

基于人体轮廓检测 的体型分类

计算机视觉第五组汇报总结

第五组：刘意，孟庆鹏，侯飞，秦风

汇报日期



原型系统



结果展示



结果分析

原型系统

01

代码部分

```
1  ## 导入相关模块
2  import os
3  import numpy as np
4  import torchvision
5  import torchvision.transforms as transforms
6  from PIL import Image, ImageDraw, ImageFont
7  import matplotlib.pyplot as plt
8  import math
9
10 COCO_INSTANCE_CATEGORY_NAMES = [
11     '__BACKGROUND__', 'person', 'bicycle', 'car', 'motorcycle',
12     'airplane', 'bus', 'train', 'trunk', 'boat', 'traffic light',
13     'fire hydrant', 'N/A', 'stop sign', 'parking meter', 'bench',
14     'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant',
15     'bear', 'zebra', 'giraffe', 'N/A', 'backpack', 'umbrella', 'N/A',
16     'N/A', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard',
17     'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard',
18     'surfboard', 'tennis racket', 'bottle', 'N/A', 'wine glass',
19     'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple',
20     'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza',
21     'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'N/A',
22     'dining table', 'N/A', 'N/A', 'toilet', 'N/A', 'tv', 'laptop',
23     'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven',
24     'toaster', 'toaster', 'sink', 'refrigerator', 'N/A', 'book', 'clock',
25     'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'
26 ]
27 # 定义能够检测出的关键点名称
28 """
29     elbow 胳膊肘, wrist 手腕, hip 臀部
30 """
```

```
31 COCO_PERSON_KEYPOINT_NAMES = ['nose', 'left_eye', 'right_eye', 'left_ear',  
32                               'right_ear', 'left_shoulder', 'right_shoulder', 'left_elbow',  
33                               'right_elbow', 'left_wrist', 'right_wrist', 'left_hip', 'right_hip',  
34                               'left_knee', 'right_knee', 'left_ankle', 'right_ankle']  
35  
36 # 加载pytorch提供的keypointtrcnn_resnet50_fpn()网络模型，可以对17个人体关键点进行检测。  
37 model = torchvision.models.detection.keypointtrcnn_resnet50_fpn(pretrained=True)  
38 model.eval()  
39  
40 #计算距离函数  
41 def euclideanDistance(p1,p2):  
42     return math.sqrt(((p1[0]-p2[0])**2)+((p1[1]-p2[1])**2))  
43  
44 def Object_Detect(model, image_path, COCO_INSTANCE_CATEGORY_NAMES, threshold=0.5):  
45     # 准备需要检测的图像  
46     image = Image.open(image_path).convert('RGB')  
47     transform_d = transforms.Compose([transforms.ToTensor()]) #将图片从0~255变到0~1  
48     image_t = transform_d(image)    ## 对图像进行变换  
49     pred = model([image_t])        ## 将模型作用到图像上  
50  
51     # 检测出目标的类别和得分  
52     pred_class = [COCO_INSTANCE_CATEGORY_NAMES[ii] for ii in list(pred[0]['labels'].numpy())]  
53     pred_score = list(pred[0]['scores'].detach().numpy())  
54  
55     # 检测出目标的边界框  
56     pred_boxes = [[ii[0], ii[1], ii[2], ii[3]] for ii in list(pred[0]['boxes'].detach().numpy())]  
57
```

```
58     ## 只保留识别的概率大约 threshold 的结果。
59     pred_index = [pred_score.index(x) for x in pred_score if x > 0.5]
60
61     ## 设置图像显示的字体
62     fontsize = np.int16(image.size[1] / 20)
63     font1 = ImageFont.truetype("./FreeMono.ttf", fontsize)
64
65     ## 可视化对象
66     draw = ImageDraw.Draw(image)
67     box = pred_boxes[0]
68     draw.rectangle(box, outline="blue")
69     texts = pred_class[0]+":"+str(np.round(pred_score[0], 2))
70     draw.text((box[0], box[1]), texts, fill="blue", font=font1)
71
72
73     pred_keypoint = pred[0]["keypoints"]
74     # 检测到实例的关键点
75     pred_keypoint = pred_keypoint[pred_index].detach().numpy()
76     # 可视化出关键点的位置
77     fontsize = np.int16(image.size[1] / 50)
78     r = np.int16(image.size[1] / 150) # 圆的半径
79     font1 = ImageFont.truetype("./FreeMono.ttf", fontsize)
80     # 可视化图像
81     image3 = image.copy()
82     draw = ImageDraw.Draw(image3)
```

基本原理

通过人体关键点检测识别出的13个点，我们选取了肩宽、臀宽，肩部与臀部之间的距离也就是人体躯干的长度，三个系数构成比例关系，以此衡量人体体型。将肩宽与体长的比例和臀宽与体长的比例进行聚类分析，将人体比例系数分为三类，由此完成体型的分类。

代码调整

人体关键点检测部分参考的<https://blog.csdn.net/csdnliwenqi/article/details/121694973>通过加载pytorch中的torchvision库中的预训练关键点检测模型keypointtrcnn_resnet50_fpn(), 对人体部位的鼻子, 左眼, 右眼, 左耳, 右耳, 左肩, 右肩, 左胳膊肘, 右胳膊肘, 左手腕, 右手腕, 左臀, 右臀, 左膝, 右膝, 左脚踝, 右脚踝这17个关键点进行检测。由于我们在后续的处理中主要需要用到左右肩和左右臀, 所以计算了肩宽、臀宽以及肩臀距离, 并在实验过程中, 将每个人体数据保存到txt文件中, 以方便承接后续聚类算法。原代码中对于检测人体模型, 会先通过对于图片里的物品进行一次完全地检测和定位, 选取概率阈值大于0.5的物品进行定位。但由于本身是人体关键点检测, 所以关键点的检测和定位只对标签为“人”有效, 所以结果图片中只会出现对标签“人”的框选。我们发现, 在照片不清晰或者人物体型不确切的情况下, 关键点检测和类别框选的准确度会下降, 导致我们在利用50张单人女性照片进行实验的时候, 实验结果出现了更多的情况(五十几个数据)。于是我们对代码进行了完善, 将原本对图片的关键点检测的循环操作改成了单次操作, 同时将优先级选定为最高概率对应的物品(优先定为“人”), 使得最后每张图片只选取对应的女性进行人体关键点检测。此外, 针对后续聚类算法的输入精度问题, 我们只保存人体关键点检测的输出数据的前六位小数。

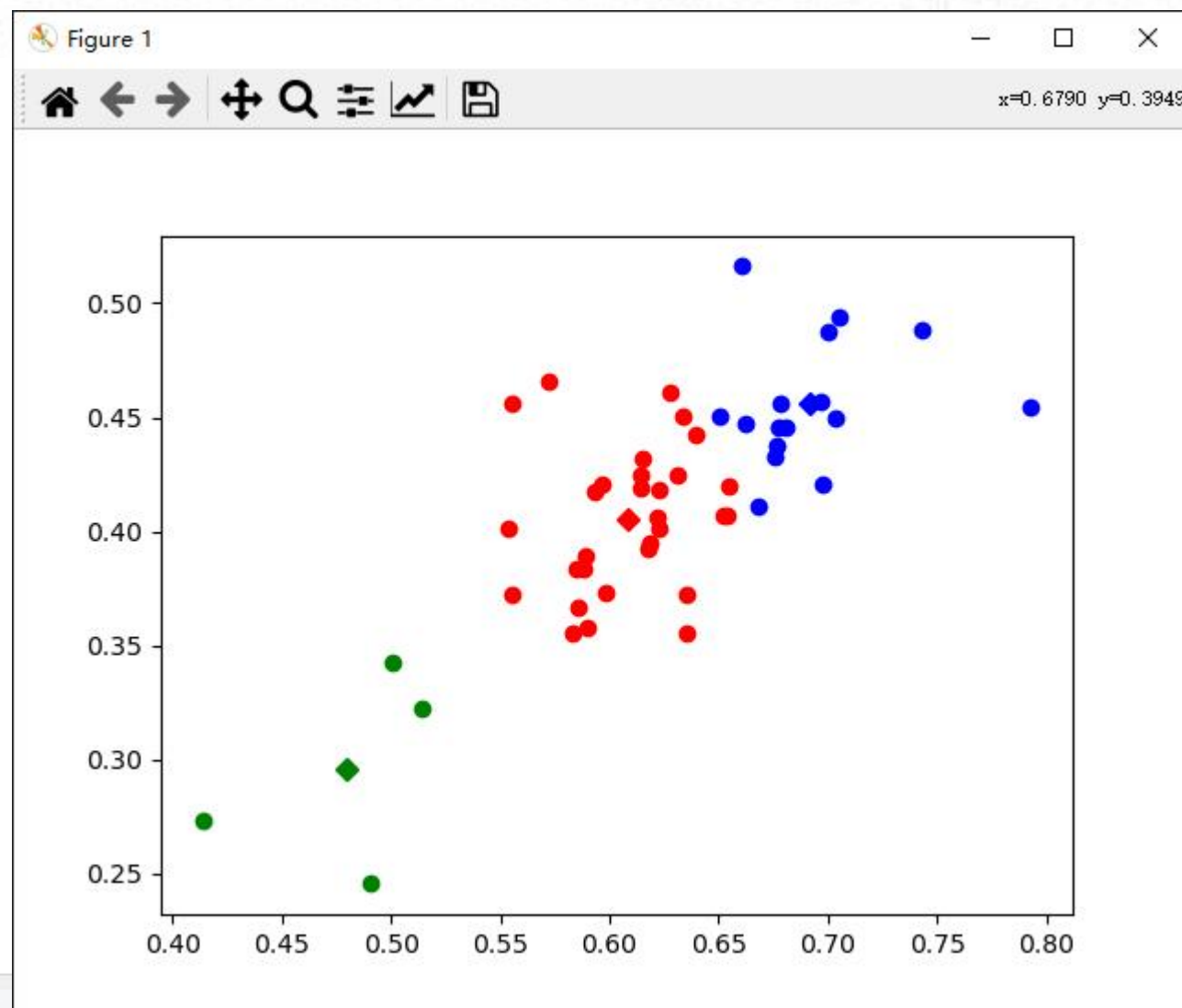
结果展示

02

聚类算法的结果

result - 记事本	result - 记事本
文件(F) 编辑(E) 格式	文件(F) 编辑
2.0	0.0
0.0	2.0
2.0	0.0
2.0	0.0
2.0	0.0
0.0	0.0
0.0	1.0
0.0	2.0
1.0	1.0
0.0	0.0
0.0	2.0
2.0	0.0
2.0	0.0
2.0	0.0
2.0	2.0
0.0	0.0
0.0	1.0
0.0	0.0
0.0	0.0
0.0	2.0

0.0
2.0
2.0
2.0
2.0
0.0
0.0
0.0
0.0
2.0
2.0
2.0
2.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
2.0



数据集的分类结果

1. 分类结果结果为2的数据



2. 分类结果结果为1的数据



3. 分类结果结果为0的数据



结果分析

03

结果分析

- 首先从我们的聚类算法结果质心分布以及数据点分布的情况来看整体为一个线性结构。比例关系为越往右上方，人体躯干越接近于正方形；越往左下方，躯干越瘦长。整个分布关系是比较明确的，证明我们在选取比例关系的时候，没有出现较大的错误。但是这两个比例系数对于结果在分布图上的位置影响，一个是往x轴偏移，一个是往y轴偏移。理想情况就是两个系数对结果影响是大致在一条直线上，因为人体形态分布就是一种线性关系。

结果分析

- 由于我们组电脑的算力有限所以数据集的量比较小导致整体结果看上去没有那么好。分类结果中也存在一些异常点。对于这种情况，我们做了如下分析：首先，数据集的选取，对于体型分类来说，身着宽松衣物的数据对于体型分类有较大的影响。然后，人体的站姿也会影响分类结果，因为我们是直接用图片中坐标的比例关系来进行计算的，而不是利用三维空间中的真实比例关系来计算，这些都造成了结果的一些误差。

A teal-tinted background image featuring a medical stethoscope and a clipboard with a pen. The clipboard has a form with a table of times and a section labeled 'POZIV'.

Thanks

POZIV	
13:30	
13:45	
14:00	

ZAKAZIVANJE