AIMS African Institute for Mathematical Sciences SENEGAL

AFRICAN MASTER'S IN MACHINE INTELLIGENCE (AMMI)

Foundations of ML/DL

Supervised by prof Moustapha Cisse

Self-Supervised Visual Representation learning : SimCLR, Deep Clustering

Authors:

Binta Sow	bsow@aimsammi.org
Armandine Sorel Kouyim Meli	askmeli@aimsammi.org
Honorine Gnonfin	hgnonfin@aimsammi.org
Mame Diarra Diop	mddiop@aimsammi.org
Samuael Adnew	sadnew@aimsammi.org
Uriel Nguefack Yefou	unyefou@aimsammi.org
Verlon Roel Mbingui	vrmbingui@aimsammi.org

April 19, 2023

Contents

Introduction

The development of machine learning applications has required manual annotation of data, often by experts. Unfortunately, not all domain of science is as data-rich as needed, and this limit algorithms in many applications. The lack of annotated data is one of the main challenges in implementing machine learning tasks. For instance, in healthcare, data is often costly to acquire and annotate and the amount of data is limited by the affected patient. To solve this problem, researchers are working on Self-Supervised Learning (SSL) techniques capable of capturing subtle nuances in data [1]. The reminder of this report is organised as follows, Self-Supervised Learning, SimCLR, deep clustering and some applications of SSL.

1 Self-Supervised Learning

1.1 Definition

A popular form of unsupervised learning, called ”self-supervised learning”, uses pretext tasks to replace the labels annotated by humans by ”pseudo-labels” directly computed from the raw input data. [2].

Self-supervised learning consist of two phases [3]:

1. **Pretraining**: In this part, we pretrain the model in a large amount of unlabeled data. The pretrain model learns representations that capture meaningful information about the data.

2. **Adaptation**: In this phase, the pretrain model is used for downstream tasks (predictions tasks) with limited labeled data (Few-Shot Learning) or even no labeled data(Zero-Shot Learning).

1.2 Benefits of self-supervised learning

1. **Improved AI capabilities**: Self-supervised learning is primarily used in computer vision for tasks such as colorization, 3D rotation, depth filling or context filling. While these tasks previously required labeled examples to build accurate models, self supervised learning can improve computer vision or speech recognition technologies by eliminating the need for examples.

2. **Scalability**: Supervised learning requires labeled data to predict the outcome of unknown data. However, it may need large data sets to build appropriate models and make accurate predictions. For large training datasets, manual labeling of data can be problematic. Self-supervised learning can automate this process and handle this task even with massive amount of data.

3. **Understanding how the human mind works**: Supervised models require human intervention to function properly. However, these interventions do not always exist. We

can then think of introducing reinforcement learning into the machines so that they start from the beginning in cases where they can get an immediate feedback without negative consequences.

1.3 Classes of Self-Supervised learning

They are many classes of self-supervised learning such that :

- Clustering pseudo-labeling : Deep Clustering, SwAV, DINO
- Consistency regularization : MixMatch, UDA, BYOL
- Contrastive learning : MoCo, SimCLR, CLIP

In this work, we will focus on **SimCLR** and **Deep Clustering**.

2 SimCLR: A simple framework for constrastive learning of visual representations

2.1 What is SimCLR?

In computer science, a framework can be defined as a set of tools and library which can be used to solve a specific problem. As example, SimCLR is a constrastive learning framework which learns representations by maximizing the agreement between different augmented views of the same data example via a constrastive loss in a latent space.

2.2 SimCLR Architecture

The SimCLR architecture is made of four important components.

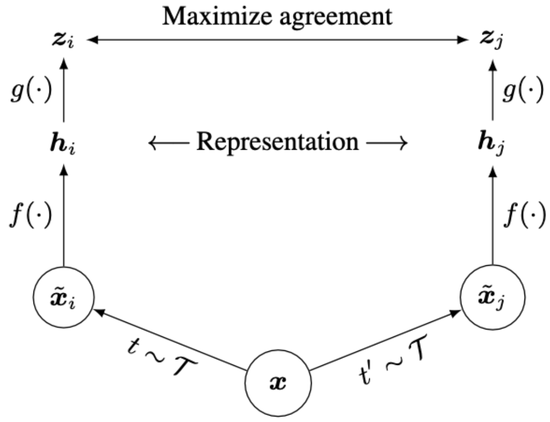


Figure 1: SimCLR Architecture [4]

1. **Data augmentation module** : It is the module which transforms a given data example in two related views of the same example. Data augmentation operations include: random cropping, random colors distortion, Gaussian blur, rotation, etc. In Figure 1, the two views of x after augmentation using operations t and t' are $\tilde{x_i}$ and $\tilde{x_j}$. If we have **N** training examples, after applying augmentation, we will have **2N** training examples.

2. **Neural network based encoder module** $f(\cdot)$: It extracts features that represent the input images. The SimCLR framework allows the choice of the neural network architecture based on an encoder module without any constraints. The most used neural network architecture is ResNet. The outputs after applying the encoder h_i and h_j where h_i=f($\tilde{x_i}$) and h_j=f($\tilde{x_j}$), where h_i and h_j are d-dimensional and are the outputs after average pooling.

3. **Prediction head encoder or neural network projection head** $g(\cdot)$: The role of this small neural network(Multi-Layer Perceptron with one hidden layer) is to map the representations into the space where we can apply the InfoNCE(Noise-Contrastive Estimation) loss. After applying the projection on the two representations, we end up with z_i and z_j where:
$z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i)$ and $z_j = g(h_j) = W^{(2)}\sigma(W^{(1)}h_j)$.

4. **A constrastive loss**: We define the constrative loss for constrastive predictive class.

$$L = -\mathbb{E}_X\left[\log \frac{\exp(s(f(x), f(x^+)))}{\exp(s(f(x), f(x^+))) + \sum_{j=1}^{N-1}\exp(s(f(x), f(x_j^-)))}\right], \tag{1}$$

where,

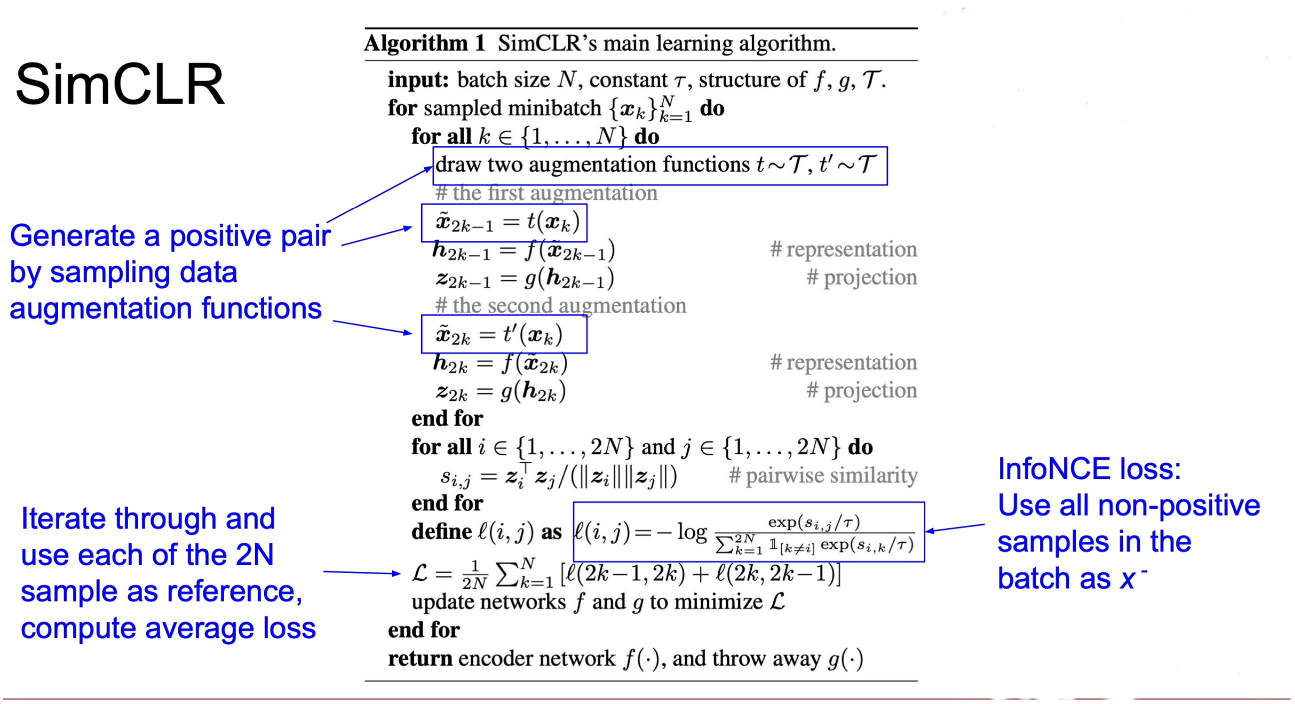
$\exp(s(f(x), f(x^+)))$: score for the positive pairs,

$\sum_{j=1}^{N-1}\exp(s(f(x), f(x_j^-)))$: score for the N-1 negative pairs, and

$$s(u, v) = \frac{u^T v}{\|u\|\|v\|}.$$

2.3 SimCLR's main learning algorithm

The SimCLR framework can be resumed by this algorithm:



SimCLR

Algorithm 1 SimCLR's main learning algorithm.

```
input: batch size $N$, constant $\tau$, structure of $f$, $g$, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ do
    for all $k \in \{1, \ldots, N\}$ do
        draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
        # the first augmentation
        $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
        $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$                # representation
        $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$                # projection
        # the second augmentation
        $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
        $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$                # representation
        $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$                # projection
    end for
    for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ do
        $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\|\|\boldsymbol{z}_j\|)$        # pairwise similarity
    end for
    define $\ell(i,j)$ as $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
    $\mathcal{L} = \frac{1}{2N}\sum_{k=1}^{N}[\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
    update networks $f$ and $g$ to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$
```

Generate a positive pair by sampling data augmentation functions

InfoNCE loss: Use all non-positive samples in the batch as x^-

Iterate through and use each of the 2N sample as reference, compute average loss

Figure 2: SimCLR Algorithm [4]

2.4 Implementation of SimCLR

Please find here the implementation of SimCLR.

3 Deep Clustering

Simple clustering is an unsupervised learning method that groups datapoints together based on their similarities. That method is unsuitable for large and high dimensional datasets since it can't identify complex patterns. An alternative way to deal with that is deep clustering which uses the neural network to identify complex and hidden patterns in the data more efficiently.

3.1 How does deep clustering works?

The input of the framework is a huge dataset(ImageNet) which is a dataset containing 1.3M images distributed into 1000 classes and then apply a Convnet on it to select the relevant features. We use the AlexNet architecture in the Convnet part. After that, the author performs clustering using one of the powerful clustering algorithm, K-Means using the selected features from the Convnet architecture. The next step is to do a classification using pseudo labels from clustering, and through the backpropagation we update the weights in the convolutional part and repeat the process. A convolutional structure gives a strong prior.

Let $\{x_1, ..., x_N\}$ a training set of N images.

Problem setup : Find the parameter θ^* such that the mapping f_{θ^*} produces good general purpose features(representation). [2]

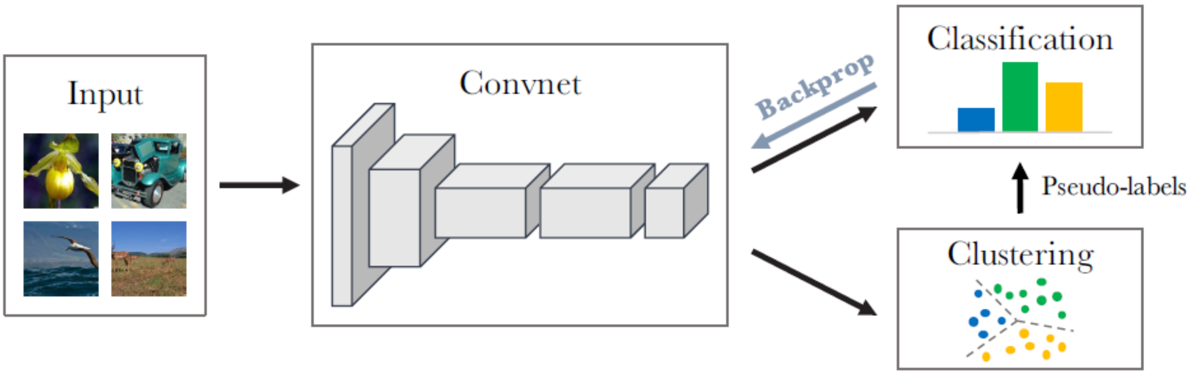


Figure 3: Illustration of DeepCluster process [2]

This approach summarized in Figure 3 consists in alternating between clustering of the image descriptors and updating the weights of the Convnet by predicting the cluster assignments. For the clustering part, the author uses K-means algorithm. The author uses it to cluster the features $f_\theta(x_n)$ into k distinct groups. The mathematics formulation is given by:

$$C \in \mathbb{R}^{d \times k} \longrightarrow \quad \text{centroid matrix}$$

$$y_n \longrightarrow \quad \text{cluster assignment of each image}$$

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^*} \| f_\theta(x_n) - C y_n \|_2^2 \tag{2}$$

such that $y_n^T \mathbf{1}_k = 1$

$$
\begin{aligned}
y_n^* \longrightarrow &\quad \text{optimal assignment (pseudo labels)} \\
C^* \longrightarrow &\quad \text{optimal centroid matrix (not used)}
\end{aligned}
$$

So, in Equation 2, we are doing an expectation minimization, where in the expectation part, we find the optimal centroid matrix and in the minimization part, we find the optimal assignment. On top of the Convnet architecture, we put a shallow neural network g_W, and the idea is t0 find the optimal parameters of θ and W which minimize the loss function given by Equation 3.

$g_W \longrightarrow$ parameterized classifier on top of the features $f_\theta(x_n)$

$$\min_{\theta,W} \frac{1}{N} \sum_{n=1}^{N} l(g_W(f_\theta(x_n)), y_n) \tag{3}$$

l is the negative log softmax.

3.2 The potential issues and their solutions

Like any method that jointly learns a discriminative classifier and the labels, we may have trivial solutions in deep clustering. Among those trivial solutions, we have empty clusters and trivial parametrization.

1. **Empty clusters**: Because of the absence of mechanisms to prevent from empty clusters, we may have during a clustering some clusters with no object while normally an optimal separation should assign all of the inputs to a single cluster [5]. To mitigate that problem, we automatically reassign points to empty clusters during the k-means optimization. More precisely, when a cluster becomes empty, the author randomly select a non-empty cluster and use its centroid with a small random perturbation as the new centroid for the empty cluster and then reassign the points belonging to the non-empty cluster to the two resulting clusters. [2]

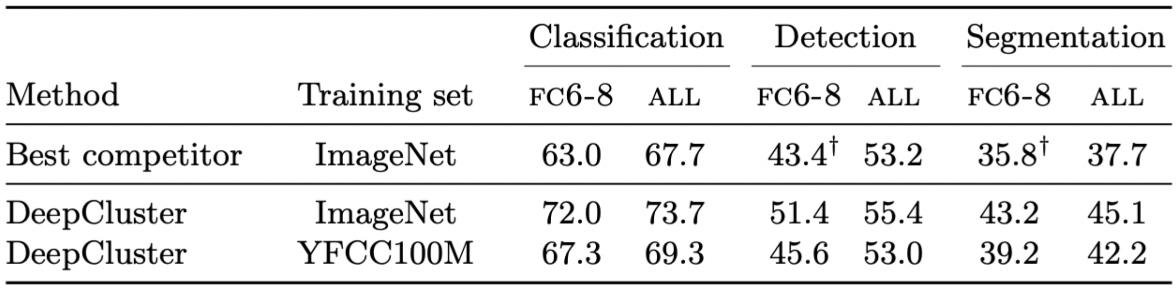
2. **Trivial parameterization**: When the number of images per class is highly unbalanced, we can meet a trivial parameterization. This means that the Convnet will be predicting the same output regardless of the input. A way to bypass this issue is to sample images based on a uniform distribution over the classes or pseudo-labels, in other words, to weight the contribution of an input to the loss function in Equation 2 by the inverse of the size of the points of his cluster [2].

3.3 Results

In Figure 4, the author compares Deepcluster with the best existing method on three downstream tasks like Classification, Detection and Segmentation. Regardless of the training set, DeepCluster outperforms the best published numbers on most tasks.

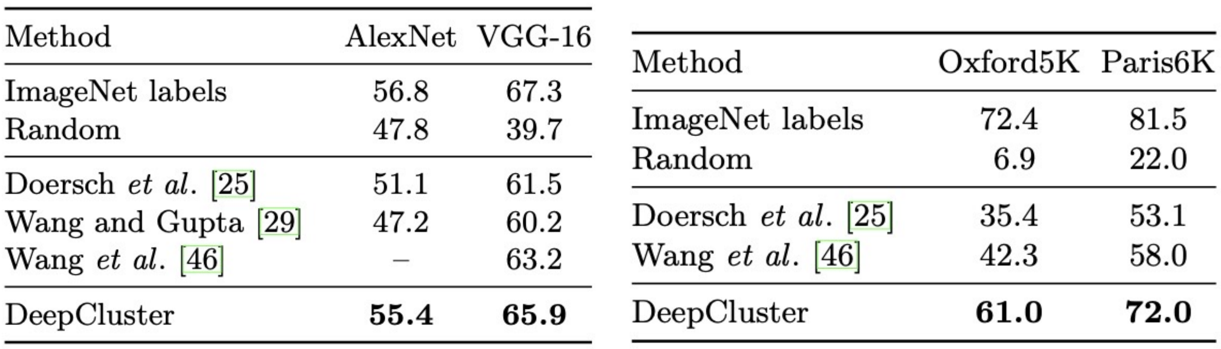
In Figure 5a, the author is comparing Deep cluster applying with two existing pre-trained models like AlexNet and VGG-16 with existing state of the art results, he shows that while using these pre-trained architecture, his model outperforms all previous work, and we get better result when using VGG-16.

In Figure 5b, the author uses DeepCluster on two datasets, i.e., Oxford Buildings [6] and Paris [7], he reports the performance of VGG-16 trained with different approaches obtained with Sobel filter.



		Classification		Detection		Segmentation	
Method	Training set	FC6-8	ALL	FC6-8	ALL	FC6-8	ALL
Best competitor	ImageNet	63.0	67.7	$43.4^\dagger$	53.2	$35.8^\dagger$	37.7
DeepCluster	ImageNet	72.0	73.7	51.4	55.4	43.2	45.1
DeepCluster	YFCC100M	67.3	69.3	45.6	53.0	39.2	42.2

Figure 4: Comparison of Deepcluster with the best competitor on several downstream tasks.



Method	AlexNet	VGG-16
ImageNet labels	56.8	67.3
Random	47.8	39.7
Doersch *et al.* [25]	51.1	61.5
Wang and Gupta [29]	47.2	60.2
Wang *et al.* [46]	–	63.2
DeepCluster	**55.4**	**65.9**

(a) Pascal VOC 2007 object detection with AlexNet and VGG- 16.

Method	Oxford5K	Paris6K
ImageNet labels	72.4	81.5
Random	6.9	22.0
Doersch *et al.* [25]	35.4	53.1
Wang *et al.* [46]	42.3	58.0
DeepCluster	**61.0**	**72.0**

(b) mAP on instance-level image retrieval on Oxford and Paris dataset with a VGG-16

Figure 5: Some other interesting performance of DeepCluster apply with AlexNet and VGG-16 on other datasets [8][9] [10]

4 Some applications of Self-Supervised Learning

- **Computer Vision**: Self-supervised learning has been used extensively in computer vision tasks such as image and video analysis, object detection, segmentation, and tracking.
- **Naturel Language Processing**: Self-supervised learning has also been applied to natural language processing tasks such as language modeling, sentiment analysis, and machine translation.
- **Robotics**: Self-supervised learning has been used in robotics applications such as robot navigation, manipulation, and control.
- **Healthcare**: Self-supervised learning has been applied to healthcare applications such as medical image analysis, patient diagnosis, and drug discovery.

Conclusion

In this work, we talked about self supervised learning which is a kind of unsupervised learning where we build models with unlabelled data. There are many types of self supervised learning algorithms but we have just focused on SimCLR and Deep Clustering. Despite some limitations, those algorithms are very useful in solving problem in different areas such as healthcare, robotics, natural language processing, etc...

References

[1] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, pages 1–7, 2022.

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[3] Andrew Ng. Cs229 lecture notes. *CS229 Lecture notes*, 1(1):1–3, 2000.

[4] 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.

[5] Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans. Maximum margin clustering. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

[6] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[7] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. pages 1 – 8, 07 2008.

[8] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. 08 2017.

[9] Carl Doersch, Abhinav Gupta, and Alexei Efros. Unsupervised visual representation learning by context prediction. 05 2015.

[10] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos, 2015.