CIS 510 Computer Vision - Project Proposal: Faking COVID-19: Fooling diagnosis with Normalizing flows

Guzman, Luis and Walton, Steven

April 2020

1 Part1

The work for part 1 contained within the code 'part1.py'. Note: Updates are in section 4.

1.1 Analysis

It appears that my network did not retrain well. If I had more time I would adjust the network more so that we get better classification. I had trained several times, dropping the learning rate as I went, and results were improving. I suspect that dropping the learning rate even more would improve the results even more. I had been fighting with COLAB for a few days where my simulation was constantly hanging and running out of resources when processing a single image and decided to move to another machine. This allowed me to improve results, but made me already late on the assignment, so I cut my losses.

Previous training gave a mAP score of 0, so they were not included. Still, the score is still quite small here with a mAP of 0.0005. This looks like it is not only due to miss classification, but the bounding boxes. In Figure 2 we can see that we correctly identified boat in the two locations but that the bounding boxes are not great. In this image no bounding box had an IoU ≥ 0.5 and thus the precision on boats was 0 here. In Figure 3 we can see that there is a lot of misclassifications here. Better retraining of the network would show an improvement in this score. We can conclude that the low mAP score is due to both of these effects.

```
Name    Recall  AP
cat     0.014013097161320104    0.011076332794830373
dog     0.0     0.0
diningtable     0.0     0.0
chair   0.0     0.0
person  0.0     0.0
pottedplant     0.0     0.0
sofa    0.0     0.0
tvmonitor       0.0     0.0
car     0.0     0.0
aeroplane       0.0     0.0
horse   0.0     0.0
bicycle 0.0     0.0
bird    0.0     0.0
train   0.0     0.0
bus     0.0     0.0
bottle  0.0     0.0
motorbike       0.0     0.0
cow     0.0     0.0
sheep   0.0     0.0
boat    0.0     0.0
mAP                     0.0005538166397415186
```

Figure 1: Precision scores for classes after retraining network

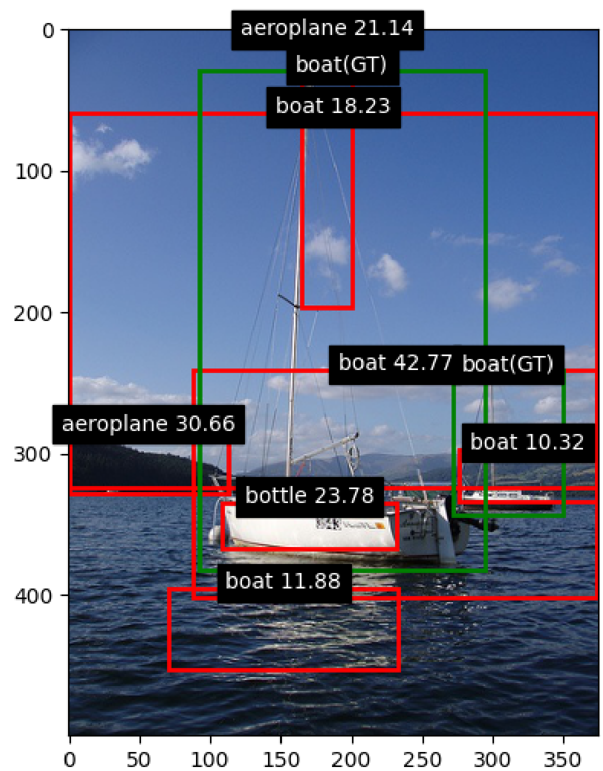


Figure 2: Sample plot with ground truth labels in green

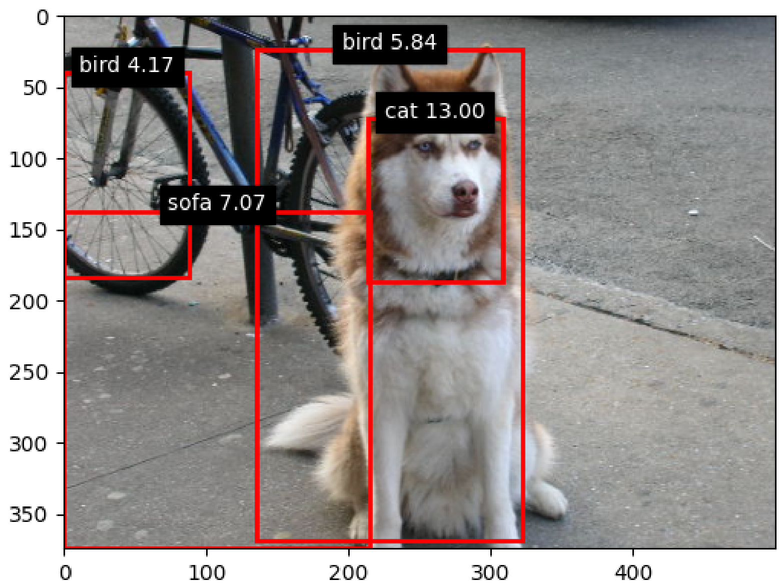


Figure 3: Sample plot without ground truth labels

2 Part2

The code for part 2 comes from modifying mmdetection and using their model zoo. I had to preprocess the VOC dataset using their included tools. The train tool allows one to select different models where I used SSD300 and Faster RNN R50 FPN 1x. The results can be seen in Figure 4 and Figure 5, respectively. Each performed much better than the results in part 1. As can be seen from the results SSD300 performed slightly better than Faster RCNN.

class	gts	dets	recall	ap
aeroplane	285	8959	0.877	0.599
bicycle	337	7240	0.911	0.694
bird	459	34008	0.856	0.512
boat	263	40408	0.852	0.308
bottle	469	40400	0.727	0.214
bus	213	8363	0.925	0.586
car	1201	47938	0.929	0.748
cat	358	8244	0.911	0.747
chair	756	89743	0.892	0.383
cow	244	4216	0.910	0.531
diningtable	206	7653	0.893	0.500
dog	489	7835	0.937	0.641
horse	348	3330	0.917	0.736
motorbike	325	4882	0.911	0.690
person	4528	217140	0.911	0.644
pottedplant	480	40962	0.738	0.284
sheep	242	5752	0.826	0.472
sofa	239	4253	0.883	0.554
train	282	17710	0.933	0.685
tvmonitor	308	30946	0.893	0.567
mAP				0.555

Figure 4: mAP results for SSD300

class	gts	dets	recall	ap
aeroplane	285	4018	0.895	0.657
bicycle	337	2154	0.884	0.639
bird	459	5485	0.808	0.494
boat	263	1506	0.654	0.377
bottle	469	1557	0.486	0.299
bus	213	2229	0.911	0.525
car	1201	9155	0.936	0.769
cat	358	2484	0.838	0.460
chair	756	5312	0.671	0.378
cow	244	4673	0.930	0.453
diningtable	206	1636	0.762	0.421
dog	489	4459	0.906	0.468
horse	348	5121	0.914	0.627
motorbike	325	2686	0.849	0.602
person	4528	39696	0.941	0.714
pottedplant	480	3504	0.663	0.333
sheep	242	1388	0.731	0.457
sofa	239	3194	0.866	0.506
train	282	4382	0.901	0.634
tvmonitor	308	1514	0.831	0.623
mAP				0.522

Figure 5: mAP results for Faster RCNN

3 Extra Credit

To perform part 1 I had to retrain the network so that it could properly use the VOC labels. This code is included in the part1.py file under the function "retrain".

4 UPDATES

Given the extra time I updated some of the parameters in the model and was able to get a better mAP score. I changed the transform function and played around with variables in the selective search. Both, as predicted originally, helped give a better score.

4.1 Selective Search

I did not measure directly the mAP scores for changes in the parameters. What I did instead is selected a few images and investigated how the change in parameters affected how accurately, under visual inspection, the search got the objects we were looking for. The problem is to get the selective search to pick up the large objects and not too small of objects. The biggest problem I was having is that the search would select a bunch of small images and ignore the large images that are closer to the ground truth. Maximal suppression helped some, but there were a lot of squares with different classifications that made this difficult.

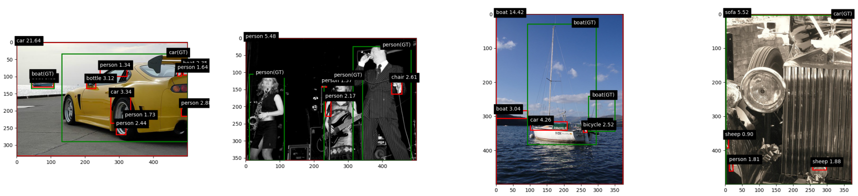
4.2 Transfer function

The transfer function the difference was that I used random resized crop and a random horizontal flip. Previously I had just resized. The normalization function is the same. Table 1 shows how different VGGs give different mAP scores. Each of these trained for 20 epochs with a learning rate of 10^{-4}.

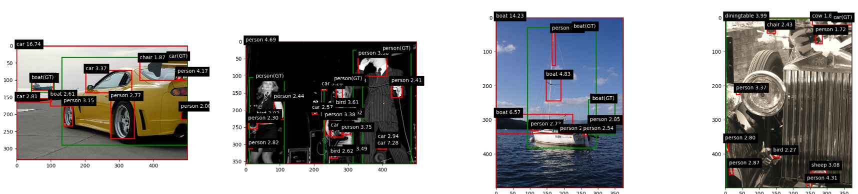
It can be seen that because we are better able to classify has a large change on the mAP score. Figure 6 shows examples of how the different models affects the accuracy. Most notably is the last image where only VGG19 correctly classifies the closeup image as a car.

Model	mAP
VGG11	0.0107
VGG16	0.0112
VGG19	0.0114

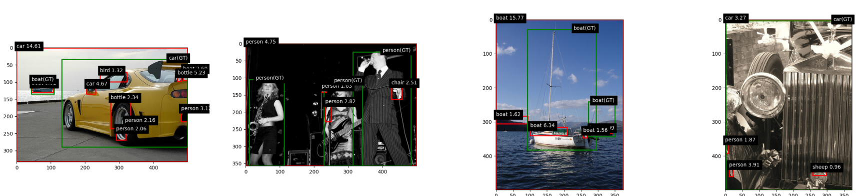
Table 1: mAP scores for different VGG models



(a) VGG11



(b) VGG16



(c) VGG19

Figure 6: Difference in VGG models