

CSDN新首页上线啦，邀请你来立即体验！(http://blog.csdn.net/)

CSDN

博客 (//blog.csdn.net/edu.csdn.net?ref=toolbar)
下载 (//download.csdn.net?ref=toolbar) GitChat (//gitbook.cn/?ref=csdn)
更多 ▾

1

1

用作者提供的net1->net2生成MTCNN的训练样本
(positive,negative,part,landmark)

原创 2017年08月31日 21:13:44 689

本代码基于作者提供的python版本代码修改，参考：
https://github.com/DuinoDu/mtcnn/blob/master/demo.py （作者提供）
https://github.com/dlunion/mtcnn/blob/master/train/gen_48net_data2.py

1，生成positive,negative,part三种样本，用作者的net1->net2生成bbox，根据预测的bbox和ground truth计算IOU：
positive: IOU >= 0.65;
negative: IOU < 0.3;
part: 0.4 <= IOU < 0.65

代码如下：

```
[python]
1. <span style="font-size:18px;">#!/usr/bin/env python
2. # -*- coding: utf-8 -*-
3.
4. import _init_paths
5. import caffe
6. import cv2
7. import numpy as np
8. #from python_wrapper import *
9. import os
10.
11. def bbreg(boundingBox, reg):
12.     reg = reg.T
13.
14.     # calibrate bouding boxes
15.     if reg.shape[1] == 1:
16.         print "reshape of reg"
17.         pass # reshape of reg
18.     w = boundingbox[:,2] - boundingbox[:,0] + 1
19.     h = boundingbox[:,3] - boundingbox[:,1] + 1
20.
21.     bb0 = boundingbox[:,0] + reg[:,0]*w
22.     bb1 = boundingbox[:,1] + reg[:,1]*h
23.     bb2 = boundingbox[:,2] + reg[:,2]*w
24.     bb3 = boundingbox[:,3] + reg[:,3]*h
25.
26.     boundingbox[:,0:4] = np.array([bb0, bb1, bb2, bb3]).T
```

创意办公室设计

(http://blog.csdn.net/xzzppp)

码云

原创	粉丝	喜欢	未开通
60	155	0	(https://gitee.com/xzzppp)

他的最新文章

更多文章 (http://blog.csdn.net/xzzppp)

用作者提供的net1->net2生成MTCNN的训练样本(positive,negative,part,landmark) (http://blog.csdn.net/XZZPPP/article/details/77753187)

用我们训练的MTCNN中o-net测试训练图片的landmark的mean error (http://blog.csdn.net/XZZPPP/article/details/7725600)

MTCNN中将自己训练的o-net接在作者提供的net1,net2后面python代码解读 (http://blog.csdn.net/XZZPPP/article/details/77652924)

相关推荐

基于Caffe的人脸检测实现 (http://blog.csdn.net/BZ22/article/details/52813031)

人脸检测——DDFD (http://blog.csdn.net/)

立即体验

内容举报

返回顶部

http://blog.csdn.net/XZZPPP/article/details/77753187

1/19

```

27. #print "bb", boundingbox
28. return boundingbox
29.
30.
31. def pad(boxesA, w, h):
32.     boxes = boxesA.copy() # shit, value parameter!!!
33.
34.     tmpw = boxes[:,3] - boxes[:,1] + 1
35.     tmpw = boxes[:,2] - boxes[:,0] + 1
36.     numbox = boxes.shape[0]
37.
38.     dx = np.ones(numbox)
39.     dy = np.ones(numbox)
40.     edx = tmpw
41.     edy = tmpw
42.
43.     x = boxes[:,0:1][:,0]
44.     y = boxes[:,1:2][:,0]
45.     ex = boxes[:,2:3][:,0]
46.     ey = boxes[:,3:4][:,0]
47.
48.
49.     tmp = np.where(ex > w)[0]
50.     if tmp.shape[0] != 0:
51.         edx[tmp] = -ex[tmp] + w-1 + tmpw[tmp]
52.         ex[tmp] = w-1
53.
54.     tmp = np.where(ey > h)[0]
55.     if tmp.shape[0] != 0:
56.         edy[tmp] = -ey[tmp] + h-1 + tmpw[tmp]
57.         ey[tmp] = h-1
58.
59.     tmp = np.where(x < 1)[0]
60.     if tmp.shape[0] != 0:
61.         dx[tmp] = 2 - x[tmp]
62.         x[tmp] = np.ones_like(x[tmp])
63.
64.     tmp = np.where(y < 1)[0]
65.     if tmp.shape[0] != 0:
66.         dy[tmp] = 2 - y[tmp]
67.         y[tmp] = np.ones_like(y[tmp])
68.
69.     # for python index from 0, while matlab from 1
70.     dy = np.maximum(0, dy-1)
71.     dx = np.maximum(0, dx-1)
72.     y = np.maximum(0, y-1)
73.     x = np.maximum(0, x-1)
74.     edy = np.maximum(0, edy-1)
75.     edx = np.maximum(0, edx-1)
76.     ey = np.maximum(0, ey-1)
77.     ex = np.maximum(0, ex-1)
78.
79.     return [dy, edy, dx, edx, y, ey, x, ex, tmpw, tmpw]
80.
81. def IoU(box, boxes):
82.     """Compute IoU between detect box and gt boxes
83.
84.     Parameters:
85.     -----
86.     box: numpy array , shape (5, ): x1, y1, x2, y2, score
87.         input box
88.     boxes: numpy array, shape (n, 4): x1, y1, x2, y2
89.         input ground truth boxes
90.
91.     Returns:
92.     -----
93.     ovr: numpy.array, shape (n, )
94.         IoU
95.     """

```



在线课程



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0iZ0qnfK9ujYzP1mznWR10Aw-



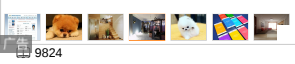
(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0iZ0qnfK9ujYzP1mznWR10Aw-



(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjmsnjD0iZ0qnfK9ujYzP1mznWR10Aw-



创意办公室设计



faster rcnn RPN之anchor(generate_anchor)源码解析 (<http://blog.csdn.net/xzzpp/article/details/52317863>)

9309

win7 64位+caffe+cuda7.5配置 (<http://blog.csdn.net/xzzpp/article/details/51510785>)

7876

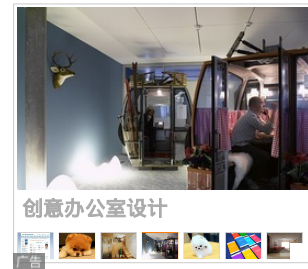
```

95.     """
96.     box_area = (box[2] - box[0] + 1) * (box[3] - box[1] + 1)
97.     area = (boxes[:, 2] - boxes[:, 0] + 1) * (boxes[:, 3] - boxes[:, 1] + 1)
98.     xx1 = np.maximum(box[0], boxes[:, 0])
99.     yy1 = np.maximum(box[1], boxes[:, 1])
100.    xx2 = np.minimum(box[2], boxes[:, 2])
101.    yy2 = np.minimum(box[3], boxes[:, 3])
102.
103.    # compute the width and height of the bounding box
104.    w = np.maximum(0, xx2 - xx1 + 1)
105.    h = np.maximum(0, yy2 - yy1 + 1)
106.
107.    inter = w * h
108.    ovr = inter / (box_area + area - inter)
109.    return ovr
110.
111.
112.    def rerec(bboxA):
113.        # convert bboxA to square
114.        w = bboxA[:, 2] - bboxA[:, 0]
115.        h = bboxA[:, 3] - bboxA[:, 1]
116.        l = np.maximum(w, h).T
117.
118.        bboxA[:, 0] = bboxA[:, 0] + w*0.5 - 1*0.5
119.        bboxA[:, 1] = bboxA[:, 1] + h*0.5 - 1*0.5
120.        bboxA[:, 2:4] = bboxA[:, 0:2] + np.repeat([l], 2, axis = 0).T
121.        return bboxA
122.
123.
124.    def nms(boxes, threshold, type):
125.        """nms
126.        :boxes:[:,0:5]
127.        :threshold: 0.5 like
128.        :type: 'Min' or others
129.        :returns: TODO
130.        """
131.        if boxes.shape[0] == 0:
132.            return np.array([])
133.        x1 = boxes[:, 0]
134.        y1 = boxes[:, 1]
135.        x2 = boxes[:, 2]
136.        y2 = boxes[:, 3]
137.        s = boxes[:, 4]
138.        area = np.multiply(x2-x1+1, y2-y1+1)
139.        I = np.array(s.argsort()) # read s using I
140.
141.        pick = []
142.        while len(I) > 0:
143.            xx1 = np.maximum(x1[I[-1]], x1[I[0:-1]])
144.            yy1 = np.maximum(y1[I[-1]], y1[I[0:-1]])
145.            xx2 = np.minimum(x2[I[-1]], x2[I[0:-1]])
146.            yy2 = np.minimum(y2[I[-1]], y2[I[0:-1]])
147.            w = np.maximum(0.0, xx2 - xx1 + 1)
148.            h = np.maximum(0.0, yy2 - yy1 + 1)
149.            inter = w * h
150.            if type == 'Min':
151.                o = inter / np.minimum(area[I[-1]], area[I[0:-1]])
152.            else:
153.                o = inter / (area[I[-1]] + area[I[0:-1]] - inter)
154.            pick.append(I[-1])
155.            I = I[np.where( o <= threshold)[0]]
156.        return pick
157.
158.
159.    def generateBoundingBox(map, reg, scale, t):
160.        stride = 2
161.        cellsize = 12
162.        map = map.T

```

内容举报

返回顶部



内容举报

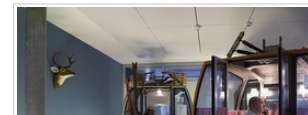
```
163.     dx1 = reg[0, :, :].T
164.     dy1 = reg[1, :, :].T
165.     dx2 = reg[2, :, :].T
166.     dy2 = reg[3, :, :].T
167.     (x, y) = np.where(map >= t)
168.
169.     yy = y
170.     xx = x
171.
172.
173.     score = map[x, y]
174.     reg = np.array([dx1[x, y], dy1[x, y], dx2[x, y], dy2[x, y]])
175.
176.     if reg.shape[0] == 0:
177.         pass
178.     boundingbox = np.array([yy, xx]).T
179.
180.     bb1 = np.fix((stride * (boundingbox) + 1) / scale).T # matlab index from 1, so with "boundi
1"
181.     bb2 = np.fix((stride * (boundingbox) + cellsize - 1 + 1) / scale).T # while python don't ha
182.     score = np.array([score])
183.
184.     boundingbox_out = np.concatenate((bb1, bb2, score, reg), axis=0)
185.
186.     return boundingbox_out.T
187.
188.
189.
190. def drawBoxes(im, boxes):
191.     x1 = boxes[:, 0]
192.     y1 = boxes[:, 1]
193.     x2 = boxes[:, 2]
194.     y2 = boxes[:, 3]
195.     for i in range(x1.shape[0]):
196.         cv2.rectangle(im, (int(x1[i]), int(y1[i])), (int(x2[i]), int(y2[i])), (0, 255, 0), 1)
197.     return im
198.
199. def drawlandmark(im, points):
200.     for i in range(points.shape[0]):
201.         for j in range(5):
202.             cv2.circle(im, (int(points[i][j]), int(points[i][j+5])), 2, (255, 0, 0))
203.     return im
204.
205.
206. from time import time
207. _tstart_stack = []
208. def tic():
209.     _tstart_stack.append(time())
210. def toc(fmt="Elapsed: %s s"):
211.     print fmt % (time() - _tstart_stack.pop())
212.
213.
214. def detect_face(img, minsize, PNet, RNet, threshold, fastresize, factor):
215.
216.     img2 = img.copy()
217.
218.     factor_count = 0
219.     total_boxes = np.zeros((0, 9), np.float)
220.     points = []
221.     h = img.shape[0]
222.
223.     w = img.shape[1]
224.     minl = min(h, w)
225.     img = img.astype(float)
226.     m = 12.0 / minsize
227.     minl = minl * m
228.
229.     ...
```

[TOP](#)
返回顶部



[内容举报](#)

[TOP](#)
返回顶部





1



1



```

229. # create scale pyramid
230. scales = []
231. while minl >= 12:
232.     scales.append(m * pow(factor, factor_count))
233.     minl *= factor
234.     factor_count += 1
235.
236. # first stage
237. for scale in scales:
238.     hs = int(np.ceil(h*scale))
239.     ws = int(np.ceil(w*scale))
240.
241.     if fastresize:
242.         im_data = (img-127.5)*0.0078125 # [0,255] -> [-1,1]
243.         im_data = cv2.resize(im_data, (ws,hs)) # default is bilinear
244.     else:
245.         im_data = cv2.resize(img, (ws,hs)) # default is bilinear
246.         im_data = (im_data-127.5)*0.0078125 # [0,255] -> [-1,1]
247.     #im_data = imResample(img, hs, ws); print "scale:", scale
248.
249.
250.     im_data = np.swapaxes(im_data, 0, 2)
251.     im_data = np.array([im_data], dtype = np.float)
252.     PNet.blobs['data'].reshape(1, 3, ws, hs)
253.     PNet.blobs['data'].data[...] = im_data
254.     out = PNet.forward()
255.
256.     boxes = generateBoundingBox(out['prob1'][0,1,:,:), out['conv4-2']
[0], scale, threshold[0])
257.     if boxes.shape[0] != 0:
258.         pick = nms(boxes, 0.5, 'Union')
259.         if len(pick) > 0 :
260.             boxes = boxes[pick, :]
261.
262.     if boxes.shape[0] != 0:
263.         total_boxes = np.concatenate((total_boxes, boxes), axis=0)
264.
265. #np.save('total_boxes_101.npy', total_boxes)
266.
267. #####
268. # 1 #
269. #####
270. # print "[1]:",total_boxes.shape[0]
271. #print total_boxes
272. #return total_boxes, []
273.
274.
275. numbox = total_boxes.shape[0]
276. if numbox > 0:
277.     # nms
278.     pick = nms(total_boxes, 0.7, 'Union')
279.     total_boxes = total_boxes[pick, :]
280.     # print "[2]:",total_boxes.shape[0]
281.
282.     # revise and convert to square
283.     regh = total_boxes[:,3] - total_boxes[:,1]
284.     regw = total_boxes[:,2] - total_boxes[:,0]
285.     t1 = total_boxes[:,0] + total_boxes[:,5]*regw
286.     t2 = total_boxes[:,1] + total_boxes[:,6]*regh
287.     t3 = total_boxes[:,2] + total_boxes[:,7]*regw
288.     t4 = total_boxes[:,3] + total_boxes[:,8]*regh
289.     t5 = total_boxes[:,4]
290.     total_boxes = np.array([t1,t2,t3,t4,t5]).T
291.     total_boxes = rerec(total_boxes) # convert box to square
292.     # print "[4]:",total_boxes.shape[0]
293.
294.     total_boxes[:,0:4] = np.fix(total_boxes[:,0:4])
295.     # print "[4.5]:",total_boxes.shape[0]
296.     #print total_boxes

```



内容举报

返回顶部



```

290.         #print total_boxes
297.         [dy, edy, dx, edx, y, ey, x, ex, tmpw, tmph] = pad(total_boxes, w, h)
298.
299.
300.     numbox = total_boxes.shape[0]
301.     if numbox > 0:
302.         # second stage
303.
304.         # construct input for RNet
305.         tempimg = np.zeros((numbox, 24, 24, 3)) # (24, 24, 3, numbox)
306.         for k in range(numbox):
307.             tmp = np.zeros((int(tmph[k]) + 1, int(tmpw[k]) + 1, 3))
308.             tmp[int(dy[k]):int(edy[k])+1, int(dx[k]):int(edx[k])+1] = img[int(y[k]):int(ey[k])+
309.             #print "y,ey,x,ex", y[k], ey[k], x[k], ex[k]
310.             #print "tmp", tmp.shape
311.
312.             tempimg[k, :, :, :] = cv2.resize(tmp, (24, 24))
313.
314.         #print tempimg.shape
315.         #print tempimg[0,0,0,:]
316.         tempimg = (tempimg-127.5)*0.0078125 # done in imResample function wrapped by python
317.
318.
319.         # RNet
320.
321.         tempimg = np.swapaxes(tempimg, 1, 3)
322.         #print tempimg[0,:,0,0]
323.
324.         RNet.blobs['data'].reshape(numbox, 3, 24, 24)
325.         RNet.blobs['data'].data[...] = tempimg
326.         out = RNet.forward()
327.
328.         score = out['prob1'][:,1]
329.         #print 'score', score
330.         pass_t = np.where(score>threshold[1])[0]
331.         #print 'pass_t', pass_t
332.
333.         score = np.array([score[pass_t]]).T
334.         total_boxes = np.concatenate( (total_boxes[pass_t, 0:4], score), axis = 1)
335.         # print "[5]:",total_boxes.shape[0]
336.         #print total_boxes
337.
338.         #print "1.5:",total_boxes.shape
339.
340.         mv = out['conv5-2'][pass_t, :].T
341.         #print "mv", mv
342.         if total_boxes.shape[0] > 0:
343.             pick = nms(total_boxes, 0.7, 'Union')
344.             # print 'pick', pick
345.             if len(pick) > 0:
346.                 total_boxes = total_boxes[pick, :]
347.                 # print "[6]:", total_boxes.shape[0]
348.                 total_boxes = bbreg(total_boxes, mv[:, pick])
349.                 # print "[7]:", total_boxes.shape[0]
350.                 total_boxes = rerec(total_boxes)
351.                 # print "[8]:", total_boxes.shape[0]
352.
353.         return total_boxes
354.
355.
356.
357.
358. def main():
359.     img_dir = "/home/xiao/code/mtcnn-caffe/prepare_data/WIDER_train/images/"
360.     imglistfile = "wider_face_train.txt"
361.     with open(imglistfile, 'r') as f:
362.         annotations = f.readlines()
363.     num = len(annotations)
364.     #print "end nice in total" % num

```

内容举报

返回顶部



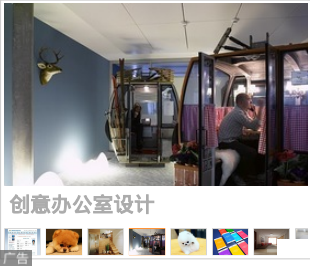
```
364. print "No pics in total %d num"
365.
366. neg_save_dir = "/media/xiao/软件/mtcnn/train/48/negative/"
367. pos_save_dir = "/media/xiao/软件/mtcnn/train/48/positive/"
368. part_save_dir = "/media/xiao/软件/mtcnn/train/48/part/"
369. image_size = 48
370. f1 = open('/media/xiao/软件/mtcnn/train/48/pos_48.txt', 'w')
371. f2 = open('/media/xiao/软件/mtcnn/train/48/neg_48.txt', 'w')
372. f3 = open('/media/xiao/软件/mtcnn/train/48/part_48.txt', 'w')
373.
374. p_idx = 0 # positive
375. n_idx = 0 # negative
376. d_idx = 0 # dont care
377. image_idx = 0
378.
379. minsize = 20
380.
381. caffe_model_path = "./model"
382. threshold = [0.6, 0.7, 0.7]
383. factor = 0.709
384.
385. caffe.set_mode_gpu()
386. PNet = caffe.Net(caffe_model_path+"/det1.prototxt", caffe_model_path+"/det1.caffemodel", ca
387. RNet = caffe.Net(caffe_model_path+"/det2.prototxt", caffe_model_path+"/det2.caffemodel", ca
388.
389. for annotation in annotations:
390.     # imgpath = imgpath.split('\n')[0]
391.     annotation = annotation.strip().split(' ')
392.     bbox = map(float, annotation[1:])
393.     gts = np.array(bbox, dtype=np.float32).reshape(-1, 4)
394.     img_path = img_dir + annotation[0] + '.jpg'
395.
396.     print "#####\n", img_path
397.     print image_idx
398.     image_idx += 1
399.     img = cv2.imread(img_path)
400.     img_matlab = img.copy()
401.     tmp = img_matlab[:, :, 2].copy()
402.     img_matlab[:, :, 2] = img_matlab[:, :, 0]
403.     img_matlab[:, :, 0] = tmp
404.
405.     boundingboxes = detect_face(img_matlab, minsize, PNet, RNet, threshold, False, factor)
406.
407.     #img = drawBoxes(img, boundingboxes)
408.     #cv2.imshow('img', img)
409.     #cv2.waitKey(1000)
410.
411.     # generate positive,negative,part samples
412.     for box in boundingboxes:
413.         x_left, y_top, x_right, y_bottom, _ = box
414.         crop_w = x_right - x_left + 1
415.         crop_h = y_bottom - y_top + 1
416.         # ignore box that is too small or beyond image border
417.         if crop_w < image_size / 2 or crop_h < image_size / 2:
418.             continue
419.         if x_left < 0 or y_top < 0:
420.             continue
421.
422.         # compute intersection over union(IoU) between current box and all gt boxes
423.         Iou = IoU(box, gts)
424.         cropped_im = img[int(y_top):int(y_bottom + 1), int(x_left):int(x_right + 1)]
425.         resized_im = cv2.resize(cropped_im, (image_size, image_size), interpolation=cv2.INTER
426.         try:
427.             # resized_im = cv2.resize(cropped_im, (image_size, image_size), interpolation=cv
428.         except Exception as e:
429.             # print "1 "
430.             # print e
431.
432.     # save negative images and write label 负样本
```

 内容举报 返回顶部 内容举报 返回顶部



1

```
433.         # Iou with all gts must below 0.3
434.         save_file = os.path.join(neg_save_dir, "%s.jpg" % n_idx)
435.         f2.write("%s/negative/%s.jpg" % (image_size, n_idx) + ' 0')
436.         f2.write(" -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1\n")
437.         cv2.imwrite(save_file, resized_im)
438.         n_idx += 1
439.     else:
440.         # find gt_box with the highest iou
441.         idx = np.argmax(Iou)
442.         assigned_gt = gts[idx]
443.         x1, y1, x2, y2 = assigned_gt
444.
445.         # compute bbox reg label, offset_x1, offset_y1相对于左上角; offset_x2, offset_y2
446.         # 相对于右上角
447.         offset_x1 = (x1 - x_left) / float(crop_w)
448.         offset_y1 = (y1 - y_top) / float(crop_h)
449.         # offset_x2 = (x2 - x_left) / float(crop_w)
450.         # offset_y2 = (y2 - y_top) / float(crop_h)
451.         offset_x2 = (x2 - x_right) / float(crop_w)
452.         offset_y2 = (y2 - y_bottom) / float(crop_h)
453.
454.         # save positive and part-face images and write labels, 正样本
455.         if np.max(Iou) >= 0.65:
456.             save_file = os.path.join(pos_save_dir, "%s.jpg" % p_idx)
457.             f1.write("%s/positive/%s.jpg" % (image_size, p_idx) + ' 1 %.6f %.6f %.6f %.6f')
458.             f1.write(" -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1\n")
459.             cv2.imwrite(save_file, resized_im)
460.             p_idx += 1
461.
462.         # part 样本
463.         elif np.max(Iou) >= 0.4:
464.             save_file = os.path.join(part_save_dir, "%s.jpg" % d_idx)
465.             f3.write("%s/part/%s.jpg" % (image_size, d_idx) + ' -1 %.6f %.6f %.6f %.6f')
466.             f3.write(" -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1\n")
467.             cv2.imwrite(save_file, resized_im)
468.             d_idx += 1
469.
470.     f.close()
471.     f1.close()
472.     f2.close()
473.     f3.close()
474.
475. if __name__ == "__main__":
476.     main()
477.
478.
479. </span>
```



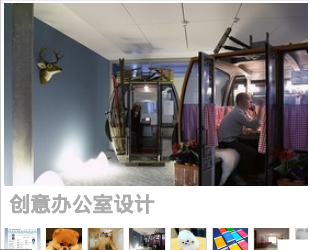
创意办公室设计

内容举报

返回顶部

2, 生成landmark样本, 用作者的net1->net2生成bbox, 根据5个landmark是否都在bbox中作为判别条件:
代码如下:

```
[python]
1. #!/usr/bin/env python
2. # -*- coding: utf-8 -*-
3.
4. import _init_paths
5. import caffe
6. import cv2
```



创意办公室设计


```

7. import numpy as np
8. #from python_wrapper import *
9. import os
10.
11. def bbreg(boundingBox, reg):
12.     reg = reg.T
13.
14.     # calibrate bounding boxes
15.     if reg.shape[1] == 1:
16.         print "reshape of reg"
17.         pass # reshape of reg
18.     w = boundingbox[:,2] - boundingbox[:,0] + 1
19.     h = boundingbox[:,3] - boundingbox[:,1] + 1
20.
21.     bb0 = boundingbox[:,0] + reg[:,0]*w
22.     bb1 = boundingbox[:,1] + reg[:,1]*h
23.     bb2 = boundingbox[:,2] + reg[:,2]*w
24.     bb3 = boundingbox[:,3] + reg[:,3]*h
25.
26.     boundingbox[:,0:4] = np.array([bb0, bb1, bb2, bb3]).T
27.     #print "bb", boundingbox
28.     return boundingbox
29.
30.
31. def pad(boxesA, w, h):
32.     boxes = boxesA.copy() # shit, value parameter!!!
33.
34.     tmpw = boxes[:,3] - boxes[:,1] + 1
35.     tmpw = boxes[:,2] - boxes[:,0] + 1
36.     numbox = boxes.shape[0]
37.
38.     dx = np.ones(numbox)
39.     dy = np.ones(numbox)
40.     edx = tmpw
41.     edy = tmpw
42.
43.     x = boxes[:,0:1][:,0]
44.     y = boxes[:,1:2][:,0]
45.
46.     ex = boxes[:,2:3][:,0]
47.     ey = boxes[:,3:4][:,0]
48.
49.     tmp = np.where(ex > w)[0]
50.     if tmp.shape[0] != 0:
51.         edx[tmp] = -ex[tmp] + w-1 + tmpw[tmp]
52.         ex[tmp] = w-1
53.
54.     tmp = np.where(ey > h)[0]
55.     if tmp.shape[0] != 0:
56.         edy[tmp] = -ey[tmp] + h-1 + tmpw[tmp]
57.         ey[tmp] = h-1
58.
59.     tmp = np.where(x < 1)[0]
60.     if tmp.shape[0] != 0:
61.         dx[tmp] = 2 - x[tmp]
62.         x[tmp] = np.ones_like(x[tmp])
63.
64.     tmp = np.where(y < 1)[0]
65.     if tmp.shape[0] != 0:
66.         dy[tmp] = 2 - y[tmp]
67.         y[tmp] = np.ones_like(y[tmp])
68.
69.     # for python index from 0, while matlab from 1
70.     dy = np.maximum(0, dy-1)
71.     dx = np.maximum(0, dx-1)
72.     y = np.maximum(0, y-1)
73.     x = np.maximum(0, x-1)
74.     edx = np.maximum(0, edx-1)

```



内容举报

返回顶部



```
75.     edx = np.maximum(0, edx-1)
76.     ey = np.maximum(0, ey-1)
77.     ex = np.maximum(0, ex-1)
78.
79.     return [dy, edy, dx, edx, y, ey, x, ex, tmpw, tmph]
80.
81. def IoU(box, boxes):
82.     """Compute IoU between detect box and gt boxes
83.
84.     Parameters:
85.     -----
86.     box: numpy array , shape (5, ): x1, y1, x2, y2, score
87.         input box
88.     boxes: numpy array, shape (n, 4): x1, y1, x2, y2
89.         input ground truth boxes
90.
91.     Returns:
92.     -----
93.     ovr: numpy.array, shape (n, )
94.         IoU
95.     """
96.     box_area = (box[2] - box[0] + 1) * (box[3] - box[1] + 1)
97.     area = (boxes[:, 2] - boxes[:, 0] + 1) * (boxes[:, 3] - boxes[:, 1] + 1)
98.     xx1 = np.maximum(box[0], boxes[:, 0])
99.     yy1 = np.maximum(box[1], boxes[:, 1])
100.    xx2 = np.minimum(box[2], boxes[:, 2])
101.    yy2 = np.minimum(box[3], boxes[:, 3])
102.
103.    # compute the width and height of the bounding box
104.    w = np.maximum(0, xx2 - xx1 + 1)
105.    h = np.maximum(0, yy2 - yy1 + 1)
106.
107.    inter = w * h
108.    ovr = inter / (box_area + area - inter)
109.    return ovr
110.
111.
112. def rerec(bboxA):
113.     # convert bboxA to square
114.     w = bboxA[:,2] - bboxA[:,0]
115.     h = bboxA[:,3] - bboxA[:,1]
116.     l = np.maximum(w,h).T
117.
118.     bboxA[:,0] = bboxA[:,0] + w*0.5 - l*0.5
119.     bboxA[:,1] = bboxA[:,1] + h*0.5 - l*0.5
120.     bboxA[:,2:4] = bboxA[:,0:2] + np.repeat([l], 2, axis = 0).T
121.     return bboxA
122.
123.
124. def nms(boxes, threshold, type):
125.     """nms
126.     :boxes:[:,0:5]
127.     :threshold: 0.5 like
128.     :type: 'Min' or others
129.     :returns: TODO
130.     """
131.     if boxes.shape[0] == 0:
132.         return np.array([])
133.     x1 = boxes[:,0]
134.     y1 = boxes[:,1]
135.     x2 = boxes[:,2]
136.     y2 = boxes[:,3]
137.     s = boxes[:,4]
138.     area = np.multiply(x2-x1+1, y2-y1+1)
139.     I = np.array(s.argsort()) # read s using I
140.
141.     pick = []
142.     while len(I) > 0:
```

 内容举报 返回顶部 内容举报

```

143.         xx1 = np.maximum(x1[I[-1]], x1[I[0:-1]])
144.         yy1 = np.maximum(y1[I[-1]], y1[I[0:-1]])
145.         xx2 = np.minimum(x2[I[-1]], x2[I[0:-1]])
146.         yy2 = np.minimum(y2[I[-1]], y2[I[0:-1]])
147.         w = np.maximum(0.0, xx2 - xx1 + 1)
148.         h = np.maximum(0.0, yy2 - yy1 + 1)
149.         inter = w * h
150.         if type == 'Min':
151.             o = inter / np.minimum(area[I[-1]], area[I[0:-1]])
152.         else:
153.             o = inter / (area[I[-1]] + area[I[0:-1]] - inter)
154.         pick.append(I[-1])
155.         I = I[np.where( o <= threshold)[0]]
156.     return pick
157.
158.
159. def generateBoundingBox(map, reg, scale, t):
160.     stride = 2
161.     cellsize = 12
162.     map = map.T
163.     dx1 = reg[0, :, :].T
164.     dy1 = reg[1, :, :].T
165.     dx2 = reg[2, :, :].T
166.     dy2 = reg[3, :, :].T
167.     (x, y) = np.where(map >= t)
168.
169.     yy = y
170.     xx = x
171.
172.
173.     score = map[x,y]
174.     reg = np.array([dx1[x,y], dy1[x,y], dx2[x,y], dy2[x,y]])
175.
176.     if reg.shape[0] == 0:
177.         pass
178.     boundingbox = np.array([yy, xx]).T
179.
180.     bb1 = np.fix((stride * (boundingbox) + 1) / scale).T # matlab index from 1, so with "boundi
1"
181.     bb2 = np.fix((stride * (boundingbox) + cellsize - 1 + 1) / scale).T # while python don't ha
182.     score = np.array([score])
183.
184.     boundingbox_out = np.concatenate((bb1, bb2, score, reg), axis=0)
185.
186.     return boundingbox_out.T
187.
188.
189.
190. def drawBoxes(im, boxes):
191.     x1 = boxes[:,0]
192.     y1 = boxes[:,1]
193.     x2 = boxes[:,2]
194.     y2 = boxes[:,3]
195.     for i in range(x1.shape[0]):
196.         cv2.rectangle(im, (int(x1[i]), int(y1[i])), (int(x2[i]), int(y2[i])), (0,255,0), 1)
197.     return im
198.
199. def drawBoxes_align(im, boxe):
200.     x1 = boxe[0]
201.     y1 = boxe[1]
202.     x2 = boxe[2]
203.     y2 = boxe[3]
204.     cv2.rectangle(im, (int(x1), int(y1)), (int(x2), int(y2)), (0,255,0), 1)
205.     return im
206.
207. def drawlandmark(im, points):
208.     for i in range(points.shape[0]):
209.         for j in range(5):

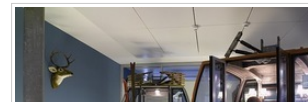
```

返回顶部



内容举报

返回顶部





1



1



```

209.         for j in range(5):
210.             cv2.circle(im, (int(points[i][j]), int(points[i][j+5])), 2, (255,0,0))
211.         return im
212.
213.     def drawlandmark_align(im, point):
214.         for j in range(5):
215.             cv2.circle(im, (int(point[j*2]), int(point[j*2+1])), 2, (255,0,0))
216.         return im
217.
218.
219. from time import time
220. _tstart_stack = []
221. def tic():
222.     _tstart_stack.append(time())
223. def toc(fmt="Elapsed: %s s"):
224.     print fmt % (time()-_tstart_stack.pop())
225.
226.
227. def detect_face(img, minsize, PNet, RNet, threshold, fastresize, factor):
228.
229.     img2 = img.copy()
230.
231.     factor_count = 0
232.     total_boxes = np.zeros((0,9), np.float)
233.     points = []
234.     h = img.shape[0]
235.     w = img.shape[1]
236.     minl = min(h, w)
237.     img = img.astype(float)
238.     m = 12.0/minsize
239.     minl = minl*m
240.
241.
242.     # create scale pyramid
243.     scales = []
244.     while minl >= 12:
245.         scales.append(m * pow(factor, factor_count))
246.         minl *= factor
247.         factor_count += 1
248.
249.     # first stage
250.     for scale in scales:
251.         hs = int(np.ceil(h*scale))
252.         ws = int(np.ceil(w*scale))
253.
254.         if fastresize:
255.             im_data = (img-127.5)*0.0078125 # [0,255] -> [-1,1]
256.             im_data = cv2.resize(im_data, (ws,hs)) # default is bilinear
257.         else:
258.             im_data = cv2.resize(img, (ws,hs)) # default is bilinear
259.             im_data = (im_data-127.5)*0.0078125 # [0,255] -> [-1,1]
260.             #im_data = imResample(img, hs, ws); print "scale:", scale
261.
262.
263.             im_data = np.swapaxes(im_data, 0, 2)
264.             im_data = np.array([im_data], dtype = np.float)
265.             PNet.blobs['data'].reshape(1, 3, ws, hs)
266.             PNet.blobs['data'].data[...] = im_data
267.             out = PNet.forward()
268.
269.             boxes = generateBoundingBox(out['prob1'][0,1,:,:), out['conv4-2']
[0], scale, threshold[0])
270.             if boxes.shape[0] != 0:
271.                 pick = nms(boxes, 0.5, 'Union')
272.                 if len(pick) > 0 :
273.                     boxes = boxes[pick, :]
274.
275.             if boxes.shape[0] != 0:

```



内容举报

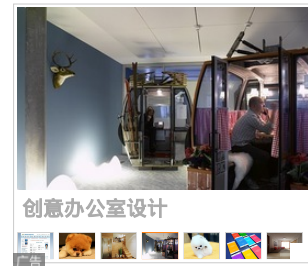
返回顶部



```
276.         total_boxes = np.concatenate((total_boxes, boxes), axis=0)
277.
278.     #np.save('total_boxes_101.npy', total_boxes)
279.
280.     #####
281.     # 1 #
282.     #####
283.     # print "[1]:",total_boxes.shape[0]
284.     #print total_boxes
285.     #return total_boxes, []
286.
287.
288.     numbox = total_boxes.shape[0]
289.     if numbox > 0:
290.         # nms
291.         pick = nms(total_boxes, 0.7, 'Union')
292.         total_boxes = total_boxes[pick, :]
293.         # print "[2]:",total_boxes.shape[0]
294.
295.         # revise and convert to square
296.         regh = total_boxes[:,3] - total_boxes[:,1]
297.         regw = total_boxes[:,2] - total_boxes[:,0]
298.         t1 = total_boxes[:,0] + total_boxes[:,5]*regw
299.         t2 = total_boxes[:,1] + total_boxes[:,6]*regh
300.         t3 = total_boxes[:,2] + total_boxes[:,7]*regw
301.         t4 = total_boxes[:,3] + total_boxes[:,8]*regh
302.         t5 = total_boxes[:,4]
303.         total_boxes = np.array([t1,t2,t3,t4,t5]).T
304.         total_boxes = rerec(total_boxes) # convert box to square
305.         # print "[4]:",total_boxes.shape[0]
306.
307.         total_boxes[:,0:4] = np.fix(total_boxes[:,0:4])
308.         # print "[4.5]:",total_boxes.shape[0]
309.         #print total_boxes
310.         [dy, edy, dx, edx, y, ey, x, ex, tmpw, tmph] = pad(total_boxes, w, h)
311.
312.
313.     numbox = total_boxes.shape[0]
314.     if numbox > 0:
315.         # second stage
316.
317.         # construct input for RNet
318.         tempimg = np.zeros((numbox, 24, 24, 3)) # (24, 24, 3, numbox)
319.         for k in range(numbox):
320.             tmp = np.zeros((int(tmph[k]) +1, int(tmpw[k]) + 1,3))
321.             tmp[int(dy[k]):int(edy[k])+1,int(dx[k]):int(edx[k])+1] = img[int(y[k]):int(ey[k])+
322.             #print "y,ey,x,ex", y[k], ey[k], x[k], ex[k]
323.             #print "tmp", tmp.shape
324.
325.             tempimg[k,:,:,:] = cv2.resize(tmp, (24, 24))
326.
327.         #print tempimg.shape
328.         #print tempimg[0,0,0,:]
329.         tempimg = (tempimg-127.5)*0.0078125 # done in imResample function wrapped by python
330.
331.
332.         # RNet
333.
334.         tempimg = np.swapaxes(tempimg, 1, 3)
335.         #print tempimg[0,:,0,0]
336.
337.         RNet.blobs['data'].reshape(numbox, 3, 24, 24)
338.         RNet.blobs['data'].data[...] = tempimg
339.         out = RNet.forward()
340.
341.         score = out['prob1'][:,1]
342.         #print 'score', score
343.         pass_t = np.where(score>threshold[1])[0]
```

内容举报

返回顶部



```

344.         #print 'pass_t', pass_t
345.
346.         score = np.array([score[pass_t]]).T
347.         total_boxes = np.concatenate( (total_boxes[pass_t, 0:4], score), axis = 1)
348.         # print "[5]:",total_boxes.shape[0]
349.         #print total_boxes
350.
351.         #print "1.5:",total_boxes.shape
352.
353.         mv = out['conv5-2'][pass_t, :].T
354.         #print "mv", mv
355.         if total_boxes.shape[0] > 0:
356.             pick = nms(total_boxes, 0.7, 'Union')
357.             # print 'pick', pick
358.             if len(pick) > 0:
359.                 total_boxes = total_boxes[pick, :]
360.                 # print "[6]:", total_boxes.shape[0]
361.                 total_boxes = bbreg(total_boxes, mv[:, pick])
362.                 # print "[7]:", total_boxes.shape[0]
363.                 total_boxes = rerec(total_boxes)
364.                 # print "[8]:", total_boxes.shape[0]
365.
366.         return total_boxes
367.
368.
369.
370.
371. def main():
372.     img_dir = "/media/xiao/学习/face_alignment/data/CelebA/Img/img_celeba.7z/img_celeba/"
373.     anno_file = "celebA_bbox_landmark.txt"
374.     with open(anno_file, 'r') as f:
375.         annotations = f.readlines()
376.     num = len(annotations)
377.     print "%d pics in total" % num
378.
379.     # 图片大小为48*48
380.     image_size = 48
381.     # landmark_save_dir = "48/landmark/"
382.     landmark_save_dir = "/media/xiao/软件/mtcnn/train/48/landmark/"
383.     # save_dir = "." + str(image_size)
384.     f1 = open('/media/xiao/软件/mtcnn/train/48/landmark_48.txt', 'w')
385.
386.     l_idx = 0 # landmark
387.     image_idx = 0
388.
389.     minsize = 40
390.     caffe_model_path = "./model"
391.     threshold = [0.6, 0.7, 0.7]
392.     factor = 0.709
393.
394.     caffe.set_mode_gpu()
395.     PNet = caffe.Net(caffe_model_path+"/det1.prototxt", caffe_model_path+"/det1.caffemodel", ca
396.     RNet = caffe.Net(caffe_model_path+"/det2.prototxt", caffe_model_path+"/det2.caffemodel", ca
397.
398.
399.     for annotation in annotations:
400.         # imgpath = imgpath.split('\n')[0]
401.         annotation = annotation.strip().split(' ')
402.
403.         im_path = annotation[0]
404.         # bbox = map(float, annotation[1:-10])
405.         pts = map(float, annotation[-10:])
406.         # boxes = np.array(bbox, dtype=np.float32).reshape(-1, 4)
407.         im_path = img_dir + im_path
408.         backupPts = pts[:]
409.
410.         #print "#####\n", im_path
411.         print image_idx

```



内容举报

返回顶部

内容举报

返回顶部

```
412. image_idx += 1
413. img = cv2.imread(im_path)
414. img_matlab = img.copy()
415. tmp = img_matlab[:, :, 2].copy()
416. img_matlab[:, :, 2] = img_matlab[:, :, 0]
417. img_matlab[:, :, 0] = tmp
418.
419. # 用作者提供的net1->net2生成人脸框
420. boundingboxes = detect_face(img_matlab, minsize, PNet, RNet, threshold, False, factor)
421.
422. #img = drawBoxes(img, boundingboxes)
423. #cv2.imshow('img', img)
424. #cv2.waitKey(1000)
425.
426. # generate landmark samples
427. for box in boundingboxes:
428.     x_left, y_top, x_right, y_bottom, _ = box # 得到人脸bbox坐标
429.     crop_w = x_right - x_left + 1
430.     crop_h = y_bottom - y_top + 1
431.     # ignore box that is too small or beyond image border
432.     if crop_w < image_size / 2 or crop_h < image_size / 2:
433.         continue
434.     if x_left < 0 or y_top < 0:
435.         continue
436.
437.     # 当五个landmark都在bbox中时, 这样的样本作为我们的landmark训练样本
438.     if pts[0] < x_left or pts[0] > x_right:
439.         continue
440.     if pts[2] < x_left or pts[2] > x_right:
441.         continue
442.     if pts[4] < x_left or pts[4] > x_right:
443.         continue
444.     if pts[6] < x_left or pts[6] > x_right:
445.         continue
446.     if pts[8] < x_left or pts[8] > x_right:
447.         continue
448.
449.     if pts[1] < y_top or pts[1] > y_bottom:
450.         continue
451.     if pts[3] < y_top or pts[3] > y_bottom:
452.         continue
453.     if pts[5] < y_top or pts[5] > y_bottom:
454.         continue
455.     if pts[7] < y_top or pts[7] > y_bottom:
456.         continue
457.     if pts[9] < y_top or pts[9] > y_bottom:
458.         continue
459.
460.     # show image
461.     #img1 = drawBoxes_align(img, box)
462.     #img1 = drawlandmark_align(img1, pts)
463.     #cv2.imshow('img', img1)
464.     #cv2.waitKey(1000)
465.
466.     # our method, x0,y0,x1,y1,x2,y2,x3,y3,x4,y4
467.     #for k in range(len(pts) / 2):
468.     #    pts[k * 2] = (pts[k * 2] - x_left) / float(crop_w);
469.     #    pts[k * 2 + 1] = (pts[k * 2 + 1] - y_top) / float(crop_h);
470.
471.     #author method, x0,x1,x2,x3,x4,y0,y1,y2,y3,y4
472.     ptsss = pts[:]
473.     # landmark标注为其相对于crop区域左上角的归一化值
474.     for k in range(len(ptsss) / 2):
475.         pts[k] = (ptsss[k * 2] - x_left) / float(crop_w);
476.         pts[5+k] = (ptsss[k * 2 + 1] - y_top) / float(crop_h);
477.
478.     # 将crop区域进行resize
```



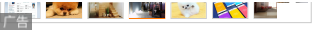
内容举报

返回顶部





```
479.         cropped_im = img[int(y_top):int(y_bottom + 1), int(x_left):int(x_right + 1)]
480.         resized_im = cv2.resize(cropped_im, (image_size, image_size), interpolation=cv2.INTER_LINEAR)
481.
482.         # box_ = box.reshape(1, -1)
483.         # 写图片名,class坐标,bbox坐标。
484.         save_file = os.path.join(landmark_save_dir, "%s.jpg" % l_idx)
485.         f1.write(str(image_size) + "/" + "landmark/" + "%s.jpg" % l_idx + ' -1 -1 -1 -1 -1')
486.
487.         # 写landmark坐标并保存图片
488.         for k in range(len(pts)):
489.             f1.write(" %f" % pts[k])
490.             f1.write("\n")
491.             cv2.imwrite(save_file, resized_im)
492.             l_idx += 1
493.
494.
495.
496.         f.close()
497.         f1.close()
498.
499.
500. if __name__ == "__main__":
501.     main()
```



版权声明：本文为博主原创文章，未经博主允许不得转载。

- 

发表评论
- (http://my.csdn.net/weixin_35068028)

 weixin_38424682 (weixin_38424682) 2017-11-04 02:00 2楼

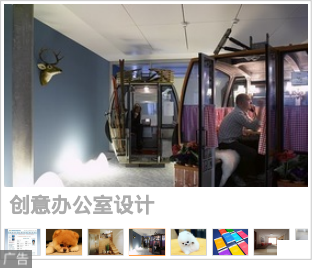
(weixin_38424682)作者提供的pr图是怎么作出来的，我用o-net输出的bbox和score 拿到widerface 的eval-tools上运行，效果非常差.....

回复
- (http://my.csdn.net/weixin_38424682)

 weixin_38424682 (weixin_38424682) 2017-11-04 01:57 1楼

(weixin_38424682)作者提供的pr图是怎么作出来的，我用o-net输出的bbox和score 拿到widerface 的eval-tools上运行，效果非常差.....

回复




1



相关文章推荐

基于Caffe的人脸检测实现 (http://blog.csdn.net/BBZZ2/article/details/52813031)

博客链接：http://blog.csdn.net/chenruiwei2/article/details/50321085 0. 引言深度学习可以说是在人脸分析相关领域遍地开花，近年来在人脸识别，深...

 BBZZ2 (http://blog.csdn.net/BBZZ2) 2016年10月14日 09:22 1030

人脸检测——DDFD (http://blog.csdn.net/shuzfan/article/details/49825751)

本文所介绍的人脸检测,主要学习和实现了ICMR-2015年雅虎实验室的文章"Multi-view Face Detection Using Deep Convolutional Neural Netw...

shuzfan (http://blog.csdn.net/shuzfan) 2015年11月22日 23:50 14107



一个普通程序员的内心独白....躺枪！躺枪！

我，一个普普通通程序员，没有过人的天赋，没有超乎寻常的好运，该如何逆袭走上人生巅峰？

广告

(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjDLnJT0lZ0qnFK9ujYzP1nsrjD10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1d9nh79rH0Luh79nWmzm17W0AwY5HDdnHc3rj6LrjD0lgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEXyN9T-KzUvdEIA-EUBqbugw9pysEn1qdAdxTvqdThP-5yF_UvTknoKzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqrjTLP0)

MTCNN训练整理 (http://blog.csdn.net/AMDS123/article/details/69568495)

MTCNN主要包括三个部分,PNet,RNet,ONet 其中PNet在训练阶段的输入尺寸为12*12,RNet的输入尺寸为24*24, ONet的输入尺寸为48*48. PNet网络参数最小, c...

AMDS123 (http://blog.csdn.net/AMDS123) 2017年04月07日 16:11 12097

人脸检测和对齐--MTCNN训练1--P-net (http://blog.csdn.net/fuwenyan/article/details/73613...)

采用数据集：人脸检测和人脸框回归：WIDER_FACE 特征点标定：Celeba（P-net未使用） P-net: P-net的任务是人脸检测和人脸框回归，所以该阶段仅需要使用WIDER FAC...

fuwenyan (http://blog.csdn.net/fuwenyan) 2017年06月23日 10:29 1953

MTCNN (Multi-task convolutional neural networks) 人脸对齐 (http://blog.csdn.net/qq_1...

该MTCNN算法出自深圳先进技术研究院，乔宇老师组，是今年2016的ECCV。（至少我知道的今年已经一篇cvpr，一篇eccv了）。 进入正题 理论基础：正如上...

qq_14845119 (http://blog.csdn.net/qq_14845119) 2016年09月27日 15:23 21558



AI 工程师职业指南

我们请来商汤、杜邦、声智、希为、58同城、爱因互动、中科视拓、鲁朗软件等公司 AI 技术一线的专家，请他们从实践的角度来解析 AI 领域各技术岗位的合格工程师都是怎样炼成的。

(http://www.baidu.com/cb.php?c=lgF_pyfqHmknjJfzrj00lZ0qnFK9ujYzP1f4Pjnd0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YdnhDYnHD1m1FBPHbsnH6z0AwY5HDdnHc3rj6LrjD0lgF_5y9YIZ0IQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1R4uhF-rA7Wuj0YmhP9PARvujmYmH0vm1qdAdxTvqdThP-5HDknWF9mnhEusKzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHnsnJOYnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPWnYPWR)

MTCNN训练数据整理 (http://blog.csdn.net/AMDS123/article/details/55509600)

分为三层网路PRO 1. 输入: 积分图像 输出: nroh1 边框概率，可以得出粗略得边框信息 conv4-2 边框偏移 2. 输入：根据第一



内容举报

返回顶部

内容举报

1

步提取的边框，提取图片，作为batch进行输入 输出：...

 AMDS123 (<http://blog.csdn.net/AMDS123>) 2017年02月17日 14:25  8677



Windows下cmake编译caffe，实现纯C++版本MTCNN人脸检测和关键点定位 (<http://blog.csdn.net/XZZPPP/article/details/52668935>)

2017.5.28 发现 caffe官方windows版本已经不提供vs工程文件了，需要用cmake编译生成sln文件。详细编译调试过程如下：
一、开发环境要求： Win...

 XZZPPP (<http://blog.csdn.net/XZZPPP>) 2017年06月14日 11:06  3546



人脸检测——MTCNN (<http://blog.csdn.net/shuzfan/article/details/52668935>)

本次介绍一篇速度还不错的人脸检测文章：《2016 Joint Face Detection and Alignment using Multi-task Cascaded Convolutional ...

 shuzfan (<http://blog.csdn.net/shuzfan>) 2016年09月26日 11:56  28960

300W数据集测试MTCNN的landmark效果代码 (<http://blog.csdn.net/hk121/article/details/7753187>)

300W数据集测试MTCNN的landmark效果，用提取其中afw数据集337张图片的预测关键点并写入到txt中，再用测试程序和标注landmark做对比。处理得到的预测landmark...

 hk121 (<http://blog.csdn.net/hk121>) 2017年08月09日 23:39  447



用我们训练的MTCNN中o-net测试训练图片的landmark的mean error (<http://blog.csdn.net/XZZPPP/article/details/7753187>)

为了验证我们训练的MTCNN的O-Net在训练集上的表现，我们写了一个测试代码，来测试训练图片的landmark的mean error。landmark标签格式如下所示： 48/landmark...

 XZZPPP (<http://blog.csdn.net/XZZPPP>) 2017年08月30日 21:13  527



mtcnn V1 版本和v2 版本的不同 (<http://blog.csdn.net/keyanxiaocaicai/article/details/71915345>)

facial5points = [105.8306, 147.9323, 121.3533, 106.1169, 144.3622; ... 109.8005, 1...

 keyanxiaocaicai (<http://blog.csdn.net/keyanxiaocaicai>) 2017年05月14日 01:05  577

VGG人脸识别训练心得 (http://blog.csdn.net/small_ARM/article/details/72835345)

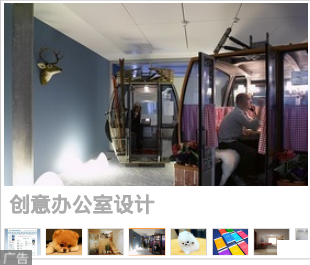
在使用VGG进行人脸训练过程中，我是用log日志记录了训练过程中的数据。训练过程中的参数对于训练而言，有着较大的影响，参数设置不当，容易出现训练结束，但未收敛到最佳状态...

 small_ARM (http://blog.csdn.net/small_ARM) 2017年06月01日 21:16  1216

300W数据集测试MTCNN的landmark效果代码 (<http://blog.csdn.net/XZZPPP/article/details/7753187>)

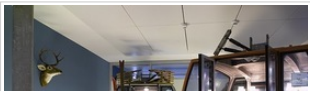
300W数据集测试MTCNN的landmark效果，用提取其中afw数据集337张图片的预测关键点并写入到txt中，再用测试程序和标注landmark做对比。处理得到的预测landmark...

 返回顶部



 内容举报

 返回顶部





1

XZZPPP (<http://blog.csdn.net/XZZPPP>) 2017年07月31日 20:26 859

人脸检测——MTCNN (<http://blog.csdn.net/u014696921/article/details/65446250>)

本次介绍一篇速度还不错的人脸检测文章：《2016 Joint Face Detection and Alignment using Multi-task Cascaded Convolut iona...

u014696921 (<http://blog.csdn.net/u014696921>) 2017年03月23日 17:25 3508

MTCNN (Multi-task convolutional neural networks) 人脸对齐 (<http://blog.csdn.net/oppo62258801>)

该MTCNN算法出自深圳先进技术研究院，乔宇老师组，是今年2016的ECCV。（至少我知道的今年已经一篇cvpr，一篇eccv了）。 进入正题 理论基础： ...

oppo62258801 (<http://blog.csdn.net/oppo62258801>) 2017年03月24日 19:56 1825

mxnet 使用自己的图片数据训练CNN模型 (<http://blog.csdn.net/u014696921/article/details/5...>)

前言 很久之前笔者就配好了mxnet环境，却因为种种原因始终没有进行系统的学习，其中一个原因是mxnet的文档不甚详细，相关教程数量和TensorFlow等框架相比也比较少，不太容易上手。因此， ...

u014696921 (<http://blog.csdn.net/u014696921>) 2017年02月24日 21:05 3779

MTCNN中将自己训练的o-net接在作者提供的net1,net2后面python代码解读 (<http://blog.csdn...>)

基于原作者提供的python版本MTCNN修改，开源地址：https://github.com/DuinoDu/mtcnn 我们的o-net训练代码参考如下开源项目：https://github.c...

XZZPPP (<http://blog.csdn.net/XZZPPP>) 2017年08月28日 19:57 866

face detection and align (<http://blog.csdn.net/andeyeluguo/article/details/62234880>)

从一个人脸数据集中获取人脸的五官位置// caffe #include #include // c++ #include #include #include // opencv #incl...

andeyeluguo (<http://blog.csdn.net/andeyeluguo>) 2017年03月15日 18:59 394

MTCNN解读：Joint Face Detection and Alignment using Multi-task Cascaded Convolutio...

解读论文为《Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks》 基于多任务级联卷积...

fuwenyan (<http://blog.csdn.net/fuwenyan>) 2017年06月14日 11:22 2153

MTCNN算法提速效果展示 (<http://blog.csdn.net/samylee/article/details/78421960>)

mtcnn算法工程应用的系列博客博主已经介绍了一部分了，可参考下列博客： 人脸检测之MTCNN训练自己的数据（切记：训练时一定要保证neg:pos:part:landmark为3:1:1:2，不然会出...

samylee (<http://blog.csdn.net/samylee>) 2017年11月02日 10:48 643



内容举报



返回顶部



1

1

1

1