Further Constrastive Training on Domain Specific Data

Valentin Villecroze
Department of Computer Science
University of Toronto
vvillecroze@cs.toronto.edu

William Ngo
Department of Computer Science
University of Toronto
wingo@cs.toronto.edu

Abstract

In computer vision, labelled data for downstream tasks (e.g. object detection) is often expensive to gather. As such, self-supervised methods such as contrastive learning are often used to pretrain the model backbone on large, general datasets such as ImageNet. In this work, we evaluate the gain in performances from further contrastive pretraining on domain specific data in an object detection task. We show that given enough data, additional contrastive training on domain specific data is helpful in improving performance and can even beat out supervised baselines. Code is available on GitHub.[1]

1 Introduction

Learning representations in a domain with limited labelled data is a common issue that arises when training neural networks for downstream tasks. One way to circumvent this problem is to use unsupervised or self-supervised methods to leverage the massive amounts of unlabelled data that are often available. Contrastive learning is one such paradigm that aims to learn rich representations in an unsupervised fashion by comparing among different samples. Samples that are similar (different views of the same object for example) are mapped closer in embedding space compared to dissimilar ones.

Typically, contrastive networks are trained using a large and general dataset such as ImageNet [1]. Once the networks have learned efficient representations, they can be used as backbone in downstream tasks such as instance segmentation or object detection. Recent work [2, 3] showed that pretraining a ResNet [4] backbone in a contrastive manner yields even better results (higher mean average precision) than pretraining in a supervised fashion. This shows that the representations learned through contrastive learning hold more relevant information for downstream tasks.

We analyze if given a pretrained contrastive network, would additional unsupervised pretraining help in the downstream task of object detection. As contrastive network are typically trained on a general dataset, we would like to see if training it further on a dataset similar to the target domain would enrich the learned representations. This is inline with the common problem in real-life applications where a large dataset often exists, but only a subset of the data is labelled.

2 Related work

2.1 Object Detection

Large advances in computer vision have rapidly improved object detection frameworks. In object detection, these advances have been driven by intuitive frameworks such as Faster R-CNN [5] and

[1] https://github.com/ngowilliam1/more-contrastive

Preprint. Under review.

Fully Convolutional Network (FCN) [6]. Two-staged detectors object detectors are driven by the idea of initially placing anchors of various sizes, performing classification and refining the anchors by recomputing features for each potential box. One staged detectors such as YOLO [7] are similar but, instead of refining the bounding box output, they directly output the bounding box, yielding in a slight loss in accuracy for a gain in speed compared to two staged detectors.

Most object detection frameworks require a convolutional backbone such as ResNet to operate. One upgrade to the vanilla ResNet is to use a Feature Pyramid Network (FPN) [8]. FPN augments a ResNet with lateral and top-down connections constructing a multi-scale feature pyramid from a single resolution input. Effectively, region of interest features from different levels of the feature pyramid are extracted and thus can be used to detect objects at different scales.

2.2 Contrastive Representation Learning

Modern contrastive visual learning frameworks are based on the idea that to learn an effective representation, different views of the same image are required [9]. These frameworks are typically designed to learn a metric space in which two views which stem from the same image are mapped closer in metric space compared to two different images.

One such method to construct this metric space is to to minimize the NCELoss [10]

$$\mathcal{L}_{\mathrm{NCE}} = -\mathbb{E}\left[\log \frac{e^{h(\mathbf{v}_1,\mathbf{v}_2)}}{\sum_{j=1}^{K} e^{h(\mathbf{v}_1,\mathbf{w}_j)}}\right]$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ are views sampled from the same image while the $\mathbf{w}_j$ are distractors sampled from other images. The score function h is usually the distance between the respective embeddings of the two images.

The current best performing framework [2] expands on this idea of using multiple views and shows that the best views to use are the ones which minimally provide the network enough information to associate two views of the same image. They argue that the best views are generated when the mutual information between the views is neither too large nor too low (U-shaped dependency). As they are generating multiple views based off a single image, labelled data is not required to perform contrastive learning in this type of framework.

3 Methodology

To perform object detection, we selected Faster-RCNN with ResNet-FPN backbone as it offers a good speed/accuracy tradeoff and can be trained using a single GPU. Since our project will not have access to a large GPU cluster, being able to train on a single GPU is essential to the project's success. One staged detectors were not considered as we did not require real-time detections.

The InfoMin [2] contrastive learning framework will be used to pretrain the ResNet-FPN backbone which will feeds into Faster-RCNN. InfoMin was chosen as it achieves state-of-the-art accuracy on unsupervised pre-training for ImageNet classification and the weights of their pre-trained network are freely available on their repository.

In this project we propose the following:

1. Starting from the contrastively pretrained ResNet-FPN weights from [2], we train Faster-RCNN on the Kitti [11] dataset ($\sim$ 15,000 images). This will establish our baseline performance.

2. Starting from the contrastively pretrained ResNet-FPN weights from [2], we first further pretrain in a contrastive manner the ResNet-FPN network on the nuImages dataset [12] ($\sim$ 90,000 images) which is similar to our target dataset. Once that is complete, we will train Faster-RCNN on the Kitti [11] dataset. As the ResNet-FPN will be trained in a contrastive manner, we do not use the labels from the nuImages dataset.

4 Experiments

We use the typical IoU-based mAP to evaluate our experiments. For the domain-specific contrastive pretraining, we train on the full NuImages dataset for 100 epochs starting from ResNet-FPN InfoMin weights made available from [2]. To train Faster-RCNN on Kitti, we split the labeled dataset in a 80/20 train and test split. Our reported results cannot be compared to the performance on the Kitti leaderboard as their test data is composed of unlabeled data, that can only be evaluated once uploaded to the leaderboard website. We train Faster-RCNN for 20 epochs, with a learning rate of 0.01 which is decreased by a factor of 10 at epoch 12, with a batch size of 6. We use a weight decay of 0.0005 and momentum of 0.9. For contrastive retraining, we train for 100 epochs with a learning rate of 0.003 with cosine annealing [13], and a batch size of 32.

4.1 Baselines and Domain Specific Contrastive Pretraining

We first trained a baseline from scratch on the Kitti dataset, without using any pretrained backbone. This model performed the worst out of all our experiments and confirms the usefulness of using pretrained weights for downstream tasks such as object detection.

A more relevant baseline is the model that uses the contrastively pretrained ResNet-FPN weights, and is then fine-tuned on the Kitti dataset. Two sets of weights are available from [2], respectively trained for 200 and 800 epochs on ImageNet. They are denoted InfoMin 200 and InfoMin 800 in subsequent parts. Additionally, the Supervised ResNet-FPN backbone trained on ImageNet is another baseline to measure the usefulness of contrastive learning.

Pretrained weights	Contrastive retraining	mAP
None	No	35.3115
Supervised	No	**62.4890**
InfoMin 200	No	60.9840
InfoMin 200	Yes ($\sim$ 90k images)	**61.9534** (+0.9694)
InfoMin 800	No	58.1231
InfoMin 800	Yes ($\sim$ 90k images)	**62.7578** (+4.6347)

Table 1: Results of our baselines and retrained models on Kitti dataset.

Pretrained weights	Contrastive retraining	Van AP	Car AP	Truck AP	Tram AP	Human AP	Cyclist AP	Human sitting AP
Supervised	No	74.741	75.571	82.237	67.27	43.655	63.986	**29.963**
InfoMin 200	No	74.192	74.97	80.043	**68.021**	41.897	61.446	26.317
InfoMin 200	Yes ($\sim$ 90k images)	75.351	75.931	82.43	66.579	43.793	64.554	25.038
InfoMin 800	No	70.093	73.673	78.094	63.163	39.969	58.676	23.195
InfoMin 800	Yes ($\sim$ 90k images)	**76.009**	**75.829**	**84.234**	67.271	**43.955**	**64.013**	27.994

Table 2: Per Class Results

As shown in Table 1, we observe that additional domain specific pretraining does indeed help performance as we can see an increase in mAP performance compared to the InfoMin weights trained on ImageNet. We observe that the increase in performance is not due to training for additional epochs as InfoMin 800 without contrastive retraining performs worse than InfoMin 200 without contrastive retraining. Potentially the network was overfitting to ImageNet when trained for 800 epochs.

Additionally, we can observe in Table 2 that for each model, it is the person and person sitting classes that are downweighting the results as mAP is simply an arithmetic mean of the per-class AP. One such method to improve performance is to merge both classes, which is done in the Kitti benchmark leaderboard.

4.2 Varying Contrastive Pretraining Dataset Size

We have also tested varying the dataset size used in the contrastive retraining phase, to see how much additional data was needed in order to improve performances. The results of the models pretrained

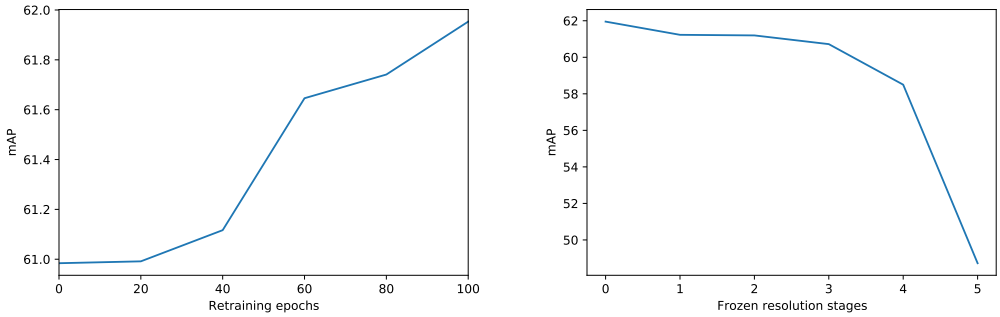
on subsets of varying sizes of the nuImages dataset can be seen Table 3. We observe a gain in performances even for relatively small datasets ($\sim$ 18k images), but if the dataset is too small, then performance degrades ($\sim$ 5k images).

Pretrained weights	Contrastive retraining dataset size	mAP
InfoMin 200	None	60.9840
InfoMin 200	$\sim$ 5k images	60.3341 (-0.6499)
InfoMin 200	$\sim$ 18k images	61.4438 (+0.4598)
InfoMin 200	$\sim$ 35k images	61.5339 (+0.5499)
InfoMin 200	$\sim$ 90k images	**61.9534** (+0.9694)

Table 3: Performances on the Kitti dataset depending on the size of the dataset used for contrastive retraining.

4.3 Number of Iterations for Contrastive Retraining

We have also evaluated changing the number of iterations for contrastive retraining. From Fig. 1a, we can observe that the backbone did not overfit to nuImages and that there is potential to train the network for more iterations. Due to computational restraints, we did not train the backbone for more iterations.



(a) Performances of InfoMin 200 depending on the number of contrastive retraining epochs.

(b) Performances of InfoMin 200 depending on the number of frozen resolution stages during fine-tuning.

4.4 Freezing Different Amount of Layers

[14] showed that depending on the dataset, it may be advantageous to keep some earlier layers fixed and only fine-tune deeper layers of the network instead of fine-tuning all the layers of the CNN. This is motivated by the idea that the earlier features of a CNN contain more generic features like edge detectors that could be useful to different tasks, but further layers of the CNN become progressively more specific to the source task and dataset. As InfoMin constructs a ResNet-FPN backbone, the network can be frozen at 5 different levels or stages of the network. The first is a convolution, and the following ones are each a group of residual blocks. We can see on Fig. 1b that fine-tuning all layers actually performs better in our case.

5 Conclusion

In summary, we showed that additional contrastive training on domain specific data is helpful in improving performance and can even beat out supervised baselines. Future works include further analysis of how similar in distribution the dataset used to perform contrastive training and the dataset used to train the detector is required to perform better than supervised baselines

References

[1] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[2] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?," 2020.

[3] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," *CoRR*, vol. abs/1911.05722, 2019.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2017.

[9] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," 2020.

[10] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[13] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," 2017.

[14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," 2014.

6 Appendix

6.1 Dataset samples

Figure 2: Samples from Kitti 2012 Object Detection Dataset

Figure 3: Samples from NuImages Dataset (Front Camera)

Figure 4: Samples from NuImages Dataset (Back Camera)