Deep Learning Project
Team 13- ShowMeYourConvolutions

Shriya Akella Sukrit Rao Shambhavi Rege Dhruv Saxena

Abstract

We present a supervised learning approach to object detection with our model achieving an AP[0.50:0.95:0.05] of 25.1% on the hidden test set used to evaluate the performance of each team.

1. Introduction

"Object detection is the task of jointly localizing and recognizing objects of interest in an image. Specifically, the object detector has to predict a bounding box around each object, and predict the correct object category" (Huang et al., 2021) (p.3).

Our task is to build a deep learning model to perform object detection using self-supervised or semi-supervised approaches. The dataset provided consists of a mix of labeled and unlabeled data. We're provided with 30,000 training images, and 20,000 validation images. Both these sets consist of 100 classes of common objects such as *person, dog, cart, bird*, etc. The unlabeled set consists of 512,000 images, which are to be used to learn useful representations that can be subsequently transferred for supervised finetuning on the labeled dataset. The evaluation metric we're judged on is the **Average Precision (AP)** computed over a range of **Intersection over Union (IoU)** values, ranging from 50% to 95% at intervals of 5%.

The best performing model we were able to obtain was a **Faster-RCNN** model with a **MobileNetv3** backbone, achieving an AP[0.50:0.95:0.05] of 25.1% on the hidden test set used to evaluate the performance of each team.

2. Related Work

The task of object detection is not new. It has an active research community and object detection benchmark challenges such as PASCAL VOC (Everingham et al., 2010), ILSVRC (Russakovsky et al., 2015) and MS COCO (Lin et al., 2014) receive numerous submissions each year. With the recent rise in the transformer architecture proposed by Vaswani et al. (2017), followed by its application to the vision domain (Dosovitskiy et al., 2020), there has been a shift in the submissions that sit atop the object detection leaderboards, with techniques applying the transformer architecture occupying a greater number of spots at the top. Before this trend, however, the top-performing techniques consisted of more "classical", convolution-based approaches. A typical characteristic of these approaches was that they were comprised of two stages. The first stage usually determines regions of interest (ROIs) using either statistical techniques such as selective search (Girshick et al., 2013) or a region proposal network as proposed by Ren et al. (2015). The second stage then refines the ROIs produced by the first stage. A drawback of these techniques is that they depend heavily on how they generated anchor boxes, reference boxes placed at different positions in the image that are used to make predictions. Typically, they also required hand-crafted components like non-maximal suppression (NMS) that prevented them from being optimized end-to-end (Zhang et al., 2022).

Redmon et al. (2015) proposed a unified, single-stage algorithm wherein they directly predict objects at predefined locations by refining the position and scale of their anchors, thus, resulting in faster inference speeds with applicability in real-time systems. However, this increase in inference speed comes at the cost of prediction quality (Huang et al., 2021).

DETR, or **DE**tection **TR**ansformer (Carion et al., 2020), is a recent transformer-based architecture that removes the need for the hand-crafted NMS component, present in Faster-RCNN, and thus, is fully differentiable and can be optimized end-to-end (Huang et al., 2021). It does so by learning how to remove redundant boxes using the Hungarian Loss (Munkres, 1957).

Sung et al. (2017) show that it is difficult to optimize the Hungarian loss. As a result, there have been several works attempting to overcome this difficulty [Zhu et al. (2020), Chen et al. (2022)]. Deformable DETR (Zhu et al., 2020), is one of the more popular techniques that proposed an improvement on DETR by using multi-scale deformable attention modules. It can attend to a small set of learned locations over multiple feature scales, rather than uniformly attending over a single-scale feature map (Huang et al., 2021).

The techniques mentioned above make use of the super-

vised learning approach, where the backbone of the model is first trained on ImageNet, after which it is transferred to object detection tasks. Self-supervised learning has recently emerged as an effective alternative supervised pretraining, where the idea is to generate labels from unlabeled data, thus the data, itself, provides the supervision. This process is known as solving the pretext task (Huang et al., 2021). Some approaches like SimCLR (Chen et al., 2020), and MOCO (He et al., 2019) have experimented with initializing object detection backbones by learning unsupervised representations on ImageNet. However, more recent works such as Bar et al. (2021) show that pretraining the object detection heads, along with the backbone, improves performance on object detection tasks.

3. Approach

In our literature review, we shortlisted the most applicable strategies and models on the basis of their performance on MS COCO, relevance to our task, and logistical viability for our project. One of the first models we selected was FasterRCNN, a popular object detector. It is the current evolutionary state of the family of region-based CNNs (R-CNN) which can accurately and swiftly predict the locations of different objects in an image. It operates in two steps :

- The backbone extracts features from the images, which are fed into the Region Proposal Network (RPN) and produces object proposals.
- Object predictions are made on the feature maps, inside every anchor box.

We experimented with a few approaches before settling on a final model for the final leaderboard, which we talk about in Section 4. All experiments were performed on the NYU HPC Cloud Bursting cluster provided.

3.1. FasterRCNN-ResNet50

The FasterRCNN-ResNet50-FPN-V2 (He et al., 2015) is a large model with about 45M parameters. We used the standard FasterRCNN_ResNet50_FPN_V2 class from PyTorch Vision Models. We set the pretrained weights to 'False' in order to train our model from scratch.

Metric	All	Small	Medium	Large
Average Precision (0.50:0.95)	18.3%	14.6%	15.5%	19.3%
Average Recall (0.50:0.95)	46.9%	29.1%	37.6%	49.1%

Table 1. Results of FasterRCNN ResNet50 on Validation Set

Table 1 shows the results of FasterRCNN-ResNet50 model. This model took extremely long to train and did not converge even with considerable amount of hyper-parameter tuning and training.

3.2. FasterRCNN-MobileNetV3 with Pretext Task

The backbone of the FasterRCNN was initialised with random weights. In order to give the backbone more context regarding the features present in the images, we used a pretext task. We trained the backbone on the unlabelled images which were rotated by 90, 180, 270. The task was to classify the unlabelled and the rotated unlabelled images into the rotation associated with the image. (Gidaris et al., 2018) Then, we used this trained backbone in our FasterRCNN model.

Metric	All	Small	Medium	Large
Average Precision (0.50:0.95)	21.9%	1.3%	7.7%	26.0%
Average Recall (0.50:0.95)	38.7%	1.4%	14.8%	46.3%

Table 2. Results of FasterRCNN MobileNetV3 with Pretext Task on Validation Set

4. Results

Metric	All	Small	Medium	Large
Average Precision (0.50:0.95)	21.3%	1.3%	6.8%	25.2%
Average Recall (0.50:0.95)	37.7%	1.4%	13.7%	45.0%

Table 3. Results of FasterRCNN MobileNetV3 on Validation Set

Metric	Test Set	Val Set
Average Precision (0.50:0.95)	25.1%	21.3%

Table 4. Comparison of Results of FasterRCNN MobileNetV3 on Validation and Test Sets

Table 3 shows the final results of our experiment. We used the FasterRCNN model with MobileNetV3(Howard et al., 2017) as the backbone. It was the smallest model in class with only 19M parameters. Table 4 shows the comparison of our model performance on the hidden test set and validation test provided

It can be observed that our model performs much better on bounding boxes with large areas as opposed to small areas. An example of this behavior is shown in Figures 2 and 3. This can be attributed to there being fewer parameters in the model.

We also observed that in a lot of situations, while the box predicted by the model is accurate, the predicted label is incorrect, resulting in an overall incorrect prediction. This can be seen from Figure 4. This leads us to believe that

our model performance can be improved by performing self-supervised pertaining on the unlabeled set. This would ensure that the backbone of our model is better at extracting features that would be useful during classification.

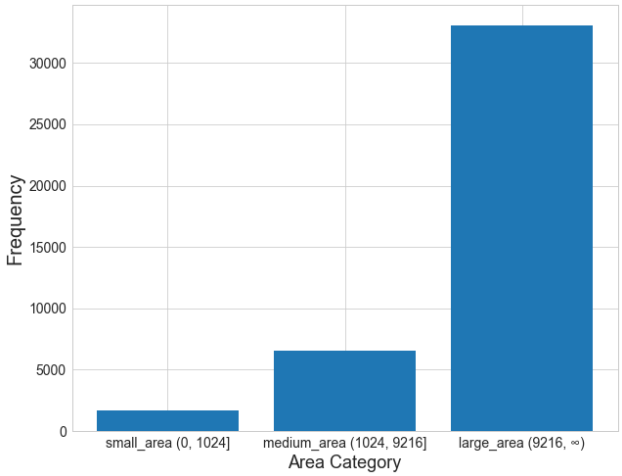

Figure 1. shows the area-wise breakdown of the boxes in the training set. It can be observed that the number of large boxes far outnumbers the number of medium and small boxes, leading to considerably better performance on large boxes

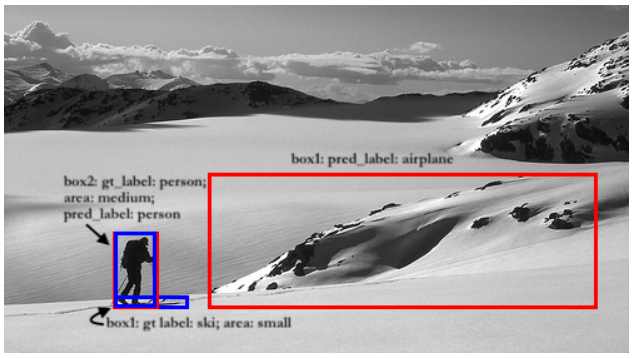

Figure 2. shows an example where our model is able to correctly identify the medium-size box but performs poorly on the small-size box

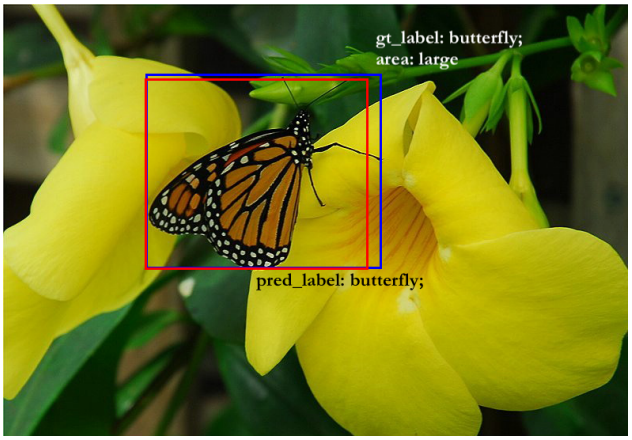

Figure 3. shows an example where our model is able to correctly identify the large-size box

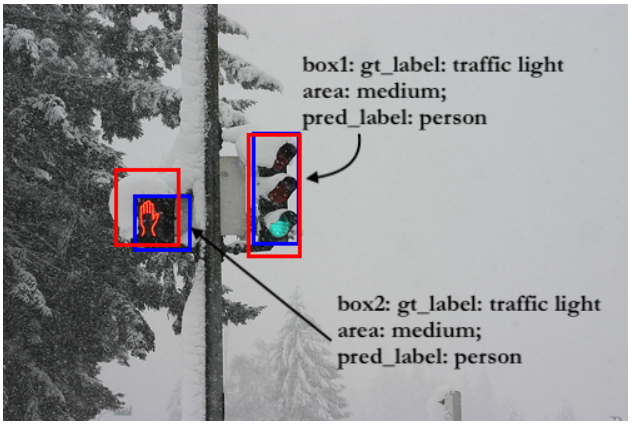

Figure 4. shows an example where our model is able to correctly identify the medium-size box but performs poorly on the small-size box

5. Future Work

Due to resource and time constraints, we leave the exploration of self-supervised approaches such as DINO (Zhang et al., 2022) and VICReg (Bardes et al., 2021), to future work. We also explored semi-supervised approaches, wherein we used a model trained on the labeled data to generate labels on the unlabeled data, using some pre-defined confidence threshold. In our experiments, we set this confidence threshold to 50%, however, a deeper exploration into determining the optimal confidence threshold is required.

References

Bar, A., Wang, X., Kantorov, V., Reed, C. J., Herzig, R., Chechik, G., Rohrbach, A., Darrell, T., and Globerson, A. Detreg: Unsupervised pretraining with region priors for object detection, 2021. URL https://arxiv.org/abs/2106.04550.

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2021. URL https://arxiv.org/abs/2105.04906.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers, 2020. URL https://arxiv.org/abs/2005.12872.

Chen, Q., Chen, X., Wang, J., Feng, H., Han, J., Ding, E., Zeng, G., and Wang, J. Group detr: Fast detr training with group-wise one-to-many assignment, 2022. URL https://arxiv.org/abs/2207.13085.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL https://arxiv.org/abs/2010.11929.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, June 2010.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations, 2018. URL https://arxiv.org/abs/1803.07728.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013. URL https://arxiv.org/abs/1311.2524.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning, 2019. URL https://arxiv.org/abs/1911.05722.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. URL https://arxiv.org/abs/1704.04861.

Huang, G., Laradji, I., Vazquez, D., Lacoste-Julien, S., and Rodriguez, P. A survey of self-supervised and few-shot object detection, 2021. URL https://arxiv.org/abs/2110.14711.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft coco: Common objects in context, 2014. URL https://arxiv.org/abs/1405.0312.

Munkres, J. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957. doi: 10.1137/0105003. URL https://doi.org/10.1137/0105003.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection, 2015. URL https://arxiv.org/abs/1506.02640.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015. URL https://arxiv.org/abs/1506.01497.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning, 2017. URL https://arxiv.org/abs/1711.06025.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L. M., and Shum, H.-Y. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022. URL https://arxiv.org/abs/2203.03605.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. Deformable detr: Deformable transformers for end-to-end object detection, 2020. URL https://arxiv.org/abs/2010.04159.