Object Recognition and Computer Vision: Project proposal

Topic D - Self-Supervised Learning of Visual Representations

Julien Gaubil, Manh-Dan Pham, M2 MVA

Introduction

In this project we study recent approaches of Self-Supervised Learning of Visual Representations with a focus on two recent ones, SimCLR [2] and VICReg [1]. After briefly reviewing the literature and describing these two approaches more thoroughly, we conduct experiments on VICReg to investigate its properties.

1. Related work

1.1. Self-Supervised Learning (SSL) of Visual Representations & Collapse

Self-supervised Learning of Visual Representations. These approaches consist in designing a task called *proxy task* for which a supervision is available while on a dataset without labels. Training for this task must enable to learn meaningful representations. First examples of proxy tasks designed were mostly geometric, for instance recovering the position of a patch relative to another [5] or predicting the degree of rotation applied to an image [6].

Recent approaches enabled rapid progress of SSL of Visual Representations, closing the gap with supervised approaches. Among these approaches are Contrastive Learning methods that aim at learning representations that are close for similar images and far away for dissimilar images. Such methods often rely on a contrastive loss [11] that ensures this behaviour and a two-branches architecture, sometimes siamese [3]. Another class of recent approaches is formed by Information Maximization methods that aim at maximizing the information encoded in the learned embedding space [7].

Collapse. A recurrent problem in Self-Supervised Representation Learning is the *collapse* of the representations in which the model outputs a constant representation, ignoring the input. Some solutions exist, for instance the use a momentum encoder and a memory bank [8] along with stop-gradient operations. The underlying reasons for the success of the existing techniques to prevent collapse are nevertheless not yet clearly understood.

1.2. SimCLR & VICReg

SimCLR. SimCLR [2] is a contrastive learning framework for SSL of visual representations. The main contribution of the authors lies in a thorough study of the components that enable good performances. They conclude to the usefulness of composing data augmentations (random cropping + color distorsion), as well as the interest of not applying directly the contrastive loss in the embedding space. They argue that projecting the representations with a non-linear function (MLP) enables to maintain information that would otherwise be lost due to the invariance to transformations (rotation, color...).

It leads to a simple framework, represented Figure 1 that does not require a memory bank.

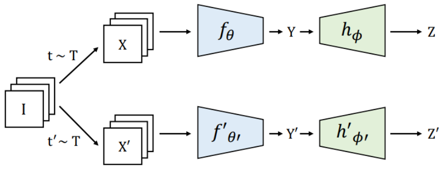

Figure 1. Model architecture used for SimCLR and VICReg. Source [1]

The framework can be divided into three steps. First, two different data augmentations t and t' are applied to a given image, yielding two augmented images called a positive pair. Then, a shared neural network encoder ($f_\theta = f'_{\theta'}$ in Figure 1), such as a ResNet [9], encodes representation vectors from the augmented images. Finally, a small neural network projects the two representation vectors to a smaller vector space where the contrastive loss (1) is applied on the two branches Z and Z'.

For a batch of N examples, among the $2N$ examples derived from data augmentation, given a positive pair, the other $2(N-1)$ augmented images are treated as negative examples. Let $sim(u,v) = \frac{u^T v}{\|u\|\|v\|}$ be the cosine similarity. The contrastive loss for a positive pair (i,j) is then defined as:

$$l_{i,j} = -log\left(\frac{exp(sim(z_i,z_j)/\tau}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} exp(sim(z_i,z_k)/\tau)}\right) \quad (1)$$

where τ is a temperature parameter. This loss aims at embedding the representation vectors of the positive pair close to each other while moving them away from the embeddings of the other images in the batch.

As the contrastive prediction task is defined at the batch level, the batch size is a crucial parameter for training the model. The authors show quantitatively that the bigger the batch size is, the better the performances are. This can be explained by the fact that bigger batch sizes allow to have more negative examples within the same batch, thus to get better separated representation vectors.

VICReg. VICReg [1] is an information maximization method that aims at explicitly prevent collapse. It uses a simple two-branches architecture (cf fig. 1) for learning visual representations in a self-supervised way, and doesn't rely on limiting techniques such as memory banks or large batch sizes. Its main contribution is the introduction of a loss composed of three regularization terms that explicitly prevent collapse.

The loss is defined on a batch of $2n$ projected representations $(z_i, z'_i)_{1\leq i\leq n} \in \mathbb{R}^d$. The first term s is an invariance term that is applied to both branches Z, Z' and simply consists in a MSE loss. The two other terms are applied separately on each branch. The covariance regularization term c aims at decorrelating the dimensions of the learned embedding space so that it encodes different information. To do so, it regularizes the off-diagonal coefficients of the covariance matrix C computed over the n representations of the branch.

The variance regularization term v explicitly prevents collapse by ensuring that the variance over the batch of each component of the representations is superior to a fixed threshold γ. This ensures that the model doesn't output a constant representation. It is defined as the hinge loss applied to the s.t.d. of the components $z^k \in \mathbb{R}^n$ of the representations over the branch Z:

$$\text{Invariance term:} \quad s(Z, Z') = \frac{1}{n} \sum_{i=1}^{n} \|z_i - z'_i\|_2^2$$

$$\text{Covariance term:} \quad c(Z) = \sum_{i,j=1, i \neq j}^{n} [C(Z)]_{i,j}{}^2$$

$$\text{Variance term:} \quad v(Z) = \frac{1}{d} \sum_{k=1}^{d} max \left(0, \gamma - \sqrt{Var(z^k) + \epsilon}\right)$$

where ϵ is a small numerical stability factor. The final loss l is a combination of the three terms, the two latter being applied symmetrically to both branches:

$$l = \lambda \, s(Z, Z') + \nu \, (c(Z) + c(Z')) + \mu \, (v(Z) + v(Z')) \quad (2)$$

The main advantage of VICReg lies in application of the last two terms of the loss separately to are applied separately to each branch while explicitly preventing collapse. This indeed enables to use separate architectures for encoders f_θ and $f_{\theta'}$, or even different modalities as input. It paves the way to its use in a wide variety of tasks.

Another advantage of VICReg is that it doesn't require techniques such as memory banks or large batch sizes that have heavy memory footprints. Authors indeed show the stability of the results for a wide range of batch sizes while memory banks are not useful, negative samples being replaced by the variance term.

2. Experiments

2.1. Experimental setup & Datasets

Experimental setup. We conduct experiments by training encoders with VICReg, using the code from the official VICReg codebase [1]. We also use pre-trained weights on ImageNet-1000 available in the codebase, in which case it will be specified.

We evaluate the backbones (i.e. everything excepted the projection heads) on classification tasks. To do so, we perform linear evaluation by training a linear classification layer on top of the frozen trained backbones.

As for the architecture, we use ResNet-50 [9] as a shared backbone for both branches and a 1-hidden layer MLP (with Batch Normalization and ReLu activation for the two first layers) as the projection head. Following the experiments in VICReg [1], the size of each layer of the projection head is fixed to 8192. The training protocol is the same as the one described in the paper. Unless specified, we use the same data augmentation pipeline that sequentially performs Random Cropping then resizing to size 224 × 224, then randomly applied Horizontal flipping, color jitter, converting to grayscale, Gaussian blurring, solarization and normalization. Following the ablations of the original paper, we set the coefficient of the invariance term to $\lambda = 25$, the coefficient of the covariance term to $\nu = 1$ and the coefficient of the variance term to $\mu = 25$.

We train and evaluate our models on four Nvidia V100 GPUs, and always train the linear classifier for 100 epochs with a batch size of 256 on 10% of the training set during each evaluation.

Datasets. We first conduct experiments on ImageNet-100 that is a subset of the classification dataset ImageNet-1000 [4]. We use the version available on Kaggle that consists in 10% of the original dataset with 100 random classes and 135000 color images.

We then evaluate the generalization power of a backbone trained with VICReg on other datasets, namely CIFAR-10 and CIFAR-100 [10]. These classification datasets are composed of 60000 32 × 32 color images divided respectively in 10 and 100 classes.

We finally evaluate the quality of the representations learned with VICReg when training on a fine-grained dataset that is CUB-200-2011 [13]. It consists in 11788 color images divided in 200 bird categories.

2.2. Experiments on a smaller dataset (*Julien*)

We first evaluate the capacity of a model trained with VICReg to learn meaningful representations from fewer data. We train each model during 1000 epochs on ImageNet-100, except a pre-trained model on ImageNet-1000 that we borrow from the codebase of VICReg for comparison. We reproduce several experiments from the paper by varying the batch size and verifying the influence of the variance term. Results are presented table 1:

Model	Top-1 Acc	Top-5 Acc
BS=256	75.4	93.2
BS=512	75.9	92.9
BS=1024	76.6	93.4
$\mu = 0$	1.0	5.0
ImageNet-1000	**81.2**	**96.0**

Table 1. Top-1 and Top-5 accuracies (in %) on Imagenet100 test set

Our experiments show the same stability with respect to the batch size as presented in the original paper, the maximal difference between the three experiments in Top-1 and Top-5 accuracy respectively being of 1.2 and 0.5%. This shows that VICReg indeed performs well with reduced batch sizes even on smaller datasets. Still, the model pre-trained on ImageNet-1000 with a batch size of 2048 yields the best results by a significant margin, around 5% in Top-1 accuracy and around 2.5% in Top-5 accuracy. This suggests that the model benefits from being trained on larger datasets to learn better representations, even though most of the classes that appear in ImageNet-1000 are not in ImageNet-100.

We then verify that the variance regularization term indeed prevents collapse by removing it, setting its coefficient μ to 0. The classification results indeed suggest that the model outputs a constant representation and the classifier selects the same class every time. We verify this by performing visualizations using t-SNE algorithm [12] on representations encoded on the test set. For clarity

[1] https://github.com/facebookresearch/vicreg

purposes, we restrict ourselves to 5 classes randomly selected, the results are presented in Figure 2:

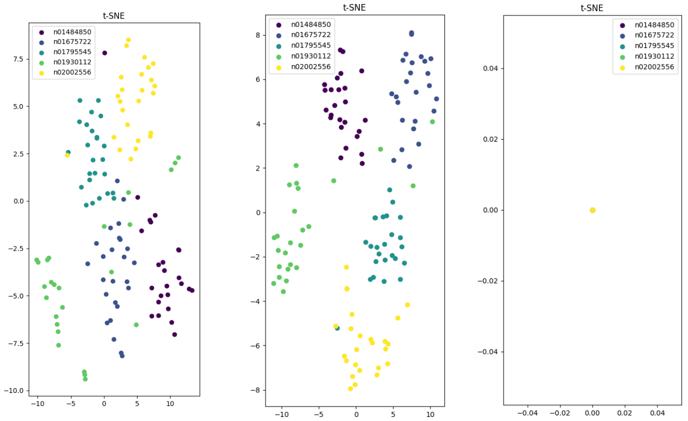

Figure 2. t-SNE visualizations on ImageNet-100 test set for VICReg models: BS=1024 (left), ImageNet-1000 (center) and $\mu = 0$ (right)

The visualization clearly demonstrates that removing the variance regularization term in Equation 2 leads to collapse. The visualization for the model pre-trained on ImageNet-1000 shows clusters that are more distinct compared to the model trained on ImageNet-100. This indicates that the representation for the model trained on ImageNet-1000 are both more dissimilar from the representations of images of different classes and closer to the representations of images of the same class. This suggests that the model trained on ImageNet-1000 captures more discriminative features for every class, leading to a better representation (especially for a classification downstream task).

2.3. Generalization evaluation (*Manh-Dan & Julien*)

In order to assess the generalization power of the representations produced by VICReg, we evaluate the model pretrained on ImageNet-1k (available in the codebase, denoted as ImageNet-1k frozen) on CIFAR-10 and CIFAR-100 datasets. We compare it to a model trained from scratch on these datasets with random weights initialization (the baseline in Tables 2 and 3) and another one finetuned on these datasets. We train models for 1000 epochs with a batch size of 256 on CIFAR.

Dataset Pretrain	Top-1 Acc	Top-5 Acc
Baseline	73.6	98.6
ImageNet-1k frozen	**90.4**	**99.8**
ImageNet-1k init	87.4	99.7

Table 2. Top-1 and Top-5 accuracies (in %) on CIFAR-10 test set

Dataset Pretrain	Top-1 Acc	Top-5 Acc
Baseline	52.6	80.3
ImageNet-1k frozen	**73.6**	**93.6**
ImageNet-1k init	67.9	90.9

Table 3. Top-1 and Top-5 accuracies (in %) on CIFAR-100 test set

For both datasets, the results show that the model pretrained on ImageNet-1k outperforms the finetuned model and the model trained from scratch. Different factors can explain the better performances of the model pretrained on ImageNet-1k. First of all, the average resolution of images from ImageNet is 469 × 387 whereas images from CIFAR have a resolution of 32 × 32 pixels. Thus, higher quality images may lead to higher quality features resulting in better classification accuracy. ImageNet-1k contains approximately 25 times more images than CIFAR, which has been observed to be useful in most unsupervised or self-supervised methods, in order to guide the training without labels.

A result that might be surprising is the fact that the model finetuned on CIFAR performs worse than the model only pretrained on ImageNet-1k. This may be due to the fact that finetuning on CIFAR does not compensate enough the deterioration of the quality of the features as explained in the previous paragraph. Indeed, as both ImageNet and CIFAR consist of natural images, their domains are similar and thus finetuning is not as beneficial as it would be with more different domains (such as medical imaging).

2.4. Fine-grained dataset (*Julien*)

We finally evaluate the capacity of a model trained with VICReg to learn meaningful representations when training on challenging fine-grained datasets. We train each model during 2000 epochs on CUB-200-2011 using a batch size of 1024, except for the model pretrained on ImageNet-1000 (denoted as VICReg-ImageNet-1000 frozen) that we borrow from the codebase of VICReg for comparison.

We investigate the influence of the scale of the crops performed. It is indeed the most important factor on CUB-200-2011 where images are not cropped and centered on the birds. We vary the lower bound of the crop scale from 0.08 (default) to 0.7, table 4:

Model	Top-1 Acc	Top-5 Acc
Crop Scale (0.08-1)	4.4	11.8
Crop scale (0.3-1)	**8.1**	18.7
Crop scale (0.5-1)	6.5	17.1
Crop scale (0.7-1)	7.6	**18.9**

Table 4. Top-1 and Top-5 accuracies (in %) on CUB-200-2011 test set

We first note that all the models trained from scratch on CUB-200-2011 don't perform well compared to the previous experiments. This is expected since CUB is much more challenging than than ImageNet, images from different bird categories being very similar. The default crop scale (0.08-1) probably leads to bad results because birds often take up a small located area on the image. A small crop may therefore completely miss the bird. There is nevertheless no clear relationship between the accuracy and the lower bound for the scale. In our next experiments, we keep the default scale (0.08-1) for fair comparison with the models pre-trained on ImageNet-1000 available in the codebase.

We then investigate how small variations in the transformation pipeline for data augmentations affect the results compared to the baseline (Crop Scale 0.08-1). We also compare with a model pre-trained on ImageNet-1000 from the code base (ImageNet-1000 frozen), and a model whose weights have been initialized with ImageNet-1000 frozen and trained on CUB (ImageNet-1000 init). The results are presented table 5:

The different transformations tested here did not enable to improve the baseline score. As previously, the model pre-trained

Model	Top-1 Acc	Top-5 Acc
Crop Scale (0.08-1)	4.4	11.8
W/o Color Jitter	2.6	8.4
W/o Gaussian Blur	2.4	8.9
Only Crop	2.4	8.3
Random ResNet	0.8	3.0
ImageNet-1000 init	17.5	39.5
ImageNet-1000 frozen	**20.9**	**42.4**

Table 5. Top-1 and Top-5 accuracies (in %) on CUB-200-2011 test set

on ImageNet-1000 outperforms the other models by a significant margin and still yields better results than the model fine-tuned on CUB-200-2011. This can be explained by the fact that CUB and ImageNet-1000 datasets could possibly overlap.

We then seek to exhibit differences in the training process between the trainings on CUB and ImageNet-100 that could explain the success of the latter. To do so, we track the evolution of the different terms of the VICReg loss (Eq.(2)) during the training. We observe a significant difference for only one of the terms that is the Covariance term. The results are presented in Figure 3.

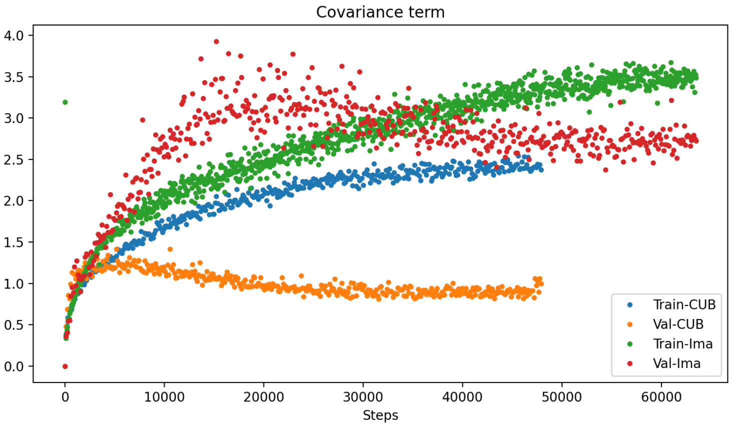

Figure 3. Evolution of the Covariance term in VICReg loss 2 during training on ImageNet-100 and CUB-200-2011

We first note that the Covariance term is the only that is consistently increasing on the training set during training, despite being a regularization term. Our interpretation is that the model learns features that are more and more accurate and specific to the training set during the process, structuring the learned embedding space in the same time. This structure and the relation between discriminative features may lead to redundant information being encoded in different dimensions, hence increasing the covariance term. Although the purpose of the covariance term is explicitly to avoid this behaviour while learning meaningful representations, this behaviour may therefore also be an indicator of a training process that learns to encode relevant representations.

The main difference between the two experiments lies in the the behaviour on the validation set: the covariance term on the validation term on CUB starts decreasing long before on ImageNet-100. This results in a relative difference between the covariance term on the validation and training set that is much higher on CUB than ImageNet (around 60% for CUB, 20% for ImageNet-100). This difference could be an indicator that the model doesn't learn to encode relevant representations on CUB.

We finally provide a visualization that highlights the lack of structure in the learned embedding space on CUB by performing visualizations using t-SNE algorithm on representations encoded on the test set (only 5 classes as previously), Figure 4:

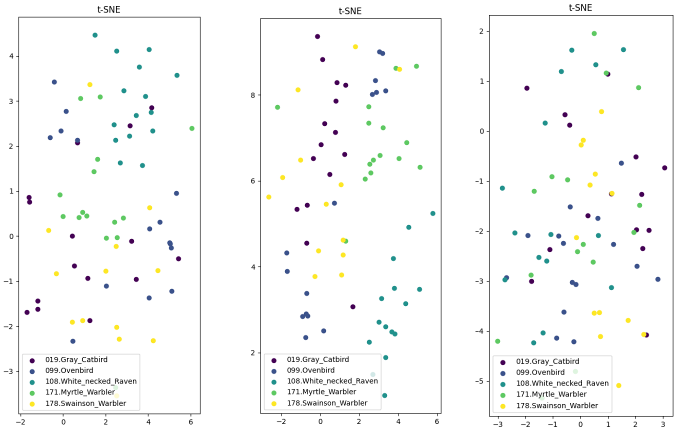

Figure 4. t-SNE visualizations on CUB-200-2011 test set for VICReg models: BS=1024 (left), ImageNet-1000 frozen (center) and a randomly initialized model not trained (right)

This visualization shows that none of the models have learnt to encode representations on CUB that are clearly distinct for different classes, therefore leading to bad results in classification. For some classes, identifiable clusters appear on the visualization for the model trained on ImageNet-1000 which is consistent with its results in classification evaluation, that are significantly better than the others.

Conclusion

In this work, we perform many experiments on VICReg [1] to highlight some of its properties. We showed that VICReg is still able to produce meaningful representations with few data. We also verified the authors claims about the role variance term used in their proposed loss (Eq.(2)) as well as its robustness to the batch size used during training. We also assessed the generalization power of VICReg by showing that the model pretrained on ImageNet-1k outperformed on CIFAR models trained from scratch or finetuned on this dataset. Finally, we showed that VICReg does not perform well on fine-grained classification task because of images being too similar which translates in too similar representations, as proven by the covariance term evolution.

References

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. *ICLR*, 2022. 1, 2, 4

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. 1

[3] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[5] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 1

[6] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 1

[7] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv preprint*, abs/2006.07733, 2020. 1

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. June 2020. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[11] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *Arxiv preprint*, abs/1807.03748, 2018. 1

[12] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 2

[13] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2