# 绪论

## 课题背景和意义

鸟类识别技术在很多应用场景中都是至关重要的。特别是如何确定一只鸟所属的类别，这可能是很多工作进行下去的前提，比如环境保护、通过观察某一鸟类的数量变化及时保护和救援一些濒危鸟类、通过对不同种类鸟类数量的分布情况来观察气候变化、及时保护环境以及生态稳定性。这些场景对于实际的生产和生活都有十分重要意义。

在图像识别应用于鸟类识别领域以前，这个工作通常是由专门的工作人员去完成，这对工作人员的专业知识具有较高的要求，另外在能见度较低以及距离较远等情况下，专人通过观察的方法确定鸟类的类别可能就会变得无能为力，并且依赖于专人进行识别是一项较为费时费力的工作，在有些情况下由于各方面条件的限制可能会十分难以进行。

近年来，深度学习取得了十分飞速的发展，卷积神经网络提出后，相比于以往使用全连接层进行图像识别正确率有了大幅度的提升，促进图像识别也取得了很多突破性的进展。因此可以使用深度学习的方式确定某一只鸟所属的类别，由于现在拍照设备可以拍出照片的分辨率远超人类肉眼可以达到的水平，以及现在计算机硬件能力的飞速提升，计算机可以以非常快的速度对于鸟类进行识别，并且识别能力相比于人类肉眼识别具有非常大的提升。相比于专人对鸟类进行识别，计算机识别的准确率相对来说更高且更稳定，另外，计算机也会降低鸟类识别的成本，可以将鸟类识别技术应用于更加广泛的领域。基于这样的背景下，本文实现了一个基于子类与深度学习的鸟类细粒度识别网络模型，同时基于该模型搭建了一个鸟类细粒度识别网站。

## 研究难点分析

与传统的粗粒度图像识别相比，细粒度图像识别的难度更高。细粒度图像识别主要会面临拍摄因素的影响、较大的类内差异以及较小的类间差异、数据标记较为困难等几个方面的挑战：

拍摄因素的影响。细粒度图像识别在样本拍摄的过程中，光照条件以及最终产生的图像的分辨率都可能对最终的识别造成较大的影响。拍摄时光照条件一个比较大的影响是可能使不同物体原本较为明显的颜色特征变得不再明显，最终会增加模型的训练难度。另外，如果最终输出的图像分辨率过小，那么也可能造成比较明显的用于区分的外观特征因为过于模糊而不能分辨，而如果图像分辨率过高，那么可能会增加后期系统使用时的拍摄难度，较难采集到可以用该系统的数据。所以拍摄时各方面的因素会对细粒度图像识别的结果造成影响。

较大的类内差异以及较小的类间差异。也即同一类别不同个体可能会有较大的外观特征差异，而所有类别由于同属于一个大的类别，所以具有比较大的类间相似性，不同类别的特征则可能只能通过较不明显的特征才能进行区分。有些属于相同类别的性别不同的样本也可能会有非常大的外观特征的差异，有些类别的个体可能会在不同年龄段会发生较大的外观特征的变化，最终进一步增加了网络模型训练的难度。同时，有些属于不同类别的个体外观差异较小，可能只能通过一些比较细微的差别加以区分，但是拍摄时个体的状态、环境因素的变化以及拍摄位置的变化都可能会导致不能观察到用于区分这些类别的特征，从而导致不能判断该样本属于哪个类别。细粒度图像识别的这个特点导致了细粒度图像识别的难度相比于粗粒度图像识别难度更高。

数据标记困难。相比于粗粒度图像，细粒度图像的标记要更加困难。细粒度图像的较大的类内差异以及较小的类间差异，增加了人工识别样本类别的难度，有些样本甚至相关领域的专家都难以做出判断。另外有些细粒度识别的算法不仅需要确定样本所属的类别，还需要圈出样本中主体所在的区域，有的甚至还需要标记出训练与验证集中主体的头、身体、尾巴等各部位的区域，如果不标记相关的区域，则可能比较难以达到期望的、可以用于实际生活生产的准确率要求，从而进一步增加了数据标注工作的难度。这些标注工作需要较高的时间成本和经济成本，且有时候根本不可能实现。因此细粒度识别的难度要比粗粒度图像识别的难度更高。

因为以上的各方面因素的影响，所以相比于粗粒度图像识别来说，细粒度图像识别要更加困难。

## 国内外研究现状

下面将分别介绍卷积神经网络研究现状和细粒度图像识别研究现状。

## 卷积神经网络研究现状

卷积神经网络是一种从生物的视觉感知机制获得灵感进而发展而来的一种广泛用于图像识别的神经网络结构。1980年，Kunihiko Fukushima在[1]中提出了一种不受位置变化影响的模式识别机制的自组织神经网络模型。之后，Y.LeCun在[2]中正式提出了一种卷积神经网络LeNet-5，该模型用于处理手写数字识别问题。但是后面由于硬件条件的限制，卷积神经网络的发展一直较为缓慢，直到A. Krizhevsky等人在[3]中提出了一种相比于LeNet-5层次更深的网络结构AlexNet，该网络在ImageNet数据集上取得了相当优异的成绩，top-5错误率仅为15.3%，比第二名低了10.8个百分点。该模型由五个卷积层、两个全连接层、一个softmax层组成，用于预测类别。该模型通过学习早期卷积层中的底层特征并将其积累为后裔卷积层中的高层语义特征。其中该论文一个重要的贡献是证明了模型的深度对于提高性能、提高模型准确率是至关重要的。由于GPU的使用，使得深层次网络的训练和多次迭代成为了可能。之后Christian Szegedy在[4]中提出了一种全新的网络结构，该网络使用1x1的卷积核来升维和降维，同时在一个网络层上使用多个不同尺寸的卷积核同时做卷积操作，避免了网络深度过大造成的过拟合以及梯度弥散、梯度爆炸等问题。进一步提升了卷积神经网络图像识别的能力。2015年，微软亚洲研究院的Kaiming He等四位华人提出了一种ResNet模型[5]（又被称为残差网络），该网络的核心是残差结构，这种残差结构允许保留之前网络层中一定比例的输出，甚至可以直接跳过一些网络层，直接学习上一个网络层的残差，并且该模型具有较好的推广性。相比于其他的网络结构，该模型能有非常大的性能提升。

## 细粒度图像识别研究现状

过去细粒度图像识别主要关注于使用局部检测提高准确率，使用这种方式可以减少类内变化对识别结果的影响。X. Wei等人在[6]中提出了一种基于局部检测的卷积网络模型Mask-CNN，该模型首先需要首先分别人工标注出训练集中样本的头和身体所在的矩形框。之后把整个图像、头部区域、身体部分所在区域分别传入三个相应的卷积神经网络进行特征提取与分类，之后把这三个网络对应的输出进行拼接并进行最终的分类操作。在训练上述三个神经网络的同时，该模型还会将模型的头部和身体等信息传入另一个神经网络进行训练。训练该网络可以对于一个给定的图像可以正确地标注出鸟的头部和身体区域。在测试的时候首先使用神经网络获得测试集鸟类的头部和身体区域，之后再将这三个部分一起传入神经网络从而得到模型最终的预测结果。之后将不断训练优化网络模型提高模型准确率。使用该模型可以实现比较高的准确率提升。也证明了基于局部检测的网络模型在细粒度图像识别方面具有较好的效果。但是，尽管使用这种局部检测模型在测试的过程中可以完全自动地进行圈出鸟类的局部区域以及训练分类网络等操作，但是在训练的过程中，需要人工标注出测试集中所有样本局部区域所在位置，比如头或者身体等的区域位置，从而训练用来标记区域的神经网络。而人工标注所有训练集数据工作量十分巨大，较为耗时耗力。

为了避免代价比较大的人工标记工作，最近有一些研究人员在探索不用局部标记的办法进行细粒度图像识别。Zhang等人在[7]中的研究可以看到，即使不使用人工标注的局部区域的信息，在细粒度图像识别方面，卷积神经网络仍然可以表现出较好的性能。

一些研究人员在神经网络的背景下尝试学习K个分类器并使用这K个分类器进行分类。1991年Jacobs等人在[8]中提出了一种门控网络结构。这种网络结构学习K个专家分类器并将其用于多演讲人元音识别系统，这种网络结构的基本思想是仅仅将输入样本的一小部分分配给一个特定的分类器。这种方式通过训练K个神经网络用来做分类，并且在训练的过程中训练一个门控网络，由这个门控网络决定将样本具体分给哪个样本进行训练。在测试的过程中，门控网络将会决定每个分类器的分类结果所占据的权重，从而将每个分类器的分类结果进行加权求和得到模型的最终预测结果。之后将不断训练优化网络模型。但是这个网络模型的一个问题是过于依赖门控网络的分类结果，同时特征提取和分类工作被分开进行了。

## 本文的主要研究内容

基于细粒度图像识别的类间差异性小、类内差异性大以及数据量比较小的特点，本文实现了一个基于子类与深度学习的鸟类细粒度识别网络模型。该模型将会分别训练K个分类器，每个分类器训练一个子类，从而可以在一定程度上解决细粒度识别的类间差异小、类内差异大的特点。同时针对数据量小的这一难点，在训练过程中使用了数据增强、迁移学习等方式避免过拟合。另外与基于局部检测进行细粒度识别不同的是，该模型在训练的过程中不用人工标注鸟的各部分所在的区域，只需要标注鸟所属的类别，减少了数据标注的工作量。

本文选择了鸟类细粒度识别的基准数据集CUB-200-2011作为本文中实现的模型的训练和测试的数据集。在训练和测试之前首先使用数据集中每个样本的边界框信息取出每个样本中鸟类所在的区域。另外，由于模型的数据量较小，为了避免模型的过拟合，本文使用了数据增强技术相对地增加了数据集的规模，此外，还使用了Dropout、权值衰减等技术进一步避免了模型的过拟合。该网络模型在GPU运算平台上基于Pytorch、sklearn等技术实现。该网络模型首先提取出训练样本的特征。之后将会根据这些特征将训练样本分成K个不同的且互不重叠的子类。之后将为每个子类分别创建一个分类器。使用每个子类分开地训练每个子分类器。在将每个子分类器训练到一个比较好的程度之后，模型将会把每个子分类器的结果进行加权求和作为模型对于某一样本的最终预测结果。之后将训练优化结合在一起的K个分类器。在训练与优化的过程中，本文尝试在迁移学习时对于不同的层设置不同的学习率，从而使模型可以达到较好的训练效果。另外本文尝试了在训练过程中在模型的loss达到不同值时将K个网络开始合在一起的训练，比较验证集的准确率，从而可以进一步提高模型的准确率。在CUB-200-2011数据集上，基于不同的经典CNN模型来实现论文中的模型进行迁移学习可以获得最高4.3%的相对准确率的提升。

同时本文基于该模型实现了一个鸟类细粒度识别网站，该网站可以识别鸟类所属的种类以及收藏鸟类的照片。

## 论文的组织结构

针对细粒度图像的国内外研究现状，同时结合相关研究的背景和难点，本文实现了一个基于子类与深度学习的鸟类细粒度识别模型。本文选择了数据集并对数据进行了预处理，同时选择了几个经典网络模型实现了该细粒度识别模型，同时对模型的结果进行了测试与评估。论文的组织结构可以分为：

第一章：绪论。本章主要介绍了细粒度图像识别的背景和意义，之后分析了细粒度图像识别的难点，之后则介绍了卷积神经网络、细粒度图像识别研究的发展现状，最后介绍了本文的主要研究内容，也即本文实现的基于子类与细粒度识别的鸟类细粒度图像识别模型。

第二章：深度学习与机器学习的相关理论。本章介绍了在本文实验中使用到一些深度学习与机器学习方面的概念和算法。介绍的深度学习中的概念主要包括卷积神经网络的概念和组成部分、过拟合的概念和解决办法、动量的概念及应用场景以及本次使用用到的几个经典的CNN网络模型。介绍的机器学习的算法主要包括K均值聚类算法和LDA降维算法，他们分别被用来将样本聚类成K个子类以及将提取出来的样本进行降维操作。

第三章：基于子类与深度学习的鸟类细粒度识别模型。本章则首先介绍了本次实验中使用到的一个比较重要的技术，即迁移学习。之后详细介绍了本次实验实现的基于子类与深度学习的鸟类细粒度识别的模型结构。该模型主要包括训练集的聚类与预训练网络和分类网络两个部分。

第四章：模型的训练与优化。本章首先介绍了实验的一些设置，包括数据集的选择以及数据集的预处理操作、本次实验使用的计算平台以及软件环境、以及模型的训练过程以及模型训练过程中遇到的一些问题以及解决办法。训练过程包括聚类过程以及网络模型的训练与优化两部分。最后则展示了基于不同的经典CNN模型训练该网络模型的训练结果，并对训练结果进行了总结。

第五章：鸟类细粒度识别系统。这一章详细介绍了基于该模型的鸟类细粒度识别网站的分析、设计与实现过程。分析过程主要包括系统的需求分析、业务流程需求分析、数据流程需求分析、非功能性需求等。设计过程则主要包括系统的整体设计、模块功能设计、数据库设计等。实现过程则主要展示了该网站的效果图。

第六章：总结与展望。这一章对本文以及本次论文所进行的实验进行了总结，并对未来可能的改进方向进行了展望。

# 深度学习与机器学习相关理论

## 深度学习

下文将介绍卷积神经网络的基本结构、在深度学习中较容易出现的过拟合问题以及本次实验中用到的解决方案，最后将介绍本次实验用到的几种常用的经典CNN网络模型。

### 卷积神经网络

卷积神经网络是使用数学上的卷积运算的一种网络结构。卷积是一种特殊的线性运算。卷积神经网络是在至少一层中使用卷积代替一般矩阵的简化的一种神经网络。卷积神经网络将一张图片作为网络的输入，然后图片将会经历一系列的卷积层、池化层以及全连接层。并将最后的结果使用Softmax函数映射到0到1的区间作为图片属于相应类别的概率。同时，应该在卷积层和池化层中间添加非线性单元来增强模型的表达能力。

卷积层是用来从一个输入图像中提取特征的层，它通过使用输入图像的一小块区域来提取特征从而来提取输入图像的像素点之间的关系，卷积层对输入图像和卷积核做数学上的卷积操作来实现输入图像特征的提取。

当图像过大时模型需要使用池化层来减小参数的数量。池化也被成为下采样，它在减小样本大小的同时仍然保留了样本中的重要信息。池化操作分为三种类型，分别是最大池化层、平均池化层以及和池化层，它们分别将待池化区域的最大值，所有值的平均值，所有值的和作为该次池化操作的结果。

如果仅仅使用卷积层和池化层对输入图像进行处理，最终模型的输入和输出也仅仅是进行线性了操作。会造成模型的表达能力不足。为了避免这个问题，需要在神经网络中加入非线性单元，也即激活函数，现实世界的问题一般也都是需要非线性模型才能做较好的拟合。在卷积神经网络中常用的激活函数是ReLU函数。ReLU函数的输出值是0和输入值的最大值。

在模型进行完卷积和池化的相应操作时，需要将最终输出的矩阵首先展平成向量，之后把它输入一个全连接层。然后模型将使用全连接层将提取出的特征结合到一起，之后需要使用一个激活函数来确定神经网络最终预测的图片所属的类别。这个激活函数可以是Softmax函数或者Sigmoid函数等。

### 过拟合

过拟合（也即overfitting）是统计学中的一个概念，它指的是过于紧密地甚至精确地匹配一个特定的数据集合，并可能因此不能较好地适应额外的数据或者预测未来地情况。在深度学习中，过拟合指的是模型对于训练集的拟合非常完美，能取得非常高的准确率，但是将模型较好地应用于新的、之前没有见过的数据，对于验证集和测试集的数据不能很好地预测。可以通过观察训练集和测试集的衡量标准来确定模型是否发生了过拟合，这些衡量标准包括准确率和损失函数等。如果模型的验证集在经过一定轮的迭代之后，验证集的衡量标准没有提升甚至开始退化，而训练集的衡量标准继续提升，那说明模型发生了过拟合。此时需要及时停止训练并保存模型的状态。由于细粒度识别样本相对较少，比较容易发生过拟合，因此在实验的过程中应尽量避免发生过拟合。避免模型的过拟合最好的方法是获取更多的训练数据，但是获取更多数据是一项费时费力的工作，并且在有些情况下根本不现实。此时就需要其他的技术来避免模型的过拟合。这些技术包括数据增强、权值衰减、Dropout等。

数据增强是深度学习中一种非常常用的技术，其主要用于提高模型的预测准确率以及防止过拟合等。在某些时候训练集的规模过小时，通过数据增强可以相对地增加训练集的规模，从而可以在一定程度上避免过拟合。常用的数据增强方式有向原始图像添加噪点、对原始图像进行缩放、裁剪、旋转、反转等操作，从而提高模型的泛化能力。由于细粒度识别的图像数据集相对较小，因此需要使用数据增强的技术。

Dropout是另一项常用的用来避免模型过拟合的技术，主要用在全连接层中。Dropout的意思是在模型的每一轮训练的过程中，会随机选择性忽视一些神经元，也即在前向传播和后向传播的时候不考虑这些神经元结构。由于一个全连接层使用了大量的参数，同时，神经元在训练的过程中互相依赖，从而抑制了每个神经元的个体能力，进而导致了模型的过拟合。加入过拟合之后则会避免神经元之间的互相依赖，从而可以很大程度避免过拟合的发生。

假如搭建的网络模型刚好符合要求，则模型能够恰好拟合数据，但是一旦搭建的网络模型复杂度过高，就很有可能造成模型对于训练集数据的过拟合。这就意味着，为了避免过拟合的发生，应该避免使用的网络模型过于复杂。但是这样的一个误区是使用较浅层的网络，这样可能根本不能拟合样本。因为更多的参数意外着可以解决更加复杂的问题，如果使用的模型复杂度过低，那么该模型就很有可能不能拟合稍微复杂一点的问题。为了解决这个问题，可以对复杂的模型做一定的惩罚。这就意味着仍然可以使用较为复杂的模型，但是应该在损失函数中加入一个关于模型参数的函数。从而可以在一定程度上保证在模型优化的过程中会自动避免过于复杂的模型。一个较为常用的惩罚方式是L2正则化，使用此方法时会在原模型损失函数上添加一定的比例乘以所有参数的平方和。该方法可以较好地防止过拟合。

### 动量

SGD是神经网络中一个较为常见的优化器。SGD的中文全称是随机梯度下降，是一个迭代的，通过不断计算loss函数对于各个变量的导数从而更新各个变量的神经网络优化器。使用随机梯度下降的一个问题是到某个程度之后可能会造成了loss函数产生震荡，不能进一步逼近最小值，甚至可能会出现loss函数最终到达一个局部最小值或者鞍点之后便不再产生变化。从而导致模型的准确率也会徘徊在一个比较低的点不能继续增加。

动量（也即Momentum）在更新变量时会在本次计算的变化量的基础上添加一定比例的上一轮的变化量，从而在梯度方向不变时学习的速度会变得更快，同时在梯度方向相反时减慢变量的更新，进而可以加快loss函数的收敛以及减小震荡。

### 几个常用的CNN网络模型

本次实验中需要用到的几个经典CNN网络模型。分别使AlexNet模型、GoogLeNet模型、ResNet模型等。下面将逐个对他们做简单的介绍。

#### AlexNet模型

AlexNet模型是一个由Alex Krizhevsky设计完成的一个卷积神经网络，其结构如图 2‑1 AlexNet模型所示，可以看到AlexNet模型包含8层；前五层是卷积层，这些卷积层被用来提取图像特征，其中前两个卷积层以及第五层后面均使用了一个最大池化层。最后三层则是全连接层。该模型的输入是一个宽高均为256像素点的RGB图像，因此如果输入图像不是256 \* 256，需要在把它用于训练网络之前先进行转化，大于该尺寸需要进行裁剪，低于该尺寸需要使用插值等方法调整图片尺寸；对于灰度图则需要把单通道的数据复制为三通道数据。AlexNet使用ReLU函数作为激活函数，该激活函数与tanh和sigmoid函数相比能显著提升训练性能。

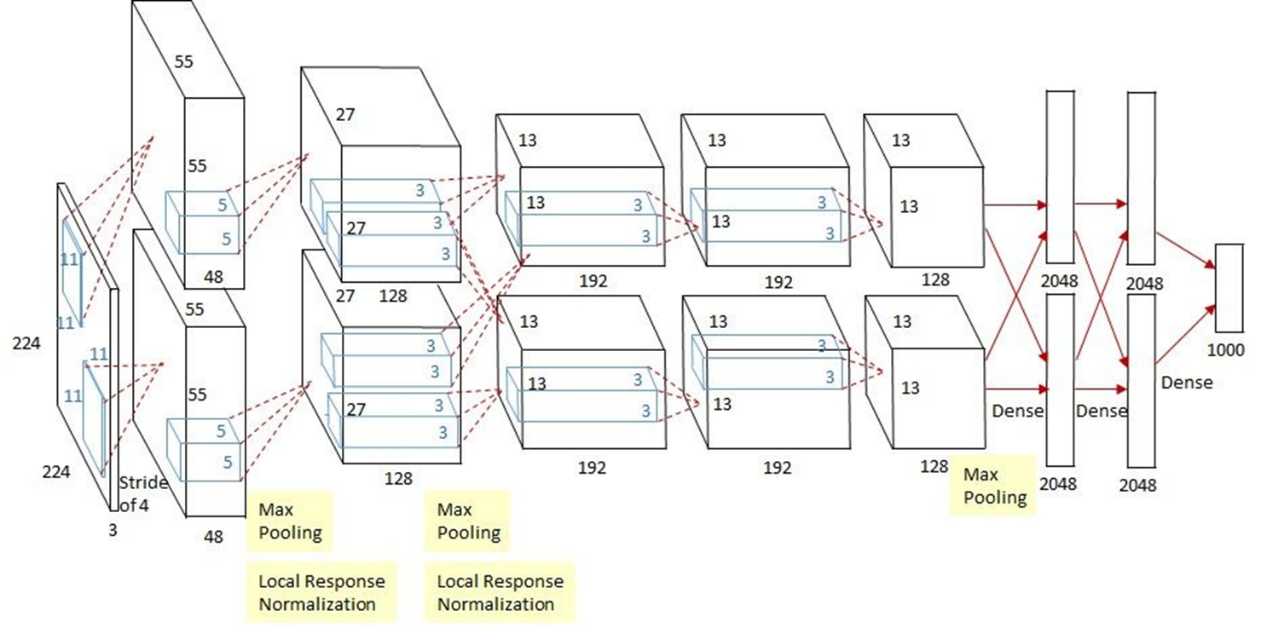


图 2‑二‑1 AlexNet模型

在ImageNet数据集上，AlexNet网络模型取得了相当优异的成绩。其top-5错误率仅为15.3%，比第二名低了10.8个百分点。原论文最重要的结果是模型的深度对于他的高性能是至关重要的。虽然较深的模型需要较高的运算能力，但是在训练过程中使用GPU使得这样的结构变得可行。

#### GoogLeNet模型

一般来说，提升网络性能最直接的办法就是增加网络的深度和宽度。但是这样也导致了模型的参数过多，一方面会造成计算复杂度较大，训练需要的内存较大且训练用时较长；另一方面会造成模型对训练集产生过拟合、以及造成梯度弥散的问题，计算梯度时后面的梯度非常小，模型难以优化等问题。

GoogleNet（又称Inception）是由Christian Szegedy提出的一种全新的深度学习结构，避免了上面的增加网络深度造成的问题。GoogLeNet的结构如图 2‑2 GoogLeNet结构所示。整个GoogLeNet结构就是由多个Inception模块串联起来的，Inception模块的结构如图 2‑3 Inception模块所示。Inception模块有两个特点：使用1x1的卷积核来升维和降维；在某一层上使用多个不同尺寸的卷积核同时进行卷积操作，再将卷积之后的结果进行聚合。使用1x1卷积核可以提取到更丰富的特征。在一层上使用不同尺寸的卷积核可以使得每一层提取到的特征更加丰富，进而可以使得模型最终的分类结果更加准确。

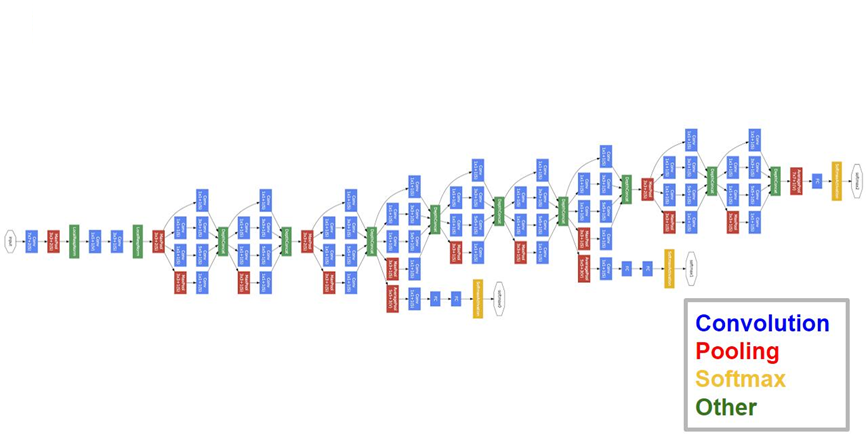


图 2‑二‑2 GoogLeNet结构

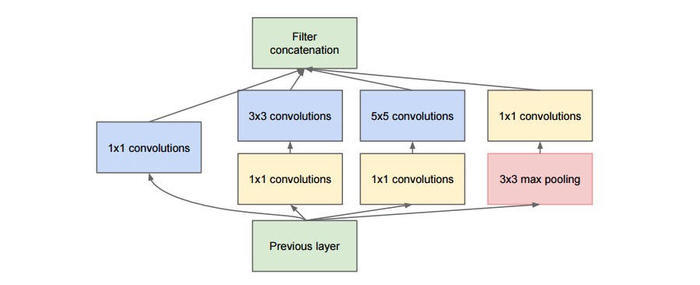


图 2‑二‑3 Inception模块

#### ResNet模型

从AlexNet模型开始，最先进的CNN模型的结构变得越来越深。但是，层数到达一定程度之后，仅仅将网络层堆叠在一起来增加网络的深度并不能带来准确率的提升了。由于梯度弥散问题，这样的比较深的网络很难进行训练。在梯度向后传递的过程中，过多的惩罚导致梯度逐渐变得无限小，从而比较靠前的网络层的参数很难获得更新。最终，随着网络变得更深，它的性能开始变得饱和甚至开始迅速下降。

ResNet又称为残差网络，是由微软研究院的Kaiming He等四位华人提出的一种卷积神经网络，其结构如图 2‑4 ResNet结构图所示。ResNet模型的结构可以加速神经网络的训练，同时可以较大幅度地提升模型地准确率。

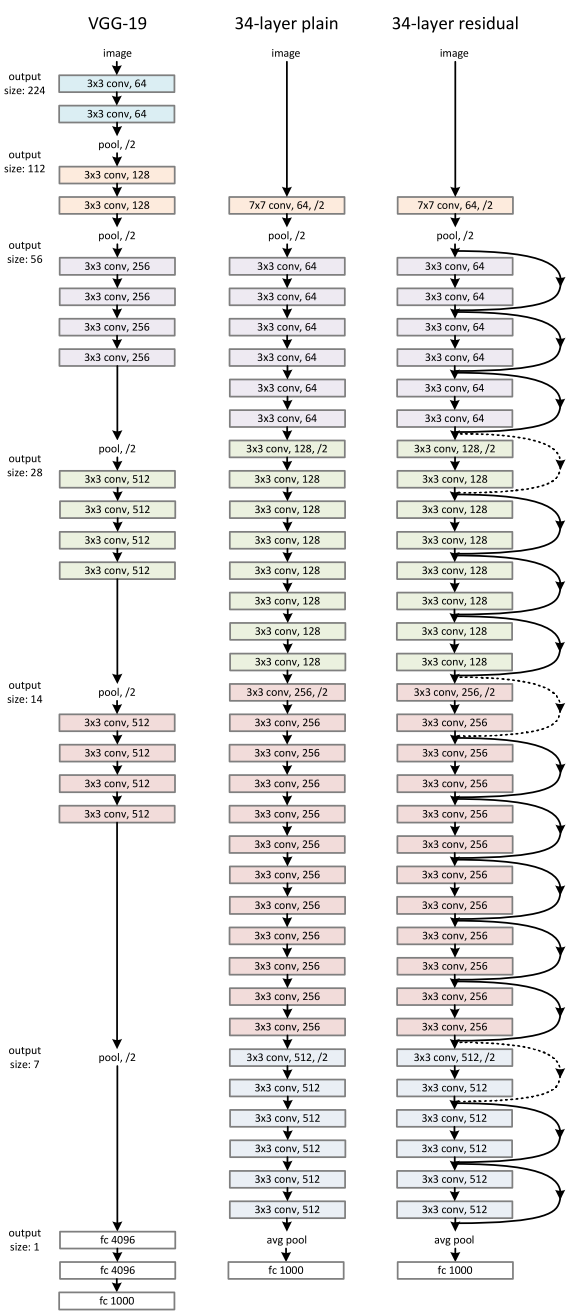


图 2‑二‑4 ResNet结构图

其核心是残差结构（如图 2‑5 一个残差块所示），这种残差结构允许保留之前网络层的一定比例的输出，甚至可以跳过一层或多层。这样的话每层的神经网络将直接学习上一个网络输出的残差值。

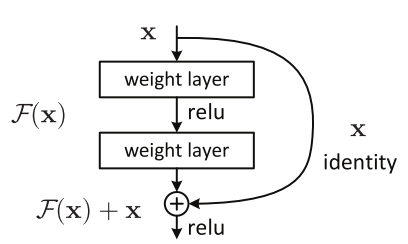


图 2‑二‑5 一个残差块

## 机器学习相关算法

本次实验使用的机器学习的相关算法有K均值聚类算法和LDA降维技术[9]，其中K均值聚类算法主要被用来根据CNN模型提取出来的特征将数据集中的所有样本分成K个互不重叠的子集。

## K均值聚类

聚类是一种最常见的探索性数据分析技术，用于对数据的结构有一个直观的认识。它可以定义为划分数据集中的子类的任务，即同一子类的数据点非常相似，而不同子类中的数据非常不同。与监督学习不同的是，聚类被认为是一种无监督学习方法。因为没有用来判断聚类算法性能的正确聚类标签。聚类主要别用来通过将数据点进行分组为不同的子组来研究数据的结构。

K均值聚类是数据科学中最常见的聚类算法之一。它是一种迭代算法。它尝试将数据集划分为K个互不重叠的子类，其中每个数据点只属于一个组。它试图使子类内的数据点尽可能地相似，同时尽可能保持子类之间的不同。它将数据点分配给一个子类，使数据点与子类型之间的距离的平方之和最小，集群内的变化最少，这样同意集群内的数据点也就越相似。

K均值聚类算法的步骤主要是：首先确定子类的数量K，随机选择K个数据点作为中心点，之后不断迭代进行下面的操作知道中心点不再发生变化为之：计算数据点和所有中心点之间距离的平方；将每个数据点分配给中心点距其最近的那个子类；通过计算属于每个子类的所有数据点的平均值来重新计算每个子类的中心点。

## LDA降维算法

LDA的中文全称是线性判别分析，是一种监督学习的降维技术，通常被用作机器学习以及模式分类应用等的一个预处理步骤。正如其名字所描述的那样，降维技术的主要目标是通过去掉多余的或者互相依赖的属性，从而将高维空间的特征转变为低维空间的特征来降低数据的维度，并尽可能保留更多的样本信息，从而尽量避免训练时出现过拟合，同时也减少了计算成本。LDA背后的思想是要将数据投影在低纬度，投影后希望每一种类别内的数据的投影点尽可能的接近，同时希望不同类别的数据的中心点之间的距离尽可能远。

进行LDA降维的步骤主要是首先计算不同类别的类间方差，也即不同类别平均值的距离，之后计算类内方差，也即计算每个类别的均值与该类别内的样本之间的距离。最后一步需要根据费雪准则构造低维空间，使类间的方差最大，类内的方差最小。

LDA与另一种机器学习中常用的降维技术PCA（中文全称是主成分分析）较为相似，但是与PCA不同的是：LDA是基于监督学习而PCA是基于无监督学习的，另外与PCA保存数据信息不同，LDA是为了使得降维后的数据点尽可能地容易被区分。

## 本章小结

本章对于实验所用到的深度学习与机器学习的相关概念以及算法。具体来说，首先介绍了深度学习中卷积神经网络的基本组成部分，以及训练神经网络中较容易出现的一个过拟合问题，并介绍了本次使用的几种避免过拟合的一些方法。之后则介绍了本次实验所使用的几个经典的CNN模型。后面介绍了实验中使用的机器学习相关算法，主要包括被用来对样本进行聚类的K均值聚类以及对从样本中提取的特征进行降维的LDA降维技术。后面将会使用这些算法完成细粒度图像模型的搭建。

# 基于子类与深度学习的鸟类细粒度识别模型

## 迁移学习

在训练CNN模型的过程中为了避免过拟合，一个可能的做法是获取尽可能多的数据。一个CNN模型通常会有数百万个参数，这使得这样的模型在数据集较小的情况下很难训练得到很好的结果。在细粒度识别问题上，由于标记样本工作较为耗时耗力，通常数据集规模相对较小。网络模型复杂度过低可能不能很好地拟合图像，而过高则可能造成模型的过拟合。因此，为了避免小数据集的问题，可以应用迁移学习[10]来使得模型能取得较好的训练效果。

迁移学习是机器学习中一个研究问题，它的核心是将在解决一个问题时所学到的知识存储起来，并将其应用到与该问题不同但相关的另一个问题上。使用迁移学习，重用以前学习任务中学到的信息将可能显著地提高目标网络地学习效率。因为细粒度图像识别与普通的图像识别有较大的相关性，所以，在所有实验中，均使用在ImageNet[11]上预训练过的网络模型，这些网络模型已进行了良好的初始化，后续的网络训练将会在这些网络上进行微调，并将测试结果与这些网络的测试结果进行比较。

在迁移学习的过程中需要给每个层不同的学习率。在最初训练过程中，如果给所有层相同的学习率，就会导致新添加的层的学习过程会对已训练好层的参数造成一定的破坏，从而会对族中模型训练的结果有一定的影响。在最初训练的过程中，只是简单地将分类模型所有层的学习率设置相同的值，没有考虑到迁移学习的过程中预训练好的层前期不需要那么高的学习率，最终使得分类模型的测试效果较差。后来通过查阅相关资料之后，终于明白了问题所在。于是在后续的训练过程中将预训练层的学习率先设置为0，而后面的自定义的层的学习率则为正常值。待模型的loss减小的幅度变得较小甚至没有变化的时候。再将预训练层的学习率设置为自定义层的学习率的0.1倍，直至最后待loss减小的幅度再次变得非常小时再将预训练层的学习率设置地跟自定义层一样。通过这样动态设置学习率的方式，模型取得了较好的预测效果。

## 细粒度识别网络模型

本文实现的网络模型通过数据分成K个不相关的子集以及为每个子类分别学习一个分类器来提高细粒度图像识别能力。在该模型中每个网络结构只需要区分总样本的一个子集，从而相对简化了每个模型需要学习的能力。这种网络模型的关键问题是输入的一个样本需要被分配给正确的网络。本文实现的网络模型是基于每个分类器对样本预测值的置信度来分配样本的。

## 训练集的聚类与预训练网络

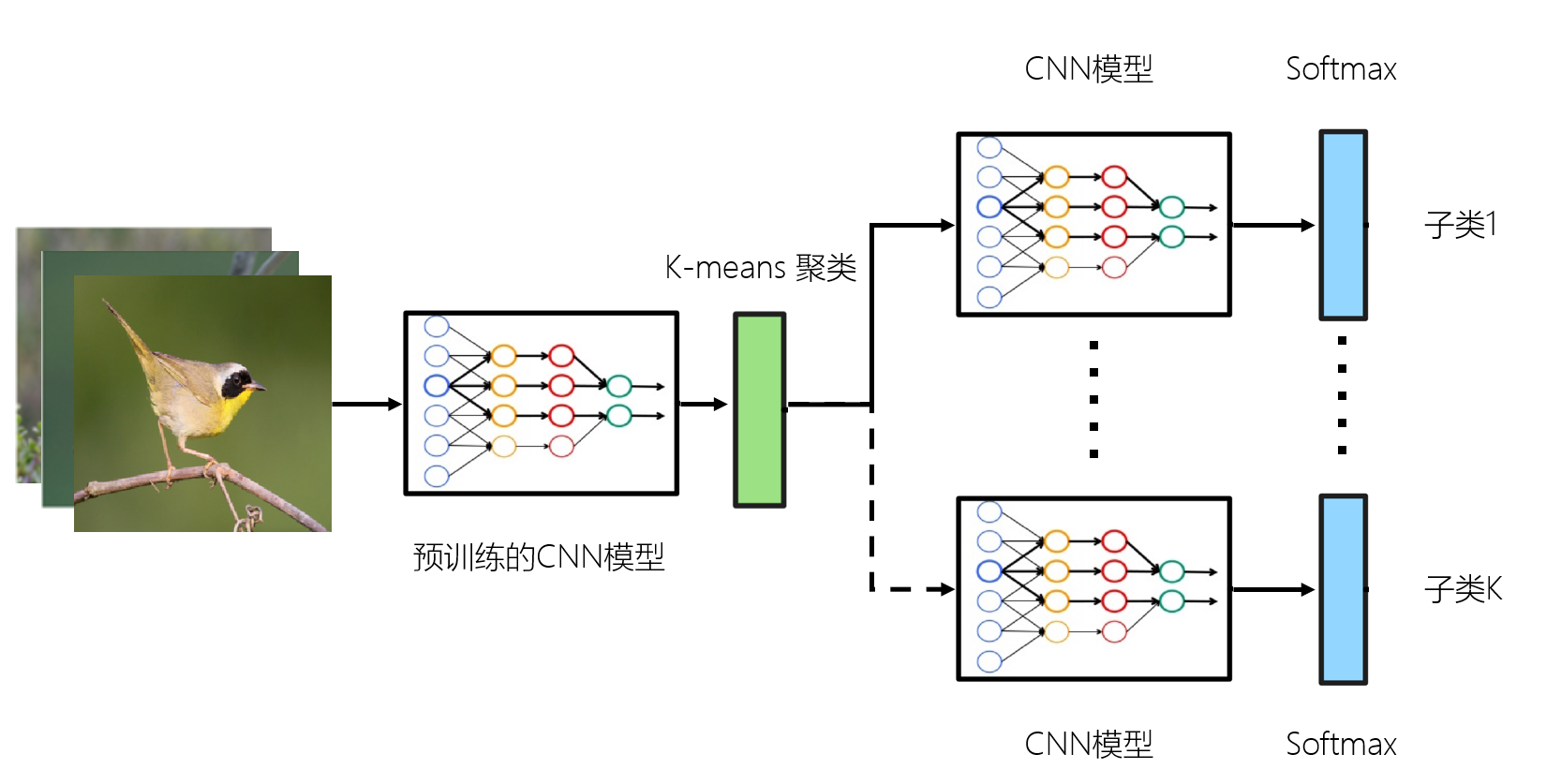


图 3‑三‑1 预训练网络模型

首先分开训练K个子分类器。如图 3‑1 预训练网络模型所示，本文首先将样本输入给一个在ImageNet数据集预训练好的CNN模型中，将模型的输出作为样本的特征，之后将根据提取出来的这些特征对训练集样本进行聚类操作，将训练集分成K个互不重叠的子类。之后将会对每一个子类分别训练一个CNN模型作为这个子类的分类器。在训练的过程中将会对每个分类器进行多次迭代训练，不断对其进行优化，直至其损失函数的值足够小且波动较小。对每个具有较高特征相似度的子类训练一个分类器避免了依赖一个CNN模型分类数据集中的所有样本，减少了神经网络的工作量，使每个模型可以更专注于如何识别每个子类中的样本，从而可以帮助神经网络达到更好的分类效果。

## 分类网络

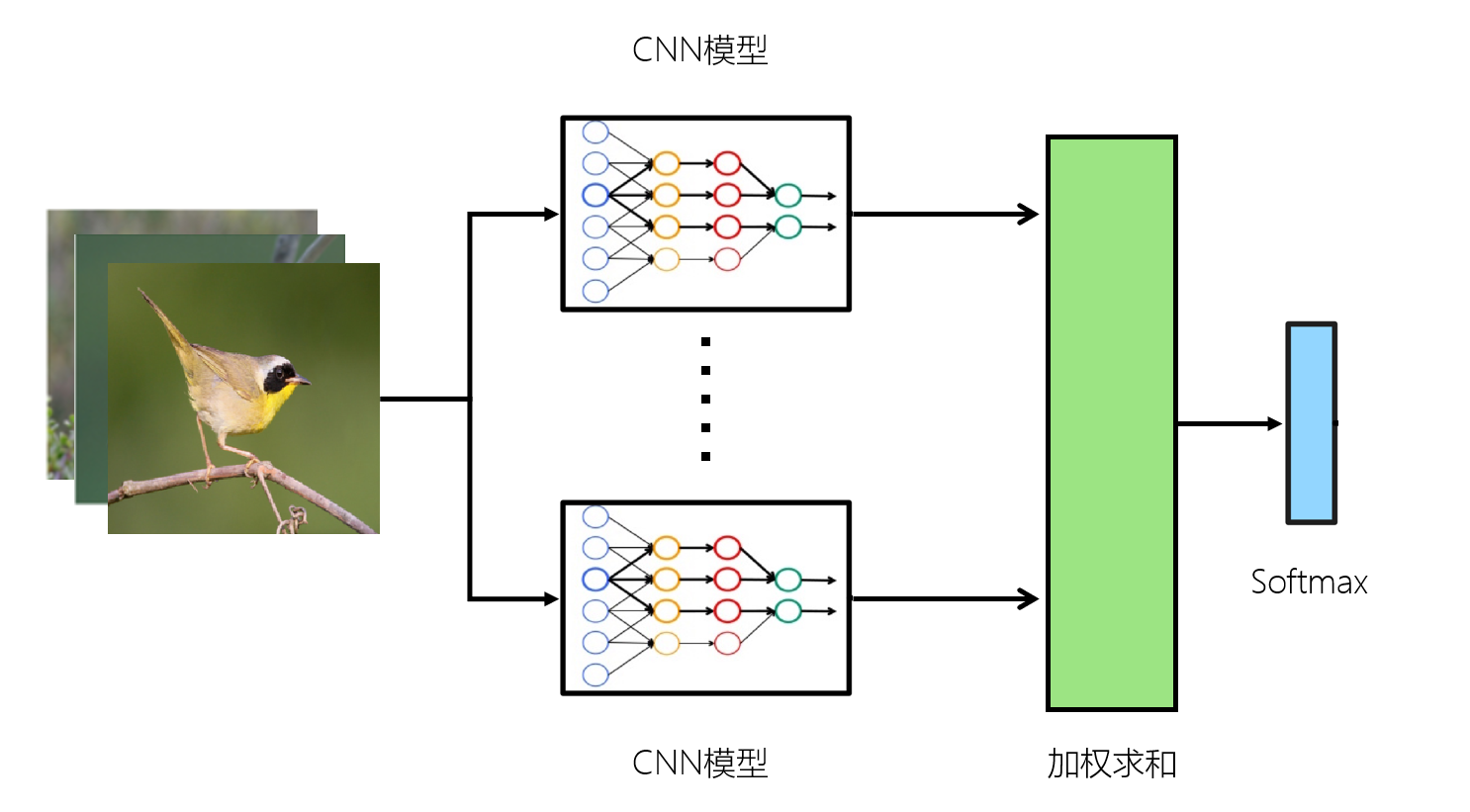


图 3‑三‑2 分类网络

在分别训练每个子网络完成之后，将开始训练模型最主要的分类网络。分类网络的结构如图 3‑2 分类网络所示。

在详细描述分类网络的结构之前，先定义一些表示。每个被训练用来分类的CNN模型的输出是一个N维的类别预测向量z，其中N表示的是类别总个数。表示第t个样本使用第k个子类对应的分类器得到的最好的分类结果，用公式可以表示为：

其中表示第t个样本在第k个分类器中属于类别n的预测值。

第k个分类器的分类结果置信度可以使用表示为：

=

样本t属于类别n的最终预测值可使用、表示为：

样本t属于类别n的概率可用表示为：

模型将分别每个训练样本和测试样本都传入K个子类对应的分类器中，得到样本t在分类器k中对于属于类别n的预测值。之后根据对这些预测值进行加权求和，结果为，其中加权求和的权值为每个分类器对应的分类结果置信度。之后将求出的结果结合到一起并进行softmax操作从而得到样本t属于类别n概率的大小。之后将使用交叉熵损失函数计算样本交叉熵，并使用随机梯度下降对模型地预测值进行不断优化，直到模型loss准确率波动很小甚至不再波动为止。

## 本章小结

本章对基于子类与深度学习的鸟类细粒度识别模型做了详细的介绍。首先本章介绍了迁移学习的概念。迁移学习是构建本细粒度识别模型使用到的非常重要的一个技术。之后分别介绍了本细粒度识别模型的两个部分：第一部分是训练集的聚类与预训练网络，在这一部分中模型首先使用一个预训练的模型提取样本特征，之后根据这些特征进行聚类，并使用聚类后的各个子类分别训练相应的分类器；第二部分是分类网络，在这一部分将各个子类对应的分类器的结果加权求和，并将求和的结果作为模型的最终预测值，对其进行不断优化。

# 模型的训练与优化

## 实验设置

## 数据集的选择与数据处理

细粒度识别的数据集的选择，要考虑到数据集的物种类别应足够多，每个物种类别分别包含的样本数量也应该足够大且不同种类的样本数量应该分布较为平均，否则将会对模型的训练结果造成一定的影响。同时，还要考虑到如果数据集含有边界框，可以使用边界框将样本中的主体提取出来，从而可以进一步提高模型的准确率。因此，本次网络模型的实现采用的数据集是Caltech-UCSD Birds-200-2011 Dataset[12]（以下简称CUB-200-2011）。

CUB-200-2011是一个用于细粒度图像分类的鸟类数据集，是CUB-200数据集的一个扩充版本。该数据集涵盖了200中鸟类，每种鸟类大概有30张训练样本和30张测试样本，总共包含11788个样本。同时，数据集中每个鸟类都可以从维基百科中查到其详细的类别信息。另外数据集中包含一个边界框信息文件，标记了每个样本鸟类所在区域的边界框，使用该信息提取出鸟类的主体所在区域将可以进一步提升模型的识别准确率。因此本网络模型采用该数据集。图 4‑1 CUB-200-2011数据集展示了数据集的样本示例。



图 4‑四‑1 CUB-200-2011数据集

在读取数据集时也读取了每个样本对应的鸟所在的区域矩形信息，并根据这些矩形信息对样本图片进行裁剪，使样本图片仅包含鸟所在的矩形区域。之后，使用数据增强技术对裁剪后的样本进行一定程度的缩放以及随机的裁剪、随机的旋转等操作，最后对数据进行归一化。除此之外实验过程中每次迭代会返回都会重新打乱样本的输入顺序，避免模型对于模型的过拟合，从而更有利于模型的学习。

## 计算平台

本次实验使用的计算平台使用的GPU平台。GPU的中文全称是图形处理器，是一种专门在个人电脑、移动设备上运行绘图运算工作的微处理器，较为适合用来做深度学习相关的研究。由于GPU的并行计算架构使得它可以同时执行大量运算，而一块单个的CPU却没有这样的能力，所以GPU相比于CPU可以更快处理图像。通常来说，GPU是为数据并行性以及将相同的操作应用于多个数据项而设计的，而CPU是为任务并行性以及执行不同的操作而设计的。本次使用GPU型号为GTX 1070 G3930，核心数为2，内存为16G，磁盘容量为2TB。

## 软件环境

本次实验是基于Pytorch进行的，Pytorch是一个开源的Python机器学习库，不同于Tensorflow，Pytorch是基于动态图的，另外Pytorch有两大特征：类似于Numpy的张量运算，并且可以使用GPU加速；可以自动求导，在用代码实现了神经网络的前向计算过程，Pytorch可以自动实现反向求导，不用手动编写。这对于搭建深度学习网络模型较为方便快捷。因此本文使用Pytorch来实现本次实验的网络结构。

同时由于本项目还用到了机器学习中的K均值聚类以及LDA降维技术等机器学习算法，由于scikit-learn对于这些算法做了较为完善的封装，并且是建立在NumPy、SciPy、matplotlib上的且是开源的，开源可商业使用，因此本文还使用了scikit-learn中相关函数实现相应的功能。

## 训练过程

模型的训练过程主要分为两部分，首先将训练集中的样本进行聚类操作，并为聚类之后的每个子类创建一个分类器。之后则是对于这些分类器进行训练和优化。

## 聚类过程

该模型需要首先使用一个预训练的CNN模型对测试集的样本进行特征提取工作。之后根据CNN模型提取出来的特征对训练集进行聚类。在聚类的过程中，模型应该关注鸟类的颜色和羽毛等特征，并且聚类的结果应该不受鸟类的姿态、背景、以及光照等外部因素变化的影响，对其应具有较好的鲁棒性。下面将会对两种聚类方法进行比较：

1. 只使用K均值进行聚类。图 4‑2 仅K均值聚类结果是直接对alexnet模型提取出来的样本特征进行K均值聚类的结果中随机抽出的部分样本。可以看到环境因素对于聚类的结果产生了较大的影响。子类4的共同特征是水面，子类5的背景都有树枝，而子类6背景都是偏纯色的背景，而在这些子类中鸟自身的特征相似性较为不明显。
2. 使用K均值加LDA降维技术进行聚类。图 4‑3 K均值加LDA聚类结果则是首先对alexnet模型提取出来的样本特征使用LDA技术对样本特征进行降维，之后再根据降维后的特征进行K均值聚类的结果中随机抽出的部分样本。聚类的结果相对于仅使用K均值聚类的结果有了一定的提升。可以看到，背景和鸟类姿态的因素对于聚类的结果相对影响较小，聚类更关注于鸟类本身的特征。甚至子类6中鸟的羽毛的颜色都是红色的，基本没有受到背景因素的干扰。

因此模型应该对使用CNN模型提取出来的样本特征先使用LDA技术进行降维操作，之后再进行K均值聚类操作，从而使得聚类可以取得更好的效果，进而帮助模型更好的学习该如何区分每个子类中不同的鸟类。



图 4‑四‑2 仅K均值聚类结果



图 4‑四‑3 K均值加LDA聚类结果

## 网络模型的训练和优化

在聚类完成之后，把聚类后的每个子类的训练样本输入到对应的子分类器中进行训练，在该部分的训练过程中，先固定预训练好的网络层，仅训练最后自定义添加的层，因为预训练的层已经达到一个较好的效果，暂时不应该改变这部分层的参数值。之后不断训练与优化自定义的网络层，直到模型的loss值达到一个非常小的值并且不再有较大的波动为止。从而为后续分类模型的训练与优化提供一个较好的初始状态，进而可以更加容易地达到更好地分类效果。

在前面的预训练网络模型训练完毕之后，将所有分类网络合在一起进行训练。在最初的那段时间，为了防止模型对训练集的过拟合，首先应该继续固定预训练的层，只训练后面的自定义层，因为这时候自定义层仍然有较大的训练优化空间。待loss波动较小时，再一起训练预训练好的网络层和自定义层，神经网络的每一层应该设置不同的学习率，这样可能更加容易达到较好的效果。

同时在整个模型的训练过程中，一直对随机梯度下降的优化器添加了一个权值衰减参数，大小一直是1e-4，用来避免模型对于训练集数据的过拟合。同时根据动态调整学习率的大小，等到loss的值波动较小的时候，把学习率降低为原来的0.1倍，一般最终模型学习率是最初时的0.01倍，从而能使得交叉熵损失函数能尽可能接近最小值。在训练的过程中，由于GPU内存等的限制，本文的实验将聚类的目标子类个数K的值设置为6。同时在聚类过程中使用的LDA降维的输出维度的值为128，相应实验表明：输出维度的值对于实验的最终结果没有较为明显的影响。

## 训练结果

在训练的过程中本文使用AlexNet、GoogLeNet、ResNet这三种网络模型，每种网络模型分别训练一个简单的迁移学习的网络以及一个基于该网络模型实现的新模型。这三种网络模型搭建的新模型都是取预训练模型最后一层输出作为聚类的输入，对于微调的迁移学习和基于这三个网络模型实现的新模型本文都是将网络的最后一个预训练的全连接层替换为新的全连接层。这两种模型分别记为CNN-tl和CNN-new。之后将会对每组模型中新搭建的网络与微调的迁移学习网络结果进行纵向比较，比较他们的差异。在不断进行训练优化之后，结果如表 4‑1 各模型测试准确率所示。可以看到相比于微调的迁移学习模型，本文构建的新模型最高有了4.3%的相对准确率提升。具体来说，其中基于AlexNet、GoogLeNet构建的新模型测试准确率相比于微调的迁移学习模型都有一定量的提升，而ResNet在经过多次训练优化以及调整超参数等多种尝试之后，准确率仍然较难有提升，甚至都没有达到微调的迁移学习的准确率，一个可能的原因是模型本身结构已比较完美，较难再做进一步的提升。

|  |  |  |
| --- | --- | --- |
|  | CNN-tl | CNN-new |
| AlexNet | 45.6% | 47.6% |
| GoogLeNet | 56.1% | 58.5% |
| ResNet | 77.8% | 76.6% |

表 4‑四‑1 各模型测试准确率

## 本章小结

本章首先介绍了本次实验所使用的鸟类细粒度识别数据集CUB-200-2011，以及对数据预处理过程中对数据进行的数据增强、归一化等一系列操作，之后介绍了搭建模型使用的计算平台以及软件环境等信息。之后则介绍了分类之前聚类的过程以及使用LDA与否对于聚类结果的影响。再之后介绍了模型训练过程以及各个超参数的设置和一些参数选择某一数值的原因。最后则介绍了使用各经典CNN模型实现新算法的测试准确率情况。

# 鸟类细粒度识别系统

## 鸟类细粒度识别系统的分析

### 系统需求分析

需求分析是软件开发周期最重要的阶段，其目的是从非正式的思想中提炼出“规格说明或规范”。在此期间，提出系统需要满足的功能和肺功能性需求，以及如何测量是否符合这些要求的准则或度量指标。

本系统用户角色分为管理员和用户两种。管理员是系统必不可少的一部分，主要负责系统的维护等工作，其角色功能包括系统的维护、数据库的维护、用户管理、用户收藏列表管理等。管理员的用例图如图 5‑1所示。

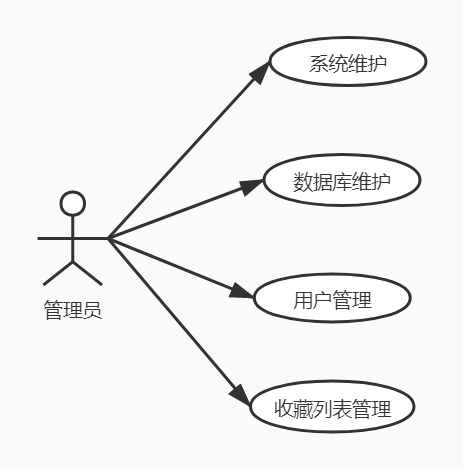


图 5‑五‑1 管理员用例图

用户可以上传鸟类图像进行识别、收藏上传的鸟类以便后续再次查看。因此本系统中用户的功能包括注册、登录、修改密码、收藏图片、查看和删除收藏夹内的图片、识别图片中的鸟的类别等。用户的角色用力如图 5‑2所示。

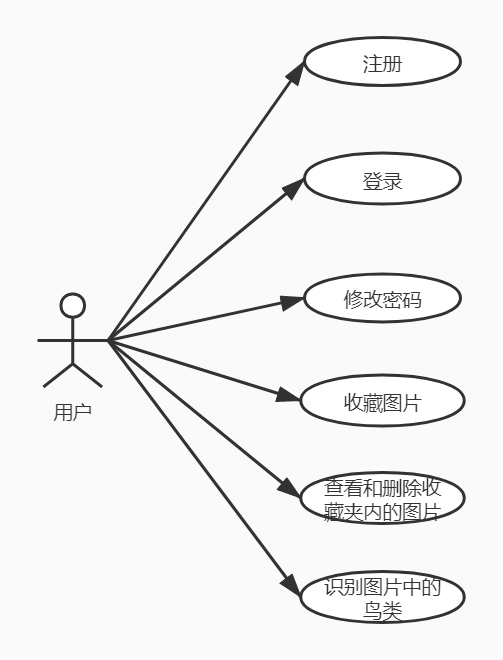


图 5‑五‑2 用户用例图

### 业务流程分析

#### 登录流程

用户和管理员的登录页面是一样的，流程图如图 5‑3所示。在登录成功之后，系统将判断登录者的身份并跳转至不同的页面。

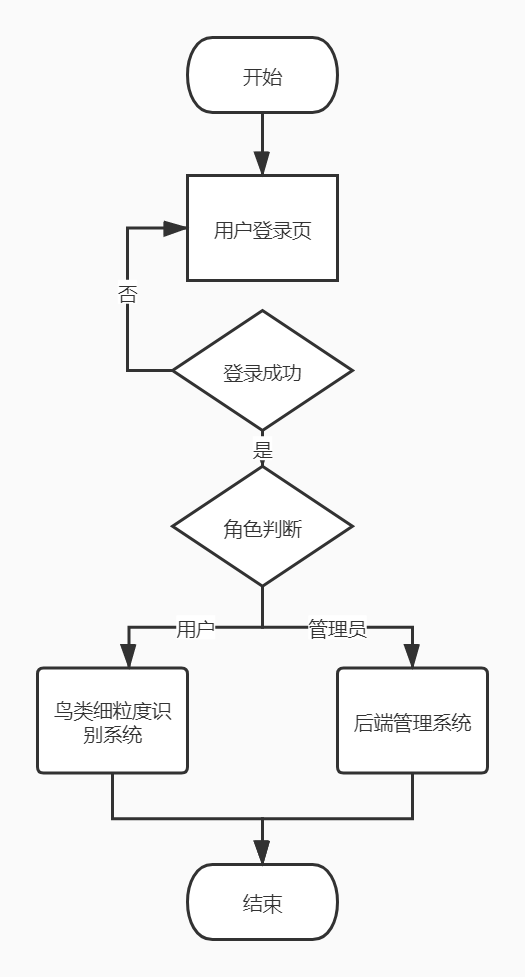


图 5‑五‑3 登录流程图

#### 图片收藏流程

用户可以从本地上传鸟类的照片并将其保存进收藏夹，其流程图如图 5‑4所示。

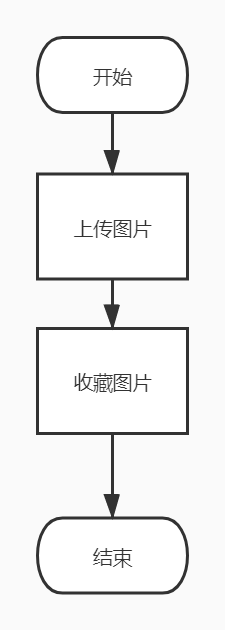


图 5‑五‑4 图片收藏流程图

#### 图片识别流程

用户可以通过两种方式选择图片进行识别，分别是本地上传图片和从收藏夹中选择图片，其流程图如图 5‑5所示。

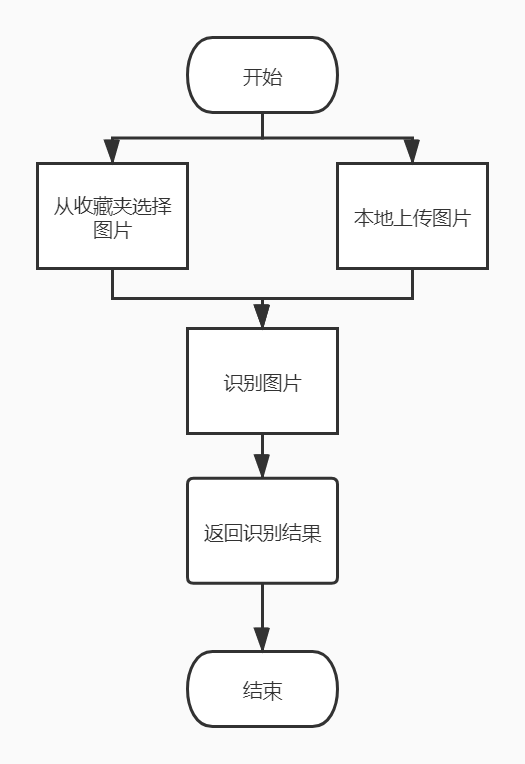


图 5‑五‑5 图片识别流程

### 数据流程分析

#### 图片的数据流图

图片的数据流图如图 5‑6所示。如图所示，用户可以收藏图片、浏览和删除收藏夹中的图片、上传图片、识别图片等，管理员可以查看和删除用户收藏的图片。

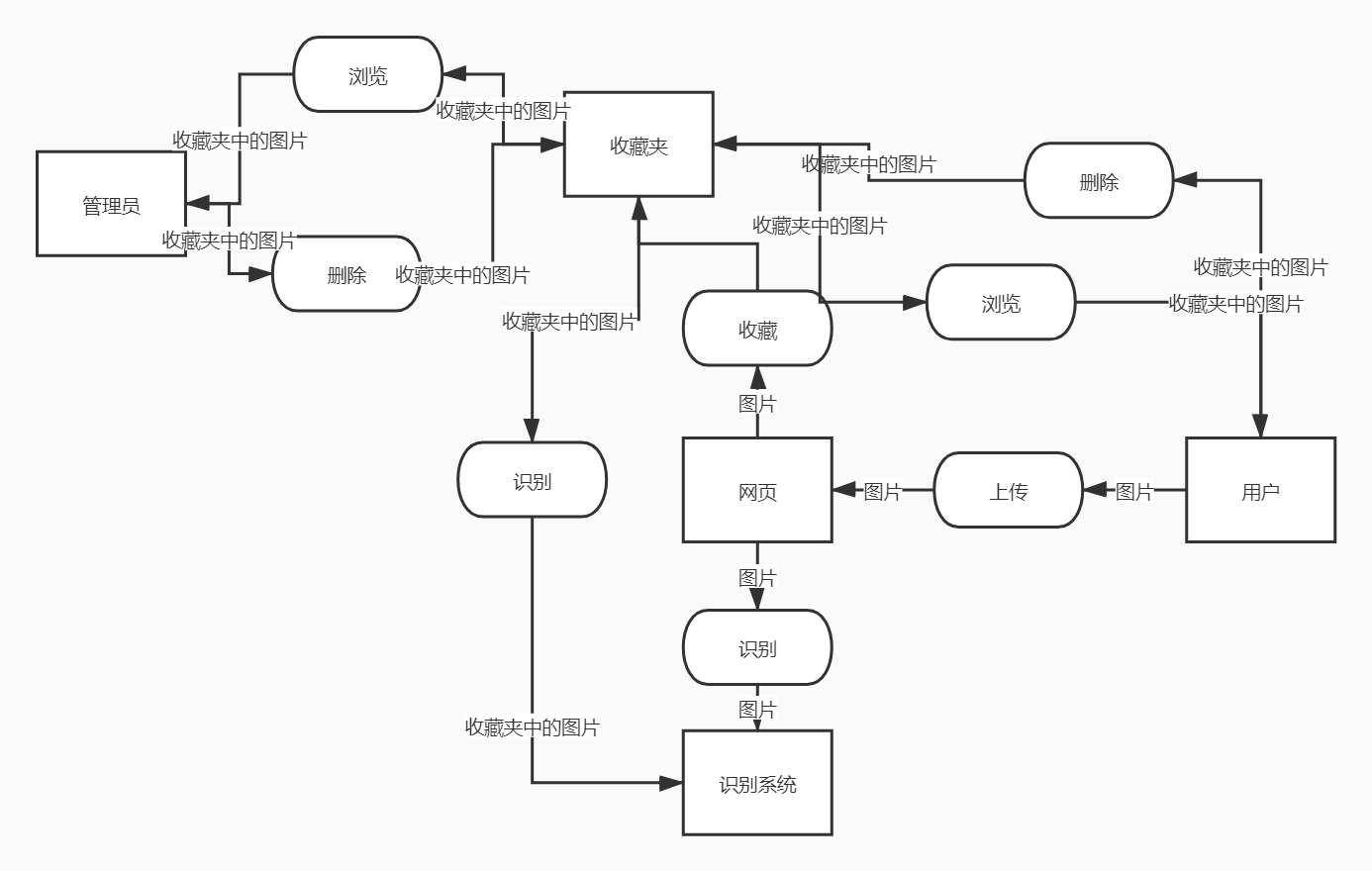


图 5‑五‑6 图片的数据流图

#### 用户信息的数据流图

用户可以通过注册创建用户信息，通过修改密码等修改用户信息。管理员可以查看和用户信息。用户信息的数据流图如图 5‑7所示。

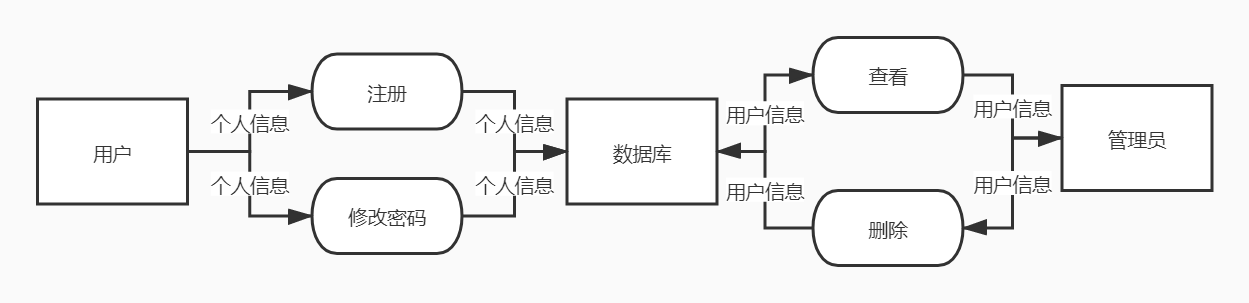


图 5‑五‑7 用户信息的数据流图

### 非功能性需求

非功能性需求是指以一些条件判断系统运作情形或其特性，而不是针对系统特定行为的需求。本系统要求进行鸟类识别的过程应不超过30秒。同时，系统应该可以支持100个用户并发进行鸟类识别的请求，可以支持1万个用户同时进行图片的收藏、查看、删除等操作。数据库的行数应该可以超过50万行。

### 鸟类细粒度识别系统的设计

### 系统整体设计

整个系统采用的是B/S架构，同时采用了前后端分离的设计。浏览器向前端服务器请求网页等信息，之后请求回来的网页再通过ajax向后端发起相应的请求。前后端之间通过Restful接口进行通信。后端分为控制层、服务层、数据库层三部分。其中控制层主要向外界提供API使得外界可以调用相应的方法。服务层主要是处理后端的业务逻辑等，而数据库层则主要处理与数据库的交互等逻辑。

该系统的功能模块主要包括用户管理模块和图片管理模块。其中用户管理模块分为浏览用户信息以及删除指定的用户，该模块只有管理员有权限进行访问和操作。图片管理模块则主要包含图片的收藏、收藏夹中图片的预览、收藏夹中图片的删除、识别收藏夹中的图片、从本地上传图片进行识别等。系统的功能模块图如图 5‑8所示。

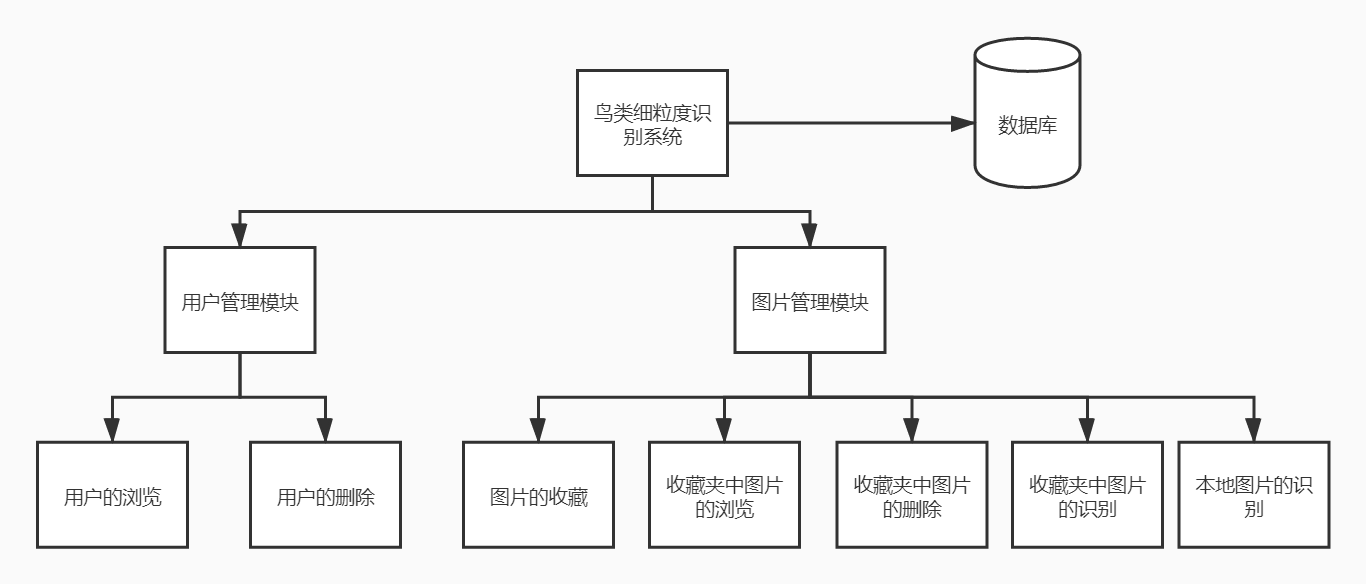


图 5‑五‑8 系统的功能模块图

### 模块功能设计

#### 用户管理模块设计

用户管理模块主要包括用户信息的浏览和用户的删除。其中用户信息的浏览将会按照用户的注册时间从早到晚显示已注册的用户。删除用户的流程图如图 5‑9所示。

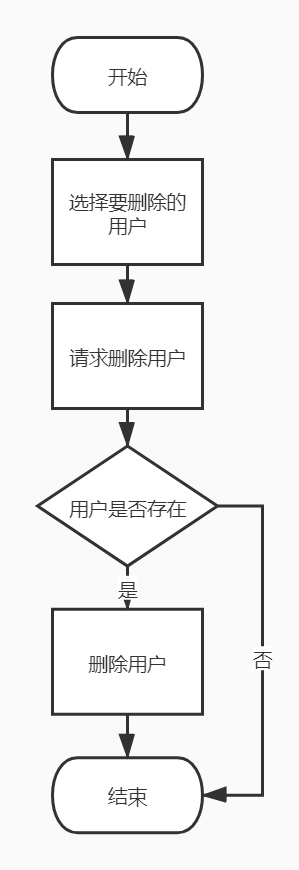


图 5‑五‑9删除用户的流程图

#### 图片管理模块设计

图片管理模块包括收藏图片、浏览收藏夹中的图片、删除收藏夹的图片、识别收藏夹中的图片、上传图片进行识别等功能。收藏图片就是用户上传图片后收藏该图片，该图片将会存储进该用户的收藏夹中。浏览收藏夹的图片则是将用户收藏夹中的图片按照用户上传的时间顺序从早到晚进行显示。删除收藏夹中的图片流程图如图 5‑10所示。

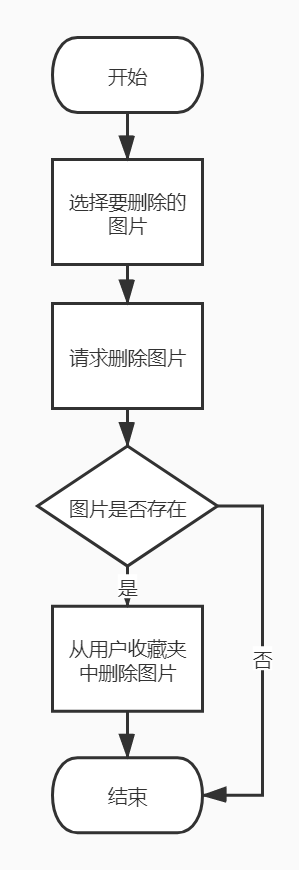


图 5‑五‑10 删除收藏夹中的图片流程图

识别收藏夹中的图片的流程为用户从收藏夹中选择图片，之后点击识别该图片将会上传给后端进行识别，之后识别的结果将会被返回并显示在网页上。上传图片进行识别的流程为用户选择一张本地的照片进行上传，点击上传之后该图片将会被上传至后端进行识别，识别结果将会展示在网页上。

### 数据库设计

本系统的数据库设计E-R图如图 五‑11所示，用到的数据表包括用户表(User)、图片表(Image)以及收藏表(Collect)三个表。

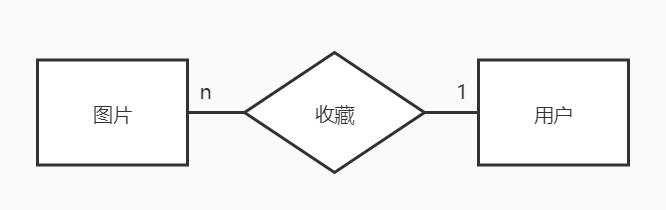


图 五‑11 E-R图

用户表的结构如表 五‑1所示。其中ID是自增的，而用户名和密码是用户注册时自己设置的。在设置的过程中要保证不能与其他人的用户名重复。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否主键 | 是否唯一 | 可否为空 | 说明 |
| ID | Long | Y | Y | N | 自增 |
| Username | Varchar(255) | N | Y | N | 用户名 |
| Password | Varchar(255) | N | N | N | 密码 |

表 五‑1 用户表

图片表的结构如表 五‑2所示。其中ID是自增的，在用户上传图片之后，会将图片存储在指定位置之后生成一个图片路径，之后将该路径存储在数据库中。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否主键 | 是否唯一 | 可否为空 | 说明 |
| ID | Long | Y | Y | N | 自增 |
| Path | Varchar(255) | N | N | N | 图片路径 |

表 五‑2 图片表

收藏表的结构如表 五‑3所示。其中ID是自增的，在用户上传图片之后，将会把用户ID和生成的图像ID在收藏表中创建一条记录，记录图片与用户的所属关系。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 类型 | 是否主键 | 是否唯一 | 可否为空 | 说明 |
| ID | Long | Y | Y | N | 自增 |
| UserID | Long | N | N | N | 用户ID |
| ImageID | Long | N | N | N | 图片ID |

表 五‑3 收藏表

## 鸟类细粒度识别系统的实现

该鸟类识别系统的实现是前端是使用jQuery、Bootstrap实现的。由于模型是使用Pytorch训练的，因此本文使用Python常用的轻量级Web框架搭建了一个小网站，该网站只有一个接口用来识别图片中鸟类所属的类别。后端的其他接口是使用Spring Boot实现的，数据库使用的是MySQL。前后端通信的接口使用的是Restful接口。

### 登录注册页面

登录界面如图 五‑13所示，注册界面如图 五‑12所示。登录注册失败时会显示相应的错误信息。同时注册时会判断用户名是否为admin，如果是则提醒用户名已存在。登录成功时会根据用户名是否为admin跳转至不同的页面。另外当用户打开登录或注册界面且已登录，则会直接跳转至相应的主页。只有用户退出登录之后才可以重新登录。

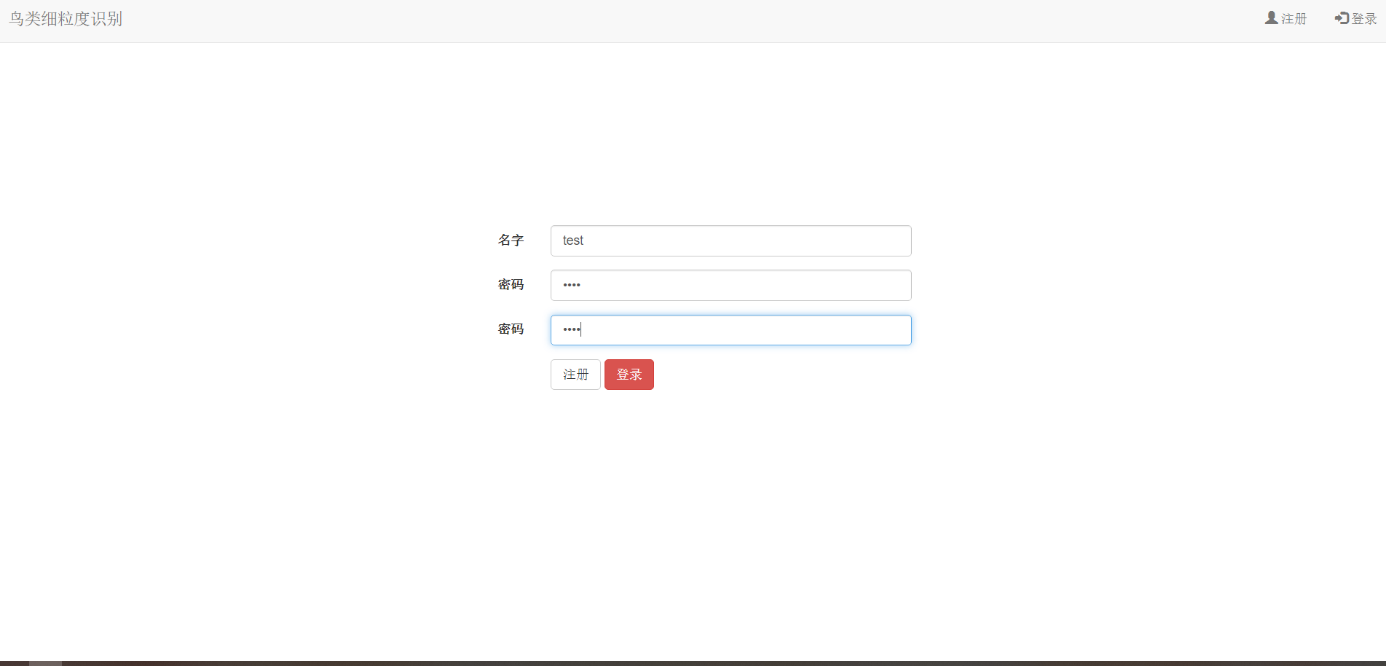


图 ‑12 注册页面

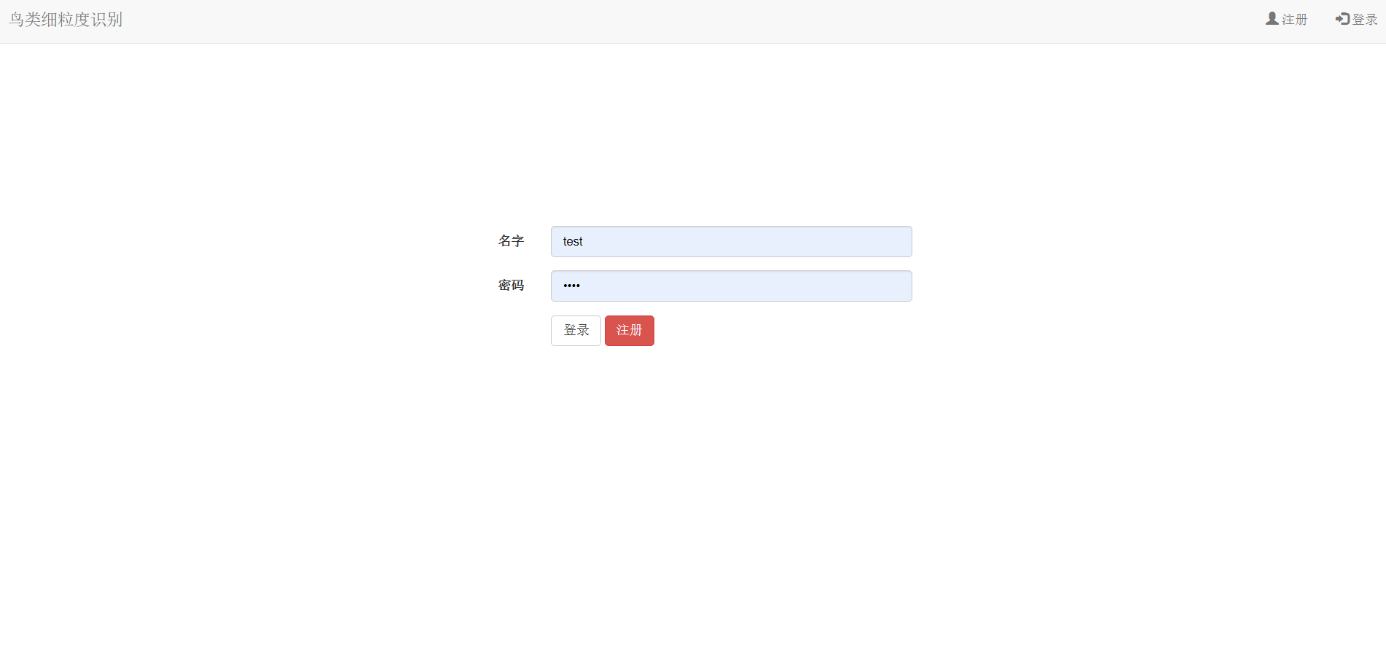


图 五‑13 登录页面

### 修改密码页面

登录之后右上角将会显示用户名和一个下拉菜单，用户点开将显示我的收藏、修改密码、退出登录操作；管理员则显示用户列表、收藏列表、退出登录。点击退出登录之后用户需要重新登录之后才能进行相关操作。点击修改密码将会显示如图 五‑14所示的修改密码页面。



图 五‑14修改密码页面

### 鸟类识别界面

鸟类识别的初始界面如图 五‑16所示，在该页面用户可以上传图片进行识别和收藏等操作。识别之后的页面如图 五‑15所示。收藏之后的页面如图 五‑17所示。

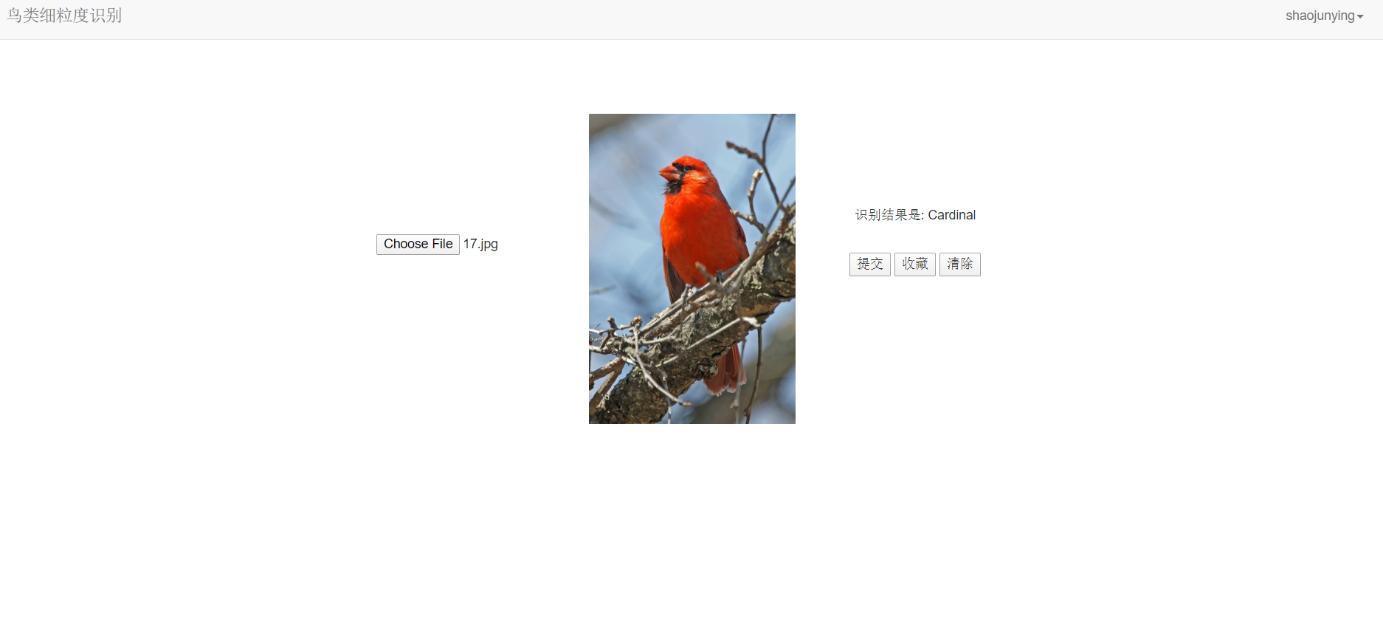


图 ‑15识别页面

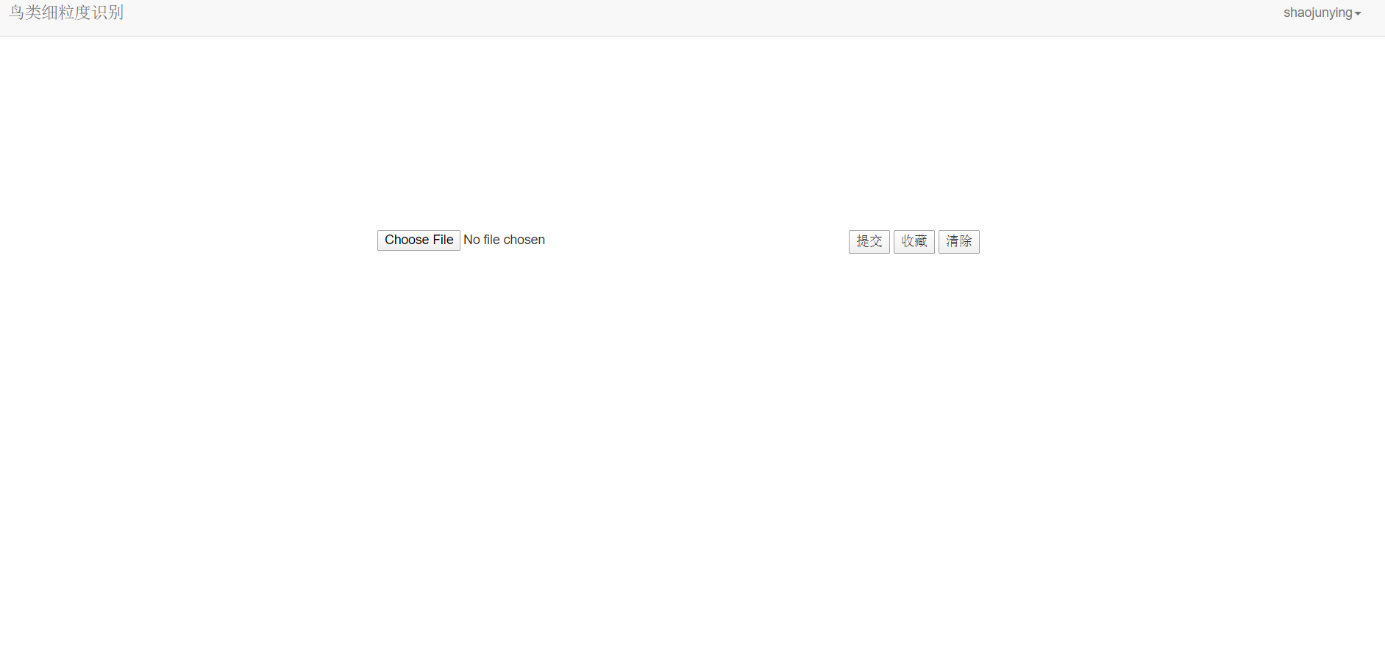


图 ‑16初始页面



图 五‑17收藏页面

### 收藏列表

用户可以查看自己的收藏列表，用户的收藏列表页面如图 五‑18所示。

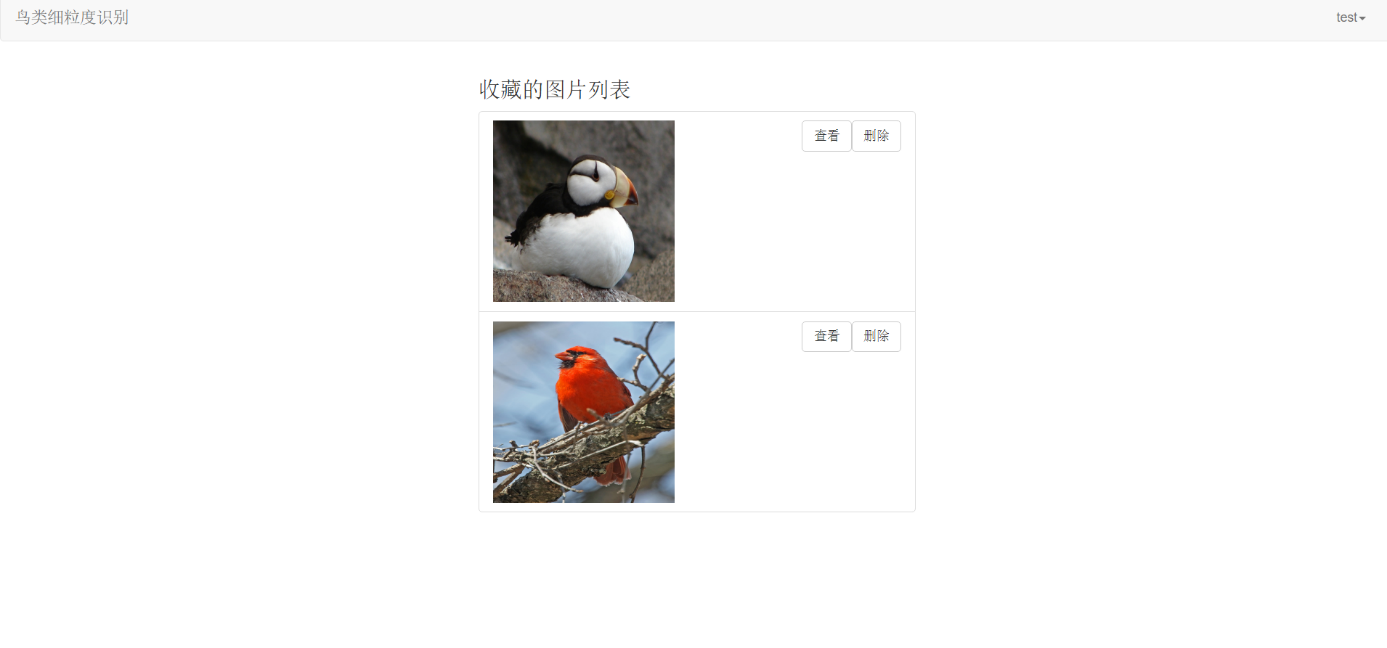


图 五‑18收藏列表

### 管理员查看用户列表

管理员可以查看用户列表以及删除指定用户，界面如图 五‑19所示。



图 五‑19 用户列表

### 管理员查看收藏列表

管理员可以查看所有用户的收藏列表，并且可以删除指定用户的收藏夹中的照片。管理员看到的收藏列表界面如图 五‑20所示

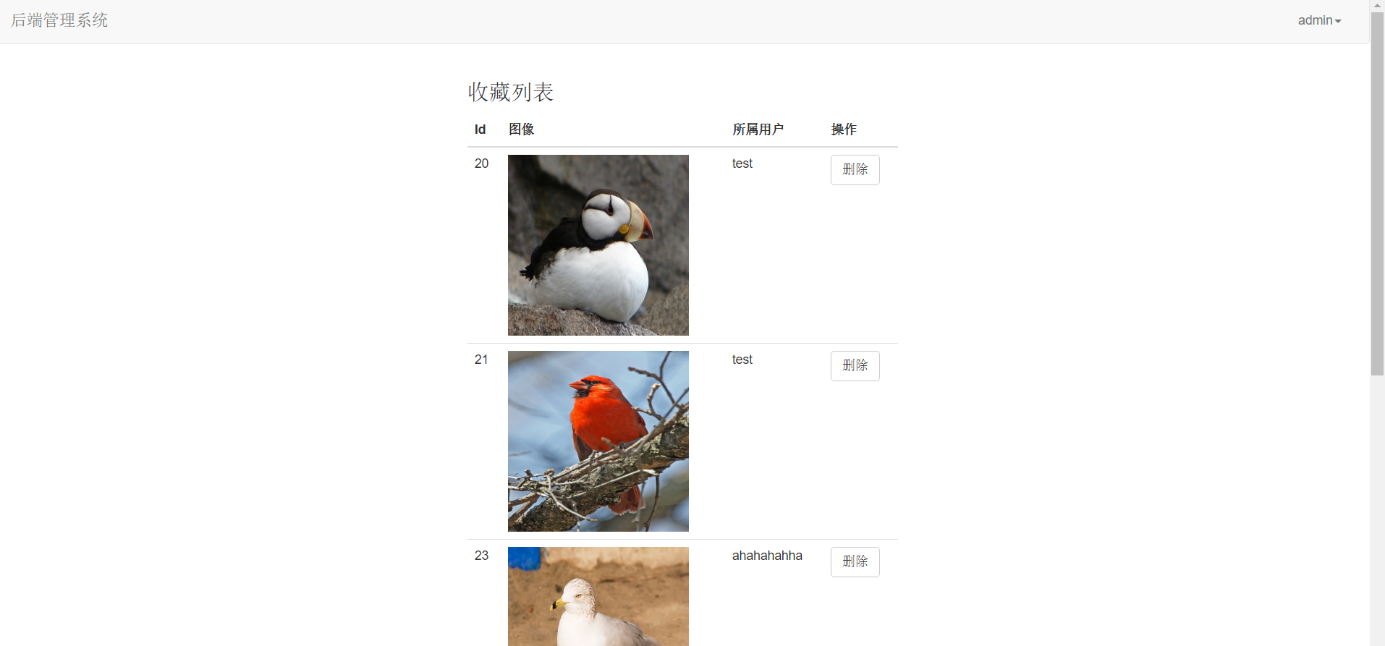


图 五‑20收藏列表

## 本章小结

本章首先介绍了鸟类细粒度识别系统的需求分析内容，主要包括系统需求分析、各种业务的流程图分析、图片和用户信息的数据流图分析以及一些非功能性的需求。之后则详细介绍了该系统的设计，包括系统的整体设计、用户和图片等各模块的功能设计以及数据库设计等。最后则介绍了该系统的具体实现。主要展示了系统的各个页面。从而较详细地展示了该鸟类细粒度识别系统分析、设计与实现过程。

# 总结与展望

## 毕业设计总结

基于细粒度图像识别的类间差异性小、类内差异性大以及数据量比较小的特点，本文实现了一个基于子类与深度学习的鸟类细粒度识别网络模型。在鸟类细粒度识别的标准数据集CUB-200-2011上，基于不同的经典CNN模型来实现论文中的模型进行迁移学习可以获得最高4.3%的相对准确率的提升。研究的主要内容概括如下：

1. 选择了鸟类细粒度识别的基准数据集CUB-200-2011作为本文中实现的模型的训练和测试的数据集。本文首先使用数据集中每个样本的边界框信息取出每个样本中鸟类所在的区域。另外，由于模型的数据量较小，为了避免模型的过拟合，本文使用了数据增强技术相对地增加了数据集的规模。此外，本文还使用了Dropout、权值衰减等技术进一步避免了模型的过拟合。
2. 基于Pytorch、sklearn等技术，本文在GPU运算平台上实现了基于子类与深度学习的鸟类细粒度识别网络模型。该网络模型首先提取出训练样本的特征。之后将根据这些特征将训练样本分成K个不同的且互不重叠的子类。之后将为每个子类分别创建一个分类器。使用每个子类分开地训练每个子分类器。在将每个子分类器训练到一个比较好的程度之后，将会把每个子分类器的结果进行加权求和作为模型对于某一样本的最终预测结果。之后将训练优化结合在一起的K个分类器。
3. 在训练与优化的过程中，在迁移学习时对于不同的层设置不同的学习率，从而使模型可以达到较好的训练效果。另外本文尝试了在训练过程中在模型的loss达到不同值时将K个网络开始合在一起的训练，比较验证集的准确率，从而可以进一步提高模型的准确率。

## 未来展望

虽然本次实验实现的网络结构取得了不错的效果，但是还是有很多地方需要进一步的改进：

1. CUB-200-2011数据集规模相对较小，可能最终的模型提升效果不明显。选择其他规模相对较大的数据集，例如Birdsnap[13]等数据集，通过不同数据集、不同底层网络结构的对比从而更加清楚地说明该网络模型对于细粒度图像识别能力相比于经典分类网络的提升情况。
2. 没有进一步验证子类的数量对于分类结果是否有影响。后续可以尝试使用不同的子类个数查看网络模型最终的测试准确率是否有较大的差别。
3. 尝试不同超参数，看模型是否还可以进一步优化。由于时间问题，以及GPU性能较差导致训练时间较长，因此如果条件允许可以调节不同超参数进行相应尝试，看准确率能否再做提升。特别是对于ResNet，看是否能相应的优化从而使新搭建的模型的准确率能达到微调的预训练ResNet模型的准确率，以及能否再做进一步的提升。

**参考文献**

1. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybernetics 36, 193–202 (1980).
2. Y. LeCun, “LeNet-5, convolutional neural networks,” November 2013.
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classiﬁcation with deep convolutional neural networks. In NIPS, pages 1097–1105, 2012.
4. C. Szegedy, W. Liu, and Y. Jia. Going deeper with convolutions. CVPR, 2014.
5. He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015:770-778
6. X. Wei, C. Xie, J. Wu, and C. Shen. Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. Pattern Recognition, 76:704–714, 2018.
7. N. Zhang, J. Donahue, and R. Girshick, et al. Partbased R-CNNs for ﬁne-grained category detection. In ECCV, pages 834–849. 2014.
8. R. A. Jacobs, M. I. Jordan, and S. J. Nowlan, et al. Adaptive mixtures of local experts. Neural Computation, 3(1):79–87, 1991.
9. C. R. Rao. The utilization of multiple measurements in problems of biological classiﬁcation. Journal of the Royal Statistical Society. Series B (Methodological), 10(2):159–203, 1948.
10. J. Yosinski, J. Clune, and Y. Bengio, et al. How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems, pages 3320–3328, 2014.
11. J. Deng, W. Dong, and R. Socher, et al. Imagenet: A large-scale hierarchical image database. In CVPR, 2009.
12. Wah C, Branson S, Welinder P, et al. The caltech-ucsd birds-200-2011 dataset[J]. 2011,1(1):1-45
13. T. Berg, J. Liu, and S. W. Lee, et al. Birdsnap: Large-scale ﬁne-grained visual categorization of birds. In CVPR, pages 2019–2026, 2014.