-circuit traces (PSCT), which indicate the true ignition source, and secondary short-circuit traces (SSCT), which are formed as a consequence of external fire exposure.

Primary short-circuit traces (PSCT) are generated when an electrical fault directly initiates ignition, typically characterized by localized melting, arc bead formation, and metallurgical features resulting from high current flow prior to fire exposure. In contrast, secondary short-circuit traces (SSCT) occur when external fire conditions degrade insulation, leading to subsequent electrical faults. These traces often exhibit irregular deformation, oxidation, and thermal damage influenced by ambient fire conditions rather than the original electrical event. This distinction is critical in fire investigation, as it determines whether the electrical system is the cause or the result of the fire. Conventionally, this distinction has relied heavily on manual inspection and expert qualitative interpretation, methods that are inherently subjective, time-consuming, and prone to inconsistency.

To automate this process, recent studies have attempted to apply deep learning to electrical fault and fire analysis. For example, Thai et al. [11] conducted a comprehensive assessment of electrical fire causes using multi-stage assessment algorithms, showing high accuracy in localized fault detection. Furthermore, Park et al. [12] introduced an arc fault detection algorithm on AC power lines by deploying deep neural networks that analyze localized voltage waveform distortions, successfully preventing potential ignitions in real-time laboratory settings. Additionally, Chen et al. [13] investigated the structural and combustion characteristics of electrical cables under variable external heat fluxes, providing data that links physical trace deformation to external thermal stress.

Recent advancements in electrical fire analysis and deep learning have significantly improved fault detection and classification capabilities. For instance, Thai et al. [11] provide a comprehensive survey of electrical fire cause assessment technologies, highlighting the effectiveness of data-driven and algorithmic approaches for identifying fire origins. Similarly, Park et al. [12] propose a deep learning-based method for detecting parallel arc faults in AC power systems, demonstrating high accuracy in real-time electrical fault detection. In addition, Chen et al. [13] investigate the physical characteristics of short-circuit and combustion behavior under varying thermal conditions, offering valuable insights into the formation mechanisms of electrical fire traces.

Despite these contributions, existing studies primarily focus on macro-level fault detection, signal-based monitoring, or physical analysis of combustion phenomena, rather than image-based forensic classification of post-fire traces. In particular, the critical task of distinguishing between primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT), which is essential for determining fire causality, remains largely underexplored in the current literature.

From a computational perspective, recent developments in edge AI and lightweight deep learning have enabled efficient on-device inference. Incel and Bursa [14] review on-device deep learning systems and emphasize their advantages in reducing latency and improving data privacy in mobile and embedded applications. Similarly, Verber et al. [15] demonstrate the feasibility of deploying lightweight deep learning models for image-based classification tasks in resource-constrained environments. Rizvi et al. [16] further highlights the importance of efficient deep learning models for real-time applications under limited computational resources, while Mittal [17] provides a comprehensive overview of lightweight object detection architectures optimized for edge devices.

However, despite these advancements, a significant research gap remains at the intersection of forensic-level electrical fire analysis and edge-based deep learning deployment. Existing lightweight models have been extensively evaluated on general-purpose datasets, but their applicability to domain-specific forensic image classification tasks, particularly for distinguishing PSCT and SSCT, has not been systematically investigated. Furthermore, most prior works either rely on laboratory-based analysis or cloud-dependent frameworks, limiting their usability in real-time field investigations where network access may be restricted.

Therefore, this paper aims to bridge this gap by developing a lightweight, accurate, and deployable deep learning framework for the real-time classification of short-circuit traces. By systematically evaluating MobileNetV2, MobileNetV3, and EfficientNet under domain-specific constraints and deploying the optimized model on an Android platform using TensorFlow Lite, this work extends existing research toward practical, on-site forensic applications.

The main contributions of this paper can be summarized as follows:

(1) Forensic-level classification of short-circuit traces: This paper addresses the underexplored problem of distinguishing primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT) using image-based deep learning, providing an objective and data-driven approach for electrical fire investigation.

(2) Development of a domain-specific dataset: A controlled experimental platform was designed to generate realistic short-circuit trace data under various electrical and thermal conditions, ensuring reproducibility and reliability of the dataset.

(3) Comparative evaluation of lightweight CNN architectures: The performance of MobileNetV2, MobileNetV3, and EfficientNet was systematically evaluated using transfer learning, analyzing their accuracy, generalization capability, and computational efficiency.

(4) On-device deployment for real-time inference: The optimized model was converted into TensorFlow Lite format and deployed on an Android platform, enabling real-time, offline classification without reliance on cloud infrastructure.

https://doi.org/10.3390/en19133184

2. Related Work and Background

2.1. Deep Learning in Fire and Electrical Fault Analysis

Deep learning techniques have been increasingly adopted in fire detection and electrical fault diagnosis due to their ability to automatically extract discriminative features from complex data. Early approaches in this domain primarily relied on handcrafted feature extraction combined with traditional machine learning algorithms such as support vector machines (SVMs) and random forests. Although these methods achieved moderate performance, they were inherently limited by their dependence on manually designed features and their sensitivity to environmental variations.

With the advancement of deep learning, convolutional neural network (CNN)-based approaches have significantly improved performance in fire-related image analysis tasks. CNNs enable hierarchical feature learning, allowing models to capture both low-level texture patterns and high-level semantic information. As a result, they have been widely applied in fire and smoke detection systems, demonstrating improved accuracy and robustness in complex environments. For example, Hasan et al. [18] proposed a deep learning framework (FireLite) that leverages transfer learning for efficient fire detection under resource-constrained conditions, achieving high accuracy while maintaining low computational cost. Thai et al. [11] presented a comprehensive survey of electrical fire cause assessment technologies, highlighting the role of data-driven approaches in improving fault diagnosis. Park et al. [12] developed a deep learning-based method for detecting parallel arc faults using electrical signal data, demonstrating high detection accuracy in controlled AC systems. Similarly, Chen et al. [13] investigated the physical and combustion characteristics of electrical cables under varying thermal conditions, providing important insights into the formation mechanisms of short-circuit traces. In addition to application-specific models, research on model efficiency has also contributed to the advancement of deep learning in real-world scenarios. Han et al. [19] introduced a deep compression technique that reduces model size through pruning, quantization, and encoding, significantly improving computational efficiency without substantial accuracy loss. Such optimization strategies are particularly relevant for deploying deep learning models in embedded or edge environments. However, these studies primarily focus on signal-based detection or physical analysis rather than image-based classification.

Despite these advancements, several limitations remain. Most existing studies address general fire detection or electrical fault classification at the system level, without considering forensic-level analysis of post-fire physical traces. In particular, the problem of distinguishing between primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT) using image-based deep learning remains largely underexplored. This gap highlights the need for specialized approaches that can analyze microscopic trace features and support accurate fire cause determination in forensic investigations.

2.2. Lightweight CNN Architectures for Efficient Vision Systems

To address the limitations of conventional deep learning models, lightweight convolutional neural network (CNN) architectures have been developed to reduce computational complexity while maintaining competitive performance. Traditional deep networks, such as VGG and ResNet, typically involve a large number of parameters and high computational cost, making them inefficient for applications where model size and inference speed are critical [20].

Lightweight CNN architectures are specifically designed to improve computational efficiency through optimized network structures. MobileNet, for example, introduces depthwise separable convolutions, which decompose standard convolution operations into depthwise and pointwise convolutions, significantly reducing both parameter count and computational cost. MobileNetV2 further enhances this approach by incorporating inverted residual blocks with linear bottlenecks, enabling efficient feature representation while minimizing information loss. MobileNetV3 extends this design by integrating squeeze-and-excitation (SE) modules and optimized activation functions, improving feature recalibration and overall efficiency.

Similarly, ShuffleNet improves efficiency by utilizing channel shuffling operations to enhance inter-channel information exchange while maintaining a lightweight structure. EfficientNet adopts a different strategy by proposing compound scaling, which systematically balances network depth, width, and input resolution to achieve improved accuracy with fewer parameters compared to conventional architectures [21].

In addition to architectural design, model optimization techniques have been widely explored to further reduce computational requirements. Wang et al. [22] demonstrated that deep neural networks can be effectively compressed through pruning, trained quantization, and encoding, significantly reducing model size without substantial degradation in performance. These techniques complement lightweight architectures by enabling more efficient model representation.

Despite these advancements, most existing studies evaluate lightweight CNN architectures on general-purpose benchmark datasets, focusing primarily on classification accuracy and computational efficiency. However, their effectiveness in domain-specific applications, particularly in scenarios involving fine-grained visual differences, has not been sufficiently investigated. In tasks such as forensic image analysis, where subtle morphological variations are critical, the performance and limitations of lightweight models require more detailed examination.

Therefore, a systematic evaluation of lightweight CNN architectures is necessary to determine their suitability for specialized classification problems involving complex and fine-grained visual features.

2.3. On-Device Deep Learning and Mobile Deployment

Recent advancements in mobile hardware and edge computing have enabled deep learning models to be executed directly on local devices. This approach, commonly referred to as on-device or edge AI, allows inference to be performed without relying on external cloud servers. As a result, it provides several advantages, including reduced latency, improved data privacy, and increased system reliability, particularly in environments where network connectivity is limited or unavailable.

Several studies have demonstrated the feasibility of on-device deep learning across different application domains. In fire detection systems, lightweight CNN models have been implemented on embedded platforms to perform real-time image classification, achieving reliable performance under constrained computational resources [23,24]. Similar approaches have been applied in areas such as healthcare diagnostics and industrial inspection, where immediate processing and decision-making are required at the point of data acquisition.

In addition to application-specific implementations, recent research in TinyML has further extended the capability of deploying deep learning models on highly resource-limited devices, including microcontrollers. TinyML focuses on optimizing neural networks through efficient architectures and model compression techniques, enabling inference with minimal memory and power consumption while maintaining acceptable accuracy [25].

Despite these advancements, many existing systems still rely partially on cloud-based processing for tasks such as data storage, model updates, or large-scale inference. Such hybrid or cloud-dependent approaches introduce several limitations, including increased latency, dependency on stable network connectivity, and potential risks related to data security and privacy. These constraints reduce their effectiveness in field-based scenarios where rapid and autonomous decision-making is required.

Therefore, there is a need for fully self-contained on-device deep learning systems that can perform accurate and efficient inference independently of cloud infrastructure. In applications such as electrical fire investigation, where immediate analysis and portability are critical, mobile-based deployment of lightweight deep learning models provides a practical and effective solution.

2.4. Research Gap

Despite the rapid progress in deep learning and mobile deployment technologies, several critical limitations remain in existing literature. First, most prior studies in fire detection and electrical fault analysis focus on macro-level detection or signal-based diagnosis, as discussed in Section 2.1, rather than the forensic-level interpretation of post-fire physical traces. In particular, the task of distinguishing between primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT), which is essential for accurate fire cause determination, has received limited attention in image-based deep learning research.

Second, although lightweight CNN architectures such as MobileNet and EfficientNet have been developed to improve computational efficiency, as described in Section 2.2, their evaluation has been largely restricted to general-purpose datasets. Their effectiveness in domain-specific applications involving fine-grained visual differences such as microscopic short-circuit trace analysis has not been systematically investigated.

Third, while on-device deep learning and edge AI have demonstrated strong potential for real-time inference, as reviewed in Section 2.3, many existing solutions still rely on cloud-based or hybrid processing frameworks. These approaches introduce latency, require stable network connectivity, and limit their applicability in field environments where immediate and offline analysis is required.

Taking together, these limitations reveal a significant research gap at the intersection of forensic-level electrical fire analysis, lightweight deep learning, and on-device deployment. Specifically, there is a lack of integrated frameworks capable of performing accurate, efficient, and real-time classification of short-circuit traces directly on mobile devices.

To address this gap, this paper proposes a lightweight deep learning-based system that combines domain-specific dataset construction, comparative evaluation of efficient CNN architectures, and real-time on-device deployment using TensorFlow Lite. This integrated approach enables practical and reliable short-circuit trace classification in real-world fire investigation scenarios.

3. Proposed Approach

This paper proposes an efficient and deployable deep learning framework for the classification of electric fire short-circuit traces into Primary Short-Circuit Traces (PSCT) and Secondary Short-Circuit Traces (SSCT). The proposed system is specifically designed to achieve high classification accuracy while maintaining low computational complexity, enabling real-time implementation on resource-constrained devices such as smartphones.

3.1. System Overview

Figure 1 illustrates the overall architecture of the proposed short-circuit trace classification system, which follows a structured end-to-end pipeline from data acquisition to mobile deployment.

The process begins with the collection of raw short-circuit trace images, which are experimentally generated and labeled into two categories: primary short-circuit traces

(PSCT) and secondary short-circuit traces (SSCT). These labeled images form the initial dataset for model development.

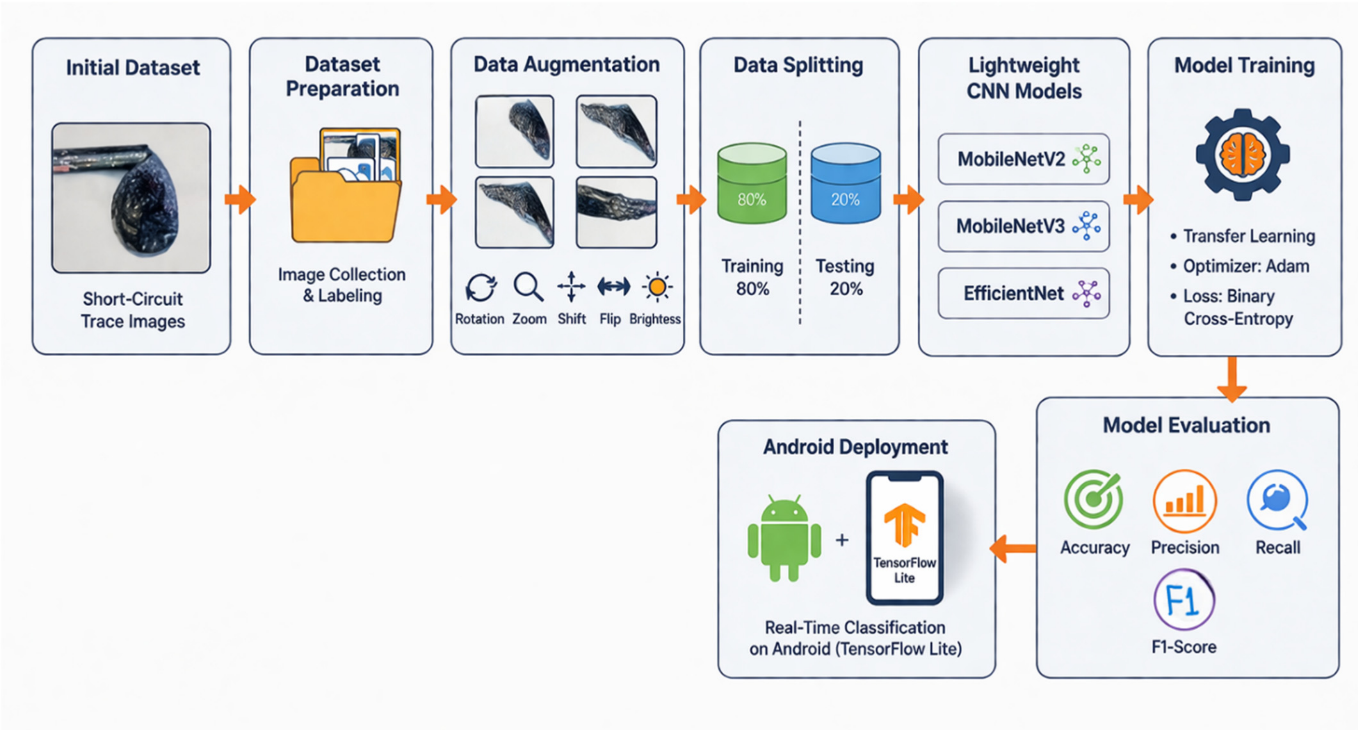

Figure 1. Overall architecture of the proposed short-circuit trace classification system.

In the next stage, data preprocessing and augmentation are applied to improve the robustness and generalization capability of the models. Various augmentation techniques, including rotation, scaling, flipping, shifting, and brightness adjustment, are used to simulate real-world variations in image conditions.

The dataset is then divided into training and testing subsets, with 80% of the data used for model training and 20% reserved for performance evaluation. This separation ensures an unbiased assessment of the model's generalization capability.

Subsequently, three lightweight convolutional neural network architectures MobileNetV2, MobileNetV3, and EfficientNet are employed using a transfer learning approach. The models are trained using the Adam optimizer and binary cross-entropy loss function to perform binary classification between PSCT and SSCT.

After training, the models are evaluated using multiple performance metrics, including accuracy, precision, recall, F1-score, and loss, to comprehensively assess their classification performance.

Finally, the best-performing model is converted into TensorFlow Lite format and deployed on an Android platform. This enables real-time, on-device inference, allowing users to classify short-circuit traces directly in field environments without relying on cloud-based processing.

3.2. Dataset Preparation and Preprocessing

In this paper, a well-structured dataset was constructed to facilitate accurate classification of electric fire short-circuit traces. In this context, a "well-structured dataset" refers to several key characteristics: (i) a balanced distribution of samples between primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT), (ii) consistent image resolution and format (224 × 224 pixels) to ensure compatibility with pretrained CNN architectures, (iii) reliable labeling obtained under controlled experimental conditions, and (iv) sufficient variability introduced through data augmentation techniques.

Such structuring plays a critical role in classification performance. A balanced and consistently formatted dataset reduces bias during training, while accurate labeling ensures meaningful feature learning. Furthermore, increased data variability improves the model's ability to generalize to unseen data and reduces the risk of overfitting. As a result, a well-structured dataset contributes to more stable convergence and improved overall classification accuracy.

The dataset used in this paper consists of a hybrid collection of both laboratory-generated and real-world short-circuit trace images. A total of 752 images were collected, among which 552 samples were generated under controlled laboratory conditions, while 200 images were obtained from real fire-damaged electrical components.

The laboratory-generated samples ensure controlled and reproducible conditions, while the real-world samples capture practical variations such as soot deposition, irregular deformation, oxidation patterns, and varying illumination. By combining these two sources, the dataset incorporates both experimental consistency and real-world complexity. All samples were randomly mixed prior to dataset splitting, and a stratified 80/20 split was applied to ensure that both laboratory and real-world samples were proportionally represented in the training and testing subsets. All images were resized to 224×224 pixels to ensure compatibility with pretrained CNN architectures.

The dataset generation process was conducted under controlled laboratory conditions and required both time and experimental resources. Each sample was produced through a sequence of steps including wire positioning, short-circuit induction, and thermal exposure, followed by a cooling period and image acquisition. On average, the generation and recording of a single sample required approximately 2–3 min. The overall dataset was constructed through repeated experiments, resulting in a total data collection time of several hours.

To ensure reproducibility and experimental consistency, a dedicated experimental platform was designed for short-circuit trace generation, as shown in Figure 2. The platform consists of a controllable power supply system, a wire positioning and transfer mechanism, and a flame generation unit for simulating fire conditions. Copper wires with two different cross-sectional areas (1.5SQ and 4SQ) were used as test samples. These wires were fixed to an electrode system and brought into contact using a controlled transfer mechanism to induce short-circuit conditions at a predefined contact point.

To simulate different fire scenarios, experiments were conducted under two temperature conditions: ambient temperature (~25 °C) and high-temperature conditions (~900 °C), which correspond to typical fire environments. Primary short-circuit traces (PSCT) were generated by continuously applying electrical current to the wires until a short circuit occurred under ambient conditions. In contrast, secondary short-circuit traces (SSCT) were produced by first exposing the wires to external flame using a flame generator for a fixed duration (approximately 2 min), followed by inducing a short circuit under elevated temperature conditions. This process simulates insulation degradation caused by fire prior to electrical failure.

In addition, molten traces were generated by exposing the wires to external flame without applying electrical current, allowing comparison between electrically induced and thermally induced damage. After trace generation, all samples were captured using imaging devices under controlled conditions. A stereomicroscope (10× magnification) and a mobile digital microscope (50× magnification) were used to obtain high-resolution images of the trace morphology.

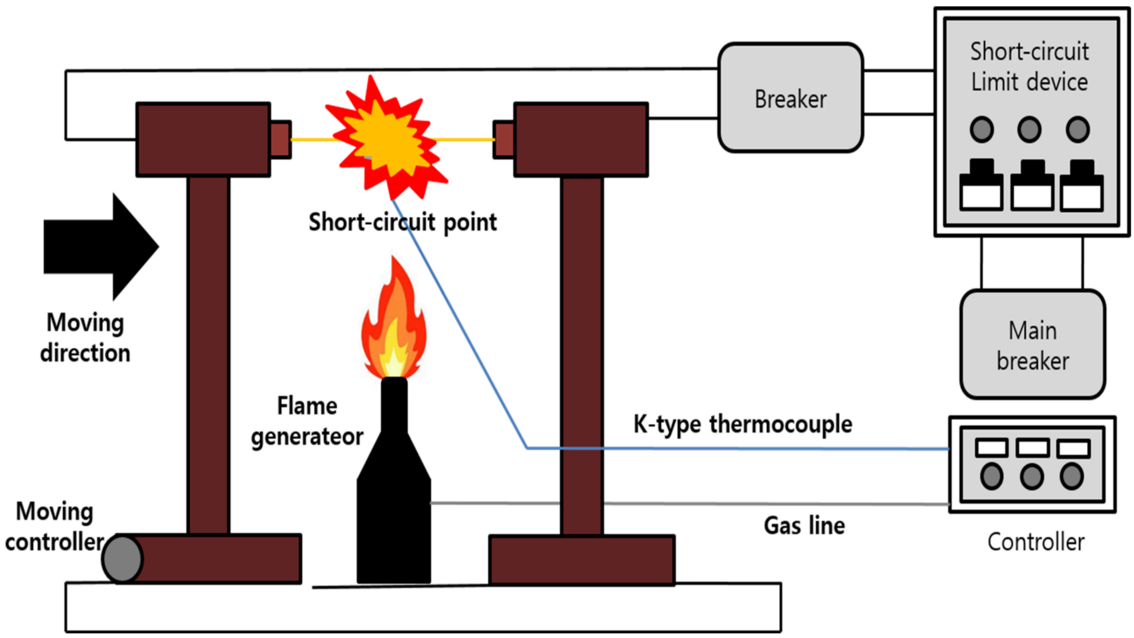

Figure 2. Data production experimental device configuration.

Figure 2 illustrates the experimental platform used for dataset generation, including the wire fixing mechanism, contact point for short-circuit induction, and the overall configuration of the system.

This setup enables controlled and repeatable generation of short-circuit traces, ensuring that the dataset reflects realistic electrical fire scenarios while maintaining experimental reliability. Representative examples of the collected short-circuit trace images are shown in Figure 3.

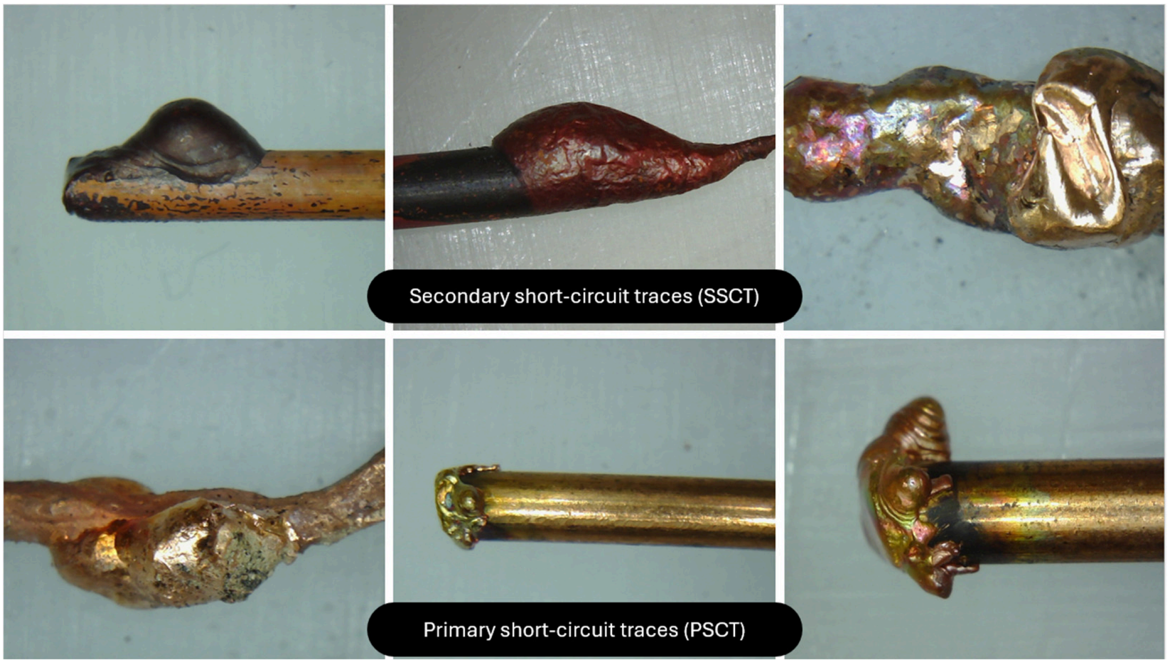

Figure 3. Presents representative samples of PSCT and SSCT images used in this paper.

This experimental setup ensures that the dataset reflects realistic variations in electrical fire conditions while maintaining controlled and reproducible sample generation.

To improve the model's generalization capability and mitigate overfitting, the training dataset underwent extensive data augmentation. This included a range of image transformations such as rotation, scaling, horizontal and vertical flipping, zooming, shifting, and brightness adjustment. These techniques significantly enriched the diversity of the training samples without the need for additional data collection, thereby enhancing the robustness and accuracy of the deep learning models. The specific parameters and objectives for each applied data augmentation technique are detailed in Table 1.

Table 1. Applied data augmentation techniques.

Technique	Value/Range	Purpose
Rescaling	1/255	Normalize pixel values to [0, 1]
Rotation	±30 degrees	Allow to learn rotational invariance
Width Shift	20% of width	Tolerate horizontal displacements
Height Shift	20% of height	Tolerate vertical displacements
Zoom	±30%	Handle variations in image size and scale
Horizontal Flip	True	Learn from mirrored images
Vertical Flip	True	Increase robustness to vertical variations
Brightness Adjustment	Range [0.7, 1.3]	Adapt to different lighting conditions

Data augmentation was applied throughout training to improve model generalization. This strategy helped both models learn from more diverse data and improved their classification performance.

3.3. Lightweight Deep Learning Models

In this paper, MobileNetV2, MobileNetV3, and EfficientNet were selected as representative lightweight CNN architectures due to their proven efficiency in resource-constrained environments. These models were chosen to reflect different design strategies for reducing computational complexity while maintaining high classification performance, making them suitable for real-time, on-device deployment scenarios. Previous studies, including our earlier work using ResNet-based architectures, have demonstrated that deeper models can improve accuracy but introduce higher computational cost, making them less suitable for on-device deployment.

The CNN-based classification framework used in this paper consists of three main components: feature extraction, feature aggregation, and classification.

First, the input image of size $224 \times 224 \times 3$ is fed into a pretrained backbone network (MobileNetV2, MobileNetV3, or EfficientNet), which serves as a feature extractor. These architectures are designed using lightweight building blocks such as depth wise separable convolutions, which significantly reduce computational cost while maintaining representational capacity.

MobileNetV2 employs inverted residual blocks with linear bottlenecks, enabling efficient feature reuse and reduced parameter count [26,27]. MobileNetV3 further enhances this design by incorporating squeeze-and-excitation (SE) modules and optimized activation functions (h-swish), which improve channel-wise feature attention and overall efficiency [28,29]. EfficientNet, on the other hand, utilizes a compound scaling method that balances network depth, width, and resolution to achieve improved performance with fewer parameters [30,31].

Following feature extraction, a global average pooling layer is applied to reduce the spatial dimensions of the feature maps, producing a compact feature vector. This vector is then passed to a fully connected (dense) layer with a sigmoid activation function to perform binary classification between primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT).

Transfer learning is employed by initializing backbone networks with ImageNet pretrained weights. During training, lower layers are frozen to retain general visual features, while higher layers are fine-tuned to adapt to the specific characteristics of short-circuit trace images.

The overall structure of the proposed CNN-based classification framework is illustrated in Figure 4, where the input image undergoes preprocessing before being passed through a lightweight feature extraction network, followed by global average pooling and a fully connected layer for final PSCT and SSCT classification.

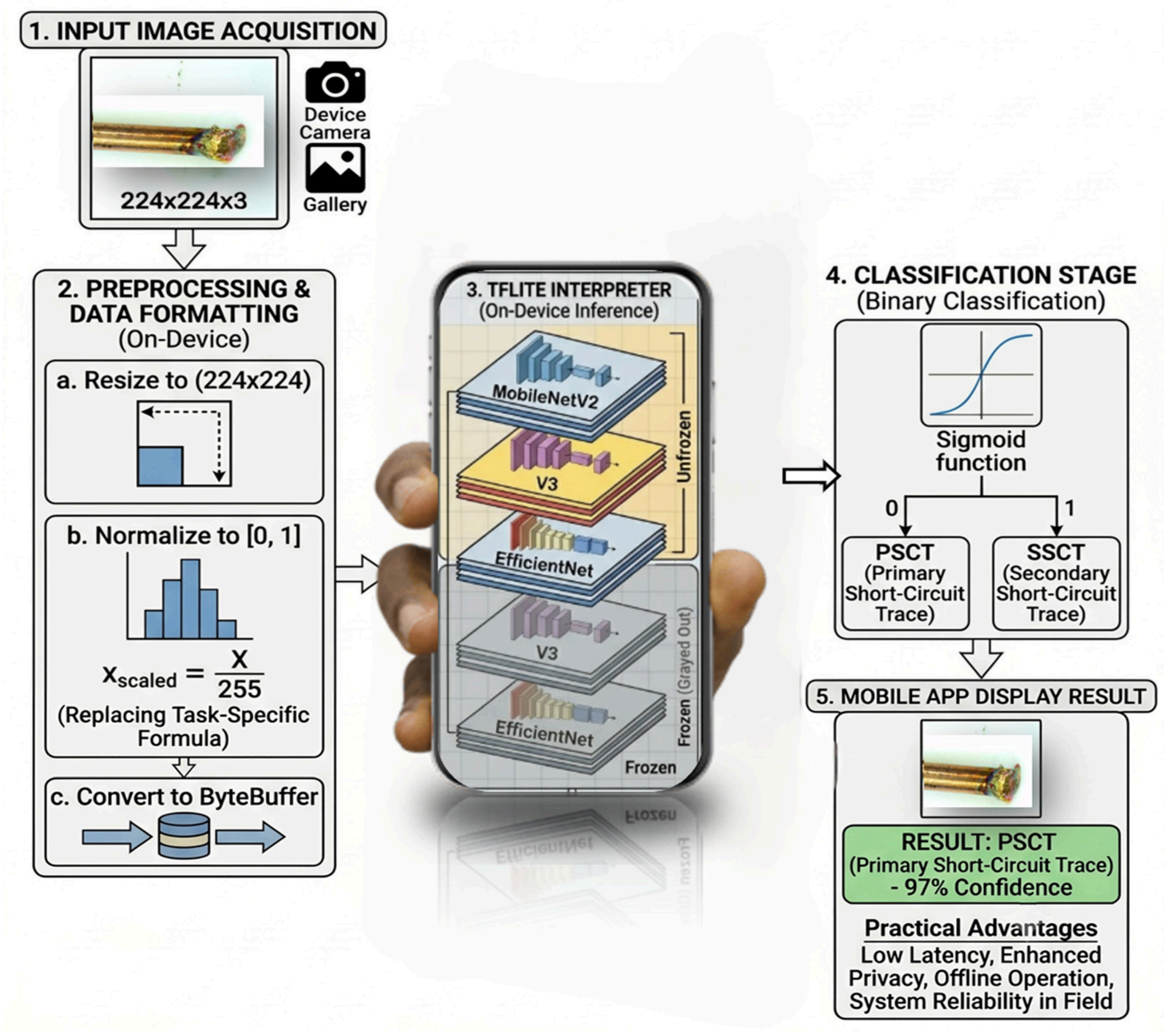

Figure 4. Schematic diagram of the CNN-based classification framework.

3.4. Training Strategy

The model training procedure was carefully designed to achieve stable convergence and optimal generalization performance. The models were implemented using transfer learning with weights pre-trained on the ImageNet dataset. The models were trained for 100 epochs to ensure sufficient convergence. Although the number of epochs may appear relatively high given the dataset size, no significant overfitting was observed during training. As shown in the training logs, both accuracy and loss exhibit gradual and stable convergence, with validation accuracy reaching approximately 96% and validation loss continuously decreasing to 0.1739 for EfficientNet and 0.1044 for MobileNetV3. Importantly, no divergence between training and validation performance was observed, indicating that the models learned generalized representations rather than memorizing the training data.

An adaptive optimization approach was employed using the Adam optimizer, which efficiently updates network weights based on first- and second-order moments of the gradients, leading to faster convergence. The learning objective was defined using the Binary Cross-Entropy loss function, which is well-suited for binary classification tasks.

To ensure numerical stability and consistent model performance, all input images were normalized by scaling pixel values to the range [0, 1].

During training, a fine-tuning strategy was adopted in which the lower layers of the network responsible for generic feature extraction were frozen, while the higher layers were selectively unfrozen and trained on the target dataset. This allows the model to retain previously learned low-level representations while adapting high-level features to the specific characteristics of short-circuit trace images.

Formally, the classification task can be defined as

$$f(x) \to y \in \{0,1\}$$

where x represents the input image and y denotes the class label (PSCT or SSCT).

This training configuration is particularly effective for small-scale datasets, as it leverages rich feature representations learned from large-scale data while reducing the risk of overfitting and improving generalization performance.

3.5. Android-Based Deployment

To validate the real-world applicability of the proposed system, the trained deep learning model was converted into the TensorFlow Lite (TFLite) format and deployed within an Android application to enable efficient on-device inference.

The mobile inference pipeline was designed to operate in real time and consists of several sequential stages. First, the input image is acquired either through the device camera or selected from the gallery. The captured image is then preprocessed, including resizing to a fixed input dimension (224 × 224) and normalization of pixel values to the range [0, 1], ensuring consistency with the training configuration.

Subsequently, the processed image is converted into a ByteBuffer representation, which facilitates efficient memory management and fast data transfer to the TFLite interpreter. The converted data is then fed into the deployed model for inference, producing a real-time classification output corresponding to either primary short-circuit traces (PSCT) or secondary short-circuit traces (SSCT).

Unlike conventional cloud-based inference systems, the proposed approach performs all computations directly on the mobile device. This on-device deployment offers several practical advantages, including reduced inference latency, elimination of network dependency, enhanced data privacy, and improved system reliability in field environments. These characteristics make the proposed system particularly suitable for real-time electrical fire investigation scenarios.

The use of MobileNetV3 further ensures efficient on-device execution due to its reduced model size (~3.58 MB) and lower computational complexity, enabling faster inference compared to heavier architectures such as EfficientNet (~15.45 MB).

4. Result and Analysis of Experiment

All experiments and model training were conducted on a 64-bit Windows 11 system equipped with an AMD Ryzen 9 5900× 12-core processor (3.70 GHz) and 32 GB of RAM, operating on x64-based architecture. The models were implemented in Python 3.13 us ing the TensorFlow framework with the Keras API. The evaluation is conducted using accuracy, precision, recall, F1-score, and loss metrics.

4.1. Quantitative Results

To quantitatively evaluate model performance, experiments were conducted on the test dataset using multiple evaluation metrics, including accuracy and loss. In addition to final classification results, the training and validation curves are analyzed to examine convergence behavior, generalization performance, and potential overfitting.

EfficientNet employs a compound scaling method to balance network depth, width, and resolution, enabling improved feature extraction capability. As shown in Figure 5, which presents the accuracy curves, loss curves, and confusion matrix within a single unified visualization, the model demonstrates stable convergence behavior during training. The accuracy curves indicate a rapid improvement in the early epoch, followed by gradual stabilization, while the loss curves show a consistent decreasing trend, confirming effective optimization.

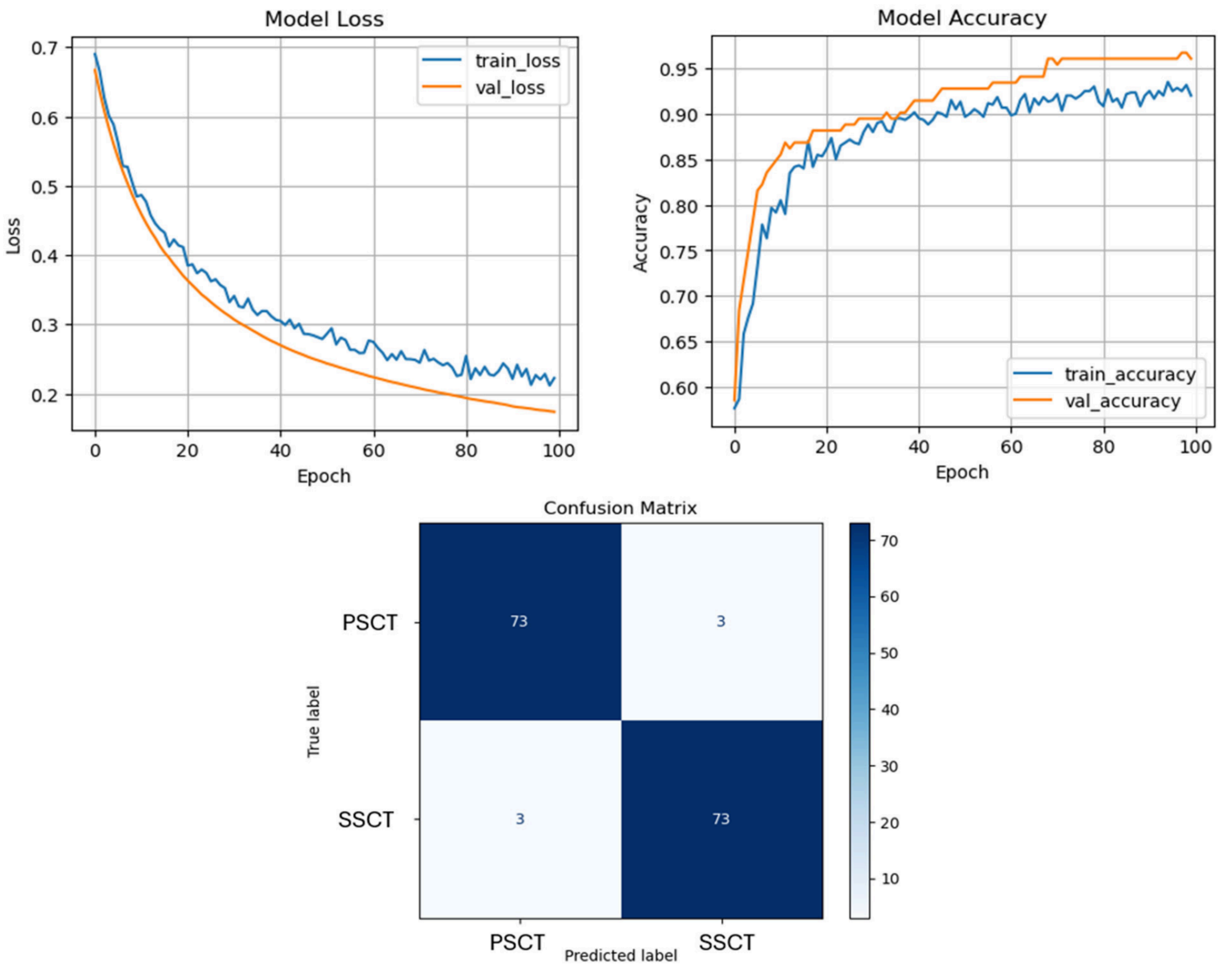

Figure 5. Training and validation accuracy curves, loss curves, and confusion matrix of EfficientNet.

The confusion matrix in Figure 5 further demonstrates the classification effectiveness of EfficientNet, showing a high number of correctly classified samples for both primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT), with minimal misclassification. This indicates that the model can capture discriminative features necessary for distinguishing between the two classes.

MobileNetV3 improves upon previous architectures by incorporating squeeze-and-excitation (SE) modules and optimized activation functions (h-swish), enabling more effective feature representation. As shown in Figure 6, which presents the training and validation accuracy curves, loss curves, and confusion matrix in a unified visualization, the model demonstrates stable convergence behavior throughout the training process.

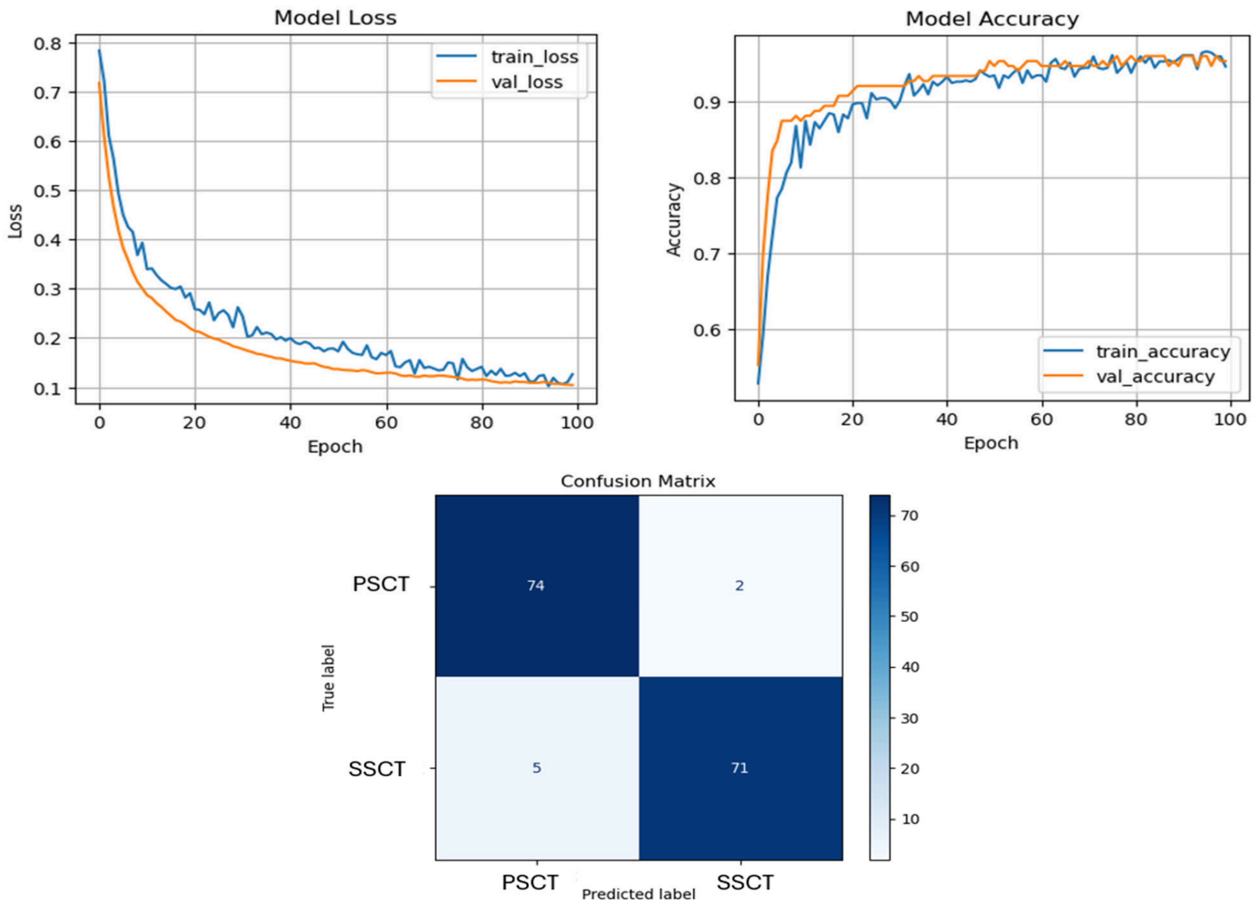

Figure 6. Training and validation accuracy curves, loss curves, and confusion matrix of MobileNetV3.

The accuracy curves indicate a rapid increase during the initial epochs, followed by gradual stabilization, while the loss curves exhibit a consistent decreasing trend, confirming effective optimization and reduced training error. Notably, MobileNetV3 achieves a lower validation loss compared to the other models, indicating superior generalization capability.

The confusion matrix in Figure 6 further highlights the classification performance of MobileNetV3, showing a high number of correctly classified samples for both primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT), with only a small number of misclassifications. This demonstrates the model's ability to effectively distinguish between the two classes, even in visually similar patterns.

MobileNetV2 is a lightweight convolutional neural network architecture designed for efficient computation using depth wise separable convolutions and inverted residual blocks. As shown in Figure 7, which presents the training and validation accuracy curves, loss curves, and confusion matrix in a unified visualization, the model demonstrates stable learning behavior throughout the training process.

The accuracy curves show a rapid increase during the initial training phase, followed by gradual convergence, indicating effective feature learning. Similarly, the loss curves exhibit a consistent decreasing trend, confirming proper optimization. However, compared to the other models, MobileNetV2 converges to slightly lower validation accuracy and a higher loss value, suggesting relatively limited feature extraction capability.

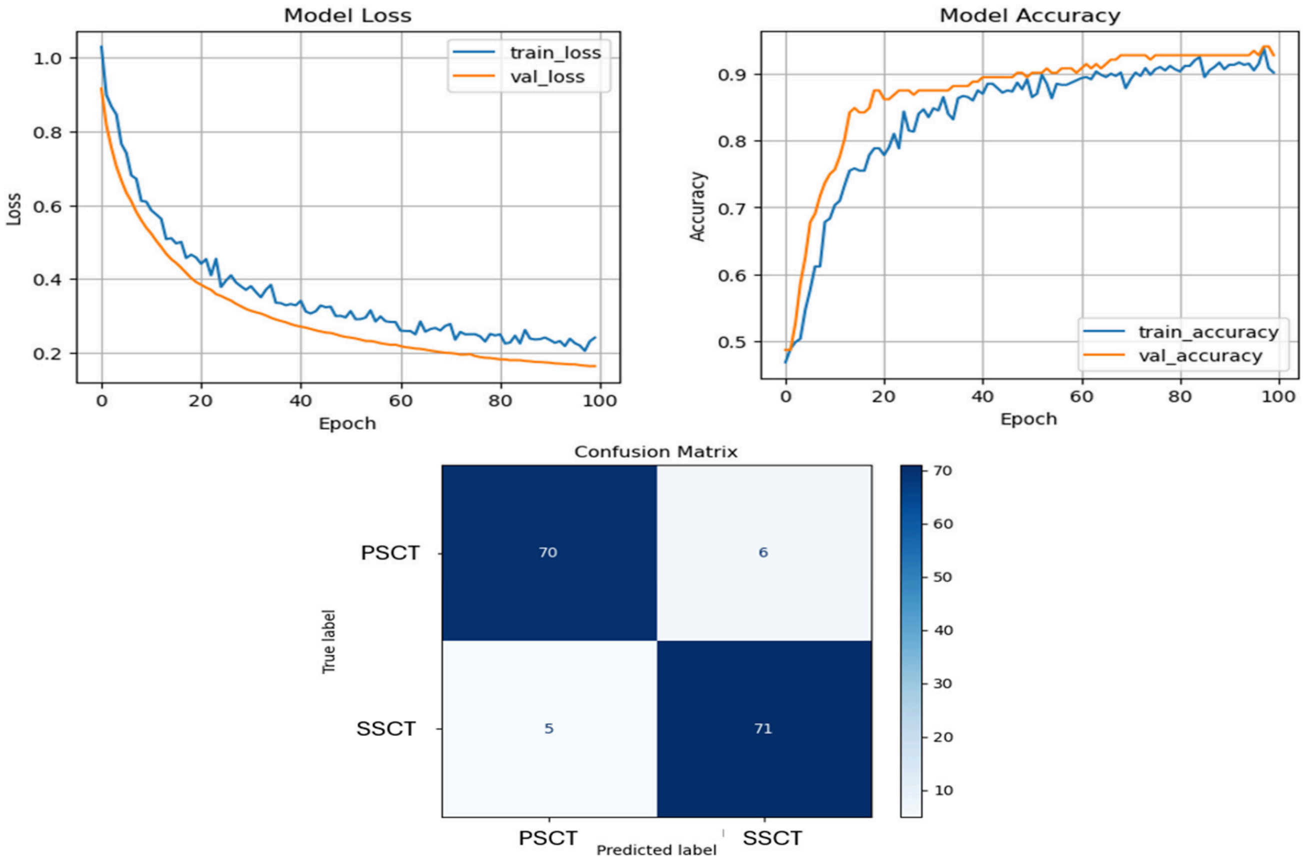

Figure 7. Training and validation accuracy curves, loss curves, and confusion matrix of MobileNetV2.

The confusion matrix in Figure 7 further illustrates the classification performance of MobileNetV2. While the model correctly classifies the majority of primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT), a higher number of misclassifications is observed compared to MobileNetV3 and EfficientNet. This indicates that the relatively simpler architecture of MobileNetV2 may struggle to capture fine-grained differences between visually similar trace patterns.

These results confirm that all three lightweight CNN models are effective for short-circuit trace classification, with EfficientNet providing the best overall accuracy.

Although the overall classification accuracy is high, a small number of misclassifications (approximately 2–6 samples per model) were observed. Visual inspection of these samples indicates that most misclassified cases correspond to borderline traces exhibiting mixed morphological characteristics between PSCT and SSCT. Some secondary short-circuit traces exposed to high temperatures exhibit localized melting patterns like primary traces, while certain primary traces show oxidation features resembling secondary damage. These ambiguous cases highlight the intrinsic difficulty of the classification task and suggest that incorporating additional contextual or multi-modal information may further improve discrimination performance.

It should be noted that the reported results are based on a single train–test split. While the models demonstrate stable convergence and strong performance, reporting statistical variability such as standard deviation or confidence intervals would provide a more rigorous evaluation.

Future work will include repeated experiments with multiple random initializations and k-fold cross-validation to quantify performance variance and ensure statistical reliability.

4.2. Comparative Performance Analysis

The observed performance differences among the evaluated models can be attributed to their architectural design and feature extraction mechanisms. As illustrated in Figure 8a,b, which present the validation accuracy and validation loss curves over 100 epochs, all models demonstrate a rapid improvement in performance during the initial training phase, followed by gradual convergence, indicating effective learning and stable optimization behavior.

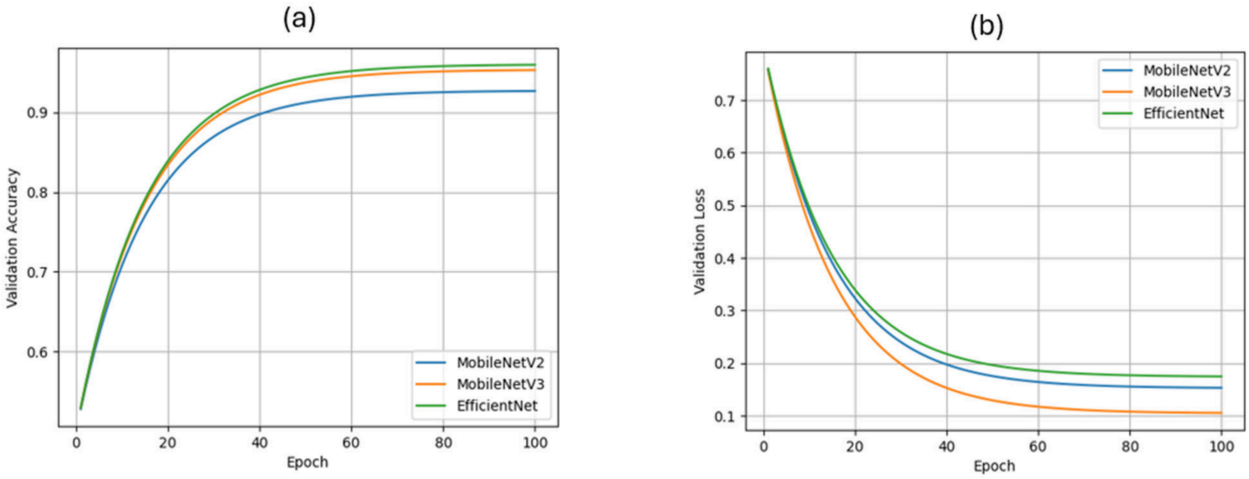

Figure 8. Comparison of validation accuracy (**a**) and validation loss (**b**) for MobileNetV2, MobileNetV3, and EfficientNet.

In terms of classification accuracy, EfficientNet achieves the highest validation accuracy of 96.05%, demonstrating superior feature extraction capability. This performance can be primarily attributed to its compound scaling strategy, which jointly optimizes network depth, width, and input resolution. Such balanced scaling enables EfficientNet to capture subtle visual differences between primary short-circuit traces (PSCT) and secondary short-circuit traces (SSCT), resulting in enhanced discriminative performance in complex classification scenarios.

MobileNetV3 also exhibits competitive performance, achieving a validation accuracy of 95.39%. Notably, it demonstrates the lowest validation loss (0.1044) among all models, as shown in Figure 8b, indicating better generalization capability and more stable convergence during training. This behavior can be attributed to the incorporation of squeeze-and-excitation (SE) modules and improved nonlinear activation functions (h-swish), which enhance the model's ability to focus on the most informative regions of the input while maintaining computational efficiency.

In contrast, MobileNetV2 achieves a slightly lower validation accuracy of 92.76% and a higher validation loss compared to the other models. Although its use of depth wise separable convolutions and inverted residual blocks significantly reduces computational complexity, its relatively simpler architecture limits its ability to capture fine-grained discriminative features necessary for distinguishing between visually similar trace patterns.

The quantitative performance comparison is summarized in Table 2, which reports accuracy, precision, recall, F1-score, and loss for each model. EfficientNet achieves the highest overall performance with an F1-score of 0.96, reflecting its balanced precision and recall. MobileNetV3, while slightly lower in accuracy, achieves strong recall (0.97) and the lowest loss, further confirming its robustness and generalization capability. MobileNetV2 shows comparatively lower performance across all evaluation metrics.

https://doi.org/10.3390/en19133184

Table 2. Quantitative performance comparison of MobileNetV2, MobileNetV3, and EfficientNet.

Model	Accuracy (%)	Precision	Recall	F1-Score	Loss
MobileNetV2	92.76	0.93	0.92	0.92	0.1523
MobileNetV3	95.39	0.94	0.97	0.95	0.1044
EfficientNet	96.05	0.96	0.96	0.96	0.1739

In addition to overall metrics, class-wise performance was evaluated. For EfficientNet, the precision and recall for PSCT were approximately 0.96 and 0.95, respectively, while for SSCT they were 0.96 and 0.97. Similar balanced performance was observed for MobileNetV3 and MobileNetV2, indicating that the models do not exhibit bias toward a specific class.

Notably, MobileNetV3 achieves near-optimal performance with significantly fewer parameters compared to EfficientNet, highlighting its suitability for deployment in resource-constrained environments.

Further insights can be obtained from the confusion matrices, which illustrate the classification results for each model. EfficientNet demonstrates the most balanced classification performance, with minimal misclassification between PSCT and SSCT classes. MobileNetV3 also shows strong performance, particularly in correctly identifying PSCT samples, while MobileNetV2 exhibits relatively higher misclassification rates, especially for samples with subtle visual differences.

From a practical deployment perspective, these results highlight a trade-off between accuracy and computational efficiency. Although EfficientNet provides the highest classification performance, its increased complexity may lead to higher inference latency on resource-constrained devices. In contrast, MobileNetV3 offers a more balanced solution, achieving competitive accuracy with lower computational cost and superior generalization, making it particularly suitable for real-time, on-device applications such as mobile-based fire investigation systems.

From a deployment perspective, computational efficiency plays a critical role in model selection. Although EfficientNet achieved the highest classification accuracy (96.05%), MobileNetV3 provides a more favorable trade-off between accuracy and computational efficiency.

Specifically, EfficientNet contains approximately 4.05 million parameters (15.45 MB), whereas MobileNetV3 requires only 0.94 million parameters (3.58 MB), representing a reduction of approximately 4.3$\times$ in model size. In addition, the training logs indicate that MobileNetV3 achieves faster inference per step (~430 ms/step) compared to EfficientNet (~720 ms/step), demonstrating significantly lower computational cost.

Despite this reduction in complexity, MobileNetV3 achieves competitive performance (95.39% accuracy) with the lowest validation loss (0.1044), indicating strong generalization capability. Therefore, MobileNetV3 was selected for deployment due to its ability to balance accuracy, efficiency, and real-time performance on resource-constrained mobile devices.

It is worth noting that the inclusion of real-world samples in both training and testing contributes to the observed generalization performance. The model maintains high classification accuracy despite the presence of real-world variability, suggesting that the learned features are not limited to laboratory-specific patterns but extend to practical fire investigation scenarios.

A comparison of model complexity further highlights the efficiency–performance trade-off. EfficientNet and MobileNetV2 contain approximately 4.05 million and 2.26 million parameters, respectively, whereas MobileNetV3 requires less than 1 million parameters. Despite this significantly lower complexity, MobileNetV3 achieves performance comparable to EfficientNet, demonstrating that lightweight architectures can effectively capture the fine-grained morphological features of short-circuit traces while maintaining computational efficiency.

4.3. Implications for Real-World Deployment

To validate real-world applicability, the trained model was deployed on an Android-based mobile platform for real-time, on-device inference. As illustrated in Figure 9, the mobile inference pipeline consists of several sequential stages to ensure efficient execution on resource-constrained devices. The best-performing model was converted into TensorFlow Lite (TFLite) format using the TensorFlow Lite converter. The mobile application was developed using Android Studio, where the TFLite model was integrated into the application for inference.

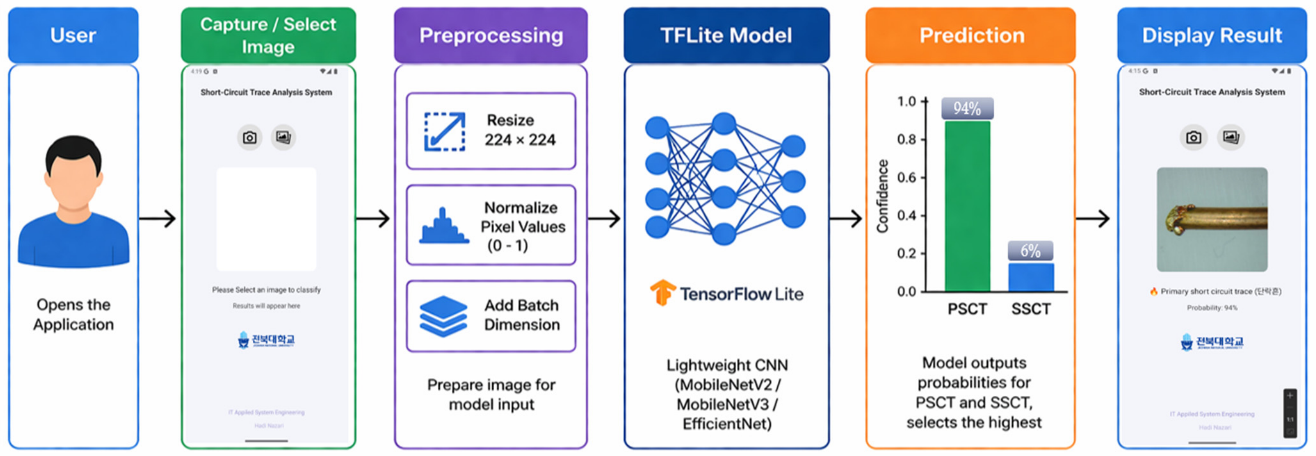

Figure 9. Mobile-based on-device inference pipeline for real-time short-circuit trace classification.

The deployment pipeline consists of the following steps:

- Image Acquisition: Input images are captured using the device camera or selected from the gallery.
- Preprocessing: Images are resized to 224 $\times$ 224 pixels and normalized to match the model's input requirements.
- Data Conversion: The preprocessed image is converted into a ByteBuffer format, which is required for efficient memory handling and compatibility with the TFLite interpreter.
- Model Inference: The processed data is passed to the TFLite model for classification.
- Result Output: The prediction result (PSCT or SSCT) is displayed in real time on the mobile interface.

The on-device inference approach eliminates the need for cloud-based processing, thereby reducing latency and enabling offline functionality. Additionally, this method enhances data privacy and ensures reliable performance in field environments where network connectivity may be limited or unavailable.

4.4. Limitations and Future Work

Despite the promising results, this paper has some limitations that should be acknowledged.

First, the dataset size used in this paper is relatively limited (752 images). Although data augmentation and transfer learning were employed to mitigate overfitting, deep learning models generally benefit from larger and more diverse datasets.

Second, while the dataset includes both laboratory-generated and real-world samples, the proportion of real-world data (200 images) remains relatively limited. As a result, the dataset may still be partially biased toward controlled experimental conditions. Additionally, the current paper does not explicitly evaluate model robustness under degraded image conditions such as low lighting, motion blur, or partial occlusion. In real fire investigation scenarios, such conditions are common and may affect classification performance. Future work will investigate model robustness under varying image quality conditions and incorporate data augmentation strategies to simulate these challenges.

Finally, the classification task is limited to binary categorization (PSCT vs. SSCT). In practical fire investigations, more complex scenarios may involve mixed or ambiguous trace characteristics. Future work will extend the current framework to multi-class classification and incorporate additional real-world variability to improve robustness and applicability.

It is worth noting that the low validation loss values observed (e.g., 0.1044 for MobileNetV3) may be partially influenced by the controlled nature of a portion of the dataset, which can reduce variability compared to real-world scenarios.

5. Conclusions

This paper proposed a lightweight deep learning-based framework for the classification of electric fire short-circuit traces, with the objective of improving safety and fault diagnosis in electrical energy systems. Through a systematic comparative analysis of MobileNetV2, MobileNetV3, and EfficientNet, it was demonstrated that model performance must be evaluated not only in terms of classification accuracy but also considering generalization capability and computational efficiency for practical deployment.

Experimental results showed that EfficientNet achieved the highest classification accuracy due to its compound scaling strategy, while MobileNetV3 exhibited the lowest validation loss and the most stable convergence behavior. Considering the trade-off between performance and computational efficiency, MobileNetV3 was identified as the most suitable model for real-time applications in resource-constrained environments.

Beyond model evaluation, the key contribution of this paper lies in the successful deployment of the optimized model on an Android-based mobile platform using TensorFlow Lite. The proposed system enables real-time, on-device classification without reliance on cloud infrastructure, ensuring low latency, enhanced data privacy, and robust operation in field conditions. This on-device inference capability is particularly valuable for electrical fire investigation scenarios, where immediate decision-making is critical.

From an energy systems perspective, accurate identification of short-circuit traces supports rapid fault localization, reduces investigation time, and enhances the reliability and safety of electrical infrastructure. The proposed approach can assist engineers, inspectors, and safety personnel in diagnosing failure sources more efficiently, thereby contributing to improved operational stability and risk mitigation in energy systems.

Overall, this paper demonstrates that lightweight convolutional neural networks, when combined with mobile deployment technologies, can provide a practical and scalable solution for intelligent fault analysis in electrical energy systems.

Author Contributions: Conceptualization, M.H.N.; methodology, M.H.N.; software, M.H.N.; validation, M.H.N.; formal analysis, M.H.N.; investigation, M.H.N.; resources, J.B.; data curation, M.H.N.; writing—original draft preparation, J.B.; writing—review and editing, J.B.; visualization, M.H.N.; supervision, M.H.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MCEE) (RS-2022-KP002707, Jeonbuk Regional Energy Cluster Training of human resources).

Data Availability Statement: The data presented in this paper are available on request from the corresponding author due to privacy/proprietary laboratory restrictions.

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

References

1. Wang, J.; Wang, S.; Zhang, Y. Deep learning on medical image analysis. *CAAI Trans. Intell. Technol.* **2025**, *10*, 1–35. [CrossRef]
2. Kumar, R.R.; Priyadarshi, R. Denoising and segmentation in medical image analysis: A comprehensive review on machine learning and deep learning approaches. *Multimed. Tools Appl.* **2025**, *84*, 10817–10875. [CrossRef]
3. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. *Inf. Fusion* **2018**, *42*, 146–157. [CrossRef]
4. Chen, B.; Chen, S. Deep Learning. In *Machine Vision Technology*; Springer: Singapore, 2026. [CrossRef]
5. Wagobera, E.K.; Evans, J.; Mukasa, P. Machine learning software for optimizing SME social media marketing campaigns. *Comput. Sci. IT Res. J.* **2024**, *5*, 1634–1647. [CrossRef]
6. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
8. Zhao, X.; Wang, L.; Zhang, Y.; Han, X.; Deveci, M.; Parmar, M. A review of convolutional neural networks in computer vision. *Artif. Intell. Rev.* **2024**, *57*, 99. [CrossRef]
9. Li, Y. Research and application of deep learning in image recognition. In Proceedings of the IEEE International Conference on Power Electron Computer Applications, Shenyang, China, 21–23 January 2022; pp. 994–999. [CrossRef]
10. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
11. Thai, H.-D.; Le, N.-B.-V.; Lee, D.; Huh, J.-H. A survey of electrical fire causes assessment technology. *IEEE Access* **2024**, *12*, 145378–145392. [CrossRef]
12. Park, C.-J.; Kim, T.-H.; Lee, B.-H. Detection algorithms of parallel arc fault on AC power lines based on deep learning techniques. *J. Electr. Eng. Technol.* **2022**, *17*, 1195–1205. [CrossRef]
13. Chen, X.K.; Sun, Y.; Li, Y.; Zhang, Y.N.; Wang, H.B.; Lyu, H.F.; Zhao, Y.T. Characteristics of short circuit and combustion in electrical cables under various external heat fluxes. *Fire Mater.* **2025**, *49*, 1088–1098. [CrossRef]
14. Incel, O.D.; Bursa, S.Ö. On-device deep learning for mobile and wearable sensing applications: A review. *IEEE Sens. J.* **2023**, *23*, 5501–5512. [CrossRef]
15. Verber, D.; Grneva, T.; Dugonik, J. Image-based waste classification using a hybrid deep learning architecture with transfer learning and edge AI deployment. *Mathematics* **2026**, *14*, 1176. [CrossRef]
16. Rizvi, S.; Scanlon, M.; McGibney, J.; Sheppard, J. Deep learning based network intrusion detection system for resource-constrained environments. In *Digital Forensics and Cyber Crime (ICDF2C 2022)*; Springer: Cham, Switzerland, 2023; pp. 355–367. Available online: https://link.springer.com/chapter/10.1007/978-3-031-36574-4_21 (accessed on 1 July 2026).
17. Mittal, P. A comprehensive survey of lightweight object detection models for edge devices. *Artif. Intell. Rev.* **2024**, *57*, 242. [CrossRef]
18. Hasan, M.; Prince, M.M.A.H.; Ansari, M.S.; Jahan, S.; Miah, A.S.M.; Shin, J. FireLite: Leveraging transfer learning for efficient fire detection in resource-constrained environments. In Proceedings of the International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 18–20 December 2024; pp. 545–550. [CrossRef]
19. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv* **2016**, arXiv:1510.00149. [CrossRef]
20. Cordova-Cardenas, R.; Amor, D.; Gutiérrez, Á. Edge AI in practice: A survey and deployment framework for neural networks on embedded systems. *Electronics* **2025**, *14*, 4877. [CrossRef]
21. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856. [CrossRef]
22. Wang, K.; He, G.; Li, X. A lightweight multi-scale context detail network for efficient target detection in resource-constrained environments. *Sensors* **2025**, *25*, 3800. [CrossRef] [PubMed]
23. Jonnalagadda, A.V.; Hashim, H.A. A segmented deep learning-based CNN approach for drones wildfire detection. *Remote Sens. Appl. Soc. Environ.* **2024**, *34*, 101181. [CrossRef]
24. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114. [CrossRef]
25. Somvanshi, S.; Kumar, A.; Sharma, P. From tiny machine learning to tiny deep learning: A survey. *ACM Comput. Surv.* **2026**, *58*, 7. [CrossRef]
26. Akay, M.; Du, Y.; Downer, J. Deep learning classification using MobileNetV2. *IEEE Open J. Eng. Med. Biol.* **2021**, *2*, 104–110. [CrossRef] [PubMed]

27. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520. [CrossRef]
28. Weng, Y.; Li, Z.; Chen, X.; He, J.; Liu, F.; Huang, X.; Yang, H. A railway track extraction method based on improved DeepLabV3+. *Electronics* **2023**, *12*, 3500. [CrossRef]
29. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
30. Ahmed, T.; Sabab, N.H.N. Classification using EfficientUNet. *SN Comput. Sci.* **2022**, *3*, 99. [CrossRef]
31. Saeed, M.I.; Sheng, J.; Al-Anisi, A. Efficient-MobileNet: A feature fusion approach. In Proceedings of the IEEE International Conference on High Performance Computing and Communications (HPCC), Chengdu, China, 18–20 December 2024; pp. 1891–1898. [CrossRef]