Bayesian Deep Learning

We all have two lives. The second one starts when we realize that we only have one. --- Tom Hiddleston

[OpenCV 3.3] OpenCV 3.3 with DNN

OpenCV 3.3
Aug 3, 2017

OpenCV 3.3 has been released with greatly improved Deep Learning module and lots of optimizations.

Adrian Rosebrock: http://www.pyimagesearch.com/author/adrian/ [nice]

The following networks have been tested and known to work:

- AlexNet
- GoogLeNet v1 (also referred to as Inception-5h)
- ResNet-34/50/...
- SqueezeNet v1.1
- VGG-based FCN (semantical segmentation network)
- ENet (lightweight semantical segmentation network)

昵称：郝一二三
园龄：6年5个月
粉丝：53
关注：5
+加关注

< 2017年12月 >

日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

随笔分类(306)

~ARM(14)
~Kernel API(11)
Algo(10)
AR(10)
Bayes(43)
C(14)
C++(22)
CV(31)

- VGG-based SSD (object detection network)
- MobileNet-based SSD (light-weight object detection network)

下面是我们将用到的一些函数。

在dnn中从磁盘加载图片：

- cv2.dnn.blobFromImage
- cv2.dnn.blobFromImages

用“create”方法直接从各种框架中导出模型：

- cv2.dnn.createCaffeImporter
- cv2.dnn.createTensorFlowImporter
- cv2.dnn.createTorchImporter

使用“读取”方法从磁盘直接加载序列化模型：

- cv2.dnn.readNetFromCaffe
- cv2.dnn.readNetFromTensorFlow
- cv2.dnn.readNetFromTorch
- cv2.dnn.readhTorchBlob

从磁盘加载完模型之后，可以用.forward方法来向前传播我们的图像，获取分类结果。

看样子就是好东西，那么，一起来安装：Installing OpenCV 3.3.0 on Ubuntu 16.04 LTS

You may meet the trouble in conflicting with python in anaconda3. Solve it as following:

```
lolo@lolo-UX303UB$ mv /usr/bin/python3
python3            python3.4-config   python3.4m-config  python3m
python3.4          python3.4m         python3-config     python3m-config
```


MyLink

积分与排名

积分 - 70101
排名 - 4602

最新评论

1. Re:[ARCore] ARCore or ARToolkit 6 ?
你好，我最近也在研究ARToolkit，能不能留个联系方式，我想请教你几个问题啊，谢谢

--imaginehui

2. Re:[Bayesian] “我是bayesian我怕谁”系列 - Variational Inference
搜elbo找过来的，别的没看，elbo那几行讲的通俗易懂

--|浅お唱|ﾍ

3. Re:[Bayesian] “我是bayesian我怕谁”系列 - Variational Inference

```
lolo: Move them away.
```

```
cmake -D CMAKE_BUILD_TYPE=RELEASE       \
-D CMAKE_INSTALL_PREFIX=/usr/local/anaconda3 \
-D INSTALL_PYTHON_EXAMPLES=ON       \
-D INSTALL_C_EXAMPLES=OFF       \
-D OPENCV_EXTRA_MODULES_PATH=/home/unsw/Android/opencv-3.3.0/opencv_contrib-3.3.0/modules
\
-D PYTHON_EXECUTABLE=/usr/local/anaconda3/bin/python3.5       \
-D BUILD_EXAMPLES=ON ..
```

```
Python 2:
  Interpreter:                 NO

Python 3:
  Interpreter:                 /usr/local/anaconda3/envs/py35/bin/python3 (ver 3.5.2)
  Libraries:                   /usr/local/anaconda3/lib/libpython3.5m.so (ver 3.5.2)
  numpy:                       /usr/local/anaconda3/envs/py35/lib/python3.5/site-packages/num
  packages path:               lib/python3.5/site-packages

Python (for build):            /usr/local/anaconda3/envs/py35/bin/python3
```

Done :-)

Installing ref:

https://hackmd.io/s/S1gWq7BwW

http://www.linuxfromscratch.org/blfs/view/cvs/general/opencv.html

https://medium.com/@debugvn/installing-opencv-3-3-0-on-ubuntu-16-04-lts-7db376f93961

@调参狗小惊喜，本没想到有人会关注贝叶斯～辛苦不过，这个领域我确实还写不出书上以外的东西，目前的能力基本就是把学时的想法记录一下，整理一下知识体系；过去影响我最大的学习障碍常常是资料的有效性和学习的顺.....

--郝一二三

4. Re:[Bayesian] “我是bayesian我怕谁”系列 - Variational Inference

观看体验太难受了啊，而且很多东西都是树上的，感觉讲的不是很明白啊，但是关注你了，希望以后讲的更明白点

--调参狗

5. Re:嵌入式读书列表 - 参考

大三，努力学习当中，我肯定要把我认为的好书学完

--sty01

阅读排行榜

1. [Pycharm] Interpreter setting in Pycharm (7248)
2. Communication - 03.RILC(4658)
3. 内核回调 之 一"读"到底(4177)
4. 嵌入式读书列表 - 参考(3505)
5. 自旋锁，旋啊旋(2500)

评论排行榜

1. A Game idea...(18)
2. Linux读书片言(17)
3. 嵌入式读书列表 - 参考(12)
4. 挂个文件系统(7)
5. 当c遇到OOA，初探(6)

Now, you have got everything. Let's practice.

From: http://www.pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/

In the first part of today's post on object detection using deep learning we'll discuss ***Single Shot Detectors*** and ***MobileNets***.

SSD Paper: http://lib.csdn.net/article/deeplearning/53059

SSD Paper: https://arxiv.org/abs/1512.02325 [Origin]

When it comes to deep learning-based object detection there are three primary object detection methods that you'll likely encounter:

- Faster R-CNNs (Girshick et al., 2015) 7 FPS
- You Only Look Once (YOLO) (Redmon and Farhadi, 2015) 155 FPS.
- Single Shot Detectors (SSDs) (Liu et al., 2015)

If we combine both the MobileNet architecture and the Single Shot Detector (SSD) framework, we arrive at a fast, efficient deep learning-based method to object detection.

The model we'll be using in this blog post is a **Caffe version**of the original TensorFlow implementation by Howard et al. and was trained by chuanqi305 (see GitHub).

In this section we will use the **MobileNet SSD + deep neural network (dnn) module in OpenC**V to build our object detector.

Code analysis:

```
# USAGE
# python deep_learning_object_detection.py --image images/example_01.jpg \
#    --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.caffemodel

# import the necessary packages
import numpy as np
import argparse
```

```
import cv2


# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
    help="path to input image")
ap.add_argument("-p", "--prototxt", required=True,
    help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
    help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
    help="minimum probability to filter weak detections")
args = vars(ap.parse_args())


# initialize the list of class labels MobileNet SSD was trained to
# detect, then generate a set of bounding box colors for each class
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
    "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
    "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
    "sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

# load our serialized model from disk
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])

# load the input image and construct an input blob for the image
# by resizing to a fixed 300x300 pixels and then normalizing it
# (note: normalization is done via the authors of the MobileNet SSD
# implementation)
image = cv2.imread(args["image"])
(h, w) = image.shape[:2]
blob = cv2.dnn.blobFromImage(image, 0.007843, (300, 300), 127.5)    # --> NCHW

# pass the blob through the network and obtain the detections and
# predictions
print("[INFO] computing object detections...")
```

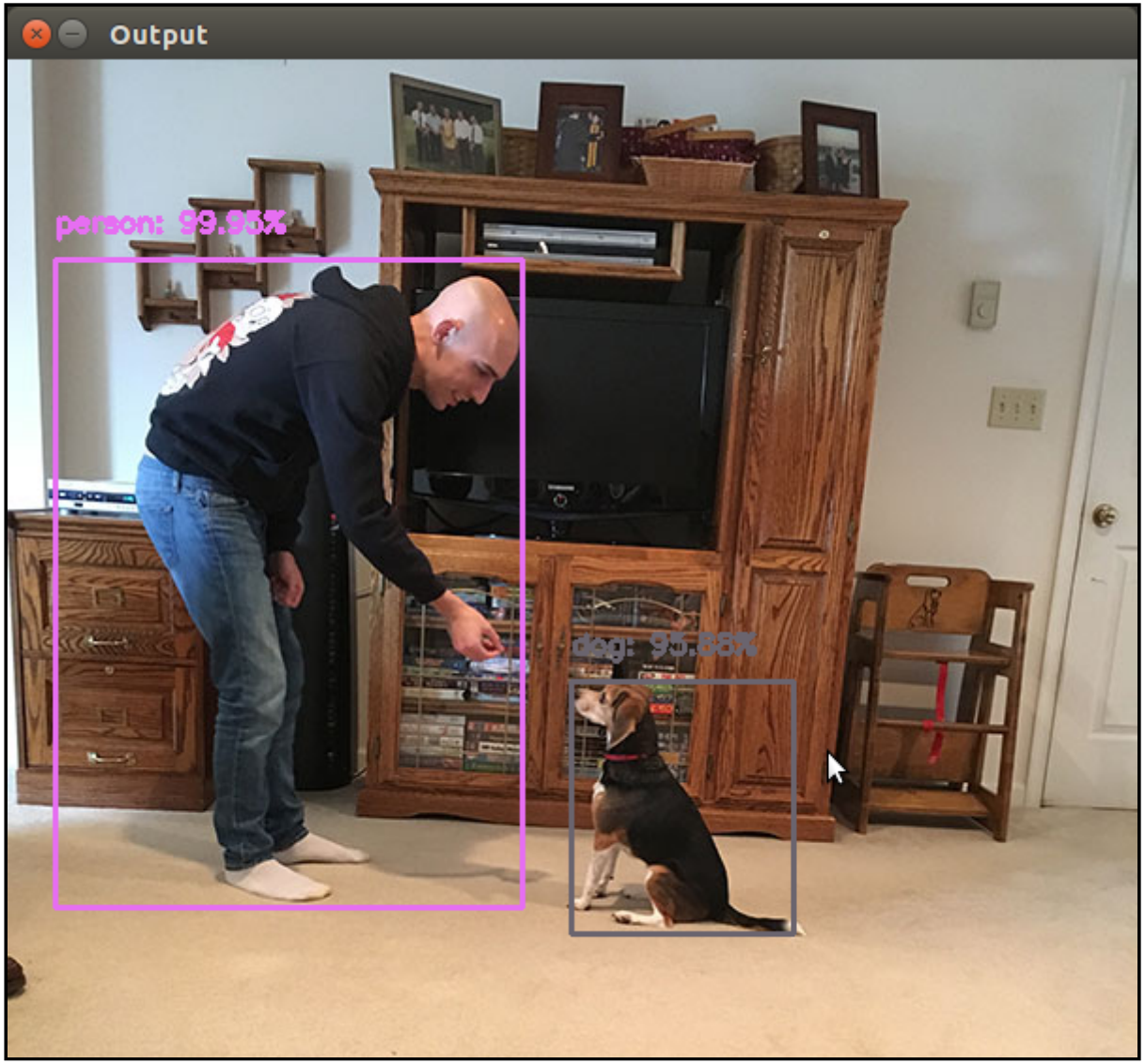
```
net.setInput(blob)
detections = net.forward()    # --> net.forward

# loop over the detections
for i in np.arange(0, detections.shape[2]):
    # extract the confidence (i.e., probability) associated with the
    # prediction
    confidence = detections[0, 0, i, 2]

    # filter out weak detections by ensuring the `confidence` is
    # greater than the minimum confidence
    if confidence > args["confidence"]:
        # extract the index of the class label from the `detections`,
        # then compute the (x, y)-coordinates of the bounding box for
        # the object
        idx = int(detections[0, 0, i, 1])
        box= detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # display the prediction
        label = "{}: {:.2f}%".format(CLASSES[idx], confidence * 100)
        print("[INFO] {}".format(label))
        cv2.rectangle(image, (startX, startY), (endX, endY), COLORS[idx], 2)
        y = startY - 15 if startY - 15 > 15 else startY + 15
        cv2.putText(image, label, (startX, y), cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)

# show the output image
cv2.imshow("Output", image)
cv2.waitKey(0)
```



NCHW

There is a comment that explains this, but in a different source file, ConvolutionalNodes.h, pasted below.

Note that the NVidia abbreviations refer to row-major layout, so to map them to column-major tensor indices are used by CNTK, you will need to reverse their order. E.g. cudnn stores images in “NCHW,” which is a [W x H x C x N] tensor in

CNTK notation (W being the fastest-changing dimension; and there are N objects of dimension [W x H x C] concatenated).

Note that the “legacy” (non-cuDNN) memory layout is old code written before NCHW became the standard, so we are likely phasing out the old representation eventually.

net.forward

```
[INFO] loading model...
[INFO] computing object detections...
(1, 1, 2, 7)

[[[[  0.          12.           0.95878285   0.49966827   0.6235761    0.69597626
0.87614471]
   [  0.          15.           0.99952459   0.04266162   0.20033446   0.45632178
0.84977102]]]]

[INFO] dog: 95.88%
[INFO] person: 99.95%
```

最新IT新闻:
· 迅雷“玩客币”更名“链克”，钱包功能14日正式实名制
· 微信支付买火车票送福利：最高优惠888元
· 微软Edge Android客户端下载量已超百万
· “游戏氪金”算赌博吗？最近各国陆续做了一些动作
· 百度AI战略背后雄心：欲重获互联网霸主地位
» 更多新闻...

最新知识库文章:
· 以操作系统的角度述说线程与进程
· 软件测试转型之路
· 门内门外看招聘
· 大道至简，职场上做人做事做管理
· 关于编程，你的练习是不是有效的？
» 更多知识库文章...