Question 1

What was the paper about?
This paper discusses a method for per-pixel object detection that overcomes some downsides of anchor-based detectors. These downsides include a lack of generalizability across resolutions/aspect ratios, extra hyper-parameters, and computationally intensive metrics (e.g. repeated IoU calculations). Regressing on a single proposed region rather than densely placed anchors also improves efficiency during inference. They go on to resolve one of the key problems with FCOS, that of resolving overlapping objects accurately, by introducing the use of Feature Pyramid Networks (FPN). The FPNs check whether a particular object is most likely at each layer versus the layer before, and since most overlapping objects are expected to have different sizes (say something a person is holding in front of that person), the ambiguity is resolved when only one layer maximizes the likelihood of there being an object at that location. While FPNs have been used with anchor detection, the specific levels of the pyramid led to certain size anchor boxes per layer, which might not be optimal since all object detection tasks are not likely to require similar size and shape boxes.

What was the key idea or intuition?
Many key vision tasks, including semantic segmentation, keypoint detection, counting, and depth estimation. can be performed using FCNs, but object detection has not followed suit. The authors decided to test whether object detection could be performed as a per-pixel prediction instead - and provide improved performance since object box sizes and shapes would be less restricted by the hyperparameters determining the potential object box shapes that are used in anchor box approaches.

What was the design of the experiments and how were the results analyzed? The model was evaluated on the COCO dataset by comparing to other models in the same problem space, and by ablation of new concepts. Best possible recall (BPR) was shown to be as good or better than anchor-based detectors. Ambiguous (overlapping) samples were shown to be largely negated by the pyramidal network structure by counting how many locations could be assigned to multiple bounding boxes with and without ctr-sampling and the FPN architecture. Model performance was tested with and without the centeredness score, and similar ablations were performed for different loss functions, normalization schemes, etc. Overall model performance was evaluated as fairly as possible against other models (i.e. removing extra features to directly compare the backbone architectures), and finally was tested on more challenging datasets like real-time data and crowded images with many ground-truth boxes. Real time analysis using a version of the FCOS trimmed down for speed was also shown to be effective.

What are the major contributions or findings?
The major contribution is an FCN that can perform object detection, speeding up training, improving accuracy, and decreasing the complexity of the training process. The authors found that anchor boxes were not necessary for state of the art object detection, though other anchor-free methods had been used before. Other important improvements include removal of

ambiguity from overlapping targets by making predictions at different feature map resolutions, and removal of low-quality bounding boxes with a “centeredness” score that penalizes suboptimal proposals.

What are the strengths and weaknesses of the method? Versus other detectors.
Strengths consist of the similarity in structure to other FCN networks along with a reduction in design complexity compared to anchor box based methods. Due to the structural similarity, design advances in other tasks can be more easily applied to frameworks like FCOS. Without bounding boxes, the number of design hyperparameters are decreased, making the training process simpler and more accessible. Finally, loss is computed against only one proposed bounding-box at a time rather than against many anchor-boxes, resulting in faster training and testing while maintaining state of the art performance. One possible weakness of a FCOS approach is that the large stride at later layers results in missing true objects (low recall), however the authors state that experimentally they did not find this to be the case.
The primary weakness of the FCOS method is that multiple ground truth objects that overlap and have similar central points tend to be difficult to resolve - which object is the intended target? It is for this reason that the authors introduce the use of an FPN and a centerness branch. While the FPN and centerness branch dramatically improve the recall of the model, the authors point out that there is still a 2.66 percent ambiguity rate, where the smallest box is chosen out of a set of overlapping boxes. They suggest that this is not important, but do not look into whether there is a specific type of instance that results in this ambiguity, so it may be that a particular type (or situation) of rare occurrence/overlap results in a mistake 100% of the time.

There is still a hyperparameter *r* that defines the size of the center box, which can vary based on the data set. Finding a way to learn this parameter during training might make the method more robust. Similarly, the loss function involves some hyperparameters (α, γ in Focal Loss) that “give better accuracy”, and perhaps a more carefully designed loss function might be either simpler or more robust.
What are your suggestions to improve this research?
Probably some sort of transformer that mimics this behavior so that the functionality is not impacted by the receptive field? The loss function, partially based on focal loss, requires two hyperparameters with only some explanation as to why certain values are chosen, α, γ. More details about appropriate values for alpha and gamma, beyond that using them improves accuracy, would be helpful in determining what these parameters should be for different use cases. The requirement for the centerness calculation to be on a separate branch rather than computed from the regression vector is mentioned, but not explained - it would have been good to give at least some cursory non-empirical explanation as to why this difference is important.

Other comments on the paper:

They had a few situations, like establishing that the BPR would not be low with an FCN, where they admit there could be a flaw, but simply stated that “empirically this was not the case.” In reality, that really means that “empirically, on the one or two data sets we tested, this was not the case”, which is not really such a strong statement.

E.g. “For FCOS, at the first glance one may think that the BPR can be much lower than anchor-based detectors because it is impossible to recall an object which no location on the final feature maps encodes due to a large stride. Here, we empirically show that”
Even though the Average Recall was higher for FCOS+centerness, that average recall was still below 60. It would be interesting to explore where the failures come in and what might be done to improve the accuracy.

Implementation

Our implementation closely followed the [torchvision] version of FCOS. For efficiency, we concatenated all FPN points into a single dimension wherever applicable, so tensors were generally of size [bs, total_points, channels]. We followed the pytorch method of multiplying regression outputs by point stride, rather than dividing targets (the other method generally led to untrainable models). Other implementation details should be obvious from the code.

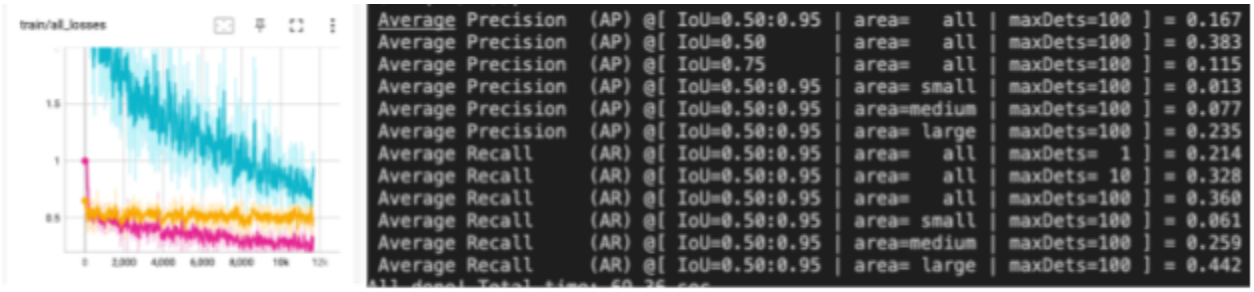

Figure 1 Training losses and evaluation performance for FCOS: Training loss (left), approximately followed the expected trajectory for classification (blue) and regression (pink). Centered-ness appeared to not change at all, likely due to a bug. Evaluation of the model (right) shows 0.383 mAP at IoU=0.5, more than 10% below the target.

Figure 2 Visualizations of model output: predicted bounding boxes from our trained model are not entirely reasonable. The left image shows a reasonable bounding box, but the predicted class is “bottle”. The box on the right appears to be around nothing and is predicted as “cat.”

Our model was able to run, but appears to not train sufficiently (**Figure 1**). It is unclear whether this is entirely due to bugs in training or bugs in inference, as the mAP at IoU=0.5 (0.383) is poor, but at least somewhat relevant, and visualizations of predictions (**Figure 2**) seem to be somewhat nonsensical. We note that our code is extensively commented, and would greatly appreciate feedback on what isn’t working. Visualization was performed by code in code/visualize.py.