

项目综合实践

识花君篇

花朵识别问题

- 花朵识别

- 我们使用预训练模型 Inception-V3 对花朵图像进行分类， Inception-V3 模型是谷歌在大型图像数据库 ImageNet 上训练好了一个图像分类模型，这个模型可以对 1000 种类别的图片进行图像分类。但现成的 Inception-V3 无法对“花”类别图片做进一步细分，因此本实验的花朵识别实验是在 Inception-V3 模型基础上采用迁移学习方式完成对“花”类别图片进一步细分的实验。

迁移学习

- 所谓迁移学习，就是将一个分类上训练好的模型通过简单的调整使其适用于一个新的图像分类问题，在训练好的 Inception-v3 模型中，通过卷积输出的节点向量再通过一个单层的全连接层神经网络就可以区分 1000 种类别的图像，所以在我们的实验中只需要把卷积层输出的节点向量作为输入来训练一个新的单层全连接神经网络处理“花”的分类问题。

- 我们采用预训练模型 Inception-V3 进行学习迁移后对五种品种的花朵识别，具体过程
 - 用户在手机 Web 端启用拍照功能，拍一张花的图像，经 Http 传输到服务器端，后端采用 Inception-V3 的学习迁移模型对花朵图像进行识别，把识别结果返回给手机 Web 端

项目实施步骤 -1

- 收集五种品种的花
 - 采用手机拍照功能，对五个品种的花进行图像收集
 - 要求：每种花至少 100 张图片，可以对同一朵花从不同角度，不同时间段进行多次拍照，也可以对同一品种不同的花朵进行多次拍照。
 - 对采集的花朵进行中心区域裁剪，并把图片缩放成 299*299 大小的图片
 - 要求：在识别使用阶段采用 canvas 对图像进行中心区域裁剪和缩放，在数据训练阶段，可以在项目实施步骤 -2 的数据增强部分统一处理。

项目实施步骤 -2

- 花朵图像数量增强：对于 Inception-V3 模型进行学习迁移，每种种类的花朵图片 100 还是不够，我们需要通过程序手段在已有的图像中再制造出更多花朵图像给我们的模型进行学习
 - 对采集的花朵进行中心区域裁剪，并把图片缩放成 299×299 大小的图片，并按每个种类一个文件夹，把缩放的图片存入到对应分类文件夹中
 - 对文件夹中的每张图像进行上下反转、左右反转、沿对角线反转、亮度饱和度和色相随机调整，每个文件夹中最终存放花朵（包括原始图像） 500 张图像

项目实施步骤 -3

- 迁移学习 Inception-V3 模型
 - Inception-V3 模型是谷歌在大型图像数据库 ImageNet 上训练好了一个图像分类模型
 - 下载 data_process.py 到程序文件夹和 tensorflow_inception_graph.pb 到程序文件夹下 “inceptionV3” 文件夹中，把花朵种类对应的文件夹拷贝到程序文件夹下的 “flower_photos” 文件夹中
 - 命令行运行 `python data_process.py`，该程序通过 Inception-V3 模型对图像进行处理，把卷积层的节点向量输出和对应花朵类型，分成训练、测试、验证三组数据写入到 `flower_processed_data.npy` 文件中
 - 下载 train.py 到程序文件夹
 - 命令行运行 `python train.py`，该程序构建了一个新的单层全连接神经网络处理 “花” 的分类问题，程序通过读取 `flower_processed_data.npy` 的数据对新的网络进行训练和测试

项目实施步骤 -4

- 构建后端花朵图片识别服务
 - 服务端采用 Http 接收 Web 端图片数据，并调用花朵图片识别函数进行识别
 - 要求：后端接受 Web 端上传的 PNG 或 JPG 图片，并把上传的图片存放到临时目录中
 - 构建基于 Inception-V3 迁移学习模型的花朵图片识别函数
 - 要求：识别函数传入一个 299*299 的彩色图片的文件路径，返回花朵的名称

实验进度安排

- 第一周
 - 采用手机相机采集 5 种品种的花朵，花朵品种自己选择，每种花朵 100 张图片
- 第二周
 - 编写 python 程序对采集的花朵进行采集和图片数量增强
- 第三周
 - 运行 data_Process.py 程序对图片进行预处理并保存到 flower_processed_data.npy 文件中，运行 train.py 程序进行迁移学习
- 第四周
 - 编写手机 Web 端花朵拍照和图片裁剪功能，并文件上传后端用 matplotlib 库把图片显示出来
- 第五周
 - 完成基于 Inception-V3 的花朵图片识别功能

学习资源 -1

- 加载二进制模型
 - [https://
www.tensorflow.org/versions/r1.11/api_docs/python/tf/import_graph_def?hl=cn](https://www.tensorflow.org/versions/r1.11/api_docs/python/tf/import_graph_def?hl=cn)
- TensorFlow 图像处理函数
 - https://tensorflow.google.cn/api_docs/python/tf/image
 - 左右反转函数 flip_left_right
 - 上下反转函数 flip_up_down
 - 对角线反转函数 transpose_image
 - 调整图像亮度、饱和度、色相函数 adjust_brightness、 adjust_saturation、 adjust_hue