人脸识别技术文档

(本技术文档根据 系统：Win10，显卡：GTX1060 )

本项目在Github上拥有源码以及文档，求star⭐。项目名：face\_dection-AI-

Github：<https://github.com/bijinmengxiang/face_recognition-AI-.git>

1.准备工作：

1.1 安装运行环境： spyder + pytorch

1.1.1 首先安装anaconda

建议在创建完python环境后再Anaconda Navigator中安装spyder环境

（当然你可以使用base空间先进去看看spyder，极度不建议把东西都装在base）

1.1.2 由于anaconda中可以创建多种不同库的编译环境，首先进入anaconda prompt，（通过conda env list命令查看现所有环境）

1.创建python环境：conda create –n Env\_Name python=3.7

（如遇提示随便看看yes一路）

（删除空间：conda remove –n Env\_Name --all ）

(清华源加速：添加通道

conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/

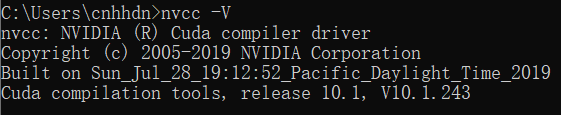
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/

conda config --set show\_channel\_urls yes）

2.安装CUDA与cuDNN(pytorch使用gpu进行运算所需组件)

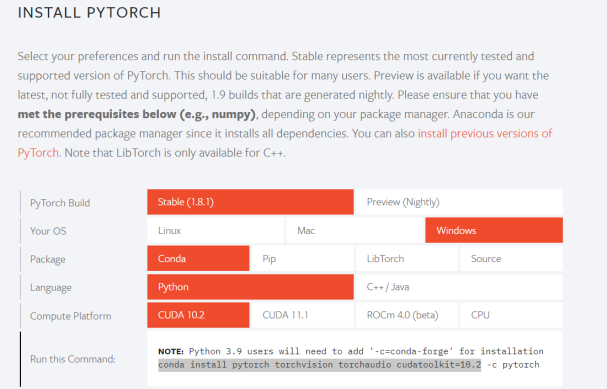
可参考<https://blog.csdn.net/sinat_23619409/article/details/84202651>

安装前需要对显卡最高支持的CUDA版本以及显卡中是否有CUDA单元进行查找，且版本需要相互对应！

<-安装完CUDA后进行的检验命令

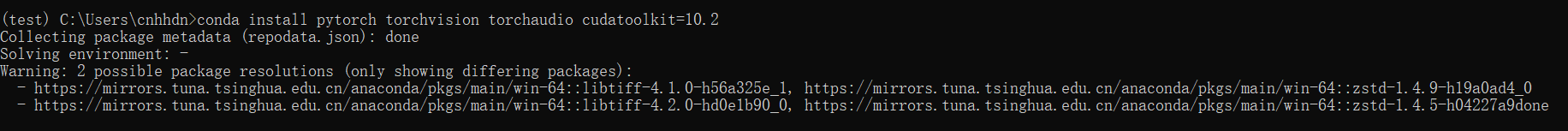
3.搭建pytorch环境（进入<https://pytorch.org/> 网站根据需求选择）

本例进行GPU版本安装



根据深颜色命令进行输入（注意将-c pytorch删除，否则使用官方源进行下载）

（安装前可以确认是否使用清华源安装，下图为实例，耗时可能较久，耐心）



检验GPU环境是否安装完成：（括号内为每一步所输入的命令或代码）

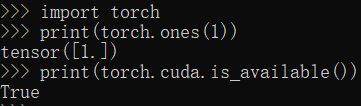
1. 一路pass下载完后激活对应的环境（activate test）

2. 输入python进入python环境中（python）

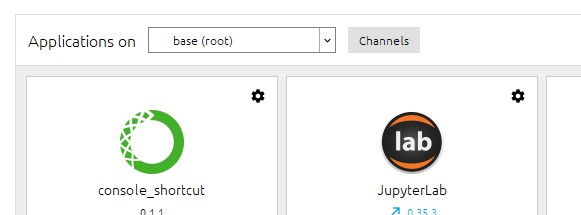
3. 首先检验pytorch是否安装成功，引入pytorch（import torch）

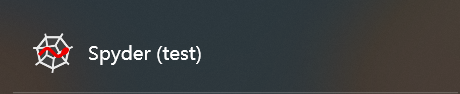
4. 输出单位矩阵（print(torch.ones(1))）

5. 查看GPU环境是否成功搭建（print(torch.cuda.is\_available())）True or False



安装对应环境的spyder：

在此选择环境，对这spyder直接install即可！

安装完成后：

1.2 数据集下载：

Pytorch中现成数据集：

测试用的Mnist手写数字集(可直接使用pytorch代码进行下载，代码见附件)

同样：pytorch中的torchvision.datasets库中还包含有以下数据集，能够较为方便的进行实验。

COCO（用于图像标注和目标检测）：专为对象检测、分割、人体关键点检测、语义分割和字幕生成而设计。

LSUN Classification ：是一个场景理解图像数据集，主要包含了卧室、固房、客厅、教室等场景图像。

Imagenet-12：拥有1000类的物品识别

CIFAR10 and CIFAR100 ：CIFAR-10由10个类60000个32x32彩色图像组成。CIFAR100每个类分别包含500，100个训练与测试图像。

CIFAR-100中的100个类被分成20个超类。每个图像都带有一个“精细”标签（它所属的类）和一个“粗糙”标签（它所属的超类）

|  |  |  |  |
| --- | --- | --- | --- |
| 水生哺乳动物 | 海狸，海豚，水獭，海豹，鲸鱼 | 大自然的户外场景 | 云，森林，山，平原，海 |
| 鱼 | 水族馆的鱼，比目鱼，射线，鲨鱼，鳟鱼 | 大杂食动物和食草动物 | 骆驼，牛，黑猩猩，大象，袋鼠 |
| 花卉 | 兰花，罂粟花，玫瑰，向日葵，郁金香 | 中型哺乳动物 | 狐狸，豪猪，负鼠，浣熊，臭鼬 |
| 食品容器 | 瓶子，碗，罐子，杯子，盘子 | 非昆虫无脊椎动物 | 螃蟹，龙虾，蜗牛，蜘蛛，蠕虫 |
| 水果和蔬菜 | 苹果，蘑菇，橘子，梨，甜椒 | 人 | 宝贝，男孩，女孩，男人，女人 |
| 家用电器 | 时钟，电脑键盘，台灯，电话机，电视机 | 爬行动物 | 鳄鱼，恐龙，蜥蜴，蛇，乌龟 |
| 家用家具 | 床，椅子，沙发，桌子，衣柜 | 小型哺乳动物 | 仓鼠，老鼠，兔子，母老虎，松鼠 |
| 昆虫 | 蜜蜂，甲虫，蝴蝶，毛虫，蟑螂 | 树木 | 枫树，橡树，棕榈，松树，柳树 |
| 大型食肉动物 | 熊，豹，狮子，老虎，狼 | 车辆1 | 自行车，公共汽车，摩托车，皮卡车，火车 |
| 大型人造户外用品 | 桥，城堡，房子，路，摩天大楼 | 车辆2 | 割草机，火箭，有轨电车，坦克，拖拉机 |

STL10 ：STL-10数据集是一个用于开发无监督特征学习、深度学习、自学学习算法的图像识别数据集。

从网络中自己寻找数据集：

<https://zhuanlan.zhihu.com/p/31378836> （人脸识别常用数据集下载，更新中）

<https://www.jianshu.com/p/44311bccbbc6> （人脸识别常用数据集下载，更新中）

从中选取几个具有特征性的数据集进行描述：

1. Labeled Faces in the Wild Home：被广泛应用于评价Face Verification算法的性能。

2.PubFig: Public Figures Face Database：主要用于非限制场景下的人脸识别。

需引用：Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar,  
*International Conference on Computer Vision (ICCV), 2009.*

3.IMDB-WIKI：每张图片都被标注了人的年龄和性别，对于年龄识别和性别识别的研究有着重要的意义。

百度大脑的AI社区中也拥有数据集，需自己寻找验证。

还能够通过爬虫从网络中形成对目标图片的搜索，下载。

然后通过分类器将带有人脸的图像进行切割。（搜索下载与切割源代码在github中）

文件路径为：face\_recognition-AI-/face-detection/crop.py(检测时需要xml文件)

face\_recognition-AI-/face-detection/ DownloadIamge.py（下载）

和crop.py在同一个文件夹中的 crop\_changed.py是本人改写的切割算法，可以防止因人脸过小或一些其他原因引起的图片尺寸错误，使得网络无法自动化训练。改进后进一步提高了自动化程度，无需对尺寸进行手动筛选

下载完数据集后如何生成标准的数据集进行使用：

数据集中是图片类型，并已经分好类

参考：<https://blog.csdn.net/zzy_pphz/article/details/104711382>

使用方法：（Process2Selfdata.py）

1. 使用ImageFolder库将图片转换为数据集

2. 加载上方数据集，参数介绍在mnist\_bumyself中注释有介绍

2：数据集的使用，模型的搭建

2.1 First Version

借鉴了Mnist的手写数字网络的思路。

由于四类图片只拥有408张的图片训练集，所以使用RandomHorizontalFlip函数进行随机镜像翻转。在第一次尝试时，使用的关键参数为：learning\_rate = 0.015

momentum = 0.5 训练次数为50次，正确率在测试集上拥有85.22%的正确率。随着下降学习率至0.008以及升高训练次数至80次后，正确率达到了86.2%。

由于在网络中加入了dropout层（随机舍弃神经元进行学习）使得不被随机遗弃的神经元被迫强壮，无法依赖其他神经元。从而减少了网络的过拟合概率，增加了网络的鲁棒性。（思路来自ImageNet Classification with Deep Convolutional Neural Networks by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E.Hinton）

在上述论文中，对提出两种关于解决训练集数据缺乏问题的方法：

1. 将数据集进行一定的翻转或随机切割处理

2. 对数据集中的图片加入一定的噪声处理（形成一张新图片）

使用上述两种方法对网络进行调整训练，能够降低过拟合的程度。

3.论文得到的思路以及想法：

1. Rethinking the Inception Architecture for Computer Vision(Christian Szegedy Google Inc., Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens)

1.1 卷积核大小意味着不同大小的感受野。最后的拼接（全连接）意味着不同尺度特征的融合。（能够通过多个卷积核处理一个目标，产生不同厚度的结果）

1.2 改变stride（步长）采用相同的卷积核，能够得到相同维度的特征

1.3 网络越深后所表达的特征越抽象，所以特征所需要的感受野也需要更大（卷积核更大）

1.4 降低参数的方法：将5\*5的卷积核分解为两个3\*3的卷积核进行操作。将3\*3的卷积核分解为1\*3与3\*1的卷积核进行操作。（理论上：n\*n的卷积核都能够分解为1\*n和n\*1的卷积核进行操作）

1.5 随着网络越来越深，特征图也越来越小，但是特征图的数量越来越多（网络厚度增加）