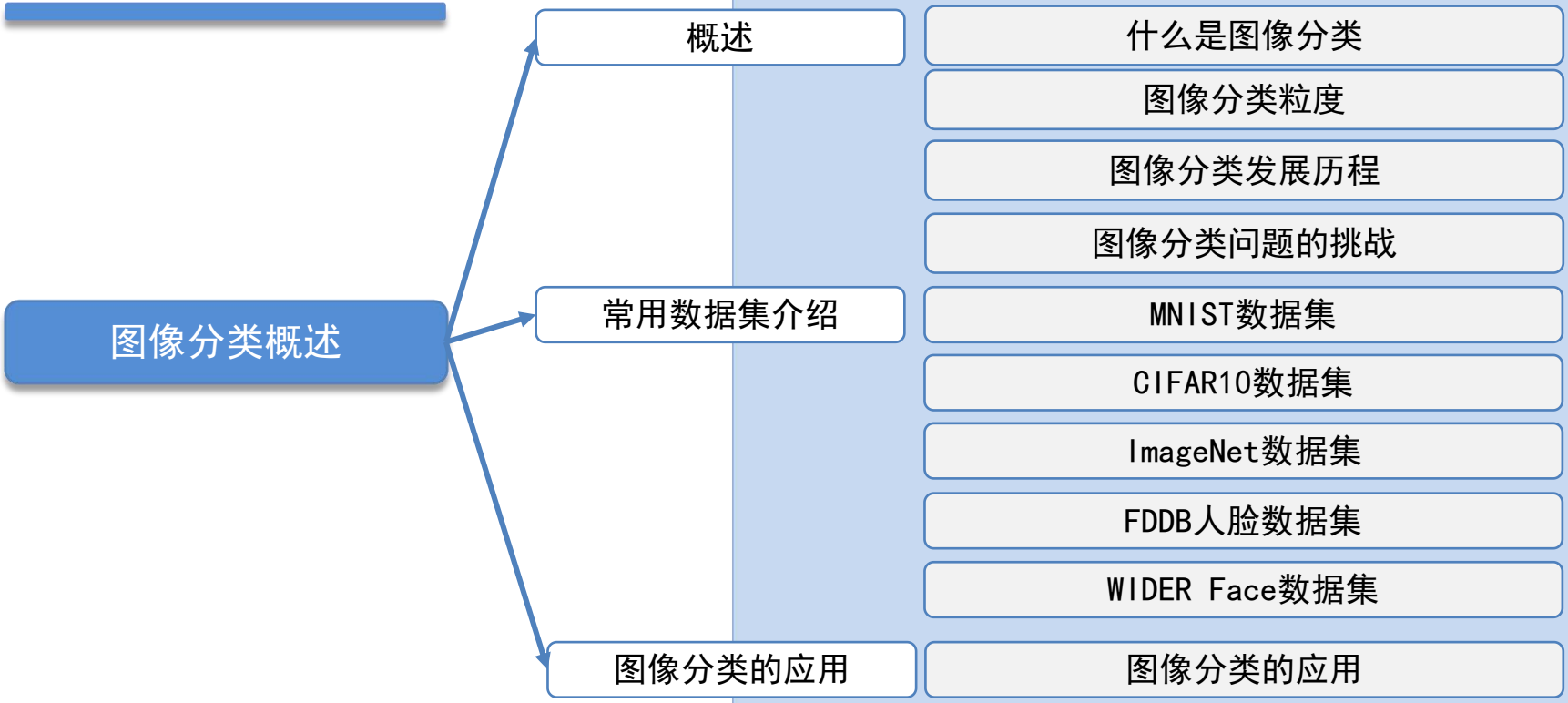
Tedu.cn
达内教育

深度学习

PaddlePaddle图像分类 DAY07

图像分类概述


图像分类概述
概述
什么是图像分类
图像分类粒度
图像分类发展历程
图像分类问题的挑战
常用数据集介绍
MNIST数据集
CIFAR10数据集
ImageNet数据集
FDDB人脸数据集
WIDER Face数据集
图像分类的应用
图像分类的应用

概述

什么是图像分类

知识讲解

- 图像分类就是将不同的图像划分到不同类别，实现最小分类误差、最高精度。手写体识别就是一个经典的图像分类问题，它将输入图像分为0~9某个数字中，实际就是将输入图像分为10类

图像分类粒度

知识讲解

- （一）跨物种级图像分类：在不同物种层次上识别不同对象，如猫狗分类

猫

狗

马

图像分类粒度（续1）

- （二）子类细粒度图像分类：同一大类下，不同子类的分类。如不同的鸟分类，不同的狗分类

知识讲解

京巴

哈士奇

吉娃娃

图像分类粒度（续2）

- （三）实例级图像分类：区分不同的个体。如人脸识别

姜文

章子怡

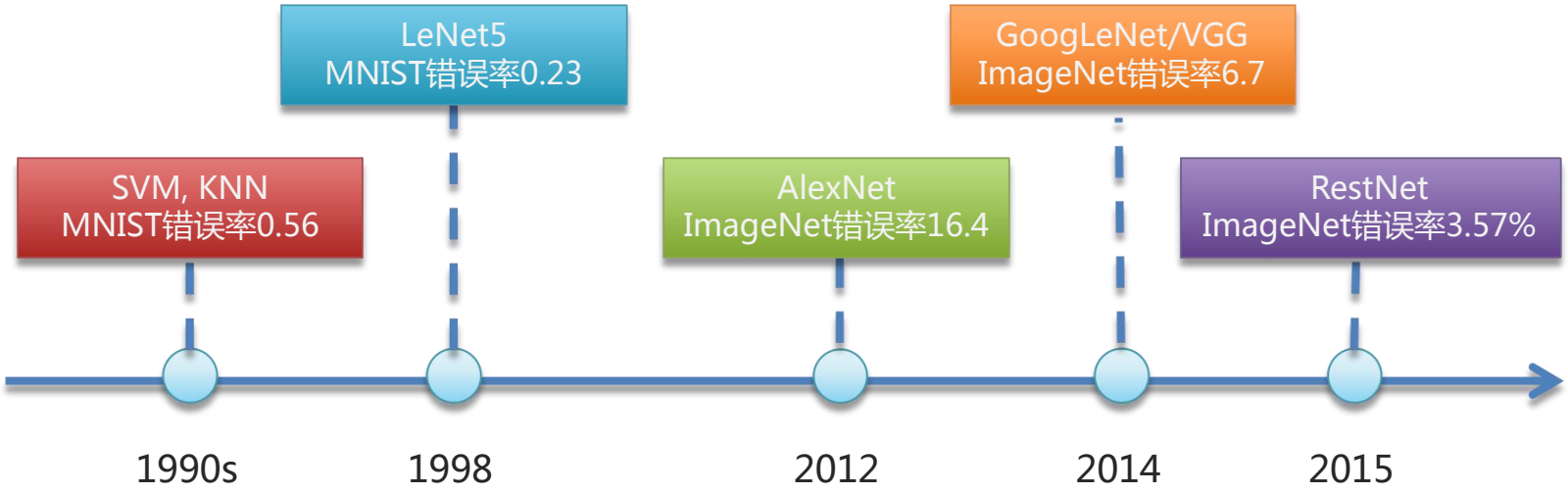
彭于晏

知识讲解

图像分类发展历程

- 图像分类任务从传统的方法到基于深度学习的方法，经历了几十年的发展

知识讲解



图像分类问题的挑战

- 虽然图像分类大赛正确率已经接近极限，但在实际工程应用中，面临诸多挑战与难题：
  - ✓ 类别不均衡
  - ✓ 数据集小
  - ✓ 巨大的类内差异
  - ✓ 实际应用复杂情况：光照、遮挡、模糊、角度变化、干扰

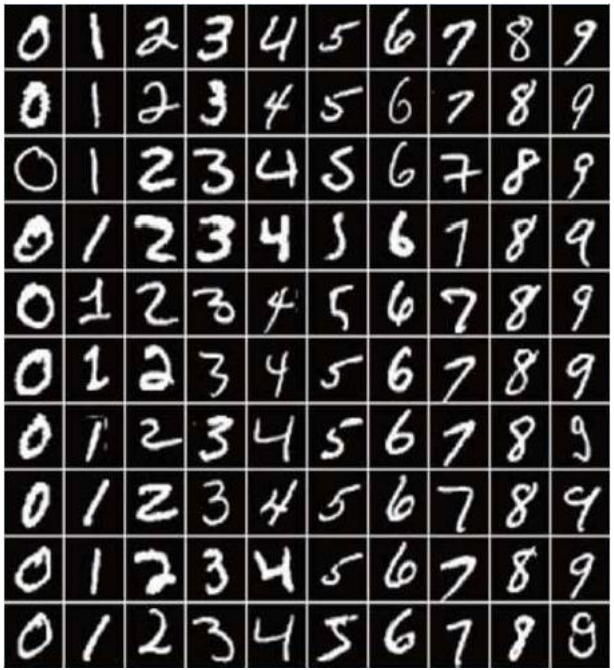
知识讲解

常用数据集介绍

MNIST数据集

知识讲解

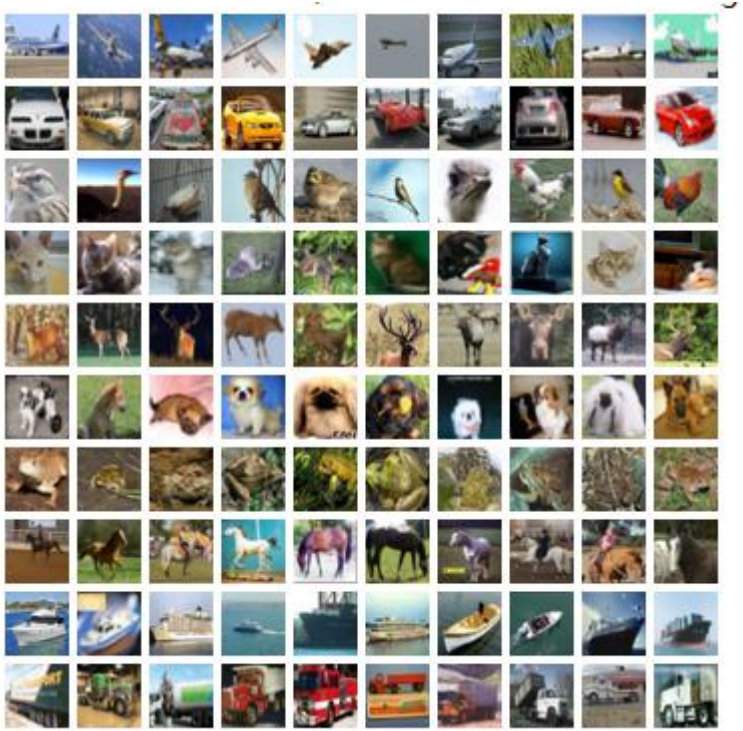
- 手写数字的数据集，来自美国国家标准与技术研究所（National Institute of Standards and Technology，NIST），发布与1998年
- 样本来自250个不同人的手写数字，50%高中学生，50%是人口普查局的工作人员
- 数字从0～9，图片大小是28×28像素，训练数据集包含60000个样本，测试数据集包含10000个样本
- 下载地址：
  http://yann.lecun.com/exdb/mnist/



CIFAR10数据集

知识讲解

- CIFAR10数据集由Alex Krizhevsky、Vinod Nair和Geoffrey Hinton等人收集
- 包含6万张彩色图像，图像大小是32×32，共有10个类，每类有6000张图。其中，5万张图组成训练集合，训练集合中的每一类均等，都有5000张图；剩余1万张图作为测试集合，测试集合中的每一类也均等，各有1000张图
- 10个类别是：airplane、automobile、bird、cat、deer、dog、frog、horse、ship和truck
- 下载地址：http://www.cs.toronto.edu/~kriz/cifar.html



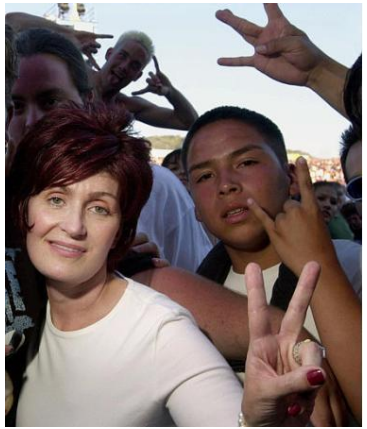
ImageNet数据集

知识讲解

- ImageNet数据集由李飞飞实验室发布于2009年
- 1400多万幅图片，涵盖2万多个类别的标注与超过百万的边界框标注，每一个类别大约有500～1000张图片
- ImageNet竞赛使用的是ImageNet完整数据集的一个子类，包括1000类，其中大部分是动物。在深度学习任务中，我们经常会使用ImageNet预训练的模型
- 下载地址：http://www.image-net.org/

FDDB人脸数据集

- 发布于2010年，是被广泛用于人脸检测方法评测的一个数据集
- 共2845张图像，包含有5171张人脸图像，大部是自然条件下拍摄的名人
- 下载地址：http://vis-www.cs.umass.edu/fddb/index.html#download



知识讲解

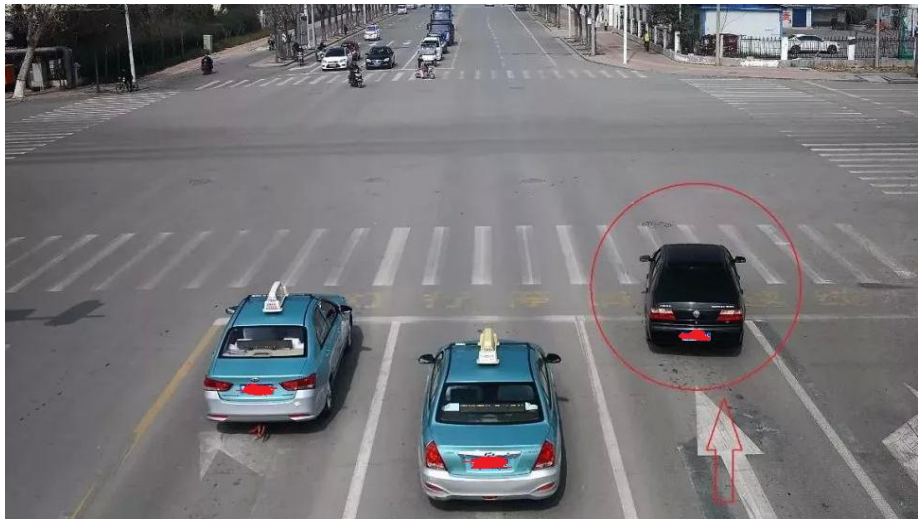
WIDER Face数据集

- 2015年由香港中文大学发布
- 32203张图像，共有393703张人脸图像，比FDDB数据集大10倍，而且在面部的尺寸、姿势、遮挡、表情、妆容和光照上都有很大的变化，自发布后广泛应用于评估性能比传统方法更强大的卷积神经网络
- 下载地址：http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/

图像分类的应用

图像分类的应用

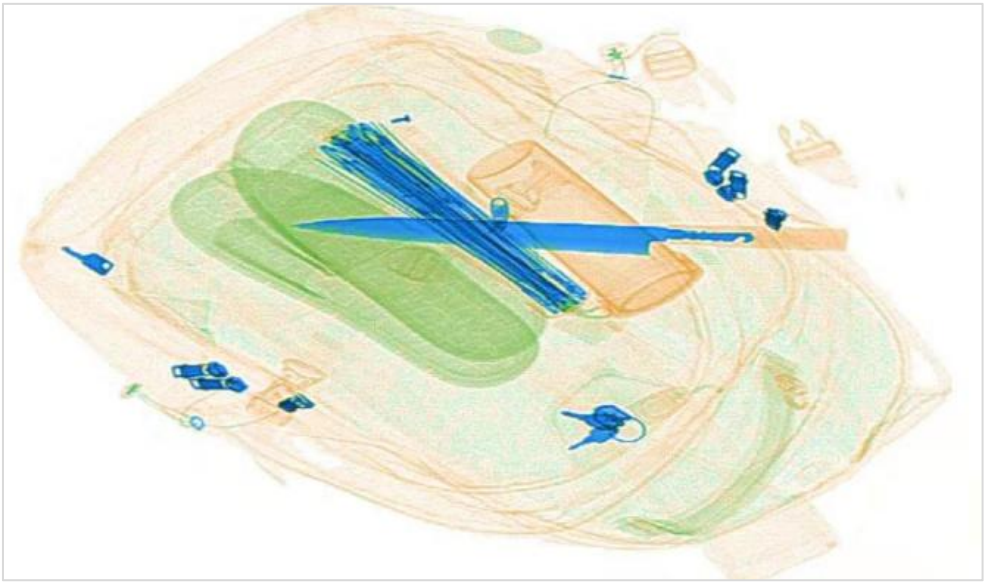
- 交通违章识别



知识讲解

图像分类的应用（续1）

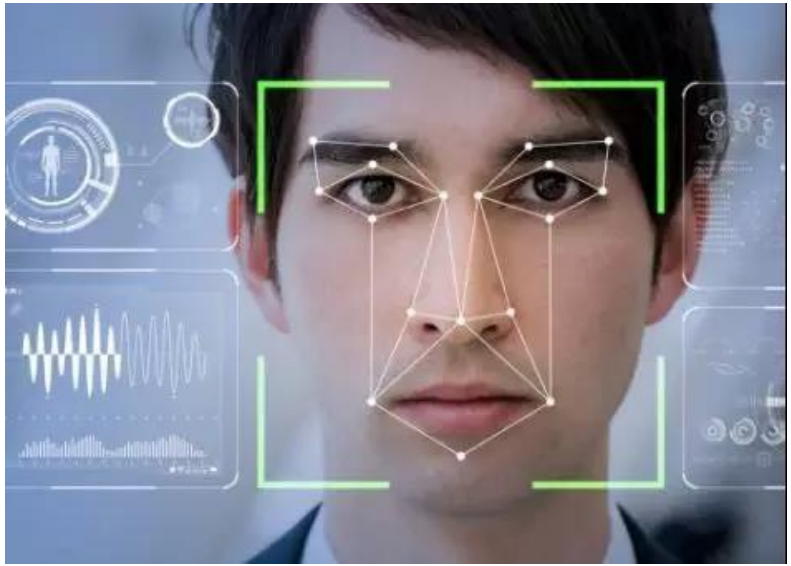
- 安检系统



知识讲解

图像分类的应用（续2）

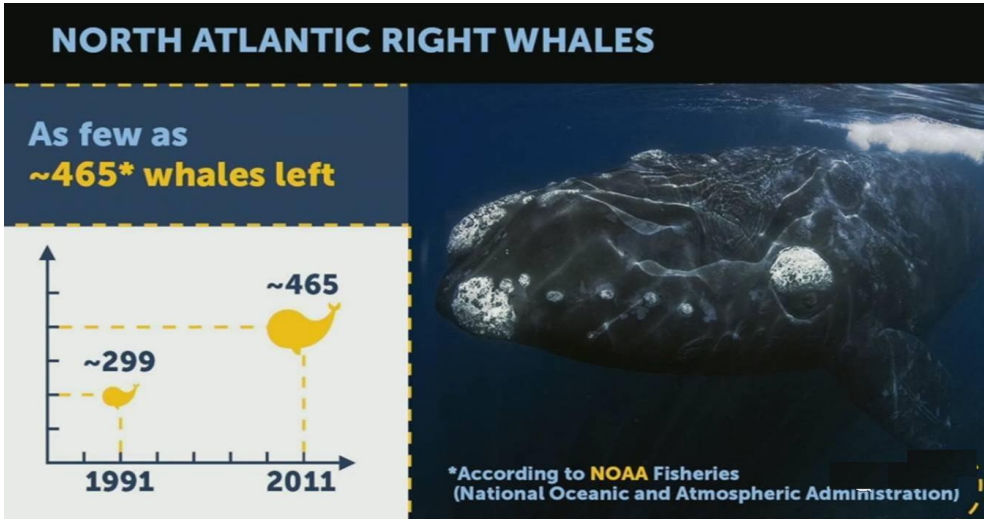
- 人脸识别



知识讲解

图像分类的应用（续3）

- 生物种群数量统计



知识讲解

图像分类的应用（续4）

- 工业质检

知识讲解

图像分类的应用（续5）

- 工地安全监测

知识讲解

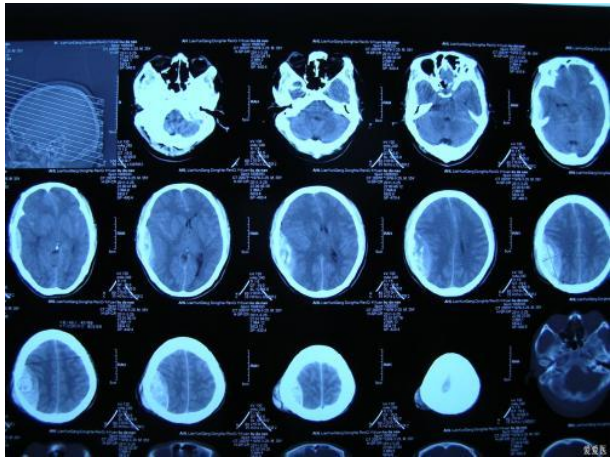
图像分类的应用（续6）

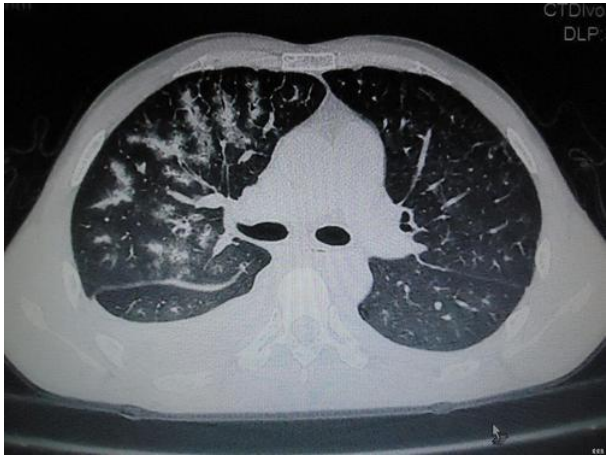
- 病虫害识别

知识讲解

图像分类的应用（续7）

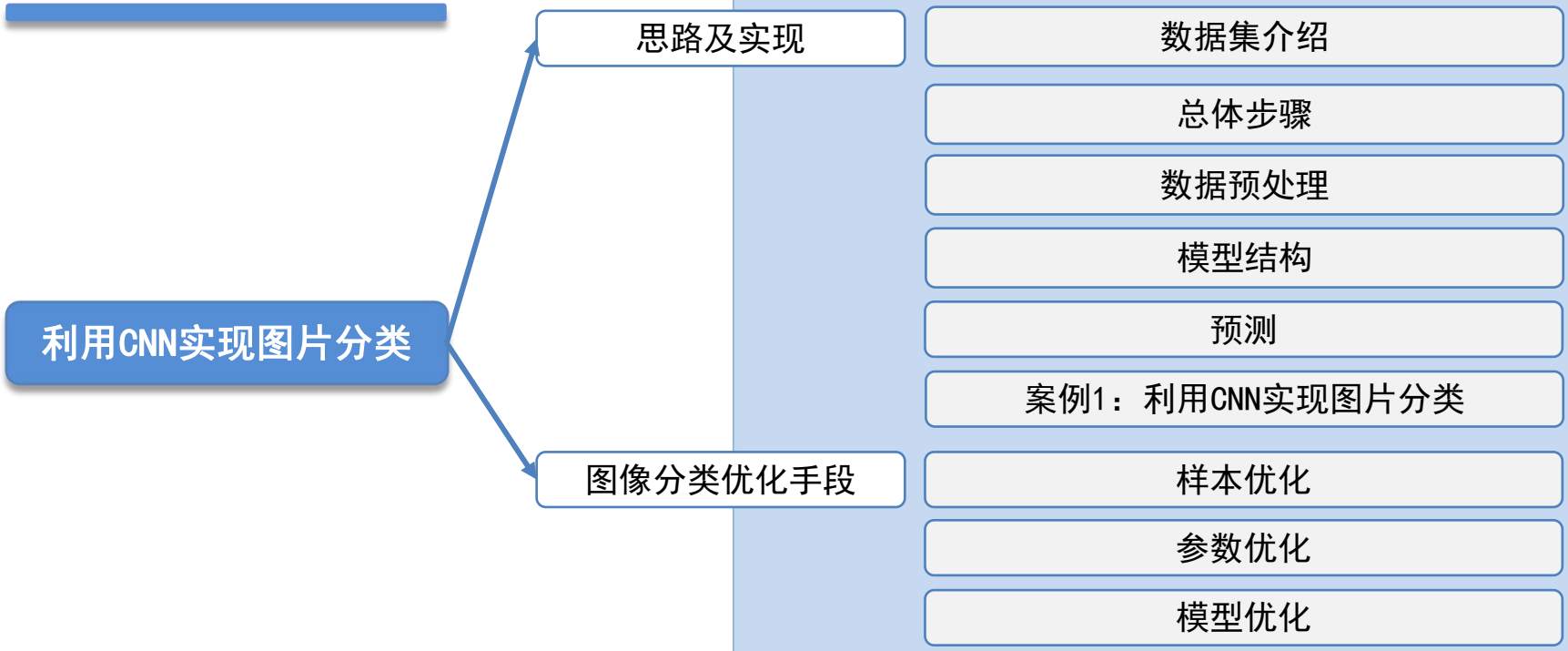
- 医疗诊断





知识讲解

利用CNN实现图片分类



思路及实现

数据集介绍

知识讲解

- 来源：爬虫从百度图片搜索结果爬取
- 内容：包含1036张水果图片，共5个类别（苹果288张、香蕉275张、葡萄216张、橙子276张、梨251张）
- 图像预处理时，将其中10%作为测试数据，90%作为训练数据

总体步骤

- 数据预处理：建立分类文件，建立训练集、测试集
- 训练与模型评估
- 读取测试图片，进行预测

知识讲解

数据预处理

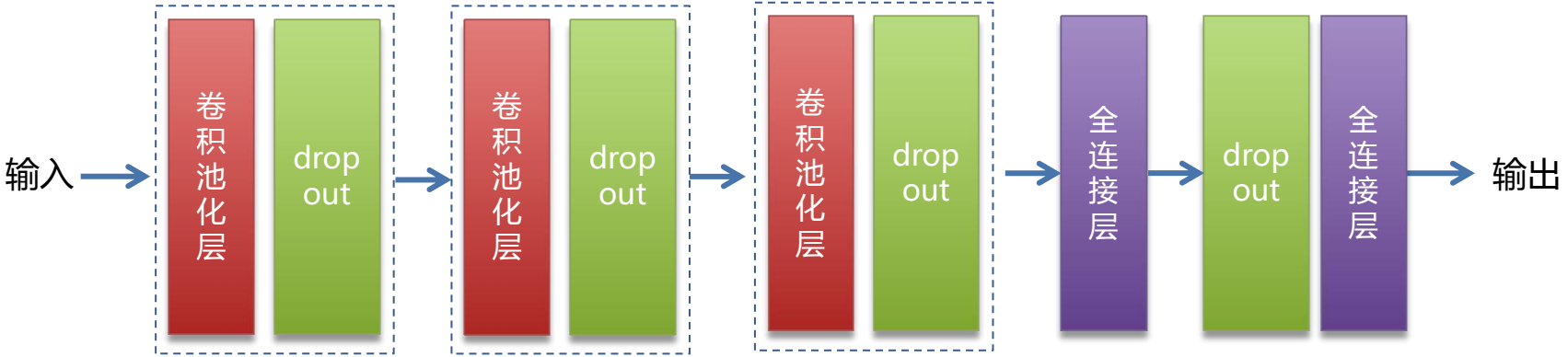
知识讲解

- 图片位于5个目录，遍历每个目录，将其中90%写入训练集文件，10%写入测试集文件，文件中记录了图片的路径，用于数据读取器进行读取
- 生成3个文件：readme.json（汇总文件）、trainer.list（训练集）、test.list（测试集）
- 注意：
  - ✓ 数据集路径是否正确
  - ✓ 生成的汇总文件、训练集文件、测试集文件是否正确

模型结构

知识讲解

- 模型


输入
卷积池化层
drop out
卷积池化层
drop out
卷积池化层
drop out
全连接层
drop out
全连接层
输出

模型结构（续1）

知识讲解

```
def convolution_neural_network(image, type_size):
    # 第一个卷积-池化层
    conv_pool_1 = fluid.nets.simple_img_conv_pool(input=image,  # 输入image
                                                  filter_size=3,  # 滤波器大小
                                                  num_filters=32,  # filter数量，与输出通道相同
                                                  pool_size=2,  # 池化层大小2*2
                                                  pool_stride=2,  # 池化层步长
                                                  act="relu")  # 激活函数
    # Dropout主要作用是减少过拟合，随机让某些权重不更新
    drop = fluid.layers.dropout(x=conv_pool_1, dropout_prob=0.5)

    # 第二个卷积-池化层
    conv_pool_2 = fluid.nets.simple_img_conv_pool(input=drop, filter_size=3, num_filters=64,
                                                  pool_size=2, pool_stride=2, act="relu")
    drop = fluid.layers.dropout(x=conv_pool_2, dropout_prob=0.5)

    # 第三个卷积-池化层
    conv_pool_3 = fluid.nets.simple_img_conv_pool(input=drop, filter_size=3, num_filters=64,
                                                  pool_size=2, pool_stride=2, act="relu")
    drop = fluid.layers.dropout(x=conv_pool_3, dropout_prob=0.5)

    # 全连接层
    fc = fluid.layers.fc(input=drop, size=512, act="relu")
    # dropout层
    drop = fluid.layers.dropout(x=fc, dropout_prob=0.5)
    # 输出层
    predict = fluid.layers.fc(input=drop, size=type_size, act="softmax")
    return predict
```

预测

知识讲解

- 加载模型并预测

```
with fluid.scope_guard(inference_scope):
    [inference_program, feed_target_names, fetch_targets] = \
        fluid.io.load_inference_model(model_save_dir, infer_exe)
    img = Image.open(test_img)
    plt.imshow(img)
    plt.show()

    #开始预测
    results = infer_exe.run(inference_program,
                            feed={feed_target_names[0]:infer_imgs},
                            fetch_list=fetch_targets)
    # name_dict = {"apple":0, "banana":1, "grape":2, "orange":3, "pear":4}
    print(results)

    result = np.argmax(results[0])
    for k, v in name_dict.items():
        if result == v:
            print("预测结果:", k)
```

案例1：利用CNN实现图片分类

- 代码见fruits.py

课堂练习

图像分类优化手段

样本优化

- 增大样本数量
- 数据增强
  - ✓ 形态变化：翻转、平移、随机修剪、尺度变换、旋转
  - ✓ 色彩变化：色彩抖动（错位的位移对图像产生的一种特殊效果）、图像白化（将图像本身归一化成 Gaussian(0,1) 分布）
  - ✓ 加入噪声：噪声扰动

知识讲解

参数优化

知识讲解

- 丢弃学习：按照一定比率丢弃神经元输出
- 权重衰减：通过为模型损失函数添加惩罚项使得训练的模型参数较小
- 批量正则化：在网络的每一层输入之前增加归一化处理，使输入的均值为 0，标准差为 1。目的是将数据限制在统一的分布下
- 变化学习率：学习率由固定调整为变化，例如由固定0.001调整为0.1, 0.001, 0.0005
- 加深网络：加深网络**可能**提高准确率，也可能降低准确率，视具体情况而定

模型优化

- 更换更复杂、精度更高的网络模型。如由简单CNN更换为VGG、GooLeNet、ResNet

知识讲解

今日总结

- 图像分类问题概述
- 常用数据集
- 图像分类的应用
- 利用CNN实现图像分类
  - 实现分类
  - 图像分类优化手段