不同深度下 ResNet、SE-ResNet 和 Swin Transformer 在图像分类中的性能特性与注意力机制调控

孔信轲

摘要

本研究深入探讨了在不同深度下非注意力机制的 CNN（ResNet）、SE-ResNet 和 Swin Transformer 在图像分类任务中的性能特性。以 ResNet 为横向对照，对 SE-ResNet 和 Swin Transformer 模型通过不同网络深度的纵向对比，系统地分析了这些模型在不同网络深度下的特点。同时，通过消融实验揭示了在不同深度的 ResNet 中引入 SE 注意力模块的影响，以深入了解 SE 模块对 ResNet 性能的具体作用。本研究提出了一种在 SE-ResNet 和 Swin Transformer 模型的注意力 Key 值机制中进行 Dropout 想法，旨在提高模型的泛化能力。The code and models are publicly available at https://github.com/qsking8/Deep-Learning-Image-Classification-Models

关键词— SENet, ResNet, Swin Transformer, Computer Vision，Dropout

1. 引言

在计算机视觉领域，深度神经网络已经成为各种图像处理任务的主流模型，如图像分类、目标检测和语义分割。ResNet（Residual Network）等经典模型以其深度残差结构成功解决了深度神经网络训练中的梯度消失和梯度爆炸问题，为模型的进一步发展提供了基础。然而，随着模型深度的增加，训练和推理的计算开销也相应增加，同时可能引发过拟合和性能下降的问题。因此，深入研究深度对不同模型性能的影响，以及引入新颖的模块如 Squeeze-and-Excitation（SE）对深度模型性能的调控，对于优化深度神经网络的设计具有重要的意义。

最近的研究进展表明，Swin Transformer 以其基于注意力机制的结构在计算机视觉任务中取得了令人瞩目的成果。Swin Transformer 通过引入局部和全局的注意力机制，使模型能够更好地捕获图像中的长距离依赖关系，从而提升了图像特征的表达能力。然而，关于深度和性能之间的关系，以及在不同深度下模型的表现仍然缺乏全面的理解。

本研究的内容关注于三个主要方面：首先，我将深入研究在不同深度下非注意力机制的 CNN（ResNet）、SE-ResNet 和 Swin Transformer 的性能特性，以全面了解它们在图像分类任务中的行为，**以 ResNet 作为横向对照，不同深度作为纵向对比，**探究 SE-ResNet 和 Swin Transformer 的性能特性。

其次，通过消融实验，我将深入探究在不同深度的 ResNet 中引入 SE 注意力模块的影响，以揭示 SE 模块对 ResNet 性能的具体作用。

最后，针对过拟合问题，我们将在 SE-ResNet 和 Swin Transformer 模型的注意力机制 **Key 值**引入 Dropout 思想，以改善模型的泛化能力。

本研究的意义在于通过深入研究这些模型在不同深度下的性能表现，为深度神经网络的设计和应用提供更深刻的理解。希望通过实验结果揭示模型性能随深度变化的规律，从而为在实际应用中选择合适深度提供指导。消融实验将有助于理解 SE 模块在 ResNet 中的具体作用机制，为模型结构的优化提供实证支持。同时，通过在注意力模型 Key 值中引入 Dropout 思想，尝试提高模型的泛化能力，应对实际任务中复杂数据集的挑战。

最新研究进展表明，深度学习领域不断涌现各种新型架构和技术，而注意力机制作为其中的一种关键技术，在提升模型性能和泛化能力方面发挥着重要作用。然而，对于不同深度下模型的特性、SE 模块在 ResNet 中的作用机制以及注意力模型的过拟合问题等方面的研究仍然相对有限。因此，本研究旨在填补这一知识空白，为深度神经网络的进一步发展提供新的见解和指导。

我将对本技术报告的主要贡献总结如下：

1) 在不同的网络深度下，探究非注意力机制的 CNN（ResNet）、SE-ResNet 和 swin_transformer 的特性

2) 对不同深度的 ResNet 设计 SE 注意力模块的消融实验

3) 对于该数据集的过拟合问题，在注意力机制 key 值上使用 DropOut 的思想，提出自己对解决 SE-ResNet、swin_transformer 注意力模型过拟合问题进行思考并提出改进想法。

2. 研究方法

本研究的首要目标是全面了解在不同深度下非注意力机制的 CNN（ResNet）、SE-ResNet 和 Swin Transformer 在图像分类任务中的性能特性。为此，我们采用了横向对照和纵向对比的研究框架。横向对照通过将 ResNet 作为基准，对比 SE-ResNet 和 Swin Transformer，纵向对比则通过选择不同深度的模型，深入探究它们在各自体系结构下的性能变化。

在**网络架构**选择上，ResNet 作为经典的非注意力机制的 CNN，我们选用 ResNet 作为横向对照的基准。通过选择不同深度的 ResNet（18 层、34 层、50 层），我们能够揭示深度对传统 CNN 性能的影响。

表 1 ResNet 架构

layer name	Output size	18-layer	34-layer	50-layer
cony1	112x112	7x7, 64, stride 2		
conv2_x	56x56	3 x3 max pool, stride 2		
		$\begin{bmatrix}3x3, & 64\\ 3x3, & 64\end{bmatrix}$x2	$\begin{bmatrix}3x3, & 64\\ 3x3, & 64\end{bmatrix}$x2	$\begin{bmatrix}1x1 & 64\\ 3x3 & 64\\ 1x1 & 256\end{bmatrix}x3$
cony3_x	28x28	$\begin{bmatrix}3x3, & 128\\ 3x3, & 128\end{bmatrix}$x2	$\begin{bmatrix}3x3, & 128\\ 3x3, & 128\end{bmatrix}$x4	$\begin{bmatrix}1x1 & 128\\ 3x3 & 128\\ 1x1 & 512\end{bmatrix}x4$
cony4_x	14x14	$\begin{bmatrix}3x3, & 256\\ 3x3, & 256\end{bmatrix}$x2	$\begin{bmatrix}3x3, & 256\\ 3x3, & 256\end{bmatrix}$x6	$\begin{bmatrix}1x1 & 256\\ 3x3 & 256\\ 1x1 & 1024\end{bmatrix}x6$
cony5_x	7x7	$\begin{bmatrix}3x3, & 512\\ 3x3, & 512\end{bmatrix}$x2	$\begin{bmatrix}3x3, & 512\\ 3x3, & 512\end{bmatrix}$x3	$\begin{bmatrix}1x1 & 512\\ 3x3 & 512\\ 1x1 & 2048\end{bmatrix}x3$
	1x1	average pool, 1000-d fc.softmax		

选择引入 SE 模块的 ResNet，旨在观察注意力机制对性能的提升效果。同样，我们选择 18 层、34 层、50 层的 SE-ResNet，以深入了解 SE 模块在不同深度下的作用。

表 2 SE-ResNet 架构

layer name	Output size	SE-18-layer	SE-34-layer	SE-50-layer
cony1	112x112	7x7, 64, stride 2		
conv2_x	56x56	3 x3 max pool, stride 2		
		$\begin{bmatrix}3x3, & 64\\ 3x3, & 64\\ \text{fc}\end{bmatrix}$x2	$\begin{bmatrix}3x3, & 64\\ 3x3, & 64\\ \text{fc}\end{bmatrix}$x2	$\begin{bmatrix}1x1 & 64\\ 3x3 & 64\\ 1x1 & 256\\ \text{fc}\end{bmatrix}$x3
cony3_x	28x28	$\begin{bmatrix}3x3, & 128\\ 3x3, & 128\\ \text{fc}\end{bmatrix}$x2	$\begin{bmatrix}3x3, & 128\\ 3x3, & 128\\ \text{fc}\end{bmatrix}$x4	$\begin{bmatrix}1x1 & 128\\ 3x3 & 128\\ 1x1 & 512\\ \text{fc}\end{bmatrix}$x4
cony4_x	14x14	$\begin{bmatrix}3x3, & 256\\ 3x3, & 256\\ \text{fc}\end{bmatrix}$x2	$\begin{bmatrix}3x3, & 256\\ 3x3, & 256\\ \text{fc}\end{bmatrix}$x6	$\begin{bmatrix}1x1 & 256\\ 3x3 & 256\\ 1x1 & 1024\\ \text{fc}\end{bmatrix}$x6
cony5_x	7x7	$\begin{bmatrix}3x3, & 512\\ 3x3, & 512\\ \text{fc}\end{bmatrix}$x2	$\begin{bmatrix}3x3, & 512\\ 3x3, & 512\\ \text{fc}\end{bmatrix}$x3	$\begin{bmatrix}1x1 & 512\\ 3x3 & 512\\ 1x1 & 2048\\ \text{fc}\end{bmatrix}$x3
	1x1	average pool, 1000-d fc.softmax		

选择 Swin Transformer 作为基于 Transformer 机制的模型，通过选择不同深度和宽度（Tiny、Small、Base），我们能够研究注意力机制对性能的影响，并与传统 CNN 进行对比。

表 3 Swin Transformer 架构

	downsp. rate (output size)	Swin-T	Swin-S	Swin-B
stage 1	4× (56×56)	concat 4×4, 96-d, LN	concat 4×4, 96-d, LN	concat 4×4, 128-d, LN
		$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 96, head 3}\end{bmatrix}\times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 96, head 3}\end{bmatrix}\times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 128, head 4}\end{bmatrix}\times 2$
stage 2	8× (28×28)	concat 2×2, 192-d , LN	concat 2×2, 192-d , LN	concat 2×2, 256-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 192, head 6}\end{bmatrix}\times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 192, head 6}\end{bmatrix}\times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 256, head 8}\end{bmatrix}\times 2$
stage 3	16× (14×14)	concat 2×2, 384-d , LN	concat 2×2, 384-d , LN	concat 2×2, 512-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 384, head 12}\end{bmatrix}\times 6$	$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 384, head 12}\end{bmatrix}\times 18$	$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 512, head 16}\end{bmatrix}\times 18$
stage 4	32× (7×7)	concat 2×2, 768-d , LN	concat 2×2, 768-d , LN	concat 2×2, 1024-d , LN
		$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 768, head 24}\end{bmatrix}\times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 768, head 24}\end{bmatrix}\times 2$	$\begin{bmatrix}\text{win. sz. } 7\times7,\\ \text{dim 1024, head 32}\end{bmatrix}\times 2$

在我的研究中，我选择了交叉熵损失函数作为模型的损失函数，通过这一选择，能够更好地适应图像分类任务，因为交叉熵损失通常用于多分类问题。相比于均方误差（MSE），交叉熵损失对分类问题更为敏感，有助于模型更准确地学习类别之间的差异。

为了优化我们的模型，采用了 Adam 优化器，其中学习率被设置为 0.0002。Adam 优化器结合了梯度的一阶矩估计和二阶矩估计，具有较好的性能和收敛速度。通过使用 Adam 优化器，我们期望能够更有效地更新模型参数，以最小化损失函数。

3. 实验与结果分析

本实验计划探究不同深度不同类型的网络共 9 个，预计训练时间长，由于手中的计算资源有限，而且报告提交截止日期提前，所以决定使用较小的数据集；并且为了尽可能地还原经典论文代码，选择图片的大小也应该和 iamgeNet 数据集大小一致，为 224*244，在网上找到了唯一一个满足两个条件的数据集——花朵数据集，此数据集包含五种不同种类的 224*224 花朵图像，用于训练的图像有 3306 张，用于验证的图像有 364 张。认为能够降低计算量的同时，保证最大程度上还原经典论文代码。

为了充分体现控制变量法，我选择在 Colab 平台上利用 Tesla V4 GPU 进行在线训练。Colab 提供了一致性的硬件环境，确保了实验在相同的计算资源下进行，从而有效地降低了变量差异性。

设计了一个消融实验和三个对比试验。在 ResNet 中插入 SE 模块，进行一个消融实验。对 ResNet 采用 18 层，34 层和 50 层的不同深度进行对比实验;对 SE-ResNet 采用 18 层，34 层和 50 层的不同深度进行对比实验;对 swin-transformer 采用 Tiny，Small 和 Base 三种不同深度和宽度进行对比实验。

3.1. 消融实验

通过消融实验的设计，我们旨在深入揭示在 ResNet 中引入 SE 模块的作用机制，并验证 SE 模块对 ResNet 性能的实质影响。

我们将 SE 模块 SE 模块插入到基本块的最后一个卷积层之后，因为在 ResNet 结构中，最后一个卷积层的输出通常具有更高的抽象特征表示，这样的位置可能更有助于 SE 模块对不同层级的特征进行调整，提升网络性能。此外，插入在这个位置上也有助于减少模型的参数数量，降低计算复杂度。

表 4 SE-ResNet 消融实验结果

	train Loss	training accuracy	val_accurate
ResNet-18	4.3E-05	1	0.744505
ResNet-34	2.4E-05	1	0.695055
ResNet-50	1.8E-4	1	0.71978
RE-ResNet-18	2.3E-05	1	0.758242
RE-ResNet-34	1.8E-05	1	0.766484
RE-ResNet-50	6.0E-06	1	0.801314

通过观察 SE-ResNet 消融实验结果，横向对照 SE-ResNet 的实验结果，随着网络深度的增加，训练损失似乎有所增加。例如，ResNet-50 的训练损失（1.8E-4）相对于 ResNet-18（4.3E-05）更高。这可能是因为更深层的网络需要更多的训练来适应数据。在验证集上，不同深度的网络表现出差异。例如，RE-ResNet-50 相对于 RE-ResNet-34 和 RE-ResNet-18 具有更高的验证准确率。这可能表明增加网络深度有助于提高模型对未见过数据的泛化能力。

将 SE-ResNet 相对于对应深度的 ResNet 进行横向对照，可以看到 SE-ResNet 在验证准确率上的提升。例如，RE-ResNet-18 相对于 ResNet-18 在验证准确率上有所提高。这表明引入 SE（Squeeze-and-Excitation）模块对模型性能有积极的影响。

在这组实验中，RE-ResNet-50 相对于其他深度的网络在训练损失和验证准确率上都表现得较好。这可能表明在这个任务上，更深层次的网络结构以及 SE 模块的引入能够提高模型性能。

综合来看，通过引入 SE 块，特别是在 ResNet-50 的基础上，模型在验证集上的性能有所提升，表现为更低的训练损失和更高的验证准确率。这表明 SE-ResNet 在这个任务上可能更有效地学习了特征。当然，具体的规律可能还受到任务特性和数据分布的影响，因此在选择模型时还需谨慎考虑。

3.2. 对比实验

对 ResNet 采用 18 层，34 层和 50 层的不同深度进行对比实验;对 SE-ResNet 采用 18 层，34 层和 50 层的不同深度进行对比实验;对 swin-transformer 采用 Tiny，Small 和 Base 三种不同深度和宽度进行对比实验。

表 5 对比实验结果

	train Loss	training accuracy	val_accurate
ResNet-18	4.3E-05	1	0.744505
ResNet-34	2.4E-05	1	0.695055
ResNet-50	1.8E-4	1	0.71978
RE-ResNet-18	2.3E-05	1	0.758242
RE-ResNet-34	1.8E-05	1	0.766484
RE-ResNet-50	6.0E-06	1	0.801314
Swin-tniy	0.003458	0.986388	0.733516
Swin-small	0.03574	0.975499	0.722527
Swin-base	0.344897	0.686328	0.664835

ResNet 不同深度对比实验，通过选择 ResNet 在不同深度下的配置，探究其性能随深度变化的趋势，为深度与性能关系提供实证分析。选用经典的 ResNet 模型，并分别配置为 18 层、34 层和 50 层，通过实验观察其在相同任务上的性能表现。

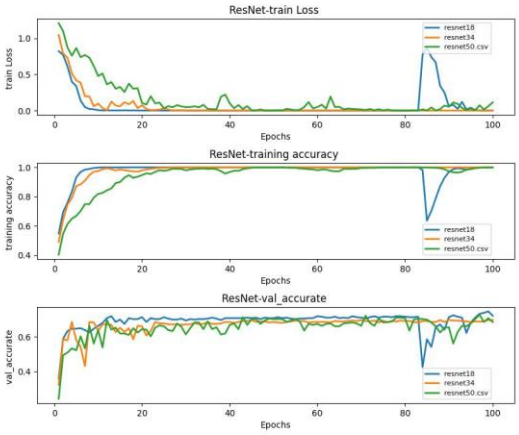

图 1 ResNet 不同深度对比

本研究中，我们通过使用 ResNet 在不同深度下的配置（分别为 18、34 和 50）来探究模型性能随深度变化的趋势。在训练过程中，我们观察到 ResNet-18 表现出极低的训练损失（4.3E-05）和完全的训练准确率（1），但在验证集上的准确率为 0.7445，表明模型可能存在一定程度的过拟合。相比之下，ResNet-34 在验证集上的准确率略低于 ResNet-18，为 0.6951，但表现出更低的训练损失（2.4E-05）。而 ResNet-50 在训练集上表现出稍高的训练损失（1.8E-4），但在验证集上的准确率为 0.7198，介于 ResNet-18 和 ResNet-34 之间。

从实验结果中可以观察到，随着 ResNet 深度的增加，训练集上的准确率基本保持满分，但在验证集上的性能表现却呈现出一定的波动。这可能表明在一定程度上，较深的 ResNet 网络在验证集上的泛化性能存在挑战，而选择适当的深度可能更有利于获得更好的性能。这对于深度神经网络的设计和选择提供了有价值的参考。

SE-ResNet 不同深度对比实验，分析引入 SE 模块后，SE-ResNet 在不同深度下的性能差异，验证 SE 模块对不同深度 ResNet 性能的调节效果。选择 18 层、34 层和 50 层的 ResNet 作为基准，分别在其基础上插入 SE 模块，通过实验观察 SE 模块在不同深度下的性能影响。

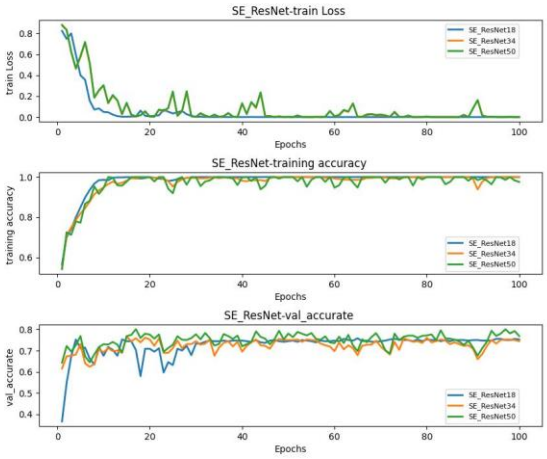

图 2 SE-ResNet 不同深度对比

在本研究中，我们聚焦于 SE-ResNet 在不同深度（RE-ResNet-18、RE-ResNet-34 和 RE-ResNet-50）下的性能表现。在训练过程中，观察到 RE-ResNet-18 表现出较低的训练损失（2.3E-05）和完全的训练准确率（1）。在验证集上的准确率为 0.7582，这显示出相对良好的泛化性能。随着深度的增加，RE-ResNet-34 表现出更低的训练损失（1.8E-05），同时在验证集上的准确率进一步提升至 0.7665。而 RE-ResNet-50 在深度增加的情况下展现出更为显著的性能提升，具有较低的训练损失（6.0E-06）和在验证集上的高准确率，达到了 0.8013。

实验结果表明，随着 SE-ResNet 深度的增加，模型在验证集上的性能逐步提升。RE-ResNet-50 相对于 RE-ResNet-34 和 RE-ResNet-18 展现出更好的性能，这可能反映了引入 SE 模块对于模型性能的正面调节效果。然而，需要进一步的实验和分析来深入理解 SE 模块在不同深度下的作用机制，以更全面地评估其对深度神经网络性能的贡献。

Swin Transformer 不同深度和宽度对比实验，研究 Swin Transformer 在不同深度和宽度下的性能表现，探索其在与传统 CNN（ResNet）和具有注意力机制的 SE-ResNet 的对比中的优势和特性。选用 Swin Transformer 模型，并分别配置为 Tiny、Small 和 Base 三种深度和宽度，通过实验观察其性能差异。

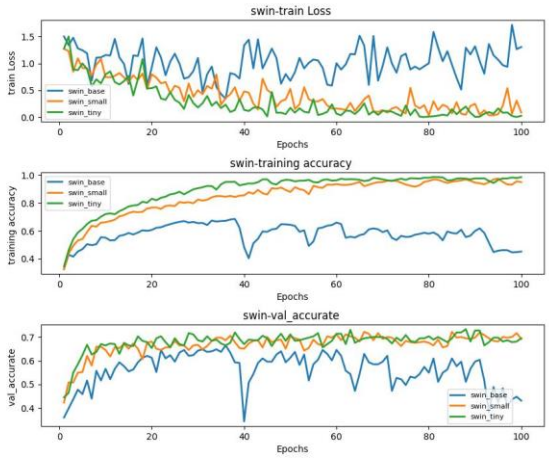

图 3 Swin Transformer 不同深度对比

在本研究中，我们对 Swin Transformer 在不同深度配置下的性能进行了深入分析，分别为 Swin-Tiny、Swin-Small 和 Swin-Base。在训练过程中，Swin-Tiny 表现出相对较低的训练损失（0.003458）和高训练准确率（0.9864），但在验证集上的准确率为 0.7335，这可能表明模型在泛化到验证集时存在一定的挑战。相比之下，Swin-Small 在训练集上的准确率稍低（0.9755），但在验证集上表现相对稳定，准确率为 0.7225。而 Swin-Base 在深度增加的情况下，显示出更高的训练损失（0.3449）和相对较低的训练准确率（0.6863），同时在验证集上的准确率为 0.6648。

实验结果揭示了在 Swin Transformer 中，随着深度的增加，模型在验证集上的性能并非一味提高，而是呈现出一定的波动。Swin-Tiny 在训练集上表现出较好的性能，但在验证集上可能存在一定的过拟合,并且在训练过程中准确率浮动较大。Swin-Small 相对稳定，而 Swin-Base 在深度增加时的性能并没有如预期般提升，这可能涉及到深度与性能的平衡问题。

4. 结论

本研究通过系统性的消融实验和对比试验，关注了 ResNet、SE-ResNet 和 Swin Transformer 在不同深度下的性能表现，着重考察了注意力机制对模型性能的调节作用。同时，对 Swin Transformer 的不同配置进行了深入研究，为深度神经网络的设计提供了有益的见解。

实验结果揭示了不同深度模型在性能上的差异，以及注意力机制对性能的调节效果。在不同深度模型的比较中，ResNet-34 表现相对稳定，而 SE-ResNet 通过引入 SE 模块实现深度增加时的性能提升。Swin Transformer 中，Swin-Tiny 表现良好但存在泛化挑战，Swin-Small 相对稳定，而 Swin-Base 在深度增加时性能波动，涉及深度与性能平衡。这些结果揭示了深度对模型性能的影响，SE 模块的正面调节效果以及 Swin Transformer 在不同深度下的多样性能特点。这对深度神经网络设计提供了有益的见解。

Swin-transformer 没有真正的收敛。观察 loss 值，发现 Swin-transformer 在训练 100epoch 后也没有真正的收敛，但是，出于控制变量的初衷、囿于算力和时间限制，没有对所有模型的训练 epoch 做调整，下一步可以将 epoch 调整到 1000，再进行所有模型的对比，以可以作为一项未来工作。

SE-ResNet 存在过拟合情况。SE 属于**通道注意力**。所谓的通道，在 CNN 中就是特征图与 Filter 进行卷积后的结果。每个 Filter 一般来讲会产生一个通道，那么一个通道可以被通俗地理解为一个**特征**。在通道注意力机制中，SE 模块主要通过 Squeeze-and-Excitation（SE）过程对通道进行加权，而在这个过程中引入 Dropout 可以帮助随机地“关闭”一些通道，减少对特定特征的过度依赖，从而提高模型的泛化性能。所以我想对 Key 值进行 Dropout，来起到降低过拟合的程度。但是我是在 1 月 18 日上午写这篇论文报告的时候产生的想法，进行了简单测试，但是已经没有时间进行完整的训练了，所以可以作为一项未来工作。

5. 参考文献

[1] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[J]. arXiv preprint arXiv:1709.01507, 2017, 7.

[2] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[3] Liu, Ze et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows.” *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021): 9992-10002.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” ICLR, 2021.

[6] Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, and C. Xu, “Transformers in computational visual media: A survey,” Computational Visual Media, vol. 8, no. 1, pp. 33–62, 2022.
[7] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, “A survey on visual transformer,” 2021.
[8] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” 2021.
[9] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-andexcitation networks,” 2019.