

3.2.x知识点总结

模型加载
session =
onnxruntime.InferenceSession('模型名')

类别标签列表加载

列表推导式加载
1、类别标签列表 = [name.strip() for name in open('类别标签文件名').readlines()]

2、with open('类别标签文件名','r') as f:
 类别标签列表 = [line.strip() for line in f.readlines()]

自定义类别标签映射字典
标签映射字典 = {'标签1': 0, '标签2': 1, ...}

图像识别

图像加载

直接打开
Image.open('图像文件名').convert('图像类型')

opencv读取
cv2.imread('图像文件名')

图像预处理

图像大小调节

图像转换为numpy数组
numpy.array(带转换图像对象, dtype = 数组的数据类型)

对图像numpy数组增加维度
np.expand_dims(图像numpy数组, axis=0)

图像归一化

Image调整
调整后的图像对象 = 待调整图像对象.resize((宽度,高度))

opencv调整
调整后的图像对象 = cv2.resize(待调整的图像对象, (宽度, 高度))

增加批次维度

增加通道维度

定义归一化的均值数组
np.array([维度1均值, 维度2均值, 维度3均值, ...])

模型运行
[输出结果1, 输出结果2, ...] = session.run([输出节点名1,输出节点名2, ...], {输入节点名1: 输入对象2, 输入节点名2: 输入对象2, ...})

取出输出结果
对于resnet.onnx模型而言, 输出节点只有一个, 故取输出结果用[0]索引取出第一个即可

输出结果 = session.run([输出节点名], {输入节点名: 输入对象})[0]

识别结果整理输出

识别结果概率化
识别结果的概率值多维数组 =
scipy.special.softmax(模型输出结果, axis=-1)

识别结果索引整理

识别结果输出

概率值转换成百分比
概率最大值百分比 = 概率值多维数组[0][最大值索引]*100

排序
待排序对象的索引从小到大的排序结果 = np.argsort(待处理列表)

取最大值
最大值索引 = np.argmax(待处理列表)

索引到类别标签
1、预测结果排序 = 类别标签列表[排序后的索引列表]

2、唯一预测结果 = 类别标签列表[最大值索引]

切片整理顺序
切片之后列表 = 待切片列表[开始索引:结束索引:取值步长]

文件路径创建
os.mkdir('路径名称')