深圳大学考试答题纸

(以论文、报告等形式考核专用)

二〇二 三 ～二〇二 四 学年度第 2 学期

课程编号 1501990016 课程名称 人工智能概述 主讲教师 高灿 评分

学 号 2022090232 姓名 彭俊杰 专业年级 22 级电子信息工程专业

教师评语:

题目: Minspore 深度学习框架学习与使用

【摘要】

MindSpore 作为华为推出的全场景深度学习框架，自其诞生以来，便凭借卓越的计算性能、灵活的 API 接口以及全场景统一部署的能力，迅速在深度学习领域崭露头角。在信息爆炸的时代，图像作为人类获取信息的重要载体，占据了显著的地位。因此，MindSpore 在图像分类、目标检测、语音识别等任务中得到了广泛的应用，受到了众多企业和研究机构的青睐，深圳大学亦开始将其纳入教学体系。

本文概述了使用 MindSpore 深度学习框架进行的两个关键实验：猫狗分类任务的网络训练与部署，以及手写体图像识别任务。在猫狗分类实验中，我们凭借 MindSpore 的易用性，迅速构建了一个能够精准区分猫和狗图像的深度学习模型。通过详细的数据预处理、模型设计、训练与优化流程，我们成功实现了高准确率的分类效果，验证了 MindSpore 在图像分类领域的卓越能力。而在手写体图像识别实验中，我们利用 MindSpore 框架对经典的 MNIST 数据集进行了训练和测试，通过精心设计的卷积神经网络架构，成功实现了对手写数字图像的高精度识别，并通过优化进一步提升了模型的性能。

这两个实验不仅充分展示了 MindSpore 在深度学习领域的广泛应用和强大功能，同时也为我们未来在更多领域和场景中探索和应用 MindSpore 提供了宝贵的经验和启示。随着技术的不断进步和应用的不断拓展，我们有理由相信，MindSpore 将在深度学习研究和应用中发挥更加重要的作用，推动人工智能技术的持续创新与发展。

【关键词】 MindSpore；深度学习框架；猫狗分类；手写体识别；神经网络

目　录

第 1 章 简介及相关工作

1. 简介

随着人工智能技术的飞速发展，深度学习框架作为支撑深度学习研究和应用的关键基础设施，其重要性日益凸显。MindSpore，作为华为推出的全场景深度学习框架，自发布以来，便凭借其卓越的计算性能、灵活的 API 接口以及全场景统一部署的能力，受到了广泛的关注和应用。在信息爆炸的时代，图像作为人类获取信息的重要载体，其处理和分析技术在多个领域都具有重要意义。因此，本文将基于 MindSpore 深度学习框架，探索图像分类和手写体图像识别这两个关键任务，旨在验证 MindSpore 在处理这些任务时的有效性和性能。

图像分类是计算机视觉领域的基础任务之一，它涉及对图像中的物体进行识别和分类。随着深度学习技术的不断发展，基于卷积神经网络的图像分类方法取得了显著的成果。猫狗分类作为图像分类的一个具体实例，具有广泛的应用场景和重要的研究价值。同时，手写体图像识别作为数字识别领域的重要分支，也具有广泛的应用前景。MNIST 数据集作为手写体数字识别的基准数据集，被广泛用于验证和比较各种算法的性能。

因此，本论文聚焦于实现猫狗分类与手写体识别两个任务，通过任务驱动学习 Mindspore 深度学习框架具体使用与部署。

2. 相关工作

在深度学习领域，已有多种深度学习框架被提出并广泛应用于各种任务中。其中，TensorFlow、PyTorch 和 Keras 等框架因其出色的性能和易用性而受到广泛的关注。然而，这些框架在全场景统一部署和硬件优化等方面仍存在一些局限性。相比之下，MindSpore 作为华为推出的全场景深度学习框架，在这些方面展现出了显著的优势.

MindSpore 采用原生设计和原生优化，支持全场景统一部署，可以方便地应用于云、边缘和端侧等场景。同时，MindSpore 还提供了灵活的 API 接口和易用的编程范式，使得用户可以轻松构建深度学习模型并进行训练和推理。此外，MindSpore 还支持动态图执行、自动并行计算、混合精度训练等先进技术，进一步提升了模型的训练速度和推理性能。

在图像分类和手写体图像识别任务中，已有大量的研究工作基于各种深度学习框架进行了探索和尝试。这些工作涉及模型的构建、优化和部署等多个方面，并取得了显著的成果。然而，这些工作大多基于传统的深度学习框架进行，对 MindSpore 的应用和性能验证相对较少。因此，本文旨在基于 MindSpore 深度学习框架，对图像分类和手写体图像识别任务进行深入研究，并验证 MindSpore 在处理这些任务时的有效性和性能。

具体而言，本文将首先介绍 MindSpore 深度学习框架的基本原理和关键技术，包括其架构、API 接口、动态图执行等。然后，本文将详细阐述基于 MindSpore 的猫狗分类实验和手写体图像识别实验的过程和结果。在猫狗分类实验中，本文将通过数据集预处理、模型设计、训练和优化等步骤，构建一个能够区分猫和狗图像的深度学习模型，并验证其在不同场景下的性能。在手写体图像识别实验中，本文将利用 MNIST 数据集构建一个卷积神经网络模型，并通过调整网络结构、优化算法和参数设置等方式，进一步提升模型的性能。最后，本文将分析 MindSpore 在处理这些任务时的优势和不足，并展望其未来的发展方向和应用前景。

第 2 章 原理及方法分析

1.原理

使用 MindSpore 进行猫狗识别和手写体识别的原理主要基于深度学习中的卷积神经网络（Convolutional Neural Networks, CNN）。猫狗识别和手写体识别是典型的图像分类问题，通过训练一个深度学习模型来识别图像中的物体。在这个过程中，模型会学习从输入图像中提取有用的特征，并使用这些特征来做出分类决策[1]。

2.方法分析与相关知识

（一）神经网络（Neural Network，NN）

人工神经元（简称神经元）是神经网络的基本组成单元，它是对生物神经元的模拟、抽象和简化。现代神经生物学的研究表明，生物神经元是由细胞体、树突和轴突组成的。通常一个神经元包含一个细胞体和一条轴突，但有一个至多个树突。

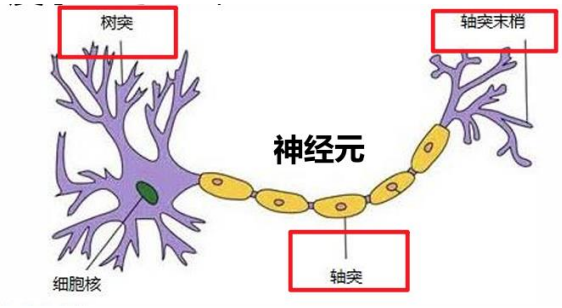


图 1 人类神经元结构

神经网络中最基本的成分是神经元（Neuron）模型。生物神经网络中的每个神经元彼此互连，当它“兴奋”时，就会向相连的神经元发送化学物质，从而改变这些神经元内的电位。如果某神经元的电位超过一个阈值，它就会被激活，即“兴奋”起来，向其他神经元发送化学物质。 1943 年，美国心理学家麦卡洛克(McCulloch)和数学家皮特斯（Pitts）按照生物神经元的结构和工作原理建立了 M-P 模型。

在 M-P 模型中，为了使得建模更加简单，以便于进行形式化表达，我们忽略时间整合作用、不应期等复杂因素，并把神经元的突触时延和强度当成常数。激活函数将输入值映射为输出值 0 或者 1，0 表示神经元抑制 1 表示神经元兴奋，然而阶跃函数具有不连续和不光滑等不太好的性质，因此实际常用 Sigmoid 函数作为激活函数。将许多神经元按一定的层次结构连接起来，就得到了神经网络

（二）感知机（Perceptron）

感知机(Perceptron)由两层神经元组成，这个结构非常简单，它其实就是输入输出两层神经元之间的简单连接。

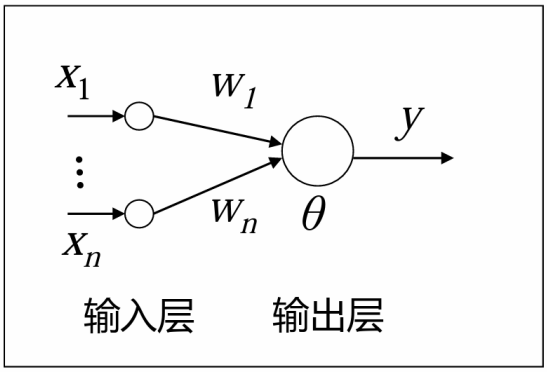


图 2 基本感知机示例

感知机是一种二分类的线性分类模型，它输入为实例的特征向量。在图像分类或图像处理领域中，通常输出+1 或者-1，对应输入空间中将实例划分为正负两个超平面，是一种判别模型。在猫狗分类中，若需要应用感知机，我们需要手机大量的猫狗图片，其中我们运用了 kagglecatsanddogs_5340 猫狗图片作为数据集。并把图片标记为+1 或者-1。例如猫标记为+1，狗标记为-1。然后从每张照片提取图像的特征，包括但不限于颜色、纹理、形状等（对应 Xi）。这些特征能够区分猫和狗，因此利用感知机能够得到一个超平面将猫与狗的特征向量正确划分为两部分。

（三）卷积神经网络（Convolutional Neural Network，CNN）

卷积神经网络（Convolutional Neural Network，CNN）是一种深度学习模型，主要用于处理具有网格结构数据的任务，如图像和视频。CNN 在计算机视觉领域中广泛应用于图像分类、目标检测、人脸识别等任务，并取得了很大的成功。

CNN 的核心思想是通过卷积层、池化层和全连接层等组件，对输入数据进行层级化的特征提取和抽象。以下是 CNN 的一些关键组件和概念：

（1）卷积层（Convolutional Layer）：

卷积层是 CNN 的核心部分，用于提取输入数据的特征。它通过在输入数据上滑动一个或多个卷积核（也称为滤波器），对局部区域进行特征提取。卷积操作可以捕捉到输入数据中的空间局部性模式，并保留空间结构信息。卷积层的输出称为特征图（Feature Map）。

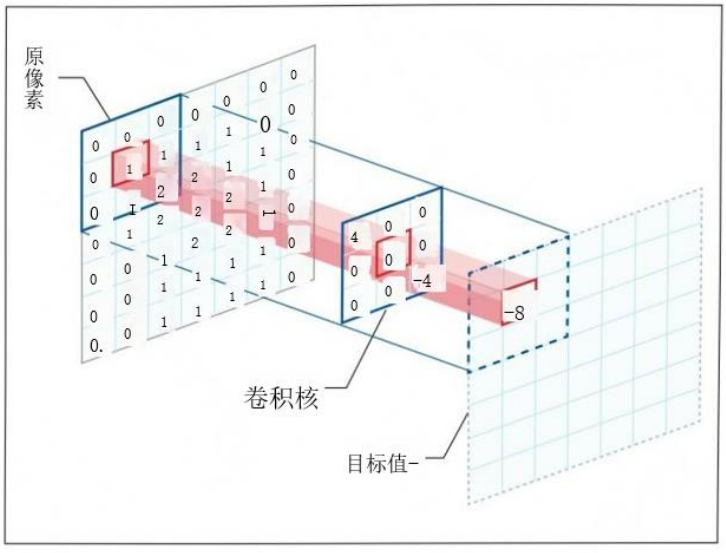


图 3 图像卷积计算示例

（2）池化层（Pooling Layer）：

池化层用于降低特征图的空间尺寸，并减少模型的参数数量。常见的池化操作包括最大池化（Max Pooling）和平均池化（Average Pooling），它们分别从局部区域中选择最大值或平均值作为池化后的值。池化操作可以提高特征的平移不变性和尺度不变性。

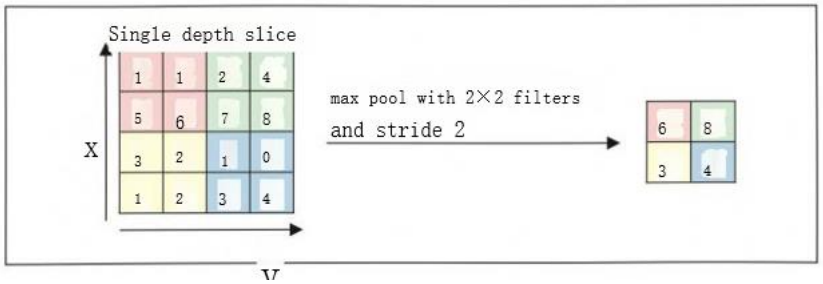


图 4 最大池化

（3）全连接层（Fully Connected Layer）：

全连接层将前面的卷积层和池化层的输出连接起来，并通过一系列的全连接操作进行特征的组合和分类。全连接层通常用于最后的分类任务，将高级抽象特征映射到类别标签。

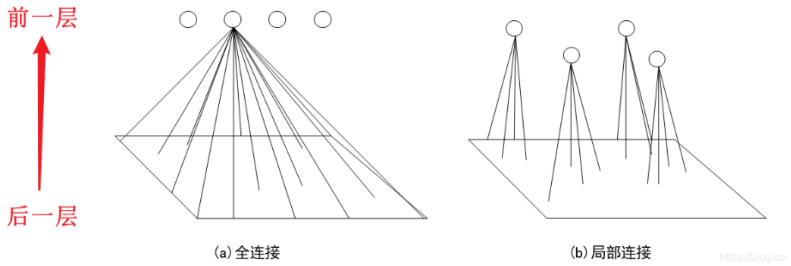


图 5 全链接与局部链接示意

（4）激活函数（Activation Function）：

激活函数引入非线性变换，增加 CNN 的表达能力。常见的激活函数包括 ReLU（Rectified Linear Unit）、Sigmoid 和 Tanh 等。激活函数在卷积层和全连接层之后被应用于特征图或神经元上，引入非线性特性。

在猫狗分类或者手写体识别等图像处理领域中，我们需要通过卷积操作提取数据集中的局部特征，并通过池化操作来降低数据的维度和复杂度。在 CNN 的最后一层，我们会使用一些优化算法如梯度下降来训练网络，达到分类猫狗和手写体的的目的。CNN 具有感知机不具备的非线性拟合能力，因此在猫狗分类实验中，其提供的代码也是基于 CNN 卷积神经网络进行训练的。

第 3 章 实验论证

猫狗分类实验主要参考助教所发的实验文档与华为 MindSpore 官网：MindSpore 官网提供的教程逐步进行实验。具体步骤分为：环境配置、数据预处理、训练模型、部署本地以及最终的探究不同参数对模型的影响。手写体识别实验主要基于 MINST 官网：http://yann.lecun.com/exdb/mnist/提供的 MNIST 数据集和 https://blog.csdn.net 所提供的教程进行学习并实现基础手写体的识别。

1. 猫狗分类网络

（一）环境配置

通过 Anaconda Prompt 创建 MindSpore 环境，指定 python 版本为 3.7.5，以便符合后续所需要包的版本要求。

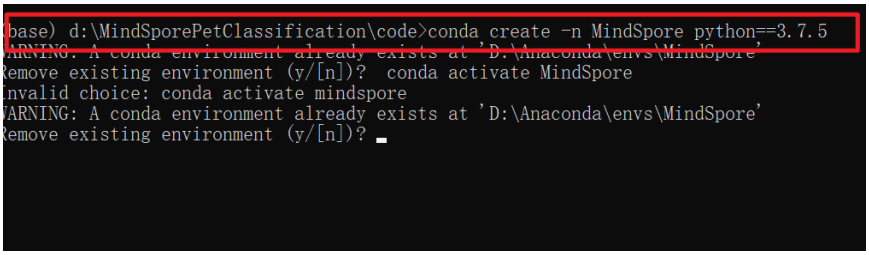


图 6 创建 MindSpore 环境

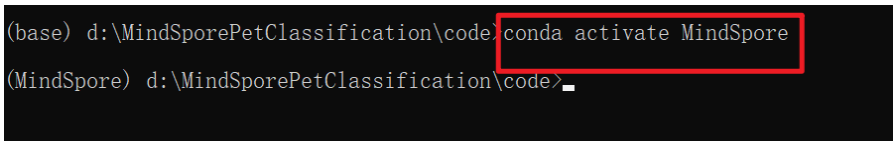


图 7 激活 MindSpore 环境

输入：

pip install https://ms-release.obs.cn-north-4.myhuaweicloud.com/1.5.0/MindSpore/cpu/x86_64/mindspore-1.5.0-cp37-cp37m-win_amd64.whl --trusted-host ms-release.obs.cn-north-4.myhuaweicloud.com -i https://pypi.tuna.tsinghua.edu.cn/simple 下载 mindspore 包，在这里我是从华为 MindSpore 官网寻找了最新版本的 CPU 版本的 mindspore 下载地址。

图 8 下载 mindspore 包

输入：python -c "import mindspore;mindspore.run_check()"检查 mindspore 库是否成功安装

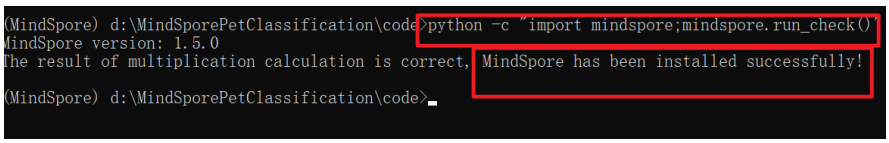


图 9 检查 mindspore 包是否成功安装

从官网下载最新数据集

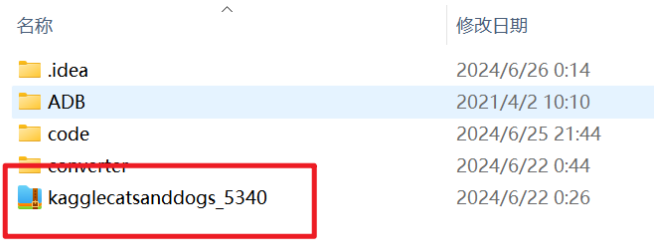


图 10 从官网下载最新数据集_5340

（二）数据预处理

输入：`cd /d d:\MindSporePetClassification\code` 改变当前根目录至 code

输入：`python preprocessing_dataset.py`

`D:\MindSporePetClassification\kagglecatsanddogs_5340.zip` 预处理数据图片

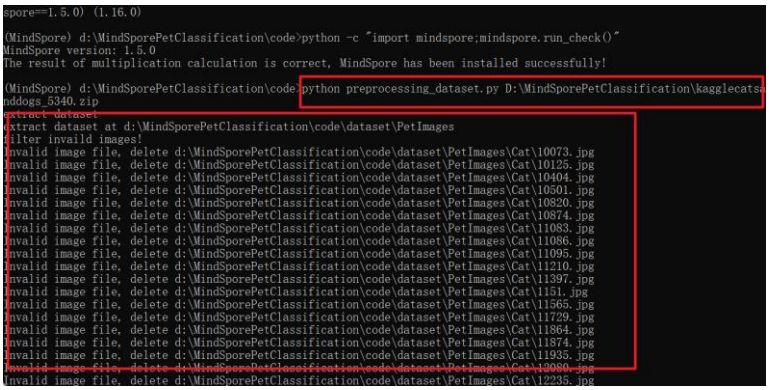


图 11 数据照片的预处理

（三）训练模型

输入：`python train.py` 开始训练模型

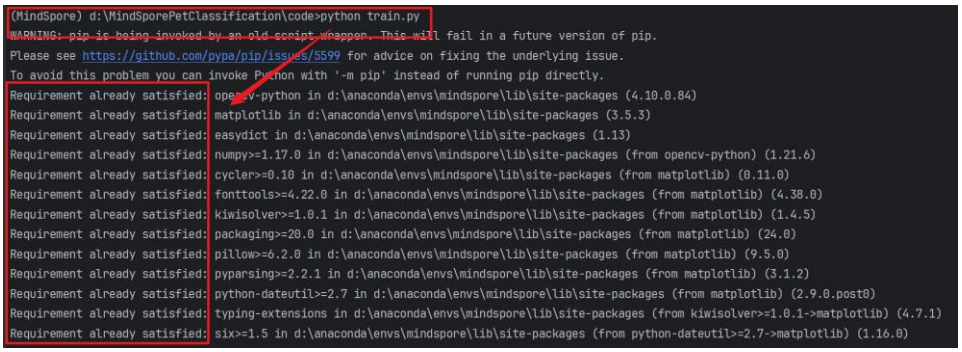


图 12 环境已准备就绪

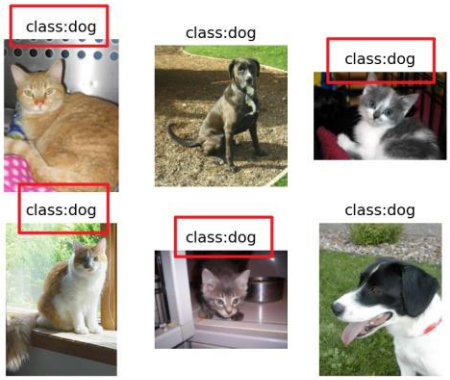


图 13 未经过训练的结果

可以看到，在环境准备好后，模型首先给出了未经过训练的结果，我们仔细观察可以发现，模型将所有的动物都打上了“dog”的标签，这很明显是错误的，因此我们关闭图形化显示窗口，开始训练模型，观察训练后的模型能否纠正先前的结果。

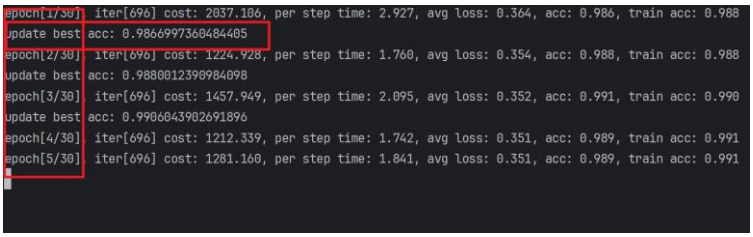


图 14 模型迭代中

训练是不断迭代的过程，通过一批批数据的输入，进行迭代计算，逼近损失/误差最小的那个模型。其中 epoch 是指训练的迭代数，可以看到随着迭代数增加，平均损失函数（avg loss）逐步降低，精度（acc）逐步提高。可以观察到，每次经过一个 epoch，若 acc 相比先前的模型有所提高，模型便会自动迭代选择最高 acc 的结果。训练最后，达到我们设置的精度阈值，或者训练最大迭代数，训练停止。我们可以看到训练耗费的时间 19 分钟，其中包含数据处理和重训两个步骤耗时，其中重训只需要 1 分钟左右，效率非常高，同时精度就能达到了 98%+，可以满足日常基本使用。

图 15 训练后的结果

训练完成后可以看到，模型能够对我们随机选择的猫狗图片做出正确的预测。

（四）部署本地

输入：`cd d:\MindSporePetClassification\converter` 改变当前根目录至 `converter`
输入：
`call converter_lite --fmk=MINDIR --modelFile=d:\MindSporePetClassification\code\mobilenetv2.mindir --outputFile=pet` 进行模型文件格式转换

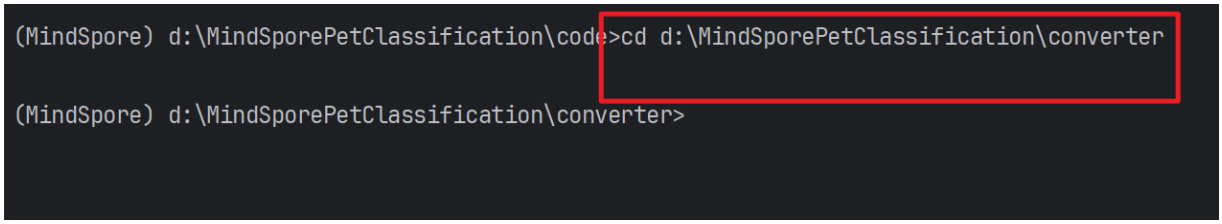


图 16 改变根目录

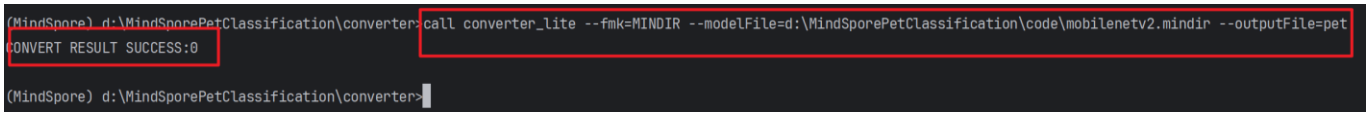


图 17 显示结果为“0”表示成功转换

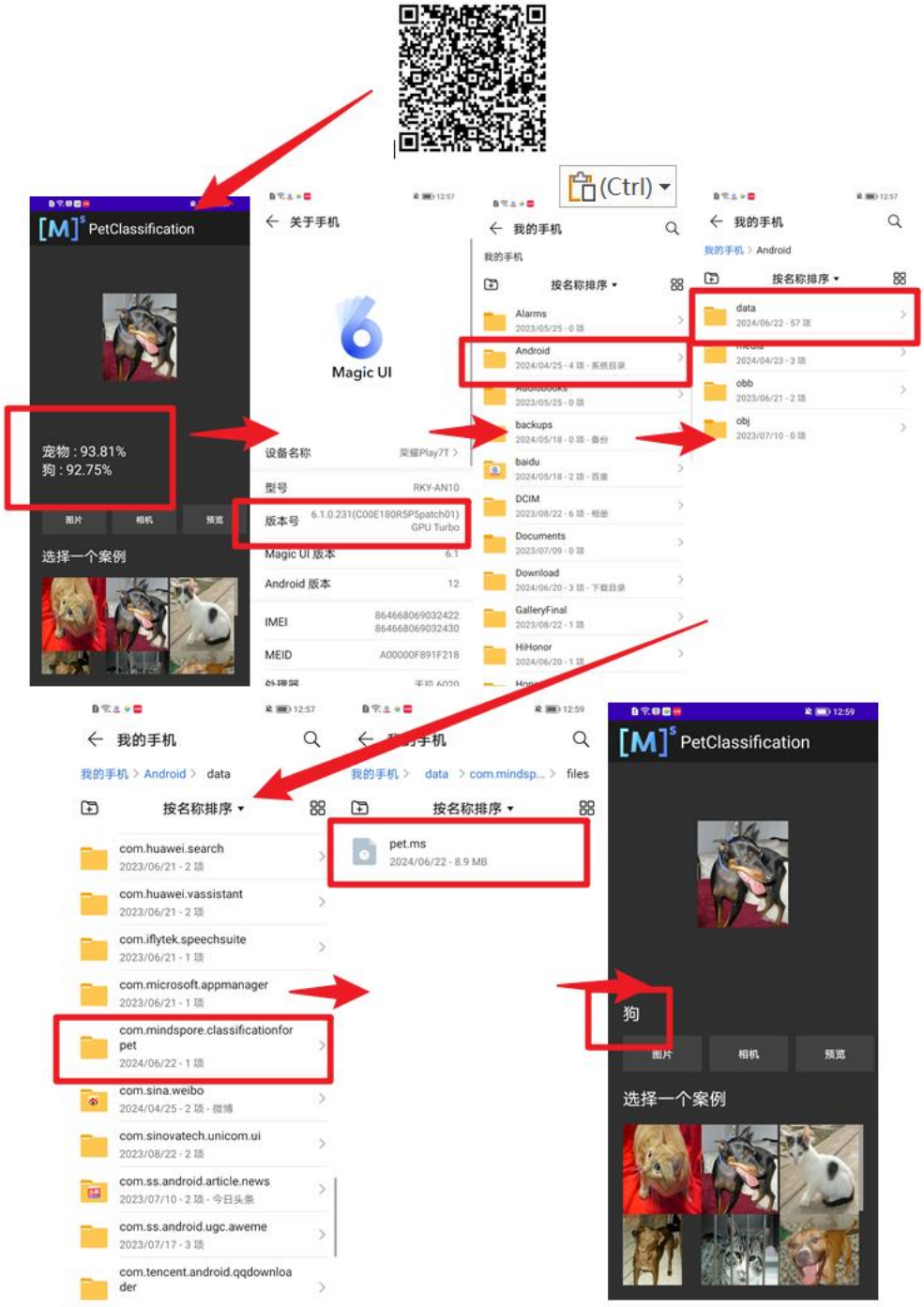


图 18 本地部署步骤

（五）探究不同参数对模型的影响

1. 通过随机选择 10 张数据库的猫狗图片并进行模型训练：修改 utils.py 的 predict_from_net 中 subplot 的参数(2,3,i+1)为 (2,5,i+1)；sample_nums 改为 10

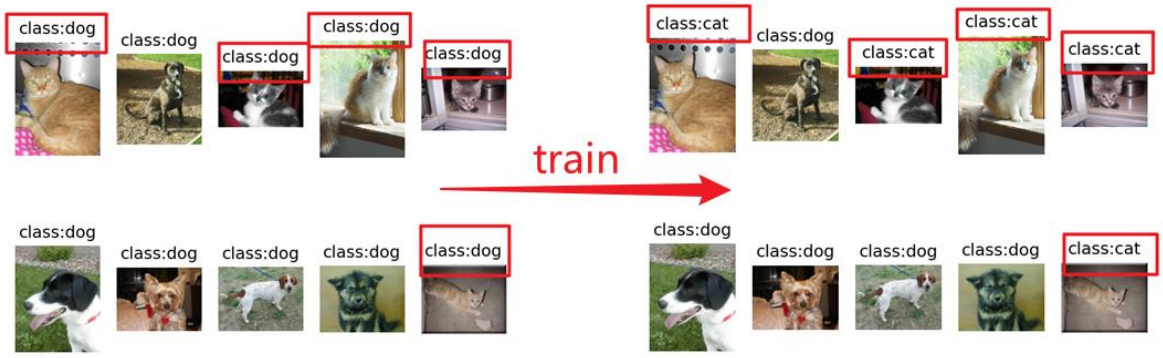


图 19 模型训练前后

2. 采用网上自己寻找的 5 张猫图片和 5 张狗图片使用本地部署的 app 进行预测：

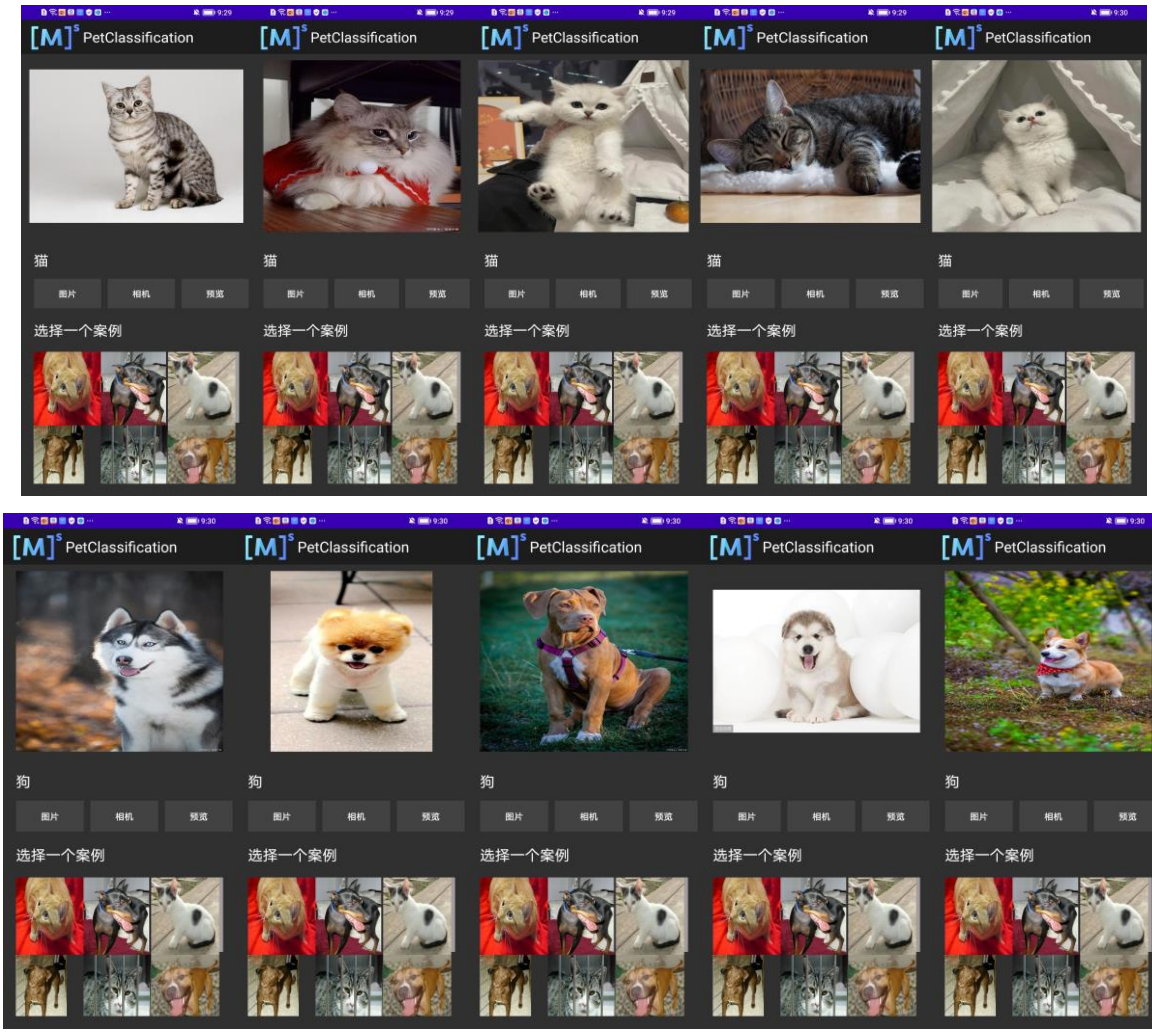


图 20 猫狗预测结果准确率 100%

通过作出迭代过程中 train_acc 和 acc 参数的变化图，可以看出，train_acc 逐渐上升并趋于较高的平稳值，总体趋势与 acc 相似。Acc 在 Epochs=5 的时候出现了小范围的下滑，可能是因为学习率突变的原因。

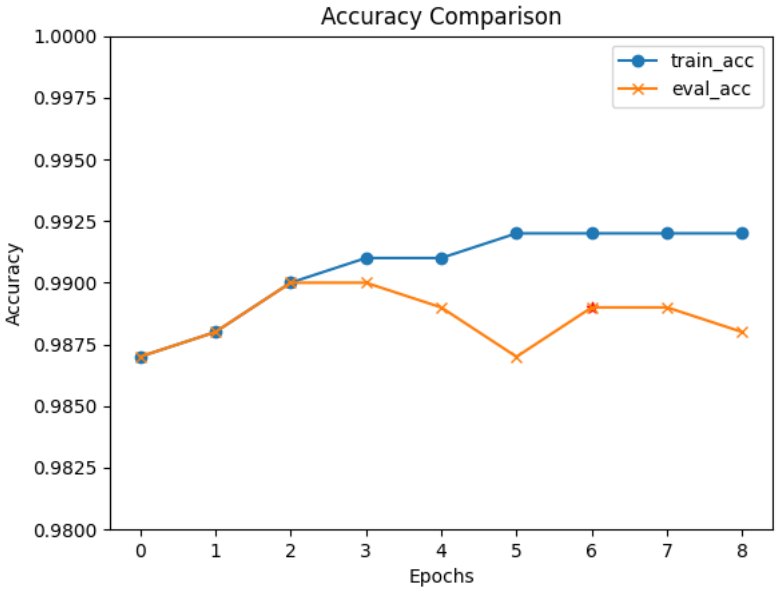
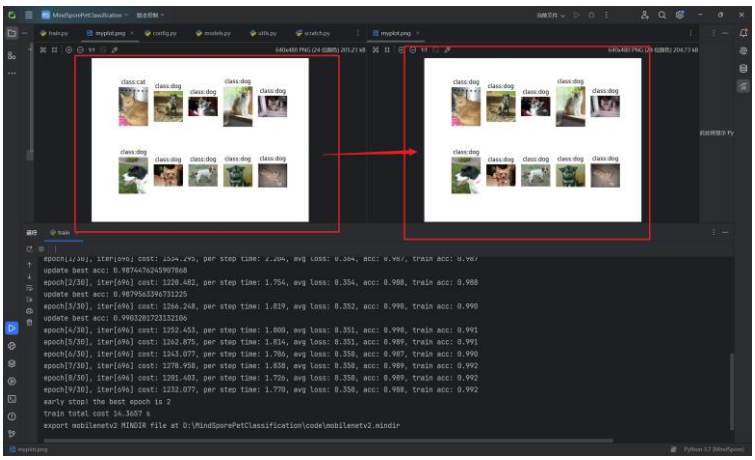


图 21 Epochs 与 train_acc、acc 关系

3. 使用 fine-tuning 和未使用 fine-tuning 的模型预测性能：
在未使用 fine-tuning 的模型预测时，我发现在训练前和训练后都无法完成猫狗识别的任务。通过观察 acc 和 train_acc 的值，包括 avg_loss 的值，暂时没能找出出现这种情况的原因。



4. batch size 从 32 改为 64：
batch size 的大小表示模型一次处理图像的个数，从 32 改为 64，能够使得模型一次处理图像的个数增加。从结果上看，模型成功完成了猫狗分类的任务。

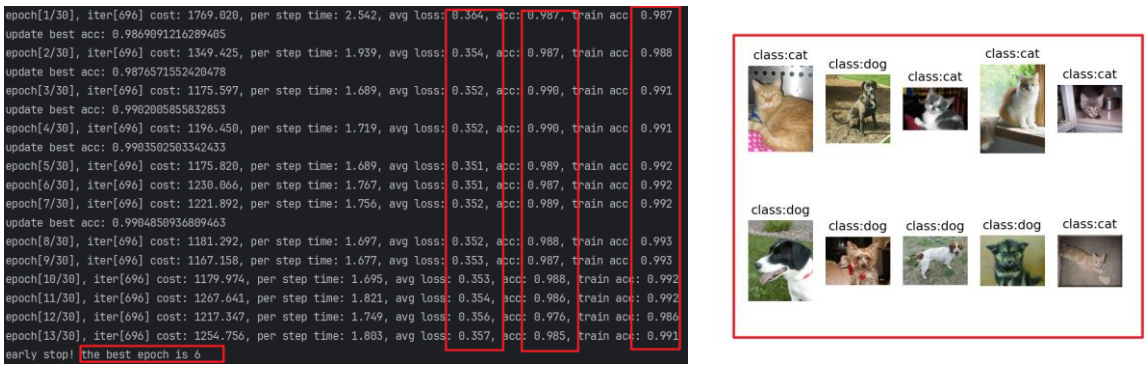


图 22 模型迭代过程（左）、分类结果 100%正确（右）

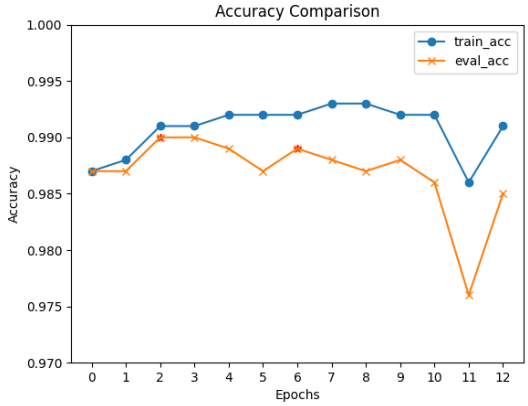


图 23 batch size=64

5. warmup_epochs 从 0 修改至 5：

warmup_epochs 时预热参数，即初始的几个 epochs 使用较低的学习率，到后面的 epochs 逐渐增大学习率的一个参数。能够观察，此时的迭代次数较少便已经收敛。模型也能够成功完成猫狗分类的任务。通过观察迭代过程中 train_acc 和 acc 参数的变化图，可以看到初始的参数相较于上面是比较低的，处于 0.900 左右，但是随着 epochs 数量的上升，参数也会慢慢上升直到稳定。而没有达到像上文的 0.99 这样的准确率可能是因为我的预热 epochs 的个数为 5，而模型迭代 9 个 epochs 就已经停止，后续增加学习率的 epochs 个数不足的原因。

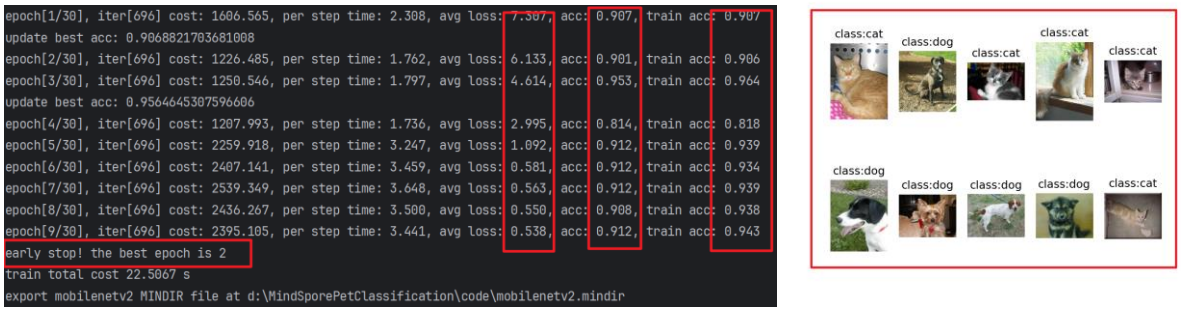


图 24 模型迭代过程（左）、分类结果 100%正确（右）

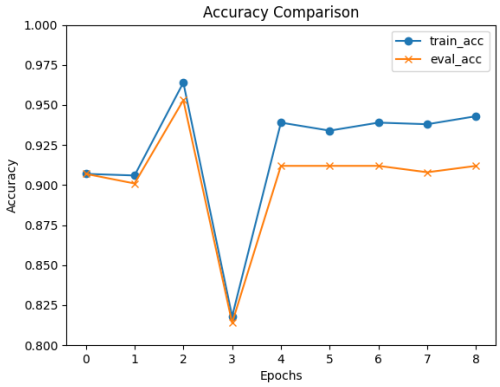


图 25 warmup_epochs=5

6. early stop 修改为 20：

通过修改 early stop 参数，能够增加模型迭代的周期，虽然最终的结果也可能是在 epochs=2 的时候，但是增加了迭代的周期或许有更多的机会找到更好的 acc，但是对硬件的要求也随之提升，不能无限地迭代。

图 26 模型迭代过程（左）、分类结果 100%正确（右）

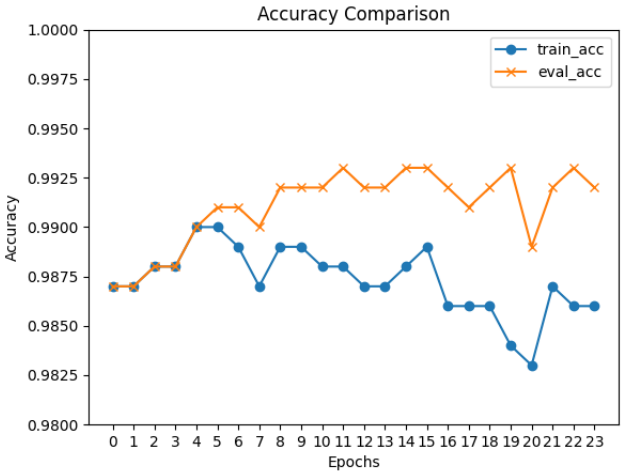


图 27 early stop=20

2. 手写体实验

（一）环境配置

进入 MindSpore 官网查询最近版本的 CPU 驱动 mindspore 包下载路径 L:

图 28 获取安装环境的命令

图 29 实验思维导图

环境配置完成后需要做类似猫狗分类实验的激活环境等操作，在这里不一一赘述。

（二）下载数据集

在 MNIST 文件夹下建立 train 和 test 两个文件夹，train 中存放 train-labels-idx1-ubyte 和 train-images-idx3-ubyte 文件，test 中存放 t10k-labels-idx1-ubyte 和 t10k-images-idx3-ubyte 文件。

图 30 MNIST 官网下载数据集

（三）代码调试

不断运行代码，进行 debug 和补充所需要的包。前前后后经历了很多 debug 和环境重新配置，在这里重点讲一个需要修改的地方。

若报错 AttributeError: ‘DictIterator’ object has no attribute ‘get_next’，这是说 MindSpore 数据类中缺少 “get_next”这个方法，但是在 MNIST 图像识别的官方代码中却使用了这个方法，这就说明 MindSpore 官方把这个变成私密方法。只需要在源码 iterators.py 中找到 DictIterator 这个类，将私有方法变成公有方法就行了（即去掉最前面的下划线）。此时你是很难找到这个 DictIterator 类的，你需要找到其他类中 get_next 方法，并手动改成_get_next，通过 ctrl 找到这个_get_next 在其他地方的使用位置，便可以找到 DictIterator 类，进而去除其前面的下划线，使得这个方法变成公有。

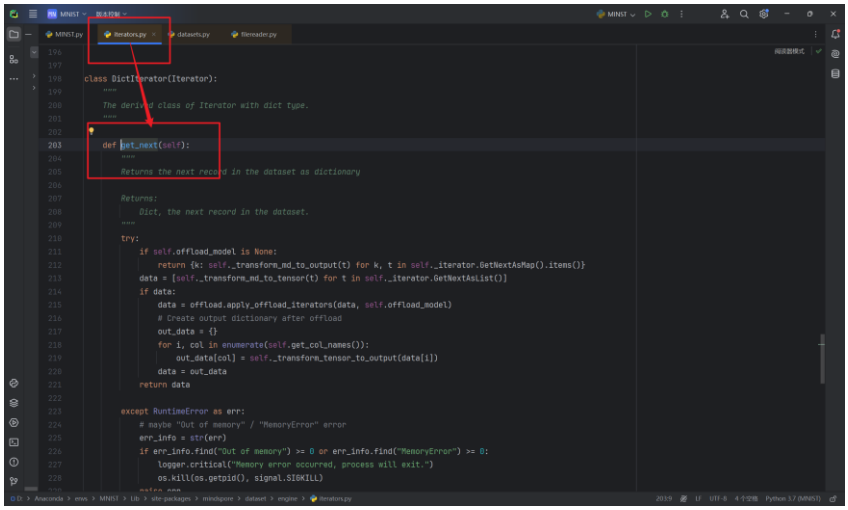


图 31 重要调试步骤

（四）代码运行

通过运行代码，最终得到的训练结果准确率为 0.816，训练步数为 1875 次，epoch 为第十次。

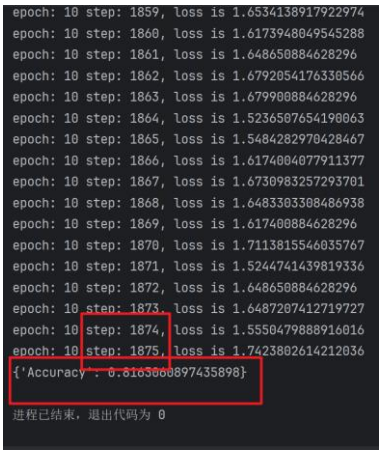


图 32 训练参数

通过训练后的模型，我们可以看到，模型能够准确地预测我们随机抽取的 10 个手写体数字，预测准确率为 100%

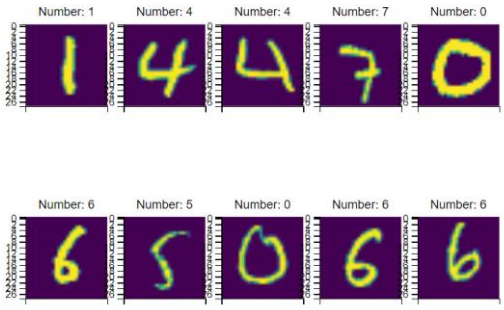


图 33 模型预测结果

原本在初始，我选择 MindSpore 深度学习框架进行手写体检测实验，是想通过猫狗分类实验的 call converter_lite—fmk=MINDIR--modelFile=d:\MindSporePetClassifiCation\code\mobilenetv2.mindir --outputFile=pet 指令将模型文件也转化成 write.ms 文件，然后进行本地部署，同时通过 Mindapp 进行可视化预测手写体，进一步观察模型训练的准确率是否达到可用的要求。可惜在我尝试的过程中由于时间不足和对部署掌握的不足，暂时没能将两个实验都部署至同一个 app。

结论

本文基于华为推出的全场景深度学习框架 MindSpore，对图像分类和手写体图像识别两个关键任务进行了深入的研究和实验。通过构建和优化深度学习模型，我们成功验证了 MindSpore 在处理这些任务时的强大能力和卓越性能。

在猫狗分类实验中，我们利用 MindSpore 的灵活性和高效性，快速构建了一个能够准确区分猫和狗图像的深度学习模型，并通过优化算法和参数设置，实现了较高的分类准确率。通过本地部署，我们验证了猫狗检测模型的准确性很高；通过修改 early stop 等参数，我对不同参数对模型造成的影响作了讨论。比如提高 early stop 参数能够使模型迭代更多次，收敛的区间更小等结论。这些成果证明了 MindSpore 在图像分类领域的有效性。

在手写体图像识别实验中，我们利用 MindSpore 对经典的 MNIST 数据集进行了训练和测试，通过构建和优化卷积神经网络模型，实现了对手写数字图像的高精度识别。虽然没有实现将训练的模型像猫狗实验一样转化成 pet.ms 文件并实现本地部署，但是我也在实现代码模型的过程中体会了从原理去理解，代码从环境部署到 Debug 的完整阶段。这考验了我们的耐心与耐力。

通过这两个实验，我们感受到了 MindSpore 作为一款全场景深度学习框架的魅力和优势。其卓越的计算性能、灵活的 API 接口以及全场景统一部署的能力。同时也首次接触了神经网络模型的相关项目，虽然只是类似“代码复现”的步骤，但是能让我们加深理解，为后续自己搭建模型搭建网络框架打下坚实的基础。

展望未来，随着人工智能技术的不断发展和应用场景的不断拓展，我们相信 MindSpore 将在深度学习研究和应用中发挥更加重要的作用。我们期待更多的研究者和开发者能够加入到 MindSpore 的大家庭中，共同推动人工智能技术的创新和发展。同时，我们也希望华为能够继续加大对 MindSpore 的投入和支持，不断完善和优化框架的功能和性能，为深度学习研究和应用提供更加强大的支撑。在此也感谢高灿老师一学期的指导！

【参考文献】

[1]https://blog.csdn.net/m0_53700832/article/details/129958252?ops_request_misc=&request_id=&biz_id=102&utm_term=mindspore%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%E5%AE%9E%E4%BE%8B&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-5-129958252.142^v100^pc_search_result_base5&spm=1018.2226.3001.4187

[2]https://download.csdn.net/download/weixin_40523201/85152959?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-download-2%7Edefault%7EBlogCommendFromBaidu%7EPaid-1-85152959-blog-129958252.235%5Ev43%5Econtrol&depth_1-utm_source=distribute.pc_relevant_t0.none-task-download-2%7Edefault%7EBlogCommendFromBaidu%7EPaid-1-85152959-blog-129958252.235%5Ev43%5Econtrol&utm_relevant_index=1

[3] MindSpore

[4]景晨凯, 宋涛, 庄雷, 刘刚, 王乐, 刘凯伦. 基于深度卷积神经网络的人脸识别技术综述[J]. 计算机应用与软件, 2018, 35(1):223-231

[5]王景. 基于深度学习算法的视频监控人脸识别系统研究[J]. 长江信息通信, 2023, 36(8):128-131

[6]史常浩、李阳、杜立伟、袁林. 基于数字图像处理技术在重点河道路口的管控[A].北京市永定河管理处, DOI:10.19695/j.cnki.cn12-1369.2023.07.27.

【附录】

一、字数：6900

二、由于先前未开始写论文的时候就做了很多额外的工作,并且我觉得这些工作和思考都很有意义，所以大部分以图表的方式呈现在论文中，导致论文的页码超出了 15 页，希望老师能理解。

三、压缩包内容：

1. 代码包：分为手写体代码包与 MindSporepet 代码包
2. 论文 pdf 格式一份；
3. 数据处理 excel 表格；
4. 预测结果图片文件夹；
5. 部署过程图片文件夹。