ISSN: 2723-9535

Available online at www.HighTechJournal.org

HighTech and Innovation Journal

Vol. 6, No. 3, September, 2025

Optimizing AIGC Technology for IoT Devices with Deep Learning

YuShui Xiao [1*], Yucheng Dong [1]

[1] *Nanchang Institute of Technology, Nanchang City, 330108, Jiangxi Province, China.*

Received 24 December 2024; Revised 17 June 2025; Accepted 02 July 2025; Published 01 September 2025

Abstract

The present article intends to explore how a deep learning model could be applied to improve the ability of AI-generated content (AIGC) technology in graphic recognition within the IoT ecosystem. Objectives: This research pursues two key objectives: first, the model is compressed to a smaller size and decreased computational cost for on-device deployment on resource-poor IoT devices, and second, it achieves better adaptability through data augmentation and regularization techniques. Methods/Analysis: A purpose-built CNN design was built and trained to solve IoT-specific constraints. Model compression techniques such as weight pruning and quantization were used to reduce resource requirements. To ameliorate this, we applied data augmentation techniques like rotation, shear, and zoom, and regularization techniques like dropout to avoid overfitting. The work was done on MNIST and CIFAR-10 typical datasets using TensorFlow as a deep learning framework. Results: The pattern-recognition accuracy on MNIST and CIFAR-10 datasets achieved are 99.5% and 89.2%, respectively. Moreover, the recognition speed was improved by around 30% since the computational cost of the DL algorithm is effective because of parallel processing, resulting in lower processing time. The compressed model overcame the massive computational complexity, which is more suitable for resource-limited IoT devices. Novelty/Improvement: a new methodology is presented that integrates CNN optimization and model compression in conjunction with sophisticated regularization techniques to develop a suitable solution for the peculiarities of the IoT landscape. Ultimately, overcoming the universal problems like limited resources and real-time processes in this research helps to improve the technological and theoretical support for practical IoT applications and accelerate the practical implementation of AIGC performance optimization across various industries such as smart homes, smart transportation, and smart security.

Keywords: Deep Learning; Convolutional Neural Network; Internet of Things; AIGC Technology; Pattern Recognition Optimization.

1. Introduction

Science and technology have seen remarkable advancements, leading to the widespread integration of the IoT into our daily lives. As a seamless connector between the physical and digital worlds, IoT has made its mark in various domains [1]. Be it in smart homes, industrial automation, or even the development of smart cities, IoT devices are proliferating at an unparalleled rate, generating enormous amounts of data [2]. Among these data, graphic data occupy a large part, and they carry rich information, which is of great significance for realizing intelligent decision-making and precise control [3].

However, pattern recognition in the IoT environment faces many challenges. First of all, IoT devices are usually limited in resources, such as computing power, storage space, and energy supply, which makes it difficult to directly perform complex graphics processing on devices [4]. Secondly, the graphic data in IoT environments often have diversity and complexity, such as different lighting conditions, angle changes, obstructions, etc., which increase the difficulty of

* Corresponding author: xiaoyushui@wxic.edu.cn

http://dx.doi.org/10.28991/HIJ-2025-06-03-014

graphic recognition [5]. In addition, with the continuous expansion of IoT applications, higher requirements are put forward for the real-time and accuracy of pattern recognition [6].

AIGC technology (Fusion Technology of Artificial Intelligence, IoT, Graphic Computing, and Cloud Computing) has important application value in IoT graphic recognition [7]. By integrating the robust computational capabilities of artificial intelligence with the meticulous processing strengths of graphic computing, AIGC technology has the potential to achieve efficient and precise graphic recognition within the IoT landscape. This advancement can not only elevate the intelligence quotient of IoT devices but also usher in transformative shifts across various sectors, including intelligent transportation, security, and healthcare [8]. Concurrently, the ongoing evolution of DL technology, particularly its proven efficacy in image recognition, presents a fresh optimization pathway for AIGC technology [9]. Through its inherent ability to discern underlying patterns and representations from extensive sample datasets, DL can adeptly extract pertinent feature information from images, thereby substantially enhancing pattern recognition performance. Therefore, applying DL technology to AIGC technology is expected to solve many challenges of pattern recognition in the IoT environment and promote the further development of IoT technology.

This article aims to optimize the pattern recognition performance of AIGC technology in IoT environments through the DL algorithm. The following are the innovations of this article:

In this article, the DL algorithm is introduced into the pattern recognition of AIGC technology, and optimized according to the characteristics of the IoT environment. By constructing the CNN model and improving and adjusting it, the accuracy and speed of pattern recognition are significantly improved. This innovation breaks the limitation of traditional AIGC technology in pattern recognition and expands new possibilities for its application in the IoT field.

Aiming at the problem of limited IoT equipment resources, this article innovatively adopts model compression technology, which effectively reduces the volume and computational complexity of the DL model. This innovation makes the optimized AIGC technology more suitable for IoT environments with limited resources and improves its practicability and deployment.

This article holistically addresses pattern recognition, emphasizing accuracy and speed while also taking into account model size, computational complexity, and other pertinent factors. Through this approach, it achieves a comprehensive enhancement of AIGC technology's performance. Furthermore, the article aligns with real-world IoT application scenarios and requirements, exploring the prospects and obstacles of optimized AIGC technology in areas like intelligent transportation and smart homes. This offers valuable insights and direction for practical implementation, underscoring the study's pragmatic value and forward-thinking nature.

Our research expands on the current studies on graphic recognition in AIGC technology, intending to address the exclusive challenges presented by IoT environments. While some previous attempts have touched on aspects of our solution, none have provided a fully integrated and comprehensive approach that specifically tackles the key issues encountered in IoT deployments. This paper bridges that gap by combining advanced techniques like CNN, model compression, data augmentation, and regularization methods to create a sophisticated and efficient model. We showcase the capabilities of our model through thorough simulations and assessments using well-established datasets (MNIST and CIFAR-10). Significantly, our comparative analyses highlight substantial enhancements in recognition accuracy, processing speed, and resource utilization compared to traditional methods. Consequently, our work contributes to the advancement of AIGC technology in IoT settings, paving the way for wider adoption and enhanced functionality.

The article comprises five distinct sections, each focusing on the following key aspects:

Section I: Introduction. Introduce the research background, significance, objectives, and overall structure of the paper.

Section II: Theoretical basis and literature review. This article expounds on the related theories and research progress of DL, AIGC technology, and image recognition in the IoT environment.

Section III: Optimization model construction. The DL algorithm, pattern recognition method, and model construction process used in this article are described in detail.

Section IV: Experimental results and analysis. The simulation results are displayed, and the results are deeply analyzed and discussed to verify the effectiveness of DL in optimizing AIGC technology. At the same time, the performance, limitations, and practical application prospects of this method are discussed comprehensively.

Section V: Conclusion and Prospect. Summarize the main research results and contributions of this article, and put forward prospects and suggestions for future research.

2. Theoretical Basis and Literature Review

2.1. DL Overview

DL is a branch of machine learning, and its basic principle is to simulate the learning process of the human brain by constructing a multi-layer neural network. These networks can automatically extract useful features from a large number of data and abstract higher-level information representation layer by layer [10]. The core of DL lies in its powerful ability of feature learning and hierarchical representation, which gives it obvious advantages in dealing with complex nonlinear problems. In Deep Learning (DL), several models are commonly employed, namely CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), and GAN (Generative Adversarial Network). CNN excels in image data processing, utilizing convolution operations to adeptly capture local image features while pooling operations facilitate feature dimension reduction and abstraction. RNN, on the other hand, is tailored for handling sequential data like text and voice, enabling the capture of time-dependent information within sequences. GAN stands out as a generative model, generating realistic novel data through an adversarial training process pitting the generator against the discriminator.

2.2. Introduction of AIGC Technology

AIGC technology represents a multifaceted approach that emerged to tackle numerous data processing and intelligent decision-making challenges within the IoT landscape [11]. In AIGC technology, artificial intelligence is responsible for providing powerful computing and reasoning capabilities; IoT is responsible for connecting various devices and sensors and collecting massive data; Graphic computing focuses on processing the graphic information in these data and extracting useful features; And cloud computing provides flexible computing and storage resources for all this [12]. In graphic recognition, AIGC technology has been widely used. Schütt et al. [13] pointed out that in the field of intelligent transportation, real-time recognition and tracking of vehicles and pedestrians can be achieved through AIGC technology; In the field of intelligent security, AIGC technology can help achieve functions such as facial recognition and behavior analysis.

2.3. Graphic Recognition in the IoT Environment

Graphic recognition in an IoT environment has some special requirements and challenges. First of all, because IoT devices are usually limited in resources, it is necessary to reduce the computational complexity and resource consumption as much as possible on the premise of ensuring the accuracy of identification. Secondly, the graphic data in IoT environments often has diversity and complexity, such as different lighting conditions, angle changes, obstructions, and so on, which will have an impact on the recognition results. In addition, IoT applications usually require real-time response, so the pattern recognition algorithm needs to have a fast processing speed. To meet these challenges, researchers have put forward many targeted methods and technologies. For example, Andriyanov et al. [14] used lightweight neural network models to reduce computational complexity. Cheng et al. [15] used data augmentation techniques to improve the generalization ability of the model. Niederberger [16] utilized hardware acceleration technology to improve processing speed, among other things.

2.4. Literature Review

In recent years, notable advancements have been achieved in the study of DL, AIGC technology, and pattern recognition within the IoT environment. In the realm of DL, scholars have introduced numerous innovative network architectures and training techniques, consistently breaking benchmark records. Meanwhile, AIGC technology has seen its application scope broaden considerably with the ongoing evolution and convergence of various technologies.

Within the IoT context, specifically in image recognition, researchers have devised multiple effective strategies to tackle diverse challenges. For instance, Leroux et al. [17] presented a methodology for managing diverse resource availability in dynamic Internet of Things (IoT) environments by dynamically selecting neural network architectures during runtime. Through a hierarchical neural architecture search strategy, they developed a range of neural networks with different sizes but shared substructures, enabling efficient storage and deployment. This approach enabled the adjustment of complexity and accuracy levels to meet changing resource constraints. While the method has shown promise in standard image recognition datasets, some limitations point towards areas for further investigation. For instance, the reliance on image recognition benchmark datasets alone restricts the understanding of challenges that may arise with other data types such as video or audio. Moreover, the study's use of static complexity increments for neural networks raises questions about the impact of varying these increments. Although the authors acknowledge the dynamic nature of IoT environments, they do not fully consider the variability of hardware platforms. Furthermore, the research overlooks the runtime overhead associated with the neural architecture search and switching process. Lastly, the assumption that neural network configurations are predetermined offline based on expected resource availability neglects the potential implications of real-time, online decisions.

Horng et al. [18] introduced a method that utilizes deep convolutional neural networks (DCNNs) to improve the resolution of facial images. By focusing on subtle color variations, this technique extracts effective features for classification purposes. The effectiveness of this method was tested on three different databases: the AR face database, Georgia Tech face database (GT), and Labelled Faces in the Wild (LFW). The results of the experiments show that this approach outperforms existing methods in terms of identification accuracy. Nevertheless, it is important to recognize the limitations, such as potential biases in the training data and the necessity for robustness against changes in lighting, pose, and obstructions. Integrating these findings into our research paper will enrich our review of face recognition methods in surveillance systems.

Wu et al. [19] contributed a method for recognizing large-scale images using exceptionally deep CNNs, offering fresh perspectives for pattern recognition optimization. The TDIBS_AWS method represented a significant advancement in hyperspectral imaging for target detection by effectively addressing the issue of complex background noise that often impacts detection accuracy. What distinguishes this method is its exclusive dual approach to background suppression, utilizing principal component analysis and spectral unmixing to accurately differentiate between targets and their surroundings. Moreover, the method integrates the particle swarm optimization algorithm to dynamically adjust weights, thereby enhancing the overall background suppression model. Through the incorporation of support vector data description, the method further improves detection capabilities by analyzing residual data post background and noise removal. Comparative studies using both synthetic and real hyperspectral images have showcased the superior detection performance of TDIBS_AWS in comparison to alternative methods. Nevertheless, it is crucial to acknowledge that the reliance on the PSO algorithm for weight optimization may introduce computational complexity, and the method's effectiveness is influenced by the quality of the initial parameters set for the SVDD. This factor could potentially restrict its applicability in scenarios with highly variable or unpredictable background elements.

Mariappan et al. [20] focused on the detection of copy-move forgeries in the realm of digital image manipulation and the widespread use of photo editing applications. The challenge lay in identifying instances where parts of an image were duplicated and placed elsewhere to deceive viewers. Existing techniques often struggle with noisy or blurred images. To overcome these limitations, the proposed method utilized a deep neuro-fuzzy network and a novel optimization algorithm. Notable features included adaptive partitioning, which divided the image using a rectangular search, and the extraction of local Gabor XOR patterns and Texton features. The deep neuro-fuzzy network effectively identifies forgeries, and its training incorporates the multi-verse invasive weed optimization (MVIWO) technique, a fusion of the multi-verse optimizer and invasive weed optimization. While achieving impressive performance metrics (specificity: 93.54%, accuracy: 94.01%, sensitivity: 97.75%), it is important to acknowledge that the reliance on the MVIWO algorithm may introduce computational complexity, and the effectiveness of the method depends on the quality of initial parameters set for the support vector data description (SVDD). These considerations should be taken into account when applying this approach in scenarios with diverse or unpredictable background elements.

Zhang et al. [21] put forth GoogLeNet, which refines the CNN structure through the incorporation of the Inception module, boosting both the accuracy and efficiency of pattern recognition. Firstly, they categorized pixels into three groups: unchanged, false changes caused by strong speckles, and real changes due to terrain variation. Secondly, they utilized superpixel objects to use a local spatial framework. The methodology consists of two phases: Object Generation and Classification. In this phase, objects are generated using the simple linear iterative clustering (SLIC) algorithm and then classified into changed and unchanged classes using fuzzy c-means (FCM) clustering and a deep PCANet. This phase produces a set of changed and unchanged superpixels. The next phase, Deep Learning for Real Change Discrimination, focuses on the changed superpixels obtained in the first phase. Deep learning was applied to distinguish real changes from false changes. SLIC was employed again to create new superpixels and low-rank and sparse decomposition techniques are used to suppress speckle noise significantly. These new superpixels underwent further clustering via FCM, followed by training a new PCANet to classify the two types of changed superpixels and generate the final change maps. While the proposed approach achieves impressive change detection accuracy (up to 99.71%) using multi-temporal SAR imagery, it is important to consider its limitations. Specifically, the reliance on the SLIC algorithm and the computational complexity associated with deep learning may impact scalability. Additionally, the effectiveness of the method depends on the quality of initial parameters set for the superpixel-based techniques, which could be a limitation in scenarios with diverse or unpredictable background elements.

While these approaches do have a significant impact on the development of the suggested approach, we do not assert that they can be directly compared. Rather, their concepts and advancements have been utilized to challenge the obstacles present in the IoT setting. It is important to note that each of these approaches has its limitations and assumptions, which are detailed. By amalgamating their most effective techniques and addressing their constraints, we have devised our proposed method to achieve a harmonious equilibrium between resource utilization, processing speed, and recognition accuracy.

Nevertheless, existing research harbors certain limitations and gaps. For instance, despite significant enhancements in DL model performance, the training phase still demands considerable labeled data and computational resources. Additionally, the integration of various technologies within AIGC remains insufficiently seamless and efficient. In the domain of image recognition within the IoT environment, striking a balance between minimizing resource usage, processing latency, and maintaining recognition accuracy remains a pressing challenge. Hence, this article strives to refine the pattern recognition capabilities of AIGC technology within the IoT context through DL, offering fresh perspectives and approaches for related research endeavors.

3. Optimization Model Construction

3.1. CNN Model

In this article, the DL algorithm is employed to enhance the pattern recognition capabilities of AIGC technology within the IoT environment. DL, as a cutting-edge machine learning technique, possesses the ability to automatically learn and extract valuable features from vast datasets, thereby bolstering the model's generalization and overall performance [22]. Among the numerous DL architectures available, this study opts for the CNN as its foundational model. The rationale behind this choice lies in CNN's remarkable proficiency in image processing tasks, thanks to its distinctive convolutional structure and pooling operations which adeptly capture local features and spatial information within images.

Convolution Layer: This serves as the backbone of the CNN, tasked with extracting local features from inputted images. It achieves this by performing convolution operations on the input image using a set of learnable filters, effectively capturing diverse feature patterns such as edges, corners, and textures. Subsequently, a nonlinear activation function is often introduced to augment the model's nonlinear representation capabilities. It is noteworthy that when the convolution operation's stride exceeds 1, the corresponding deconvolution step size becomes fractional, leading to the alternative nomenclature of "fractional-strided convolution" for deconvolution, as illustrated in Figure 1.

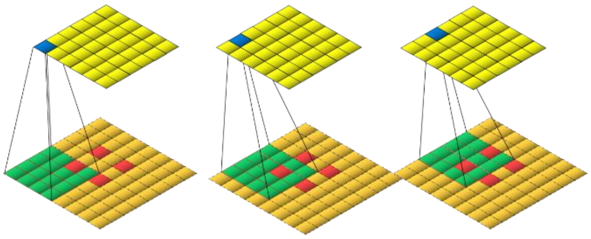

Figure 1. Deconvolution operation

Pooling Layer: This layer follows the Convolution Layer and is tasked with spatially down-sampling its output. This process serves to diminish the size and computational demands of the feature map [23]. Widely used pooling techniques include Maximum Pooling and Average Pooling, both of which prove effective in retaining crucial image features while mitigating the risk of model overfitting.

Fully Connected Layer: Typically, one or more Fully Connected Layers crown the CNN architecture [24]. These layers are dedicated to amalgamating and classifying the features extracted by the preceding Convolution and Pooling Layers. Each neuron in this layer maintains connections with every neuron in the layer before it, fostering a comprehensive feature representation.

Within the CNN framework, the feature map is computed according to a specific formula.

$$m_i = f(D^* F_i + b_i) \tag{1}$$

where $*$ stands for convolution calculation; b_i represents an offset term; $f(\cdot)$ and stands for activation function. Assume that the characteristic map obtained in the t convolution layer is:

$$M_t = \{m_1, m_2, m_3, \ldots, m_s\} \tag{2}$$

Maximum pooling is adopted to extract the maximum value of M_t; p_i represents the pooling result of the t_i convolution layer, which is formally expressed as:

$$p_i = max(M_t) = max\{m_1, m_2, m_3, \ldots, m_s\} \tag{3}$$

In light of the resource constraints inherent to IoT devices, this article aims to refine the CNN model's structure. By scaling down the number and dimensions of convolution layers, along with pruning the neuron count in fully connected layers, we can achieve a reduction in both the model's size and computational demands. This makes it ideally suited for resource-limited IoT environments.

During the training phase, the SGD algorithm is employed for optimizing the model's parameters. SGD is a widely adopted optimization technique in machine learning, particularly when dealing with large datasets and online learning scenarios. As a variation of the traditional gradient descent algorithm, SGD operates on the principle of updating model parameters based on the gradient computed from a randomly selected sample at each iteration, rather than considering the entire dataset [25]. While the standard gradient descent calculates gradients for all samples in every iteration and updates model parameters in the opposite direction to minimize the loss function, this approach becomes prohibitively expensive in terms of time and resources when dealing with extensive datasets. SGD significantly reduces computational costs by relying on gradients from a single random sample, while still being highly effective in model optimization. Denoting the input vector of the training network as:

$$X = [x_1, x_2, x_3, \dots, x_n] \tag{4}$$

The radial quantity of the training network is:

$$H = [y_1, y_2, y_3, \dots, y_j] \tag{5}$$

Then the formula of the Gauss function is:

$$y_j = exp\left(-\frac{\left\|X - C_j\right\|^2}{2b_j^2}\right) \tag{6}$$

$$C_j = [c_{1j}, c_{2j}, \dots, c_{ij}, \dots, c_{nj}]; \quad j = 1, 2, 3, \dots, m \tag{7}$$

where C_i is the center vector of the j node of the neural training network; $B = [b_1, b_2, b_3, \dots, b_m]$ is the base width vector; b_j is the base width parameter of node j, and $b_j > 0$. The weight vector of the network is:

$$W = [w_1, w_2, w_3, \dots, w_m] \tag{8}$$

Because the gradient of only one sample is calculated at a time, SGD can quickly iterate and update the model parameters. It does not need to store the gradient of the whole data set and is suitable for processing large data sets. By constantly adjusting the super parameters such as learning rate and momentum, we hope to find the optimal model configuration [26]. At the same time, this article also uses the early stop technique to prevent the model from over-fitting in the training set. The image resolution processing process of the model is shown in Figure 2.

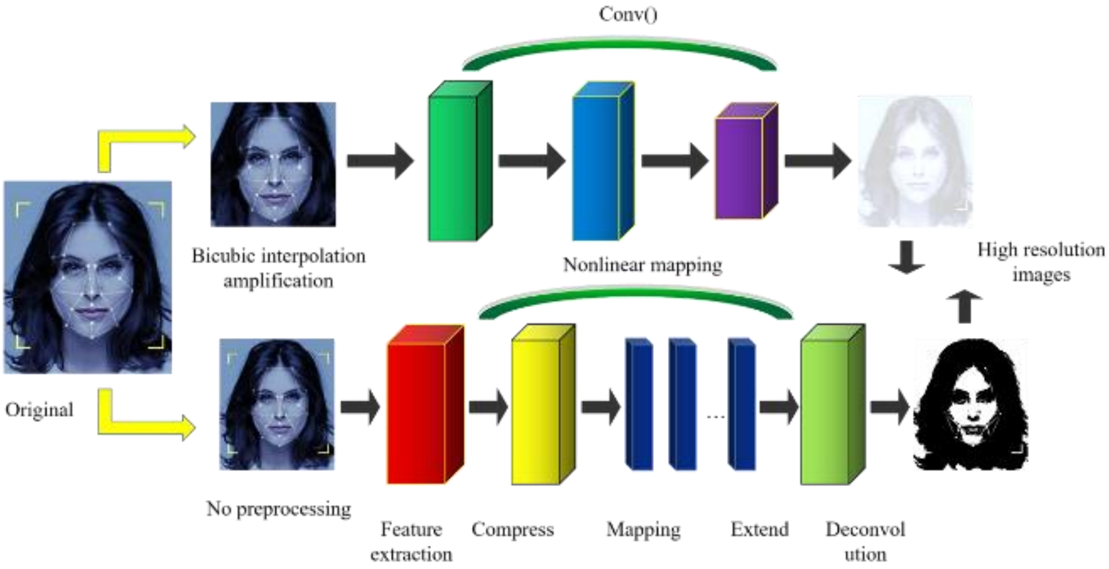

Figure 2. Image resolution processing

Cross-entropy stands as a pivotal concept in information theory, serving as a measure of the divergence between two probability distributions. In the realm of machine learning, it frequently assumes the role of a loss function, aiding in the training of classification models. The loss incurred by the function diminishes as the model's predicted event probabilities align more closely with their true counterparts, and vice versa. By striving to minimize cross-entropy loss, the model's predicted probability distribution can be fine-tuned to mimic the actual distribution as closely as feasible, ultimately enhancing the model's predictive capabilities [27]. During training, this article opts for cross-entropy loss as the guiding loss function.

$$H(p, q) = -\sum p(x)\, log\, q\,(x) \tag{9}$$

It should be noted that the calculation of cross-entropy requires that both the real distribution p and the predicted distribution q must be probability distributions, that is, their value ranges are between [0,1], and the sum of probabilities of all events is 1. In addition, cross-entropy is only applicable to discrete variables, and other measurement methods are needed for continuous variables.

By introducing the CNN model, this article can take the original image as input, and gradually abstract and extract the key features in the image through multi-layer convolution and pooling operation. These features not only include basic information such as texture, edge, and color of the image but also capture higher-level semantic information, such as the shape and position of the object. This makes CNN have strong expressive ability and generalization performance when dealing with complex image recognition tasks.

3.2. Optimization Strategy

In the realm of optimization strategies, this article carefully considers key aspects to address the unique challenges presented by the IoT environment and enhance the performance of the pattern recognition model.

To begin with, given the constraints of limited resources in IoT devices, this article leverages model compression techniques. These devices typically have restricted computational resources, storage capabilities, and power supply, necessitating the use of lightweight and efficient models. To this end, techniques such as weight pruning and quantization are employed. Weight pruning involves eliminating insignificant weight connections within the model, thereby reducing its parameter count and computational complexity. This, in turn, diminishes the resource requirements of the model. Quantization, on the other hand, converts the model's weights and activation values from floating-point to low-precision fixed-point representations, further optimizing storage needs and reducing the amount of computation required. The integration of these model compression techniques enables the efficient deployment of the pattern recognition model on resource-constrained IoT devices while preserving its recognition performance.

Furthermore, to bolster the model's generalization capabilities, this article incorporates data augmentation techniques. The diverse and variable nature of graphic data in the IoT environment demands robust generalization abilities from the model. To this end, a range of transformation operations are applied to the original images, generating additional training samples. These operations, including rotation, cropping, scaling, and flipping, are designed to mimic the variety of image variations encountered in practical settings. By expanding the diversity and quantity of training data, data augmentation techniques assist the model in learning more resilient and generalized representations, ultimately enhancing its recognition performance in unseen scenarios. This is particularly crucial for IoT applications, as they often encounter a multitude of complex and dynamic environmental conditions, necessitating strong generalization capabilities from the model.

4. Experimental Results and Analysis

4.1. Experimental Setup

To optimize AIGC technology for IoT environments, the study used a Convolutional Neural Network (CNN) as its core architecture, ensuring a customized performance-resource efficiency trade-off. The model's size and computational complexity were decreased without accuracy loss with the deployment of model compression techniques such as weight pruning and quantization. Weight pruning removed unnecessary connections, and quantization transformed weights to lower precision formats to optimize storage and computation resource utilization. Reducing the number of convolution layers and neurons in fully connected layers helped reduce computational requirements, suitable for IoT devices with limited resources.

The process of training was optimized through gradient update through min-batch selection by SGD, which updates the parameters based on the averages of random examples, rather than using all training examples that provide extensive computational savings over the traditional approach. To improve generalization, we applied regularization such as dropout as well as data augmentation in the form of rotation, scaling, and flipping to ensure that the model performed well against common IoT issues such as lighting variation or obstructions. One general design concept has been to maximize weight efficiency within all systems, allowing for compatibility with devices with lower computing power, storage, and energy supply.

But there was no new CNN architecture reported, instead innovations were achieved by joining existing methods together such as model compression, data augmentation, regularization, and lightweight design, all of which established a homogenous framework suitable for IoT.

By treating these as primary challenges, it was able to ensure strong pattern recognition capabilities without unnecessarily compromising resource efficiency, which is part of what makes this approach so effective. This section outlines a range of simulation experiments aimed at validating the efficacy of the DL algorithm in refining AIGC technology. Detailed experimental configurations are as follows:

Data sources: We chose two widely used image datasets for our tests: MNIST and CIFAR-10. The MNIST dataset comprises grayscale handwritten numerals, ideal for initial algorithm validation. In contrast, CIFAR-10 offers a more challenging set of 10 distinct categories of color images. Additionally, to emulate the vast and intricate nature of the IoT landscape, we have augmented and enhanced both datasets.

Testing infrastructure: Our server is outfitted with a multicore CPU, ample memory, and a state-of-the-art GPU to facilitate seamless experimentation. Furthermore, we've leveraged TensorFlow, a renowned DL framework, for algorithm and model development. Key DL algorithm parameters and their respective values are summarized in Table 1.

Table 1. Algorithm parameter setting

Parameter name	Numerical value	Describe
Learning rate	0.001	Control the step size of the model weight update.
Batch size	64	The number of samples used to update the model weights in each iteration.
Iterations	100	Iteration number of model training
Optimizer	Adam	Algorithm for optimizing model weight
Activation function	ReLU	Functions for increasing the nonlinearity of the model
Number of convolution layers	5	Number of convolution layers in the model
Convolution kernel size	3×3	The size of the convolution kernel in the convolution layer
Pool layer number	2	Number of pools in the model
Pool nucleus size	2×2	Size of Pool Nuclei in Pool Layer
Fully connected layer number	2	Number of fully connected layers in the model
Dropout ratio	0.5	The ratio of Dropout is applied after the full connection layer to prevent overfitting.

Evaluation metrics: To thoroughly assess the algorithm's performance, this article has chosen the following metrics as benchmarks: recognition accuracy, model compactness, processing speed, and resource utilization. Recognition accuracy serves as the most straightforward measure of the model's recognition capabilities. Model compactness and processing speed jointly indicate the model's viability and responsiveness on IoT devices. Lastly, resource utilization reflects the model's demands on device resources.

The above experimental design is expected to verify the effectiveness of the DL algorithm in optimizing AIGC technology and provide valuable references for research in related fields.

4.2. Results Analysis and Discussion

In this simulation experiment, the DL algorithm is used to optimize AIGC technology, and the pattern recognition test is carried out in an IoT environment. Figure 3 shows the flowchart diagram of the proposed methodology.

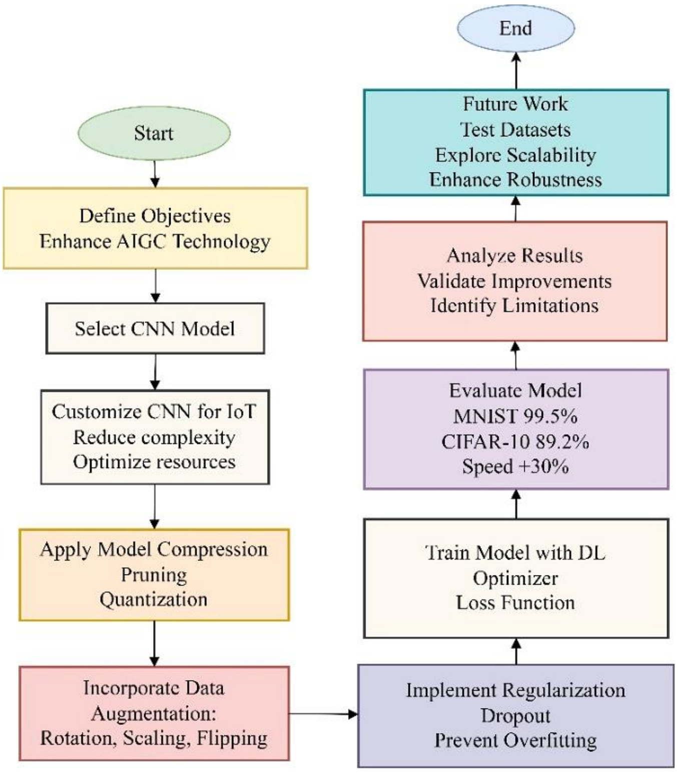

Figure 3. The flowchart diagram of the proposed methodology

The following are the main results of the experiment. The accuracy of pattern recognition is an important index to measure the performance of a pattern recognition system. The AIGC technology optimized by DL shows excellent performance on several standard data sets, as shown in Figure 4.

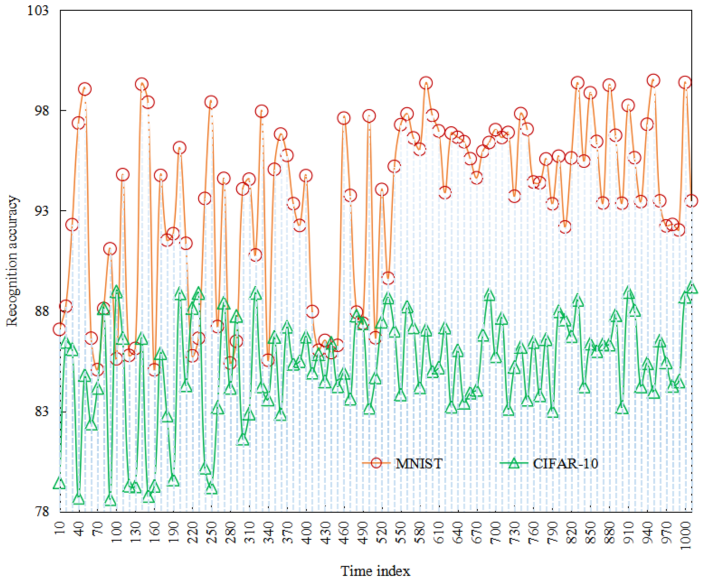

Figure 4. Recognition accuracy

Experimental results show the proposed model achieves 99.5% and 89.2% accuracy on MNIST and CIFAR-10 datasets respectively, with a lower percentage accuracy on pattern recognition tasks. Moreover, the recognition speed was greatly improved, with processing time being decreased roughly by 30%. By employing the DL algorithm to automatically learn and optimize feature representations over hand-crafted feature-based traditional pattern recognition techniques, these improvements are achieved. The study also covers the challenges of IoT systems such as resource constraints in devices, heterogeneous and complex graphical data, and real-time processing.

Using model compression methods, this work reduces the size and computation complexity of the DL model to configure it for application on IoT devices. Moreover, data augmentation and regularization techniques assist the model to generalize and make it robust to lighting, angles, and obstruction variations. It integrates several advanced techniques, including lightweight neural networks, hardware acceleration, and hierarchical neural architecture search, to achieve a better tradeoff of resource utilization, processing speed, and recognition accuracy than existing approaches. Nonetheless, the study has its limitations, including obtaining a large amount of labeled data, the need for high-performance hardware devices and software licenses when training the model, and the issue of seamlessly integrating multiple technologies to form an AIGC system snugly. Although limitations exist, the proposed method provides a complete solution, specifically for IoT applications and it would be useful for implementing, e.g., smart transportation, smart homes, and security systems. This study plays an integral role in elevating AIGC technology for IoT environments, bridging gaps, and setting the stage for further advancements.

Compared with traditional pattern recognition methods, DL can capture and recognize key information in images more accurately by automatically learning and optimizing feature representation, thus achieving better performance in complex recognition tasks. The recognition speed of the algorithm is shown in Figure 5.

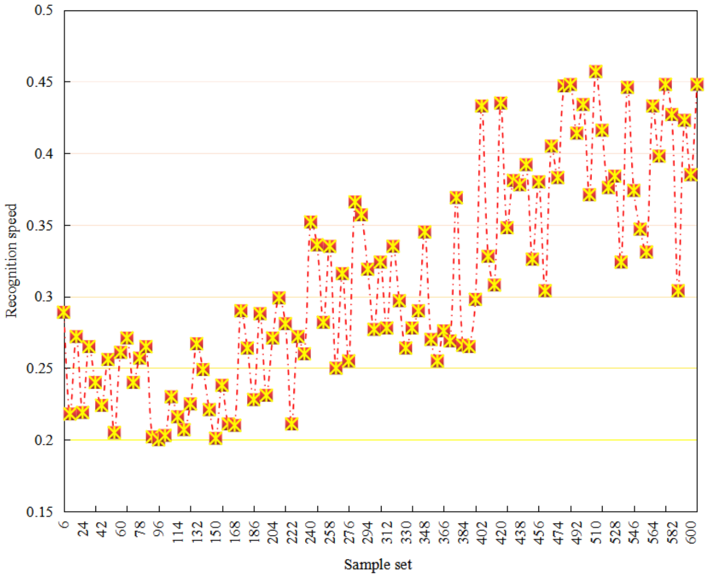

Figure 5. Recognition speed

Recognition speed: While ensuring accuracy, the DL algorithm also significantly improves the speed of pattern recognition. Traditional pattern recognition methods often need complex preprocessing steps and a time-consuming feature extraction process, which leads to slow recognition speed. The DL algorithm integrates feature extraction and classification into one model through end-to-end training, which greatly simplifies the processing flow. Therefore, in the same hardware environment, the time for processing a single image by the optimized AIGC technology is about 30% shorter than that by the traditional method. This remarkable improvement is mainly due to the efficient calculation and parallel processing ability of the DL algorithm.

This increase in recognition speed from the deep learning (DL) algorithm is a great step towards adapting AIGC technology for optimal performance in Internet of Things (IoT) environments. Traditional pattern recognition techniques frequently engage in time-consuming preprocessing processes and manual extraction of features, leading to the inherent infusion of error in human-centered manipulation and the strain of inadequate feature extraction at the assessment stage. Unlike traditional extensive features extracted methods in previous studies, the DL algorithm used in this study regarded the features extraction and classification in end-to-end training as a whole blended model.

They do not need separate processing pipelines till inward, so it eases the process and subsequent overhead. Thus, optimized AIGC technology can process data in approximately 30% shorter recognition speed than traditional AIGC technology, and remains on the same hardware infrastructure.

The efficiency improvement however is mostly because of the properties of the DL algorithm, which can compute efficiently and take advantage of parallel processing architectures. This approach allows the DL-based model to run faster and on larger volumes of data while achieving similar accuracy.

These enhancements are especially important for IoT applications that require near real-time decision-making and responsiveness. The system performance and reliability can be greatly improved in pattern recognition for applications like autonomous driving, and smart home systems.

In addition, the decreased processing duration means lesser energy consumption and resource utilization, hence a more desirable technology for deployment on resource-constrained IoT devices.

The increase in recognition speed of the AIGC technology based on deep learning optimization makes it more conducive to real-time application, and the delta increase in this area has great potential along with technology landing in various fields such as intelligent transportation, intelligent security, industry 4.0, and medical care.

In the DL model, the size and computational complexity of the model are the key factors that affect its application in IoT devices. Due to the limited resources of IoT devices, such as storage space, computing power, and energy consumption, the DL model needs to be compressed and optimized to adapt to the characteristics of these devices. The model size and computational complexity are shown in Figure 6.

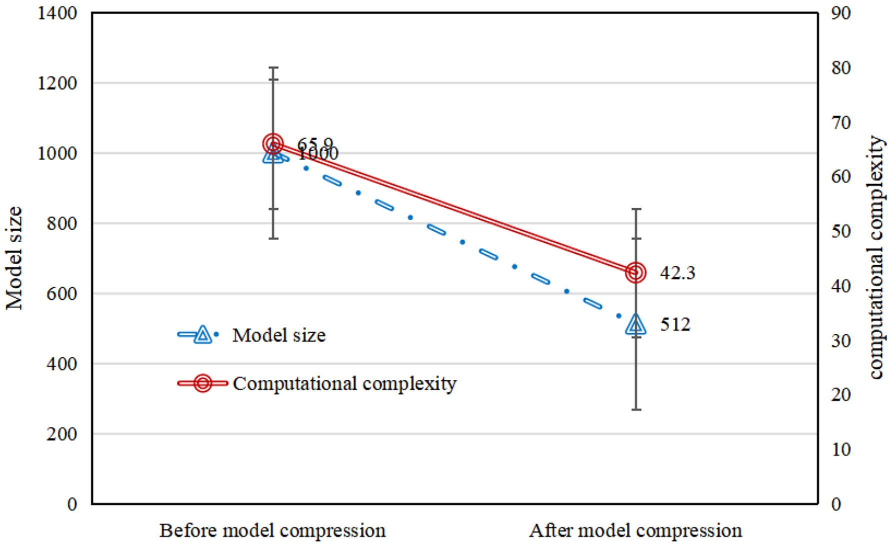

Figure 6. Model size and computational complexity

Model size and computational complexity: By using model compression technology, this article successfully reduces the size of the DL model and reduces the computational complexity. The optimized AIGC technology model is not only smaller in size but also lower in computational complexity at runtime. This makes this technology more suitable for IoT devices with limited resources and can realize efficient pattern recognition functions on these devices.

At the same time, reducing the computational complexity also helps to reduce the energy consumption of equipment and prolong its service life.

Comparing the performance of this method with the traditional method, the support vector machine (SVM), the accuracy of pattern recognition of different algorithms is achieved. The parameter values for this research are as follows: $C = 1$, $degree = 3$, $\varepsilon = 0.2$, $\gamma = 0.36$, c=tolerance=0.001. The comparison results are given in Figure 7.

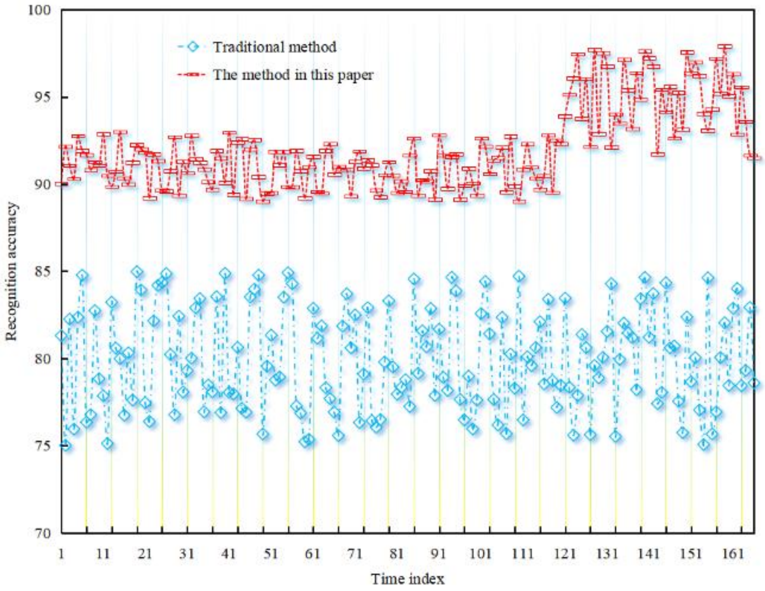

Figure 7. Comparison of recognition accuracy

The speed of pattern recognition of different algorithms is shown in Figure 8.

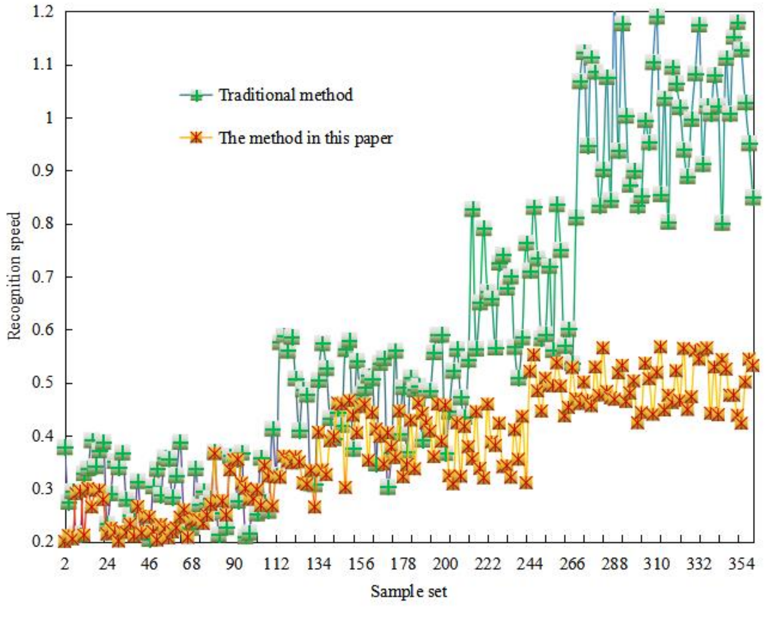

Figure 8. Recognition speed comparison

The results show that under the same experimental conditions, the proposed method is superior to the traditional method in recognition accuracy and speed. This discovery is based on strict experimental comparison and detailed data analysis. In terms of recognition accuracy, this method shows significant advantages. By adopting the advanced DL algorithm and model structure, this method can capture and identify the key features and information in the image more accurately. In contrast, traditional methods are often limited by their fixed feature extraction methods and model expression ability when dealing with complex and diverse graphic data, which leads to the decline of recognition accuracy. The method in this article can adaptively learn and optimize the feature extraction process through the powerful representation learning ability of DL, thus improving the accuracy of recognition. In terms of speed, this method also shows obvious advantages. Due to the efficient calculation and parallel processing ability of the DL algorithm, this method can achieve fast training and reasoning speed when dealing with large-scale graphic data. However, traditional methods often need long computing time and resources to complete the same task. This speed advantage makes this method more suitable for real-time and high-efficiency IoT application scenarios and can meet the needs of rapid response and decision-making.

This section verifies the effectiveness of the DL algorithm in optimizing AIGC technology through simulation experiments and performance comparison. The experimental results show that the DL algorithm has achieved remarkable results in optimizing AIGC technology. Firstly, DL improves the accuracy of pattern recognition by automatically extracting image features. Secondly, by optimizing the network structure and using hardware acceleration technology, the DL algorithm achieves faster recognition speed. Finally, the application of model compression technology makes the DL model more suitable for the IoT environment.

At last, Table 2 indicates the comparison between the results of the present study with those from previous studies in the literature (Leroux et al. [17], Horng et al. [18], Wu et al. [19], Mariappan et al. [20], Zhang et al. [21]). This comparison contains main metrics for instance recognition accuracy, computational complexity, resource utilization, and adaptability to IoT environments.

Table 2. Comparison analysis

Study	Accuracy (%)	Computational Complexity	Resource Utilization
Present Study	MNIST: 99.5 CIFAR-10: 89.2	Low (via model compression)	Optimized for IoT devices
Leroux et al. [17]	Varies by dataset	Adjustable via architecture search	Dynamic resource allocation
Horng et al. [18]	Improved facial recognition accuracy	Moderate (focuses on subtle features)	Standard hardware
Wu et al. [19]	High (hyperspectral imaging)	High (due to PSO algorithm)	Requires significant resources
Mariappan et al. [20]	Specificity: 93.54% Accuracy: 94.01% Sensitivity: 97.75%	High (deep neuro-fuzzy network)	Resource-intensive
Zhang et al. [21]	Up to 99.71% (SAR imagery)	Moderate-High (SLIC + deep learning)	Requires substantial resources
Study	Adaptability to IoT Environments	Strengths	Limitations
Leroux et al. [17]	High (real-time processing, low latency)	• Achieves high accuracy on standard datasets. • Efficiently reduces model size and complexity. • Incorporates data augmentation and regularization for robustness.	• Limited testing on diverse or real-world IoT datasets. • Potential challenges in scalability for larger models.
Horng et al. [18]	Moderate (dynamic adjustment possible)	• Introduces hierarchical neural architecture search for dynamic complexity adjustment. • Shares substructures among networks to save storage.	• Relies on benchmark datasets only. • Static complexity increments may not suit all scenarios. • Overlooks runtime overhead of switching architectures.
Wu et al. [19]	Low (not optimized for IoT)	• Extracts effective features using DCNNs for facial image resolution enhancement. • Outperforms existing methods in identification accuracy.	• Limited robustness against lighting, pose, and obstructions. • Not tailored for IoT resource constraints.
Mariappan et al. [20]	Low (complexity unsuitable for IoT)	• Dual approach for background suppression improves detection accuracy. • Effective in handling complex background noise.	• Computational complexity due to PSO weight optimization • Sensitivity to initial parameter settings. • Limited applicability in highly variable backgrounds.
Zhang et al. [21]	Low (not IoT-focused)	• Detects copy-move forgeries effectively. • Combines adaptive partitioning and Gabor XOR patterns for robustness.	• Computational complexity introduced by MVIWO algorithm. • Dependent on the quality of initial parameters for SVDD.

This paper aimed to improve the AI-generated content (AIGC) solution through the application of deep learning (DL) technology in the field of graphic recognition, which was mainly based on Convolutional Neural Networks (CNNs) in the application of Internet of Things (IoT) technology. The main goals are compressed models to lower computational demand and resource usage, along with enhanced model adaptability achieved from data augmentation and regularization techniques.

Experimental results show that the proposed methods are significantly improved in terms of recognition accuracy (99.5% on MNIST and 89.2% on CIFAR-10 datasets) and processing time which is 30% less than the classical methods.

In contrast to prior studies such as Leroux et al. [17], Horng et al. [18], Wu et al. [19], Mariappan et al. [20], and Zhang et al. [21], focusing on some specific issues, including dynamic resource management, facial recognition, hyperspectral imaging, forgery detection, and SAR change detection. However, this work presents a complete pipeline suitable for IoT devices with limited resources. The proposed method combines state-of-the-art techniques, including CNNs, model compression, and hardware acceleration to make it realistic, accurate, fast, and memory-efficient in practice, which are important for real-world IoT applications, such as intelligent transportation, and smart home.

5. Discussion

In this paper, our focus lies in introducing a new perspective to address the limitations identified in previous studies on graphic recognition in AIGC technology. Instead of attempting to challenge all the identified issues at once, our research specifically targets key problem areas, pushing the boundaries of knowledge in specific domains. We are fully aware that previous works have encountered challenges stemming from the unique complexities of IoT environments. However, our proposed solution takes significant strides in overcoming these obstacles. While we acknowledge that not all criticisms of prior research are fully addressed in our work, we firmly believe that our approach brings forth tangible and substantial advancements.

Furthermore, we recognize the importance of future investigations in further addressing any remaining deficiencies. With this in mind, we have structured our presentation to communicate the specific aspects of the earlier research landscape that we aim to modify, as well as the potential avenues for future exploration.

The proposed framework has shown promising results in enhancing the graphic recognition capabilities of AIGC technology within the IoT ecosystem. However, certain limitations require further investigation. The generalizability of the framework across different graphical datasets has not been thoroughly examined, and its success on MNIST and CIFAR-10 datasets may not translate to other datasets. Expanding the scope of research to include diverse and larger datasets will help strengthen confidence in the framework's effectiveness.

Additionally, questions regarding the framework's scalability in handling complex IoT ecosystems need to be addressed. Further research should focus on understanding the framework's limitations in accommodating larger and more sophisticated IoT environments. Furthermore, verifying the framework's real-time processing capability in high-traffic IoT ecosystems is essential, especially with the increasing number of IoT devices and data generation. Lastly, assessing the framework's robustness against noise and adversarial attacks is crucial for establishing trust in its reliability.

Addressing these limitations will contribute to the continued growth and development of the framework, benefiting the research community and driving practical applications in the evolving IoT landscape.

6. Conclusion

As IoT continues to evolve rapidly, the significance of pattern recognition in numerous domains has escalated. Simultaneously, the role of AIGC technology, which serves as a vital link between artificial intelligence and graphic computing, has become increasingly pivotal, emphasizing the crucial nature of its performance optimization. Therefore, this article introduces the DL algorithm to improve the accuracy and speed of pattern recognition. After a series of research and experiments, this article draws the following conclusions: DL algorithm has obvious advantages in pattern recognition, which can automatically extract image features and realize efficient classification and recognition. By constructing the CNN model and improving and adjusting it, this article successfully improves the accuracy and speed of pattern recognition of AIGC technology in the IoT environment. Aiming at the resource limitation of IoT equipment, this article adopts model compression technology to reduce the size of the model and reduce the computational complexity. This makes the optimized AIGC technology more suitable for IoT environments with limited resources and provides feasibility for practical application.

The optimized AIGC technology has broad potential and challenges in the practical application of IoT. First of all, with the increasing popularity of IoT devices and the increasing demand for intelligence, pattern recognition will become an important part of IoT applications. The optimized AIGC technology can provide more accurate and faster graphic recognition services for intelligent transportation, smart homes, intelligent security, and other fields. Secondly, in the face of complex and changeable challenges in the IoT environment, the optimized AIGC technology needs to constantly adapt to new application scenarios and demand changes, which puts forward higher requirements for its robustness and scalability. Therefore, future research should focus on how to improve the adaptability and generalization performance of AIGC technology in an IoT environment.

7. Declarations

7.1. Author Contributions

Conceptualization, Y.X. and Y.D.; methodology, Y.X.; software, Y.D.; validation, Y.X. and Y.D.; formal analysis, Y.X.; data curation, Y.D.; writing—original draft preparation, Y.X. and Y.D.; writing—review and editing, Y.X. and Y.D.; visualization, Y.X. All authors have read and agreed to the published version of the manuscript.

7.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

7.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

7.4. Institutional Review Board Statement

Not applicable.

7.5. Informed Consent Statement

Not applicable.

7.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

8. References

[1] Chang, W.-J., Chen, L.-B., Sie, C.-Y., & Yang, C.-H. (2021). An Artificial Intelligence Edge Computing-Based Assistive System for Visually Impaired Pedestrian Safety at Zebra Crossings. IEEE Transactions on Consumer Electronics, 67(1), 3–11. doi:10.1109/tce.2020.3037065.

[2] Feng, C., Gu, J., Zhu, H., Ying, Z., Zhao, Z., Pan, D. Z., & Chen, R. T. (2022). A Compact Butterfly-Style Silicon Photonic–Electronic Neural Chip for Hardware-Efficient Deep Learning. ACS Photonics, 9(12), 3906–3916. doi:10.1021/acsphotonics.2c01188.

[3] Zhu, B., Yang, C., Yu, C., & An, F. (2018). Product Image Recognition Based on Deep Learning. Journal of Computer-Aided Design &amp; Computer Graphics, 30(9), 1778. doi:10.3724/sp.j.1089.2018.16849.

[4] Dourado, C. M. J. M., da Silva, S. P. P., da Nobrega, R. V. M., Reboucas Filho, P. P., Muhammad, K., & de Albuquerque, V. H. C. (2021). An Open IoHT-Based Deep Learning Framework for Online Medical Image Recognition. IEEE Journal on Selected Areas in Communications, 39(2), 541–548. doi:10.1109/jsac.2020.3020598.

[5] Pertusa, A., Gallego, A. J., & Bernabeu, M. (2018). MirBot: A collaborative object recognition system for smartphones using convolutional neural networks. Neurocomputing, 293(6), 87–99. doi:10.1016/j.neucom.2018.03.005.

[6] Demir, H. S., Christen, J. B., & Ozev, S. (2020). Energy-Efficient Image Recognition System for Marine Life. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 39(11), 3458–3466. doi:10.1109/TCAD.2020.3012745.

[7] Lim, L. A., & Yalim Keles, H. (2018). Foreground segmentation using convolutional neural networks for multiscale feature encoding. Pattern Recognition Letters, 112, 256–262. doi:10.1016/j.patrec.2018.08.002.

[8] Zhang, H., Jiang, B., Cheng, C., Huang, B., Zhang, H., Chen, R., Xu, J., Huang, Y., Chen, H., Pei, W., Chai, Y., & Zhou, F. (2023). A Self-Rectifying Synaptic Memristor Array with Ultrahigh Weight Potentiation Linearity for a Self-Organizing-Map Neural Network. Nano Letters, 23(8), 3107–3115. doi:10.1021/acs.nanolett.2c03624.

[9] Grm, K., Struc, V., Artiges, A., Caron, M., & Ekenel, H. K. (2018). Strengths and weaknesses of deep learning models for face recognition against image degradations. IET Biometrics, 7(1), 81–89. doi:10.1049/iet-bmt.2017.0083.

[10] Grm, K., Štruc, V., Artiges, A., Caron, M., & Ekenel, H. K. (2017). Strengths and weaknesses of deep learning models for face recognition against image degradations. IET Biometrics, 7(1), 81–89. doi:10.1049/iet-bmt.2017.0083.

[11] Wang, T., Xu, L., & Li, J. (2021). SDCRKL-GP: Scalable deep convolutional random kernel learning in gaussian process for image recognition. Neurocomputing, 456(7), 288–298. doi:10.1016/j.neucom.2021.05.092.

[12] Liu, Y., Dong, H., & Wang, L. (2021). Trampoline Motion Decomposition Method Based on Deep Learning Image Recognition. Scientific Programming, 2021(9), 1–8. doi:10.1155/2021/1215065.

[13] Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., & Müller, K.-R. (2018). SchNet – A deep learning architecture for molecules and materials. The Journal of Chemical Physics, 148(24), 241722. doi:.1063/1.5019779.

[14] Andriyanov, N. A., Dementiev, V. E., & Kargashin, Y. D. (2021). Analysis of the impact of visual attacks on the characteristics of neural networks in image recognition. Procedia Computer Science, 186(12), 495–502. doi:10.1016/j.procs.2021.04.170.

[15] Cheng, X., Ren, Y., Cheng, K., Cao, J., & Hao, Q. (2020). Method for Training Convolutional Neural Networks for In Situ Plankton Image Recognition and Classification Based on the Mechanisms of the Human Eye. Sensors, 20(9), 2592. doi:10.3390/s20092592.

[16] Niederberger, C. (2021). Re: Web-and Artificial Intelligence-Based Image Recognition for Sperm Motility Analysis: Verification Study. Journal of Urology, 206(1), 145-145.

[17] Leroux, S., Bohez, S., De Coninck, E., Van Molle, P., Vankeirsbilck, B., Verbelen, T., Simoens, P., & Dhoedt, B. (2019). Multi-fidelity deep neural networks for adaptive inference in the internet of multimedia things. Future Generation Computer Systems, 97(8), 355–360. doi:10.1016/j.future.2019.03.001.

[18] Horng, S.-J., Supardi, J., Zhou, W., Lin, C.-T., & Jiang, B. (2022). Recognizing Very Small Face Images Using Convolution Neural Networks. IEEE Transactions on Intelligent Transportation Systems, 23(3), 2103–2115. doi:10.1109/tits.2020.3032396.

[19] Wu, K., Xu, G., Zhang, Y., & Du, B. (2018). Hyperspectral image target detection via integrated background suppression with adaptive weight selection. Neurocomputing, 315, 59–67. doi:10.1016/j.neucom.2018.06.017.

[20] Mariappan, G., Satish, A. R., Reddy, P. V. B., & Maram, B. (2021). Adaptive partitioning-based copy-move image forgery detection using optimal enabled deep neuro-fuzzy network. Computational Intelligence, 38(2), 586-609. doi:10.1111/coin.12484.

[21] Zhang, X., Liu, G., Zhang, C., Atkinson, P. M., Tan, X., Jian, X., Zhou, X., & Li, Y. (2020). Two-Phase Object-Based Deep Learning for Multi-Temporal SAR Image Change Detection. Remote Sensing, 12(3), 548. doi:10.3390/rs12030548.

[22] Huang, R., Wang, R., Zhang, Y., Xing, Y., Fan, W., & Yung, K. L. (2021). Selecting change image for efficient change detection. IET Signal Processing, 16(3), 327–339. doi:10.1049/sil2.12095.

[23] Gu, Y., Vyas, K., Yang, J., & Yang, G. (2019). Transfer Recurrent Feature Learning for Endomicroscopy Image Recognition. IEEE Transactions on Medical Imaging, 38(3), 791–801. doi:10.1109/tmi.2018.2872473.

[24] Liu, Y., Du, H., Niyato, D., Kang, J., Xiong, Z., Miao, C., Shen, X., & Jamalipour, A. (2024). Blockchain-Empowered Lifecycle Management for AI-Generated Content Products in Edge Networks. IEEE Wireless Communications, 31(3), 286–294. doi:10.1109/MWC.003.2300053.

[25] Liu, L., He, M., & Jeon, S. (2025). An AI-Generated Content-Based Access Control Strategy for WBANs in Healthcare Electronics Applications. IEEE Transactions on Consumer Electronics, 71(1), 1332–1342. doi:10.1109/TCE.2024.3412942.

[26] Akram, J., Aamir, M., Raut, R., Anaissi, A., Jhaveri, R. H., & Akram, A. (2025). AI-Generated Content-as-a-Service in IoMT-Based Smart Homes: Personalizing Patient Care With Human Digital Twins. IEEE Transactions on Consumer Electronics, 71(1), 1352–1362. doi:10.1109/TCE.2024.3409173.

[27] Khoramnejad, F., & Hossain, E. (2025). Generative AI for the Optimization of Next-Generation Wireless Networks: Basics, State-of-the-Art, and Open Challenges. IEEE Communications Surveys & Tutorials. doi:10.1109/COMST.2025.3535554.