Capstone Proposal: Multiple object image detection

Trey Bean*
treybean@gmail.com

1 Domain Background

Computer vision is a field of study focusing on the process in which a computer derives information from the contents of images or video. Starting in the 1960's, researchers sought to design computer programs that could deduce information from images beyond their pixel representation, for example trying to derive 3D geometrical information from 2D perspective images, [8]. Over the subsequent decades, computer scientists continued to advance the techniques to extract more and higher quality information from images. Organizations began holding annual competitions to allow researchers to benchmark and compare the performance of the latest algorithms and challenge research groups to make advances in the field. For example, the PASCAL Visual Object Classes (VOC) Challenge, which ran from 2005-2012, [5], and the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which started in 2010 and continues to be held each year. Over the course of the last 6 years, the ILSVRC has seen a 7.9x reduction in image classification error (from 28.2% to 3.57%), surpassing the human error rate on the same dataset of 5.1%, while we've seen object detection accuracy increase 2.9x since the introduction of this task, from 22.6% mean average precision (mAP) in ILSVRC2013 to 66.3% mAP in ILSVRC2016, [12] and [9].

In the project at the end of the deep learning section of the Udacity Machine Learning nanodegree, we used a convolutional neural network to classify images of the CIFAR-10 dataset into one of the ten categories. While impressive, the requirements for many computer visualization applications, e.g. self-driving cars, warehouse robots, augmented reality applications, etc. require much more sophisticated information retrieval from image-based sensory inputs, like the object detection category of the ILSVRC. When I look up from my computer, for example, the image that is presented to my brain can't be reduced to a single class, maybe "cluttered living room scene", without losing a great deal of information. In terms of object detection, my brain is quickly able to detect, classify, and locate hundreds, if not thousands, of objects, all of which can be used by my brain to make evaluations and decisions. If we're going to achieve near-human or super-human performance in many artificial intelligence applications, it's very likely these applications will need to leverage, if not surpass, this level of complexity of input.

Personally, from the first time I was introduced to a relational database and how it can organize information for storage and retrieval, not to mention exploring the data for additional insights, I've been excited about and fascinated with the challenge of modeling human intelligence and the capacity for learning with computers. In fact, it was at this point in college, when I shifted my focus from neuroscience to computer science. We've got a long way to go to approach the full complexity and potential contained within the human brain, but, since humans are largely a visual species, it makes sense to me that great gains in the field of artificial intelligence will come from recreating that key ability of ours in computers.

2 Problem Statement

Building off the final project in the deep learning section of the Udacity Machine Learning Nanodegree, where I trained a convolutional neural network to classify images from the CIFAR-10 dataset into a single classes, in this capstone project, I will explore how to use similar deep learning

*Udacity profile: https://profiles.udacity.com/p/7636093896

techniques to detect multiple objects of different classes as well as multiple instances of the same class in a single image. Given a set of images containing multiple objects of different classes or different instances of the same class, the algorithm should correctly identify and classify at least n different objects, where n depends on the dataset.

3 Datasets and Inputs

For this project, I'm planning on using the PASCAL VOC 2007 dataset [1] because it is the one I see cited the most most frequently in related papers, e.g. [6, 10]. This will provide me existing benchmarks on the same dataset by leading researchers in this topic area and allow me to measure my own attempts at solving the problem.

There are other datasets available, e.g. the PASCAL VOC 2012 dataset [2], the ImageNet ILSVRC datasets, as well as the MS COCO dataset. I'm electing not to use the ILSVRC2017 dataset because it is quite large it isn't publicly available without registration and permission. I have registered and downloaded the dataset and may explore performance on it as well as including it in the training on my own, but won't use it as part of the capstone project for the reasons cited above. I will also experiment with the MS COCO dataset on my own, but won't use it for the capstone project, primarily due to the newness of the dataset and, therefore, fewer benchmarks and paper citations as well as the larger size, which will impact Udacity's reviewers. I will definitely explore using the PASCAL VOC 2012 dataset as well as a combined PASCAL VOC 2007 & 2012 dataset as described by Ren et al. [10].

The PASCAL VOC 2007 dataset [1] is comprised of 5,011, 3-layer, color images, constrained to preserve aspect ratio with the maximum width or height to be 500px. The images have been annotated to identify bounding boxes and a corresponding object class label for each object in one of the twenty classes present in the image. In total, 12,608 object instances are identified in the 5,011 images. The classes of objects identified are: Aeroplane, Bicycle, Bird, Boat, Bottle, Bus, Car, Cat, Chair, Cow, Diningtable, Dog, Horse, Motorbike, Person, Pottedplant, Sheep, Sofa, Train, and Tvmonitor. The annotations for each object identified include the class for each object identified along with a bounding box, the pose, whether it is truncated or occluded, and a 'difficult' flag indicating that the object is considered difficult to recognize, "for example an object which is clearly visible but unidentifiable without substantial use of context" [4].

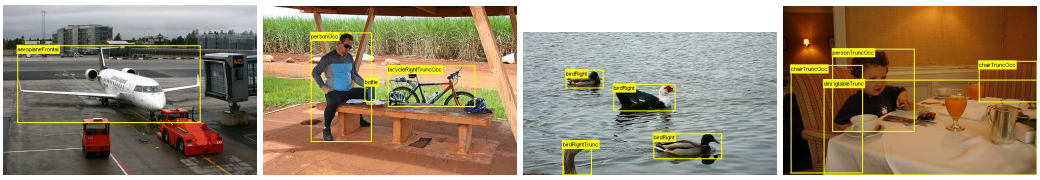

Figure 1: Examples of annotated images in the PASCAL VOC 2012 dataset.

While the class distribution isn't equal within the training or the validation sets, as seen in Figure 2, the distributions are approximately equal across training and validation sets [3]. I expect to start by using the existing split, but if I experiment with different splitting strategies, I will make sure I maintain class distributions between train and validation sets.

4 Solution Statement

As an initial approach, after preprocessing the images by mean subtraction and normalization, one could leverage traditional computer vision techniques like edge detection to isolate potential objects in a single image and then pass those to a CNN similar to the one I created to classify the CIFAR-10 images, basically breaking the problem down to smaller sizes and then do what we know how to do, classify a single image with a single class. From my research, this is cumbersome, slow, not very accurate, and doesn't leverage any contextual clues from other objects in the image. Other approaches that have been explored are to utilize a sliding-window detector to pass small frames of the underlying image to a CNN for classification. The model can then use regression techniques to modify the predicted bounding box. The challenge with this is, again, performance, and it's somewhat

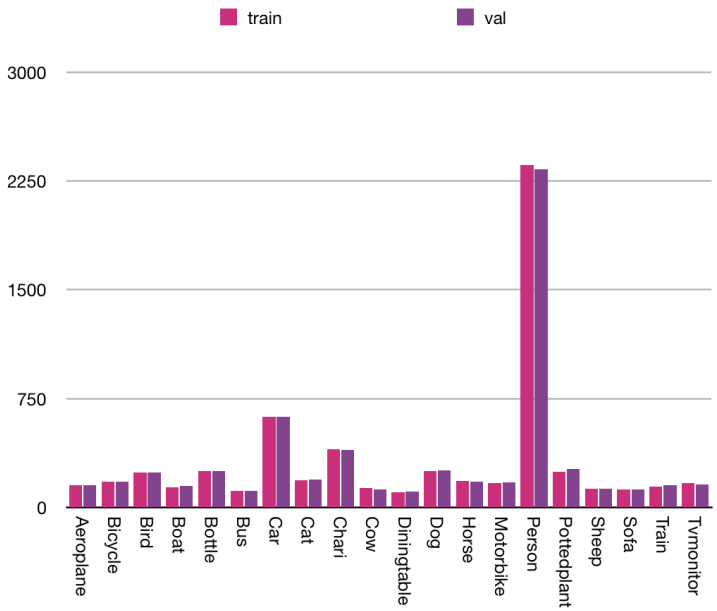

Figure 2: Class distribution across PASCAL VOC 2007 training and test datasets.

disappointing in its inelegance. Others suggested using different techniques to propose likely regions containing objects [7]. These made great strides in accuracy, but still were lacking in performance.

The current state-of-the art technique, Faster R-CNN, replaces the original region proposal steps in the above approaches and elegantly uses a CNN for proposing regions [10] that shares layers with the detection network. I intend to implement this method on the PASCAL VOC 2012 dataset and make observations about how it works and explore alternate approaches to increase either accuracy or performance.

5 Benchmark Model

As described above, I will be using the results of the Faster R-CNN model described in the paper by Girshick [6] as my benchmark model. The authors were able to achieve 69.9% mean Average Precision (mAP) on just the PASCAL VOC 2007 dataset, which is 4.5% better than the selective search (SS) approach used by Fast R-CNN [6] on the same dataset. The authors were able to achieve a significantly better mAP by training on a combined dataset that incorporated PASCAL VOC 2007, PASCAL VOC 2012, and MS COCO. I will not be doing that as part of this project for dataset size guidelines, though, I will explore it on my own.

These results are trained using state-of-the-art networks and computational hardware. While I intend to aim to meet these, in order to arrive at a more reasonable benchmark model, I downloaded the Faster R-CNN model that was implemented in Caffe [11]. With this implementation, trained and validated against the PASCAL VOC 2007 dataset, I was able to achieve **60.9% mAP**. This will be my benchmark metric.

6 Evaluation Metrics

Most papers and competitions in this space are using mean Average Precision (mAP) as the primary evaluation metric. This is calculated by creating a precision/recall curve and calculating the area underneath the curve, giving you the average precision for a given class. The mAP is the mean across all of the 20 classes in the dataset.

The determination on whether a predicted classification and bounding box is accurate, a, is calculated by analyzing if the area, A, of the intersection between the predicted bounding box, B_p, and the

ground truth bounding box, B_{gt}, divided by the area of the union of the two bounding boxes is greater than a given threshold, t. For the PASCAL VOC 2012 challenge $t = 0.5$.

$$a = \frac{A(B_p \cap B_{gt})}{A(B_p \cup B_{gt})} > t$$

Because performance of identification of the trained model is important for real-world applications, an additional metric I intend to evaluate is the frames per second (fps), which I'm defining as the number of images that the trained model can process per second. The benchmark model of Fast R-CNN was able to achieve 0.5 fps and the Faster R-CNN was able to achieve 5 fps when using the Simonyan and Zisserman model (VGG-16) for the Region Proposal Network (RPN) and 17 fps when using the Zeiler and Fergus model (ZF) for the RPN, [10].

7 Project Design

As described above, the key insight of the Faster R-CNN model is to replace the region proposal portion of its predecessor Fast R-CNN, which used a selective search algorithm and produced about 2000 region proposals to feed to the detector network, with a Region Proposal Network (RPN), an additional network that shares a common set of convolutional layers with the detector network. This approach generates multiple orders of magnitudes fewer, and better, region proposals to be processed by the detector network resulting in much *faster* R-CNN, [10], see Figure 3.

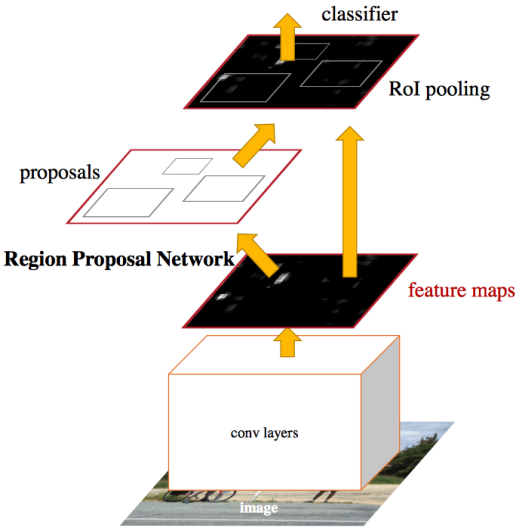

Figure 3: Overview of Faster R-CNN workflow.

In order to share the convolutional layers, I will start by preprocessing the images by applying a mean subtraction and normalization step. I will also re-scale the images such that the shorter side is 600px, as is done in the Faster R-CNN paper, [10]. Next, I will implement the shared Faster R-CNN model described by Girshick, [6]. I will follow the 4-step Alternating Training algorithm described in that paper. First, training the RPN by initializing an ImageNet-pre-trained model and fine-tuning for the RPN task. Next, train the Fast R-CNN detection network, also initialized by the ImageNet-pre-trained model, using the proposals generated by the RPN. In order to start sharing the convolutional layers, in the third step, Girshick then used the detector network trained in step 2 to initialize RPN training, but with fixed shared convolutional layers, only fine-tuning the layers unique to RPN. Finally, with the convolutional layers still fixed, we go back to the detector network and fine-tune the unique Fast R-CNN layers.

Once I have the basic Faster R-CNN working, I will compare it to the benchmarks reported in the paper. At this point, I will experiment with alternate architectures for the RPN, different training algorithms, and compare the results.

8 Conclusion

In summary, I'm fascinated by computer vision and believe that many of the promises of machine learning and artificial intelligence will rely on advances in this field of study. My intention of the my capstone project is to focus on the object detection challenge and review approaches used in recent years by state-of-the-art models, which have won, or placed highly, in challenges like the ILSVRC to better understand how object detection works in these models, so that I can begin exploring how to use them in practical applications as well as what the current challenges are.

References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, .

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, . URL http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[3] Mark Everingham and John Winn. The pascal visual object classes challenge 2007 (voc2007) development kit, 2007. URL http://host.robots.ox.ac.uk/pascal/VOC/voc2007/devkit_doc_07-Jun-2007.pdf.

[4] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit, 2012. URL http://host.robots.ox.ac.uk/pascal/VOC/voc2012/devkit_doc.pdf.

[5] Mark Everingham, Luc van Gool, Chris Williams, John Winn, and Andrew Zisserman. The pascal visual object classes homepage, . URL http://host.robots.ox.ac.uk/pascal/VOC/.

[6] Ross Girshick. Fast r-cnn, 2015. URL https://arxiv.org/pdf/1504.08083v2.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013. URL https://arxiv.org/pdf/1311.2524v5.

[8] T. S. Huang. Computer vision: Evolution and promise. *CERN School of Computing*, 19, 1996. URL http://cds.cern.ch/record/400313/files/p21.pdf.

[9] Fei-Fei Li, Justin Johnson, and Serena Yeung, 2017. URL http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture1.pdf.

[10] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015. URL https://arxiv.org/abs/1506.01497.

[11] (rbgirshick) Ross Girshick. Faster r-cnn (python implementation). URL https://github.com/rbgirshick/py-faster-rcnn.

[12] et al. Russakovsky, Olga. Imagenet large scale visual recognition challenge. 2014. URL https://arxiv.org/pdf/1409.0575.pdf.