

## 第9讲 视觉应用部署

### 学习目标:

能够简述计算机视觉应用的开发流程

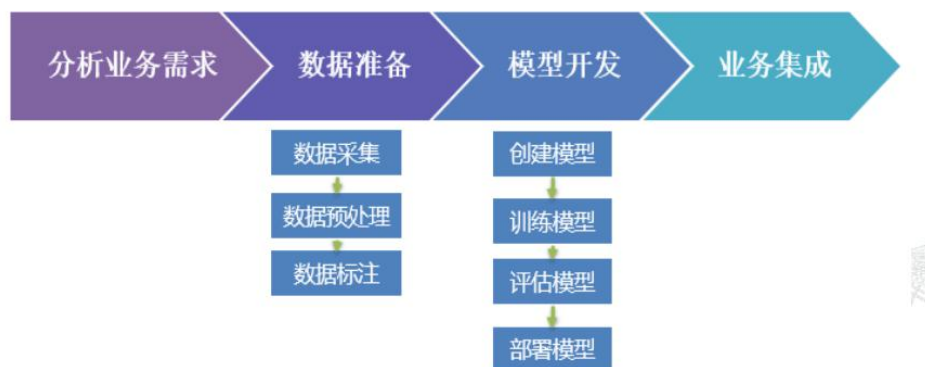
能够使用模型评估的方法，完成简单模型评估

能够使用百度 EasyDL 平台完成基本的图像应用开发

能够运用模型的云端部署与调用的方法，完成模型的在线调用

EasyDL 是百度推出的定制化 AI 训练平台，可利用其进行计算机视觉应用开发。

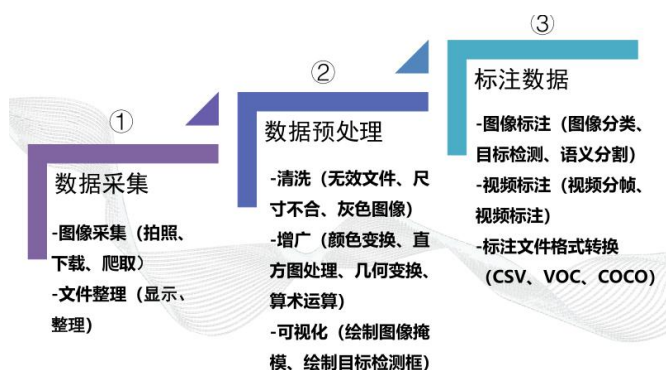
### 9.1 开发流程



#### 1. 分析业务需求

- (1) 监控保安是否在岗—通过图像分类模型进行判断
- (2) 是否有异常人员进入—通过人脸识别模型进行判断
- (3) 各区域垃圾桶是否已满

#### 2. 数据准备



#### 3. 模型开发

- (1) 创建模型：图像分类模型、物体检测模型、图像分割模型、视频分类 moxing
- (2) 训练模型

### (3) 评估模型

#### 混淆矩阵

混淆矩阵				
		预测值 ( Predict )		
		True	False	
实际值 ( Actual )	True	TP	FN	TP(True Positive) : 样本真实类别为正, 预测类别也为正。
	False	FP	TN	FN(False Negative) : 样本真实类别为正, 预测类别为负。
				FP(False Positive) : 样本真实类别为负, 预测类别为正。
				TN(True Negative) : 样本真实类别为负, 预测类别也为负。

准确率: 正确样本数与总样本数之比

$\text{accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$   $(18 + 30) / 50$

精确率: 正确预测的正样本数占有所有预测为正样本的数量的比值

$\text{precision} = \text{TP} / (\text{TP} + \text{FP})$   $18 / (18 + 0)$

召回率: 正确预测的正样本数占真是正样本总数的比例

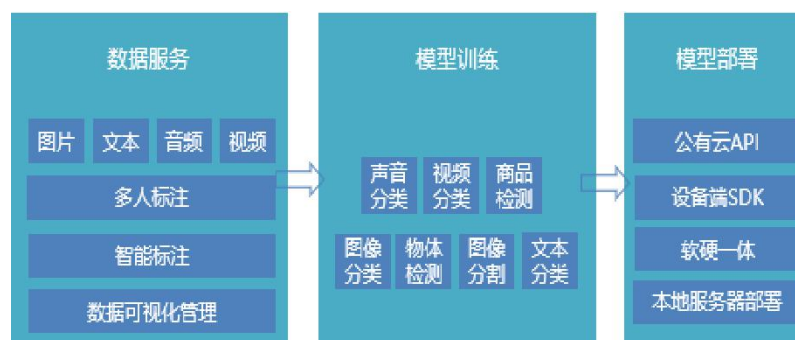
$\text{recall} = \text{TP} / (\text{TP} + \text{FN})$   $18 / 20$

### (4) 部署模型

#### 4.业务集成

## 9.2 EasyDL 的使用

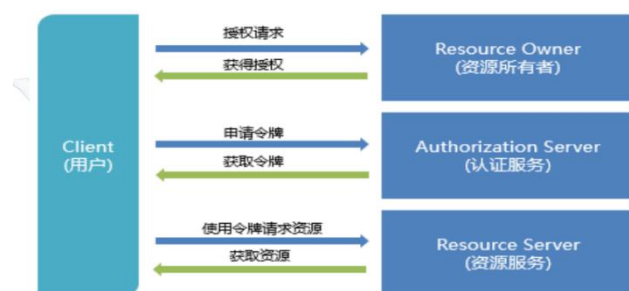
<https://ai.baidu.com/easydl/lite>



### (1) EasyDL 的使用方法



### (2) 调用模型 (API)



令牌 token

百度应用授权服务地址：

<https://aip.baidubce.com/oauth/2.0/token>

grant\_type: 必须参数，固定为 client\_credentials;

client\_id: 必须参数，应用的 API Key;

client\_secret: 必须参数，应用的 Secret Key;

返回参数：

access\_token: 要获取的 Access Token;

expires\_in: Access Token 的有效期(秒为单位，一般为 1 个月);

## 9.3 视觉应用项目实施流程



## 9.4 项目任务

### 9.4.1 数据分类与标注

#### 1.数据分类并打包

首先，创建 5 个空白文件夹，并将不同的图片分别对应存放到不同的文件夹中，再将文件夹进行更名，即完成图像分类与标注。



如本次的数据集为五种标签，即五个按如下标签命名的文件夹：

## 2. 访问植物识别应用



注意：图片类型为 jpg/png/bmp/jpeg，并且大小限制在 4M 内，长宽比在 3:1 以内，其中最长边需要小于 4096px，最短边需要大于 30px。图片应与实际业务场景(光线、角度、采集设备)尽可能一致，且无重复图片。

将以上 5 个文件夹打成 1 个 zip 包，注意压缩包的大小要在 5GB 以内。

## 2. 上传数据集

进入百度 AI 开放平台：<https://ai.baidu.com/>

图像分类模型

总览

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

EasyData数据服务

数据总览

标签组管理

在线标注

智能标注

我的数据总览 > 蔬菜图片数据集/V1/导入

标签个数0

待确认0

大小0M

数据清洗

暂未做过数据清洗任务

导入数据

数据标注状态

☐ 无标注信息

☒ 有标注信息

导入方式

本地导入

上传压缩包

标注格式

☒ 以文件夹命名分类

☐ json (平台通用)

☐ xml (特指voc)

☐ json (特指coco)

上传压缩包

上传压缩包

确认并返回

点击"开始上传"之后会自动跳转到数据集页面，这时选择"确认并返回"，即可进行上传。

### 3.数据标注

## 9.4.2 模型创建与训练

### 1.创建模型

单击模型中心"创建模型"，然后输入"模型名称"，选择"个人"，输入"邮箱地址" "联系方式" "功能描述"，并单击"下一步"。

模型中心

模型列表 > 创建模型

模型类别: 图像分类

模型名称: flowers-demo

模型归属: 公司 个人

邮箱地址:

联系方式:

功能描述: 识花应用使用demo

下一步

### 2.训练模型

点击"训练"，并配置训练模型。

飞桨 EasyDL

产品介绍 操作平台 应用案例 使用文档

社区 飞桨官网 AI主站 winfrey2010

图像分类模型

训练模型

选择模型: 蔬菜识别

训练配置

部署方式: 公有云部署 EasyEdge本地部署 如何选择部署方式?

选择设备: 服务器 通用小型设备 专项适配硬件

选择算法: 高精度 高性能 AI市场已购模型 查看算法性能及适配硬件

高级训练配置

添加数据

添加数据集 + 请选择

数据集	版本	分类数量	操作
蔬菜图片数据集	V1	7	移除

选择训练模型条件，其中部署方式选择"公有云 API"，选择算法选择"高精度"，然后单击"添加训练数据"。

选择算法介绍：

高精度适合数据量比较少的数据集，如训练数据在 1000 张图片内，优势是准确率效果更

高。

高性能适合大一些的数据集，优势是相同训练数据量的情况下，训练耗时短，模型预测速度快，但准确率效果平均要比高精度算法低 3%-5%。

点击"开始训练"，耐心等待训练完成即可。

### 9.4.3 模型评估与发布

#### 1.启动校验服务

单击"校验"，进入校验模型界面。

【图像分类】蔬菜识别 <input checked="" type="checkbox"/> 模型ID: 207709 <span>免训练迭代模式 new</span> <span>训练</span> <span>历史版本</span> <span>删除</span>					
部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	● 训练完成	未发布	top1准确率: 100.00% top5准确率: 100.00% <a href="#">完整评估结果</a>	<a href="#">查看版本配置</a> <a href="#">申请发布</a> <a href="#">校验</a>

检查模型无误后，点击"启动模型校验服务"。

校验模型

选择模型: flowers-demo

部署方式: 公有云部署 (目前仅支持)

选择版本: V1

启动模型校验服务

#### 2.进行校验

选择"点击添加图片"，并选择测试图片进行提交。

选择模型: 蔬菜识别

部署方式: 公有云API

选择版本: V1

当前模型准确率 100.00% [评估报告](#)



识别结果:

胡萝卜

红苋菜

广东菜心

地瓜叶

黑茄子

识别结果 [如何优化效果?](#)

调整阈值 0.03

预测分类	置信度 > 3.00%
胡萝卜	48.12%
红苋菜	23.70%
广东菜心	12.19%
地瓜叶	7.40%
黑茄子	5.93%

### 9.4.4 发布公有云部署并调用服务

#### 1.发布模型配置

输入"服务名称"，"接口地址"，点击"提交申请"即可。（可留意右侧接口参数说明，最

终编写代码时需对应解析其参数)

图像分类模型

总览

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

EasyData数据服务

数据总览

标签组管理

在线标注

智能标注

发布模型

选择版本V1

服务名称 \* 蔬菜识别

接口地址 \* https://aip.baidubce.com/rpc/2.0/ai\_custom/v1/classification/ vegsong

其他要求

若接口无法满足您的需求, 请描述希望解决的问题, 500汉字以内

0/500

☒ 同意云服务数据回流服务条款并开通服务

提交申请

单击服务详情

模型列表

操作文档 常见问题 新手教程 提交工单

创建模型

【图像分类】蔬菜识别 模型ID: 207709

免训练迭代模式 new

训练

历史版本

删除

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	已发布	top1准确率: 100.00% top5准确率: 100.00% 完整评估结果	查看版本配置 服务详情 校验 体验H5

检查接口地址, 单击"立即使用"

模型列表

操作文档 常见问题 新手教程 提交工单

创建模型

【图像分类】蔬菜识别 模型ID: 207709

免训练迭代模式 new

训练

历史版本

删除

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	已发布	top1准确率: 100.00% top5准确率: 100.00% 完整评估结果	查看版本配置 服务详情 校验 体验H5

服务详情

服务名称: 蔬菜识别

模型版本: V1

接口地址: https://aip.baidubce.com/rpc/2.0/ai\_custom/v1/classification/vegsong

服务状态: 已发布

立即使用

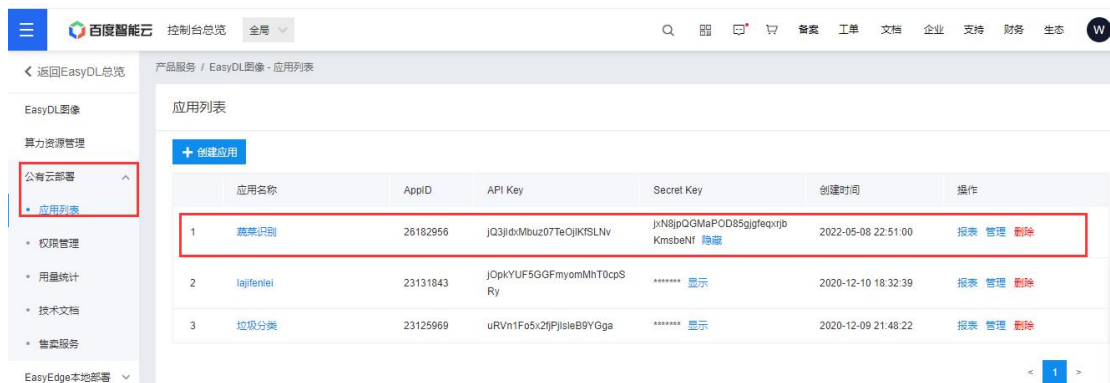
查看API文档

2. 应用创建与配置

返登录百度账号之后将自动跳转到"百度智能云", 找到"公有云服务管理"->"应用列表",

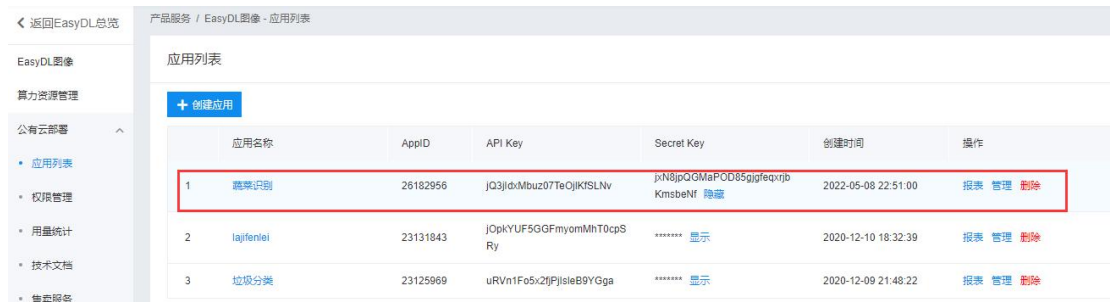


单击"马上创建", 来进行创建应用。



输入"应用名称", 选择"应用类型" "接口选择", 填写"应用描述", 最后点击"立即创建"即可。

出现如图所示即创建成功, 点击"查看应用详情", 即可看到 API Key 和 Secret Key。



### 3.编写应用调用代码

接口地址: [https://aip.baidubce.com/rpc/2.0/ai\\_custom/v1/classification/vegsong](https://aip.baidubce.com/rpc/2.0/ai_custom/v1/classification/vegsong)

首先导入所需要的库 requests、json 库, 通过 API Key 和 Secret Key 获取的 access\_token, 参考代码:

```
"""
EasyDL 图像分类 调用模型公有云 API Python3 实现
"""
```

```
import json
import base64
import requests
```

```
"""
```

使用 requests 库发送请求

使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库, 执行命令

```
pip freeze | grep requests
```

若返回值为空, 则安装该库

```
pip install requests
```

```
"""
```



```

# 目标图片的 本地文件路径，支持 jpg/png/bmp 格式
IMAGE_FILEPATH = r"D:\03 教学\21-22-01,02 计算机视觉应用开发\HU. jpeg"

# 可选的请求参数
# top_num: 返回的分类数量，不声明的话默认为 6 个
PARAMS = {"top_num": 2}

# 服务详情 中的 接口地址
MODEL_API_URL =
"https://aip.baidubce.com/rpc/2.0/ai_custom/v1/classification/vegsong"

# 调用 API 需要 ACCESS_TOKEN。若已有 ACCESS_TOKEN 则于下方填入该字符串
# 否则，留空 ACCESS_TOKEN，于下方填入 该模型部署的 API_KEY 以及
SECRET_KEY，会自动申请并显示新 ACCESS_TOKEN
ACCESS_TOKEN = ""
API_KEY = "jQ3jIdxMbuz07Te0j1KfSLNv"
SECRET_KEY = "jxN8jpQGmaPOD85gjgfeqxrjbKmsbeNf"

print("1. 读取目标图片 ' {}'.format(IMAGE_FILEPATH))
with open(IMAGE_FILEPATH, 'rb') as f:
    base64_data = base64.b64encode(f.read())
    base64_str = base64_data.decode('UTF8')
print("将 BASE64 编码后图片的字符串填入 PARAMS 的 'image' 字段")
PARAMS["image"] = base64_str

if not ACCESS_TOKEN:
    print("2. ACCESS_TOKEN 为空，调用鉴权接口获取 TOKEN")
    auth_url =
"https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id={}&client_secret={}".format(API_KEY, SECRET_KEY)
    auth_resp = requests.get(auth_url)
    auth_resp_json = auth_resp.json()
    ACCESS_TOKEN = auth_resp_json["access_token"]
    print("新 ACCESS_TOKEN: {}".format(ACCESS_TOKEN))
else:
    print("2. 使用已有 ACCESS_TOKEN")

print("3. 向模型接口 'MODEL_API_URL' 发送请求")
request_url = "{}?access_token={}".format(MODEL_API_URL, ACCESS_TOKEN)
response = requests.post(url=request_url, json=PARAMS)
response_json = response.json()

```

```
response_str = json.dumps(response_json, indent=4, ensure_ascii=False)
print("结果:\n{}".format(response_str))
```

输出结果如下:

1. 读取目标图片 'D:\03 教学\21-22-01, 02 计算机视觉应用开发\HU.jpeg'  
将 BASE64 编码后图片的字符串填入 PARAMS 的 'image' 字段
  2. ACCESS\_TOKEN 为空, 调用鉴权接口获取 TOKEN  
新 ACCESS\_TOKEN: 24.3b7e810e24f0beae75e67a1860d9999e.2592000.1654783076.282335-26182956
  3. 向模型接口 'MODEL\_API\_URL' 发送请求
- 结果:

```
{
  "log_id": 1500595951998248177,
  "results": [
    {
      "name": "胡萝卜",
      "score": 0.48117491602897644
    },
    {
      "name": "红苋菜",
      "score": 0.23695452511310577
    },
    {
      "name": "广东菜心",
      "score": 0.12187977880239487
    }
  ]
}
```

接下来解析 API 调用返回结果, 参考代码:

```
""" 解析 API 调用返回结果 """
```

```
def get_result(content):
```

```
    text = ""
    if content:
        # print(content)
        jsonstr = json.loads(content)
        for result in jsonstr["results"]:
            print(result["name"], result["score"])
            # round 函数将 W 保留两位小数
            result["score"] = round(result["score"], 3)
```

```
        text = text + str(result["name"]) + " " + str(result["score"]) + "\n"
    return text
```

```
result_str=get_result(response_str)
```

输出结果如下：

```
胡萝卜 0.48117491602897644
```

```
红苋菜 0.23695452511310577
```

```
广东菜心 0.12187977880239487
```

最后在图片上显示检测结果，参考代码：

```
import cv2
import matplotlib.pyplot as plt
img = cv2.imread(IMAGE_FILEPATH)
%matplotlib inline
plt.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体（解决中文无法显示的问题）
plt.rcParams['axes.unicode_minus'] = False # 解决保存图像时负号“-”显示方块的问题

y0, dy = 50, 30
for i, txt in enumerate(result_str.split('\n')):
    y = y0 + i * dy
    cv2.putText(img, txt, (300, y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
img=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
plt.show()
```

结果如图所示：

