


# A reliable and low-cost deep learning model integrating convolutional neural network and transformer structure for fine-grained classification of chicken *Eimeria* species

Pengguang He <sup>\*,†,‡</sup> Zhonghao Chen,<sup>\*,†,‡</sup> Yefan He,<sup>\*,†,‡</sup> Jintian Chen,<sup>\*,†,‡</sup> Khawar Hayat,<sup>\*,†,‡</sup>  
Jinming Pan <sup>\*,†,‡</sup> and Hongjian Lin <sup>\*,†,‡,1</sup>

<sup>\*</sup>College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310058, China; <sup>†</sup>Key Laboratory of Intelligent Equipment and Robotics for Agriculture of Zhejiang Province, Hangzhou 310058, China; and <sup>‡</sup>Key Laboratory of Equipment and Informatization in Environment Controlled Agriculture, Ministry of Agriculture and Rural Affairs of China, Hangzhou 310058, China

**ABSTRACT** Chicken coccidiosis is a disease caused by *Eimeria* spp. and costs the broiler industry more than 14 billion dollars per year globally. Different chicken *Eimeria* species vary significantly in pathogenicity and virulence, so the classification of different chicken *Eimeria* species is of great significance for the epidemiological survey and related prevention and control. The microscopic morphological examination for their classification was widely used in clinical applications, but it is a time-consuming task and needs expertise. To increase the classification efficiency and accuracy, a novel model integrating transformer and convolutional neural network (CNN), named Residual-Transformer-Fine-Grained (ResTFG), was proposed and evaluated for fine-grained classification of microscopic images of seven chicken *Eimeria* species. The results showed that ResTFG achieved the best performance with high accuracy and low cost compared with traditional models. Specifically, the parameters,

inference speed and overall accuracy of ResTFG are 1.95M, 256 FPS and 96.9%, respectively, which are 10.9 times lighter, 1.5 times faster and 2.7% higher in accuracy than the benchmark model. In addition, ResTFG showed better performance on the classification of the more virulent species. The results of ablation experiments showed that CNN or Transformer alone had model accuracies of only 89.8% and 87.0%, which proved that the improved performance of ResTFG was benefit from the complementary effect of CNN's local feature extraction and transformer's global receptive field. This study invented a reliable, low-cost, and promising deep learning model for the automatic fine-grain classification of chicken *Eimeria* species, which could potentially be embedded in microscopic devices to improve the work efficiency of researchers and extended to other parasite ova, and applied to other agricultural tasks as a backbone.

**Key words:** chicken *Eimeria* classification, deep learning, convolutional and transformer structure, complementary effect

2023 Poultry Science 102:102459

<https://doi.org/10.1016/j.psj.2022.102459>

## INTRODUCTION

The global demand for protein products has been steadily increasing (Cao and Li, 2013). The poultry industry provides a large amount of meat and egg products for human consumption and its production scale is expected to further increase in the next decade (Mottet and Tempio, 2017; Blake et al., 2020). Chicken coccidiosis is a widespread and economically significant

disease caused by protozoan parasite of the genus *Eimeria* (Chapman et al., 2013; Mesa et al., 2021), costing the global broiler industry more than 14 billion dollars per year (Adams et al., 2022). There are 7 recognized *Eimeria* species, including *E. Tenella*, *E. Acervulina*, *E. Maxima*, *E. Brunetti*, *E. Mitis*, *E. Necatrix*, and *E. Praecox*. Chickens infected with different *Eimeria* species may occur clinical or subclinical symptoms because of significant differences in pathogenicity and virulence among different species (Shirley, 1997). Clinical coccidiosis is more harmful, which not only affects the yield and quality of meat and eggs, but also can cause chicken death with a high probability, while subclinical coccidiosis generally does not cause death (Engidaw and Getachew, 2018). In addition, different *Eimeria* species may have different drug resistance (Fatoba and

© 2022 The Authors. Published by Elsevier Inc. on behalf of Poultry Science Association Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Received October 2, 2022.

Accepted December 25, 2022.

<sup>1</sup>Corresponding author: [linhongjian@zju.edu.cn](mailto:linhongjian@zju.edu.cn)

Adeleke, 2018). Therefore, the successful identification of *Eimeria* species can provide guidance for treatment measures. And understanding the prevalence of coccidiosis can help the government to formulate macro-control policies, and rapid and accurate identification of *Eimeria* species can provide convenience for relevant researchers. Overall, it is of practical significance to distinguish *Eimeria* species for epidemiological survey and related prevention and control.

The molecular biological methods and microscopic morphological examination were widely adopted to identify parasites (Huang et al., 2017; Mattiello et al., 2000; Fotouhi-Ardakani et al., 2021; Hendershot et al., 2021). The former are accurate and sensitive but require sophisticated protocols, and the latter is a very challenging task for naked eyes due to the small morphological differences among chicken *Eimeria* species. Therefore, there is an urgent need to develop an automatic identification process for chicken *Eimeria* species. In some studies, The morphological characteristics of *Eimeria* oocysts were extracted and semi-automatic recognition was carried out by machine learning algorithms (Kucera and Reznicky, 1991; Castañón et al., 2007; Abdalla and Seker, 2017). Castañón et al. (2007) achieved the best overall accuracy of 85.75%. However, the semi-automatic methods requires manually designed features, which is cumbersome and the model accuracy is insufficient. The rapid development of convolutional neural network (CNN) has provided a powerful tool for the image recognition task (Esteva et al., 2017). Due to the superiority of CNN, it has been used for species identification of various parasites with good results and has been embedded in automated devices (Yang et al., 2020; Butploy et al., 2021; Lee et al., 2021; Thevenoux et al., 2021; Abade et al., 2022). However, there are few studies focusing on the classification of chicken *Eimeria* species using deep learning methods. Monge and Beltrán (2019) proposed a CNN model to classify chicken *Eimeria* species and the accuracy was improved to 90.42%, which still has room for improvement.

It is observed that the CNN-based models could achieve better results than traditional models that requires manual feature extraction. But these studies did not realize that the species recognition of some parasites, for example, chicken *Eimeria*, is a fine-grained classification task, which focusing on the classifying objects of similar but different subtypes (Zhao et al., 2020). The Transformer structure, originally proposed for Natural Language Processing (NLP) tasks (Vaswani et al., 2017; Jacob et al., 2019), which has been successfully applied in major computer vision tasks including fine-grained classification (Carion et al., 2020; Chen et al., 2021; Dosovitskiy et al., 2021; Zheng et al., 2021). And Transformer-Fine-Grained model (TransFG) achieved State-of-The-Art (SOTA) performance on five popular fine-grained classification benchmarks (He et al., 2021). The feature of local region connection makes CNN good at capturing local features, but lacks the ability to capture global features. Transformer can capture global features well, but is less capable of

capturing local features. Therefore, integrating CNN and Transformer structure could improve the model performance (Dai et al., 2021; Lu et al., 2022).

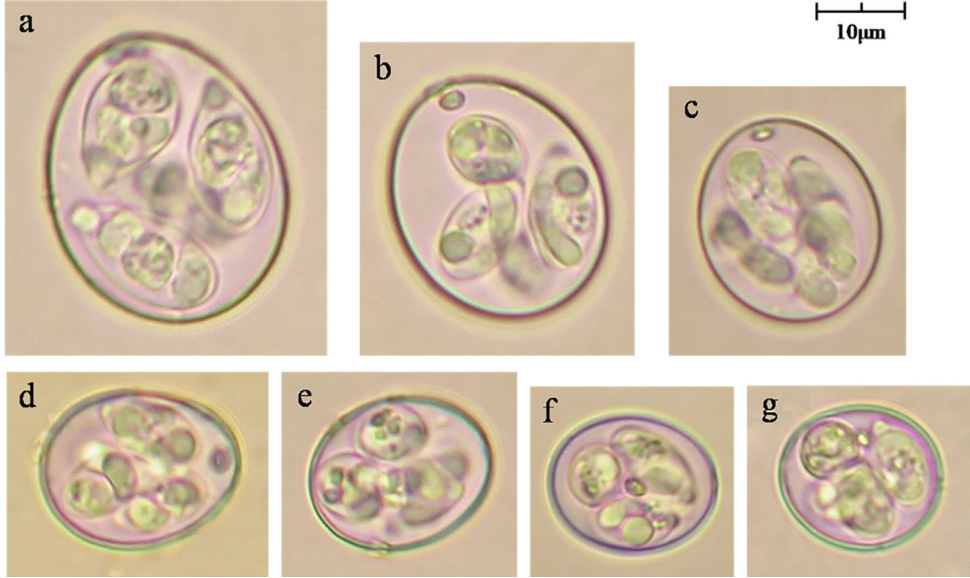
In this study, a novel model, named Residual-Transformer-Fine-Grained (**ResTFG**), was proposed for the classification of chicken *Eimeria* species based on the residual block (He et al., 2016) and TransFG (He et al., 2021). The main objectives of the present study were to 1) investigate the complementary effects of integrating the Transformer and CNN on the model performance; 2) optimize the number of layers and hyperparameters of the new model for better performance. The study eventually designed and validated a high performance and lightweight model suitable for automatic and real-time micrograph classification of seven chicken *Eimeria* oocysts.

## MATERIALS AND METHODS

### Dataset

**Dataset description** The dataset used in this study was from a publicly available website (<http://www.coccidia.icb.usp.br/>), created by the Laboratory of Molecular Biology of Coccidia at the Department of Parasitology of the Institute of Biomedical Sciences and the Cybernetic Vision Research Group at the Institute of Physics, the University of Sao Paulo, firstly published in 2007 (Castañón et al., 2007). The generality of data sources was considered. Several samples of each species were used, which were collected from different geographic sources, in order to dilute possible intra-specific variations and maximize inter-specific discrimination (Castañón et al., 2007). The dataset provides RGB digital micrographs of oocysts of seven *Eimeria* species of domestic fowl. Each micrograph contains multiple oocysts which are of the same species. To construct the classification image dataset, single oocyst was isolated from micrographs manually. In total, there were 7 categories, 4,243 labeled microscopic images of isolated oocysts. Since the morphological differences of seven oocysts, these images were of different sizes, and the maximum size was up to  $447 \times 642$  pixels (width by height) and the minimum size was  $177 \times 225$  pixels (width by height). Figure 1 shows the characteristic morphology of the 7 chicken *Eimeria* species.

**Dataset augmentation and splitting** There is an originally large difference in the number of different oocyst categories with uneven distribution, which would tend to result in a model with false classification of some uncertain samples into the category with more samples. Appropriate data augmentation was conducted for categories of images with fewer samples. Specifically, all *E. Maxima* images were flipped horizontally, and 300 *E. Brunetti* images and 200 *E. Necatrix* images were randomly selected for horizontal flipping, which is a commonly used method for dataset augmentation (Wang et al., 2019; Ye et al., 2020; Lu et al., 2022). Considering that the imaging environment of micrographs is controllable and consistent, and the original



**Figure 1.** Typical micrographs of oocysts of the seven chicken *Eimeria* species. Samples: (a) *E. Maxima*, (b) *E. Brunetti*, (c) *E. Tenella*, (d) *E. Necatrix*, (e) *E. Praecox*, (f) *E. Acervulina*, and (g) *E. Mitis*.

micrographs are sufficiently representative, it is not necessary to augment the original dataset by utilizing some morphological or color adjustment methods. After balancing the dataset, the total number of images increased from 4,243 to 5,103. Seventy percent of the samples of each category were randomly selected as the training set and the rest thirty percent as the test set. To ensure reproducible evaluation results, a constant random seed “2022” was set. The details of the dataset are shown in [Table 1](#).

## Methods

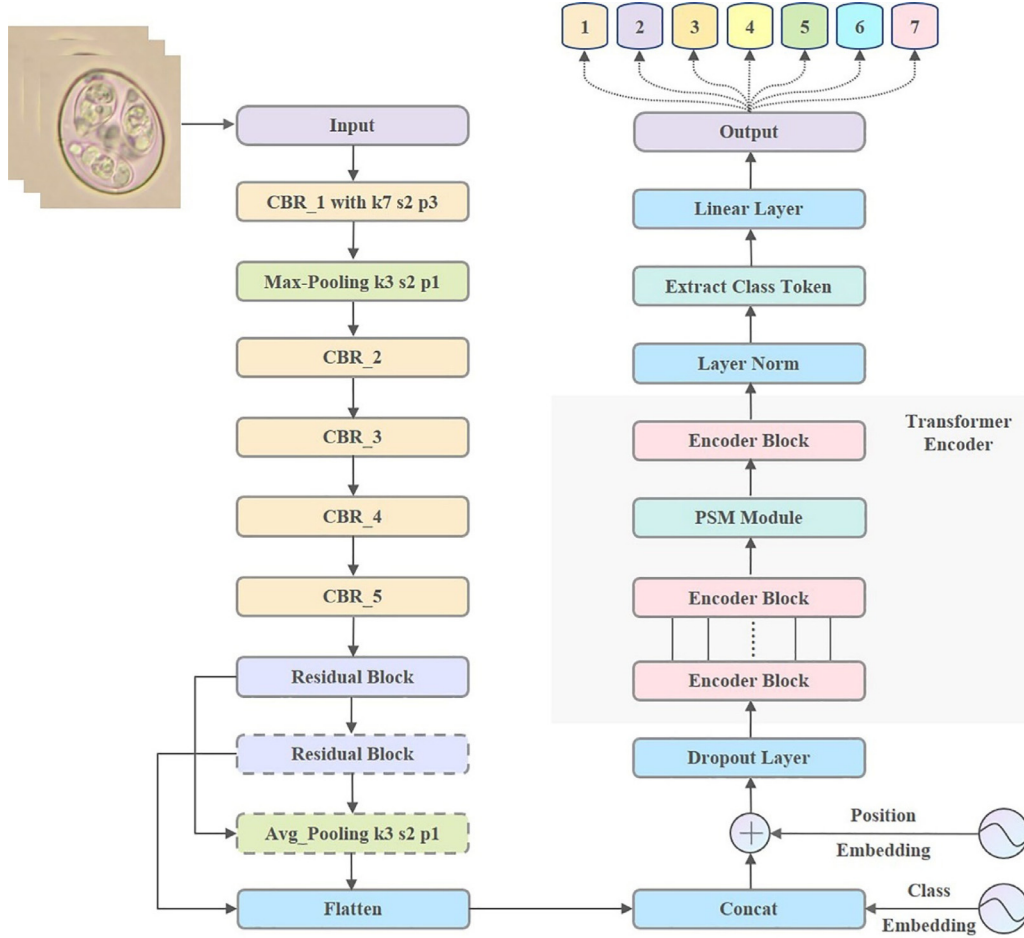
**Proposed model** The framework of the proposed ResTFG model is shown in [Figure 2](#). The left and the right parts are the CNN and Transformer branches, respectively.

The CNN branch consists of an input layer, a maximum pooling layer, an average pooling layer, a flatten layer, 5 CBR modules (acronym for Convolution Batch-Normalization ReLU (rectified linear unit)), and one or two residual blocks. The  $k$ ,  $s$ , and  $p$  in [Figure 2](#) represent the kernel size, stride, and padding of the convolution layer, respectively. As shown in [Figure 3\(a\)](#), the CBR module consists of three layers, a convolution layer with

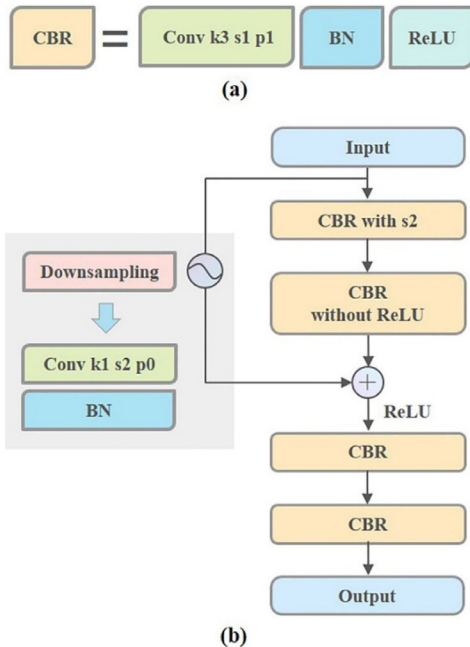
a kernel size of  $3 \times 3$ , a stride of  $1 \times 1$  and a padding of 1, followed by a batch normalization (BN) layer and a ReLU layer. The BN layer can speed up the training and convergence of the network, alleviate the gradient dispersion and mitigate the overfitting problem. In order to make the network capable of characterizing nonlinear mappings and further addressing the gradient disappearance problem, the ReLU activation function was added after each BN layer. The residual block is chosen as it can improve the trainability of the deep network with less computation cost. The structure of the residual block is shown in [Figure 3\(b\)](#), composed of an input layer, four CBR modules, a downsampling operation and an output layer. The downsampling operation is implemented by a convolution layer with a kernel size of  $1 \times 1$ , a stride of  $2 \times 2$  and no padding, and is connected to a BN layer afterward. The number of the residual block is uncertain (shown in the dashed box in [Figure 2](#)) since the structure in the model is subject to optimization. When there is only one residual block, it is then connected to the average pooling layer and the flattening layer. The second residual block is directly connected to the flattened layer when there are two residual blocks. This design aims to match the feature dimensions after the flattened layer with the input dimensions of the Transformer branch.

**Table 1.** The number of images of the chicken *Eimeria* oocyst dataset.

| Class label  | Species name         | Original | After data augmentation | Partitioning of the dataset (7:3) |       |
|--------------|----------------------|----------|-------------------------|-----------------------------------|-------|
|              |                      |          |                         | Training                          | Test  |
| ACE          | <i>E. Acervulina</i> | 742      | 742                     | 520                               | 222   |
| BRU          | <i>E. Brunetti</i>   | 442      | 742                     | 520                               | 222   |
| MAX          | <i>E. Maxima</i>     | 360      | 720                     | 504                               | 216   |
| MIT          | <i>E. Mitis</i>      | 825      | 825                     | 578                               | 247   |
| NEC          | <i>E. Necatrix</i>   | 502      | 702                     | 492                               | 210   |
| PRA          | <i>E. Praecox</i>    | 676      | 676                     | 474                               | 202   |
| TEN          | <i>E. Tenella</i>    | 696      | 696                     | 488                               | 208   |
| Total number |                      | 4,243    | 5,103                   | 3,576                             | 1,527 |



**Figure 2.** The overview of the proposed Residual-Transformer-Fine-Grained (ResTFG) model. The left and right parts are the convolutional neural network (CNN) and Transformer branch, respectively. The  $k$ ,  $s$ , and  $p$  represent the kernel size, stride, and padding of the convolution layer, respectively. When there is only one residual block, it is then connected to the average pooling layer and the flatten layer; when there are two residual blocks, the second one is directly connected to the flatten layer.



**Figure 3.** The structure of the Convolution Batch-Normalization ReLU (rectified linear unit) module (CBR) (a), and the residual block (b). The  $k$ ,  $s$ , and  $p$  represent the kernel size, stride, and padding of the convolution layer, respectively.

The Transformer branch was designed based on a previous study (He et al., 2021). The benefit of this intentionally simplified setting is to reduce the impact of other techniques on model performance (Dai et al., 2021). The final output vector dimension was set to seven to match the number of the chicken *Eimeria* species in this study. The Transformer branch is described in detail, including token and embedding operations, Transformer encoder block, and the optimization of loss function.

- (1) Tokens and Embeddings: For the Transformer in NLP, each word in the input sentence is divided into tokens, and a token representing semantic information is a class token. But for the Transformer in computer vision, it is not realistic to regard each pixel as a token due to the limitation of computation, so the image is cut into multiple patches, and each patch is regarded as a token for subsequent processing. In ResTFG models, the patch embedding operation is implemented by the CNN branch. The class token, that is, a learnable vector, is embedded in patch embedding to enable the model for categorization. Different from CNN, the Transformer structure

requires position embeddings to encode the location information of patch tokens, so a learnable vector, that is, position token, is added to the patch embedding. Assuming the input image is represented as  $x_p^N$  after being mapped to patch embedding space by the CNN branch, the output after position embedding is:

$$x_0 = [E_c; x_p^1; x_p^2; \dots; x_p^N] + E_{pos} \quad (1)$$

where  $x_0$  with category and spatial information is the input of the Transformer branch,  $N$  is the number of image patches,  $E_c$  is the class embedding operation, and  $E_{pos}$  is the position embedding operation.

(2) Transformer Encoder: Transformer Encoder is composed of  $L$  encoder block layers in total and one Part Select Module (**PSM**), where  $L$  is 12 by default.

a) Encoder block: As shown in Figure 4, the encoder block is composed of the alternating Layer Normalization (**LN**) layer, Multi-Head Self-Attention (**MSA**) layer, and Multi-layer Perceptron (**MLP**) block. The MLP block contains two linear layers, two dropout layers with a dropout rate of 0.1, and one Gaussian Error Linear Unit (**GeLU**) activation function layer. For each encoder block, the LN layer is applied before MSA and MLP layer, and the residual connection is applied after each LN layer. Therefore, the output of layer  $L$  can be expressed as:

$$\hat{x}_l = MSA(LN(x_{l-1})) + x_{l-1} \quad (2)$$

$$x_l = MLP(LN(\hat{x}_l)) + \hat{x}_l \quad (3)$$

where  $x_{l-1}$  is the output of layer  $L-1$ ,  $\hat{x}_l$  is the output of MSA layer.

MSA is developed from Self-Attention (**SA**) mechanism. The image classification task is abstracted into a query task through SA. Every patch token can be linked to the class token through SA. Patches mapped to the

embedding space are calculated by matrix  $Q$  (query) and matrix  $K$  (key) to obtain the attention weight. The weight is then fed into the Softmax function to calculate the dot product of it with matrix  $V$  (value) for the final output, i.e., the dot-product attention (Vaswani et al., 2017), expressed as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

where  $d_k$  is the dimension of matrix  $Q$  and  $K$ . The scaling of dot-product is necessary because the large  $d_k$  will result in a large value after  $QK^T$ , leading to a minimal gradient after the Softmax function, which is not conducive to the network training.

Rich characteristic information can generally be obtained by deepening the number of CNN channels, and each channel can be used to identify a different pattern. Similarly,  $K$  attention heads are set in the MSA for better feature capture capability. In this study, the number of attention heads were 12 by default. The output of MSA could be expressed as (Vaswani et al., 2017):

$$\begin{aligned} MultiHead(Q, K, V) \\ = Concat(head_1, \dots, head_i)W^O \end{aligned} \quad (5)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (6)$$

b) PSM module: The input of the last encoder block is modified with the PSM layer to take full advantage of the attention information. Previous studies have pointed out that raw attention weights do not necessarily correspond to the relative importance of input tokens, so it is necessary to integrate the attention weights of all previous layers (Abnar and Zuidema, 2020). Specifically, PSM recursively applies matrix multiplication to the raw attention weights in all encoder block layers to get  $a_{final}$ . The tokens corresponding to the index  $A_1, A_2, \dots, A_K$  of the maximum  $K$  attention heads in  $a_{final}$  are selected and spliced with the class token as the input to the last encoder block (He et al., 2021), which can be expressed as:

$$x_{select} = [x_{l-1}^0; x_{l-1}^{A_1}, x_{l-1}^{A_2}, \dots, x_{l-1}^{A_K}] \quad (7)$$

$$a_{final} = \prod_{l=0}^{L-1} a_l \quad (8)$$

$$a_l = [a_l^0, a_l^1, \dots, a_l^K] \quad (9)$$

$$a_l^i = [a_l^{i_0}, a_l^{i_1}, \dots, a_l^{i_K}] \quad (10)$$

(3) Loss Function: The contrastive loss is introduced into the model to minimize the similarity of the class

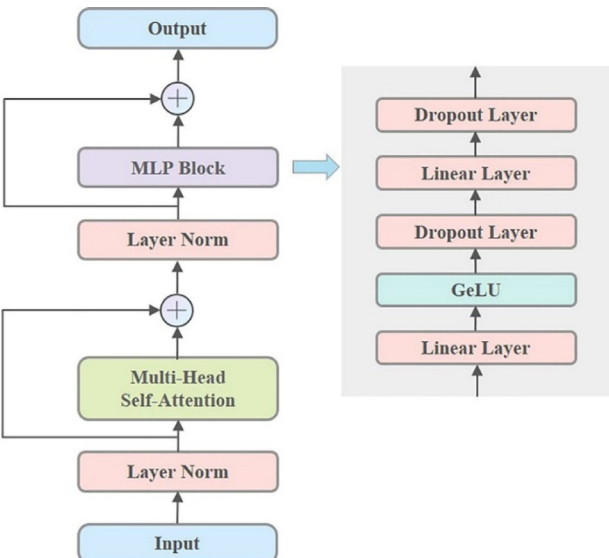


Figure 4. The structure of the transformer encoder block.

tokens of different categories and maximize the similarity of the class tokens of the same category. The total loss  $L_{total}$  is the sum of contrastive loss  $L_{con}$  and cross-entropy loss  $L_{cross}$ , which can be expressed as (He et al., 2021):

$$L_{total} = L_{con}(x) + L_{cross}(y, y') \quad (11)$$

$$L_{con} = \frac{1}{N^2} \sum_i^N \left[ \sum_{j: y_i=y}^N (1 - sim(x_i, x_j)) + \sum_{j: y_i \neq y}^N \max(sim(x_i, x_j) - \alpha, 0) \right] \quad (12)$$

$$L_{cross} = - \sum_{i=1}^n y'_i \log(y_i) \quad (13)$$

where  $N$  is the batch size,  $x_i$  and  $x_j$  are pre-processed with L2 normalization,  $sim$  is cosine similarity,  $\alpha$  is an artificially constant to avoid the loss affected by simple negative samples,  $y$  is the ground-truth label, and  $y'$  is the predicted label.

**Equipment and Environment** To facilitate intensive computation in model training, a professional deep learning platform, SYS-4029GP-TRT was used, equipped with  $2 \times$  Intel© Xeon(R) Gold 6147M CPU @ 2.50GHz, a total of 260 GB memory, and 8 graphics cards including  $4 \times$  Nvidia TITAN RTX and  $4 \times$  Nvidia GeForce RTX 2080 Ti, a total of 140 GB video memory. The testing and inference speed measurement of models were run on a desktop computer with GeForce RTX 3080 GPU and Inter(R) Core (TM) i9-10900KF CPU @3.70GHz. In terms of the software environment, Python-3.8, PyCharm-Professional-2021.2.3, and Pytorch-GPU-1.8.1 framework were used.

**Experimental setting** First, 7 existing SOTA models were compared, including VGG11 (Simonyan, 2014), MobilenetNet\_V3\_Small, MobilenetNet\_V3\_Large (Howard et al., 2019), ResNet34 (He et al., 2016), DenseNet121 (Huang et al., 2016), Shufflenet\_V2\_x1\_0 (Ma et al., 2018), and TransFG\_B16 (He et al., 2021). VGGNet11, ResNet34, and DenseNet121 with their good feature extraction capability and generalization performance have become the preferred backbone of many downstream tasks, and have achieved competitive accuracy in many image recognition applications. Mobilenet-Net\_V3\_Small, MobilenetNet\_V3\_Large, and Shufflenet\_V2\_x1\_0 models are a class of lightweight models with relatively low accuracy but fast inference speed. The TransFG\_B16 with a batch size of 16 is one of the TransFG models, which is a Transformer based model specifically designed for fine-grained image classification task. After the comprehensive evaluation of the above 7 SOTA models, a benchmark model was obtained for the subsequent comparison with ResTFG models.

Then a series of experiments were conducted to optimize ResTFG models. First, the effect of 3 hyper-parameters and the number of encoder block layers in the Transformer branch on model performance were evaluated. After determining the optimal setting of the

Transformer branch, the structure of the CNN branch was further optimized. Furthermore, the effectiveness of the integration of CNN and Transformer structure was proved through ablation experiments.

### Image Preprocessing and Hyperparameters setting

The computational cost of the CNN is significantly related to the input image size, which should be sufficiently reduced but without affecting the model performance. In this study, to modify images as little as possible, only two preprocessing steps were adopted before inputting the images into the CNN structure. First, each image was adjusted to a commonly used format for the image classification task with a resolution of  $224 \times 224$  using the resize method in PyTorch deep learning framework. Second, each pixel value was normalized from to  $[0, 1]$  to speed up the convergence of the model.

All models used were trained from scratch. The initial learning rate is 0.001, and the decay rate is 0.5 for every 20 epochs. In addition, the number of epochs is 100, the batch size is 64, and the optimizer is Stochastic Gradient Descent (SGD) with a 0.9 momentum and a  $1e-4$  weight decay. The models proposed in this paper use the sum of cross-entropy loss and contrastive loss as the total loss, while other models only use cross-entropy to calculate the loss.

**Evaluation Metrics** In practical application scenarios, the deployment of models will be limited by computational resources. In this study, three metrics, that is, accuracy, the number of parameters, and inference speed (FPS, frames per second), are used for a comprehensive evaluation of model performance to obtain high performance and lightweight model. The receiver operating characteristic (ROC) curve, area under the ROC curve (AUC) and the loss curve of the training stage were also used to compare the performance. In addition, to evaluate the recognition performance of the model for each category of chicken *Eimeria* species, confusion matrix, precision, recall, and F1 score were adopted. All results are the average value of 3 tests. The calculation formulas of metrics are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (15)$$

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

Where *Precision* is the proportion of the correct sample to the total sample, *Accuracy* is the proportion of all samples correctly predicted by the model, *Recall* is the proportion of all real samples that the model predicts to be correct, *F1score* is the harmonic mean of *Precision* and *Recall*, True Positive (*TP*) is the number of samples correctly predicted to be positive, True Negative (*TN*)

is the number of samples correctly predicted to be negative, False Positive ( $FP$ ) is the number of samples falsely predicted to be positive, and False Negative ( $FN$ ) is the number of samples falsely predicted to be negative.

## RESULTS

### Performance of Existing Models

The test results of the seven existing SOTA models are shown in Table 2. As expected, the accuracy of MobilenetNet\_V3\_Small, MobilenetNet\_V3\_Large and Shufflenet\_V2\_x1\_0 was relatively low. Shufflenet\_V2\_x1\_0 has the lowest number of parameters and accuracy (80.4%), which could not meet the recognition requirement. It was surprising VGG11, with the maximum number of parameters, achieved the fastest inference speed of 409 FPS, but its memory consumption is high. DenseNet121 had a good accuracy, but very slow inference speed. TransFG\_B16 also had good accuracy but high memory consumption. ResNet34 was identified as a balanced model with 21.29M parameters, 166FPS and 94.2% accuracy, and therefore, was selected as the benchmark model.

### Optimization of Transformer Branch

Two residual blocks with 13 convolution layers were adopted in ResTFG models by default. The initial value of three hyperparameters in the Transformer branch, that is, hidden size, MLP dimension and the number of multi-attention heads was 768, 3,072, and 12 respectively, which were set according to TransFG\_B16. MLP

dimension was set to four times of hidden size, so it was not considered a separate hyperparameter. The more multi-attention heads mean the more patterns of correlation between different patches can be learned. The number of the encoder block layer was adjusted at 8 to limit the size of ResTFG models. The initial model was named ResTFG (C13, H12, L8) (a), where  $[C]$  represents the number of convolution layers,  $[H]$  represents the number of multi attention head,  $[L]$  represents the number of the encoder block layer, and the letter suffixes correspond to the different hidden size and MLP dimension.

The hidden size and MLP dimension were first optimized. As shown in Table 3, ResTFG (C13, H12, L8) (a) achieved the highest accuracy of 97.2%, but it had too many parameters. When the hidden size was set to 384, the ResTFG (C13, H12, L8) (b) obtained a tradeoff among parameters, inference speed and accuracy. Then, the hyperparameter  $H$  was optimized. It was found that the number of parameters was almost unaffected by  $H$ , but the model accuracy was negatively affected by a decreasing  $H$ , so  $H$  was always set to 12 in subsequent models. Furthermore, the evaluation of hyperparameter  $L$  showed the accuracy and inference speed increased with the decrease of  $L$ , which might imply a saturation of the model. A similar result occurred in Lu et al.’s study (Lu et al., 2022). Therefore, ResTFG (C13, H12, L2) achieved the best results with 10.86M number of parameters, 216FPS, and 97.1% accuracy. Compared with ResTFG (C13, H12, L8) (a), the number of parameters in ResTFG (C13, H12, L2) was reduced by 86%, and the inference speed was doubled, but the accuracy was only decreased by 0.1%, which is significantly superior to the performance of ResNet34.

**Table 2.** The performance of State-of-The-Art (SOTA) models in the classification of chicken *Eimeria* species. TransFG\_B16 with a batch size of 16 is one of the Transformer-Fine-Grained (TransFG) models.

| Model Name            | Parameters (M) | Speed (FPS) | Accuracy (%) |
|-----------------------|----------------|-------------|--------------|
| VGG11                 | 128.80         | <b>409</b>  | 86.9         |
| MobilenetNet_V3_Small | 1.53           | 157         | 86.3         |
| MobilenetNet_V3_Large | 4.21           | 107         | 90.7         |
| <b>ResNet34</b>       | 21.29          | 166         | <b>94.2</b>  |
| DenseNet121           | 6.96           | 56          | 92.7         |
| Shufflenet_V2_x1_0    | <b>1.26</b>    | 146         | 80.4         |
| TransFG_B16           | 85.80          | 104         | 93.5         |

**Table 3.** The performance comparison of the Residual-Transformer-Fine-Grained (ResTFG) for the Transformer branch with different hyperparameters and the number of the encoder block layer. TransFG\_B16 with a batch size of 16 is one of the Transformer-Fine-Grained (TransFG) models.

| Model Name                   | Hidden size | MLP dimension | Number heads | Number Layers | Parameters (M) | Speed (FPS) | Accuracy (%) |
|------------------------------|-------------|---------------|--------------|---------------|----------------|-------------|--------------|
| TransFG_B16                  | 768         | 3,072         | 12           | 12            | 85.80          | 104         | 93.5         |
| ResTFG (C13, H12, L8) (a)    | 768         | 3,072         | 12           | 8             | 82.14          | 105         | <b>97.2</b>  |
| ResTFG (C13, H12, L8) (b)    | <b>384</b>  | <b>1,536</b>  | 12           | 8             | 21.50          | 112         | 97.0         |
| ResTFG (C13, H12, L8) (c)    | 288         | 1,024         | 12           | 8             | 11.90          | 113         | 95.7         |
| ResTFG (C13, H12, L8) (d)    | 192         | 768           | 12           | 8             | <b>6.12</b>    | 116         | 95.7         |
| ResTFG (C13, H8, L8)         | 384         | 1,536         | 8            | 8             | 21.50          | 114         | 96.7         |
| ResTFG (C13, H4, L8)         | 384         | 1,536         | 4            | 8             | 21.50          | 114         | 96.6         |
| ResTFG (C13, H2, L8)         | 384         | 1,536         | 2            | 8             | 21.50          | 114         | 96.0         |
| ResTFG (C13, H12, L6)        | 384         | 1,536         | <b>12</b>    | 6             | 17.96          | 134         | 96.5         |
| ResTFG (C13, H12, L4)        | 384         | 1,536         | 12           | 4             | 14.41          | 165         | 96.9         |
| ResTFG (C13, H12, L3)        | 384         | 1,536         | 12           | 3             | 12.63          | 183         | 97.0         |
| <b>ResTFG (C13, H12, L2)</b> | 384         | 1,536         | 12           | <b>2</b>      | 10.86          | <b>216</b>  | 97.1         |

**Table 4.** The performance comparison of the Residual-Transformer-Fine-Grained (ResTFG) for the convolutional neural network (CNN) branch with different kernel parameters and the number of the convolution layer.

| Model Name                      | Hidden size | MLP dimension | Parameters (M) | Speed (FPS) | Accuracy (%) |
|---------------------------------|-------------|---------------|----------------|-------------|--------------|
| ResTFG (C13, H12, L2)           | 384         | 1,536         | 10.86          | 216         | <b>97.1</b>  |
| ResTFG (C9, H12, L2) (a)        | 384         | 1,536         | 10.09          | 254         | 96.5         |
| ResTFG (C9, H12, L2) (b)        | 180         | 720           | 2.50           | 256         | 96.7         |
| <b>ResTFG (C9, H12, L2) (c)</b> | <b>156</b>  | <b>624</b>    | 1.95           | 256         | 96.9         |
| ResTFG (C5, H12, L2)            | 156         | 624           | <b>1.80</b>    | <b>299</b>  | 96.2         |

### Optimization of CNN Branch

The convolution layers of the CNN branch were pruned and the parameters of the convolution kernel were adjusted in order to make the model more lightweight. As shown in Table 4, after pruning one residual block, i.e., four convolution layers, the accuracy of ResTFG (C9, H12, L2) (a) only decreased by 0.6%, but there was a 17% improvement in inference speed. It is a good result, but the model is not light enough and there is a possibility to further reduce its memory consumption. Therefore, the kernel size (same value with hidden size) of convolution layers in the residual block was optimized. The results showed that the kernel size strongly influences the number of parameters of the model. When it was reduced from 384 to 156, the number of parameters was reduced from 10.09M to 1.95M which the accuracy was improved by 0.2%. When the number of convolution layers was further reduced, the bonus in the number of model parameters was negligible, but the decrease in accuracy was significant. Therefore, ResTFG (C9, H12, L2) (c) (referred to as optimized ResTFG) obtained the advantages of both high performance and lightweight, with the number of parameters of 1.95M, an inference speed of 256FPS, and an accuracy of 96.9%, which is 10.9 times lighter, 1.5 times faster, and 2.7% higher in accuracy than ResNet34. Table 5 shows the related parameters setting and output shape of each layer of ResTFG (C9, H12, L2) (c).

### Ablation Studies on ResTFG

To demonstrate that the integration of the CNN and Transformer structure have a positive effect on model performance improvement, the ablation experiments were designed and implemented. Specifically, the Transformer branch and the CNN branch in ResTFG (C9, H12, L2) (c) were removed separately so that only one branch could be used for testing. The test results are shown in Table 6. There is no doubt that the number of parameters would be reduced and the inference speed would increase with only the CNN or Transformer branch. But as Table 6 shows, the accuracy of these two models decreased significantly by 7.1% and 9.9%, respectively. The results showed sufficient evidence that the hybrid model integrating the CNN branch and Transformer branch can fully utilize the advantages of both, which is highly superior to the model with a single branch.

**Table 5.** The parameters setting and corresponding output shape of each layer of Residual-Transformer-Fine-Grained (ResTFG) (C9, H12, L2) (c).

| Layer name     | Kernel size | Stride | Padding | Output shape   |
|----------------|-------------|--------|---------|----------------|
| Input          |             |        |         | 224 × 224 × 3  |
| CBR_1          | 7 × 7       | 2      | 3       | 112 × 112 × 64 |
| Max-pooling    | 3 × 3       | 2      | 1       | 56 × 56 × 64   |
| CBR_2          | 3 × 3       | 1      | 1       | 56 × 56 × 64   |
| CBR_3          | 3 × 3       | 1      | 1       | 56 × 56 × 64   |
| CBR_4          | 3 × 3       | 1      | 1       | 56 × 56 × 64   |
| CBR_5          | 3 × 3       | 1      | 1       | 56 × 56 × 64   |
| Residual Block |             |        |         | 28 × 28 × 156  |
| Avg-pooling    | 3 × 3       | 2      | 1       | 14 × 14 × 156  |
| Flatten        |             |        |         | 196 × 156      |
| Embedding      |             |        |         | 197 × 156      |
| Encoder blocks |             |        |         | 197 × 156      |
| Linear         |             |        |         | 7              |

**Table 6.** The performance of the Residual-Transformer-Fine-Grained (ResTFG) with only convolutional neural network (CNN) branch or Transformer branch.

| Model Name                      | Parameters (M) | Speed (FPS) | Accuracy (%) |
|---------------------------------|----------------|-------------|--------------|
| CNN Only                        | <b>0.92</b>    | <b>549</b>  | 89.8         |
| Transformer Only                | 1.03           | 403         | 87.0         |
| <b>ResTFG (C9, H12, L2) (c)</b> | 1.95           | 256         | <b>96.9</b>  |

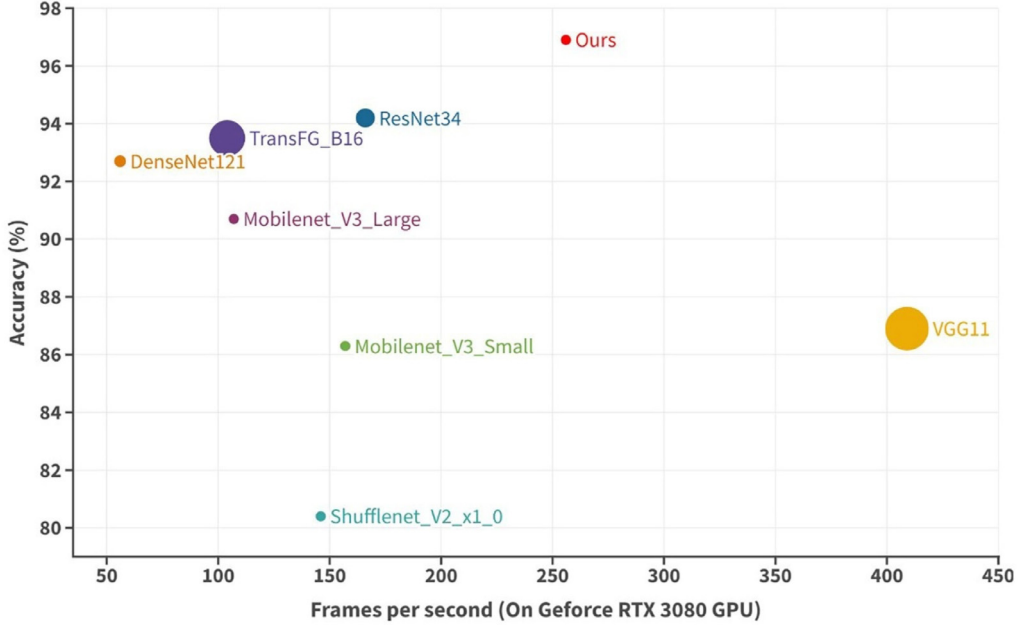
## DISCUSSION

### Balance of Accuracy and Inference Speed

Due to the limited computing resources of embedded and mobile devices, in addition to accuracy, memory consumption and inference speed of the model are also important. If a model has the advantages of high accuracy and low cost at the same time, it has more potential to be applied to the actual scenarios to automatically identify chicken *Eimeria* species.

Based on the collected microscopic image dataset of chicken *Eimeria* oocysts, 7 existing SOTA models were tested. Although VGG11 had the fastest inference speed (409FPS), its parameters reached 128.80M, which could not meet the requirements of lightweight model. ShuffleNet\_V2\_x1\_0 has an advantage in the number of parameters (1.26M), but its accuracy is only 80.4%. Compared with other models, ResNet34 achieved good number of parameters (21.29M), inference speed (166FPS) and accuracy (94.2%), but it was still inferior to our proposed model. And there is only two ResTFG models have an accuracy below 96%, with the highest





**Figure 5.** The bubble plots of accuracy (%), inference speed (frames per second) and the number of parameters of seven State-of-The-Art (SOTA) models and the optimized Residual-Transformer-Fine-Grained (ResTFG) model (Ours).

accuracy of 97.2%. Among all the ResTFG models, ResTFG (C5, H12, L2) achieved the fewest number of parameters (1.80M) and the fastest inference speed (299FPS). But ResTFG (C9, H12, L2) (c) (referred to as Ours) is regarded as a balanced model with the advantages of both high performance and lightweight. Figure 5 shows the bubble plots of accuracy, inference speed and the number of parameters of 7 existing SOTA models and Ours, with larger bubbles representing more parameters and vice versa. Ours is in the upper right of the figure, and the bubble size is significantly smaller than ResNet34.

In addition to the overall accuracy, the ability of models to identify the more virulent *Eimeria* species is also critical. Table 7 shows the recognition accuracy and recall of each *Eimeria* species of the benchmark model ResNet34, ResTFG (C13, H12, L8) (b), ResTFG (C13, H12, L2) and ResTFG (C9, H12, L2) (c) (Ours). The comprehensive recognition performance was evaluated by F1 score values, which was calculated by Equation 15. ResTFG (C13, H12, L8) (b) and ResTFG (C13, H12, L2) were selected because of their superior performance in the model optimization process. Among the seven *Eimeria* species, *E. Tenella* showed the highest

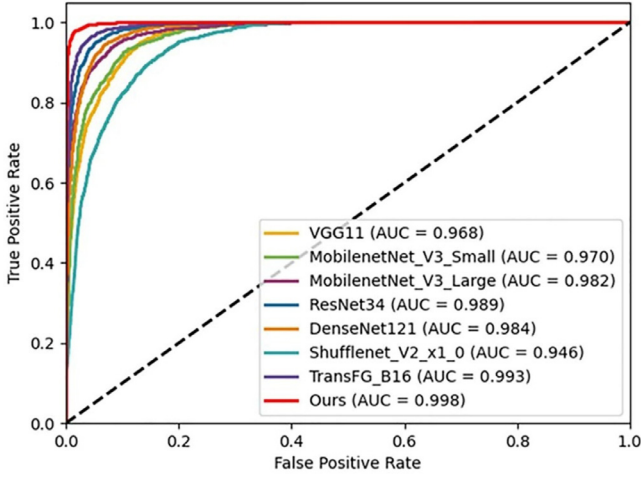
virulence, followed by *E. Necatrix* (Shirley, 1997). Therefore, the identification performance of models on these two species has been focused on. As can be seen from Table 7, compared with the ResNet34, the optimized model (Ours) has larger F1 score values, that is, the comprehensive recognition performance of *E. Tenella* and *E. Necatrix* is better. In addition, compared with ResTFG (C13, H12, L2) and ResTFG (C13, H12, L8) (b), Ours improved the recognition performance of *E. Necatrix* without affecting the recognition ability of *E. Tenella*. These results showed that the model proposed in this study is of practical significance, and the optimization strategy is feasible.

## ROC Curve

The performance of multiple methods on the same task can be easily compared by the ROC curve, the area enclosed by the curve and the coordinate axes is called AUC, which is insensitive to the category distribution. The vertical coordinate of the ROC curve is TPR (True Positive Rate) and the horizontal coordinate is the FPR (False Positive Rate), and the closer the curve is to the

**Table 7.** Precision, recall, and F1 score of the optimized Residual-Transformer-Fine-Grained (ResTFG) and ResNet34 model for each class.

| Models<br>Class label | ResTFG (C9, H12, L2) (c) (Ours) |        |             | ResNet34  |        |          | ResTFG (C13, H12, L8) (b) |        |          | ResTFG (C13, H12, L2) |        |          |
|-----------------------|---------------------------------|--------|-------------|-----------|--------|----------|---------------------------|--------|----------|-----------------------|--------|----------|
|                       | Precision                       | Recall | F1 score    | Precision | Recall | F1 score | Precision                 | Recall | F1 score | Precision             | Recall | F1 score |
| ACE                   | 0.97                            | 0.95   | 0.96        | 0.95      | 0.91   | 0.93     | 0.97                      | 0.95   | 0.96     | 0.97                  | 0.99   | 0.98     |
| BRU                   | 0.96                            | 0.98   | 0.97        | 0.90      | 0.98   | 0.94     | 0.96                      | 0.99   | 0.98     | 0.96                  | 0.97   | 0.97     |
| MAX                   | 0.98                            | 0.99   | 0.99        | 0.98      | 0.95   | 0.96     | 0.99                      | 0.99   | 0.99     | 0.98                  | 0.99   | 0.98     |
| MIT                   | 0.98                            | 0.98   | 0.98        | 0.96      | 0.97   | 0.96     | 0.97                      | 0.97   | 0.97     | 0.98                  | 0.98   | 0.98     |
| NEC                   | 0.97                            | 0.96   | <b>0.97</b> | 0.95      | 0.91   | 0.93     | 0.98                      | 0.95   | 0.96     | 0.97                  | 0.94   | 0.96     |
| PRA                   | 0.93                            | 0.95   | 0.94        | 0.88      | 0.94   | 0.91     | 0.93                      | 0.97   | 0.95     | 0.93                  | 0.97   | 0.95     |
| TEN                   | 0.99                            | 0.97   | <b>0.98</b> | 0.98      | 0.93   | 0.96     | 0.98                      | 0.97   | 0.98     | 0.99                  | 0.97   | 0.98     |



**Figure 6.** The ROC curves of seven State-of-The-Art (SOTA) models and the optimized Residual-Transformer-Fine-Grained (ResTFG) model (Ours).

upper left as well as the larger the ACU value represents the better model performance.

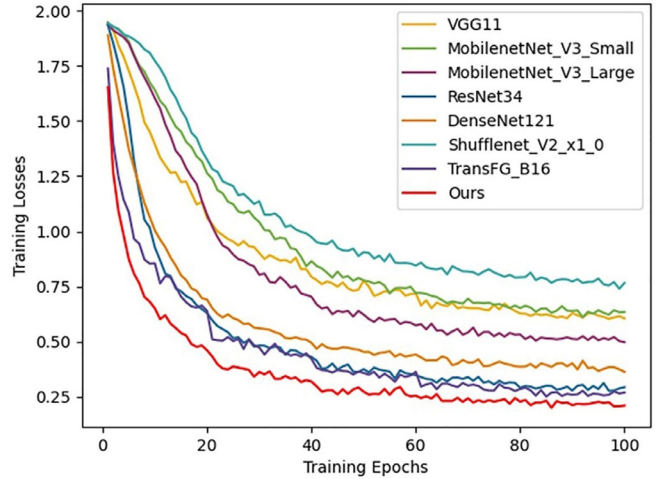
As shown in Figure 6, the ROC curve of the optimized ResTFG (Ours) is closest to the upper left and the AUC value reaches 0.998, which is better than all other models, where the AUC value of ResNet34 is 0.989. Therefore, our model achieved the best performance.

### Convergence Situation

Besides focusing on the performance of the model after being well-trained, the convergence situation of the model during the training stage is also one of the most important concerns for researchers. Figure 7 shows the convergence process of the average loss of seven SOTA models and the optimized ResTFG model with the increase of training iterations. As shown in Figure 7, the average loss value of all the eight models decreased rapidly in the first 20 epochs and converged after about 80 epochs. As can be seen that the convergence of VGG11, MobileNet\_V3\_Small, MobileNet\_V3\_Large, and Shufflenet\_V2\_x1\_0 in this task is relatively poor, and of ResNet34 and TransFG\_B16 is similar except that the initial loss value of TransFG\_B16 is lower. The solid red line represents the optimized ResTFG. Its loss value is always the lowest among all models no matter when the training is stopped. It also has the fastest convergence speed, and it reaches the best performance of ResNet34 after training 50 epochs. In addition, the fluctuation of its loss value is the smallest in the late training period. The results proved that our model has a robust and stable learning capability.

### Confusion Matrix

The confusion matrix of the optimized ResTFG is illustrated in Figure 8, which shows the classification



**Figure 7.** The loss curves of seven State-of-The-Art (SOTA) models and the optimized Residual-Transformer-Fine-Grained (ResTFG) model (Ours).

performance of the model for each class. According to Equation 14, the overall accuracy is 96.9%.

The identification accuracy, recall and F1 score of each class are calculated with Equations 15 to 17, and the results are shown in Table 7. As the table shows, the samples of *E. Praecox* species are more likely to be misclassified, which is the same as in César et al.'s study (2007). In their study, the morphological features and a Bayesian classifier were used to identify seven *Eimeria* species, and the results showed that *E. Praecox* species had the worst classification accuracy (74.2%), followed by *E. Necatrix* species (74.9%). And in this study, *E. Praecox* species also had the worst classification accuracy of 93%, while *E. Necatrix* species achieved 97%. It can be seen that the classification accuracy has been greatly improved. The poor performance of different models for the classification of *E. Praecox* species suggests that *E. Praecox* is the most difficult to be differentiated correctly among the 7 *Eimeria* species, which is due to the objective fact of morphological similarities between *E. Praecox* and the other species (Kucera and Reznicky, 1991)

In this study, the reason preventing the further improvement of the model accuracy is the mutual misclassification between *E. Acervulina* and *E. Praecox* species. Eight *E. Acervulina* samples were misclassified as *E. Acervulina* and five *E. Praecox* samples were misclassified as *E. Acervulina*. The result provides a direction for the further optimization of the mode. On the one hand, the penalty on *E. praecox* misclassification can be increased during model training to force the model to learn more features of *E. praecox* species. On the other hand, for data-driven deep learning methods, increasing the number of samples can be tried, especially *E. Praecox* species which is hard to distinguish, to make the model have better recognition ability and enhance its generalization performance.

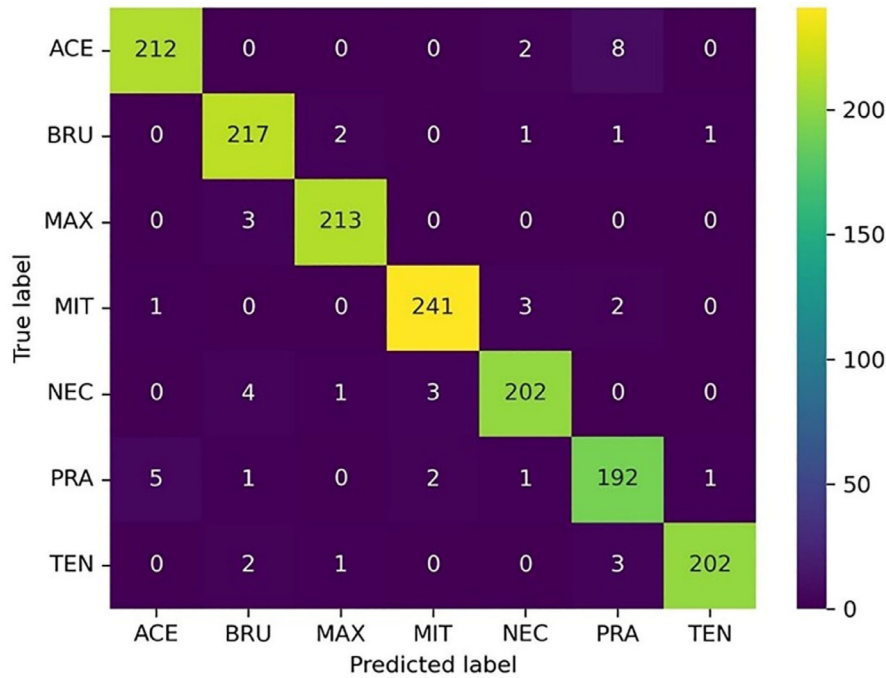


Figure 8. Confusion matrix of the optimized Residual-Transformer-Fine-Grained (ResTFG) model (Ours).

## CONCLUSIONS

To achieve the automatic fine-grained classification of chicken *Eimeria* species, a novel deep-learning model, named ResTFG, which integrates the advantages of the CNN and Transformer structure, was proposed in this study. The CNN structure containing residual blocks was used as the backbone, which has a powerful feature extraction ability, and compensated for the defect of CNN lacking a global receptive field through the deployment of the multihead attention mechanism in Transformer. The ablation experiments proved the synergistic effect of integrating the CNN and Transformer structure. Overall, the proposed ResTFG model performs well, achieving an accuracy of 96.9%, an inference speed of 256 FPS, and a memory consumption of 1.95M, which has the advantages of both high accuracy and low cost. This model can improve the work efficiency of researchers. More importantly, for people who do not have the ability to identify *Eimeria* species with the naked eye, they can obtain species distribution information to infer the severity of the disease with the help of this automatic identification system, which can provide guidance for subsequent medication and the basis for effective control measures. In future work, the ResTFG model would be further optimized and applied to other computer vision and pattern recognition tasks in agricultural engineering.

## ACKNOWLEDGMENTS

This work was supported by the Key R&D Program of Zhejiang Province (2021C02026) and the Agriculture Research System of China (CARS-40).

## DISCLOSURES

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.

## REFERENCES

- Abade, A., L. F. Porto, P. A. Ferreira, and F. D. Vidal. 2022. NemaNet: a convolutional neural network model for identification of soybean nematodes. *Biosyst. Eng.* 213:39–62.
- Abdalla, M. A. E., and H. Seker. 2017. Recognition of protozoan parasites from microscopic images: *Eimeria* species in chickens and rabbits as a case study. 2017 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC).
- Abnar, S., and W. Zuidema. 2020. Quantifying attention flow in transformers. *Proc. 58th Annu. Meeting Assoc. Comput. Linguist. (ACL) Online*.
- Adams, D. S., R. R. Kulkarni, J. P. Mohammed, and R. Crespo. 2022. A flow cytometric method for enumeration and speciation of coccidia affecting broiler chickens. *Vet. Parasitol.* 301:109634.
- Blake, D. P., J. Knox, B. Dehaeck, B. Huntington, T. Rathinam, V. Ravipati, S. Ayoade, W. Gilbert, A. O. Adebambo, I. D. Jatau, M. Raman, D. Parker, J. Rushton, and F. M. Tomley. 2020. Re-calculating the cost of coccidiosis in chickens. *Vet. Res.* 51:115.
- Butploy, N., W. Kanarkard, and P. M. Intapan. 2021. Deep learning approach for *Ascaris lumbricoides* parasite egg classification. *J. Parasitol. Res.* 2021:6648038.
- Cao, Y., and D. Li. 2013. Impact of increased demand for animal protein products in Asian countries: implications on global food security. *Anim. Front.* 3:48–55.
- Carion, N., F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. 2020. End-to-end object detection with transformers. 2020 Eur. Conf. Comput. Vision (ECCV). Scottish Event Campus.
- Castañón, C. A. B., J. S. Fraga, S. Fernandez, A. Gruber, and L. da F. Costa. 2007. Biological shape characterization for automatic image recognition and diagnosis of protozoan parasites of the genus *Eimeria*. *Pattern Recognit.* 40:1899–1910.

- Chapman, H. D., J. R. Barta, D. Blake, A. Gruber, M. Jenkins, N. C. Smith, X. Suo, and F. M. Tomley. 2013. A selective review of advances in coccidiosis research. *Adv. Parasit.* 83:93–171.
- Chen, H. T., Y. H. Wang, T. Y. Guo, C. Xu, Y. P. Deng, Z. H. Liu, S. W. Ma, C. J. Xu, C. Xu, and W. Gao. 2021. Pre-trained image processing transformer. 2021 IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR).
- Dai, Y., Y. F. Gao, and F. Y. Liu. 2021. TransMed: transformers advance multi-modal medical image classification. *Diagnostics* 11:1384.
- Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Housby. 2021. An image is worth  $16 \times 16$  words: transformers for image recognition at scale. 2021 Int. Conf. Learn. Represent. (ICLR).
- Engidaw, A., and G. Getachew. 2018. A review on poultry coccidiosis. *Abbyss. J. Sci. Technol.* 3:1–12.
- Esteva, A., B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542:115–118.
- Fatoba, A. J., and M. A. Adeleke. 2018. Diagnosis and control of chicken coccidiosis: a recent update. *J. Parasit. Dis.* 42:483–493.
- Fotouhi-Ardakani, R., S. M. Ghafari, P. D. Ready, and P. Parvizi. 2021. Developing, modifying, and validating a TaqMan real-time PCR technique for accurate identification of *Leishmania* parasites causing most leishmaniasis in Iran. *Front. Cell. Infect. Microbiol.* 11:731595.
- He, J., J. N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. T. Bai, and C. H. Wang. 2021. TransFG: a transformer architecture for fine-grained recognition. arXiv:2103.07976, doi:10.48550/arXiv.2103.07976. Accessed August 2022.
- He, K. M., X. Y. Zhang, S. Q. Ren, and J. Sun. 2016. Deep residual learning for image recognition. 2016 IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR).
- Hendershot, A. L., E. Esayas, A. C. Sutcliffe, S. R. Irish, E. Gadisa, F. G. Tadesse, and N. F. Lobo. 2021. A comparison of PCR and ELISA methods to detect different stages of *Plasmodium vivax* in *Anopheles arabiensis*. *Parasite Vector* 14:473.
- Howard, A., M. Sandler, G. Chu, L. C. Chen, B. Chen, M. X. Tan, W. J. Wang, Y. K. Zhu, R. M. Pang, V. Vasudevan, Q. V. Le, and H. Adam. 2019. Searching for MobileNetV3. 2019 IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR).
- Huang, G., Z. Liu, L. V. D. Maaten, and K. Q. Weinberger. 2016. Densely connected convolutional networks. 2017 IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR).
- Huang, Y. Y., X. C. Ruan, L. Li, and M. H. Zeng. 2017. Prevalence of *Eimeria* species in domestic chickens in Anhui province, China. *J. Parasit. Dis.* 41:1014–1019.
- Jacob, D., M. W. Chang, L. Kenton, and T. Kristina. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. Proc. 2019 Conf. North Am. Chapt. Assoc. Comput. Linguist.: Hum. Lang. Technol..
- Kucera, J., and M. Reznicky. 1991. Differentiation of species of *Eimeria* from the fowl using a computerized image-analysis system. *Folia Parasit.* 38:107–113.
- Lee, C. C., P. J. Huang, Y. M. Yeh, P. H. Li, C. H. Chiu, W. H. Cheng, and P. Tang. 2021. Helminth egg analysis platform (HEAP): an opened platform for microscopic helminth egg identification and quantification based on the integration of deep learning architectures. *J. Microbiol. Immunol.* 55:395–404.
- Lu, X. Y., R. Yang, J. Zhou, J. Jiao, F. Liu, Y. F. Liu, B. F. Su, and P. W. Gu. 2022. A hybrid model of ghost-convolution enlightened transformer for effective diagnosis of grape leaf disease and pest. *J. King Saud Univ.-Com.* 34:1755–1767.
- Ma, N. N., X. Y. Zhang, H. T. Zheng, and J. Sun. 2018. ShuffleNet V2: practical guidelines for efficient CNN architecture design. 2018 Eur. Conf. Comput. Vision (ECCV).
- Mattiello, R., J. D. Boviez, and L. R. McDougald. 2000. *Eimeria brunetti* and *Eimeria necatrix* in chickens of Argentina and confirmation of seven species of *Eimeria*. *Avian Dis.* 44:711–714.
- Mesa, C., L. M. Gomez-Osorio, S. Lopez-Osorio, S. M. Williams, and J. J. Chaparro-Gutierrez. 2021. Survey of coccidia on commercial broiler farms in Colombia: frequency of *Eimeria* species, anticoccidial sensitivity, and histopathology. *Poult. Sci.* 100:101239.
- Monge, D. F., and C. A. Beltrán. 2019. Classification of *Eimeria* species from digital micrographies using CNNs. 2019 10th Int. Conf. Pattern Recognit. Syst. (ICPRS).
- Mottet, A., and G. Tempio. 2017. Global poultry production: current state and future outlook and challenges. *Worlds Poult. Sci. J.* 73:245–256.
- Shirley, M. W. 1997. *Eimeria* spp. from the chicken: occurrence, identification and genetics. *Acta Vet. Hung.* 45:331–347.
- Simonyan, K., and A. Zisserman. 2024. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, doi:10.48550/arXiv.1409.1556. Accessed August 2022.
- Thevenoux, R., V. L. Le, H. Villesseche, A. Buisson, M. Beurton-Aimar, E. Grenier, L. Folcher, and N. Parisey. 2021. Image based species identification of *Globodera* quarantine nematodes using computer vision and deep learning. *Comput. Electron. Agr.* 186:106058.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. 2017 21st Conf. Neural Inform. Process. Syst. (NIPS).
- Wang, J. T., M. X. Shen, L. S. Liu, Y. Xu, and C. Okinda. 2019. Recognition and classification of broiler droppings based on deep convolutional neural network. *J. Sensors* 2019:3823515.
- Yang, F., M. Poostchi, H. Yu, Z. Zhou, K. Silamut, J. Yu, R. J. Maude, S. Jaeger, and S. Antani. 2020. Deep learning for smartphone-based malaria parasite detection in thick blood smears. *IEEE J. Biomed. Health* 24:1427–1438.
- Ye, C. W., Z. W. Yu, R. Kang, K. Yousafa, C. Qi, K. J. Chen, and Y. P. Huang. 2020. An experimental study of stunned state detection for broiler chickens using an improved convolution neural network algorithm. *Comput. Electron. Agric.* 170:105284.
- Zhao, J. J., Y. X. Peng, and X. T. He. 2020. Attribute hierarchy based multi-task learning for fine-grained image classification. *Neurocomputing* 395:150–159.
- Zheng, S. X., J. C. Lu, H. S. Zhao, X. T. Zhu, Z. K. Luo, Y. B. Wang, Y. W. Fu, J. F. Feng, T. Xiang, P. H. S. Torr, and L. Zhang. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. 2021 IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR).