

实现一个如下要求的简易图像检索系统：

1. 在标准的 resnet18 的 classifier 前增加一层64维的全连接层
2. 学习 FAISS 的使用方法 <https://github.com/facebookresearch/faiss>
3. 实现如下的方法：

```
def extract_feature(model, img):  
    '''  
    用模型提取一张图片的特征向量  
    param:  
        model 为上面定义的模型  
        img 为一张图片  
    return:  
        fc_feat 为上面模型中定义的64维全连接层的输出  
        conv_feat 为最后一层卷积层输出的特征图，并变形成一条一维向量  
        classify 为上面模型的预测向量  
    '''  
    return fc_feat, conv_feat, classify  
  
def extract(model, img_folder, file_name):  
    '''  
    提取一个文件夹中的所有图片的特征，并以 .h5格式存储到指定位置  
    param:  
        model 为上面定义的模型  
        img_folder 为一个文件夹路径  
        file_name 为存储的文件地址  
    return:  
        无返回  
    '''  
  
def search(gallery, query):  
    '''  
    返回 gallery 中与 query 最近邻的3个结果  
    param:  
        gallery 为一个 faiss 对象  
        query 为一张图片的 fc_feat向量  
    return:  
        res 为一个 list，对应前三个最近邻图片的名称  
    '''  
    return res
```

4. 准备一个有5张图片的文件夹，用上面的方面提取这个文件夹里图片的特征并存储
5. 用 Flask在本地实现一个服务接口，接口接收一张图片的 base64编码，返回这张图片的前三个最近邻图片的名称（也可以不用 Flask，选自己熟悉的即可）
6. 写一个可以运行的脚本程序来演示上述服务接口查询

7. 以上功能实现后，做好必要的注释，将代码打包发到 [taujiang@outlook.com](mailto:taujiang@outlook.com)