16824- Visual Learning and Recognition assignment2 Report

Siwei Zhu
ID: szhu1

Part I: Quesions:

Q1: Explain the functionality of `SecretAssignmentLayer`

Answer: The bottom for this layer is cls_prob, which is a R * 20 (R denote the number of ROI)matrix indicate the element-wise product of classification matrix (which is also R * 20) and detection matrix (R * 20). And this `SecretAssignmentLayer` will reshape the R * 20 matrix to be 1 * 20 * R (top), and this matrix will then go through the "Reduction Layer" along axis = 2 which will deduct the dimension of the matrix by 1 along the last axis. So we could eventually get a 1 * 20 matrix which we will use for computing loss.

Q2: There are a few similar layers implemented in Caffe. Name them and explain why they cannot be used in our case?

Answer: There are two layers in caffe which are Flatten layer and Reshape layer.Flatten layer will change the input of shape n * c * h * w to a simple vector output of shape n * (c*h*w). For Reshape layer, it is a more general version of Flatten layer (When reshape_param { shape { dim: 0 dim: -1 } } it is the same as Flatten layer). The reason why they can not be used here is that before we go through this layer let's say we have a R* 20 matrix, and we want to have a 1 * 20 * R matrix, what Reshape layer does is it will fulfill each row of the new matrix along the rows of the original matrix, so if R is not equal to 20 (here we suppose R = 100), the rows of the new matrix will be filled with 100/20 = 5 ROIs with different label of class. This is not what we want.

Q3: If you have attempted the first part, you probably have realized that defining the loss function in the network definition prototxt is not as straightforward as adding one more layer. In the assignment submission PDF, explain why?

Answer: The main reason is that for different neural network, the loss maybe different, and most of the time we may can not use a single caffe layer to represent the loss.
For example, in our assignment, the loss function is defined in the paper and we need to implement it ourself.

Q4: There are ways to avoid implementing a Python based loss layer i.e. the loss can be expressed as a combination of existing Caffe layers. In the assignment submission PDF, describe one such combination - try to submit the most concise combination.

Answer: One combination is: Log Layer —> Eltwise Layer(SUM)
Or maybe we could use Log Layer —> Reduction Layer.

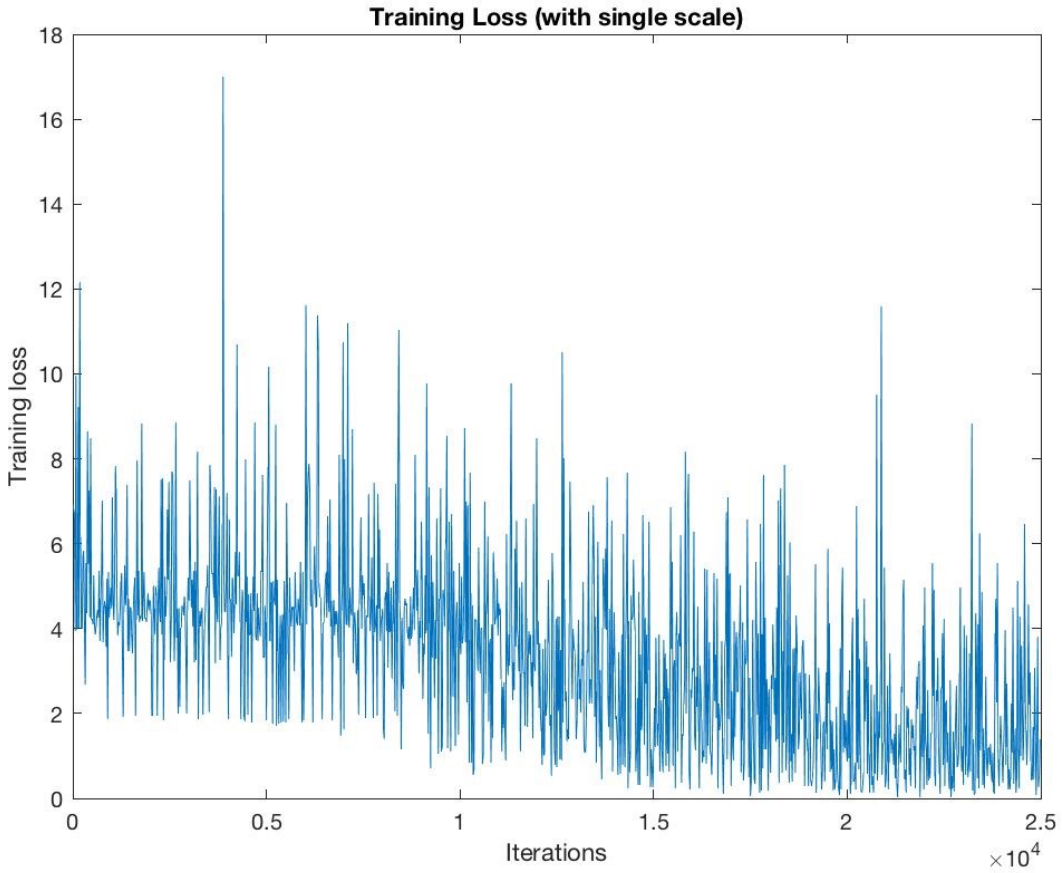
Q5: In the assignment submission PDF, explain how multi-scale training works and why it is useful?

Answer: Multi-scale training will randomly initialize the input image with different scales then send the image into training. This is a way of data augmentation in deep learning, which could effectively increase our dataset as well as avoid overfitting.

Part II: Experiment results.

1. Training with single scale:

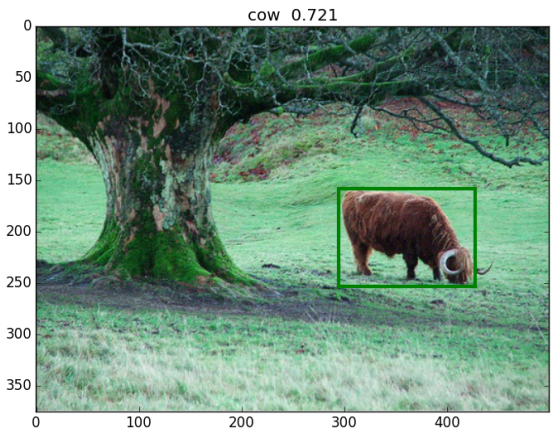
1.1 the training loss plot:

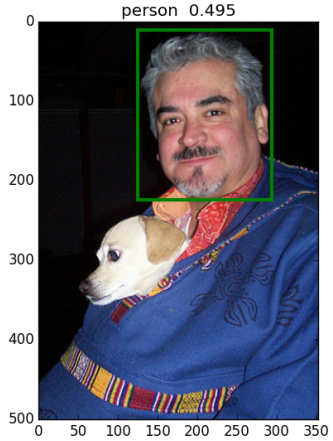


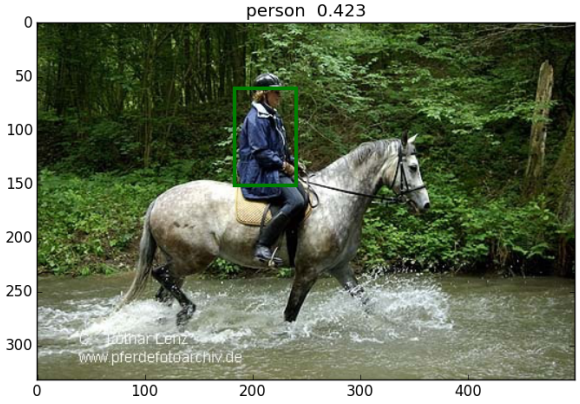
1.2 Class-wise AP and mAP:

AP for aeroplane = 0.2190
AP for bicycle = 0.3167
AP for bird = 0.1801
AP for boat = 0.1347
AP for bottle = 0.0173
AP for bus = 0.4867
AP for car = 0.4329
AP for cat = 0.2421
AP for chair = 0.0107
AP for cow = 0.1673
AP for diningtable = 0.0992
AP for dog = 0.1838
AP for horse = 0.3684
AP for motorbike = 0.3938
AP for person = 0.0528
AP for pottedplant = 0.0757
AP for sheep = 0.1434
AP for sofa = 0.2199
AP for train = 0.3896
AP for tvmonitor = 0.2187
Mean AP = 0.2176

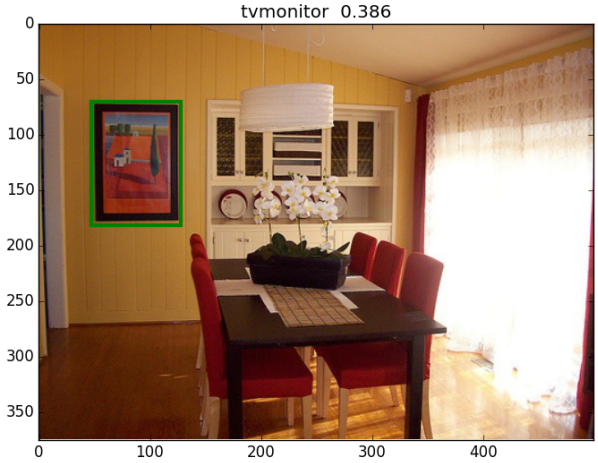
1.3 Visualization of predicted bounding boxes

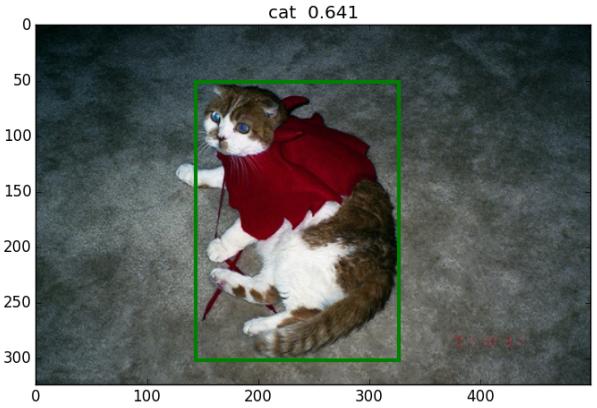


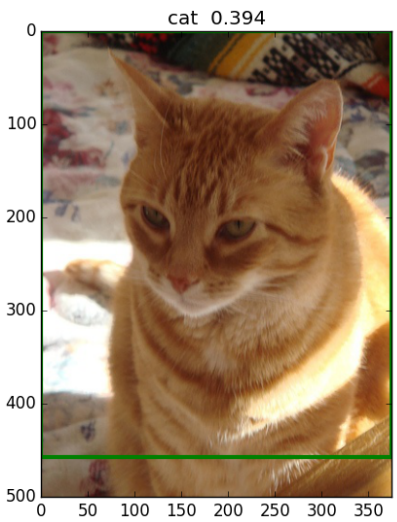


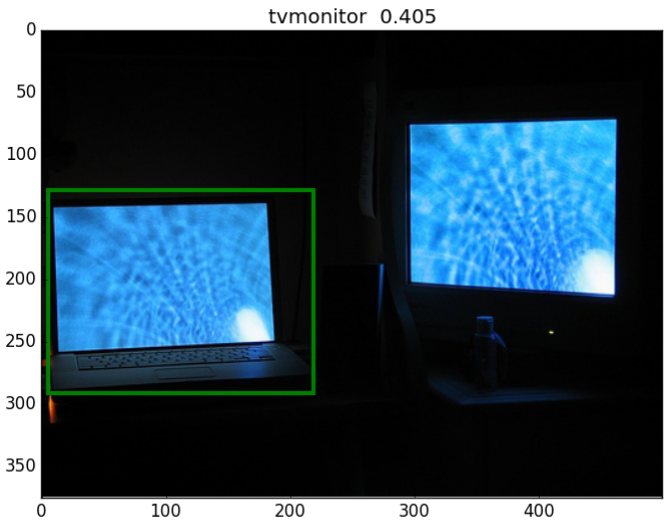

person 0.423
© Dothar Lenz
www.pferdefotoarchiv.de

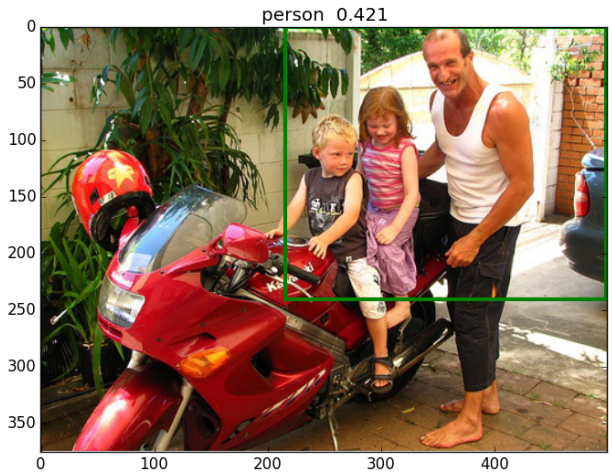

car 0.968

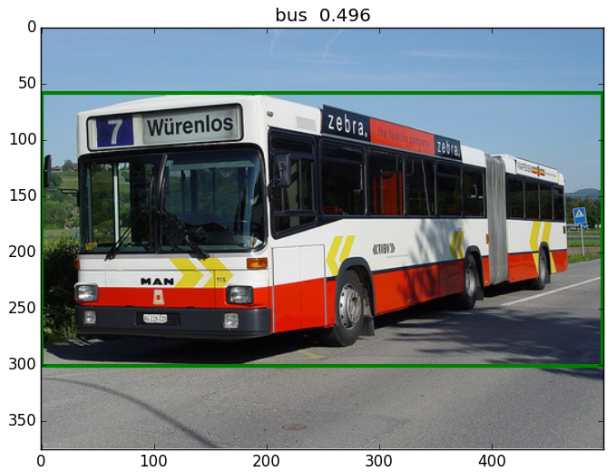

tvmonitor 0.386


cat 0.641


cat 0.394

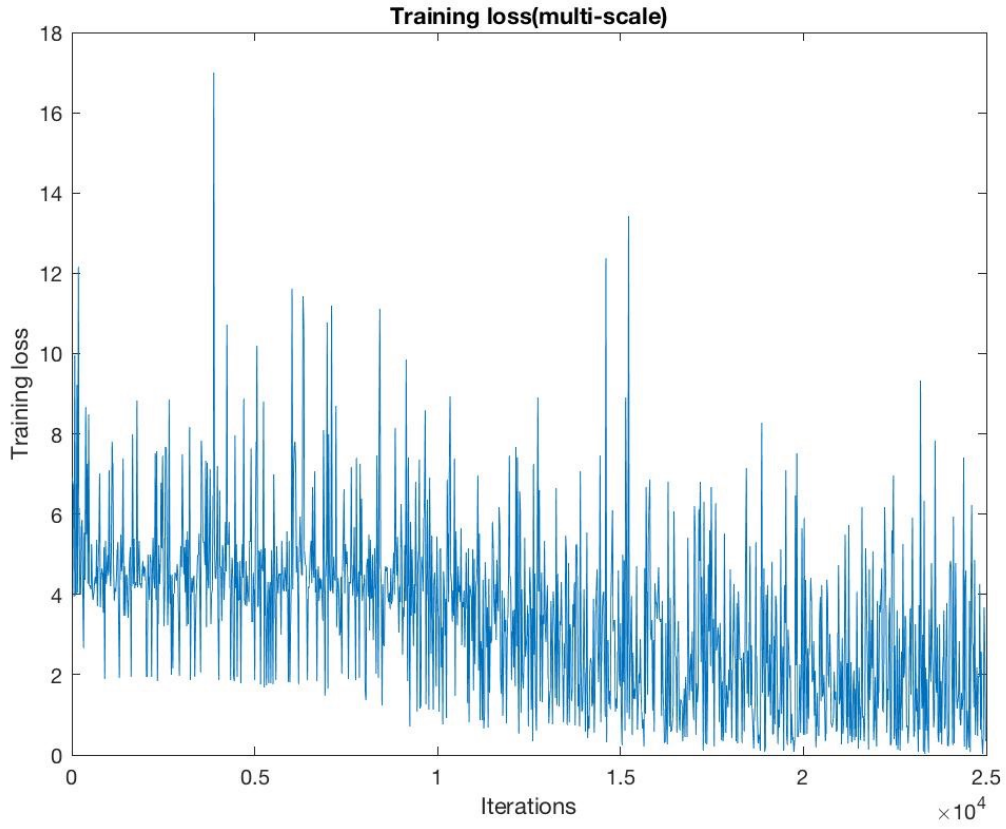

tvmonitor 0.405





2. Training with multi-scale:

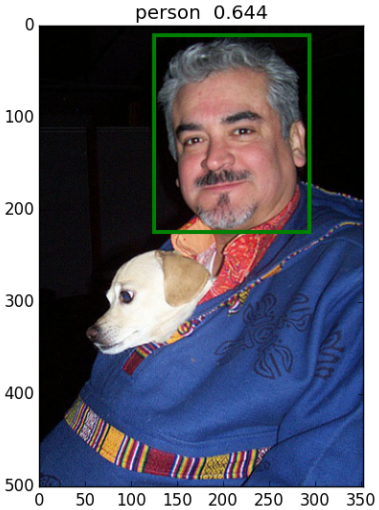
2.1 Training loss plot:

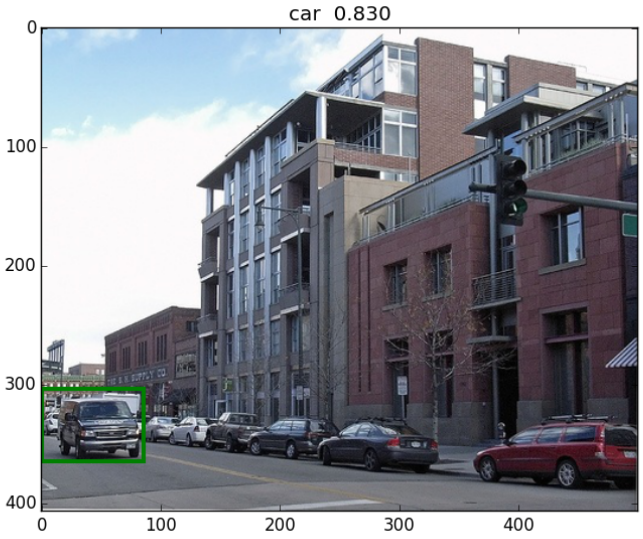


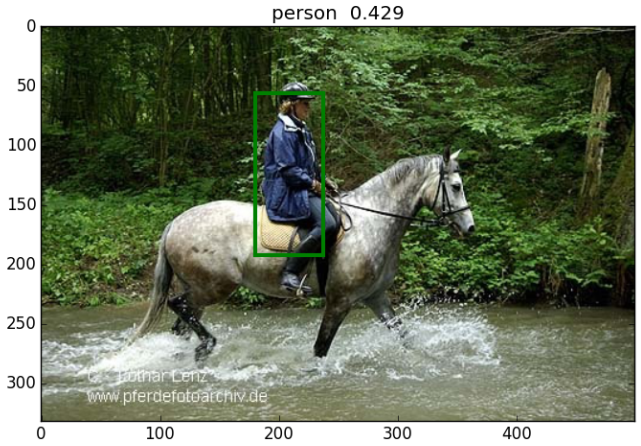
2.2 Class-wise AP and mAP:

AP for aeroplane = 0.2672
AP for bicycle = 0.2891
AP for bird = 0.1857
AP for boat = 0.1985
AP for bottle = 0.0372
AP for bus = 0.4667
AP for car = 0.4385
AP for cat = 0.2928
AP for chair = 0.0159
AP for cow = 0.1990
AP for diningtable = 0.0867
AP for dog = 0.1781
AP for horse = 0.3025
AP for motorbike = 0.3606
AP for person = 0.0669
AP for pottedplant = 0.0874
AP for sheep = 0.1457
AP for sofa = 0.2409
AP for train = 0.3701
AP for tvmonitor = 0.2790
Mean AP = 0.2254

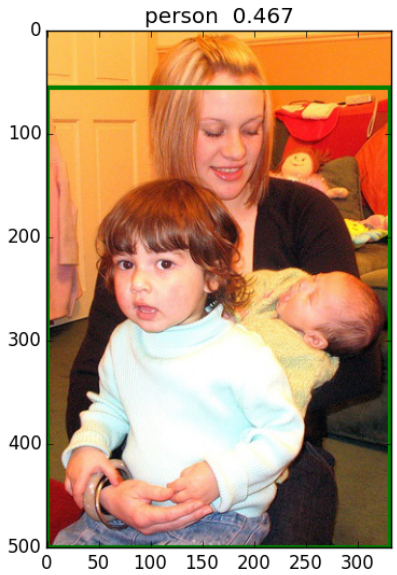
2.3 Visualization of predicted bounding boxes:

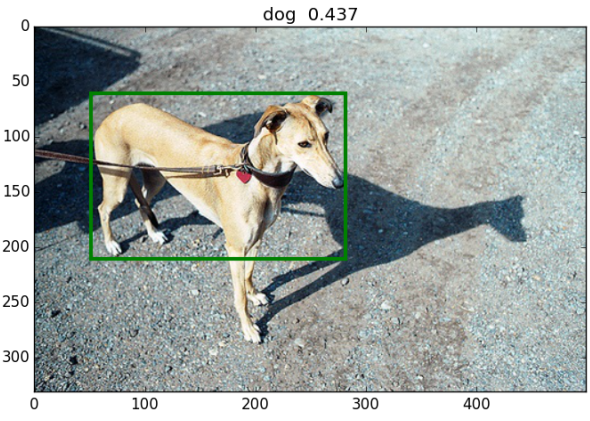


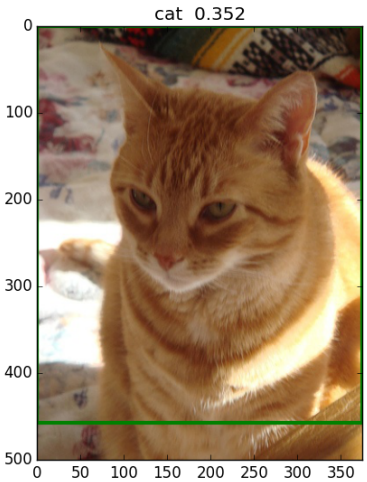

car 0.830

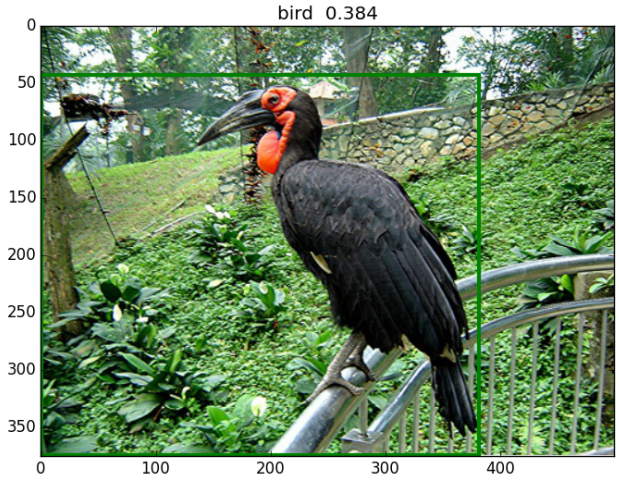

person 0.429
© Gothar Lenz
www.pferdefotoarchiv.de

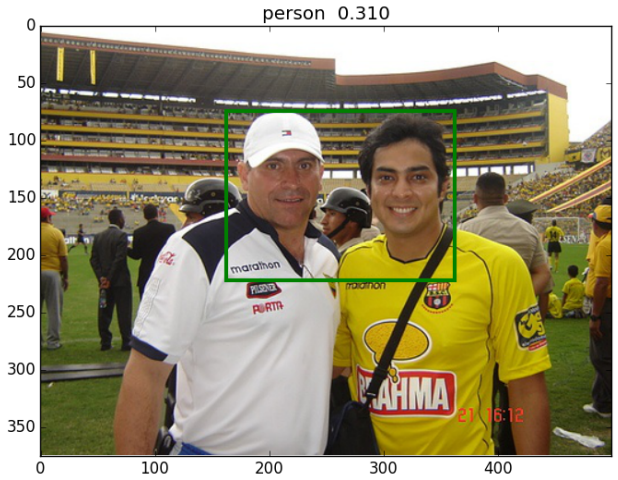

person 0.300


person 0.467


dog 0.437


cat 0.352


bird 0.384
0
50
100
150
200
250
300
350
0
100
200
300
400


person 0.310
0
50
100
150
200
250
300
350
0
100
200
300
400
marathon