WEAKLY SUPERVISED OBJECT DETECTION USING SUB REGIONS

APPLIED DEEP LEARNING FINAL PROJECT

Soroush Khadem
University of Colorado, Boulder
soroush.khadem@colorado.edu

December 10, 2020

ABSTRACT

Weakly supervised object detection is a challenging problem in computer vision. Using only image-level labels for an image, a model must learning bounding boxes for objects in an image. The state of the art for this task has gotten better in recent years, but the state of the art is far behind supervised object detectors. I propose a model that uses multiple regions centered on each proposal region to provide spatial consistency in the object being detected, and combat the local minimum problem with MIL detectors. The model is trained in an end-to-end fashion and produces good results on the PASCAL VOC dataset.

1 Introduction

Weakly Supervised Object Detection (WSOD) can enable detection of objects without the laborious task of hand-labelling bounding boxes, which is prone to error and is time consuming. Hand-drawn bounding boxes can also lead to bias, and prevent scaling to thousands of classes. In addition, WSOD allows for much more efficient dataset collection, for example by using a keyword search on the internet. Currently, WSOD performance is far behind the supervised version of object detection, but the task can still be useful. For example, a WSOD can be run as a first pass to generate pseudo ground truth boxes to later be refined, or the network could be used for some production system where labelling the data with bounding boxes is too hard. In addition, the task provides some useful insight into how deep learning can learn image representation. A typical WSOD network uses the fact that Convolutional Neural Networks (CNN) are very effective at extracting features from an image.

Most WSOD methods can be broken down into three main steps: region proposal, feature extraction, and region detection. Region proposal is how candidate bounding boxes are generated, and is typically done using either Edge Boxes [15] or Selective Search [13]. These methods use more classical computer vision techniques to extract regions of the image with high "object-ness" score, that is, high likelihood that there is an object in the region. This is done essentially by counting the number of edges, as defined by some kernel that slides over the image to detect edges. The feature extraction stage involves transforming the image into a better representation, which is usually done by applying a CNN backbone trained on a classification task. This part of the pipeline is relatively well-understood as many other tasks from supervised object detection to segmentation to attention networks employ the same tactic, and it has been shown that feature extractors, especially those trained on ImageNet generalize well to many tasks. The last stage is assign a score to each region $\mathcal{R}$ from the region proposal step, and then use ultimately use some method to turn these into a label for the image in order to back propagate losses.

2 Related Work

Advancements in the WSOD field come in a variety of different facets, and because the problem is so challenging and relatively unsolved, there are many possible improvements that can be introduced. One family of improvements lies in the region proposal part of WSOD. This is especially challenging, since there is no direct information about the bounding boxes. Thus, a loss function must be designed especially carefully to allow for proper parameter updates.

In [3], a cascade of convolutional neural nets is used. A Multiple Instance Learning (MIL) detector is trained, and then a method like Fast R-CNN is trained on the pseudo ground truth boxes. MIL treats each training image as a bag, and then selects instances from each bag to learn. The main problem with MIL is that the detection becomes fixated on the highest activating region within an object, rather than the whole object. This is the main drawback of [1] (the model I used), and was one of the focal points for the designed changes to the model. In [**?**], the MIL detector and box regression is combined to be two branches, and attention is used to better guide the localization. Attention is another popular method for WSOD, since it has inherit localization information. [5] utilizes this fact in a fascinating way: multiple transforms are applied to the same image during training and the attention maps are compared between them to encourage consistency between augmentations.

Another family of improvements has come to improve the region proposal or region refinement. Due to the nature of the problem, because bounding boxes are not regressed, the detector is only as good as the region proposals that are fed into it. Tang et al. [12] approach the problem by training a multi-stage region refinement step, where various feature maps from the backbone are combined to train a separate CNN to score proposals before feeding into the WSDDN-based head. This approach is highly intuitive, but it still uses the initial boxes as a seed to the refinement, so the first proposal must be good enough. Tang et al. [11] combine spatially overlapping regions, and the regions are refined online. That is, there is an iterative method where each step "grows" the proposed regions to include the whole object. This method is very promising, but is computationally expensive.

Other methods have been used to improve the training process itself. Sangineto et al. [9] use a self paced strategy to select easy images, and augment the training data iteratively. Liu et al. [7] show that there is instability in which region gets selected by a WSOD network, and utilize this information to train an ensemble of methods.

3 Data

In order to train a weakly supervised object detector, by definition, only image level annotations are needed. However, for evaluation purposes, it makes sense to also use a dataset that has ground truth bounding box annotations. Thus, the PASCAL VOC 2012 dataset is chosen [4]. This dataset contains a diverse set of images, spanning across 20 classes. There is a mix of cluttered and uncluttered images, which can make it particularly challenging for weakly supervised methods. There are 22,531 images in the entire dataset, and the data is split into `train`, `val`, and `test` sets, following a 25%, 25%, 50% split, respectively. For evaluation purposes, the `val` set was used, in order to save some computational cost. The data is distributed relatively evenly, although there is a bias towards the `person` class. See Fig. (1) for an analysis of the class distribution for the `train` and `val` data splits. As can be seen in Fig. (2), the images are from a variety of scenes, some more complex than others.

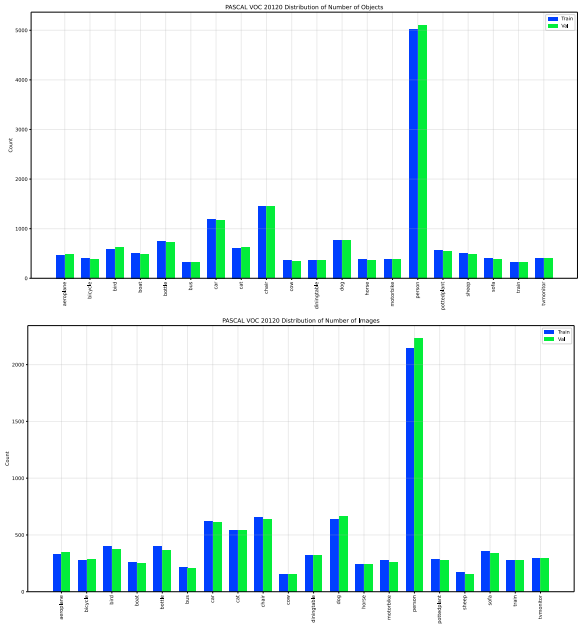

Figure 1: The PASCAL VOC 2012 dataset. Top: Distribution of number of objects, defined by the number of bounding boxes. Bottom: Number of images with at least one instance of each class.

Figure 2: A collection of random images from the `val` set. Drawn are the ground truth bounding boxes with class label on the top left corner.

4 Model

The primary base model used is the WSDDN of Bilen and Vedaldi [1], due to the simple to understand yet powerful nature of this model. This model is a foundational one, and the majority of work advancing the field uses WSDDN, while introducing improvements to various aspects of the model [5, 6, 11, 12]. Almost all work at least mentions WSDDN as a benchmark or baseline. The network consists of a few main components: a feature extraction backbone, a region proposal input, and two separate detection and recognition streams, discussed in depth below. Many variations of this model were attempted in order to improve performance of the model.

Region Proposals The most common region proposal methods are EdgeBoxes [15] and Selective Search [13], as these methods give good results very quickly. The authors of WSDDN tried both methods, and achieved better results using EdgeBoxes. In addition, EdgeBoxes provided more satisfactory results (qualitatively) on the PASCAL VOC data. Because there is no regression done on the regions, it is critical that each potential object is detected as one of the many (noisy) proposed regions. Thus, these regions have a large impact on the final evaluation score, so it was key to tune the parameters of the algorithm. Following [8], parameters were selected to give the network the best chance at finding the best box. Even so, many of the failure cases were seen to be partially because there were no better regions for a given class, showing one of the avenues for further improvement.

Feature Extraction Backbone As common in the field of computer vision, a model pre-trained on a larger dataset is chosen to use as a way to generalize feature extraction to any task. It has been shown that the weights for the convolutional layers learned by training on many images can be used to process new images and perform other tasks extremely well. Thus, the authors use VGG-16 pretrained on ImageNet as the primary backbone.

Classification and Detection Streams The network is split at its first fully connected layer (ϕ^6_{fc}). The image passes through the convolutional layers, resulting in a 512-d feature vector. Spatial Pyramid Pooling is applied using the the region proposals, and the resulting tensor is of size $|R| \times 512 \times 7 \times 7$, where $|R|$ denotes the number of regions $\{R_1, R_2, ...\}$. This is then passed through the two fully connected layers from VGG, ϕ^6_{fc} and ϕ^7_{fc}. The output is then split into two separate streams and passed through a classification layer $\phi^8_{fc\_c}$ and a detection layer $\phi^8_{fc\_d}$, before being multiplied element-wise, resulting in a score matrix. These two layers are both fully connected layers, but what is novel is that the classification stream does a softmax over the class scores, for each region, and the detection stream does a softmax over the regions, for each class. The resulting score matrix thus represents regions for each row, and class scores for each column. That is, every region (row) has a distribution for which class it belongs to, and each class (column) has a distribution for which region represents it.

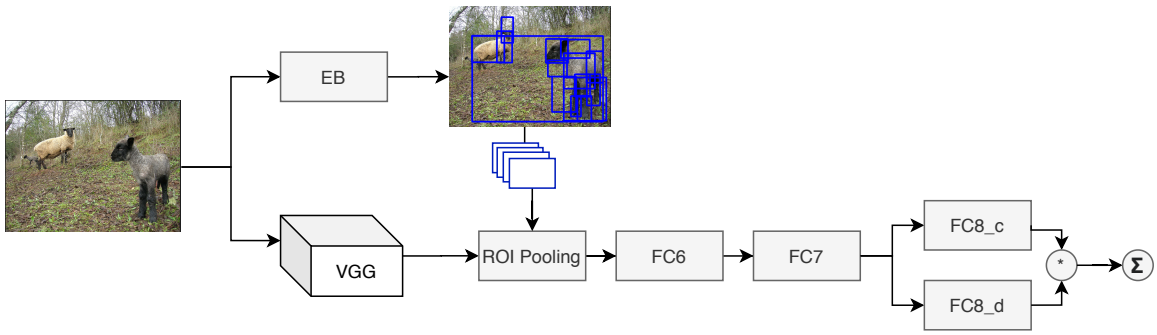

Figure 3: The original WSDDN network.

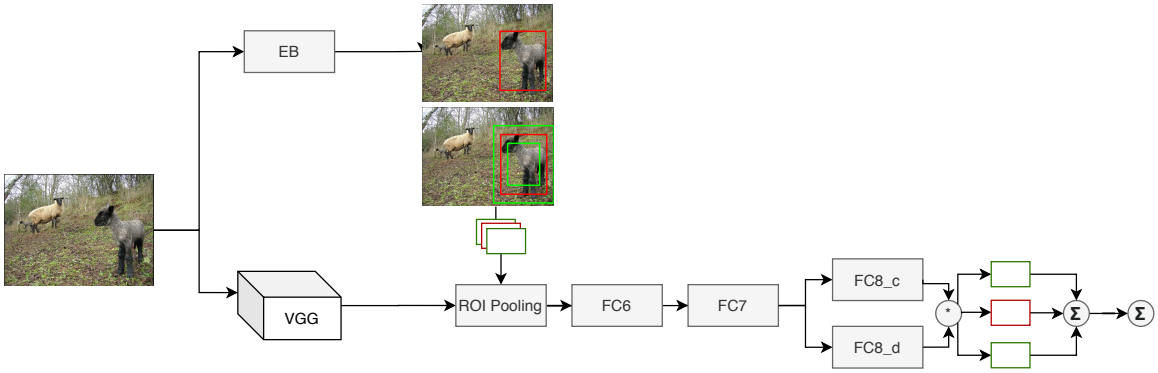

Figure 4: Modified WSDDN to use sub regions centered on the region proposals. Note that only one region is shown for simplicity, but the same operations are happening for each proposed region. The oroginal region propsal is shown in red while the centered regions are shown in green.

4.1 Modifications

As noted in the paper, there are two main failure cases for WSDDN: grouping multiple object instances with a single bounding box, and focusing on discriminate parts of the object (for example selecting the face of a person instead of the whole body). To attempt to combat these, I tried two new ideas: a new per-class regularization, and an addition of sub boxes.

Sub Regions In order to combat the shortcoming of the network where a sub region of the object gets detected, the idea of sub regions is introduced. The idea is that for each proposed region r, a certain number of new regions $\hat{r}$ are introduced. Different strategies to create $\hat{r}$ were experimented with, two of which are outlined below. Then, all regions are passed through the two classification and detection streams, and the outputs combined based on the original region. Thus, the output of the network remains the same, but with each region's score being the sum of itself and its sub regions. See Fig (4) for more details. This method is inspired by [6], where the region is converted explicitly to context and object by putting 0s in the center of a sub region, but differs in that here, the whole sub region is used.

In order to parameterize the calculation of these sub boxes, two parameters are introduced, s_1 and s_2. Given an original region $r = \{x_1, y_1, x_2, y_2\}$, these factors are used in the following way:

$$\hat{r}_1 = \{x_1 + w * s_1, y_1 + h * s_1, x_2 - w * s_1, y_2 - h * s_1\}$$
$$\hat{r}_2 = \{x_1 + w * s_1 * s_2, y_1 + h * s_1 * s_2, x_2 - w * s_1 * s_2, y_2 - h * s_1 * s_2\}$$

Here, w and h represent the width and the height of the region, respectively. So this means that the parameters define the percentage of the height and width to add to the two opposite corners of the bounding box. Thus it can be seen that by modifying the signs of s_1 and s_2 either both sub boxes can be bigger than the original region, both smaller than the original region, or one bigger and one smaller.

Per Class Regularization One key insight from a few different WSOD papers is using the fact that objects from the same class should look the same. For example, Song et al. [10] and **?**] use an initialization framework that frames the problem as retrieval of similar images from a graph. Wang et al. [14] embed multiple latent categories for each class that are learned for all images. Bilen et al. [2] encourage regions to be similar among learned clusters during training. These ideas are highly intuitive: one aspect that can help distinguish objects from background is their similarity to other objects of the same class. In WSDDN, Bilen and Vedaldi [1] introduce a spatial regularization term, shown here:

$$\frac{1}{nC}\sum_{k=1}^{C}\sum_{i=1}^{N_k^+}\sum_{r=1}^{|\bar{R}|}\frac{1}{2}(\phi_{k*i}^{y})^2(\phi_{k*i}^{fc7}-\phi_{kri}^{fc7})^T(\phi_{k*i}^{fc7}-\phi_{kri}^{fc7}) \tag{1}$$

Here, n is the number of images (indexed by i), C is the number of classes (indexed by k), $\bar{R}$ is the set of regions with 60% IOU with the highest scoring region, denoted by $* = \texttt{argmax}_r \phi_{kri}^{y}$ (indexed by r). What this is essentially doing is using the highest scoring region for each class, and penalizing other regions spatially near that region for having different activations (ϕ^{fc7}). While this works for helping to ensure spatial continuity in the distribution of regions associated with the same class, it does not have any effect on other objects in the same image (in different parts) that share the class label. This can be part of the reason the network tends to miss many of the same object in one image. That is, if there are many people in one image, the network will detect only one of them. Thus, a per-class regularization is introduced:

$$\lambda\sum_{k=1}^{C}\sum_{r=1}^{|\bar{\hat{R}}|}(\phi_{k*i}^{y})\cdot(\phi_{k*}^{fc7}-\phi_{kr}^{fc7})^T(\phi_{k*}^{fc7}-\phi_{kr}^{fc7}) \tag{2}$$

While similar to spatial regularization, this term differs importantly in the choice of $\hat{R}$. Here, for a given k:

$$\hat{R} = \{r \in R \mid \texttt{argmax}_c \phi_{cr}^{y} = k, \phi_{kr}^{y} > \rho\}$$

That is: all regions which share k as their highest scores, and that have a score greater than a threshold ρ, are enforced to have similar feature activations. The idea here is that if the network is confident enough that multiple regions (within the same image) belong to the same class, these regions should be similar in feature space. The threshold ρ is used to only enforce this loss on regions that have high confidence. Otherwise, all regions (including background ones) would be included in the term. The parameter λ is a hyperparameter used to balance regularization.

5 Results

Sub Boxes By introducing a weighting factor to each sub box, the results were quite encouraging. The mean average precision increased from the baseline for each threshold IOU value. Qualitatively, also, the results showed that the network was learned to recognize entire regions more accurately. Note that in the images that follow ground truth boxes are in green and detections are in red, with confidence scores shown next to the label.

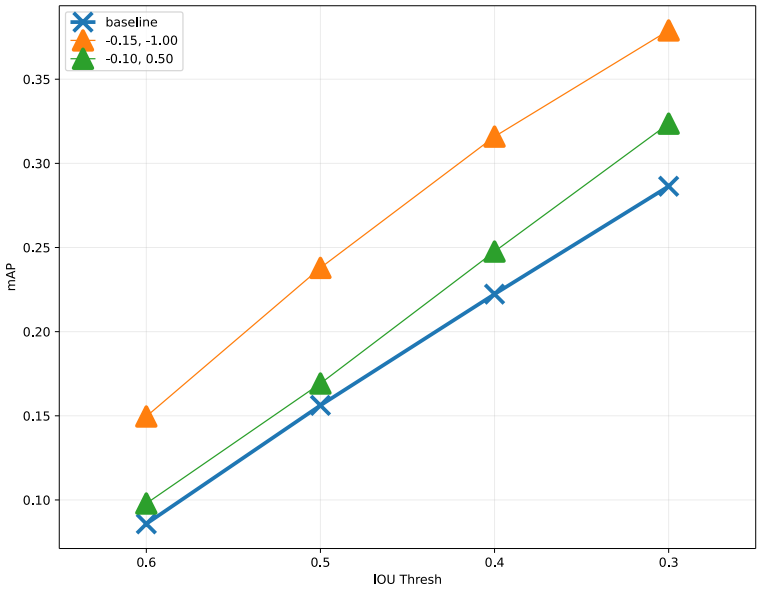

Figure 5: The mean AP, calculated using the COCO metric, at different IOU thresholds. Using sub boxes improved the vanilla WSDDN. The orange line corresponds to $s_1 - 0.15$, $s_2 = -1$, which means that one region is smaller and one is larger.

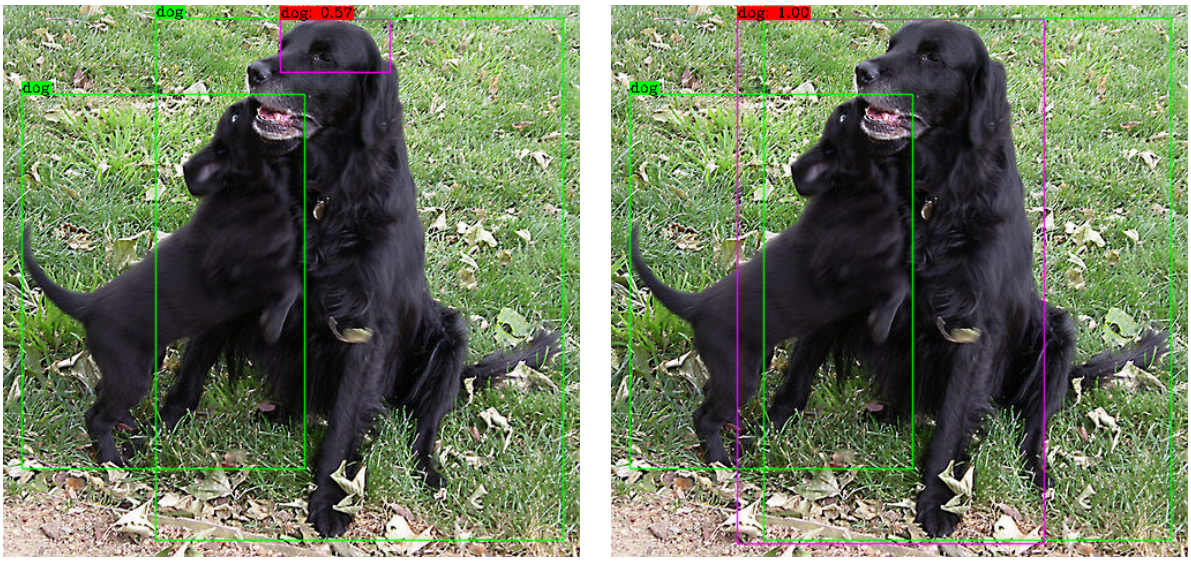

Figure 6: Left: baseline WSDDN. Right: sub region WSDDN. The network was able to detect the entire bounding box

Figure 7: Left: baseline WSDDN. Right: sub region WSDDN. while still not the full region, the network did improve its detection

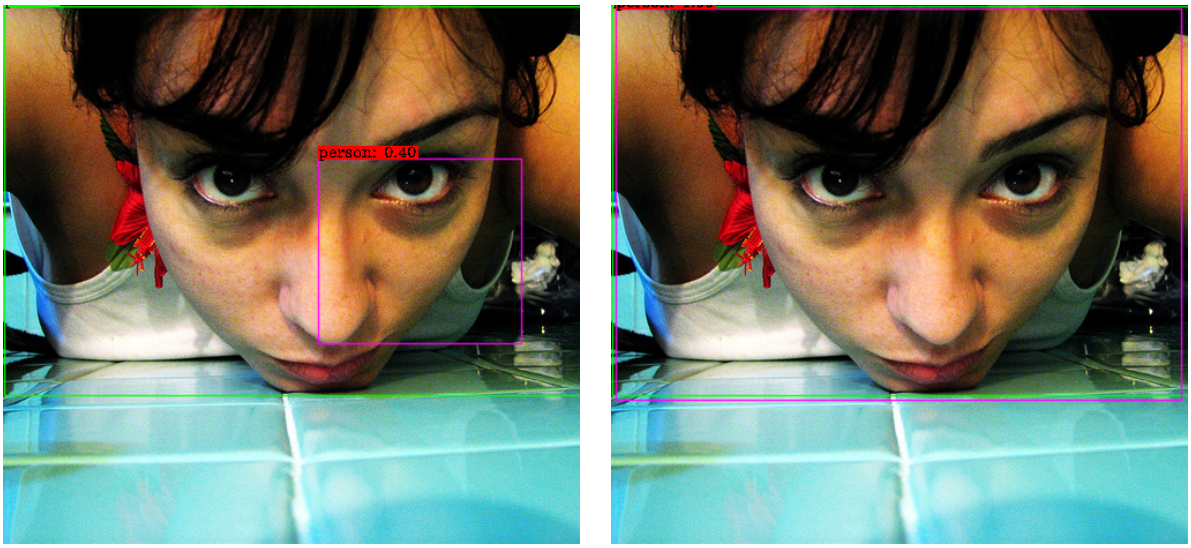

Figure 8: Left: baseline WSDDN. Right: sub region WSDDN. In this case, the entire region was correctly detected

Class Regularization For the class regularization, the goal was to pick up on the multiple objects that are usually missed within the same image. The results for the two trials run improved the mAP of the baseline WSDDN method. A few different parameters were chosen for the λ and the score threshold. These values must be chosen carefully, as there were instances where the model actually diverged and the losses went to infinity.

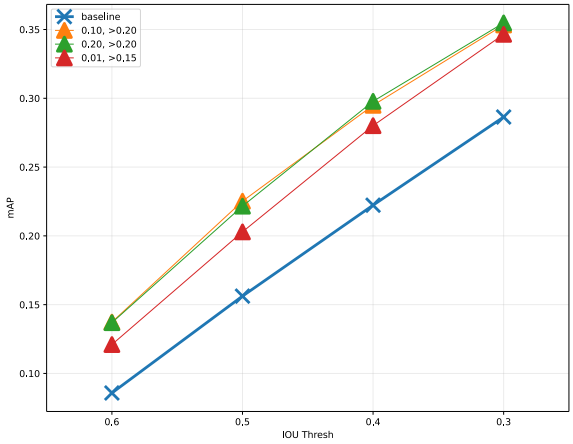

Figure 9: The mean AP, calculated using the COCO metric, at different IOU thresholds. Using regularization across the image for the same class improved results

Figure 10: Left: baseline WSDDN. Right: sub region WSDDN. A clear improvement! The network detects both bottles correctly

Figure 11: Left: baseline WSDDN. Right: sub region WSDDN. While not perfect, the network detects multiple bounding boxes, one for the front cat and one for the cat behind it, while the baseline focuses on the activation of the head of the cat

6 Ablation

Many different configurations of scales were attempted. All except for the two shown in the results section performed worse than baseline. Below is a plot showing those results:

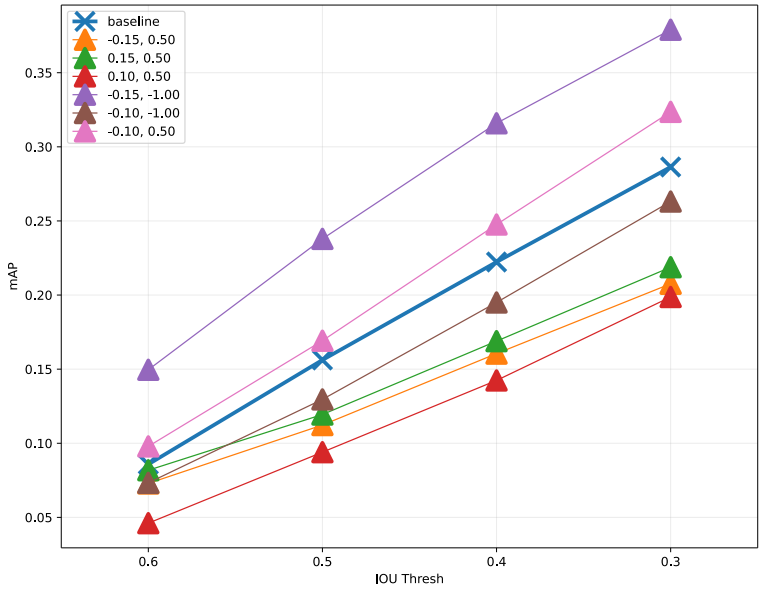

Figure 12: The mean AP, calculated using the COCO metric, at different IOU thresholds. Using sub boxes improved the vanilla WSDDN.

In addition, another sub box strategy was attempted, by defining overlapping regions on the interior of the main region, as shown below in Fig. 13. The intuition here was that within the correct region, there would exist high activations for the same class, so those should help each other. However, these losses would never converge low enough, and the mAP never reached over 0.5, so the plots showing them are ommitted. Along with these changes, a few different architecture changes were attempted, such as adding a different fully connected layer for each sub region, but this also never converged.

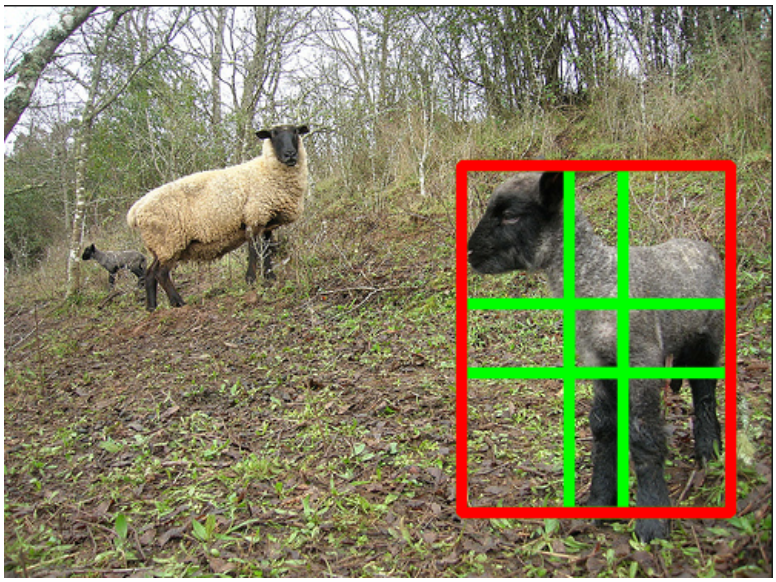

Figure 13: An attempt at defining the sub regions as interior overlapping regions. The method never worked.

7 Conclusion

Weakly supervised object detection is an incredibly interesting problem that can illuminate some of the core ideas from object detection. While the performance of WSOD will never reach that of supervised object detection, advancements in the field will help bring advancements to the field of computer vision as whole.

In this work, I took the WSDDN model, one of the benchmarks for WSOD, and suggested two different improvements. One improvement is to add multiple sub regions per proposed region, and sum the scores of those regions together to get the final score for the original region. In addition, I propose a regularization across other regions from the same class, which can encourage the network to detect multiple objects in the same image. It should be noted that I used my own implementation of WSDDN, which admittedly performs worse than the official paper's results [1]. Thus, instead of comparing to Bilen and Vedaldi [1]'s results, I compare to my own baseline, so any bugs that were in the code are consistent across methods.

In the end, I believe that the main factor that degrades performance is the subpar region proposals. I think it would be interesting to analyze how a trained WSOD model performs when passing in the ground truth boxes. If the networks fail on perfect input data (which I don't think they will) then maybe the focus should be on that part of it. Thus, I think the most promising work is similar to those of [3, 5, 11, 12], since these methods refine the regions themselves.

References

[1] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. arXiv:1511.02853, 2015.

[2] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with convex clustering. CVPR, 2015.

[3] Ali Diba, Vivek Sharma, Ali Mohammad Pazandeh, Hamed Pesiavash, and Luc Van Gool. Weakly supervised cascaded networks. CVPR, 2017.

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

[5] Zeyi Huang, Yang Zou, Vijayakumar Bhagavatula, and Dong Huang. Comprehensive attention self-distillation for weakly-supervised object detection. arXiv:2010.12023, 2020.

[6] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. arXiv:1609.04331, 2016.

[7] Boxiao Liu, Yan Gao, Nan Guo, Xiaochun Ye, Fang Wan, Haihang You, and Dongrui Fan. Utilizing the instability in weakly supervised object detection. CVPR, 2019.

[8] Sean McMahon, Niko Sunderhauf, Ben Upcroft, and Michael Milford. How good are edge boxes, really? ECCV, 2015.

[9] Enver Sangineto, Moin Nabi, Dubravko Culibrk, and Nicu Sebe. Self paced deep learning for weakly supervised object detection. arXiv:1605.07651, 2018.

[10] Hyun Oh Song, Ross Girshick, Stefanie Jegelka, Julien Mairal, Zaid Harchaoui, and Trevor Darrell. On learning to localize objects with minimal supervision. arXiv:1403.1024, 2014.

[11] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. arXiv:1704.00138, 2017.

[12] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. http://sunw.csail.mit.edu/2015/papers/13_McMahon_SUNw.pdf, 2018.

[13] J.R.R Uijlings, K.E.A van de Sande, T. Gevers, and A.W.M Smeulders. Selective search for object recognition. IJCV, 2013.

[14] Chong Wang, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan. Weakly supervised object localization with latent category learning. ECCV, 2014.

[15] C. Lawrence Zitnick and Piotr Dollar. Edge boxes: Locating object proposals from edges. ECCV, 2014.