Proceedings of CONF-CDS 2025 Symposium: Data Visualization Methods for Evaluation
DOI: 10.54254/2755-2721/2025.PO24818

Research on Image Classification Models Based on Deep Learning

Hao Zhang

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China
spaceexecutionagd@outlook.com

Abstract. With the development of deep learning technologies, image recognition tasks have become increasingly important in the field of computer vision. Traditional image classification methods based on handcrafted features have been gradually replaced by Convolutional Neural Networks (CNNs), which have recently been further advanced by Transformer-based visual models such as Vision Transformer (ViT). This paper reviews the evolution of image classification techniques, systematically introduces the core structures and underlying principles of CNNs and ViT, and focuses on the mechanism by which ViT linearly embeds image patches into the Transformer architecture for feature modeling. Experiments conducted on the ImageNet 1000 (mini) dataset demonstrate that the ViT model outperforms traditional CNN models such as ResNet101 and ResNet152 in Top-1 accuracy, Top-5 accuracy, and average accuracy, thereby validating its superior performance and promising application prospects in image recognition tasks.

Keywords: deep learning, Convolutional Neural Network (CNN), Vision Transformer (ViT), image classification

1. Introduction

With the advancement of information technology, image data have become increasingly important for scientific research, production, and daily life. Manual image classification is no longer sufficient due to inefficiency, making rapid image classification via computers an urgent need. Image classification has become a vital task in computer vision, playing a key role across various fields and practical scenarios. With the rapid development of artificial intelligence and deep learning, image classification methods based on deep learning have gained widespread attention for their efficiency. Using deep learning for image classification has become a major trend in the computer vision community, drawing considerable academic interest.

Initially, researchers employed traditional approaches to conduct image classification tasks, typically involving two steps: feature extraction and feature classification. Scholars studied each step individually. For example, David G. Lowe proposed the Scale-Invariant Feature Transform (SIFT) method, which uses local invariant features for robust global matching. This method extracted highly distinctive features and addressed the issue of matching handcrafted features under complex transformations [1]. Gongde Guo and colleagues proposed a novel K-Nearest Neighbors (KNN)

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.PO24818

model that dynamically determines the optimal local value of k, replacing manual selection and enhancing classification efficiency [2].

With the introduction of Convolutional Neural Networks (CNNs) in the 1980s, researchers began applying them to image classification. LeCun et al. introduced the LeNet-5 network, which used gradient-based backpropagation for supervised CNN training. In handwritten digit recognition, it achieved a low error rate of 0.8%. However, the model suffered from high training costs, small dataset sizes, and weak generalization [3]. Krizhevsky et al. proposed the AlexNet model, which replaced LeNet-5's Sigmoid activation with ReLU, solving issues of computational cost and vanishing gradients in deeper networks while alleviating overfitting [4]. Building on AlexNet, Simon Yan et al. introduced the VGG network, which adopted a multi-scale training strategy and employed deeper networks with smaller kernels. This reduced the required iterations for convergence and demonstrated a proportional relationship between network depth and performance, though excessively deep networks led to degradation issues [5]. He K. et al. proposed ResNet V1, which uses shortcut connections to directly map low-level features to deeper layers, simplifying the learning process and resolving degradation issues, thereby significantly improving accuracy [6]. He K. later proposed ResNet V2, which replaces nonlinear activation in shortcuts with identity mapping and uses batch normalization (BN) to enhance network performance [7]. S. Targ et al. introduced the ResNet-in-ResNet (RiR) dual-stream deep architecture, which utilizes residual and transient streams with convolutional kernels for inter-stream information transfer. This architecture enables dynamic selection of features to retain or discard, addressing information redundancy and limited expressiveness in traditional ResNets, while maintaining computational efficiency and significantly improving classification performance [8]. Tan M. and colleagues designed EfficientNet using Neural Architecture Search (NAS), balancing depth, width, and resolution via a compound scaling method. Combining NAS and compound scaling significantly improved classification accuracy, achieving 84.4% Top-1 and 97.1% Top-5 accuracy on ImageNet [9].

Transformer, introduced by Google in 2017, is an attention-based model that forgoes convolutional structures and employs multi-head attention and positional encoding, achieving remarkable performance in natural language processing tasks such as translation [10]. Radford A. and others improved Transformer and developed GPT-2, trained on WebText, which learned tasks like question answering, translation, and reading comprehension without explicit supervision, showcasing its potential in unsupervised learning [11]. Some researchers recognized the advantages of Transformer over convolution and began applying attention mechanisms to image classification. Chen M. et al. proposed the iGPT model, which employs autoregressive and BERT objectives for pretraining. The model's performance is evaluated using linear probes or fine-tuning, demonstrating results comparable to supervised learning models [12]. Dosovitskiy A. et al. introduced the Vision Transformer (ViT), which transforms images into patches, linearly embeds them as sequences, and feeds them into a Transformer. By treating image patches similarly to NLP tokens, ViT achieves state-of-the-art performance on several image classification datasets at a lower pretraining cost, surpassing CNN models such as EfficientNet [13].

2. Methodology

2.1. Introduction to CNN

A Convolutional Neural Network (CNN) is a typical feedforward neural network with a deep structure that incorporates convolution operations. It generally consists of multiple convolutional layers, pooling layers, fully connected layers, and activation functions. CNN training includes two

DOI: 10.54254/2755-2721/2025.PO24818

main processes: forward propagation and backpropagation. In forward propagation, the raw image data is passed through the network layers to generate the output. In backpropagation, the network adjusts the parameters of the convolutional layers based on the computed loss value.

The convolutional layer extracts spatial features from the input image via convolution operations to generate feature maps. The general definition of convolution is shown in Equation (1), and its discrete form is shown in Equation (2).

$$(f*g)(n) = \int_{-\infty}^{\infty} f(\gamma)g(n-\gamma)d\gamma \tag{1}$$

$$\left(f*g\right)\left[x\right] = \sum_{y=-\infty}^{\infty} f(y)g(x-y) \tag{2}$$

In a convolutional network, the 2D convolution operation for an image is defined as in Equation (3):

$$Output\left(i,j\right) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} Input(i\cdot s+m, j\cdot s+n)\cdot W(m,n) + b \tag{3}$$

Where, Output(i,j) represents the value at position (i, j) in the output matrix, Input(x,y) represents the corresponding value in the input image, s is the stride of the convolutional kernel, M and N are the width and height of the kernel, and,b is the bias term. A schematic diagram of the 2D convolution process is shown in Figure 1.

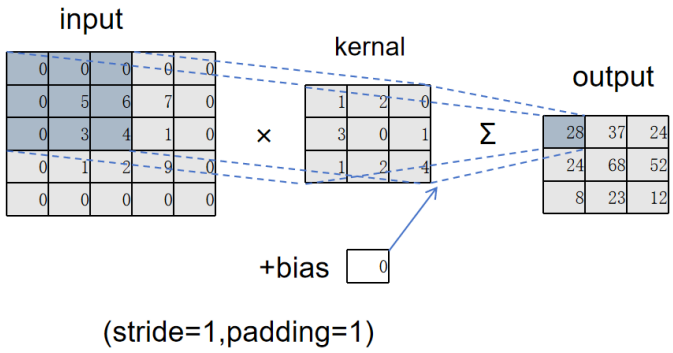

Figure 1: Schematic diagram of 2D convolution operation on an image

Given a specific kernel size, stride, and input image size, the output dimensions $w_0 \times h_0$ are calculated by Equation (4):

$$w_0 = \frac{(w+2P-M)}{s} + 1$$

$$h_0 = \frac{(h+2P-N)}{s} + 1 \tag{4}$$

Here, w×h denotes the input dimensions, and P is the padding size. Padding is typically used to preserve boundary features of the input image, usually by padding with zeros.

The pooling layer primarily extracts the most significant features to reduce feature dimensionality and the size of the feature maps, thereby lowering computational complexity and improving

DOI: 10.54254/2755-2721/2025.PO24818

efficiency. There are two common methods for selecting representative values: average pooling, which takes the average value, and max pooling, which takes the maximum value.

The fully connected layer performs discriminative analysis on comprehensive high-level features by integrating and aggregating low-level features. Each node in a fully connected layer is connected to all nodes in the preceding layer, and each connection has a weight. By recombining features through these weighted connections, new feature representations are formed. Typically, multiple fully connected layers are stacked, with the final output passed through an activation function—such as the Softmax or Sigmoid function—to produce a probability distribution over the target classes.

2.2. ViT(Vision Transformer)

The Vision Transformer (ViT) is an architecture that applies the Transformer model—originally designed for natural language processing—to the field of computer vision. The structure of the Vision Transformer is shown in Figure 2.

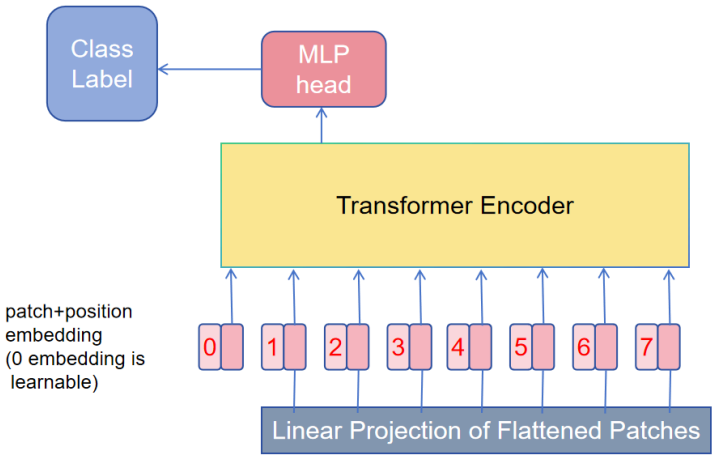

Figure 2: Structure of the Vision Transformer (ViT)

2.2.1. Self-attention mechanism and multi-head self-attention

The attention mechanism simulates human perception and attention behavior, allowing the model to focus on different parts of the input. The attention function, known as Scaled Dot-Product Attention, is a function of three input vectors: Query (Q), Key (K), and Value (V). Its output is a weighted representation of the input.

To compute the correlation between different input vectors, the model multiplies the embedded input matrices to obtain the Q, K, and V vectors. Then, it computes the dot product of Q and K to obtain similarity scores, applies a softmax function to normalize these scores, and finally multiplies the result with the V vectors. The weighted sum of these vectors forms the output. The computation is defined in Equation (5):

$$Attention\left(Q,K,V\right)=SoftMax\left(\frac{QK^{T}}{\sqrt{d_{k}}}\right)V \tag{5}$$

DOI: 10.54254/2755-2721/2025.PO24818

Where, Q is the query vector, K is the key vector, d_k is the dimensionality of the input vectors, V is the value vector.

The multi-head self-attention mechanism is essentially an extension of the self-attention mechanism, using multiple sets of projection matrices: W_Q, W_K, and W_V for queries, keys, and values, respectively. Each set of weights is referred to as an attention head. Assume the multi-head attention mechanism uses n heads; the weight matrices for the i -th head are denoted as W_i^Q , W_i^K , W_i^V. The attention computation for each head is shown in Equation (6):

$$h_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{6}$$

Each attention head performs its own self-attention computation as in Equation (7):

$$Attention\left(Q_i, K_i, V_i\right) = SoftMax\left(\frac{Q_i {K_i}^T}{\sqrt{d_k}}\right)V_i \tag{7}$$

The results from all attention heads are concatenated and passed through a final linear projection as in Equation (8):

$$Multihead\left(Q, K, V\right) = Concat\left(h_1, h_2, \ldots, h_n\right)W^O \tag{8}$$

2.2.2. Vision transformer architecture

The standard Transformer takes a one-dimensional sequence of word embeddings as input [14], while image data consists of three-channel pixel matrices. Therefore, images must first be transformed into a one-dimensional form suitable for input into the Transformer.

The specific method involves dividing the image into patches, then linearly projecting each patch into a one-dimensional vector suitable for Transformer input [15]. The transformation is shown in Equation (1.9):

$$\mathrm{x} \in \mathrm{R}^{\mathrm{H}\times\mathrm{W}\times\mathrm{C}} \;\longrightarrow\; \mathrm{x_p} \in \mathrm{R}^{\mathrm{N}\times(\mathrm{P}^2\bullet\mathrm{C})} \tag{9}$$

where (H,W) is the resolution of the original image, C is the number of channels, (P,P) is the resolution of each patch, and $\mathrm{N} = \frac{\mathrm{HW}}{\mathrm{P}^2}$ is the number of patches.

After flattening the image patches, they are projected via a learnable linear mapping into D-dimensional vectors, becoming patch embeddings. Learnable positional embeddings are then added to these patch embeddings to retain spatial information, as shown in Equation (10):

$$z_0 = \left[x_{class}; x_p^1 E; x_p^2 E; \ldots; x_p^N E\right] + E_{pos}, E \in R^{(P^2\cdot C)\times D}, E_{pos} \in R^{(N+1)\times D} \tag{10}$$

Analogous to the [CLS] token in BERT, a learnable class embedding can be prepended, whose output from the Transformer encoder is used as the image-level representation (Equation 11):

$$z_0^0 = x_{class} \tag{11}$$

The encoder consists of alternating layers of multi-head self-attention (MSA) and multi-layer perceptrons (MLPs). Each block is preceded by Layer Normalization (LN) and followed by residual connections.

DOI: 10.54254/2755-2721/2025.PO24818

3. Experiments

3.1. Dataset description

ImageNet 1000 (mini) is a streamlined subset of the full ImageNet 1000 (ILSVRC 2012) dataset. The training and validation sets each contain 1,000 classes. The number of images per class is relatively small—typically a few dozen—and all images are in color, unprocessed, and vary in resolution. Each image is stored in a folder named after its corresponding class label, and the filenames are composed of the class label followed by an image index. This dataset is well-suited for experimentation and training on resource-constrained devices and supports various image classification, object detection, and image recognition tasks.

3.2. Experimental setup

The experiments were conducted on a Linux operating system with an NVIDIA RTX 3090 GPU and a 14-core CPU.

The training and validation sets used were from ImageNet 1000 (mini), with 1,000 classes each. The class labels are embedded in both the filenames and the folder structure. The main model used in the experiment was the Vision Transformer Base, with an input resolution of 224×224 pixels and an initial training epoch set to 0.

A two-phase training strategy was adopted: Epochs 0–100: Only the classification head was trained while the backbone network remained frozen. Epochs 100–200: The entire network was unfrozen for full fine-tuning. The batch size was set to 16 throughout the training process, and model weights were saved every 10 epochs. The optimizer used was Momentum SGD, with a momentum coefficient of 0.9. The initial learning rate was set to 0.01, the minimum learning rate to 0.0001, and a cosine annealing schedule was applied to accelerate convergence and improve final accuracy. The hyperparameters used in the experiments are shown in Figure 3.

keys	values
num_classes	1000
backbone	vit_b_16
model_path	
input_shape	[224, 224]
Init_Epoch	0
Freeze_Epoch	100
UnFreeze_Epoch	200
Freeze_batch_size	16
Unfreeze_batch_size	16
Freeze_Train	True
Init_lr	0.01
Min_lr	0.0001
optimizer_type	sgd
momentum	0.9
lr_decay_type	cos
save_period	10
save_dir	logs
num_workers	4
num_train	1000
num_val	1000

Figure 3: Hyperparameters used in the experiment

DOI: 10.54254/2755-2721/2025.PO24818

The performance of the ViT model was compared with that of ResNet101 and ResNet152. The Top-1 accuracy, Top-5 accuracy, and average accuracy were computed for each model on the dataset.

3.3. Experimental results and analysis

As described above, we compared three models. The Top-1, Top-5, and average accuracies obtained are shown in Table 1. It is evident that the ViT model outperforms both ResNet101 and ResNet152 across all three metrics.

Table 1: Training results of the three models

Network Architecture	Top-1 Accuracy	Top-5 Accuracy	Average Accuracy
Resnet101	91.38%	94.12%	90.08%
Resnet152	92.71%	95.08%	90.86%
ViT	92.93%	95.71%	91.16%

To further evaluate performance, we compared the per-class accuracy of the ViT and ResNet152 models for selected categories from the dataset. As shown in Table 2, although ResNet152 outperformed ViT in a few categories, ViT achieved higher accuracy in the majority of them. This further demonstrates the superior performance of ViT in classification tasks.

Table 2: Per-class accuracy comparison between ResNet152 and ViT

Class ID	Class Label	ResNet152 Accuracy (%)	ViT Accuracy (%)
n01440764	Carp	93	92
n01773797	Spider	91	93
n02412080	Goat	83	85
n01629819	Gecko	77	77
n02999410	Chain	92	90
n13133613	Corn	96	99
n02090379	Labrador Retriever	97	100
n01631663	Salamander	87	92
n03721384	Guqin	90	92
n10148035	Wedding	83	87

4. Conclusion

This paper systematically reviewed deep learning-based image classification models, tracing the evolution from traditional Convolutional Neural Networks (CNNs) to the attention mechanism-driven Vision Transformer (ViT). A series of comparative experiments were conducted using the ImageNet 1000 (mini) dataset to evaluate ViT against two classical CNN architectures: ResNet101 and ResNet152. ViT achieved a Top-1 accuracy of 92.93%, Top-5 accuracy of 95.71%, and average accuracy of 91.16%, outperforming ResNet101 (91.38%, 94.12%, 90.08%) and ResNet152 (92.71%, 95.08%, 90.86%) across all metrics. These results validate the advantage of ViT's architecture, which divides an image into patches and linearly embeds them into the Transformer framework. Leveraging

DOI: 10.54254/2755-2721/2025.PO24818

the global self-attention mechanism, ViT effectively captures global contextual information in images, thereby surpassing traditional convolutional networks in classification tasks.

Future research in deep learning-based image classification can advance in the following directions: 1. Incorporating large-scale pretraining strategies to enhance the model's generalization capability across diverse image types. 2. Exploring hybrid CNN–Transformer architectures to achieve joint modeling of local and global features. 3. Extending ViT to higher-resolution or more complex vision tasks such as object detection and semantic segmentation to further unlock its application potential in broader domains.

References

[1] Lowe D G. Distinctive image features from scale-invariant keypoints [J]. International journal of computer vision, 2004, 60: 91-110.

[2] Guo G, Wang H, Bell D, et al. KNN model-based approach in classification [C]//On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings. Springer Berlin Heidelberg, 2003: 986-996.

[3] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[4] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [J]. Advances in neural information processing systems, 2012, 25.

[5] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv: 1409.1556, 2014.

[6] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[7] He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks [C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. Springer International Publishing, 2016: 630-645.

[8] Targ S, Almeida D, Lyman K. Resnet in resnet: Generalizing residual architectures [J]. arXiv preprint arXiv: 1603.08029, 2016.

[9] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks [C]//International conference on machine learning. PMLR, 2019: 6105-6114.

[10] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [J]. Advances in neural information processing systems, 2017, 30.

[11] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners [J]. OpenAI blog, 2019, 1(8): 9.

[12] Chen M, Radford A, Child R, et al. Generative pretraining from pixels [C]//International conference on machine learning. PMLR, 2020: 1691-1703.

[13] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale [J]. arXiv preprint arXiv: 2010.11929, 2020.

[14] Han K, Xiao A, Wu E, et al. Transformer in transformer [J]. Advances in neural information processing systems, 2021, 34: 15908-15919.

[15] Han K, Wang Y, Chen H, et al. A survey on vision transformer [J]. IEEE transactions on pattern analysis and machine intelligence, 2022, 45(1): 87-110.