

[CSDN首页 \(http://www.csdn.net?ref=toolbar\)](http://www.csdn.net?ref=toolbar)[学院 \(http://edu.csdn.net?ref=toolbar\)](http://edu.csdn.net?ref=toolbar)[下载 \(http://download.csdn.net?ref=toolbar\)](http://download.csdn.net?ref=toolbar)

更多 ▾

[下载 CSDN APP \(http://www.csdn.net/app?ref=toolbar\)](http://www.csdn.net/app?ref=toolbar)[写博客 \(http://write.blog.csdn.net/postedit?ref=toolbar\)](http://write.blog.csdn.net/postedit?ref=toolbar)[登录 \(https://passport.csdn.net/account/login?ref=toolbar\)](http://blog.csdn.net/) | [注册 \(http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister\)](http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister)[首页 \(http://blog.csdn.net/\)](http://blog.csdn.net/)[移动开发 \(http://blog.csdn.net/mobile/newarticle.html\)](http://blog.csdn.net/mobile/newarticle.html)

全部 ▾

CSDN (<http://www.csdn.net>)目录 **SSD框架训练自己的数据集**

喜欢 转载 2017年05月02日 08:03:43

656

0

0



收藏



评论



分享

SSD框架训练自己的数据集 (<http://www.cnblogs.com/objectDetect/p/57>)

SSD (<https://github.com/weiliu89/caffe/tree/ssd#traineval>) demo中详细介绍了如何在VOC数据集上使用SSD进行物体检测的训练和验证。
本文介绍如何使用SSD实现对自己数据集的训练和验证过程，内容包括：

mindfusion_heu ([http://bl...](http://blog.csdn.net/jqw11))

+ 关注

(<http://blog.csdn.net/jqw11>)

原创

粉丝

喜欢

码云

20

0

0

他的最新文章

[更多文章 \(http://blog.csdn.net/jqw11\)](http://blog.csdn.net/jqw11)

linux下gcc编译使用opencv的源文件时报错的处理：undefined reference to symbol

'_ZNSSD1Ev@@GLIBCXX_3.4'
([/jqw11/article/details/74455376](http://blog.csdn.net/jqw11/article/details/74455376))

Python-OpenCV 处理图像（一）：基本操作 cv2
([/jqw11/article/details/73826014](http://blog.csdn.net/jqw11/article/details/73826014))

关于python 和C++使用cv画矩形并填充颜色同时填充文字
([/jqw11/article/details/73332166](http://blog.csdn.net/jqw11/article/details/73332166))

Chipscope信号无法找到的解决方案
([/jqw11/article/details/73123453](http://blog.csdn.net/jqw11/article/details/73123453))

- 1 数据集的标注
- 2 数据集的转换
- 3 使用SSD如何训练
- 4 使用SSD如何测试

1 数据集的标注



目录



喜欢



收藏



评论



分享

数据的标注使用BBox-Label-Tool (<https://github.com/puzzledqs/BBox-Label-Tool>)

工具，该工具使用python实现，使用简单方便。修改后的工具支持多label的标签标注。

该工具生成的标签格式是：

object_number

className x1min y1min x1max y1max

classname x2min y2min x2max y2max

...

1.1 labelTool工具的使用说明

BBox-Label-Tool工具实现较简单，原始的git版本使用起来有一些小问题，进行了简单的修改，修改后的版本

编辑推荐

最热专栏

将数据集做成VOC2007格式用于Faster...

Faster-RCNN+ZF用自己的数据集训练...

SSD框架训练自己的数据集 (/yx2017/a...

SSD框架训练自己的数据集 (/haima199...

在线课程



Presto的服务治理与架构

在京东的实践与应用

(<http://edu.csdn.net/huiyi>

Course/series_detail/64?

utm_source=blog9)



深入掌握Kubernetes应用

实践

([http://edu.csdn.net/course/detail/6080?](http://edu.csdn.net/course/detail/6080?utm_source=blog9)

utm_source=blog9)



目录



喜欢



收藏



评论



分享

```

#-----
# Name:      Object bounding box label tool
# Purpose:   Label object bboxes for ImageNet Detection data
# Author:    Qiushi
# Created:   06/06/2014
#
#-----
from __future__ import division
from Tkinter import *
import tkMessageBox
from PIL import Image, ImageTk
import os
import glob
import random

# colors for the bboxes
COLORS = ['red', 'blue', 'yellow', 'pink', 'cyan', 'green', 'black']
# image sizes for the examples
SIZE = 256, 256

classLabels=['mat', 'door', 'sofa', 'chair', 'table', 'bed', 'ashcan', 'shoe']

class LabelTool():
    def __init__(self, master):
        # set up the main frame
        self.parent = master
        self.parent.title("LabelTool")
        self.frame = Frame(self.parent)
        self.frame.pack(fill=BOTH, expand=1)
        self.parent.resizable(width = False, height = False)

        # initialize global state
        self.imageDir = ''
        self.imageList= []
        self.egDir = ''
        self.egList = []
        self.outDir = ''
        self.cur = 0
        self.total = 0
        self.category = 0

```

```
self.imagename = ''
self.labelfilename = ''
self.tkimg = None

# initialize mouse state
self.STATE = {}
self.STATE['click'] = 0
self.STATE['x'], self.STATE['y'] = 0, 0

# reference to bbox
self.bboxIdList = []
self.bboxId = None
self.bboxList = []
self.hl = None
self.vl = None
self.currentClass = ''

# ----- GUI stuff -----
# dir entry & load
self.label = Label(self.frame, text = "Image Dir:")
self.label.grid(row = 0, column = 0, sticky = E)
self.entry = Entry(self.frame)
self.entry.grid(row = 0, column = 1, sticky = W+E)
self.ldBtn = Button(self.frame, text = "Load", command = self.loadDir)
self.ldBtn.grid(row = 0, column = 2, sticky = W+E)

# main panel for labeling
self.mainPanel = Canvas(self.frame, cursor='tcross')
self.mainPanel.bind("<Button-1>", self.mouseClick)
self.mainPanel.bind("<Motion>", self.mouseMove)
self.parent.bind("<Escape>", self.cancelBBox) # press <Escape> to cancel current bbox
self.parent.bind("s", self.cancelBBox)
self.parent.bind("a", self.prevImage) # press 'a' to go backforward
self.parent.bind("d", self.nextImage) # press 'd' to go forward
self.mainPanel.grid(row = 1, column = 1, rowspan = 4, sticky = W+N)

# showing bbox info & delete bbox
self.lb1 = Label(self.frame, text = 'Bounding boxes:')
self.lb1.grid(row = 1, column = 2, sticky = W+N)
self.listbox = Listbox(self.frame, width = 22, height = 12)
self.listbox.grid(row = 2, column = 2, sticky = N)
```



目录



喜欢



收藏



评论



分享



目录



喜欢



收藏



评论



分享

```

self.btnDel = Button(self.frame, text = 'Delete', command = self.delBBox)
self.btnDel.grid(row = 3, column = 2, sticky = W+E+N)
self.btnClear = Button(self.frame, text = 'ClearAll', command = self.clearBBox)
self.btnClear.grid(row = 4, column = 2, sticky = W+E+N)

#select class type
self.classPanel = Frame(self.frame)
self.classPanel.grid(row = 5, column = 1, columnspan = 10, sticky = W+E)
label = Label(self.classPanel, text = 'class:')
label.grid(row = 5, column = 1, sticky = W+N)

self.classbox = Listbox(self.classPanel, width = 4, height = 2)
self.classbox.grid(row = 5, column = 2)
for each in range(len(classLabels)):
    function = 'select' + classLabels[each]
    print classLabels[each]
    btnMat = Button(self.classPanel, text = classLabels[each], command = getattr(self,
function))
    btnMat.grid(row = 5, column = each + 3)

# control panel for image navigation
self.ctrPanel = Frame(self.frame)
self.ctrPanel.grid(row = 6, column = 1, columnspan = 2, sticky = W+E)
self.prevBtn = Button(self.ctrPanel, text='<< Prev', width = 10, command = self.prevImage)
self.prevBtn.pack(side = LEFT, padx = 5, pady = 3)
self.nextBtn = Button(self.ctrPanel, text='Next >>', width = 10, command = self.nextImage)
self.nextBtn.pack(side = LEFT, padx = 5, pady = 3)
self.progLabel = Label(self.ctrPanel, text = "Progress:      /      ")
self.progLabel.pack(side = LEFT, padx = 5)
self.tmpLabel = Label(self.ctrPanel, text = "Go to Image No.")
self.tmpLabel.pack(side = LEFT, padx = 5)
self.idxEntry = Entry(self.ctrPanel, width = 5)
self.idxEntry.pack(side = LEFT)
self.goBtn = Button(self.ctrPanel, text = 'Go', command = self.gotoImage)
self.goBtn.pack(side = LEFT)

# example pannel for illustration
self.egPanel = Frame(self.frame, border = 10)
self.egPanel.grid(row = 1, column = 0, rowspan = 5, sticky = N)

```



目录



喜欢



收藏



评论



分享

```

self.tmpLabel2 = Label(self.egPanel, text = "Examples:")
self.tmpLabel2.pack(side = TOP, pady = 5)
self.egLabels = []
for i in range(3):
    self.egLabels.append(Label(self.egPanel))
    self.egLabels[-1].pack(side = TOP)

# display mouse position
self.disp = Label(self.ctrPanel, text='')
self.disp.pack(side = RIGHT)

self.frame.columnconfigure(1, weight = 1)
self.frame.rowconfigure(10, weight = 1)

# for debugging
self.setImage()
self.loadDir()

def loadDir(self, dbg = False):
    if not dbg:
        s = self.entry.get()
        self.parent.focus()
        self.category = int(s)
    else:
        s = r'D:\workspace\python\labelGUI'
        if not os.path.isdir(s):
            tkMessageBox.showerror("Error!", message = "The specified dir doesn't exist!")
            return
    # get image list
    self.imageDir = os.path.join(r'./Images', '%d' %(self.category))
    self.imageList = glob.glob(os.path.join(self.imageDir, '*.jpg'))
    if len(self.imageList) == 0:
        print 'No .JPEG images found in the specified dir!'
        return

# set up output dir
self.outDir = os.path.join(r'./Labels', '%d' %(self.category))
if not os.path.exists(self.outDir):
    os.mkdir(self.outDir)

labeledPicList = glob.glob(os.path.join(self.outDir, '*.txt'))

```



目录



喜欢



收藏



评论



分享

```

for label in labeledPicList:
    data = open(label, 'r')
    if '0\n' == data.read():
        data.close()
        continue
    data.close()
    picture = label.replace('Labels', 'Images').replace('.txt', '.jpg')
    if picture in self.imageList:
        self.imageList.remove(picture)
# default to the 1st image in the collection
self.cur = 1
self.total = len(self.imageList)
self.loadImage()
print '%d images loaded from %s' %(self.total, s)

def loadImage(self):
    # load image
    imagepath = self.imageList[self.cur - 1]
    self.img = Image.open(imagepath)
    self.imgSize = self.img.size
    self.tking = ImageTk.PhotoImage(self.img)
    self.mainPanel.config(width = max(self.tking.width(), 400), height = max(self.tking.height(), 400))
    self.mainPanel.create_image(0, 0, image = self.tking, anchor=NW)
    self.progLabel.config(text = "%04d/%04d" %(self.cur, self.total))

    # load labels
    self.clearBBox()
    self.imagename = os.path.split(imagepath)[-1].split('.')[0]
    labelname = self.imagename + '.txt'
    self.labelfilename = os.path.join(self.outDir, labelname)
    bbox_cnt = 0
    if os.path.exists(self.labelfilename):
        with open(self.labelfilename) as f:
            for (i, line) in enumerate(f):
                if i == 0:
                    bbox_cnt = int(line.strip())
                    continue
                tmp = [int(t.strip()) for t in line.split()]
                print tmp
    ##

```



目录



喜欢



收藏



评论



分享

```

        self.bboxList.append(tuple(tmp))
        tmpId = self.mainPanel.create_rectangle(tmp[0], tmp[1], \
                                                tmp[2], tmp[3], \
                                                width = 2, \
                                                outline = COLORS[(len(self.bboxLis
t)-1) % len(COLORS)])
        self.bboxIdList.append(tmpId)
        self.listbox.insert(END, '(%d, %d) -> (%d, %d)' %(tmp[0], tmp[1], tmp[2], t
mp[3]))
        self.listbox.itemconfig(len(self.bboxIdList) - 1, fg = COLORS[(len(self.bbo
xIdList) - 1) % len(COLORS)])

def saveImage(self):
    with open(self.labelfilename, 'w') as f:
        f.write('%d\n' %len(self.bboxList))
        for bbox in self.bboxList:
            f.write(' '.join(map(str, bbox)) + '\n')
        print 'Image No. %d saved' %(self.cur)

def mouseClicked(self, event):
    if self.STATE['click'] == 0:
        self.STATE['x'], self.STATE['y'] = event.x, event.y
        #self.STATE['x'], self.STATE['y'] = self.imgSize[0], self.imgSize[1]
    else:
        x1, x2 = min(self.STATE['x'], event.x), max(self.STATE['x'], event.x)
        y1, y2 = min(self.STATE['y'], event.y), max(self.STATE['y'], event.y)
        if x2 > self.imgSize[0]:
            x2 = self.imgSize[0]
        if y2 > self.imgSize[1]:
            y2 = self.imgSize[1]
        self.bboxList.append((self.currentClass, x1, y1, x2, y2))
        self.bboxIdList.append(self.bboxId)
        self.bboxId = None
        self.listbox.insert(END, '(%d, %d) -> (%d, %d)' %(x1, y1, x2, y2))
        self.listbox.itemconfig(len(self.bboxIdList) - 1, fg = COLORS[(len(self.bboxIdList)
- 1) % len(COLORS)])
        self.STATE['click'] = 1 - self.STATE['click']

def mouseMove(self, event):
    self.disp.config(text = 'x: %d, y: %d' %(event.x, event.y))

```




目录



喜欢



收藏



评论



分享

```

        if self.tking:
            if self.hl:
                self.mainPanel.delete(self.hl)
            self.hl = self.mainPanel.create_line(0, event.y, self.tking.width(), event.y, width
= 2)

            if self.vl:
                self.mainPanel.delete(self.vl)
            self.vl = self.mainPanel.create_line(event.x, 0, event.x, self.tking.height(), width
h = 2)

        if 1 == self.STATE['click']:
            if self.bboxId:
                self.mainPanel.delete(self.bboxId)
            self.bboxId = self.mainPanel.create_rectangle(self.STATE['x'], self.STATE['y'], \
                event.x, event.y, \
                width = 2, \
                outline = COLORS[len(self.bboxList)
% len(COLORS)])

    def cancelBBox(self, event):
        if 1 == self.STATE['click']:
            if self.bboxId:
                self.mainPanel.delete(self.bboxId)
                self.bboxId = None
                self.STATE['click'] = 0

    def delBBox(self):
        sel = self.listbox.curselection()
        if len(sel) != 1 :
            return
        idx = int(sel[0])
        self.mainPanel.delete(self.bboxIdList[idx])
        self.bboxIdList.pop(idx)
        self.bboxList.pop(idx)
        self.listbox.delete(idx)

    def clearBBox(self):
        for idx in range(len(self.bboxIdList)):
            self.mainPanel.delete(self.bboxIdList[idx])
        self.listbox.delete(0, len(self.bboxList))
        self.bboxIdList = []
        self.bboxList = []

```



目录



喜欢



收藏



评论



分享

```
def selectmat(self):
    self.currentClass = 'mat'
    self.classbox.delete(0,END)
    self.classbox.insert(0, 'mat')
    self.classbox.itemconfig(0,fg = COLORS[0])

def selectdoor(self):
    self.currentClass = 'door'
    self.classbox.delete(0,END)
    self.classbox.insert(0, 'door')
    self.classbox.itemconfig(0,fg = COLORS[0])

def selectsofa(self):
    self.currentClass = 'sofa'
    self.classbox.delete(0,END)
    self.classbox.insert(0, 'sofa')
    self.classbox.itemconfig(0,fg = COLORS[0])

def selectchair(self):
    self.currentClass = 'chair'
    self.classbox.delete(0,END)
    self.classbox.insert(0, 'chair')
    self.classbox.itemconfig(0,fg = COLORS[0])

def selecttable(self):
    self.currentClass = 'table'
    self.classbox.delete(0,END)
    self.classbox.insert(0, 'table')
    self.classbox.itemconfig(0,fg = COLORS[0])

def selectbed(self):
    self.currentClass = 'bed'
    self.classbox.delete(0,END)
    self.classbox.insert(0, 'bed')
    self.classbox.itemconfig(0,fg = COLORS[0])

def selectashcan(self):
    self.currentClass = 'ashcan'
    self.classbox.delete(0,END)
    self.classbox.insert(0, 'ashcan')
```



目录



喜欢



收藏



评论



分享

```
        self.classbox.itemconfig(0, fg = COLORS[0])

def selectshoe(self):
    self.currentClass = 'shoe'
    self.classbox.delete(0, END)
    self.classbox.insert(0, 'shoe')
    self.classbox.itemconfig(0, fg = COLORS[0])

def prevImage(self, event = None):
    self.saveImage()
    if self.cur > 1:
        self.cur -= 1
        self.loadImage()

def nextImage(self, event = None):
    self.saveImage()
    if self.cur < self.total:
        self.cur += 1
        self.loadImage()

def gotoImage(self):
    idx = int(self.idxEntry.get())
    if 1 <= idx and idx <= self.total:
        self.saveImage()
        self.cur = idx
        self.loadImage()

## def setImage(self, imagepath = r'test2.png'):
##     self.img = Image.open(imagepath)
##     self.tking = ImageTk.PhotoImage(self.img)
##     self.mainPanel.config(width = self.tking.width())
##     self.mainPanel.config(height = self.tking.height())
##     self.mainPanel.create_image(0, 0, image = self.tking, anchor=NW)

if __name__ == '__main__':
    root = Tk()
    tool = LabelTool(root)
    root.mainloop()
```

main.py

使用方法：

- (1) 在BBox-Label-Tool/Images目录下创建保存图片的目录，目录以数字命名(BBox-Label-Tool/Images/1), 然后将待标注的图片copy到1这个目录下;
- (2) 在BBox-Label-Tool目录下执行命令 `python main.py`
- (3) 在工具界面上, Image Dir 框中输入需要标记的目录名(比如 1), 然后点击load按钮, 工具自动将Images/1目录下的图片加载进来;

需要说明一下, 如果目录中的图片已经标注过, 点击load时不会被重新加载进来.

- (4) 该工具支持多类别标注, 画bounding boxes框标定之前, 需要先选定类别, 然后再画框.

- (5) 一张图片标注完后, 点击Next>>按钮, 标注下一张图片, 图片label成功后, 会在BBox-Label-



目录 Tool/Labels对应的目录下生成与图片文件名对应的label文件.



喜欢



收藏



评论



分享

2 数据集的转换

caffe训练使用LMDB格式的数据, ssd框架中提供了voc数据格式转换成LMDB格式的脚本。
所以实践中先将BBox-Label-Tool标注的数据转换成voc数据格式, 然后再转换成LMDB格式。

2.1 voc数据格式

是xml格式的label信息

(1)Annotations中保存的



目录



喜欢



收藏



评论



分享

```
<?xml version="1.0" ?>
<annotation>
  <folder>VOC2007</folder>
  <filename>1.jpg</filename>
  <source>
    <database>My Database</database>
    <annotation>VOC2007</annotation>
    <image>flickr</image>
    <flickrid>NULL</flickrid>
  </source>
  <owner>
    <flickrid>NULL</flickrid>
    <name>idaneel</name>
  </owner>
  <size>
    <width>320</width>
    <height>240</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>door</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>109</xmin>
      <ymin>3</ymin>
      <xmax>199</xmax>
      <ymax>204</ymax>
    </bndbox>
  </object>
</annotation>
```

VOC XML内容信息

	(2) ImageSet 目录下的Main 目录里存放的是用于表示训练的图片集和测试的图片集	
		(3) JPEGImages 目录
下存放所有图片集		
	(4) label 目录下保存的是BBBox-Label-Tool 工具标注好的bounding box 坐标文件， 该目录下的文件就是待转换的label 标签文件。	



目录



喜欢



收藏



评论



分享

2.2 Label转换成VOC数据格式

BBox-Label-Tool工具标注好的bounding box坐标文件转换成VOC数据格式的形式。

具体的转换过程包括了两个步骤：

(1) 将BBox-Label-Tool下的txt格式保存的bounding box信息转换成VOC数据格式下以xml方式表示；

(2) 生成用于训练的数据集和用于测试的数据集。

用python实现了上述两个步骤的转换。

createXml.py 完成txt到xml的转换； 执行脚本./createXml.py



目录



喜欢



收藏



评论



分享

```
#!/usr/bin/env python

import os
import sys
import cv2
from itertools import islice
from xml.dom.minidom import Document

labels='label'
imgpath='JPEGImages/'
xmlpath_new='Annotations/'
foldername='VOC2007'

def insertObject(doc, datas):
    obj = doc.createElement('object')
    name = doc.createElement('name')
    name.appendChild(doc.createTextNode(datas[0]))
    obj.appendChild(name)
    pose = doc.createElement('pose')
    pose.appendChild(doc.createTextNode('Unspecified'))
    obj.appendChild(pose)
    truncated = doc.createElement('truncated')
    truncated.appendChild(doc.createTextNode(str(0)))
    obj.appendChild(truncated)
    difficult = doc.createElement('difficult')
    difficult.appendChild(doc.createTextNode(str(0)))
    obj.appendChild(difficult)
    bndbox = doc.createElement('bndbox')

    xmin = doc.createElement('xmin')
    xmin.appendChild(doc.createTextNode(str(datas[1])))
    bndbox.appendChild(xmin)

    ymin = doc.createElement('ymin')
    ymin.appendChild(doc.createTextNode(str(datas[2])))
    bndbox.appendChild(ymin)
    xmax = doc.createElement('xmax')
    xmax.appendChild(doc.createTextNode(str(datas[3])))
    bndbox.appendChild(xmax)
    ymax = doc.createElement('ymax')
```



目录



喜欢



收藏



评论



分享


```

if '\r' == str(datas[4])[-1] or '\n' == str(datas[4])[-1]:
    data = str(datas[4])[0:-1]
else:
    data = str(datas[4])
ymax.appendChild(doc.createTextNode(data))
bndbox.appendChild(ymax)
obj.appendChild(bndbox)
return obj

```



def create():

目录



喜欢



收藏



评论



分享

```

for walk in os.walk(labels):
    for each in walk[2]:
        fidin=open(walk[0] + '/' + each, 'r')
        objIndex = 0
        for data in islice(fidin, 1, None):
            objIndex += 1
            data=data.strip('\n')
            datas = data.split(' ')
            if 5 != len(datas):
                print 'bounding box information error'
                continue
            pictureName = each.replace('.txt', '.jpg')
            imageFile = imgpath + pictureName
            img = cv2.imread(imageFile)
            imgSize = img.shape
            if 1 == objIndex:
                xmlName = each.replace('.txt', '.xml')
                f = open(xmlpath_new + xmlName, "w")
                doc = Document()
                annotation = doc.createElement('annotation')
                doc.appendChild(annotation)

                folder = doc.createElement('folder')
                folder.appendChild(doc.createTextNode(foldername))
                annotation.appendChild(folder)

                filename = doc.createElement('filename')
                filename.appendChild(doc.createTextNode(pictureName))
                annotation.appendChild(filename)

                source = doc.createElement('source')

```



目录



喜欢



收藏



评论



分享

```
database = doc.createElement('database')
database.appendChild(doc.createTextNode('My Database'))
source.appendChild(database)
source_annotation = doc.createElement('annotation')
source_annotation.appendChild(doc.createTextNode(foldername))
source.appendChild(source_annotation)
image = doc.createElement('image')
image.appendChild(doc.createTextNode('flickr'))
source.appendChild(image)
flickrid = doc.createElement('flickrid')
flickrid.appendChild(doc.createTextNode('NULL'))
source.appendChild(flickrid)
annotation.appendChild(source)

owner = doc.createElement('owner')
flickrid = doc.createElement('flickrid')
flickrid.appendChild(doc.createTextNode('NULL'))
owner.appendChild(flickrid)
name = doc.createElement('name')
name.appendChild(doc.createTextNode('idaneel'))
owner.appendChild(name)
annotation.appendChild(owner)

size = doc.createElement('size')
width = doc.createElement('width')
width.appendChild(doc.createTextNode(str(imgSize[1])))
size.appendChild(width)
height = doc.createElement('height')
height.appendChild(doc.createTextNode(str(imgSize[0])))
size.appendChild(height)
depth = doc.createElement('depth')
depth.appendChild(doc.createTextNode(str(imgSize[2])))
size.appendChild(depth)
annotation.appendChild(size)

segmented = doc.createElement('segmented')
segmented.appendChild(doc.createTextNode(str(0)))
annotation.appendChild(segmented)
annotation.appendChild(insertObject(doc, datas))
else:
    annotation.appendChild(insertObject(doc, datas))
```

```
try:
    f.write(doc.toprettyxml(indent = '    '))
    f.close()
    fidin.close()
except:
    pass
```

```
if __name__ == '__main__':
    create()
```



目录



喜欢



收藏



评论



分享

createXml.py

createTest.py 生成训练集和测试集标识文件；执行脚本

./createTest.py %startID% %endID% %testNumber%



目录



喜欢



收藏



评论



分享

```
#!/usr/bin/env python

import os
import sys
import random

try:
    start = int(sys.argv[1])
    end = int(sys.argv[2])
    test = int(sys.argv[3])
    allNum = end-start+1
except:
    print 'Please input picture range'
    print './createTest.py 1 1500 500'
    os._exit(0)

b_list = range(start,end)
blist_webId = random.sample(b_list, test)
blist_webId = sorted(blist_webId)
allFile = []

testFile = open('ImageSets/Main/test.txt', 'w')
trainFile = open('ImageSets/Main/trainval.txt', 'w')

for i in range(allNum):
    allFile.append(i+1)

for test in blist_webId:
    allFile.remove(test)
    testFile.write(str(test) + '\n')

for train in allFile:
    trainFile.write(str(train) + '\n')
testFile.close()
trainFile.close()
```

createTest.py

说明：由于BBox-Label-Tool实现相对简单，该工具每次只能对一个类别进行打标签，所以转换脚本每一次也是对一个类别进行数据的转换，这个问题后续需要优化改进。

优化后的BBox-Label-Tool工具，支持多类别标定，生成的label文件中增加了类别名称信息。

使用时修改classLabels，改写成自己的类别，修改后的工具代码参见1.1中的main.py

2.3 VOC数据转换成LMDB数据

SSD提供了VOC数据到LMDB数据的转换脚本 data/VOC0712/create_list.sh 和 ./data/VOC0712/create_data.sh，这两个脚本是完全针对VOC0712目录下的数据进行的转换。

实现中为了不破坏VOC0712目录下的数据内容，针对我们自己的数据集，修改了上面这两个脚本，将脚本中涉及到VOC0712的信息替换成我们自己的目录信息。

在处理我们的数据集时，将VOC0712替换成indoor。

具体的步骤如下：

(1) 在 \$HOME/data/VOCdevkit目录下创建indoor目录，该目录中存放自己转换完成的VOC数据集；

(2) \$CAFFE_ROOT/examples目录下创建indoor目录；

(3) \$CAFFE_ROOT/data目录下创建indoor目录，同时将data/VOC0712下的create_list.sh,create_data.sh,labelmap_voc.prototxt

这三个文件copy到indoor目录下，分别重命名为create_list_indoor.sh,create_data_indoor.sh, labelmap_indoor.prototxt

(4)对上面新生成的两个create文件进行修改，主要修改是将VOC0712相关的信息替换成indoor
修改后的这两个文件分别为：



目录



喜欢



收藏



评论



分享



目录



喜欢



收藏



评论



分享

```
#!/bin/bash

root_dir=$HOME/data/VOCdevkit/
sub_dir=ImageSets/Main
bash_dir="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"

for dataset in trainval test
do
    dst_file=$bash_dir/$dataset.txt
    if [ -f $dst_file ]
    then
        rm -f $dst_file
    fi
    for name in indoor
    do
        if [[ $dataset == "test" && $name == "VOC2012" ]]
        then
            continue
        fi
        echo "Create list for $name $dataset..."
        dataset_file=$root_dir/$name/$sub_dir/$dataset.txt

        img_file=$bash_dir/$dataset"_img.txt"
        cp $dataset_file $img_file
        sed -i "s/^/$name\/JPEGImages\/g" $img_file
        sed -i "s/$/.jpg/g" $img_file

        label_file=$bash_dir/$dataset"_label.txt"
        cp $dataset_file $label_file
        sed -i "s/^/$name\/Annotations\/g" $label_file
        sed -i "s/$/.xml/g" $label_file

        paste -d' ' $img_file $label_file >> $dst_file

        rm -f $label_file
        rm -f $img_file
    done
    # Generate image name and size infomation.
    if [ $dataset == "test" ]
    then
        $bash_dir/../../build/tools/get_image_size $root_dir $dst_file $bash_dir/$dataset"_name_siz
```

```
e.txt"
fi

# Shuffle trainval file.
if [ $dataset == "trainval" ]
then
    rand_file=$dst_file.random
    cat $dst_file | perl -MList::Util=shuffle -e 'print shuffle(<STDIN>);' > $rand_file
    mv $rand_file $dst_file
fi
done
```



目录

create_list_indoor.sh



喜欢



收藏



评论



分享

```

cur_dir=$(cd $( dirname ${BASH_SOURCE[0]} ) && pwd )
root_dir=$cur_dir/../../

cd $root_dir

redo=1
data_root_dir="$HOME/data/VOCdevkit"
dataset_name="indoor"
mapfile="$root_dir/data/$dataset_name/labelmap_indoor.prototxt"
anno_type="detection"
db="lmbd"
min_dim=0
max_dim=0
width=0
height=0

extra_cmd="--encode-type=jpg --encoded"
if [ $redo ]
then
    extra_cmd="$extra_cmd --redo"
fi
for subset in test trainval
do
    python $root_dir/scripts/create_annotset.py --anno-type=$anno_type --label-map-file=$mapfile -
-min-dim=$min_dim --max-dim=$max_dim --resize-width=$width --resize-height=$height --check-labe
l $extra_cmd $data_root_dir $root_dir/data/$dataset_name/$subset.txt $data_root_dir/$dataset_na
me/$db/$dataset_name_"$subset_"$db examples/$dataset_name
done

```

create_data_indoor.sh

(5)修改labelmap_indoor.prototxt，将该文件中的类别修改成和自己的数据集相匹配，注意需要保留一个label 0，background类别

⊕ ⊞


```

item {
  name: "none_of_the_above"
  label: 0
  display_name: "background"
}
item {
  name: "door"
  label: 1
  display_name: "door"
}

```



目录 labelmap_indoor.prototxt



喜欢



收藏



评论



分享

完成上面步骤的修改后，可以开始LMDB数据数据的制作，在\$CAFFE_ROOT目录下分别运行：

```
./data/indoor/create_list_indoor.sh
```

```
./data/indoor/create_data_indoor.sh
```

命令执行完毕后，可以在\$CAFFE_ROOT/indoor目录下查看转换完成的LMDB数据数据。

3 使用SSD进行自己数据集的训练

训练时使用ssd demo中提供的预训练好的VGGnet model : VGG_ILSVRC_16_layers_fc_reduced.caffemodel (<https://gist.github.com/weiliu89/2ed6e13bfd5b57cf81d6>)

将该模型保存到\$CAFFE_ROOT/models/VGGNet下。

将ssd_pascal.py copy一份 ssd_pascal_indoor.py文件， 根据自己的数据集修改ssd_pascal_indoor.py

主要修改点：

(1) train_data和test_data修改成指向自己的数据集LMDB

```
train_data = "examples/indoor/indoor_trainval_lmdb"
```

```
test_data = "examples/indoor/indoor_test_lmdb"
```

(2) num_test_image该变量修改成自己数据集中测试数据的数量

(3) num_classes 该变量修改成自己数据集中 标签类别数量数 + 1

针对我的数据集，ssd_pascal_indoor.py的内容为：





目录



喜欢



收藏



评论



分享

```

from __future__ import print_function
import caffe
from caffe.model_libs import *
from google.protobuf import text_format

import math
import os
import shutil
import stat
import subprocess
import sys

# Add extra layers on top of a "base" network (e.g. VGGNet or Inception).
def AddExtraLayers(net, use_batchnorm=True):
    use_relu = True

    # Add additional convolutional layers.
    from_layer = net.keys()[-1]
    # TODO(weiliu89): Construct the name using the last layer to avoid duplication.
    out_layer = "conv6_1"
    ConvBNLayer(net, from_layer, out_layer, use_batchnorm, use_relu, 256, 1, 0, 1)

    from_layer = out_layer
    out_layer = "conv6_2"
    ConvBNLayer(net, from_layer, out_layer, use_batchnorm, use_relu, 512, 3, 1, 2)

    for i in xrange(7, 9):
        from_layer = out_layer
        out_layer = "conv{}_1".format(i)
        ConvBNLayer(net, from_layer, out_layer, use_batchnorm, use_relu, 128, 1, 0, 1)

        from_layer = out_layer
        out_layer = "conv{}_2".format(i)
        ConvBNLayer(net, from_layer, out_layer, use_batchnorm, use_relu, 256, 3, 1, 2)

    # Add global pooling layer.
    name = net.keys()[-1]
    net.pool6 = L.Pooling(net[name], pool=P.Pooling.AVE, global_pooling=True)

    return net

```



目录



喜欢



收藏



评论



分享

```

### Modify the following parameters accordingly ###
# The directory which contains the caffe code.
# We assume you are running the script at the CAFFE_ROOT.
caffe_root = os.getcwd()

# Set true if you want to start training right after generating all files.
run_soon = True
# Set true if you want to load from most recently saved snapshot.
# Otherwise, we will load from the pretrain_model defined below.
resume_training = True
# If true, Remove old model files.
remove_old_models = False

# The database file for training data. Created by data/VOC0712/create_data.sh
train_data = "examples/indoor/indoor_trainval_lmdb"
# The database file for testing data. Created by data/VOC0712/create_data.sh
test_data = "examples/indoor/indoor_test_lmdb"
# Specify the batch sampler.
resize_width = 300
resize_height = 300
resize = "{}x{}".format(resize_width, resize_height)
batch_sampler = [
    {
        'sampler': {
            'max_trials': 1,
            'max_sample': 1,
        },
        'sample_constraint': {
            'min_jaccard_overlap': 0.1,
        },
        'max_trials': 50,
        'max_sample': 1,
    },
    {
        'sampler': {
            'min_scale': 0.3,
            'max_scale': 1.0,
            'min_aspect_ratio': 0.5,
            'max_aspect_ratio': 2.0,
        },
        'sample_constraint': {
            'min_jaccard_overlap': 0.1,
        },
        'max_trials': 50,
        'max_sample': 1,
    },
]

```



目录



喜欢



收藏



评论



分享

```
},
{
    'sampler': {
        'min_scale': 0.3,
        'max_scale': 1.0,
        'min_aspect_ratio': 0.5,
        'max_aspect_ratio': 2.0,
    },
    'sample_constraint': {
        'min_jaccard_overlap': 0.3,
    },
    'max_trials': 50,
    'max_sample': 1,
},
{
    'sampler': {
        'min_scale': 0.3,
        'max_scale': 1.0,
        'min_aspect_ratio': 0.5,
        'max_aspect_ratio': 2.0,
    },
    'sample_constraint': {
        'min_jaccard_overlap': 0.5,
    },
    'max_trials': 50,
    'max_sample': 1,
},
{
    'sampler': {
        'min_scale': 0.3,
        'max_scale': 1.0,
        'min_aspect_ratio': 0.5,
        'max_aspect_ratio': 2.0,
    },
    'sample_constraint': {
        'min_jaccard_overlap': 0.7,
    },
    'max_trials': 50,
    'max_sample': 1,
},
{
```



目录



喜欢



收藏



评论



分享

```

        'sampler': {
            'min_scale': 0.3,
            'max_scale': 1.0,
            'min_aspect_ratio': 0.5,
            'max_aspect_ratio': 2.0,
        },
        'sample_constraint': {
            'min_jaccard_overlap': 0.9,
        },
        'max_trials': 50,
        'max_sample': 1,
    },
    {
        'sampler': {
            'min_scale': 0.3,
            'max_scale': 1.0,
            'min_aspect_ratio': 0.5,
            'max_aspect_ratio': 2.0,
        },
        'sample_constraint': {
            'max_jaccard_overlap': 1.0,
        },
        'max_trials': 50,
        'max_sample': 1,
    },
]
train_transform_param = {
    'mirror': True,
    'mean_value': [104, 117, 123],
    'resize_param': {
        'prob': 1,
        'resize_mode': P.Resize.WARP,
        'height': resize_height,
        'width': resize_width,
        'interp_mode': [
            P.Resize.LINEAR,
            P.Resize.AREA,
            P.Resize.NEAREST,
            P.Resize.CUBIC,
            P.Resize.LANCZOS4,
        ],
    },

```



目录



喜欢



收藏



评论



分享

```

        },
        'emit_constraint': {
            'emit_type': caffe_pb2.EmitConstraint.CENTER,
        }
    }
test_transform_param = {
    'mean_value': [104, 117, 123],
    'resize_param': {
        'prob': 1,
        'resize_mode': P.Resize.WARP,
        'height': resize_height,
        'width': resize_width,
        'interp_mode': [P.Resize.LINEAR],
    },
}

# If true, use batch norm for all newly added layers.
# Currently only the non batch norm version has been tested.
use_batchnorm = False

# Use different initial learning rate.
if use_batchnorm:
    base_lr = 0.0004
else:
    # A learning rate for batch_size = 1, num_gpus = 1.
    base_lr = 0.00004

# Modify the job name if you want.
job_name = "SSD_{}".format(resize)
# The name of the model. Modify it if you want.
model_name = "VGG_VOC0712_{}".format(job_name)

# Directory which stores the model .prototxt file.
save_dir = "models/VGGNet/VOC0712/{}".format(job_name)
# Directory which stores the snapshot of models.
snapshot_dir = "models/VGGNet/VOC0712/{}".format(job_name)
# Directory which stores the job script and log file.
job_dir = "jobs/VGGNet/VOC0712/{}".format(job_name)
# Directory which stores the detection results.
output_result_dir = "{}data/VOCdevkit/results/VOC2007/{}/Main".format(os.environ['HOME'], job_name)

```



目录



喜欢



收藏



评论



分享

```

# model definition files.
train_net_file = "{}/train.prototxt".format(save_dir)
test_net_file = "{}test.prototxt".format(save_dir)
deploy_net_file = "{}deploy.prototxt".format(save_dir)
solver_file = "{}solver.prototxt".format(save_dir)
# snapshot prefix.
snapshot_prefix = "{}{}/{}".format(snapshot_dir, model_name)
# job script path.
job_file = "{}{}/{}.sh".format(job_dir, model_name)

# Stores the test image names and sizes. Created by data/VOC0712/create_list.sh
name_size_file = "data/indoor/test_name_size.txt"

# The pretrained model. We use the Fully convolutional reduced (atrous) VGGNet.
pretrain_model = "models/VGGNet/VGG_ILSVRC_16_layers_fc_reduced.caffemodel"

# Stores LabelMapItem.
label_map_file = "data/indoor/labelmap_indoor.prototxt"

# MultiBoxLoss parameters.
num_classes = 2
share_location = True
background_label_id=0
train_on_diff_gt = True
normalization_mode = P.Loss.VALID
code_type = P.PriorBox.CENTER_SIZE
neg_pos_ratio = 3.
loc_weight = (neg_pos_ratio + 1.) / 4.
multibox_loss_param = {
    'loc_loss_type': P.MultiBoxLoss.SMOOTH_L1,
    'conf_loss_type': P.MultiBoxLoss.SOFTMAX,
    'loc_weight': loc_weight,
    'num_classes': num_classes,
    'share_location': share_location,
    'match_type': P.MultiBoxLoss.PER_PREDICTION,
    'overlap_threshold': 0.5,
    'use_prior_for_matching': True,
    'background_label_id': background_label_id,
    'use_difficult_gt': train_on_diff_gt,
    'do_neg_mining': True,
    'neg_pos_ratio': neg_pos_ratio,
    'neg_overlap': 0.5,
    'code_type': code_type,

```

```

    }
    loss_param = {
        'normalization': normalization_mode,
    }

    # parameters for generating priors.
    # minimum dimension of input image
    min_dim = 300
    # conv4_3 ==> 38 x 38
    # fc7 ==> 19 x 19
    # conv6_2 ==> 10 x 10
    # conv7_2 ==> 5 x 5
    # conv8_2 ==> 3 x 3
    # pool6 ==> 1 x 1
    mbox_source_layers = ['conv4_3', 'fc7', 'conv6_2', 'conv7_2', 'conv8_2', 'pool6']
    # in percent %
    min_ratio = 20
    max_ratio = 95
    step = int(math.floor((max_ratio - min_ratio) / (len(mbox_source_layers) - 2)))
    min_sizes = []
    max_sizes = []
    for ratio in xrange(min_ratio, max_ratio + 1, step):
        min_sizes.append(min_dim * ratio / 100.)
        max_sizes.append(min_dim * (ratio + step) / 100.)
    min_sizes = [min_dim * 10 / 100.] + min_sizes
    max_sizes = [[]] + max_sizes
    aspect_ratios = [[2], [2, 3], [2, 3], [2, 3], [2, 3], [2, 3]]
    # L2 normalize conv4_3.
    normalizations = [20, -1, -1, -1, -1, -1]
    # variance used to encode/decode prior bboxes.
    if code_type == P.PriorBox.CENTER_SIZE:
        prior_variance = [0.1, 0.1, 0.2, 0.2]
    else:
        prior_variance = [0.1]
    flip = True
    clip = True

    # Solver parameters.
    # Defining which GPUs to use.
    gpus = "0"
    gpulist = gpus.split(",")

```




目录



喜欢



收藏



评论



分享

```

num_gpus = len(gpulist)

# Divide the mini-batch to different GPUs.
batch_size = 4
accum_batch_size = 32
iter_size = accum_batch_size / batch_size
solver_mode = P.Solver.CPU
device_id = 0
batch_size_per_device = batch_size
if num_gpus > 0:
    batch_size_per_device = int(math.ceil(float(batch_size) / num_gpus))
    iter_size = int(math.ceil(float(accum_batch_size) / (batch_size_per_device * num_gpus)))
    solver_mode = P.Solver.GPU
    device_id = int(gpulist[0])

if normalization_mode == P.Loss.NONE:
    base_lr /= batch_size_per_device
elif normalization_mode == P.Loss.VALID:
    base_lr *= 25. / loc_weight
elif normalization_mode == P.Loss.FULL:
    # Roughly there are 2000 prior bboxes per image.
    # TODO(weiliu89): Estimate the exact # of priors.
    base_lr *= 2000.

# Which layers to freeze (no backward) during training.
freeze_layers = ['conv1_1', 'conv1_2', 'conv2_1', 'conv2_2']

# Evaluate on whole test set.
num_test_image = 800
test_batch_size = 1
test_iter = num_test_image / test_batch_size

solver_param = {
    # Train parameters
    'base_lr': base_lr,
    'weight_decay': 0.0005,
    'lr_policy': "step",
    'stepsize': 40000,
    'gamma': 0.1,
    'momentum': 0.9,
    'iter_size': iter_size,

```



目录



喜欢



收藏



评论



分享

```
'max_iter': 60000,
'snapshot': 40000,
'display': 10,
'average_loss': 10,
'type': "SGD",
'solver_mode': solver_mode,
'device_id': device_id,
'debug_info': False,
'snapshot_after_train': True,
# Test parameters
'test_iter': [test_iter],
'test_interval': 10000,
'eval_type': "detection",
'ap_version': "11point",
'test_initialization': False,
}

# parameters for generating detection output.
det_out_param = {
    'num_classes': num_classes,
    'share_location': share_location,
    'background_label_id': background_label_id,
    'nms_param': {'nms_threshold': 0.45, 'top_k': 400},
    'save_output_param': {
        'output_directory': output_result_dir,
        'output_name_prefix': "comp4_det_test_",
        'output_format': "VOC",
        'label_map_file': label_map_file,
        'name_size_file': name_size_file,
        'num_test_image': num_test_image,
    },
    'keep_top_k': 200,
    'confidence_threshold': 0.01,
    'code_type': code_type,
}

# parameters for evaluating detection results.
det_eval_param = {
    'num_classes': num_classes,
    'background_label_id': background_label_id,
    'overlap_threshold': 0.5,
```

```
'evaluate_difficult_gt': False,
'name_size_file': name_size_file,
}
```

```
### Hopefully you don't need to change the following ###
```

```
# Check file.
```

```
check_if_exist(train_data)
```

```
check_if_exist(test_data)
```

```
check_if_exist(label_map_file)
```

```
check_if_exist(pretrain_model)
```

```
make_if_not_exist(save_dir)
```

```
make_if_not_exist(job_dir)
```

```
make_if_not_exist(snapshot_dir)
```

```
# Create train net.
```

```
net = caffe.NetSpec()
```

```
net.data, net.label = CreateAnnotatedDataLayer(train_data, batch_size=batch_size_per_device,
        train=True, output_label=True, label_map_file=label_map_file,
        transform_param=train_transform_param, batch_sampler=batch_sampler)
```

```
VGGNetBody(net, from_layer='data', fully_conv=True, reduced=True, dilated=True,
        dropout=False, freeze_layers=freeze_layers)
```

```
AddExtraLayers(net, use_batchnorm)
```

```
mbox_layers = CreateMultiBoxHead(net, data_layer='data', from_layers=mbox_source_layers,
        use_batchnorm=use_batchnorm, min_sizes=min_sizes, max_sizes=max_sizes,
        aspect_ratios=aspect_ratios, normalizations=normalizations,
        num_classes=num_classes, share_location=share_location, flip=flip, clip=clip,
        prior_variance=prior_variance, kernel_size=3, pad=1)
```

```
# Create the MultiBoxLossLayer.
```

```
name = "mbox_loss"
```

```
mbox_layers.append(net.label)
```

```
net[name] = L.MultiBoxLoss(*mbox_layers, multibox_loss_param=multibox_loss_param,
        loss_param=loss_param, include=dict(phase=caffe_pb2.Phase.Value('TRAIN')),
        propagate_down=[True, True, False, False])
```

```
with open(train_net_file, 'w') as f:
```

```
    print('name: "{}_train"'.format(model_name), file=f)
```

```
    print(net.to_proto(), file=f)
```



目录



喜欢



收藏



评论



分享

```

shutil.copy(train_net_file, job_dir)

# Create test net.
net = caffe.NetSpec()
net.data, net.label = CreateAnnotatedDataLayer(test_data, batch_size=test_batch_size,
        train=False, output_label=True, label_map_file=label_map_file,
        transform_param=test_transform_param)

VGGNetBody(net, from_layer='data', fully_conv=True, reduced=True, dilated=True,
        dropout=False, freeze_layers=freeze_layers)

AddExtraLayers(net, use_batchnorm)

mbox_layers = CreateMultiBoxHead(net, data_layer='data', from_layers=mbox_source_layers,
        use_batchnorm=use_batchnorm, min_sizes=min_sizes, max_sizes=max_sizes,
        aspect_ratios=aspect_ratios, normalizations=normalizations,
        num_classes=num_classes, share_location=share_location, flip=flip, clip=clip,
        prior_variance=prior_variance, kernel_size=3, pad=1)

conf_name = "mbox_conf"
if multibox_loss_param["conf_loss_type"] == P.MultiBoxLoss.SOFTMAX:
    reshape_name = "{}_reshape".format(conf_name)
    net[reshape_name] = L.Reshape(net[conf_name], shape=dict(dim=[0, -1, num_classes]))
    softmax_name = "{}_softmax".format(conf_name)
    net[softmax_name] = L.Softmax(net[reshape_name], axis=2)
    flatten_name = "{}_flatten".format(conf_name)
    net[flatten_name] = L.Flatten(net[softmax_name], axis=1)
    mbox_layers[1] = net[flatten_name]
elif multibox_loss_param["conf_loss_type"] == P.MultiBoxLoss.LOGISTIC:
    sigmoid_name = "{}_sigmoid".format(conf_name)
    net[sigmoid_name] = L.Sigmoid(net[conf_name])
    mbox_layers[1] = net[sigmoid_name]

net.detection_out = L.DetectionOutput(*mbox_layers,
        detection_output_param=det_out_param,
        include=dict(phase=caffe_pb2.Phase.Value('TEST')))
net.detection_eval = L.DetectionEvaluate(net.detection_out, net.label,
        detection_evaluate_param=det_eval_param,
        include=dict(phase=caffe_pb2.Phase.Value('TEST')))

with open(test_net_file, 'w') as f:

```



目录



喜欢



收藏



评论



分享

```

        print('name: "{}_test".format(model_name), file=f)
        print(net.to_proto(), file=f)
    shutil.copy(test_net_file, job_dir)

# Create deploy net.
# Remove the first and last layer from test net.
deploy_net = net
with open(deploy_net_file, 'w') as f:
    net_param = deploy_net.to_proto()
    # Remove the first (AnnotatedData) and last (DetectionEvaluate) layer from test net.
    del net_param.layer[0]
    del net_param.layer[-1]
    net_param.name = '{}_deploy'.format(model_name)
    net_param.input.extend(['data'])
    net_param.input_shape.extend([
        caffe_pb2.BlobShape(dim=[1, 3, resize_height, resize_width]))])
    print(net_param, file=f)
shutil.copy(deploy_net_file, job_dir)

# Create solver.
solver = caffe_pb2.SolverParameter(
    train_net=train_net_file,
    test_net=[test_net_file],
    snapshot_prefix=snapshot_prefix,
    **solver_param)

with open(solver_file, 'w') as f:
    print(solver, file=f)
shutil.copy(solver_file, job_dir)

max_iter = 0
# Find most recent snapshot.
for file in os.listdir(snapshot_dir):
    if file.endswith(".solverstate"):
        basename = os.path.splitext(file)[0]
        iter = int(basename.split("{}_iter_".format(model_name))[1])
        if iter > max_iter:
            max_iter = iter

train_src_param = '--weights("{}" \\n'.format(pretrain_model)
if resume_training:

```



目录



喜欢



收藏



评论



分享



目录



喜欢



收藏



评论



分享

```

    if max_iter > 0:
        train_src_param = '--snapshot="{_iter_}.solverstate" \\n'.format(snapshot_prefix, max_iter)

if remove_old_models:
    # Remove any snapshots smaller than max_iter.
    for file in os.listdir(snapshot_dir):
        if file.endswith(".solverstate"):
            basename = os.path.splitext(file)[0]
            iter = int(basename.split("{}_iter_".format(model_name))[1])
            if max_iter > iter:
                os.remove("{}_iter_".format(snapshot_dir, file))
        if file.endswith(".caffemodel"):
            basename = os.path.splitext(file)[0]
            iter = int(basename.split("{}_iter_".format(model_name))[1])
            if max_iter > iter:
                os.remove("{}_iter_".format(snapshot_dir, file))

# Create job file.
with open(job_file, 'w') as f:
    f.write('cd {}\n'.format(caffe_root))
    f.write('./build/tools/caffe train \\n')
    f.write('--solver="{}" \\n'.format(solver_file))
    f.write(train_src_param)
    if solver_param['solver_mode'] == P.Solver.GPU:
        f.write('--gpu {} 2>&1 | tee {}/{}.log\n'.format(gpus, job_dir, model_name))
    else:
        f.write('2>&1 | tee {}/{}.log\n'.format(job_dir, model_name))

# Copy the python script to job_dir.
py_file = os.path.abspath(__file__)
shutil.copy(py_file, job_dir)

# Run the job.
os.chmod(job_file, stat.S_IRWXU)
if run_soon:
    subprocess.call(job_file, shell=True)

ssd_pascal_indoor.py

```

训练命令：

```
python examples/ssd/ssd_pascal_indoor.py
```

4 测试

SSD框架中提供了测试代码，有C++版本和python版本



目录

4.1 C++版本



喜欢

编译完SSD后，C++版本的的可执行文件存放目录：`.build_release/examples/ssd/ssd_detect.bin`



收藏

测试命令 `./build_release/examples/ssd/ssd_detect.bin models/VGGNet/indoor/deploy.prototxt models/VGGNet/indoor/VGG_VOC0712_SSD_300x300_iter_60000.caffemodel pictures.txt`

其中中保存的是待测试图片的list



评论

4.2 python版本



分享

python 版本的测试过程参见examples/detection.ipynb

参考：

1 将数据集做成VOC2007格式用于Faster-RCNN训练 (http://blog.csdn.net/sinat_30071459/article/details/50723212)

2 SSD的配置安装与测试 (<http://blog.csdn.net/samylee/article/details/51822832>)

分类: Computer Vision (<http://www.cnblogs.com/objectDetect/category/844155.html>), object Detection (<http://www.cnblogs.com/objectDetect/category/821690.html>)

好文要顶 () 关注我 () 收藏该文 () ()

(<http://home.cnblogs.com/u/objectDetect/>)

machineLearning (<http://home.cnblogs.com/u/objectDetect/>)

关注 - 14 (<http://home.cnblogs.com/u/objectDetect/followees>)

粉丝 - 5 (<http://home.cnblogs.com/u/objectDetect/followers>)

+加关注 ()



目录 « (<http://www.cnblogs.com/objectDetect/p/5682473.html>) 上一篇 : object detect links

(<http://www.cnblogs.com/objectDetect/p/5682473.html>)

» (<http://www.cnblogs.com/objectDetect/p/5849605.html>) 下一篇 : 神经网络向量化求反向传播梯度

喜欢 (<http://www.cnblogs.com/objectDetect/p/5849605.html>)

posted @ 2016-08-17 16:36 machineLearning (<http://www.cnblogs.com/objectDetect/>) 阅读(4639) 评论

收藏 (27) 编辑 (<https://i.cnblogs.com/EditPosts.aspx?postid=5780006>) 收藏



评论



 举报

分享标签 : [caffe](http://so.csdn.net/so/search/s.do?q=caffe&t=blog) (<http://so.csdn.net/so/search/s.do?q=caffe&t=blog>)



相关文章推荐

将数据集做成VOC2007格式用于Faster-RCNN训练 (/sinat_30071459/article/details/50723212)

0.文件夹名 首先，确定你的数据集所放的文件夹名字，例如我的叫logos。(因为后面做xml会用到这个文件夹名字) 1.图片命名 虽然说图片名对训练没什么影响，但建议还是按VOC2007那样，如“00...”



sinat_30071459 (http://blog.csdn.net/sinat_30071459) 2016-02-23 15:35 48007

Faster-RCNN+ZF用自己的数据集训练模型(Python版本)

目录(/sinat_30071459/article/details/51332084)



说明：本博文假设你已经做好了自已的数据集，该数据集格式和VOC2007相同。下面是训练前的一些修改。（做数据集的过喜欢程可以看这里） Faster-RCNN源码下载地址： Matlab版本：[http...](http://...)



sinat_30071459 (http://blog.csdn.net/sinat_30071459) 2016-05-06 17:33 53689

收藏



评论



分享



零基础的学习心路：12个机器学习案例实战！

都说今年是AI开发元年，为了转型AI技术程序员，这小半年来看了几本书，总结了一些学习的方法和踩过的坑儿，今天我想谈谈关于机器学习该如何入门以及学习方法....

(http://www.baidu.com/cb.php?c=lgF_pyfqHmsrHTYrjb0IZ0qnfK9ujYzP1D4P1Tk0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1Y1nymdrjnYuADLPvDkPAmL0AwY5HDdnjTsP1fdnHR0IgF_5y9YIZ0IQzq-uZR8mLPbUB48ugfEpZNGXy-jULNzTvRETVNzpyN1gvw-IA7GUatvPHqdIAdxTvqdThP-5yF_UvTkn0KzujYk0AFV5H00TZcqn0KdpYfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1civrHnz0AqLUWYs0ZK45HcsP6KWThnqn16kn1m)

SSD框架训练自己的数据集 (/yx2017/article/details/70344803)

SSD demo中详细介绍了如何在VOC数据集上使用SSD进行物体检测的训练和验证。本文介绍如何使用SSD实现对自己数据集的训练和验证过程，内容包括：1 数据集的标注2 数据集的转换3 使用SSD如何...



yx2017 (<http://blog.csdn.net/yx2017>) 2017-04-22 09:32 520

SSD框架训练自己的数据集 (/haima1998/article/details/77482788)

转自：http://www.cnblogs.com/objectDetect/p/5780006.html SSD demo中详细介绍了如何在VOC数据集上使用SSD进行物体检测的训练和验证。本...



haima1998 (http://blog.csdn.net/haima1998) 2017-08-22 17:12 128

SSD配置、训练、测试以及应用到自己的数据集 (/wei_guo_xd/article/details/73729472)

目录

git clone https://github.com/weiliu89/caffe.git (上面的版本可能存在问题,最好是在https://github.com/weiliu89/caffe/...



wei_guo_xd (http://blog.csdn.net/wei_guo_xd) 2017-06-25 20:54 786

喜欢



SSD框架训练自己的数据集 (/u014696921/article/details/53353896)

收藏



SSD demo中详细介绍了如何在VOC数据集上使用SSD进行物体检测的训练和验证。本文介绍如何使用SSD实现对自己数据集的训练和验证过程,内容包括:1 数据集的标注2 数据集的转换3 使用SSD如何...

评论



u014696921 (http://blog.csdn.net/u014696921) 2016-11-26 19:08 8018

分享

利用SSD和自己训练好的模型进行目标检测 (/wei_guo_xd/article/details/75376937)


本文翻译自: /caffe-ssd/examples/ssd_detect.ipynb 首先怎么安装jupyter以及使用jupyter 安装: sudo pip install jupyter 使...



wei_guo_xd (http://blog.csdn.net/wei_guo_xd) 2017-07-19 14:31 124


SSD: Single Shot MultiBox Detector 训练KITTI数据集 (1) (/jesse_mx/article/details/65634482)

前言 之前介绍了SSD的基本用法和检测单张图片的方法,那么本篇博客将详细记录如何使用SSD检测框架训练KITTI数据集。SSD项目中自带了用于训练PASCAL VOC数据集的脚本,基本不用做修改就可...

 Jesse_Mx (http://blog.csdn.net/Jesse_Mx) 2017-03-25 12:56 5691

SSD安装及训练自己的数据集 (/zhang_shuai12/article/details/52346878)

最近一直在搞object detection玩，之前用的是faster-rcnn，准确率方面73.2%，效果还不错，但是识别速度有点欠缺，我用的GPU是GTX980ti，识别速度大概是15fps.最...

 zhang_shuai12 (http://blog.csdn.net/zhang_shuai12) 2016-08-28 17:58 16572

目录

ssd训练kitti数据集和测试过程 (/flztiii/article/details/76850661)

喜欢

训练过程 训练过程可以参考http://blog.csdn.net/Jesse_Mx/article/details/65634482，这篇博客从kitti数据集的转化到训练文件的修改，训练过程，乃...

收藏

 flztiii (<http://blog.csdn.net/flztiii>) 2017-08-07 16:19 162

评论

用SSD训练自己的数据集 (/dongfang1984/article/details/74640219)


分享

1构建 数据集 先来看一下我们构建数据集应该是什么样的,假设总数据为1000张。为了方便，我们将数据放在 / home / bin golwang/data 文件夹下。/home/bingolwa...

 dongfang1984 (<http://blog.csdn.net/dongfang1984>) 2017-07-07 11:11 285

caffe-ssd训练kitti、lisa数据集 (/xiji321/article/details/70171511)

目的：将kitti、Lisa数据集合并，进行训练 一、数据集准备，将两种数据集准备成VOC格式 kitti数据集（车辆行人等）：<http://www.cvlibs.net/datasets...>

 xiji321 (<http://blog.csdn.net/xiji321>) 2017-04-14 15:42 838

SSD(Single Shot MultiBox Detector):ubuntu16安装及训练自己的数据集(VOC2007格式)过程记录 (/10km/article/details/70168526)

安装SSD# SSD代码clone到 caffe-ssd文件夹下 git clone --recursive https://github.com/weiliu89/caffe.git caffe-s...



10km (<http://blog.csdn.net/10km>) 2017-04-14 10:48 844



SSD: Single Shot MultiBox Detector 训练KITTI数据集 (2)

目录 (/jesse_mx/article/details/70048255)



前言 博主在上篇中花了很大篇幅讲解如何一步步把KITTI原始数据做成了SSD可以训练的格式，接下来就可以使用相关caffe代码实现SSD的训练了。下载VGG预训练模型 将 SSD 用于自己的检测任务，...



Jesse_Mx (http://blog.csdn.net/Jesse_Mx) 2017-04-11 10:52 3521

收藏



Ubuntu上用caffe的SSD方法训练umdfaces数据集 (/u013738531/article/details/61934587)

评论

实验目的 继前一段时间用SSD训练过VOC数据集以后，这一次使用SSD+K80服务器来训练自己的人脸识别应用，选择的数
据集还是之前下载的umdfaces，总共36w张人脸图像。 实验环境 ...



分享 u013738531 (<http://blog.csdn.net/u013738531>) 2017-03-13 22:42 1867

用SSD训练自己的数据集(VOC2007格式) (/zhy8623080/article/details/73188594)



用SSD训练自己的数据集(VOC2007格式)一. 配置caffe环境ubunt16.04下caffe环境安装二. 下载,编译及测试ssd源码(一)下载
源码github链接或者执行 git clone...



zhy8623080 (<http://blog.csdn.net/zhy8623080>) 2017-06-13 17:11 995

使用caffe框架利用faster-rcnn来训练自己的数据集 (/u013738531/article/details/53769643)

最近在研究caffe，前前后后差不多快一周了，论文看得比较少，直接上手来做的，期间遇到无数问题，大大小小的无数问题，不过通过上网，看别人的博客，几乎踩了很多大坑，这里给大家总结一下，希望后续同样做深度...

 u013738531 (<http://blog.csdn.net/u013738531>) 2016-12-20 21:40  4531



http://download.csdn.net/detail/howard_shooter/9971089
目录**人工智能训练数据集** (http://download.csdn.net/detail/howard_shooter/9971089)



喜欢 2017-09-09 12:23 5.67MB [下载](#)



收藏



评论



分享