实验 1 使用华为云 ModelArts 进行 AI 流程开发

1.实验的目的， 意义

1.1 实验目的

熟悉 AI 的通用开发流程以及熟悉华为云平台使用。

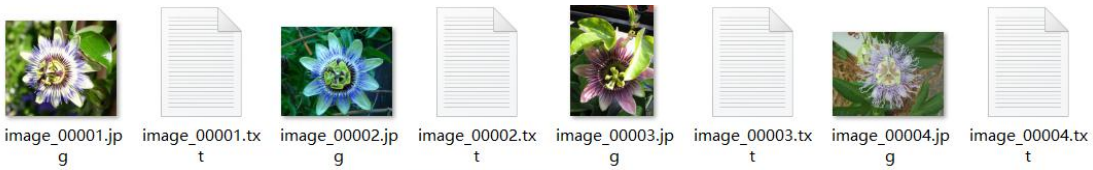
1.2 实验的意义

掌握准备模型所需数据集及运用相关工具处理数据集的方法，熟悉在云平台上训练和部署 AI 模型的一般流程，体会在华为云 ModelArts 平台开发和本地开发的不同之处，为之后的开发打下基础。

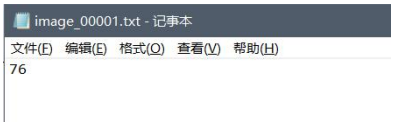
2.记录自己的实验过程（要有必要的截图以及文字说明）

2.1 准备数据

先准备好实验所需的数据集，将所有的花卉图片拷贝到 dataset 文件夹中，并为每个样本图片生成标签，标签是命名与对应图片名称相同的 txt 文件，标签的取值为[0,101]，对应着数据集中的 102 种花卉。



如 image_00001.jpg 对应的花卉编号为 76，所以 image_00001.txt 的内容为：



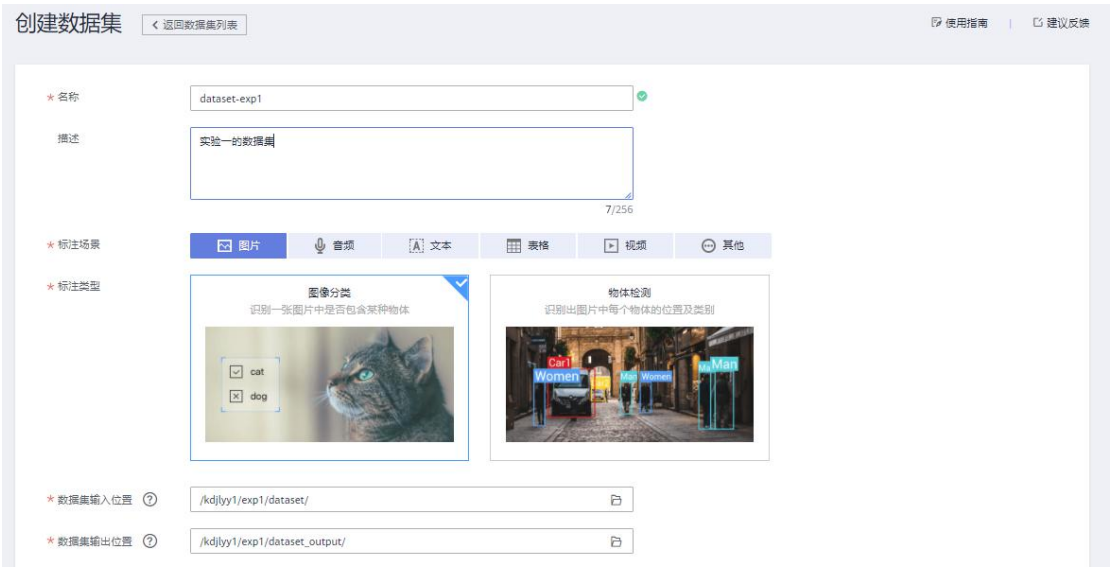
2.2 创建桶并上传数据集

先进入华为云的控制台，建立自己的桶，并建立对应的文件夹，用来存放数据集：

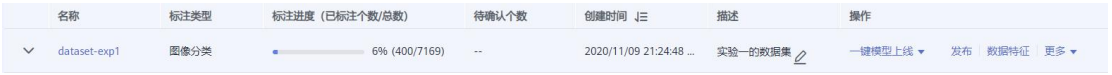
通过 OBS Broswer Plus 软件将本地准备好的数据集上传到桶的对应文件夹中：

2.3 创建数据集并发布

进入华为云的 ModelArts 管理控制台，创建数据集，指定数据集的输入和输出文件夹：



创建成功后会自动进行标注：

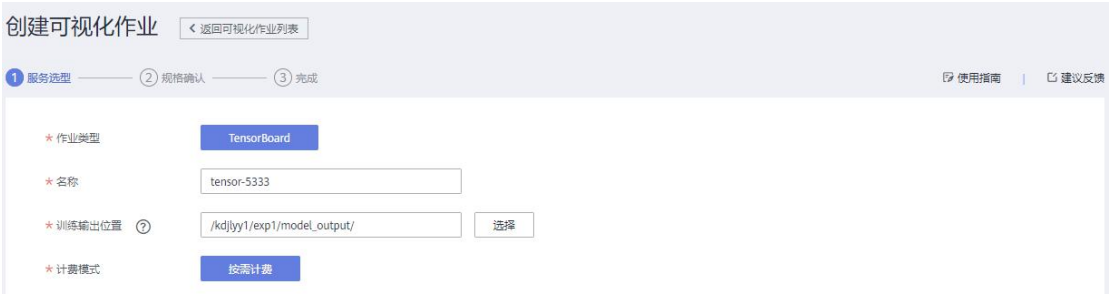


标注完成后划分训练集验证集的比例并发布：

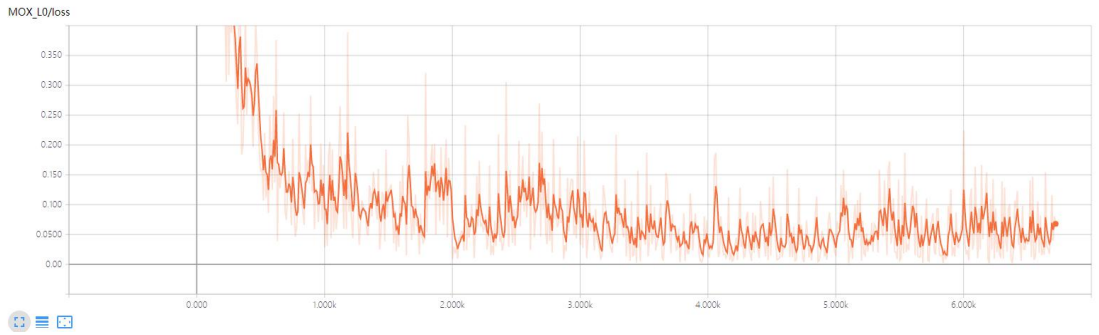
2.4 订阅模型，创建训练作业和可视化作业

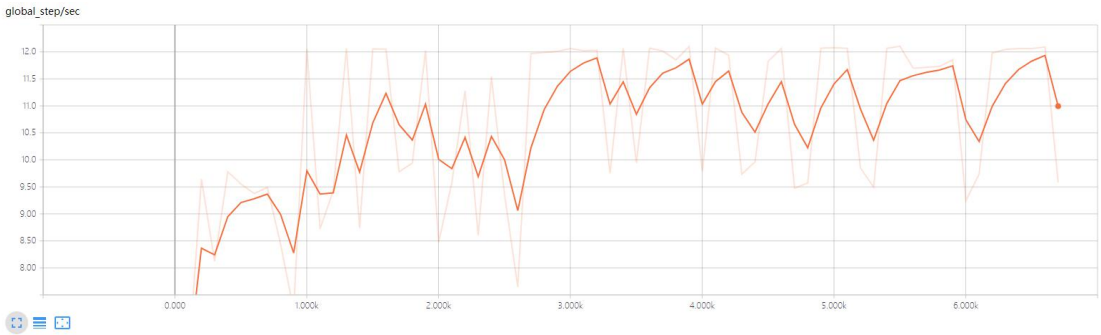
在市场订阅图像分类-ResNet_v1_50 模型，点击创建训练作业，输入之前准备好的数据集和模型输出位置：

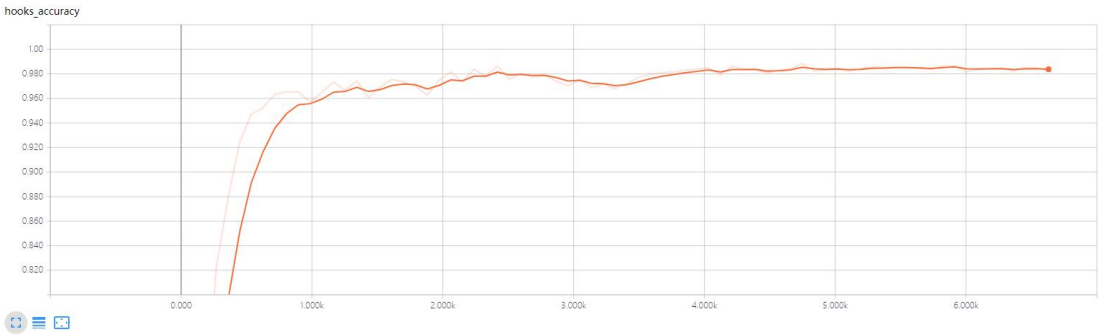
创建可视化作业，训练输出位置与模型输出位置相同：

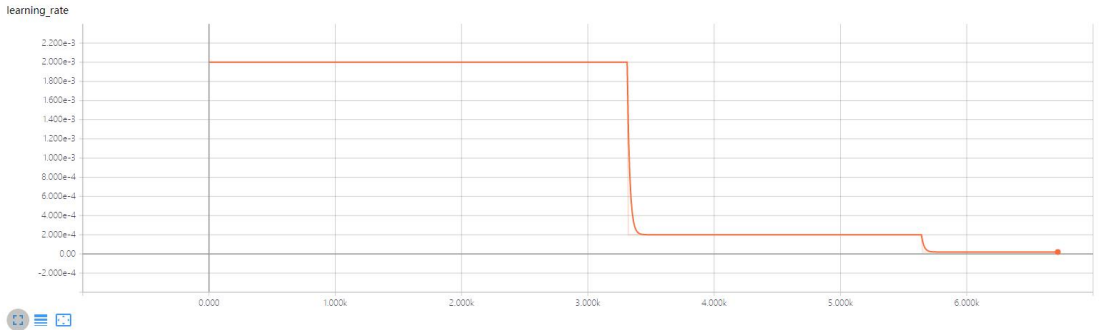


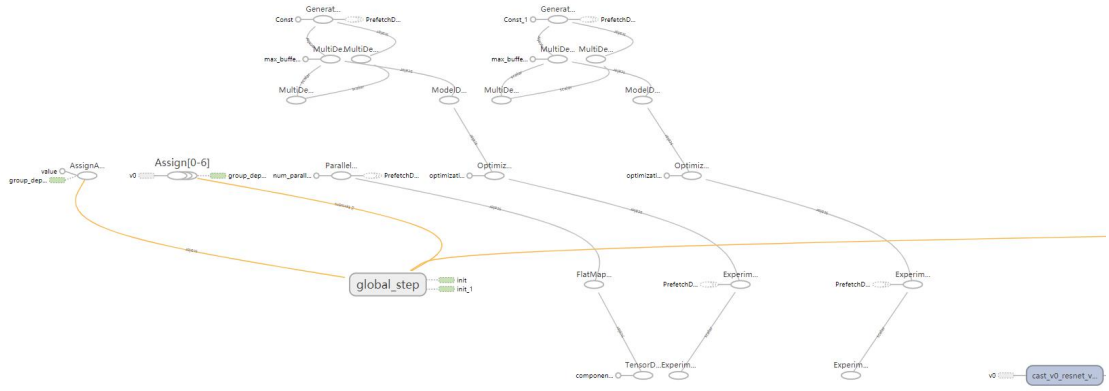
等待模型完成训练，可以在 Tensorboard 上看到模型的各项监控指标和 Tensorflow 可视化的计算图：











2.5 导入模型并部署

模型训练完成后，导入刚才训练好的模型：

并成功部署：

2.6 预测结果

上传测试集的图片，可以在右侧看到用该模型预测的结果：

3.实验过程中碰到的问题，以及你是如何解决的？

3.1 处理数据集的问题。

实验提供的数据集不能直接输入模型，要提前对数据集进行处理：首先将样本集的训练集和验证集的花卉图片拷贝到同一个文件夹下（在 ModelArts 平台训练模型时再划分训练集、验证集），然后利用代码，为每张花卉图片生成对应的标签，写入正确的标签值，完成数据集的准备工作。

3.2 生成可视化作业的问题。

第一次生成可视化作业时，等模型训练完成后，打开可视化作业发现 Tensorboard 上没有模型的各项监控指标和可视化的计算图。通过查看官方的教程发现，可视化作业的训练输出位置应该与模型的训练输出位置保持一致，重新生成可视化作业即可正常显示。

4.实验验收时所涉及的问题

4.1 在训练模型之前有没有认真查看数据集，是否发现什么特点？

花卉的种类共有 102 种，但是各个种类之间的样本数量差距比较大，部分种类花卉的样本总数在 30 张左右，有些种类的花卉样本数量达到 200 多。数据集的花卉图片特征比较清晰，噪声比较小，无太多干扰项。但是少数种类的花卉在颜色、形状上的个体差异比较大。

4.2 根据本次实验，简述在云平台进行 AI 开发的一般流程。

首先，要准备训练模型所需要的训练集并生成对应的标签，然后将模型上传到云平台的对应文件夹中，再在云平台创建数据集，对数据进行标注。数据集创建完成后需要订阅模型创建作业，输入准备好的数据集和各项超参来训练模型，并可以在 Tensorboard 上查看模型的训练结果和各项参数，最后可以用训练好的模型在测试集上进行测试。

4.3 如果训练出的模型太大，怎样进行模型压缩？

进行模型压缩的方法主要分为两大类：采用新的卷积计算方法和在已训练好的模型上做裁剪。

① 采用新的卷积计算方法：这种方法直接修改网络结构或者使用新的卷积计算方式，从而减少参数，达到压缩模型的效果，例如 SqueezedNet，MobileNet。

② 在已训练好的模型上做裁剪：可以通过剪枝、权值共享、量化、神经网络二值化等方法实现，具体如下：

剪枝：神经网络是由一层一层的节点通过边连接，每个边上会有权重，所谓剪枝，就是当我们发现某些边上的权重很小，可以认为这样的边不重要，进而可以去掉这些边。在训练的过程中，在训练完大模型之后，看看哪些边的权值比较小，把这些边去掉，然后继续训练模型；

权值共享：就是让一些边共用一个权值，达到缩减参数个数的目的。假设相邻两层之间是全连接，每层有 1000 个节点，那么这两层之间就有 1000*1000=100 万个权重参数。可以将这一百万个权值做聚类，利用每一类的均值代替这一类中的每个权值大小，这样同属于一类的很多边共享相同的权值，假设把一百万个权值聚成一千类，则可以把参数个数从一百万降到一千个；

量化：一般而言，神经网络模型的参数都是用的 32bit 长度的浮点型数表示，实际上不需要保留那么高的精度，可以通过量化，比如用 0~255 表示原来 32 个 bit 所表示的精度，通过牺牲精度来降低每一个权值所需要占用的空间；

神经网络二值化：比量化更为极致的做法就是神经网络二值化，也即将所有的权值不用浮点数表示了，用二进制的数表示，要么是+1,要么是-1，用二进制的方式表示，原来一个 32bit 权值现在只需要一个 bit 就可以表示，可以大大减小模型尺寸。