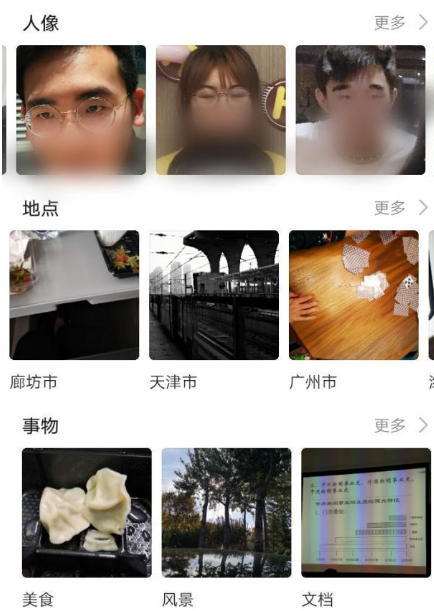
# 项目简介

不知你们是否有过这样的烦恼，每次节假日或者特殊的节日跟家人朋友游玩结束疲惫回到家后，还要自己手工从一堆拍摄的杂乱照片中分类整理出来哪些是风景照，哪些是人物合照，哪些是美食照...，面对每次出去拍摄回来的成百上千张照片，恨不得请个人来专门整理，然而又有谁能真正了解自己所想？正所谓科技改变生活，其实现如今许多智能手机自带的AI照片识别就可以帮到我们。



苹果手机简单的相册分类功能（非AI）



华为P30智能相册（AI）

虽然部分手机拥有了智能相册分类功能，但是不免有些人对照片高质量的追求者还会在后期使用Adobe Photoshop对照片进行美化休整，那么面对PC端的相册整理任务，我们是不是好像又回到了问题的起点呢？

基于上述问题痛点，本项目结合飞桨PaddleClas提出了利用图像分类技术训练常见照片图像分类模型解决PC端照片智能分类问题。飞桨图像识别套件PaddleClas是飞桨为工业界和学术界所准备的一个图像识别任务的工具集，助力使用者训练出更好的视觉模型和应用落地。



项目中利用图像模型智能分类流程

本项目完全基于飞桨PaddleClas开发套件，采用ResNet50\_vd模型在小数据量短训练时长下在AI Studio上实现模型的训练学习，后期经过测试，模型在测试集上准确率高达96.5%，在少量新数据图片情况下测试也可以正确分类。接下来将对项目进行详细介绍：

1. 环境配置
2. 数据集选取及处理
3. 模型选择及参数配置
4. 模型训练及测试
5. WatchDog文件监控
6. 智能相册本地配置使用

# 环境配置

环境要求：

Python3

CUDA == 10.1

cuDNN ==7.6.5

PaddlePaddle v2.0或更高版本

PaddleX == 2.0.0

安装过程

安装PaddleClas套件：

1. *# 下载PaddleClas套件源码*
2. git clone https://gitee.com/paddlepaddle/PaddleClas.git -b develop

安装套件依赖库：

1. cd PaddleClas
2. pip install --upgrade -r requirements.txt -i https://mirror.baidu.com/pypi/simple
3. pip install paddlex -q

# 数据集选取及处理

出于我们日常拍摄照片种类的众多，项目优先选取了：动物、食物、任务、风景、文本这样一个五分类的数据，该数据完全采用公开数据，数据干净且清晰：



部分食物分类数据集展示

由于公开的原始数据集的庞大不利于我们缩短训练快速开发模型，所以整个项目选取了每个类别的部分数据进行使用，并对选取的数据进行了统计，统计结果如下表：



原始公开数据集：

风景数据集：https://github.com/yuweiming70/Landscape-Dataset

食物数据集：https://www.datafountain.cn/datasets/5423

人物数据集：http://www.seeprettyface.com/mydataset\_page2.html

文本数据集：https://challenge.xfyun.cn/topic/info?type=ad-2021&ch=dc-wbzmt-05

动物数据集：https://blog.csdn.net/sunshine\_\_sun/article/details/115558151

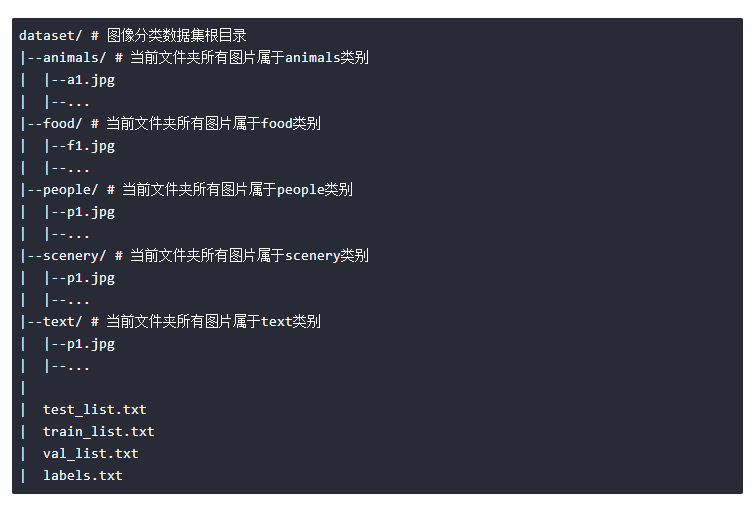
整理后的数据集：<https://aistudio.baidu.com/aistudio/datasetdetail/110303>

选取完成数据集以后我们就要对数据集进行处理了，解压数据集到工作文件夹然后利用PaddleX小工具对数据集进行划分：



数据集划分【前】的文件结构

1. *# 数据集切分： 训练：测试：验证=8：1：2*
2. paddlex --split\_dataset --format ImageNet --dataset\_dir './dataset/' --val\_value 0.2 --test\_value 0.1



数据集划分【后】的文件结构

Tips：可以看到，在这里我们使用了PaddleX的一个小工具对数据集进行了划分，PaddleX是一个飞桨全流程开发工具，因为有了它我们划分数据集开始变得更加简单快速！当然，如果你不满足使用它，也可以去参考写一个划分数据集的代码。

# 模型选择与参数配置

数据预处理之后，需要选择并训练网络。本项目选择的网络是ResNet50\_vd，关于ResNet网络结构的代码分析，可单击链接：

https://aistudio.baidu.com/aistudio/projectdetail/438756

在PaddleClas/ppcls/configs/quick\_start/ResNet50\_vd.yaml中的修改**必要参数**：

* save\_interval：10 每隔多少个epoch保存模型
* validate：True 是否在训练时评估
* valid\_interval：10 每隔多少个epoch进行模型评估
* epochs：50 训练总epoch数
* classes\_num：5 分类数
* image\_shape： [3, 224, 224] 图片大小

Train：

* image\_root: ./dataset/ 数据集文件
* cls\_label\_path: ./dataset/train\_list.txt 数据集标签文件

Eval：

* image\_root: ./dataset/ 数据集文件
* cls\_label\_path: ./dataset/val\_list.txt 数据集标签文件

Tips：关于学习率的设置问题，大部分的神经网络选择的初始学习率为0.1，batch\_size是256，根据实际的模型大小和显存情况，可以将学习率设置为0.1×k，batch\_size设置为256×k。

修改好配置文件，我们再把labels.txt文件做一个映射修改就可以训练了：



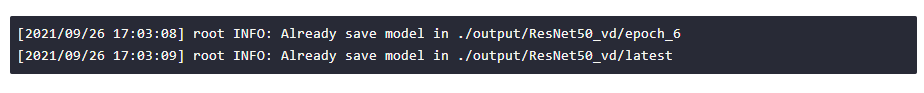
类别标签映射（1映射改成2）

当然，如果你想要让你的模型更加优秀，性能更加出众的话，你可以在PaddleClas技术文档找到更多的训练技巧和参数设置可以关注PaddleClas技术文档，其对相关参数进行了详细的解释说明：<https://paddleclas.readthedocs.io/zh_CN/latest/models/Tricks.html>

# 模型训练及测试

使用已经配置好的训练文件就可以执行模型的训练了，指令如下：

1. python3 tools/train.py \
2. -c ./ppcls/configs/quick\_start/ResNet50\_vd.yaml \
3. -o Arch.pretrained=True



每个epochs训练完成后，保存评估的最好模型

本次项目中设置的是阶段性模型评估，同时保存评估结果最好的模型，参数在下述目录：

PaddleClas/output/ResNet50\_vd/best\_model。best\_model文件夹下包含ppcls.pdmodel、ppcls.pdopt、ppcls.pdparams三个文件用来进行后续的评估测试推理使用。

在训练完成后，使用训练评估保存的最好模型去预测一张随机找到的食物照片看一下测试结果：

1. python tools/infer.py -c ./ppcls/configs/quick\_start/ResNet50\_vd.yaml \
2. -o Infer.infer\_imgs=./demo/food/1043283.jpg \
3. -o Global.pretrained\_model=./output/ResNet50\_vd/latest/latest



推理图片及结果

# WatchDog文件监控

分类模型已经训练好了，但是本地使用该模型还需要更加自动化的操作，这里项目使用了一个文件监控库-WatchDog，那么WatchDog是什么呢？WatchDog用来监控指定目录/文件的变化，如添加删除文件或目录、修改文件内容、重命名文件或目录等，每种变化都会产生一个事件，且有一个特定的事件类与之对应，然后再通过事件处理类来处理对应的事件，怎么样处理事件完全可以自定义，只需继承事件处理类的基类并重写对应实例方法。

上面这段文字你可能有些迷茫，还是不知道WatchDog是干什么的，那么形象举个例子来说，我们把文件监控看作是两个部分，一个叫做Listener（松鼠）和Handler（老虎），当Listener（松鼠）站在书上看着远处，如果来了一个事件（狮子）到来，那么Listener告诉Handler（老虎）让它去执行准备战斗的操作，如果看到了另外一个事件（人）到来，Listener（松鼠）会告诉Handler（老虎）让它执行逃跑操作。



形象化的WatchDog理论讲解（比喻）

如上描述，我们把一个文件夹使用WatchDog做成一个Listener，然后当Listener监控到有照片文件放进来就会触发Paddle模型去预测照片种类，并将该照片移动到相应的相册分类文件夹中。

如果你对WatchDog感兴趣可以运行下以下一个Demo程序：

1. from watchdog.observers import Observer
2. from watchdog.events import FileSystemEventHandler
3. import time
4. import os
5. folder\_to\_track = "python\\movingFiles\\src"
6. folder\_destination\_pic = "python\\movingFiles\\des\\pic"
7. folder\_destination\_music = "python\\movingFiles\\des\\music"
8. folder\_destination\_kaoyan = "python\\movingFiles\\des\\kaoyan"
9. class MyHandler(FileSystemEventHandler):
10. def on\_any\_event(self, event):
11. for filename in os.listdir(folder\_to\_track) :
12. src = folder\_to\_track +"\\"+ filename
13. if filename.endswith("jpg") or filename.endswith("png"):
14. new\_destination=folder\_destination\_pic + "\\"+filename
15. elif filename.endswith("mp3") :
16. new\_destination=folder\_destination\_music + "\\" + filename
17. elif filename.\_\_contains\_\_("ziliao") :
18. new\_destination = folder\_destination\_kaoyan + "\\"+filename
19. os.rename(src, new\_destination)
20. event\_handler = MyHandler()
21. observer = Observer()
22. observer.schedule(event\_handler, folder\_to\_track, recursive=True)
23. observer.start()
24. try:
25. while True:
26. time.sleep(10)
27. except KeyboardInterrupt:
28. observer.stop()
29. observer.join()

# 智能相册本地配置使用

借助之前完成的模型开发和WatchDog监控，项目就可以部署在本地使用了。在这里我们需要完成之前的核心文件监控代码：

1. class MyHandler(FileSystemEventHandler):
2. def on\_any\_event(self, event):
3. for filename in os.listdir(folder\_to\_track) :
4. src = folder\_to\_track +"/"+ filename
5. if filename.endswith("jpg") or filename.endswith("png"):
6. *#new\_destination=folder\_destination\_picture + "/"+filename*
7. print(filename)
8. os.system(r"python PaddleClas/tools/infer.py -c ./PaddleClas/ppcls/configs/quick\_start/ResNet50\_vd.yaml -o Infer.infer\_imgs=./src/"+str(filename)+" -o Global.pretrained\_model=./PaddleClas/output/ResNet50\_vd/latest/latest > ./result.txt")
9. with open('result.txt', 'r', encoding='utf-8') as f:  *# 打开文件*
10. lines = f.readlines()  *# 读取所有行*
11. last\_line = lines[-1]  *# 取最后一行*
12. *#print(eval(last\_line))*
13. result = eval(last\_line)[0]['label\_names'][0]
14. *#print(type(eval(last\_line)))*
15. if result=='animals':
16. new\_destination=animals + "/" +filename
17. if result=='people':
18. new\_destination=people + "/" +filename
19. if result=='food':
20. new\_destination=food + "/" +filename
21. if result=='scenery':
22. new\_destination=scenery + "/" +filename
23. if result=='text':
24. new\_destination=text + "/" +filename
25. elif filename.endswith("mp3") :
26. new\_destination=folder\_destination\_music + "/" + filename
27. elif filename.endswith("avi") or filename.endswith("mp4"):
28. new\_destination = folder\_destination\_video + "/"+filename
29. os.rename(src, new\_destination)

可以部署本地使用的整体项目目录结构：



项目代码结构

# 心得体会

本项目使用了飞桨开源深度学习框架以及PaddleClas套件，在AI Studio上完成了数据处理、模型训练、模型评估推理等工作。PaddleClas套件让图像分类技术变得更为简单便捷，降低了开发者的上手难度。

项目开源地址：<https://aistudio.baidu.com/aistudio/projectdetail/2408523>