哈尔滨工业大学

《模式识别与深度学习》实验

题目：**基于 Vision Transformer 的花卉图像分类**

学　　号：120L022221

姓　　名：秦宇航

班　　级：2003602

专　　业：人工智能

学　　院：计算学部

指导教师：金野

完成日期：2023 年 6 月 12 日

摘　要

图像分类是计算机视觉的方向之一，随着不同模型的出现，图像分类提供越来越精准的分类预测。本实验的工作内容是基于 Vision Transformer (ViT)的花卉图像分类，数据集采用的是包括雏菊、玫瑰、向日葵、蒲公英、郁金香等五种花卉的数据，采用 Vision Transformer 的深度模型框架对模型训练，并进行最后的预测。并且工作中还采用 Resnet 和 Alexnet 两种模型训练和预测结果与 ViT 进行结果比对。最后将 Alexnet 模型的卷积网络加入到 ViT 模型中进行训练，训练效果相较于原始 ViT 和 Alexnet 准确率都有提升。

实验结果在没有采用预训练模型或迁移学习的情况下，ViT 的准确率并不理想，但是在使用预训练模型的情况下，准确率可以达到 97%。并且在改进的 ViT 模型上准确率也有比较大的提升。

关键词：Vision Transformer；注意力机制；花卉图像分类；Alexnet

目 录

第一章 绪论

第一节 引言

图像分类是在给定的图像数据中按照事先规定的物品种类进行区分的办法，可以按照特征分类、子类细粒度分类以及实例级图像分类。图像分类不仅是计算机视觉中比较基础的工作，也是后续的图像分割、图像识别、物体跟踪的基础操作。图像分类检测技术在众多领域得到广泛应用，本实验使用图像分类技术对花卉进行分类。

第二节 研究目的和内容

传统的机器学习分类方法采用通过识别花朵的颜色、纹理、形态等特征通过相似度来计算花卉种类的概率，随着深度学习的发展，计算机视觉领域也得到提升，图像分类的技术也取得了比较大的成功，但是对于花卉的分类研究，由于花卉的相似程度较高，在深度学习过程中图像的一些特征也许无法准确识别，这也是深度学习关于花卉分类研究首要任务。

如今在计算机视觉中最活跃的便是 Transformer 模型。Transformer 是一种基于注意力机制的编码器解码器模型，最早使用在自然语言处理领域，它是由 Google 团队在 2017 年发表的，如今也应用于像 BERT、GPT 等语言模型中。Transformer 相比于 RNN 网络复杂度从 O(n)降低到 O(1)。Transformer 由编码模块和解码模块两个部分组成，编码模块和解码模块都包含 6 个块。模型将输入的每个单词表示为向量并加上一个位置信息组成，这是因为 Transformer 本身是不能记住单词的顺序信息的，因此需要在输入的末尾中添加位置 Embedding，之后将词向量输入到编码模块中得到编码信息矩阵，再将编码信息矩阵传输到 D 解码模块通过 softmax 层后开始预测[1]。而在 Transformer 模型中最重要的是他的自注意力机制结构，它是由多个 Self-Attention 组合成 Multi-Head Attention 的发挥作用，它可以用来计算单词之间多维的相关系数。Self-Attention 自注意力机制结构是用到的 K、Q、V 三个矩阵通过输出进行线性变换得到。后来随着科学家对于 Transformer 的自注意力机制研究与发展，并且在计算机视觉领域也在发展，

受到 Transformer 的启发，将其自注意力机制应用到计算机视觉领域当中。2020 年，Dosovitskiy 等人提出了 Vision Transformer(ViT)[2]模型改变了计算机视觉领域被卷积神经网络的统治阶段，他将图像分成图像块并将图像块输入到模型当中利用 Transformer 的机制，在各种计算机视觉的数据集上取得了非常好的效果超过了当时最好的神经网络模型，但是 ViT 模型只能在大规模百万级数据集上才能取到良好的结果，对于日常普通的训练集并不如卷积神经网络[3]。随后科学家们便开始对 Transformer 模型进行各种改进和创新，Transformer 模型在层次过深的时候便会进入饱和状态。2021 年科学家提出了采用 Re-attention 模型的 Deep ViT 重新生成 attention map 增强更层次之间的多样性。科研团队有在接下来的工作中提出了 Token-to-Token ViT，相较于传统 ViT 模型参数量大大减少，然而性能得到很好的提升，训练又快准确度也更高，模型通过设计的 Transformer-based 网络可以不需要在大型数据集上训练也能实现比卷积神经网络效果更好的结果，而且通过实验还证明了 deep-narrow 网络能够增加特征的丰富性。同时也有将卷积神经网络和 Vision Transformer 结合起来达到更快的速度的 LeViT 模型。CNN 网络的卷积操作可以是相邻的像素点得到比较相似的梯度，Transformer 中 Attention 结构是不重叠的，而且通过数据增强操作也在一定程度上使数据的空间结构更加平滑，于是作者通过将两个网络结合以达到更高的精度和更快的速度。

第二章 数据集预处理

本节是对数据集的介绍和对于数据集的处理方面进行讲解。由于在数据集中图片的大小、清晰度等不相同，所以需要数据预处理的方法进行提前加工处理从而达到比较好的训练效果，其次由于数据集规模较小，通过数据集的预处理可以有效地扩大数据集的规模，对于实验的验证更有说服力。

第一节 数据集介绍

实验中使用的数据集是 Kaggle 网站上花卉图像五分类数据集，其中的图片数据来自于 Flickr、Google 图片和 Yandex 图片网站。其中的花卉种类包括雏菊、玫瑰、向日葵、蒲公英和郁金香五种类别，其中雏菊的图片共 633 张，蒲公英的图片共 898 张，玫瑰的图片 641 张，向日葵的图片 699 张，郁金香的图片共 799 张，总共 3670 张。五类图像展示如图 2.1、图 2.2、图 2.3、图 2.4 和图 2.5：


图 2.1 雏菊


图 2.2 蒲公英


图 2.3 玫瑰


图 2.4 向日葵


图 2.5 郁金香

第二节 数据集的划分

再进行训练过程中数据集中包括训练集、验证集和测试集三个部分，合理的划分数据集有利于对实验的训练，可以更好地提高准确度。并且在划分数据集的时候，训练集、验证集和测试集三个部分中的数据不能重复，否则会影响实验训练的效果和最终的准确率。数据集的划分主要使用的方法是留出法和交叉验证法两种。

留出法划分数据集是在保证数据一致性的前提下，采用分层抽样的方法来进行划分，即在数据集的每一个字集中按照规定的比例进行抽取，并且抽取方法必须按照随机的概率进行，一般是按照 7：3 或者 8：2 进行划分。缺点便是由于只是通过一次划分数据的分类不具有普遍性。

交叉验证法主要是 k 折交叉验证法。就是将原数据集化分为 k 个不相关的子数据集，然后进行 k 次的训练和验证，每一次训练 k-1 个子训练集，并用第 k 个训练集进行验证，最后将求得验证和训练的误差进行平均值计算，平均值的计算方法可以采用算术平均和几何平均的方法[4]。

本实验采用的是留出法划分数据集，留出法划分数据集时将原始数据集按照一定的比例并进行分层抽样的方法分为训练集和验证集，此过程采用的是代码进行划分，可以保证随机性。由于使用的数据集中数据的数量过少，所以本实验采用的比例是按照 9：1 的方式划分为训练集和验证集，其中测试集数据也是预测数据集，本实验从每个种类花卉中分别选取一张图片作为最后实验的预测图片，将图片输入进行预测的代码并给出对五种类型花卉对应各个种类的概率，其中概率最大的就是最终结果，最终对比最后预测花卉图片数据集的准确率来进行多种模型的评价。

第三节 数据的预处理

本实验所使用的数据集规模有限，在训练过程中为了防止发生过拟合现象和提升训练的准确性需要对原始数据集进行一定预处理作为数据集的扩充。预处理技术包括图像的增广技术，通过对图像进行多种随机的图像变化，进而产生相似但是却并不相同的训练样本，从而实现训练集的扩充操作，而且本文还通过叠加多种增广方式来处理数据。由于图像训练的过程当中，图像的品质有所不同，所以对于训练过程的影响也比较大，所以在训练过程之前通过一系列的预处理方法可以提升模型训练的准确度。由于原数据集中图像数据的大小不同，所以在做实验工作时，首先要进行图像的裁剪工作，将数据集中不同大小的图片统一处理成为 224×224 大小的图片。并且数据集中存在一些灰度图像，灰度图像对训练模型和特征提取等工作也有一定影响，所以还需要将数据集中的灰度图像进行删除。之后需要对图像进行增强数据的操作，增强图像中的可以被利用的信息，主要是为了改善图像的视觉效果，而且在不同模式下的图像需要使用不同方式的图像增强方法对强调图像的整体或局部特性，将原来效果不好的图像进行优化，更有利于后续训练过程中的图像特征提取。

在本实验中采用预处理的方法有随机裁剪、随机水平翻转、随机上下翻转、亮度调整等图像增广方法。之后还需要将所有数据集中的图片转换格式并作归一化处理。在裁剪的时候需要将所有图片处理为 224×224 的图像，因为在后面 ViT 模型中输入的图像必须是 224×224 格式的。在水平和上下翻转图片的操作将概率设置为 50%的情况，并且通过调整亮度饱和度等方式也可以提高数据集的多样性，同时对模型的训练有利于增强模型的准确性和泛化性。并且在训练集和验证集当中的图像预处理方法需要保持一致，保证模型的准确运行。图像预处理及原图像如表 2.1 所示，第一行为所经过的图像预处理方法，图片为图像处理后的结果。

表 2.1 图像处理结果

原始图片	上下翻转	左右翻转	随机裁剪	调整亮度
				
				

第三章 基于 Vision Transformer 的图像分类设计

近些年来，ViT 在计算机视觉的崛起，将图像分类技术推向新的高度。花卉图像相较于其他的图像分类，由于花卉在一些特征上比较相似，所以分类也有一定难度，但是在深度学习下，通过超强的算力也可以取到良好的结果。

本章节主要从 ViT 模型的介绍、改进的模型讲解、最终结果的评价等方面进行安排撰写。

第一节 Transformer 模型

Transformer 模型是在 CNN、RNN、MLP 之后的第四大模型，它是一种基于注意力机制的模型，并没有采用卷积和循环的操作，论文在 2017 年由 Google 团队发表，在论文中对机器翻译进行实验，击败了当时机器翻译中的各个模型，而且模型还大大提高了训练时间，所以在之后几年 Transformer 模型在自然语言处理领域十分火热。2020 年 Transformer 开始在 CV 领域大放异彩，随着 Google 团队将 Transformer 模型应用到计算机视觉领域，在后来的这几年大量的关于 Transformer 的计算机视觉模型开始出现并且在计算机视觉领域占据主导地位。

3.1.1 Self-Attention 机制

注意力函数可以描述为映射一个 *query* 和一组关于维度 *dimension* 的 *key-value* 对的一个输出，模型如图 3.1。在计算时通过计算 *query* 和 *key* 的点积除以 $\sqrt{d_k}$，再将结果通过 softmax 层进行处理并与 *value* 进行乘积。而在训练的时候将输入的数据节点通过 InputEmbedding 映射，再将映射结果通过变化矩阵 W_q、W_k、W_v，同时并行计算一组的 *query*、*key* 和 *value* 得到 Q、K、V 三个矩阵并进行最后注意力计算，公式如下:

$$\text{Attention(Q,K,V)} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})\text{V} \tag{3.1}$$

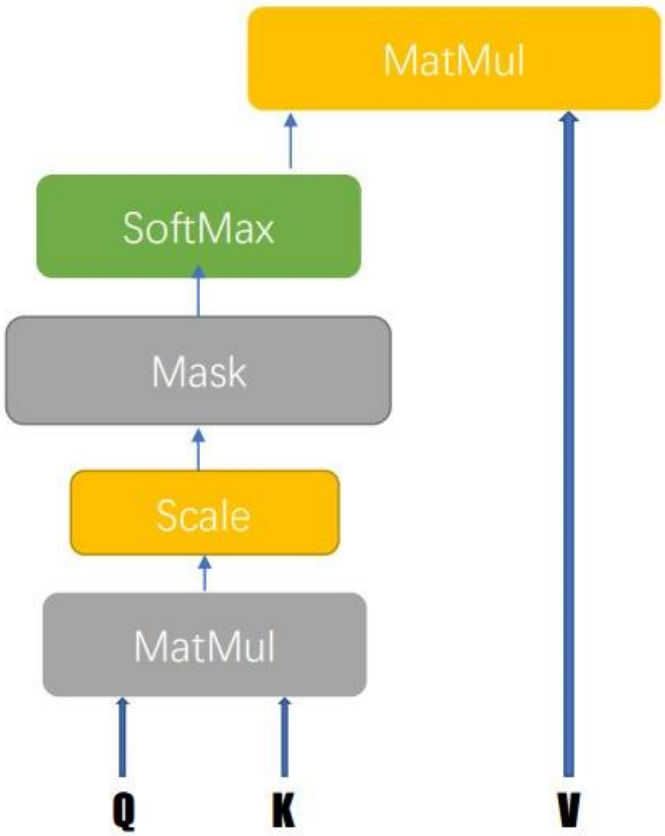

图 3.1 Self-Attention 计算模型

3.1.2 Multi-Head Attention 机制

Transformer 中另一个重要的机制就是 multi-head Attention 结构，多头注意力机制能够联合来自不同头部学习到的信息，他其实是在自注意力机制上构建的，能够达到更好的训练效果。他将每个输入节点 a_i 通过 W^q、W^k、W^v 矩阵得到 q_i、k_i、v_i，之后根据所使用的 *head* 数量将结果分为 *head* 份，将每个 $head_i$ 里面的 q_i、k_i、v_i 进行拼接得到 Q_i、K_i、V_i，接着通过自注意力机制的公式(3.1)得出每个 *head* 的注意力结果进行拼接，再通过可学习的参数 W^O 进行融合得到最终结果，公式如下：

$$\text{MultiHead}(Q, K, V)=\text{Concat}(\text{head}_1,...,\text{head}_h)\ W^O$$

$$\text{其中 head}_i=\text{Attention}\ (QW_i^Q,KW_i^K,VW_i^V) \qquad (3.2)$$

并且在 Transformer 模型中，首先在编码器解码器中 q 来自于前一个的解码器，k 和 v 来自编码器的输出。而且编码器中包含自注意力层，q、k、v 都来自编码器前一层的输出。解码器中的自注意力层允许解码器关注所有位置，通过删掉 softmax 输入中非法连接的 v 来实现缩放点积注意力。多头注意力机制如图 3.2 所示：

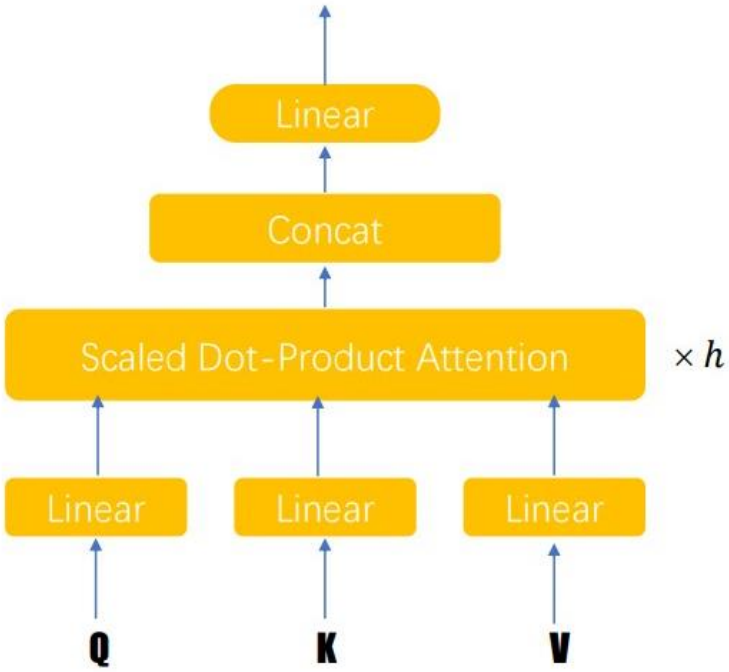

图 3.2 Multi-Head Attention 机制模型

3.1.3 Transformer 其他部分

首先比较重要的是位置编码，由于 Transformer 模型不使用循环和卷积的操作，为了保证输入的序列的顺序问题，必须添加一个位置编码信息。所以模型在编码器和解码器的输入中添加了相同维度的位置编码，最后再和输入进行相加。在位置编码中可以选择学习和固定的两种方法。

其次是 Add&Norm 模块，他是连接在 Encoder 和 Decoder 后面的，其中 Add 表示残差连接，Norm 表示的是 LayerNorm，是一种正则归一化处理防止过拟合现象的发生，通过对每一层的所有向量进行均值和方差并归一化到正态分布并通过学习调整到合适的数值。

第二节 Vision Transformer 模型

ViT 模型主要由三个模块组成，分别是 Linear Project of Flattened Patches、Transformer Encoder 、MLP Head。Transformer 模型将图像拆分成块，并将这些块按照顺序并添加上位置信息作为输入，经过上面三个模块最后得出分类概率。ViT 模型在中小数据集上的准确率并不是非常理想，是由于 ViT 模型缺少 CNN 一样的 biases，所以训练模型并不具备泛化能力，但是在大规模数据集上则取得了更加好的结果。

模型的操作步骤：首先是对数据进行处理，将图片数据分成块将输入的 224×224 的图片按照 16×16 的 *Patch* 划分，之后通过线性映射转变为一维向量，在对向量加入一个可学习的位置信息，再将分类向量与输入向量进行拼接，最后进行分类操作。模型结构如图 3.3。

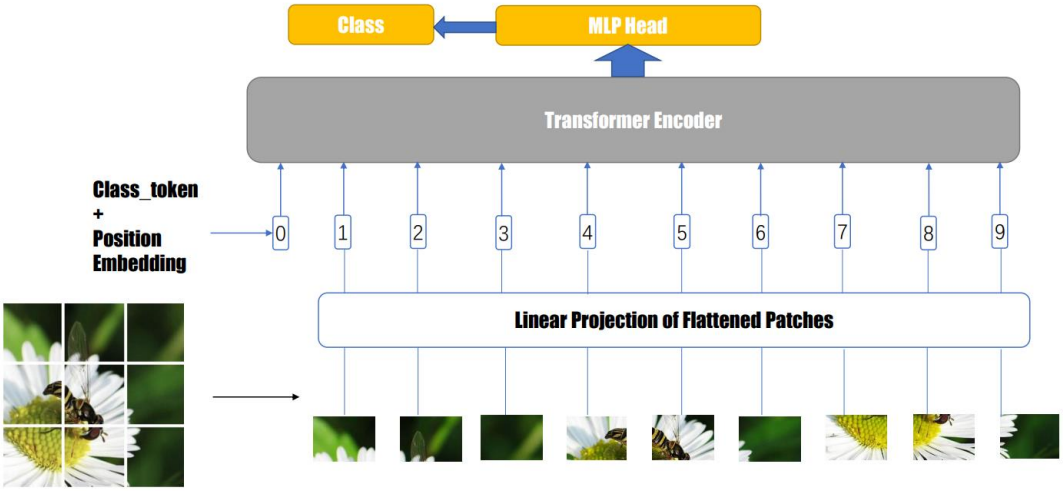

图 3.3 Vision Transformer 结构

1. Patch Embeddings 层设计

在这层中，transformer 模型要求输入为一个二维矩阵[*number_token,token_dim*],但是由于图像数据是三维矩阵[*H,W,C*]所以需要进行变换。首先将图片分为 16×16 的 *Patch*，划分后会得到 196 个 P，接着再将每个映射到一维向量中，这就变为一个二维矩阵[196, 768],其中 768 这个 *token* 维度是预先设置的。之后在刚刚得到的向量上插入一个进行分类判断的 *class_token*，而且这个向量是一个可以进行训练而得来的参数并且要和刚刚求得的 *token* 形式相

同，由于 ViT 当中并没有解码层，所以这个向量的作用充当 *Query* 的作用，又来寻找其他九个输入向量的图像类别。最后对于位置信息 Position Embedding 采用的是可训练的参数直接叠加在先前的 *token* 上，并且其形状也要对应，对位置编码进行训练可以得到位置越接近，往往具有更相似的位置编码，而且使用了位置编码可以对实验的准确度有比较好的提升。

2. Transformer Encoder 层设计

Transformer Encoder 层是一个重复堆叠 Encoder Block 的结构，其中包括多头注意力层，多层感知机块，还有归一化层进行处理，在每个块后还需要残差连接，在本实验中对其重复叠加 16 次。

首先是 Layer Norm 结构，这个是应用在自然语言处理领域的一种归一化处理方法用来加速网络的收敛,*E(x)*代表均值，*VAR(x)*为方差，ε是一个非常小的常量，防止分母为零，γ，β是两个可训练的参数。在这层需要将每个 *token* 进行归一化处理。Layer Norm 公式如下：

$$y = \frac{x - E(x)}{\sqrt{VAR(x) + \varepsilon}} * \gamma + \beta \tag{3.3}$$

多头注意力机制和在上面 Transformer 结构中讲述一样是能够联合来自不同头部学习到的信息，是在自注意力机制上构建的，能够达到更好的训练效果。他将每个输入节点 a_i 通过 W_q、W_k、W_v 矩阵得到 q_i、k_i、v_i，之后根据所使用的 *head* 数量将结果分为 *head* 份，将每个 $head_i$ 里面的 q_i、k_i、v_i 进行拼接得到 Q_i、K_i、V_i, 接着通过自注意力机制的公式(3.1)得出每个 *head* 的注意力结果进行拼接，再通过可学习的参数 W^O 进行融合得到最终结果。

Dropout 是在训练过程中按照一定概率将神经网络单元暂时从网络中丢弃，是一种防止过拟合的有效方法。

MLP Block 是一个全连接层加上 GELU 激活函数再加上 Dropout 组合而成的结构，再通过第一个全连接层后会将节点变为原来的 4 倍，之后通过 GELU 激活函数用非线性函数进行处理增加训练的泛化，接着再通过 Dropout、全连接层将节点还原为原来的形式，最后通过 Dropout 进行输出。

对于这一 Encoder 整体结构便是再通过 Embedding 后的数据进入编码器后先通过 Lay Norm 归一化，再将结果传到多头注意力层得出的结果与最开始的输入进行相加形成一个残差结构，接着将输出继续归一化处理，通过 MLP 层后与

第一次的输出进行相加有构成一次残差连接，在这一过程中形成一次Transformer Encoder的编码。总体结构如图3.4。

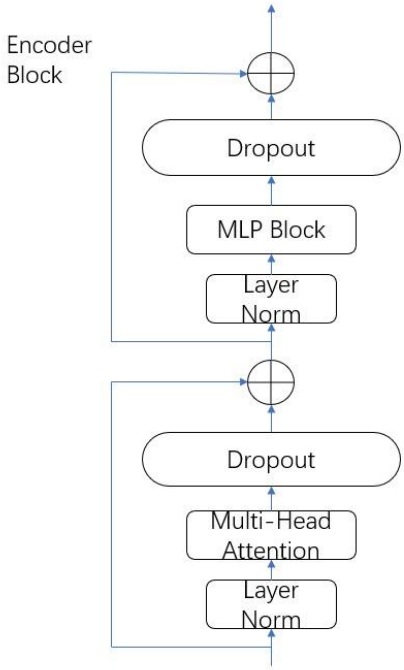

图3.4 Transformer Encoder

3. MLP Head层设计

通过Transformer Encoder后输出的向量形状和输入的形状是保持不变的，在ViT-B/16中输入与输出均是[197,768]的形状，这里本文工作只是需要分类的信息，所以只需要提取出 *class_token* 生成的对应结果就行。接着通过 MLP Head 得到最终的分类结果。MLP Head是由Linear全连接层和tanh激活函数再加上Linear全连接层组成。如果在迁移学习或者加再预训练权重过程中只需要Linear层。

第三节 学习率衰减方法对模型训练的影响

学习率是在训练模型的时候，计算损失函数梯度调整网络里面的一个超参数，学习率的选取对于训练模型的准确率有着非常重要的作用，学习率过大或者过小都会影响损失函数，一个合适的学习率和良好的学习率更新算法可以保证模型更快地到达损失函数的最小值，保证收敛的损失值是比较优的解。

学习率衰减的策略各种各样，本实验采用轮数衰减和余弦学习率衰减[5]。轮数衰减的学习率是指在指定特定的训练次数后学习率按照设定的比例进行衰减，

每经过特定训练次数就会衰减一次，最后达到收敛效果。余弦学习率衰减方法是按照训练过程按照余弦函数曲线下降，在初始和结束的时候曲线下降较慢，在中间过程中学习率衰减较快。余弦学习率衰减的公式如下：

$$L_t=\frac{1}{2}(1+\cos(\frac{t\pi}{T}))L \tag{3.4}$$

T 是 batch 总数，L_t 是在第 t 个批时的学习率大小。

下面两幅图，图 3.5，图 3.6 是两种学习率衰减策略下学习率随着训练次数变化图像。

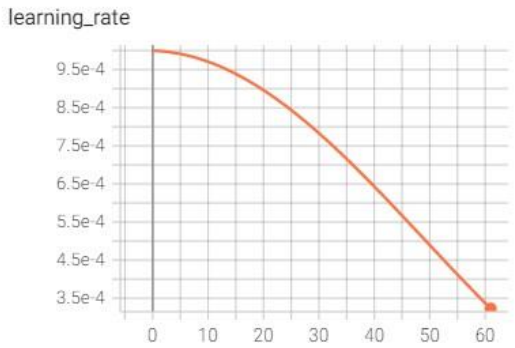

图 3.5 余弦学习率衰减

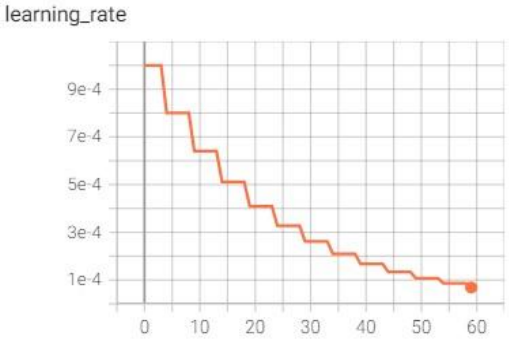

图 3.6 轮数衰减

第四节 基于 Alexnet 改进的 ViT

Alexnet 是 A. Krizhevsky 在 2010 年提出的并在同年的 ILSVRC 比赛中夺得桂冠。而在本实验的改进中主要是通过改变 ViT 模型中 Patching Embedding 层中的卷积网络实现的。在原本的模型中，只通过一次卷积操作，卷积核大小为 16×16，步长为 16 的卷积，将图片变成 14×14 的图片，并且深度为 768。在卷积操作中卷积层越多，越有利于提升训练的准确率，而 Alexnet 是一个包含五层卷积操作的网络，所以在实验改进时首先想到 Alexnet 网络加深网络层数来优化准确率问题[6]。设计的模型图如图 3.7 所示。

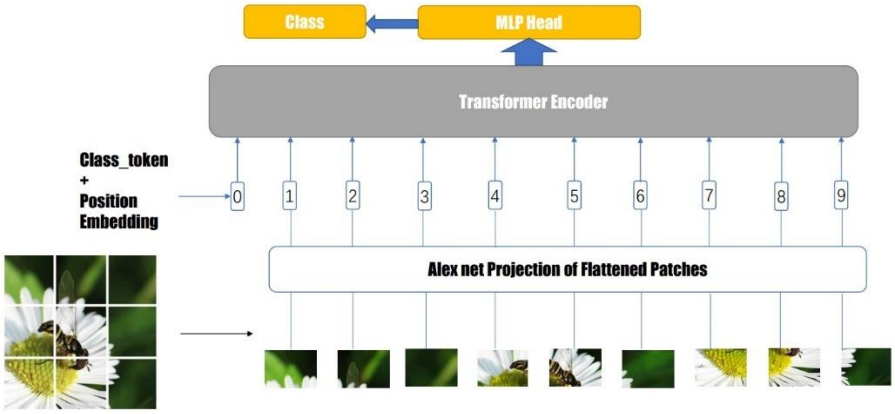

图 3.7 基于 Alexnet 改进的 Vision Transformer 模型

然而 Alexnet 网络最终处理图片数据得到的特征是 6×6×256 的特征矩阵，而 ViT 模型需要传入的特征矩阵是 14×14×768，所以需要在 Alexnet 网络基础上对卷积操作和池化操作进行更改，本实验中 Alexnet 网络结构如图 3.8 所示，图中标注了每次处理之前图片的特征大小和每次卷积操作所需要的 *kernel size*、*stride* 和 *padding* 的大小，并且最终得到所期待的特征矩阵大小。

图片首先经过卷积层 1，将[224,224,3]的图片输出为[109,109,48]的特征，之后经过 ReLu 激活函数和 Maxpool 输出为[54,54,48],按照图 3.8 的 5 层卷积操作后得到最终[14,14,768]的特征矩阵形状。

矩阵尺寸大小计算公式如下：

$$N = \frac{W-F=2P}{S} + 1 \tag{3.5}$$

W是输入图片大小，F是卷积核大小，P是 padding 的像素个数，S是步距。

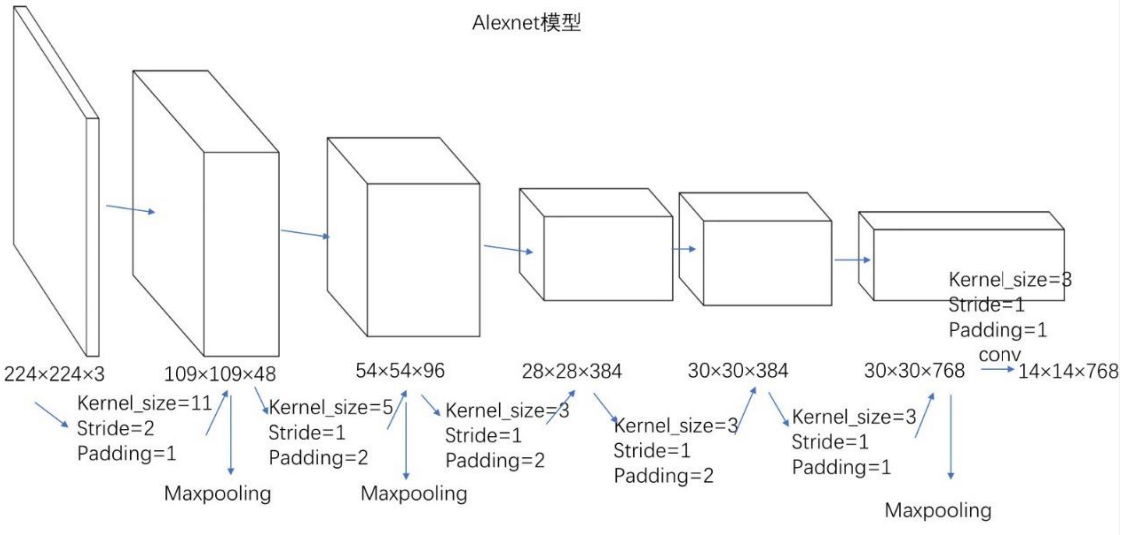

图 3.8 改进的 Alexnet 模型

第五节　实验结果与对比分析

3.5.1 实验环境配置

花卉分类图像数据集包括雏菊、玫瑰、向日葵、蒲公英和郁金香五种类别，共计3670张图片，采用图像增广的操作进行数据集的扩充。

实验的硬件环境：使用的GPU是NVIDIA GeForce GTX 1660Ti，CPU是Intel(R) Core(TM) i7-9750H。

本实验使用的编程语言是python 3.9，深度学习系统框架是pytorch，此外还使用了tensorboard进行可视化操作。

在训练过程中，采用SGD优化器，学习率大小为0.001，动量系数是0.9，权重衰减是0.00005，训练的循环次数是100个循环，批大小设置为8。本实验还采用预训练模型，预训练模型未使用模型训练参数一致，但是循环次数调整为60循环，因为经过多次实验发现在60循环时已经趋近收敛。

3.5.2 实验评价指标

对于实验结果的评价，本实验的评价标准是对于验证集的准确度和测试集5张图片预测的准确度概率。评价标准中还有例如实验性能、训练损失值等指标。

对于分类问题验证集准确率计算公式如下：

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.6}$$

3.5.3 实验结果与对比分析

实验结果是从ViT模型、Resnet模型、Alexnet模型三种模型中对比实验得到的，并且在实验中还采用了在大数据集上预训练模型进行比较，并且还通过在预训练模型上轮数衰减的学习率衰减和余弦学习率衰减进行对比，主要的评价指标是准确度、性能、损失值、还会对测试集每张图片预测的准确率进行比较。其中性能为训练一个循环所需要时间单位为循环/秒。ViT_Cosinelr、Resnet_Cosinelr、Alexnet_Cosinelr是未采用预训练模型，preViT_Cosinelr、preRes_Cosinelr是是用

于训练权重进行训练的模型，其中前五个模型采用的是余弦学习率衰减，PreViT_Steplr 使用的是轮数衰减的学习率，实验结果如表 3.1：

表 3.1 实验结果及对比

模型	准确度	损失值	训练时间(s)
ViT_Cosinelr	0.758	0.544	305
Resnet_Cosinelr	0.881	0.220	57
Alexnet_Cosinelr	0.844	0.352	30
PreViT_Cosinelr	0.978	0.013	270
PreRes_Cosinelr	0.963	0.041	50
PreViT_Steplr	0.985	0.0009	263

通过实验结果对比分析，首先在未采用预训练模型下的对比，在准确度并定的指标下，Resnet 的准确度最高，其次是 Alexnet，最后是 ViT 模型，这是由于 ViT 模型更适合在大规模数据集下进行试验，而在本次实验的数据集属于小数据集，所以准确度较低，而且由于 ViT 模型的参数更多所以训练时间也较长。

但是在加入预训练模型的情况下，ViT 的准确率要好于 Resnet 模型，准确率可以达到 97.8%，而且损失值也只有 0.013，所以在有大规模数据集进行预训练的情况下，ViT 模型对于本次实验的分类效果更好

再通过学习率衰减策略不同的对比发现，轮数衰减的学习率相较于余弦学习率衰减策略结果的准确率进一步提高，高出了 1%的准确率。

3.5.4 改进的 Alex_ViT 与原始 ViT 结果对比分析

表 3.2 实验结果及对比

模型	准确度	损失值	训练时间(s)
ViT	0.758	0.544	305
Alex_ViT	0.848	0.200	332
Resnet	0.881	0.220	57
Alexnet	0.844	0.352	30

在表 3.2 中所示，由于改进实验之后并没有预训练模型，所以四种模型比较是在无预训练模型下进行训练，且均使用余弦学习率衰减策略，从实验对比中发

现经过加深 Patching Embedding 层中的卷积神经网络，ViT 准确率大大提升，上升了 9%，而且相较于 Alexnet 模型准确率也有略微提升，但是依然低于 Resnet 模型，这是由于模型的大体结构依然是 Vision Transformer 结构，所以在小规模数据集上的表现力还是有所欠缺，但是已经高于融合的这两种模型的准确度，但是训练的损失下降非常明显，在四个模型中处于最低。但是性能方面，随着卷积层数的增加，参数量的增加，训练时长增加。

3.5.5 预测结果对比

预测的 5 张图片如图 3.9 所示。

图 3.9 预测图片展示，

从左至右依次为雏菊、蒲公英、玫瑰、向日葵、郁金香

预测结果展示在表 3.3 中，从预测结果的对比中也可以看出，改进的 Alex_ViT 的整体预测准确率也是最高。

表 3.3 预测结果对比

模型	雏菊(%)	蒲公英(%)	玫瑰(%)	向日葵(%)	郁金香(%)
ViT	91.44	65.55	81.88	96.91	99.56
Alex_ViT	99.61	100	96.13	99.64	99.99
Resnet	99.76	100	95.99	99.97	99.93
Alexnet	99.96	99.74	85.95	93.65	100

第四章 总结

第一节 实验总结

本实验是基于 Vision Transformer 的花卉图像分类工作。

本次实验工作的重点是 ViT 模型的实现，本文详细介绍了 ViT 模型的基础 Transformer 模型中自注意力机制和多头注意力机制，以及 ViT 模型的内部框架，分别介绍了 embedding 层、transformer encode 层和 MLP Head 的功能和理论基础，并在最后介绍了轮数学习率衰减和 cosine 学习率衰减策略下对于训练的速度和验证集准确性的影响，得出轮数学习率衰减策略相较于余弦学习率衰减策略对于准确率的提升更有帮助的结论。在实验的最后一部分，通过研究 Alexnet 和 ViT 最终改进的 Alex_ViT 模型，将模型中的一层卷积网络替换成改进的 Alexnet 网络，改进后的混合模型准确率大大提升。

本次实验结果的对比与分析主要是以分类的准确率作为主要指标，评价这四种模型的实验效果。而且实验中还使用了预训练模型，使得实验结果更上一层，而且对比分析可以从更多方面进行比较观察 Alex_ViT 模型的优势。

第二节 实验的不足

在做本次实验的时候还有许多不足方面，首先是实验环境 GPU 的性能不能支持对实验进行过多次的训练循环，所以实验结果可能并未寻找到最佳结果。其次由于 ViT 并不能在小数据集上取得比较好的结果，但由于时间问题并未尝试或搜集如何在小数据集上改进 ViT 从而达到想要的结果。对于数据集方面，没有使用大规模数据集直接训练模型，所以未得到在大规模数据集下的准确率关系。

参考文献

[1] Kitaev N, Kaiser Ł, Levskaya A. Reformer: The efficient transformer[J]. arXiv preprint arXiv:2001.04451, 2020

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

[3] Liu Y, Zhang Y, Wang Y, *et al*. A survey of visual transformers[J]. the Institute of Electrical and Electronics Engineers Transactions on Neural Networks and Learning Systems, 2023

[4] T.Fushiki." Estimation of prediction error by using K-fold cross-validation." Statistics and Computing, 2011, 21: 137-146.

[5] He Tong, Zhang Zhi, Zhang Hang, et al. " Bag of Tricks for Image Classification with Convolutional Neural Networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Dec.2018.

[6] M.Maaz, A.Shaker, H.Cholakkal, et al. "Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications."Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII. Cham: Springer Nature Switzerland, 2023: 3-20.