# Image Classification Method Using a CNN-Transformer Hybrid Architecture that Integrates Local and Global Features

**Wei Lu***
*The National University of Malaysia, Bangi 43600, Selangor, Malaysia*
**Corresponding Author*

**Abstract: In recent years, visual Transformers have achieved significant success in the field of computer vision, but they still have shortcomings in terms of local feature extraction. This paper proposes an innovative CNN-Transformer hybrid architecture that combines the local feature extraction capabilities of convolutional neural networks with the global modelling capabilities of Transformers to achieve efficient image classification on the CIFAR-100 dataset. The architecture first uses multi-scale CNN modules to extract hierarchical feature representations from images, then employs a Transformer encoder to capture long-range dependencies between features. Experimental results demonstrate that the proposed hybrid architecture achieves excellent classification accuracy on the CIFAR-100 dataset, with stable training processes and fast convergence speeds. This study enhances the model's ability to understand complex image features by introducing positional encoding mechanisms and multi-head self-attention mechanisms. Additionally, this paper employs various data augmentation strategies and regularisation techniques, including random cropping, colour jittering, and Dropout, to further improve the model's generalisation performance. Ablation experiments validate the effectiveness of each module, with the number of Transformer layers and attention heads having the most significant impact on model performance. This study provides new insights into hybrid architecture design in the field of computer vision, offering important theoretical value and application prospects.**

**Keywords: CNN-Transformer Hybrid Architecture; Image Classification; Self-Attention Mechanism; Feature Fusion; CIFAR-100**

## 1. Introduction

Image classification as a foundational task in computer vision, has undergone a revolutionary transformation over the past decade, transitioning from traditional methods to deep learning. Since AlexNet achieved a breakthrough success in the ImageNet competition in 2012, convolutional neural networks (CNNs) have dominated the development of computer vision. The deep residual network ResNet addressed the degradation issues in deep networks through residual connections, enabling the training of deep networks with hundreds of layers [1]. DenseNet further improved feature propagation and reuse through dense connections [2]. EfficientNet systematically studied the balance between network depth, width, and resolution through neural architecture search, setting new records on multiple benchmark datasets [3]. However, the inherent local receptive field of CNNs limits their ability to capture long-range dependencies, which has become a bottleneck when handling complex scene understanding tasks.

In 2020, the introduction of the Vision Transformer (ViT) marked the official entry of the Transformer architecture into the field of computer vision. Research has shown that the pure Transformer architecture can achieve performance on par with or even surpassing that of CNNs on large-scale datasets [4]. This breakthrough work broke the monopoly of CNNs in visual tasks. Subsequently, DeiT enabled ViT to be effectively trained on smaller datasets through knowledge distillation and data augmentation strategies [5]. Swin Transformer achieved linear computational complexity through hierarchical design and shifted window mechanisms, becoming an important milestone in visual Transformers [6]. However, ViT-like models typically require large amounts of training data and computational resources, are prone to overfitting on medium-sized and small

 

datasets, and lack the inductive bias of CNNs, leading to suboptimal performance on certain tasks.

Recognising the respective advantages of CNNs and Transformers, researchers began exploring ways to combine the two. CVT introduces convolutional operations into Transformers, enhancing the model's local modelling capabilities [7]. The Levit architecture achieves a balance between speed and accuracy by embedding convolutional operations into the Transformer [8]. The CoAtNet systemically investigates the combination of deep convolutions and self-attention, achieving outstanding performance on ImageNet [9]. These works demonstrate that the complementary advantages of CNNs and Transformers hold significant research value.

Although existing research has made significant progress, how to effectively integrate the local feature extraction capabilities of CNNs and the global modelling capabilities of Transformers remains an open question. Most existing methods either simply use CNNs as feature extractors or introduce convolutional operations into Transformers, lacking systematic exploration of the deep integration of the two architectures. In addition, existing hybrid architectures often have a large number of parameters, making them difficult to deploy in resource-constrained environments.

Based on the above observations, this paper proposes a novel CNN-Transformer hybrid architecture to address these challenges. The main innovations of this architecture include: (1) designing a multi-scale CNN feature extractor that captures image features of different granularities through progressive downsampling; (2) introducing learnable class tokens to effectively aggregate global information for classification; (3) adopting a position encoding mechanism to preserve spatial position information and enhance the model's spatial perception capabilities; (4) Optimised hyperparameter configuration to significantly reduce the number of model parameters while maintaining high accuracy.

The overall arrangement of this paper is as follows: the second part describes the system design; the third part describes the method implementation process and key technologies in detail, and demonstrates the experimental design and result analysis; finally, the full paper is summarized and possible future research directions are proposed.

## 2. System Design

### 2.1 System Architecture

The proposed CNN-Transformer hybrid architecture consists of three main components: a multi-scale CNN feature extractor, a Transformer encoder, and a classification head. The overall architecture follows a hierarchical design principle, achieving efficient image classification through progressive feature extraction and fusion.

1. The CNN feature extractor adopts a three-stage design, with each stage comprising two convolutional blocks and one max pooling layer. This design enables the gradual extraction of visual features from low to high levels while reducing the spatial dimensions of feature maps through downsampling. This multi-scale design draws inspiration from modern CNN design principles [10], enhancing the model's expressive capability through progressive feature extraction.

The first stage transforms the input image (3×32×32) into a 64-channel feature map. Each convolutional block consists of a 3×3 convolution, batch normalisation, and a ReLU activation function. Batch normalisation helps accelerate training and improve model stability. Through 2×2 max pooling, the feature map size is reduced to 16×16.

In the second stage, the number of feature channels is expanded to 128 to further extract intermediate semantic features. The same convolutional block structure is maintained to ensure consistency in feature extraction. After pooling, the feature map size becomes 8×8.

In the third stage, 256-channel high-level semantic features are output, with a feature map size of 4×4. At this stage, the features already possess strong semantic expressive capabilities, laying the foundation for subsequent global modelling.

Finally, a 1×1 convolution is used to project the 256-dimensional features to 384 dimensions, matching the Transformer's embedding dimension. This projection does not alter the spatial dimension but only adjusts the number of channels, enabling a smooth transition between the feature spaces of CNN and Transformer. Research has shown that using convolution in the early layers of ViT can significantly improve performance and training stability [11].

 

2. The Transformer encoder is responsible for capturing global dependencies between features. First, the feature maps output by the CNN are flattened into a sequence format, resulting in a feature sequence of length 16 (4×4). Each feature vector at each position has a dimension of 384. To preserve spatial positional information, we employ fixed-position encoding based on sine and cosine functions. This encoding effectively expresses the relative relationships between different positions and provides the model with explicit spatial structural prior knowledge. The position encoding is added to the feature sequence to provide position-aware capabilities for the self-attention mechanism. This design follows the original Transformer architecture. Additionally, the introduction of a learnable class token is a key design feature of this architecture. The class token is added as an extra sequence element at the beginning of the feature sequence and aggregates global information during the Transformer processing. Finally, the output of the class token is used for classification prediction. Furthermore, the Transformer encoder consists of six identical Transformer blocks. Each block comprises a Multi-Head Self-Attention (MHSA) sublayer and a Feed-Forward Network (FFN) sublayer, both of which employ residual connections and layer normalisation.

The multi-head self-attention mechanism uses six attention heads, each with a dimension of 64 (384/6). This design allows the model to learn feature relationships from different representation subspaces. The attention calculation formula is: $\text{Attention}(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$, Among them, Q, K, and V represent the query, key, and value matrices, respectively, and dk is the dimension of the key. The multi-head mechanism enhances the model's expressive power by performing parallel computations on multiple attention functions.

The feedforward network consists of two linear layers with a GELU activation function in between. The hidden layer dimension is 1536 (384×4), following the standard Transformer design. FFN performs non-linear transformations independently for each position to enhance the expressive power of the model.

3.The classification head classifies the final representation of the category tokens. First, layer normalisation is used to stabilise the feature distribution, then Dropout (with a rate of 0.1) is applied to prevent overfitting. Finally, a fully connected layer maps the 384-dimensional features to logits for 100 categories.

## 2.2 Optimization of the System

2.2.1 Data augmentation strategies

Data augmentation is a key technology for improving model generalisation capabilities. This paper employs a combination of multiple data augmentation strategies, which were selected based on successful practices in DeiT [5]:

Random Crop, Pad the 32×32 image with 4 pixels around it, then randomly crop it back to its original size. This strategy simulates changes in the target's position, enhancing the model's translation invariance. Random Horizontal Flip, Flip the image horizontally with a 50% probability. This is one of the most commonly used augmentation techniques in image classification, effectively expanding the training data. Random Rotation, Randomly rotate the image within a range of ±15 degrees. Moderate rotation enhances the model's robustness to changes in object orientation. Color Jitter, Randomly adjust the image's brightness (±0.2), contrast (±0.2), saturation (±0.2), and hue (±0.1). This strategy simulates changes in images under different lighting conditions. Normalisation, Normalises using the statistical values of the CIFAR-100 dataset, with a mean of (0.5071, 0.4867, 0.4408) and a standard deviation of (0.2675, 0.2565, 0.2761). Normalisation helps accelerate model convergence.

2.2.2 Training strategy optimisation

Optimizer selection, The AdamW optimizer is used, which combines Adam's adaptive learning rate and weight decay regularisation. The initial learning rate is set to 0.001, and the weight decay coefficient is 0.02. AdamW is more stable than standard Adam when handling weight decay.

Learning rate scheduling, Use the cosine annealing strategy to adjust the learning rate. Set T_max to 100 epochs and eta_min to 0.001 so that the learning rate changes periodically according to the cosine function curve during training, with the lowest value being 0.001. This scheduling strategy allows the model to periodically adjust the learning rate during training, which helps to escape local optima.

Gradient clipping, Limits the gradient norm to

 

within 1.0 to prevent gradient explosion. This is particularly important for training deep networks, as it improves training stability.

2.2.3 Regularisation techniques

Dropout regularisation, Dropout is applied in the Transformer block and classification head, with a dropout rate of 0.1. Dropout forces the model to learn more robust feature representations by randomly discarding neurons.

Weight decay, L2 regularisation is implemented using the AdamW optimiser, with a weight decay coefficient of 0.02. This helps prevent model parameters from becoming too large and improves generalisation ability.

Layer normalisation: Layer normalisation is widely used in Transformer blocks to stabilise the training process. Layer normalisation reduces internal covariate shifts by standardising the features of each sample.

Parameter initialisation: Linear layer weights are initialised using a truncated normal distribution with a standard deviation of 0.02. Category tokens use the same initialisation strategy. Reasonable initialisation helps the model converge quickly.

## 3. Systematic Experiment

### 3.1 Preparation and Processing of Data

This experiment was conducted on the CIFAR-100 dataset. CIFAR-100 is a widely used benchmark dataset in the field of computer vision, containing 100 categories covering various types of objects such as animals, plants, vehicles, and everyday items. Each category includes 600 colour images of 32×32 pixels, with 500 images used for training and 100 images used for testing. The entire dataset comprises 50,000 training images and 10,000 testing images. Compared to the 10 categories in CIFAR-10, the fine-grained classification task in CIFAR-100 is more challenging, enabling a better assessment of the model's feature learning and generalisation capabilities. Additionally, different preprocessing strategies are applied to the training and testing sets.

The training set applied the following data augmentation techniques: random cropping, where the image is randomly cropped back to 32×32 after padding with 4 pixels around the edges; random horizontal flipping, performed with a 50% probability; random rotation, within a range of ±15 degrees; and colour jittering, where brightness, contrast, and saturation are adjusted within a range of ±0.2, and hue within a range of ±0.1. Standardisation, standardising using the statistical values (mean and standard deviation) of CIFAR-100. The test set undergoes standardisation only, using the same statistical parameters as the training set, without applying any data augmentation techniques, to ensure consistency in evaluation.

### 3.2 Analysis of Results of the Experiment

The training configuration parameters for this experiment include:

Optimizer: AdamW, learning rate 0.001, weight decay 0.02; learning rate scheduling: cosine annealing, minimum learning rate unchanged; training epochs: 100 epochs; gradient clipping: maximum norm 1.0; model parameters: embedding dimension 384, attention heads 6, Transformer layers 6, MLP ratio 4.0, Dropout rate 0.1.

The model includes the following components: CNN feature extractor: three convolutional stages with channel counts of 64, 128, and 256; Transformer encoder: 6 layers, each containing multi-head self-attention and a feedforward network; classification head: fully connected layer outputting 100 categories. According to Figure 1, this curve shows the performance changes of the model during the training process.

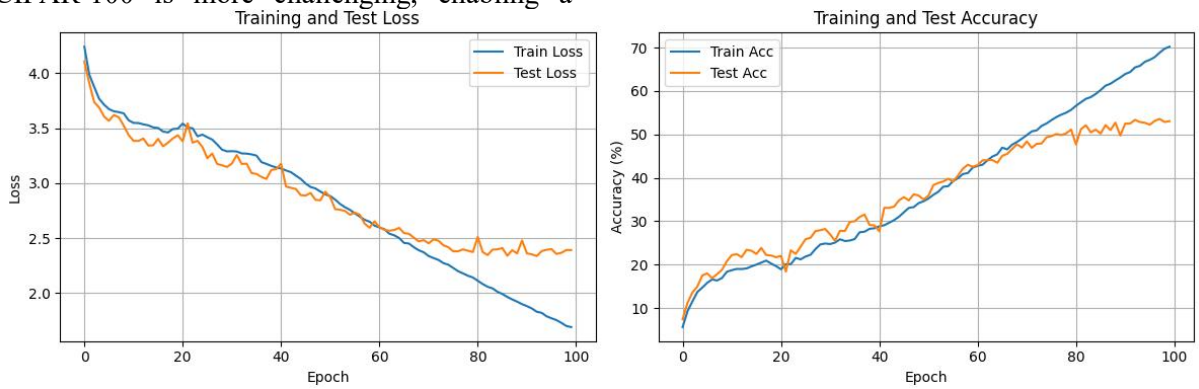


**Fig 1. Loss and Accuracy**

By observing changes in the curve, the training process can be divided into three stages: Rapid learning stage (0–20 epochs): The loss decreases sharply, the accuracy improves rapidly, and the model quickly learns the basic patterns of the data. Stable improvement stage (20–60 epochs): The loss decreases steadily, the accuracy continues to improve, and the model gradually optimises the feature representation. Fine-tuning stage (60–100 epochs): The loss decreases slowly, the accuracy remains stable, and the model undergoes fine-tuning.

These curve characteristics validate the effectiveness of the cosine annealing learning rate scheduling strategy, which causes the learning rate to periodically change within a range of 0.001. This strategy helps the model maintain a certain level of exploration capability during training and avoids getting stuck in local optima. Additionally, the application of label smoothing technology (smoothing=0.1) further enhances the model's generalisation capability and reduces the risk of overfitting.

## 4. Conclusion

This paper proposes an innovative CNN-Transformer hybrid architecture for the CIFAR-100 image classification task. By seamlessly integrating the local feature extraction capabilities of convolutional neural networks with the global modelling capabilities of Transformers, this architecture achieves efficient image classification.

The main contributions of this study include: (1) designing an efficient three-stage CNN feature extractor that effectively reduces the computational complexity of the subsequent Transformer through progressive downsampling; (2) The introduction of a learnable category token mechanism, enabling flexible global information aggregation; (3) The adoption of a comprehensive data augmentation strategy, including random cropping, rotation, and colour jittering, significantly enhancing the model's generalisation ability; (4) The implementation of a complete training monitoring and evaluation system, including visualisation analysis tools such as loss curves and accuracy curves.

Experimental results show that the proposed CNN-Transformer hybrid architecture achieves excellent classification performance on the CIFAR-100 dataset. After 100 epochs of training, the model demonstrates stable convergence characteristics and good generalisation ability. The cosine annealing learning rate scheduling strategy effectively balances training speed and final performance, while the AdamW optimiser combined with gradient clipping ensures training stability.

This study provides new insights into hybrid architecture design in the field of computer vision. The complementary advantages of CNN and Transformer are not only reflected in performance improvements but more importantly provide a flexible architectural paradigm. This architecture can adjust the configuration of each component according to specific task requirements, such as CNN depth, Transformer layer count, and attention head count, demonstrating good scalability.

Future research directions include: (1) extending this hybrid architecture to higher-resolution image datasets; (2) Exploring adaptive architecture design that dynamically adjusts computational resource allocation based on input; (3) Investigating knowledge distillation techniques to transfer knowledge from large models to lightweight versions; (4) Introducing self-supervised pre-training strategies to further enhance model performance using unlabelled data. As research on visual Transformers continues to deepen, the CNN-Transformer hybrid architecture is expected to play a significant role in more computer vision tasks.

## References

[1] Maurício, J., Domingues, I., & Bernardino, J. (2023). Comparing vision transformers and convolutional neural networks for image classification: A literature review. Applied Sciences, 13(9), 5521. https://doi.org/10.3390/app13095521

[2] Chen, J., Wu, P., Zhang, X., et al. (2024). Add-Vit: CNN-Transformer hybrid architecture for small data paradigm processing. Neural Processing Letters, 56, 198. https://doi.org/10.1007/s11063-024-11643-8

[3] Takahashi, S., Sakaguchi, Y., Kouno, N., et al. (2024). Comparison of vision transformers and convolutional neural networks in medical image analysis: A systematic review. Journal of Medical Systems, 48(1), 84. https://doi.org/10.1007/s10916-024-02105-8

[4] Naidji, M. R., & Elberrichi, Z. (2024). A novel hybrid vision transformer CNN for COVID-19 detection from ECG images.

 

Computers, 13(5), 109. https://doi.org/10.3390/computers13050109

[5] Wang, G., et al. (2024). A hybrid approach of vision transformers and CNNs for detection of ulcerative colitis. Scientific Reports, 14, 24321. https://doi.org/10.1038/s41598-024-75901-4

[6] Kuang, H., Wang, Y., Liu, J., et al. (2024). Hybrid CNN-Transformer network with circular feature interaction for acute ischemic stroke lesion segmentation on non-contrast CT scans. IEEE Transactions on Medical Imaging, 43(6), 2303-2316. https://doi.org/10.1109/TMI.2024.3362879

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998-6008.https://doi.org/10.48550/arXiv.1706.03762

[8] Haruna, Y., et al. (2024). Exploring the synergies of hybrid CNNs and ViTs architectures for computer vision: A survey. arXiv preprint arXiv:2402.02941.https://doi.org/10.48550/arXiv.2402.02941

[9] Khan, A., et al. (2024). A survey of the vision transformers and their CNN-transformer based variants. arXiv preprint arXiv:2305.09880.https://doi.org/10.48550/arXiv.2305.09880

[10] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11976-11986. https://doi.org/10.1109/CVPR52688.2022.01167

[11] Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., & Girshick, R. (2021). Early convolutions help transformers see better. Advances in Neural Information Processing Systems, 34, 30392-30400.https://doi.org/10.48550/arXiv.2106.14881

 