ASM Sc. J., 20(1), 2025
https://doi.org/10.32802/asmscj.2025.2002

Optimisation Solutions and Simple Innovative Solution Research on ResNet50 Model

C.H. Zhao[1] and W.Y. Leong[2]*

[1]Inti International University, Persiaran Perdana BBN Putra Nilai, 71800 Nilai, Negeri Sembilan.

This review explores optimisation strategies and innovative modifications to the ResNet50 model, a widely used deep learning architecture in computer vision. ResNet50, with its hallmark skip connections, addresses vanishing gradient issues in deep networks, enabling efficient feature extraction. However, the model's performance can be enhanced through various optimisation solutions. This paper systematically reviews existing approaches, including pruning, quantisation, hyperparameter tuning, and advanced training techniques such as knowledge distillation and transfer learning. Additionally, the integration of simple innovative solutions—such as lightweight architectural modifications, enhanced data augmentation techniques, and fusion with attention mechanisms—is examined for their ability to improve accuracy and computational efficiency. The review synthesises findings from recent studies to provide actionable insights for researchers and practitioners. Emphasis is placed on balancing model performance with resource constraints, ensuring applicability in real-world scenarios. The outcomes aim to guide further research in ResNet50 optimisation, fostering advancements in efficient and scalable deep learning solutions.

Keywords: residual neural networks; model optimisation; product innovation

I. INTRODUCTION

ResNet50 is a classic backbone network in deep learning. With its residual structure, it has demonstrated excellent performance in key visual tasks such as image classification, object detection, and image segmentation (Shafiq, M *et al.*, 2022).

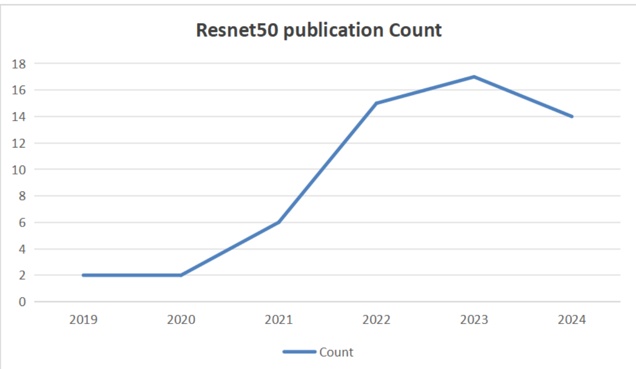

Figure 1. Research trend of ResNet50 diagram

As a milestone in the field of deep learning, ResNet50's optimisation has become the focus of research. It can be seen from the publication volume of web of science literature that the research trend of ResNet50 is shown in the Figure 1.

The number of related research papers has increased significantly, especially in the fields of image classification and target detection (Leong, 2024a). This trend reflects the rapid development of deep learning technology and the extensive application of ResNet50 in computer vision (Leong, 2024b).

Through the study of existing published literature, as shown in the following table, ResNet50, as a classic deep learning model, performs well in many visual tasks but still has many shortcomings in some specific applications, such as gradient vanishing, long-distance dependency processing, overfitting, and low computational efficiency. In response to these shortcomings, scholars have proposed a variety of improvement schemes to improve its performance in specific tasks (Leong, 2024c).

*Corresponding author's e-mail: waiyie.leong@newinti.edu.my

Table 1. Limitations of the ResNet50

Authors	**Paper Title**	**ResNet50 Limitation(s)**
H Jabnouni *et al.*	ResNet-50 based fire and smoke images classification	Poor classification performance for low-resolution images
M Tripathi *et al.*	Analysis of convolutional neural network-based image classification techniques	For tasks with less data, overfitting is prone to occur
K He *et al.*	Deep residual learning for image recognition	As the depth increases, the training time and computational cost increase significantly
Kimberley Whitehead *et al.*	Developmental trajectory of movement-related cortical oscillations during active sleep in a cross-sectional cohort of pre-term and full-term human infants	Although it reduces the training error, it increases the model complexity
Kaiming He *et al.*	Identity Mappings in Deep Residual Networks	Complex structures perform poorly on some tasks
ZHOU Tao *et al.*	ResNet and Its Application to Medical Image Processing: Research Progress and Challenges	Large calculation and memory consumption.
FEI Wenqian *et al.*	Diamond particle clarity detection method based on CBAM-ResNet50	It is easy to lose information in detail detection.
LI Qing *et al.*	A survey of person re-identification based on deep learning	There are problems with feature extraction and speed.

With the growing market demand for applications and diverse scenarios, optimising the ResNet50 model has become a research focus in academia and industry. The purpose of this article is to comprehensively review and analyse the current ResNet50 optimisation techniques, hoping to provide guidance and reference for the diverse applications of the model. By exploring different optimisation strategies, this article will discuss how to balance the accuracy and cost of the model, and how to adjust the model structure according to different application requirements, to achieve optimal model performance under multiple requirements.

II. MATERIALS AND METHOD

Deep learning models such as attention mechanism, residual connection, and multi-scale feature fusion have made relatively deep progress. According to the data of papers published in the web of science database, researchers are working to improve the performance and generalisation ability of models through these technologies.

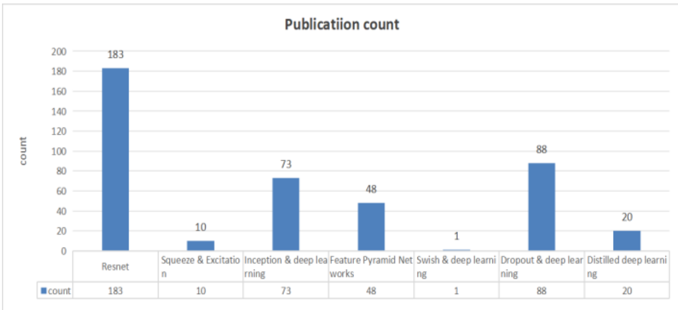

Figure 2. Data of papers published diagram

These optimised models demonstrate the application potential and performance improvement space of ResNet50 in different fields (Zhou, Y *et al.*, 2024). By combining advanced technologies and strategies, ResNet50 and its derivative models have achieved remarkable results in tasks such as image classification and object detection.

A. Attention Mechanism Integration

SE-ResNet50 is a deep learning model that combines the residual network (ResNet) and the Squeeze-and-Excitation (SE) module. SE-ResNet50 is based on the ResNet50 architecture and enhances the model's feature expression ability by introducing the SE module (Gu, X *et al.*, 2024). The SE module is a lightweight channel attention mechanism module. It learns the relationship between channels and adaptively adjusts the weight of each channel to improve the performance of the model. The input image of SE-ResNet50 first passes through a stem module, which includes a 7x7 convolutional layer and three 3x3 convolutional layers for feature extraction. The model contains four residual blocks (layer 1 to layer 4), and each block consists of multiple residual units. These units solve the vanishing gradient problem in deep network training by introducing residual connections. In each residual block, the SE module is embedded into the residual unit. The SE module consists of two main parts: squeeze and excitation. The squeeze operation compresses spatial information through global average pooling, while the excitation operation learns the importance of the channel through two fully connected layers and applies it to the original feature map (Jin, X *et al.*, 2022). Finally, SE-ResNet50 outputs a feature map with 2048 channels, which can be used for image classification or other visual tasks.

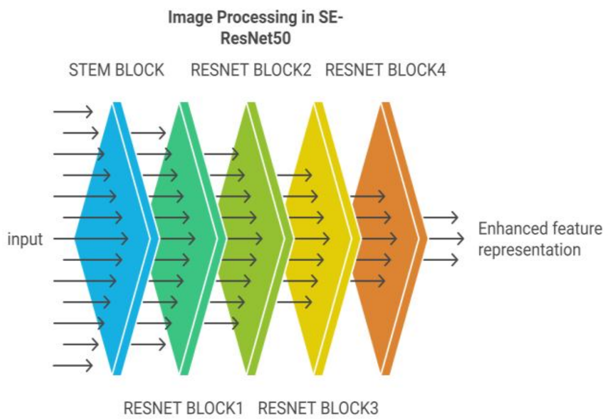

Figure 3. SE-ResNet50 structure diagram

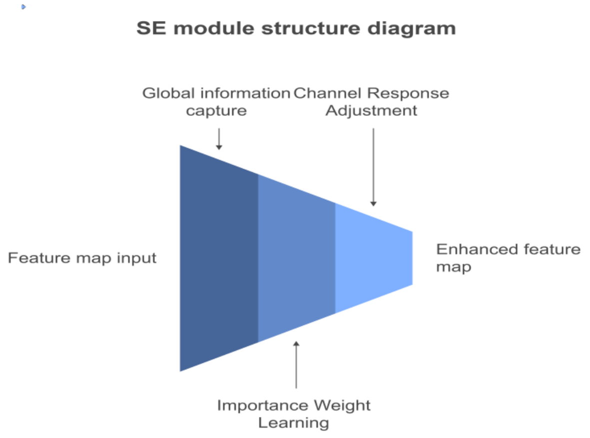

Figure 4. SE module structure diagram

SE-ResNet50 is widely used in a variety of computer vision tasks due to its powerful feature extraction capabilities and adaptive adjustment of channel relationships. SE-ResNet50 effectively improves the performance and generalisation capabilities of the model by combining the deep residual structure of ResNet and the channel attention mechanism of the SE module, enabling it to achieve good results in a variety of visual tasks.

B. Residual Block Design Optimisation

ResNeXt is a deep learning model that combines the ideas of ResNet and Inception and improves the expressive ability of the model by introducing grouped convolution and multi-branch structure (Dhillon, A *et al.*, 2020). The infrastructure of ResNeXt is inspired by ResNet's Bottleneck module. It divides the residual part into multiple branches; each branch contains a 3x3 convolutional layer. The number of branches is defined as "cardinality". The main part of ResNeXt50 consists of four stages, each stage contains a series of ResNeXt modules. These stages are usually referred to as layer 1, layer 2, layer 3, and layer 4. Each module contains three sub-layers, where the middle layer uses 3x3 grouped convolutions, while the two sides use 1x1 convolutions.

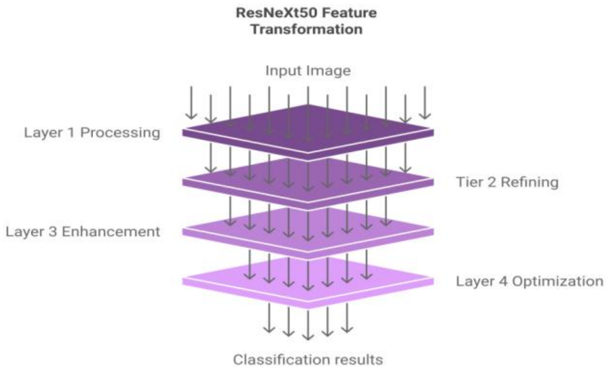

Figure 5. ResNeXt module structure diagram

ResNeXt uses novel architectural designs to improve model performance while maintaining computational efficiency. It has demonstrated strong competitiveness and broad application prospects in multiple vision tasks.

C. Multi-scale Feature Fusion

FPN-ResNet50 is a deep learning model that combines Feature Pyramid Networks (FPN) and residual networks (Abdelrahman, A. *et al.*, 2023). FPN-ResNet50 uses ResNet50 as its backbone network, responsible for extracting deep features of the image. FPN fuses information at different scales by building a feature pyramid. The FPN network structure mainly consists of two parts, the underlying feature extraction network and the top-level feature regression network. The bottom feature extraction network extracts feature maps of different sizes through multiple convolutional layers, which are up sampled and merged into the top feature pyramid in subsequent processing. FPN contains a top-down path, a bottom-up path, and lateral connections. The top-down path fuses high-level feature maps with low-level feature maps through up sampling, while the bottom-up path delivers information from the low-level feature maps. The output layer of FPN-ResNet50 usually includes an adaptive average pooling layer and a fully connected layer to convert the fused feature maps into the final classification or target detection results.

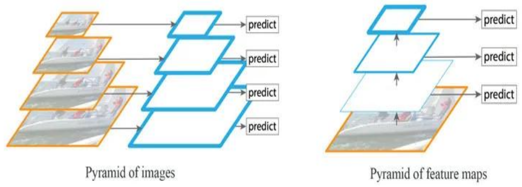

Figure 6. FPN-ResNet50 module structure diagram (Lin, T. Y *et al.*, 2017)

The FPN-ResNet50 model combines the deep feature extraction capabilities of ResNet50 and the multi-scale information fusion capabilities of FPN, enabling it to achieve excellent performance in computer vision tasks such as target detection and image segmentation. This structure is relatively simple and clear, easy to understand and implement, and has become one of the preferred network structures for many deep learning researchers and practitioners.

D. Activation Function Optimisation

Swish-ResNet50 is a deep learning model that replaces the traditional ReLU activation function with the Swish activation function to improve the performance of the model. The Swish activation function is a new type of activation function proposed by the Google Brain team (Chieng, HA *et al.*, 2018). Swish is a smooth continuous function. This smoothness makes Swish have derivatives throughout the domain, which is conducive to optimisation. Swish allows a small number of negative weights to be propagated, while ReLU thresholds all negative weights to zero. This is crucial for the use of non-monotonic smooth activation functions in increasingly deeper neural networks. Experiments show that on some challenging datasets, models using Swish as the activation function perform better than models using ReLU.

E. Introducing Regularisation Techniques

Dropout-ResNet50 is a deep learning model that introduces Dropout regularisation technology into the traditional ResNet50 architecture. Dropout is a commonly used regularization method to prevent overfitting of neural networks (Salehin, I *et al.*, 2023). By randomly "dropping" a portion of neurons during the training process (i.e., setting their output to zero), Dropout can reduce the complex coadaptation relationships between neurons and enhance the generalisation ability of the model. In the Dropout-ResNet50 model, a Dropout layer is usually added before the fully connected layer of the network, or after the output of each residual block. Doing so can introduce randomness into the feature extraction process, reduce the risk of overfitting, and improve the robustness of the model.

F. Optimising Network Width and Depth

The architecture of Wide-ResNet50 is based on ResNet50, but its number of channels per residual block is twice that of ResNet50. This design idea is to improve the performance and expressiveness of the model by increasing the width of the network (that is, the number of feature channels) rather than increasing the depth of the network (Khan, A *et al.*, 2020). The basic blocks used by Wide-ResNet50 are residual blocks with identity mapping, which contain two 3x3 convolutional layers with batch normalisation and ReLU activation functions in between. By increasing the number of channels per block, Wide-ResNet50 can capture richer feature information. Wide-ResNet50 is widely used in computer vision tasks such as image classification, target detection, and image segmentation due to its excellent performance and efficiency.

Deep-ResNet50 usually refers to the depth-extended version of ResNet50, which improves the performance of the model by increasing the depth of the network. In PyTorch, Deep-ResNet50 can be achieved by inheriting the ResNet model and adjusting the number of layers. For example, you could use torchvision.models.Resnet50 as the base model and add or modify layers as needed. Deep-ResNet50 improves the performance of the model by increasing the depth of the network while maintaining the efficiency and accuracy of the ResNet series models (Arslan, M. Mubeen, *et al.*, 2024). This depth-expanded model can capture richer feature information when processing complex visual tasks, thereby improving the model's generalisation ability and accuracy.

G. Introducing Knowledge Distillation

Distilled-ResNet50 is a ResNet50 model optimised by knowledge distillation. Knowledge Distillation is a model compression technology that allows a small student model to learn the knowledge of a large teacher model to improve the performance of the student model (Gou, J *et al.*, 2021). Distilled-ResNet50 retains the basic architecture of ResNet50, but introduces an additional loss function, distillation loss, during training. This loss function combines the cross-entropy loss of the true label and the KL divergence loss of the teacher model output to encourage the output of the student model to be close to the output of the teacher model. Distilled-ResNet50 uses distillation technology to improve performance without increasing model complexity. This approach is particularly suitable for scenarios where the model needs to be deployed on resource-constrained devices, as it can improve model accuracy while keeping the model size constant.

The above-mentioned optimisation models show the application potential and performance improvement space of ResNet50 in different fields. By combining advanced technologies and strategies, ResNet50 and its derivative models have achieved remarkable results in tasks such as image classification and target detection.

H. ResNet50 Innovative Optimisation Model

Based on the above theory and the ResNet50 classic network, this paper innovates the models.

1. Hierarchical Feature Fusion Squeeze-and-Excitation Net

This paper adopts the attention mechanism and hierarchical feature fusion mechanism in the above optimisation direction to implement a deep learning model named Hierarchical Feature Fusion Squeeze-and-Excitation Net, abbreviated as HFF-SEnet. The following is a detailed description of the model structure.

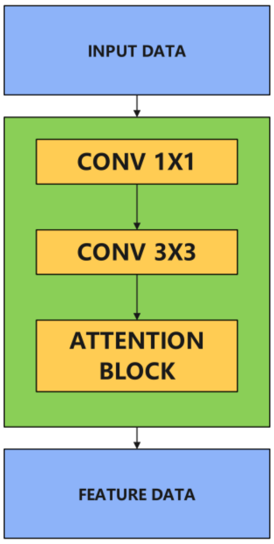

Figure 7. HFF-SEnet convolutional layer module structure diagram

Define the convolutional layer, conv1: 1x1 convolution, which is used to reduce the number of input channels. conv2: 3x3 convolution, used to extract features. The attention module is a sequence of convolution and activation functions, used to calculate the channel attention weight.

Define the ResNet structure, convolution batch normalisation, activation function, and maximum pooling.

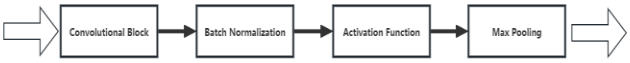

Figure 8. ResNet structure of HFF-SEnet diagram

Defining the forward propagation method, the input X first passes through multiple layers of ResNet (convolution, batch normalisation, activation function, and max pooling). After each layer of ResNet, feature fusion is performed through the corresponding HFF module. Finally, after an average pooling and flattening operation, the final result is output through the fully connected layer.

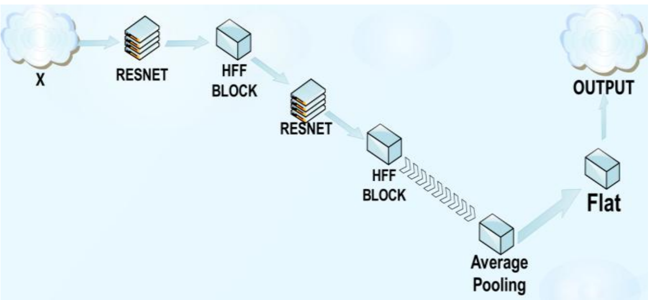

Figure 9. Forward propagation method of HFF-SEnet diagram

This new optimisation model improves model performance by fusing features at different levels of RESNET. Hierarchical feature fusion can help the model better capture feature information at different levels, thereby improving the generalisation ability and robustness of the model.

The model code is as follows:

```python
import torch
import torch.nn as nn
import torch.nn.functional as F
class HierarchicalFeatureFusion(nn.Module):
    def __init__(self, in_channels, out_channels):
        super(HierarchicalFeatureFusion, self).__init__()
        self.conv1 = nn.Conv2d(in_channels, out_channels, kernel_size=1)
        self.conv2 = nn.Conv2d(out_channels, out_channels, kernel_size=3, padding=1)
        self.attention = nn.Sequential(
            nn.Conv2d(out_channels, out_channels // 4, kernel_size=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(out_channels // 4, out_channels, kernel_size=1),
            nn.Sigmoid()
        )
    def forward(self, x):
        x = self.conv1(x)
        x = self.conv2(x)
        attention = self.attention(x)
        x = x * attention
        return x
class ResNet50HFF(nn.Module):
    def __init__(self, num_classes=1000):
        super(ResNet50HFF, self).__init__()
        self.resnet = torchvision.models.resnet50(pretrained=True)
        self.hff1 = HierarchicalFeatureFusion(256, 256)
        self.hff2 = HierarchicalFeatureFusion(512, 512)
        self.hff3 = HierarchicalFeatureFusion(1024, 1024)
        self.hff4 = HierarchicalFeatureFusion(2048, 2048)
        self.fc = nn.Linear(2048, num_classes)

    def forward(self, x):
        x = self.resnet.conv1(x)
        x = self.resnet.bn1(x)
        x = self.resnet.relu(x)
        x = self.resnet.maxpool(x)
        x = self.resnet.layer1(x)
        x = self.hff1(x)
        x = self.resnet.layer2(x)
        x = self.hff2(x)
        x = self.resnet.layer3(x)
        x = self.hff3(x)
        x = self.resnet.layer4(x)
        x = self.hff4(x)
        x = self.resnet.avgpool(x)
        x = torch.flatten(x, 1)
        x = self.fc(x)
        return x
```

Figure 10. Code of HFF-SEnet model diagram

2. *Multi-Modal Feature Fusion Resnet*

The classic Resnet50 model is weaker than the transformer architecture in multi-modal data fusion, but if it is a small number of modal fusions, it can be achieved by constructing a multi-modal function when designing the convolution block. Based on the concept of multi-modal data fusion, this paper designs a multi-modal feature extraction module by defining multi-modal functions, extracts feature of different modalities respectively, and designs a feature fusion module to fuse the features of different modalities into Resnet. , and named Multi-Modal Feature Fusion Resnet (M-MFFnet). The following describes the model details.

Define the convolution block and initialise the two convolution layers conv1 and conv2, which are used to process the two input modal data (in_channels1 and in_channels2) respectively. Use the fusion convolution layer to further process the result of splicing the two feature maps. The forward propagation method is to input X1 and

X2 in two channels and perform the convolution operation through conv1 and conv2 respectively. The two processed feature maps are spliced along the channel dimension. The fused feature map is then obtained through the fusion layer.

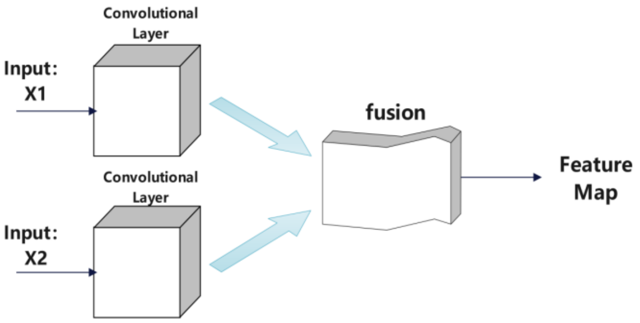

Figure 11. M-MFFnet convolution block diagram

The pre-trained ResNet50 model is introduced. X1 and X2 pass through each layer of ResNet50 respectively, extract features, use the mmff module to fuse the extracted features, and finally obtain the final output through global average pooling and fully connected layers.The model code is as follows:

```
import torch
import torch.nn as nn
import torchvision.models as models
class MultiModalFeatureFusion(nn.Module):
    def __init__(self, in_channels1, in_channels2):
        super(MultiModalFeatureFusion, self).__init__()
        self.conv1 = nn.Conv2d(in_channels1, 256, kernel_size=1)
        self.conv2 = nn.Conv2d(in_channels2, 256, kernel_size=1)
        self.fusion = nn.Conv2d(512, 256, kernel_size=1)
    def forward(self, x1, x2):
        x1 = self.conv1(x1)
        x2 = self.conv2(x2)
        x = torch.cat([x1, x2], dim=1)
        x = self.fusion(x)
        return x
class ResNet50MMFF(nn.Module):
    def __init__(self, num_classes=1000):
        super(ResNet50MMFF, self).__init__()
        self.resnet = models.resnet50(pretrained=True)
        self.mmff = MultiModalFeatureFusion(2048, 2048)
        self.fc = nn.Linear(256, num_classes)
    def forward(self, x1, x2):
        x1 = self.resnet.conv1(x1)
        x1 = self.resnet.bn1(x1)
        x1 = self.resnet.relu(x1)
        x1 = self.resnet.maxpool(x1)
        x1 = self.resnet.layer1(x1)
        x1 = self.resnet.layer2(x1)
        x1 = self.resnet.layer3(x1)
        x1 = self.resnet.layer4(x1)
        x2 = self.resnet.conv1(x2)
        x2 = self.resnet.bn1(x2)
        x2 = self.resnet.relu(x2)
        x2 = self.resnet.maxpool(x2)
        x2 = self.resnet.layer1(x2)
        x2 = self.resnet.layer2(x2)
        x2 = self.resnet.layer3(x2)
        x2 = self.resnet.layer4(x2)
        x = self.mmff(x1, x2)
        x = self.resnet.avgpool(x)
        x = torch.flatten(x, 1)
        x = self.fc(x)
        return x
```

Figure 12. Code of M-MFFnet model diagram

The use of Multi-Modal Feature Fusion Resnet in the multimodal direction makes up for the shortcomings of Resnet50 in multimodal applications to a certain extent. A model with small parameters such as Resnet50 can be used to solve the use problems of simple multimodal models.

III. RESULT AND DISCUSSION

This paper reviews various optimisation techniques for the ResNet50 model and reveals their advantages and limitations in specific application scenarios. The results show that no single optimisation scheme is suitable for all situations, and each method provides unique improvements based on its specific needs. Therefore, the choice of optimisation strategy should be based on the specific requirements and constraints of the application. Future research should explore the synergistic effects of different optimisation techniques to meet the growing and diverse needs in practical applications (Leong, 2009). By comprehensively applying multiple optimisation methods, it is expected to achieve the best balance between performance and efficiency of the ResNet50 model and promote the further development of deep learning model optimisation.

IV. CONCLUSION

This article reviews the optimisation techniques of the ResNet50 model and shows that each optimisation method has its applicable scenarios and unique advantages. According to the analysis of national and regional literature publications in the Web of Science data, the United States and China are in the leading position in ResNet50 research.

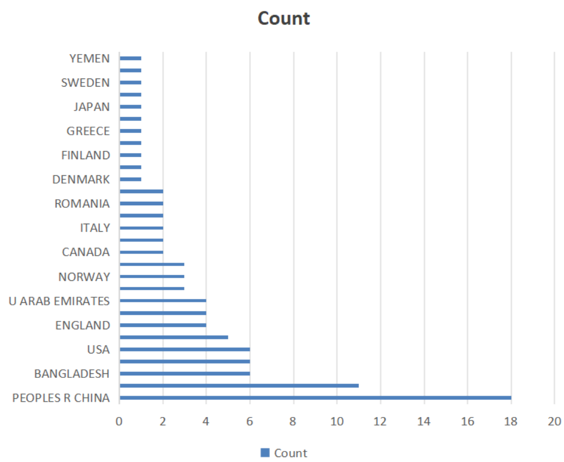

Figure 13. National and regional literature publications data diagram

Through a review of the literature, it can be seen that the research on ResNet50 and its optimisation model is continuously deepening, and more breakthroughs are expected to be made in the next few years. Future research can combine a variety of optimisation strategies to improve model performance and meet diverse practical needs.

V. REFERENCES

Abdelrahman, A & Viriri, S 2023, 'FPN-SE-ResNet model for accurate diagnosis of kidney tumors using CT images', Applied Sciences, vol. 13, no. 17, p. 9802.

Arslan, M, Mubeen, M, Akram, A, Abbasi, SF, Ali, MS & Tariq, MU 2024, 'A Deep Features Based Approach Using Modified ResNet50 and Gradient Boosting for Visual Sentiments Classification', in 2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, pp. 239-242.

Chieng, HH, Wahid, N, Ong, P & Perla, SRK 2018, 'Flatten-T Swish: a thresholded ReLU-Swish-like activation function for deep learning', arXiv preprint arXiv:1812.06247.

Dhillon, A & Verma, GK 2020, 'Convolutional neural network: a review of models, methodologies and applications to object detection', Progress in Artificial Intelligence, vol. 9, no. 2, pp. 85-112.

Fei, W, Zhao, F, Du, Q & Wang, Q 2024, 'Diamond particle clarity detection method based on CBAM-ResNet50', Diamond & Abrasives Engineering, vol. 44, no. 5, pp. 588-598. Doi: 10.13394/j.cnki.jgszz.2023.0153

Gou, J, Yu, B, Maybank, SJ & Tao, D 2021, 'Knowledge distillation: A survey', International Journal of Computer Vision, vol. 129, no. 6, pp. 1789-1819.

Gu, X, Tian, Y, Li, C, Wei, Y & Li, D 2024, 'Improved SE-ResNet Acoustic–Vibration Fusion for Rolling Bearing Composite Fault Diagnosis', Applied Sciences, vol. 14, no. 5, p. 2182.

He, K, Zhang, X, Ren, S & Sun, J 2016, 'Deep residual learning for image recognition', in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778.

Jabnouni, H, Arfaoui, I, Cherni, MA, Bouchouicha, M & Sayadi, M 2022, 'ResNet-50 based fire and smoke images classification', in 2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), IEEE, pp. 1-6.

Jin, X, Xie, Y, Wei, XS, Zhao, BR, Chen, ZM & Tan, X 2022, 'Delving deep into spatial pooling for squeeze-and-excitation networks', Pattern Recognition, vol. 121, p. 108159.

Khan, A, Sohail, A, Zahoora, U & Qureshi, AS 2020, 'A survey of the recent architectures of deep convolutional neural networks', Artificial Intelligence Review, vol. 53, pp. 5455-5516.

Leong, WY, Leong, YZ & Leong, WS 2024a, 'Miniature THz Antenna Design', 2024 IEEE International Workshop on Electromagnetics: Applications and Student Innovation Competition (iWEM), Taoyuan County, Taiwan, pp. 1-2.

Leong, WY, Leong, YZ & Leong, WS 2024c, 'Nuclear Technology in Electronic Communications', 2024 IEEE 4th International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), Taipei, Taiwan, 2024, pp. 684-689.

Leong, WY & Liu, W 2009, 'Structural health monitoring: Subsurface defects detection', in 2009 35th Annual Conference of IEEE Industrial Electronics, IEEE, pp. 4326-4330.

Leong, WY 2024b, 'Industry 5.0: Design, standards, techniques and applications for manufacturing', Institution of Engineering and Technology.

Li, Q, Hu, W, Li, J, Liu, Y & Li, M 2022, 'A survey of person re-identification based on deep learning', Chinese Journal of Engineering, vol. 44, no. 5, pp. 920-932. Doi: 10.13374/j.issn2095-9389.2020.12.22.004

Lin, TY, Dollár, P, Girshick, R, He, K, Hariharan, B & Belongie, S 2017, 'Feature pyramid networks for object detection', in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117-2125.

Salehin, I & Kang, DK 2023, 'A review on dropout regularization approaches for deep neural networks within the scholarly domain', Electronics, vol. 12, no. 14, p. 3106.

Shafiq, M & Gu, Z 2022, 'Deep residual learning for image recognition: A survey', Applied Sciences, vol. 12, no. 18, p. 8972.

Tripathi, M 2021, 'Analysis of convolutional neural network-based image classification techniques', Journal of Innovative Image Processing (JIIP), vol. 3, no. 2, pp. 100-117.

Whitehead, K, Meek, J & Fabrizi, L 2018, 'Developmental trajectory of movement-related cortical oscillations during active sleep in a cross-sectional cohort of pre-term and full-term human infants', Scientific Reports, vol. 8, p. 17516. Doi: 10.1038/s41598-018-35850-1

Zhou, T, Liu, Y, Lu, H, Ye, X & Chang, X 2022, 'ResNet and its application to medical image processing: Research progress and challenges', Journal of Electronics & Information Technology, vol. 44, no. 1, pp. 149-167. Doi: 10.11999/JEIT210914

Zhou, Y, Wang, Z, Zheng, S, Zhou, L, Dai, L, Luo, H, ... & Sui, M 2024, 'Optimization of automated garbage recognition model based on resnet-50 and weakly supervised cnn for sustainable urban development', Alexandria Engineering Journal, vol. 108, pp. 415-427.