# BKVisionAlgorithms 使用文档

## 简介

BKVisionAlgorithms 是一个为整合各种计算机视觉算法而设计的Python框架。提供统一接口实现 cv 任务

## 算法支持：

图像分类：timm(timm 本身就是分类器整合框架,timm支持 1000+ 分类器模型结构)
  
对象检测：
  
 - yolo: yolov5, yolovX,yolov6,yolov8
  
 - transformer: swin,dino
  
 - 3D:...
  
 - other: EfficientDet , Mask R-CNN
  
  
图像分割：
  
 - Mask R-CNN.FCN,PID,U-Net,DeepLab系列

## 安装

pip install bkvisionalgorithms

## 快速开始

### 导入框架

import bkvisionalgorithms as bkva

### 使用示例1 图像分类

from tqdm import tqdm
  
  
from bkvisionalgorithms.algorithms.base.property import ClassificationProperty, DetectionResult, ImageFolderLoader, \
  
 ImageDetectionDirector, AlgorithmFactory, ImageAdjustBase
  
  
if \_\_name\_\_ == "\_\_main\_\_":
  
 # use AlgorithmFactory to create a detection model
  
 property = ClassificationProperty("demo/classification\_timm\_test1")
  
 if property.debug:
  
 property.save = True
  
  
 classificationModel = AlgorithmFactory().create(property)
  
 print(classificationModel)
  
 imageFolderLoader = ImageFolderLoader(r"E:\clfData\r5",recursion=True,remove=True) # 删除原来的文件
  
 print(imageFolderLoader)
  
 director = ImageDetectionDirector(imageFolderLoader, classificationModel,ImageAdjustBase())
  
 print(director)
  
 for result in tqdm(director):
  
 result:DetectionResult
  
 print(result)
  
 imageFolderLoader.close()
  
 print("end")

### 使用示例2 目标检测

from tqdm import tqdm  
  
from algorithms.base.property import DetectionProperty, DetectionResult, ImageFolderLoader, ImageAdjustSplit, \  
 ImageDetectionDirector, AlgorithmFactory  
  
from ultralytics.utils import USER\_CONFIG\_DIR  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # use AlgorithmFactory to create a detection model  
 property = DetectionProperty("demo/detection\_yolo\_test1")  
  
 # show and save control not use Thread please don't set True in production environment  
 # if character is chinese please install font Arial.Unicode.ttf in /font folder  
  
 if property.debug:  
 USER\_CONFIG\_DIR.mkdir(parents=True, exist\_ok=True)  
 property.save\_dir = USER\_CONFIG\_DIR  
 property.show = True  
 property.show\_all = False  
 property.save = True  
 property.save\_all = False  
  
 detectionModel = AlgorithmFactory().create(property)  
 print(detectionModel)  
 imageFolderLoader = ImageFolderLoader(r"E:\clfData\鼎信\分割\image")  
 print(imageFolderLoader)  
 director = ImageDetectionDirector(imageFolderLoader, detectionModel,ImageAdjustSplit())  
 print(director)  
 for result in tqdm(director):  
 result:DetectionResult  
 imageFolderLoader.close()  
 print("end")

## 核心模块

algorithms.base.property

包含基础属性类（BaseProperty），及专用属性类（DetectionProperty 和 ClassificationProperty）

## 配置文件

使用 Property 实例化 yaml 以创造cv模型  
例： config.yaml

#--encoding:utf-8--  
  
type: detection  
name: yolov5s  
weights: yolov5s.pt  
names: Steel\_RZ.yaml # dataset labels :  
# names: ['\_\_background\_\_', 'tuopi', 'bahen', 'jiaza', 'yiwuyaru', 'huashang', 'bianlie', 'yanghuatiepi',  
# 'gunyin', 'liewen', 'daitougunyin', 'qipi', 'shezhuangqipi', 'zhalan']  
  
img-size: 640 # inference size (pixels)  
conf-thres: 0.25 # confidence threshold  
iou-thres: 0.45 # NMS IoU threshold  
max-det: 1000 # maximum detections per image  
device: 0 # cuda device, i.e. 0 or 0,1,2,3 or cpu  
view-img: false # show results  
save-txt: true # save results to \*.txt  
save-conf: true # save confidences in --save-txt labels  
save-crop: true # save cropped prediction boxes  
nosave: false # do not save images/videos  
batch-size: 32 # inference batch size  
  
debug: true # debug mode

# 依赖

pypattyrn
  
onnxruntime
  
pyyaml
  
torch
  
timm
  
tqdm

# 拓展

# 其他支持

除了 BKVisionAlgorithms ， 下列框架受支持  
 - BKVisionTrain 训练  
 - BKVisionCamera 统一的相机接口，对 面阵，线阵，甚至3D相机 的 适配器框架  
 - BKVisionData 统一的数据支持 对 PLC ， 数据库，TCP/IP ，串口 的 适配器框架  
业务框架：  
 - BKVisionServer 服务端支持  
 - BKVisionBusiness 根据现场的业务逻辑管理

# 许可证

由北京科技大学设计研究院所有。C:\Program Files\Pandoc