Images Classification Integrating Transformer with Convolutional Neural Network

Yulin Peng

School of Sciences, Chang'an University, Xi'an 710064, China.

2020901049@chd.edu.cn

Abstract. Convolutional neural networks (CNN) are one of the most widely used deep learning methods in computer vision, which can effectively extract local spatial information from images, but lack global understanding and dependency modelling of image features. As a result, contextual information cannot be fully utilized by the network. For example, on coordinate modelling tasks (such as object detection, image generation, etc.), CNN may not be able to accurately locate or reconstruct the position and shape of objects. In contrast to traditional CNN models such as ResNet, Transformers rely on their global attention mechanism to capture long-distance dependencies between patches. The thesis presents an enhanced lightweight method which integrates Transformer with five convolutional neural layers. Model based on CNN and Transformer is tested on the two benchmark datasets MNIST and CIFAR-10. After a few epochs, the model is convergent and reaches high accuracy of 99.34% in MNIST and 92.04% in CIFAR-10. This model outperforms the single CNN and some state-of-the-art models in classifying both datasets, especially in distinguishing similar images like '6' and '9', 'bird' and 'plane'. These results indicate the model's good robustness and generality.

Keywords: Classification; Transformer; Convolutional Neural Network.

1. Introduction

The history of CNN can be traced back to the late 1970s and early 1980s, when researchers explored the idea of using neural networks with local connectivity and shared weights to model the visual cortex of animals. Computer scientists such as Fukushima[1], LeCun[15], and Rummelhart[16] developed artificial neural networks that mimicked the biological model, using backpropagation to train the network parameters.

CNN has the capability to acquire hierarchically organized features from data, beginning with basic patterns like edges and corners, and progressing to more intricate ones such as faces and objects. It can be utilized for a wide range of objectives including image classification, segmentation, detection, recognition, generation, and enhancement. Some examples of CNN architectures are LeNet-5[1], AlexNet[2], VGGNet[3], ResNet[4] etc.

Recognizing handwritten digits and object recognition are crucial tasks in various linguistic and real-world scenarios that involve identifying hand-written numbers, such as processing bank checks, forms, and invoices. However, the recognition of handwritten digits poses a challenge due to the significant variability in writing styles and formats. Unlike words, numbers do not benefit from contextual or semantic correction. Traditional approaches such as CNN architectures like Lenet-5 are mostly used in classifying numerals.

However, CNN has some disadvantages that limit its performance and generalization ability. Some of these disadvantages are:

Classification of images with different positions: CNN can struggle to recognize objects that are rotated, scaled, or translated in the input image, especially if they are not present in the training data. This is because CNN is not fully translation-invariant or rotation-invariant[5], due to the use of fixed filters.

Lacks global understanding: CNN are vulnerable to adversarial examples, which can trick the model into making wrong predictions or classifications. These inputs are created by some subtle changes on the original image that are hard to notice by humans but can significantly alter the model's output. Adversarial examples expose the limitations of CNN in capturing the global understanding

and dependencies of image features, which leads to misclassifications of objects and numerals. Adversarial examples are a serious challenge for the security and reliability of CNN, especially in critical applications such as face recognition or self-driving cars. Various defense methods have been developed to protect CNN from adversarial examples, such as adversarial training[6], [7], defensive distillation[8], or gradient masking[9]. However, these methods are not perfect and can also compromise the model's performance on normal inputs.

Other minor disadvantages: CNN also has some other minor disadvantages, such as requiring a large amount of labeled data for training, being prone to overfitting if not regularized properly, being sensitive to hyperparameters[10] such as learning rate or filter size and being computationally expensive and memory-intensive for large-scale applications. These disadvantages can be mitigated by using various techniques such as transfer learning, dropout, batch normalization, or parallel computing. However, these techniques also introduce some trade-offs and challenges for optimizing and deploying CNN.

Meanwhile, Transformer, presented by google in 2017, relies on their global attention mechanism to capture long-distance dependencies between patches, can complementary strengths with CNN.

Transformer uses a mechanism called attention[11], which allows the network to learn the relevance and dependency of different elements in the input sequence. Transformer can also encode and decode the input sequence in parallel, rather than sequentially, which makes it more efficient and scalable. Transformer can handle various tasks, such as machine translation, text summarization, text generation, and natural language understanding. Transformer has also been extended to computer vision tasks such as image classification and image retrieval. This approach was first proposed by Dosovitskiy et al.[12], who introduced the Vision Transformer (ViT) model and showed that it can achieve state-of-the-art results on large-scale image classification datasets, such as ImageNet-1k, when trained with sufficient data.

Transformer can address some of the disadvantages that convolutional neural networks (CNN) have, such as:

Classification of images with different positions: Transformer can learn to recognize objects regardless of their position, orientation, or scale in the input image, by using a self-attention mechanism that captures the global context of the image. Transformer can also use positional encodings to represent the spatial information of the image pixels[13]. Transformer can outperform CNN on tasks such as object detection, segmentation, and recognition.

Adversarial examples: Transformer can be more robust to adversarial examples than CNN, by using a pre-training and fine-tuning strategy[14] that leverages large-scale unlabeled data. Transformer can also use a contrastive learning objective[15] that encourages the network to learn invariant and discriminative features from the data. Transformer can also use a generative adversarial network (GAN) framework to generate realistic and diverse images that can fool CNN.

Coordinate frame: Transformer can process data that have different coordinate frames or representations, such as graphs, point clouds, or 3D shapes, by using a graph attention mechanism that learns the relations between the nodes or points in the data[16]. Transformer can also use a geometric learning approach that incorporates geometric priors and constraints into the network.

Inspired by the above advantages and disadvantages, this thesis combines Transformer with CNN, and applies Transformer to image recognition. That is, using CNN to extract and pool image features, and then sending data into Transformer and using attention mechanism to learn features. This method can complement each other's advantages. The above advantages and disadvantages also demonstrate the scientific and feasibility of this method.

This thesis's major objective is to novel an image classification method that can be used in recognizing objects and numerals in a high accuracy, identifying the rotated numerals, and sheltered, low luminosity pictures.

The major contributions of this thesis can be summarized in the following points:

This thesis proposes an improved images classification method focusing on different situations.

Transformer is integrated with CNN for images classification.

Enhanced recognition accuracy demonstrates the superiority of the proposed system over existing methods.

2. Related work

Similar works focus on improving the accuracy and reducing the learning epochs of images recognition have got impressive performance.

2.1 Transformer

Vision Transformer (ViT) suffers from poor performance on small datasets, due to the lack of inductive bias and data efficiency.

To address this issue, several methods have been proposed to enhance ViT on these tasks. For example, Chen et al.[17] proposed CrossViT. It combines patches of different sizes in a dual-branch transformer architecture and uses a cross-attention module to exchange information between branches, which reduces computational and memory costs and improves the feature quality. El-Nouby et al.[18] proposed to train vision transformers for image retrieval, using a metric learning objective that consists of a contrastive loss and a differential entropy regularizer, which encourages the model to learn discriminative and diverse features. This method outperforms CNN-based approaches on several category-level retrieval datasets and is competitive for object retrieval on challenging datasets.

2.2 Convolutional Neural Network

CNN achieved a breakthrough in 2012 with AlexNet [2], which used a deep architecture with eight layers and several techniques to improve training and generalization. The introduction of techniques like dropout, data augmentation, and rectified linear units (ReLU) further contributed to improving CNN's training and generalization capabilities. Since then, many variants and improvements of CNN have been proposed, such as VGG[3], ResNet [4], DenseNet[19], and EfficientNet[20]. These models aim to increase the depth, width, or efficiency of the network, while simultaneously maintaining or reducing the number of parameters and computational cost.

3. Mixed Model

The mixed model of this paper is illustrated in Figure 1. As the Figure 1 shows,

the model consists of three main components: Convolutional layer, Transformer layer, and SAM optimizer.

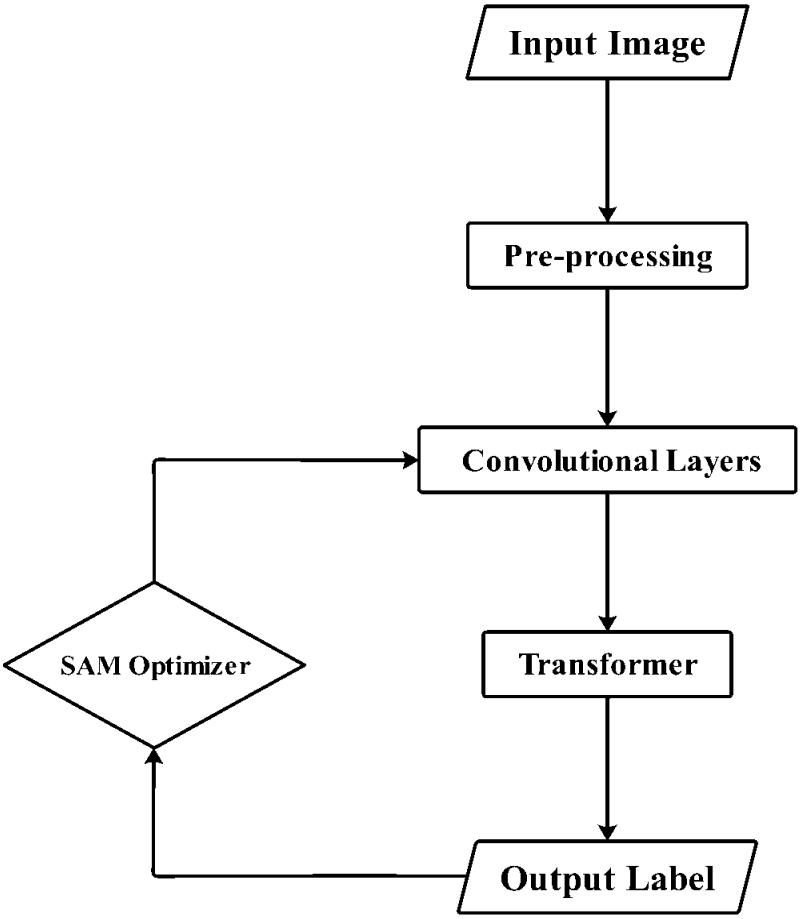

Figure 1. The main parts of the nixed model

3.1 Convolutional Layer

The convolutional layer is responsible for extracting high-level features from the input images. It is composed of five residual blocks, each of which has two convolutional layers with batch normalization and Mish[21] activation function, followed by a shortcut connection that adds the input to the output of the block. The residual blocks help to avoid the problem of vanishing gradients and improve the representation power of the network. After each residual block, we apply a max-pooling layer with a kernel size of 2 and a stride of 2 to reduce the spatial dimension of the feature maps. And finally, each image will be divided into 512 parts and turned into the encoders of Transformer.

Let the input image be:

$$x \in R^{b\times 3\times 32\times 32}$$

where b is the batch size of the input. First, the input image passes through the residual convolutional layer and the maximum pooling layer to obtain the image features:

$$f \in R^{b\times d\times 1\times 1}$$

where,

$$f_{i,j,k,l} = \max\left(0, w_j * x_j + b_j + s_j * x_i\right)_{k,l}$$

d is model dimension, w_j and b_j is the weight and bias of the jth convolutional layer, s_j is the weight of the jth residual connection, * represents convolution operation, $\max(0,\cdot)$ represents activation function.

The Mish function is a new type of activation function proposed by Sergey Ioffe in 2019 [21]. It is an improvement on Leaky ReLU, by introducing the softplus function and the tanh function, the activation function is made more smooth and non-monotonous. It is superior to unsmooth activation functions in generalization like ReLU. The formula for its function is:

$$\text{Mish}(x) = x * \tanh(\text{softplus}(x))$$

where,

$$\text{softplus}(x) = \ln^{1+e^x}$$

The function formula of Leaky ReLU is:

$$\text{LeakyReLu} = \max(0.1 * x, x)$$

The function image of Mish and Leaky ReLU is (Figure 2):

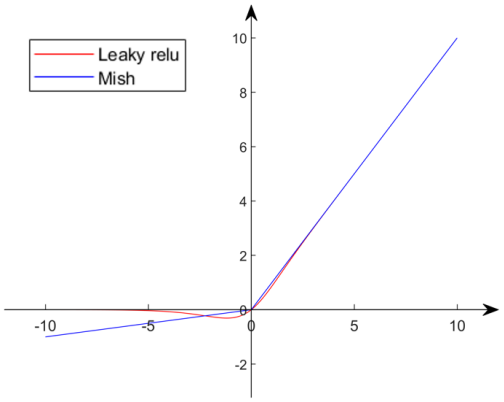

Figure 2. Graph of Mish and Leaky ReLU activation functions.

When x>0, the images of the two functions basically overlap; When x < 0, the Mish function gradually tends to 0, which can keep the function with a small negative value to prevent the gradient from disappearing. However, Leaky ReLU does not do a good job in keeping small negative values, which is not conducive to model updates.

The Mish function shows better results and faster convergence speed in experiments. It is adopted by target detection models such as YOLOv4, which greatly improves the accuracy of detection[22]. Therefore, the model uses the Mish function.

Then, Image features are flattened and passed through a fully connected layer to obtain encoded vectors:

$$e \in R^{b \times d}$$

where,

$$e_{i,j} = v_j * f_i + c_j$$

v_j and f_i are weights and biases of fully connected layers.

Then, the image feature will be sent to the encoder of transformer.

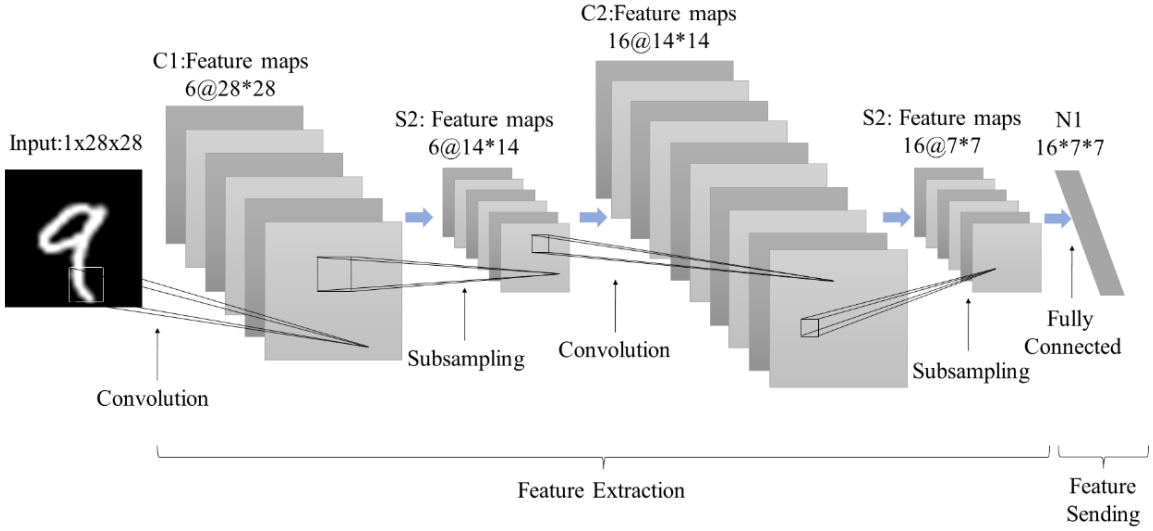

Figure 3.The overall architecture of Convolutional layers in extracting feature of MNIST.

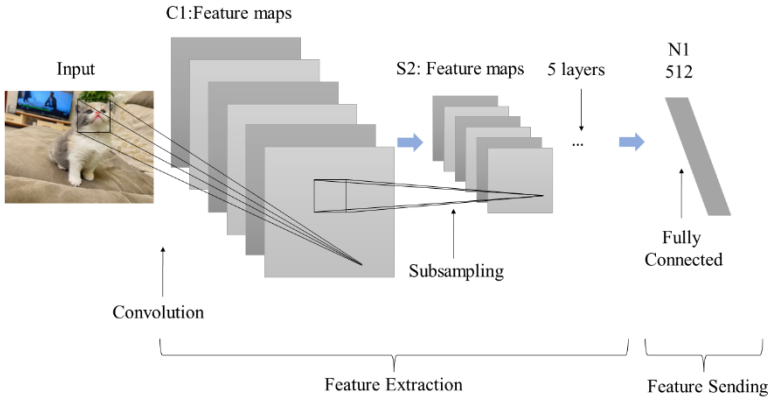

Figure 4.The overall architecture of Convolutional layers in extracting feature of CIFAR-10.

3.2 Transformer Layer

The transformer layer is designed to capture long-range dependencies and global context from the feature maps[11]. It is a standard encoder-decoder architecture with 8 self-attention mechanism. This allows the model to attend to different parts of the input and output sequences. Both the encoder and the decoder consist of several identical layers, with a multi-head self-attention sublayer and a feed-forward sublayer, connected by residual connections and layer normalization. The encoder takes the flattened feature maps from the convolutional layer as input and encodes them into a sequence of hidden states. Similarly, the decoder takes the same input as the encoder and generates a sequence of output states. In the multi-head self-attention sublayer, attention scores are computed between different positions of the input or output sequence and combined with a weighted sum. In the feed-forward sublayer, two linear layers with Mish activation function in between are used to obtain more detailed information to learn, with 4096 dimensions, twice the original size.

In the Convolutional Layer, image features have been processed as encoded vectors. The encoding vector is then added by one dimension and used as input and output to the Transformer, transformation vector can be got:

$$t \in R^{b\times 1\times d}$$

Then, the output of encoder is:

$$z \in R^{b\times 1\times d}$$

$$z = \mathrm{LayerNorm}(x + \mathrm{FFN}(\mathrm{LayerNorm}(x + \mathrm{MultiHeadAttention}(x, x, x))))$$

LayerNorm represents layer normalization, FFN is feedforward neural network, MultiHeadAttention represents the multi-head self-attention mechanism.

Decoder also consists of multiple identical layers, each contains three sublayers: the multi-head self-attention mechanism, multi-head encoder-decoder attention mechanism and feedforward neural networks. Each sublayer also has a residual connection and a layer normalization. Let the input be:

$$y \in R^{b\times m\times d}$$

where m is the target sequence length. The output of decoder is:

$$o = R^{b\times m\times d}$$

$$o = (\mathrm{LayerNorm}(y + \mathrm{FFN}(\mathrm{LayerNorm}(y + \mathrm{MultiHeadAttention}(o, o, o) + \mathrm{MultiHeadAttention}(z, z, y)))))$$

The final linear layer is a fully connected layer that maps the output of the decoder to the size of the target. Set the target size to v, then the output of the final linear layer is:

$$p \in R^{b\times m\times v}$$

where,

$$p_{i,j,k} = w_k * o_{i,j} + b_k$$

w_k and $o_{i,j}$ is the weight and bias of the fully connected layer.

The output of the final linear layer is used for softmax classification.

3.3 SAM Optimizer

The SAM optimizer[23] is a sharpness-aware optimizer used in this model to enhance the model's generalization performance and stability. It follows a two-step update process during each iteration. In the first step, the optimizer moves the parameters towards a local maximum along the gradient direction. Then, in the second step, a gradient descent update is performed from this local maximum. This approach helps the optimizer avoid sharp minima and instead locate flat minima, which are more resilient to noise and perturbations.

4. Experimental Results

4.1 Overall setting

The mixed model trains on python using torch. The experiment was carried out in Python 3.10 Anaconda environment on Windows 10 systems, with the configuration of Intel(R) i7- 10875H CPU @ 2.3GHz and Nvidia RTX 2060 6GB GPU.

The initial learning rate of the model is set at 0.0002, using the Adam optimizer for training, and use CosineAnnealingLR scheduler, the scheduler adjusts the learning rate in the way of cosine annealing.

Based on the characteristics of the cosine function, cosine annealing makes the function value show a downward trend of first slowing, then accelerating, and then slowing down with the increase of independent variables. In every epoch, the decay of learning rate is based on the following formula:

$$\eta_t = \eta_{min}^i + \frac{1}{2}\left(\eta_{max}^i - \eta_{min}^i\right)\left(1 + \cos\left(\frac{T_{cur}}{T_i}\pi\right)\right)$$

where η_t is learning rate, η_{min}^i and η_{max}^i are the minimum and maximum values of the learning rate, which define the range of learning rate. T_{cur} is the number of steps currently executed, T_i is the total number of steps in the ith run. Run refers to the training process after each restart. Restart refers to the method of jumping out of the local optimal solution and finding the global optimal solution by suddenly increasing the learning rate.

Cosine annealing can also be combined with warmup and hold strategies to improve the convergence speed and accuracy of the model. Warmup refers to the use of a small learning rate at the beginning of training, and linearly increases to the preset learning rate so that the model can gradually stabilize. Hold refers to keeping the learning rate constant for a period after the warmup is over before starting cosine annealing.

This method can make the model converge rapidly in the early stage of training, and then reduce the learning rate, which is helpful to further refine the training and improve the generalization ability.

When training on the MNIST, the original image is input as the training set.

When training on the CIFAR-10, to improve the generalization ability of the model and increase the number of data, the thesis preprocesses the images of the training set with RandomCrop, RandomResized, RandomHorizontalFlip and RandomErasing.

4.2 Pre-processing of the datasets

Regarding the MNIST, we do some rotation of 0-20 degrees with a probability of 0.5. This allows us to simulate the actual scenario of writing numbers more realistically. When it comes to the CIFAR-10, we employ several pre-processing ways. First, we resize and crop a random region of the images, using a scale factor between 0.8 and 1. Next, we flip the images horizontally with a probability of 0.5. Additionally, we erase a random rectangular region of the images, with an area between 4% and 20% of the image area, and an aspect ratio between 0.5 and 2. The erased region is then filled with random pixel values. The erased region is filled with random pixel values. Such pre-processing ways can strengthen the datasets.

4.3 MNIST

The MNIST[1] dataset is one of the most popular and widely used datasets for image classification and machine learning research. It consists of 70,000 grayscale images of handwritten digits (0-9), each of size 28 * 28 pixels. The dataset is divided into 60,000 training images and 10,000 testing images. This dataset has become a benchmark of image classification model. The thesis compares the result with other adopted convolutional structure like Lenet5. The results are shown in Table 1.

Table 1. Comparison with existing models on MNIST

Model		Accuracy(%)
Maxout[24]		99.55
NIN[25]		99.53
DSN[26]		99.61
#layers(Convolutional)		
Lenet5[1]	2	98.59
Mixed model	1	98.78
Mixed model	2	99.34
Mixed model	3	99.24

The mixed model didn't use any preprocessing, while other models used whiting and global contrast normalization.

When the numbers are random rotated for 0-20 degrees, the advantages of the model are revealed. The results are shown in Table 2.

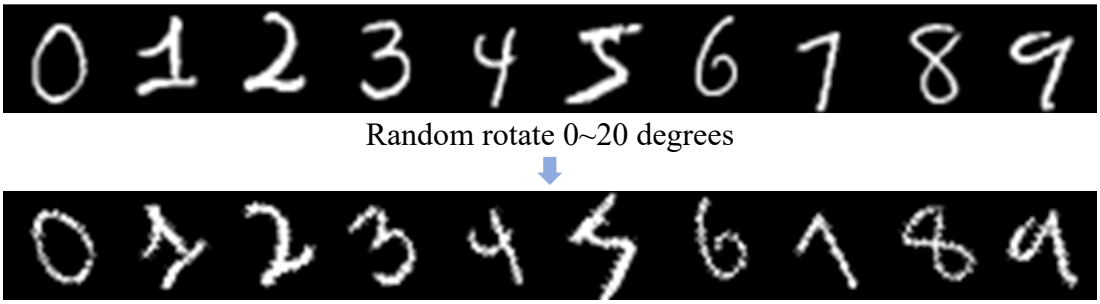

Figure 5. Some examples of MNIST in abnormal scenarios.

Table 2. Comparison with existing models on MNIST rotated.

Model	#layers	Accuracy (%)
Lenet5	3	97.23
Mixed model	1	95.62
Mixed model	2	99.18
Mixed model	3	98.07

4.4 CIFAR-10

The CIFAR-10 dataset is another common and widely used dataset for image classification and machine learning research. It consists of 60,000 color images of size 32 * 32 pixels, divided into 10 classes, with 6,000 images per class. The 10 classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset is split into 50,000 training images and 10,000 testing images. The thesis compares the result with other adopted convolutional structure like Resnet. The results are shown in Table 3.

Table 3. Comparison with existing models on CIFAR-10

Model		Accuracy(%)
Maxout[24]		90.32
NIN[25]		89.59
DSN[26]		90.31
#layers		
Resnet-20[4]	20	91.25

Mixed model	4	91.50
Mixed model	5	92.04

The mixed model only uses five convolutional layers to train, and exceeds the accuracy of ResNet-20 which has 20 residual layers. The mixed model reducing model complexity and compute resources.

When the images are random rotated for 0-10 degrees and random horizontal fliped, the advantages of the model are revealed. The results are shown in Table 4.

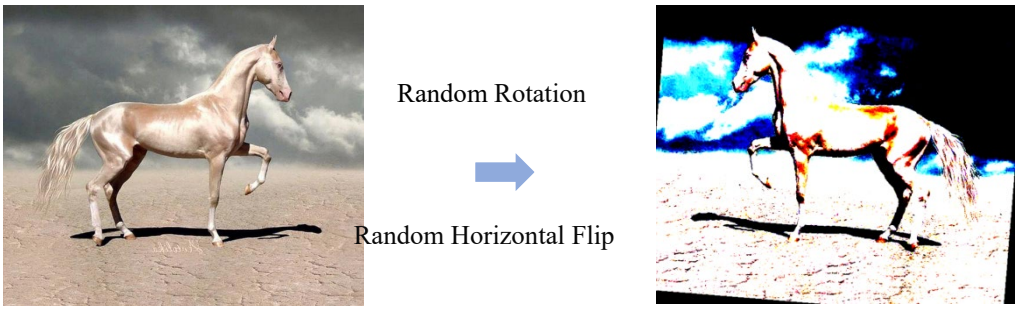

Figure 5. Example of CIFAR-10 in abnormal scenarios.

Table 4. Comparison with existing models on CIFAR-10 in abnormal scenarios

Model	#layers	Accuracy (%)
Resnet-20	20	82.44
Mixed model	4	82.53
Mixed model	5	84.63

5. Conclusion

Inspired by the disadvantages of CNN and the advantages of transformer, the thesis built a mixed model of CNN and transformer for image recognition. The model's main concept is to use a shallow CNN for extracting and pooling image features, which are then passed to the transformer for feature learning through the attention mechanism. By doing so, the model can learn more about the image features and the correlation between different locations without adding excessive depth. This approach improves the model's generalization ability. In experimental evaluations, the model outperformed existing models, particularly in abnormal scenarios. Additionally, the model can be further improved by adjusting the number of attention heads and convolutional layers for specific datasets. A challenging task now is how to reduce model parameters without affecting model accuracy, which might be a promising future research direction.

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, 'Gradient-based learning applied to document recognition', Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, 'ImageNet classification with deep convolutional neural networks', Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[3] K. Simonyan and A. Zisserman, 'Very Deep Convolutional Networks for Large-Scale Image Recognition'. arXiv, Apr. 10, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, 'Deep Residual Learning for Image Recognition', in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[5] R. Jiang and S. Mei, 'Polar Coordinate Convolutional Neural Network: From Rotation-Invariance to Translation-Invariance', in 2019 IEEE International Conference on Image Processing (ICIP), Sep. 2019, pp. 355–359. doi: 10.1109/ICIP.2019.8802940.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, 'Explaining and Harnessing Adversarial Examples'. arXiv, Mar. 20, 2015.

[7] H. Wu, R. Ding, H. Zhao, P. Xie, F. Huang, and M. Zhang, 'Adversarial Self-Attention for Language Understanding'. arXiv, Feb. 08, 2023.

[8] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, 'Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks'. arXiv, Mar. 14, 2016.

[9] I. Goodfellow, 'Gradient Masking Causes CLEVER to Overestimate Adversarial Perturbation Size'. arXiv, Apr. 20, 2018.

[10] C. Garbin, X. Zhu, and O. Marques, 'Dropout vs. batch normalization: an empirical study of their impact to deep learning', Multimed. Tools Appl., vol. 79, no. 19–20, pp. 12777–12815, May 2020, doi: 10.1007/s11042-019-08453-9.

[11] A. Vaswani et al., 'Attention Is All You Need', arXiv, 2017.

[12] A. Dosovitskiy et al., 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. arXiv, Jun. 03, 2021.

[13] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, 'General Multi-label Image Classification with Transformers', in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA: IEEE, Jun. 2021, pp. 16473–16483. doi: 10.1109/CVPR46437.2021.01621.

[14] L. Pan, C.-W. Hang, A. Sil, and S. Potdar, 'Improved Text Classification via Contrastive Adversarial Training'. arXiv, Feb. 17, 2022.

[15] Z. Jiang, T. Chen, T. Chen, and Z. Wang, 'Robust Pre-Training by Adversarial Contrastive Learning'. arXiv, Oct. 26, 2020.

[16] D. Chen, L. O'Bray, and K. Borgwardt, 'Structure-Aware Transformer for Graph Representation Learning'. arXiv, Jun. 13, 2022.

[17] C.-F. Chen, Q. Fan, and R. Panda, 'CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification'. arXiv, Aug. 22, 2021.

[18] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, 'Training Vision Transformers for Image Retrieval'. arXiv, Feb. 10, 2021.

[19] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, 'Densely Connected Convolutional Networks'. arXiv, Jan. 28, 2018. doi: 10.48550/arXiv.1608.06993.

[20] M. Tan and Q. V. Le, 'EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks'. arXiv, Sep. 11, 2020. doi: 10.48550/arXiv.1905.11946.

[21] D. Misra, 'Mish: A Self Regularized Non-Monotonic Activation Function'. arXiv, Aug. 13, 2020.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, 'YOLOv4: Optimal Speed and Accuracy of Object Detection'. arXiv, Apr. 22, 2020.

[23] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, 'Sharpness-Aware Minimization for Efficiently Improving Generalization'. arXiv, Apr. 29, 2021.

[24] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, 'Maxout Networks'. arXiv, Sep. 20, 2013.

[25] M. Lin, Q. Chen, and S. Yan, 'Network In Network'. arXiv, Mar. 04, 2014.

[26] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, 'Deeply-Supervised Nets'. arXiv, Sep. 25, 2014.