Deep Learning Fall 2022 Final Project Report - Team 21

Caglar Dogan [1] **Aryann Dharamsey** [1] **Kristin Mullaney** [1] **Tyler Niiyama** [1]

Abstract

The scarcity and expansiveness of labeled training data often serve as limiting factors in the performance of deep learning models. In recent years, this has led to studies of self-supervised learning algorithms that allow for the learning of efficient feature representations from unlabeled data and the use of such algorithms as a part of semi-supervised learning procedures of deep learning models. In this paper, we present an empirical study on the use of such methods in object detection. Our model utilizes a hybrid approach that combines a self-supervised learning stage for the training of a backbone that provides feature representations for images and a supervised learning stage for object detection and classification.

1. Introduction

Semi-supervised object detection is an active area of research within the broader umbrella of semi-supervised learning. This paradigm aims to achieve better results with the same amount of labeled training data by building an internal model of the world through the learning of structures present in unlabeled datasets and using this model during the supervised training stage. This lessens the need to pay for expensive, manually labeled data to increase model performance.

In semi-supervised object detection, self-supervised representation learning algorithms are often used for the inference of structures in unlabeled datasets with the training of backbone networks that output feature vectors to efficiently represent input images. Amongst these, joint-embedding energy-based models, which can provide efficient representations that can be used to distinguish objects of distinct nature but are invariant to arbitrary details of exact initiations of objects, are particularly predominant.

Joint embedding architectures' output representations of input images are most efficient for object detection when similar images have close representations and distinct images have different representations. For this reason, different training methodologies have been constructed to help minimize the distance between the representations of similar objects while keeping the representations of distinct objects far. These different approaches can be summarized under the categories of contrastive and regularized methods.

Contrastive methods stimulate the learning of efficient representations by requiring a low distance between the representations of images obtained by the augmentations of the same image (positive samples) and requiring a high distance between the representations of augmentations of different images (negative samples). Using the negative samples to prevent the collapse of representations, contrastive methods have been successful in image recognition tasks in recent years (Chen et al., 2020). However, these methods are computationally expensive, and their efficiency diminishes with increasing representation space dimensionality.

Regularized methods, on the other hand, achieve the same result by utilizing a loss function whose value increases with a lower variance of different representations while still being positively related to the distance between the representations of augmented images coming from the same sample.

After the self-supervised phase, a supervised learning stage is employed to train a full model for object detection. This stage includes the training of the necessary networks to predict bounding boxes for objects and to classify the objects in each bounding box. A fine-tuning of the backbone for the specific classes to be distinguished can also be carried out simultaneously in this stage.

Building upon these approaches, we demonstrate the effectiveness of self-supervised representation learning in object detection by following this semi-supervised framework in this paper. For this, we present our results from a new artificial neural network trained on a dataset predominantly composed of unlabeled data and signify the changes in performance stimulated by changes to our architectures between our intermediate iterations.

2. Related Work

Our work in the presented semi-supervised object detection model is built upon recent results from three particular fields: Backbone Architectures, Regularized Representation Learning Algorithms, and Object Detection Models.

2.1. Backbone Architectures

Self-supervised representation learning is often carried out on backbone architectures built upon variations of convolutional neural networks (CNNs). An architecture of particular interest is the ResNet architecture (He et al., 2015), in which residual connections allow the model to overcome the issue of vanishing gradients during training. ResNets50 is known to work particularly well as a backbone and thus is utilized by our model.

2.2. Regularized Representation Learning Algorithms

One particular area of research relevant to our work here is the study of regularized representation learning algorithms. Some recently developed methods have been successful in image recognition (Zbontar et al., 2021) and have stimulated the use of such methods in object detection, with VICReg (Bardes et al., 2021) being a particularly powerful example. VICReg regularizes variance and decorrelates variables through redundancy reduction and covariance regularization, achieving state-of-the-art results on downstream tasks. With this, the need for negative samples is removed, and the computational power required for achieving the same performance is greatly reduced. Our approach utilizes this learning method for this reason.

2.3. Object Detection Models

One type of CNN that was designed for object detection is R-CNN (Girshick et al., 2013), which generates regions of interest (RoIs) that are processed by a CNN to extract features and classify objects. While R-CNN was successful for object detection, it has been surpassed by more efficient models such as Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren et al., 2015), which can process RoIs faster and achieve better performance overall. Furthermore, Faster R-CNN is generally considered the most efficient and effective model for object detection.

3. Approach

In our project, we decided to use semi-supervised learning to utilize the information from our large unlabeled dataset alongside the labeled training data. For this, we trained a backbone with self-supervised learning to get efficient representations of features in objects before combining this with an object detection model to get final predictions.

In our self-supervised backbone training phase, we chose a ResNet50 architecture as our backbone for processing unlabeled data. We chose ResNet50 because it performs well on object classification tasks but is more efficient than deeper models in the ResNet family. To effectively train this backbone, our plan was to include several additional features in what would be a customized ResNet50 representation learning algorithm.

First, we added stochastic data augmentation to randomly alter the input data in various ways and improve the generalization performance of the model (Shorten & Khoshgoftaar, 2019). Our hope was to use distinct samples from the same images to train our backbone to minimize the distance between the representations of similar objects.

Next, we implemented an information maximization technique for regularization (Zbontar et al., 2021). This adds constraints to the model's optimization process to reduce the risk of overfitting while maximizing the amount of information shared between the model's input and output. We did this to encourage the backbone to learn more meaningful features without the need for computationally expensive contrastive methods. For this, we choose to use VICReg (Bardes et al., 2021) loss, as it allowed the use of symmetric networks in training and works well for object detection. Then, to be able to use the VICReg (Bardes et al., 2021) loss, we added a multi-layer perceptron expander with an output size of 8192.

We then utilized this backbone in a Faster R-CNN network to train for object detection using the given labeled data. We chose Faster R-CNN because it achieves strong performance on a variety of object detection benchmarks and because it is faster, more efficient, and more accurate than its predecessors, Fast R-CNN and R-CNN.

4. Implemented Iterations and Results

For our initial implementation, we trained the backbone for 10 epochs using a batch size of 8 and a learning rate of 0.001. The loss lowered for the first few thousand steps of training before stagnating at 45. We then incorporated the backbone weights into the Faster R-CNN and trained on the labeled training data for five epochs. In the supervised component, 3 out of the 5 convolutional layers in the backbone were unfrozen and continued to train. The preliminary evaluation results generated using the training set left a lot to be desired. The average precision on various box sizes was between 0.000 and 0.001, and the average recall was between 0.000 and 0.004. This indicated to us that our model was not learning properly.

Upon further investigation of the bounding box predictions and predicted classifications, we noticed that the model predicted fewer and fewer boxes during training and that predicted boxes were often wrongly labeled as being in the class "dog". After inspecting the labeled training data, we observed that the class "dog" appeared 8341 times, almost twice as frequently as the next most common class "person", which appeared 4331 times.

To troubleshoot, we experimented by retraining the backbone using the same batch size and number of epochs with different learning rates (0.1, 0.01, 0.0001, and 0.00001) but found the training loss stalled around 45. We then switched from using stochastic gradient descent as our backbone optimizer to using ADAM and saw our training loss lower from around 45 to 38.

At this stage, we also decided to set the number of trainable backbone layers during the supervised stage to zero. We found that this had no noticeable effect on the trained model's performance but did speed up training significantly.

Having not achieved enough of an improvement in model precision with these changes, we experimented with a frozen ResNet50 backbone that had been pre-trained on ImageNet to investigate whether the issue was in the backbone or the object-detection networks. With our custom self-supervised component being out-of-the-box, we found that our average precision on the training set climbed to 0.13, and our average recall climbed to 0.262 after just 3 epochs of training.

Following this, we switched to Facebook Research's implementation for VICReg training (Bardes et al., 2021) and customized it to train our backbone from scratch with our unlabeled dataset. Using this new implementation and batch size of 256, we were able to reach a training loss of around 20 for VICReg loss. With this, we were able to train Faster R-CNN models with reasonable outputs—but still noted problems with our precision.

Following this, we turned to optimize our Faster R-CNN model structure. Upon further investigation, we decided to implement a feature pyramid network (Lin et al., 2016) on top of our backbone in the Faster R-CNN model, which was shown to improve performance by allowing the model to use early feature maps alongside the final features in detecting objects.

We then considered that the anchor boxes for the model might not be representative of the sized objects that were in the labeled training images. To investigate this, we created histograms of the length and width of the training objects (See Fig. 1). To better account for this, we customized our anchor boxes to match the distribution of the labeled boxes. After examining the resulting bounding box predictions and the convergence of bath errors on the training set, it was clear that our new anchors improved box placement and size predictions.

Lastly, we changed the optimizer of the supervised component to ADAM with a learning rate of 0.001 and implemented a learning rate scheduler that varied the learning rate between 0.001 and 0.00001 between epochs. We trained the model for 7 epochs (after which training loss had started to saturate) and were able to achieve an average precision of 0.051 on the validation set and an average recall of 0.231.

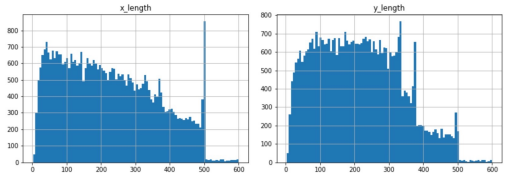


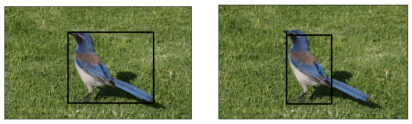
Figure 1. Length and width distribution of anchor boxes

5. Model Analysis

We now compare the outputs of our final model with the ground truth denoted by the labeled datasets to qualitatively demonstrate the quality of our results and inspect the feature maps at different levels of our backbone model to see the nature of the internal representations.

5.1. Model Behavior In Successful Cases

As an example of a close-to-correct output, the following results can be seen:

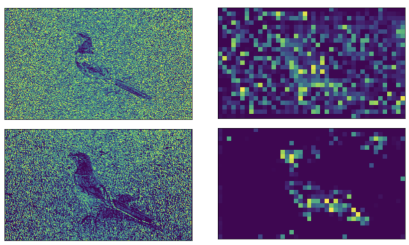


(a) Ground Truth (b) Model Output

As can be seen in figures (a) and (b), the model successfully detected the object. In this case, the predicted class was also correct. However, the outputted bounding box can be seen to be smaller than the one presented as ground truth, leaving small extensions of the object at hand (In this case, the tail of the bird) outside the predicted bounding box.

In our experiments, we have noted that this kind of result is really common for input images containing single instances of objects commonly found in the training dataset.

To see how the internal representations learned by our self-supervised section might have been used by the model, the following selected feature maps from layer 1 and layer 4 of the ResNet50 backbone can be seen:



(c) Features at Backbone Layer 1 (d) Features at Backbone Layer 4

As can be see seen, the features captured at Layer 1 (figure (c)) are close to how we would describe the generic orientation of the surfaces and likely signify basic features related to line patterns and curvature.

On the other hand, the Layer 4 (figure (d)) output demonstrates how some feature maps might contain information about the existence of an object directly. Looking at the feature map presented at the bottom of this figure, we can clearly see that the related internal representation is triggered by the existence of a bird in a particular region—likely as a result of high-level features present in bird pictures at these pixel locations.

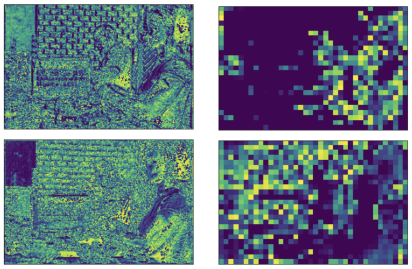
5.2. Shortcomings

Now, we demonstrate an example in which the model fails to detect some objects with the following result:

(e) Ground Truth (f) Model Output

Figures (e) and (f) demonstrate a case in which the model was not able to detect two objects of the same class (humans) when they were spatially really close. This behavior was seen consistently in our results.

To see the potential causes of this phenomenon, the following selected feature maps from layer 1 and layer 4 of the ResNet50 backbone can be seen:



(g) Features at Backbone Layer 1 (h) Features at Backbone Layer 4

In figure (h), it can be seen that while there are distinguishable activations in the feature maps showing high-level features of the underlying objects, the activations around the two people in the image are almost blended together. This likely is a cause of the model's inability to detect both humans separately.

This phenomenon was common during our inspections and is the largest known shortcoming of our implemented model.

6. Conclusion

Through this project, we demonstrated the effectiveness of the semi-supervised object detection, using a ResNet50 backbone trained using VICReg alongside a Faster R-CNN with a feature pyramid network. Although limited by the use of only 8 epochs of training due to time constraints, the model showed promising results with its ability to detect objects and displayed signs of ongoing learning.

References

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

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.

Girshick, R. Fast r-cnn, 2015. URL https://arxiv.org/abs/1504.08083.

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.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection, 2016. URL https://arxiv.org/abs/1612.03144.

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.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning, 2019. URL https://doi.org/10.1186/s40537-019-0197-0.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction, 2021. URL https://arxiv.org/abs/2103.03230.