

Self-Supervised Classification Network

Elad Amrani[1,3(✉)], Leonid Karlinsky[2], and Alex Bronstein[3]

[1] IBM Research-AI, Haifa, Israel
[2] MIT-IBM Watson AI Lab, Cambridge, US
[3] Technion, Haifa, Israel
elad.amrani@ibm.com

Abstract. We present Self-Classifier – a novel self-supervised end-to-end classification learning approach. Self-Classifier learns labels and representations simultaneously in a single-stage end-to-end manner by optimizing for same-class prediction of two augmented views of the same sample. To guarantee non-degenerate solutions (i.e., solutions where all labels are assigned to the same class) we propose a mathematically motivated variant of the cross-entropy loss that has a uniform prior asserted on the predicted labels. In our theoretical analysis, we prove that degenerate solutions are not in the set of optimal solutions of our approach. Self-Classifier is simple to implement and scalable. Unlike other popular unsupervised classification and contrastive representation learning approaches, it does not require any form of pre-training, expectation-maximization, pseudo-labeling, external clustering, a second network, stop-gradient operation, or negative pairs. Despite its simplicity, our approach sets a new state of the art for unsupervised classification of ImageNet; and even achieves comparable to state-of-the-art results for unsupervised representation learning. Code is available at https://github.com/elad-amrani/self-classifier.

Keywords: Self-supervised classification · Representation learning

1 Introduction

Self-supervised visual representation learning has gained increasing interest over the past few years [2,5–7,10,18,23,29]. The main idea is to define and solve a pretext task such that semantically meaningful representations can be learned without any human-annotated labels. The learned representations are later transferred to downstream tasks, e.g., by fine-tuning on a smaller dataset. Current state-of-the-art self-supervised models are based on contrastive learning (Sect. 2.1). These models maximize the similarity between two different augmentations of the same image while simultaneously minimizing the similarity between different images, subject to different conditions. Although they attain impressive

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19821-2_7.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
S. Avidan et al. (Eds.): ECCV 2022, LNCS 13691, pp. 116–132, 2022.
https://doi.org/10.1007/978-3-031-19821-2_7

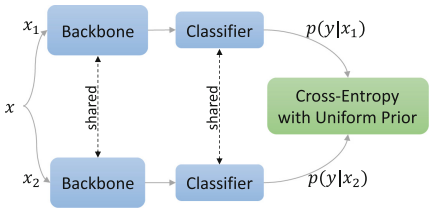


Fig. 1. ***Self-Classifier*** **architecture**. Two augmented views of the same image are processed by a shared network comprised of a backbone (e.g. CNN) and a classifier (e.g. projection MLP + linear classification head). The cross-entropy of the two views is minimized to promote same class prediction while avoiding degenerate solutions by asserting a uniform prior on class predictions. The resulting model learns representations and discovers the underlying classes in a single-stage end-to-end unsupervised manner.

overall performance, for some downstream tasks, such as unsupervised classification (Sect. 6.1), the objective of the various proposed pretext tasks might not be sufficiently well aligned. For example, instance discrimination methods, such as [7,18] used for pre-training in the current state-of-the-art unsupervised classification method [28], decrease similarity between all instances, even between those that belong to the same (unknown during training) class, thus potentially working against the set task. In contrast, in this paper we propose a classification-based pretext task whose objective is directly aligned with the end goal in this case. Knowing only the number of classes C we learn an unsupervised classifier (*Self-Classifier*) such that two different augmentations of the same image are classified similarly. In practice, such a task is prone to degenerate solutions, where all samples are assigned to the same class. To avoid them, we assert a uniform prior on the standard cross-entropy loss function, such that a solution with an equipartition of the data is an optimal solution. In fact, we show that the set of optimal solutions no longer includes degenerate ones.

Our approach can also be viewed as a form of deep unsupervised clustering (Sect. 2.2) [1,4,17,20,28,30–32] combined with contrastive learning. Similarly to deep clustering methods, we learn the parameters of a neural network and cluster (class) assignments simultaneously. Recently, clustering has been combined with contrastive learning in [2,32] with great success, yet in both studies clustering was employed as a separate step used for pseudo-labeling. In contrast, in this work we learn representations and cluster labels in a single-stage end-to-end manner, using only minibatch SGD.

The key contributions of this paper are:

1. A simple yet effective self-supervised single-stage end-to-end classification and representation learning approach. Unlike previous unsupervised classification works, our approach does not require any form of pre-training, expectation-maximization algorithm, pseudo-labeling, or external clustering. Unlike previous unsupervised representation learning works, our approach does not

require a memory bank, a second network (momentum), external clustering, stop-gradient operation, or negative pairs.
2. Although simple, our approach sets a new state of the art for unsupervised classification on ImageNet with 41.1% top-1 accuracy, achieves results comparable to state of the art for unsupervised representation learning, and attains a significant –2% AP) improvement in transfer to COCO det/seg compared to other self. sup. methods.
3. We are the first to provide quantitative analysis of self-supervised classification predictions alignment to a set of different class hierarchies (defined on ImageNet and its subpopulations), and show significant (up to 3.4% AMI) improvement over previous state of the art in this new metric.

2 Related Work

2.1 Self-Supervised Learning

Self-supervised learning methods learn compact semantic data representations by defining and solving a pretext task. In such tasks, naturally existing supervision signals are utilized for training. Many pretext tasks were proposed in recent years in the domain of computer vision, including colorization [35], jigsaw puzzle [24], image inpainting [25], context prediction [9], rotation prediction [14], and contrastive learning [2,5–7,10,18,23,29] just to mention a few.

Contrastive learning has shown great promise and has become a *de facto* standard for self-supervised learning. Two of the earliest studies of contrastive learning are Exemplar CNN [10], and Non-Parametric Instance Discrimination (NPID) [29]. Exemplar CNN [10], learns to discriminate between instances using a convolutional neural network classifier, where each class represents a single instance and its augmentations. While highly simple and effective, it does not scale to arbitrarily large amounts of unlabeled data since it requires a classification layer (softmax) the size of the dataset. NPID [29] tackles this problem by approximating the full softmax distribution with noise-contrastive estimation (NCE) [16] and utilizing a memory bank to store the recent representation of each instance to avoid computing the representations of the entire dataset at each time step of the learning process. Such approximation is effective since, unlike Exemplar CNN, it allows training with large amounts of unlabeled data. However, the proposed memory bank by NPID introduces a new problem - lack of consistency across representations stored in the memory bank, i.e., the representations of different samples in the memory bank are computed at multiple different time steps. Nonetheless, Exemplar CNN and NPID have inspired a line of studies of contrastive learning [2,5–7,18,21].

One such recent study is SwAV [2] which resembles the present work the most. SwAV takes advantage of contrastive methods without requiring to compute pairwise comparisons. More specifically, it simultaneously clusters the data while enforcing consistency between cluster assignments produced for different augmentations (or "views") of the same image, instead of comparing features

directly. To avoid a trivial solution where all samples collapse into a single cluster, SwAV alternates between representation learning using back propagation, and a separate clustering step using the Sinkhorn-Knopp algorithm. In contrast to SwAV, in this work we propose a model that allows learning both representations and cluster assignments in a single-stage end-to-end manner.

2.2 Deep Unsupervised Clustering

Deep unsupervised clustering methods simultaneously learn the parameters of a neural network and the cluster assignments of the resulting features using unlabeled data [1,4,17,20,28,30–32]. Such a task is understandably vulnerable to degenerate solutions, where all samples are assigned to a single cluster. Many different solutions that were proposed to avoid the trivial outcome are based on one or few of the following: a) pre-training mechanism; b) Expectation-Maximization (EM) algorithm (i.e., alternating between representation learning and cluster assignment); c) pseudo-labeling; and d) external clustering algorithm such as k-means.

Two of the earliest studies of deep clustering are DEC [30] and JULE [31]. DEC [30] initializes the parameters of its network using a deep autoencoder, and its cluster centroids using standard k-means clustering in the feature space. It then uses a form of EM algorithm, where it iterates between pseudo-labeling and learning from its own high confidence predictions. JULE [30], similarly to DEC, alternates between pseudo-labeling and learning from its own predictions. However, unlike DEC, JULE avoids a pre-training step and instead utilizes the prior on the input signal given by a randomly initialized ConvNet together with agglomerative clustering.

More recent approaches are SeLa [32] and IIC [20]. SeLa [32] uses a form of EM algorithm, where it iterates between minimization of the cross entropy loss and pseudo-labeling by solving efficiently an instance of the *optimal transport problem* using the Sinkhorn-Knopp algorithm. IIC [20] is a single-stage end-to-end deep clustering model conceptually similar to the approach presented in this paper. IIC maximizes the mutual information between predictions of two augmented views of the same sample. The two entropy terms constituting mutual information – the entropy of a sample and its negative conditional entropy given the other sample compete with each other, with the entropy being maximal when the labels are uniformly distributed over the clusters, and the negative conditional entropy being maximal for sharp one-hot instance assignments.

In this work, we follow a similar rationale for single-stage end-to-end classification without the use of any pseudo-labeling. Unlike IIC, our proposed loss is equivalent to the cross-entropy classification loss under a uniform label prior that guarantees non-degenerate, uniformly distributed optimal solution as explained in Sect. 3. Although many deep clustering approaches were proposed over the years, only two of them (SCAN [28] and SeLa [32]) have demonstrated scalability to large-scale datasets such as ImageNet. In fact, the task of unsupervised classification of large-scale datasets remains an open challenge.

3 Self-Classifier

Let x_1, x_2 denote two different augmented views of the same image sample x. Our goal is to learn a classifier $y \triangleq f(x_i) \in [C]$, where C is the given number of classes, such that two augmented views of the same sample are classified similarly, while avoiding degenerate solutions. A naive approach to this would be minimizing the following cross-entropy loss:

$$\tilde{\ell}(x_1, x_2) = -\sum_{y \in [C]} p(y|x_2) \log p(y|x_1), \tag{1}$$

where $p(y|x)$ is a row softmax with temperature τ_{row} [29] of the matrix of logits $\mathcal{S}$ produced by our model (backbone + classifier) for all classes (columns) and batch samples (rows). However, without additional regularization, an attempt to minimize (1) will quickly converge to a degenerate solution in which the network predicts a constant y regardless of the x. In order to remedy this, we propose to invoke Bayes and total probability laws, obtaining:

$$p(y|x_2) = \frac{p(y)p(x_2|y)}{p(x_2)} = \frac{p(y)p(x_2|y)}{\sum_{\tilde{y} \in [C]} p(x_2|\tilde{y})p(\tilde{y})}, \tag{2}$$

$$p(y|x_1) = \frac{p(y)p(y|x_1)}{p(y)} = \frac{p(y)p(y|x_1)}{\sum_{\tilde{x_1} \in B_1} p(y|\tilde{x}_1)p(\tilde{x}_1)}, \tag{3}$$

where B is a batch of N samples (B_1 are the first augmentations of samples of B), and $p(x|y)$ is a column softmax of the aforementioned matrix of logits $\mathcal{S}$ with the temperature τ_{col}. Now, assuming that $p(x_1)$ is uniform (under the reasonable assumption that the training samples are equi-probable), and, since we would like all classes to be used, assuming (an intuitive) uniform prior for $p(y)$, we obtain:

$$\ell(x_1, x_2) = -\sum_{y \in [C]} \frac{p(x_2|y)}{\sum_{\tilde{y}} p(x_2|\tilde{y})} \log \left(\frac{N}{C} \frac{p(y|x_1)}{\sum_{\tilde{x}_1} p(y|\tilde{x}_1)} \right), \tag{4}$$

where $p(y)$ and $p(\tilde{y})$ cancel out in (2), and $p(y)/p(\tilde{x}_1)$ becomes N/C in (3). In practice, we use a symmetric variant of this loss (that we empirically noticed to be better):

$$\mathcal{L} = \frac{1}{2} \Big(\ell(x_1, x_2) + \ell(x_2, x_1) \Big). \tag{5}$$

Note that the naive cross entropy in (1) is in fact mathematically equivalent to our proposed loss function in (4), under the assumption that $p(y)$ and $p(x)$ are uniform. Finally, despite being very simple (only few lines of PyTorch-like pseudocode in Algorithm 1) our method sets a new state of the art in self-supervised classification (Sect. 6.1).

Algorithm 1. *Self-Classifier* PyTorch-like Pseudocode

```
# N: number of samples in batch
# C: number of classes
# t_r / t_c: row / column softmax temperatures
# aug(): random augmentations
# softmaxX(): softmax over dimension X
# normX(): L1 normalization over dimension X
for x in loader:
  s1, s2 = model(aug(x)), model(aug(x))
  log_y_x1 = log(N/C * norm0(softmax1(s1/t_r)))
  log_y_x2 = log(N/C * norm0(softmax1(s2/t_r)))
  y_x1 = norm1(softmax0(s1/t_c))
  y_x2 = norm1(softmax0(s2/t_c))
  l1 = - sum(y_x2 * log_y_x1) / N
  l2 = - sum(y_x1 * log_y_x2) / N
  L = (l1 + l2) / 2
  L.backward()
  optimizer.step()
```

4 Theoretical Analysis

In this section, we show mathematically how *Self-Classifier* avoids trivial solutions by design, i.e., a collapsing solution is not in the set of optimal solutions of our proposed loss function (4). Proofs are provided in Supplementary.

Theorem 1 (Non-Zero Posterior Probability). *Let B be a batch of N samples with two views per sample, $(x_1, x_2) \in B$. Let $p(y)$ and $p(x)$ be the class and sample distributions, respectively, where $y \in [C]$. Let (5) be the loss function. Then, each class $y \in [C]$ will have at least one sample $y \in [C]$ with non-zero posterior probability $p(x|y) > 0$ assigned into it, and each sample $x \in [N]$ will have at least one class $y \in [C]$ with $p(x|y) > 0$.*

Theorem 2 (Optimal Solution With Uniform Prior). *Let B be a batch of N samples with two views per sample, $(x_1, x_2) \in B$. Let $p(y)$ and $p(x)$ be the class and sample distribution, respectively, where $y \in [C]$. Then, the uniform probabilities $p(y) = \frac{1}{C}$, $p(x) = \frac{1}{N}$ constitute a global minimizer of the loss (4).*

5 Implementation Details

5.1 Architecture

In all our experiments, we used ResNet-50 [19] backbone (as customary for all compared SSL works) initialized randomly. Following previous work, for our projection heads we used an MLP with 2 layers (of sizes 4096 and 128) with BN, leaky-ReLU activations, and ℓ_2 normalization after the last layer. On top of the projection head MLP we had 4 classification heads into $1K, 2K, 4K$ and $8K$

classes respectively. Each classification head was a simple linear layer without additive bias term. Row-softmax temperature τ_{row} was set to 0.1, while column-softmax temperature τ_{col} – to 0.05. Unless mentioned otherwise, evaluation for unsupervised classification (Sect. 6.1) was done strictly using the $1K$-classes classification head. For linear evaluation (Sect. 6.2) the MLP was dropped and replaced with a single linear layer of $1K$ classes.

5.2 Image Augmentations

We followed the data augmentations of BYOL [15] (color jittering, Gaussian blur and solarization), multi-crop [2] (two global views of 224×224 and six local views of 96×96) and nearest neighbor augmentation [11] (queue for nearest neighbor augmentation was set to $256K$). We refer to Table 8 in Sect. 7 for performance results without multi-crop and nearest neighbor.

5.3 Optimization

Unsupervised Pre-Training/Classification. Most of our training hyper-parameters are directly taken from SwAV [2]. We used a LARS optimizer [33] with a learning rate of 4.8 and weight decay of 10^{-6}. The learning rate was linearly ramped up (starting from 0.3) over the first 10 epochs, and then decreased using a cosine scheduler for 790 epochs with a final value of 0.0048 (for a total of 800 epochs). We used a batch size of 4096 distributed across 64 NVIDIA V100 GPUs.

Linear Evaluation. Similarly to [8] we used a LARS optimizer [33] with a learning rate of 0.8 and no weight decay. The learning rate was decreased using a cosine scheduler for 100 epochs. We used a batch size of 4096 distributed across 16 NVIDIA V100 GPUs. We have also tried the SGD optimizer in [18] with a batch size of 256, which gives similar results.

6 Results

6.1 Unsupervised Image Classification

We evaluate our approach on the task of unsupervised image classification using the large-scale ImageNet dataset (Tables. 1 to 3). We report the standard clustering metrics: Normalized Mutual Information (NMI), Adjusted Normalized Mutual Information (AMI), Adjusted Rand-Index (ARI), and Clustering Accuracy (ACC).

Our approach sets a new state-of-the-art performance for unsupervised image classification using ImageNet, on all four metrics (NMI, AMI, ARI and ACC), even when trained for a substantial lower number of epochs (Table 1). We compare our approach to the latest large-scale deep clustering methods [28,32] that have been explicitly evaluated on ImageNet. Additionally, we also compare our

Table 1. ImageNet unsupervised image classification using ResNet-50. NMI: Normalized Mutual Information, AMI: Adjusted Normalized Mutual Information, ARI: Adjusted Rand-Index, ACC: Clustering accuracy. †: produced by fitting a k-means classifier on the learned representations of the training set (models from official repositories were used), and then running inference on the validation set (results for SimCLRv2 and InfoMin are taken from [36]). SimSiam provide only 100-epoch model in their official repository. ∗: best result taken from the paper's official repository. Top-3 best methods per-metric are underlined. Best in bold

Method	Epochs	NMI	AMI	ARI	ACC
Representation learning methods					
SimCLRv2† [6]	1000	61.5	34.9	11.0	22.4
SimSiam† [8]	100	62.2	34.9	11.6	24.9
SwAV† [2]	800	64.1	38.8	13.4	28.1
MoCoV2† [7]	800	66.6	45.3	12.0	30.6
DINO† [3]	800	66.2	42.3	15.6	30.7
OBoW† [13]	200	66.5	42.0	16.9	31.1
InfoMin† [27]	800	68.8	48.3	14.7	33.2
BarlowT† [34]	1000	67.1	43.6	17.6	34.2
Clustering based methods					
SeLa* [32]	280	65.7	42.0	16.2	30.5
SCAN [28]	800+125	72.0	51.2	27.5	39.9
Self-Classifier	100	71.2	49.2	26.1	37.3
Self-Classifier	200	72.5	51.6	28.1	39.4
Self-Classifier	400	72.9	52.3	28.8	40.2
Self-Classifier	800	**73.3**	**53.1**	**29.5**	**41.1**

approach to the latest self-supervised representation learning methods (using ImageNet-pretrained models provided in their respective official repositories) after fitting a k-means classifier to the learned representations computed on the training set. For all methods we run inference on the validation set (unseen during training).

The current state-of-the-art approach, SCAN [28], is a multi-stage algorithm that involves: 1) pre-training (800 epochs); 2) offline k-nearest neighbor mining; 3) clustering (100 epochs); and 4) self-labeling and fine-tuning (25 epochs). In contrast, *Self-Classifier* is a single-stage simple-to-implement model (Algorithm 1) that is trained only with minibatch SGD. At only 200 epochs *Self-Classifier* already outperforms SCAN with 925 epochs.

SCAN provided an interesting qualitative analysis of alignment of its unsupervised class predictions to a certain (single) level of the default (WordNet) ImageNet semantic hierarchy. In contrast, here we propose a more diverse set of

Table 2. ImageNet-superclasses unsupervised image classification accuracy using ResNet-50. We define new datasets that contain broad classes which each subsume several of the original ImageNet classes. See Supplementary for details of each superclass. †: produced by fitting a k-means classifier on the learned representations of the training set (models from official repositories were used), and then running inference on the validation set. Results for SCAN and SeLa were produced using ImageNet-pretrained models provided in their respective official repositories

	Number of ImageNet superclasses					
Method	10	29	128	466	591	1000
Representation learning methods						
SwAV† [2]	79.1	69.4	58.0	46.3	34.5	28.1
MoCoV2† [7]	80.0	72.8	63.8	51.4	36.8	30.6
DINO† [3]	79.7	71.3	60.7	49.2	37.8	30.7
OBoW† [13]	83.9	76.5	67.4	53.5	35.7	31.1
BarlowT† [34]	80.2	72.1	62.7	52.7	40.9	34.2
Clustering based methods						
SeLa [32]	55.2	44.9	40.6	36.6	37.8	30.5
SCAN [28]	85.3	79.3	71.2	59.6	44.7	39.9
Self-Classifier	**85.7**	**79.7**	**71.8**	**60.0**	**46.7**	**41.1**

quantitative metrics to evaluate the performance of self-supervised classification methods on various levels of the default ImageNet hierarchy, as well as on several hierarchies of carefully curated ImageNet subpopulations (BREEDS [26]). We believe that this new set of hierarchical alignment metrics expanding on the leaf-only metric used so far, will allow deeper investigation of how self-supervised classification approaches perceive the internal taxonomy of classes of unlabeled data they are applied to, exposing their strength and weaknesses in a new and interesting light. We use these new metrics to compare our proposed approach to previous unsupervised clustering work [28,32], as well as state-of-the-art representation learning work [2,3,6–8,13,27,34].

In Table 2 we report results for different numbers of ImageNet superclasses (10, 29, 128, 466 and 591) resulting from cutting the default (WordNet) ImageNet hierarchy on different levels. See Supplementary for details of each superclass. The results in this table, that are significantly higher then the result for leaf (1000) classes for any hierarchy level, indicate that examples misclassified on the leaf level tend to be assigned to other clusters from within the same superclass. Furthermore, we see that *Self-Classifier* consistently outperforms previous work on all hierarchy levels.

In Table 3 we report the results on four ImageNet subpopulation datasets of BREEDS [26]. These datasets are accompanied by class hierarchies re-calibrated by [26] such that classes on same hierarchy level are of the same visual granular-

Table 3. ImageNet-subsets (BREEDS) unsupervised image classification using ResNet-50. The four BREEDS datasets are: Entity13, Entity30, Living17 and Nonliving26. NMI: Normalized Mutual Information, AMI: Adjusted Normalized Mutual Information, ARI: Adjusted Rand-Index, ACC: Clustering accuracy. †: produced by fitting a k-means classifier on the learned representations of the training set (models from official repositories were used), and then running inference on the validation set. Results for SCAN and SeLa were produced using ImageNet-pretrained models provided in their respective official repositories

	Entity13				Entity30				Living17				Nonliving26			
Method	NMI	AMI	ARI	ACC	NMI	AMI	ARI	ACC	NMI	AMI	ARI	ACC	NMI	AMI	ARI	ACC
Representation learning methods																
SwAV† [2]	64.8	39.9	15.2	75.6	64.6	39.4	15.1	70.5	61.0	40.3	15.7	85.2	62.0	41.1	19.2	63.1
MoCoV2† [7]	67.3	46.6	14.7	79.0	66.4	45.8	15.1	74.6	61.2	45.7	16.3	89.7	63.3	46.2	19.3	66.2
DINO† [3]	67.2	43.7	18.0	78.2	66.8	43.2	18.1	73.7	63.8	45.1	19.6	88.2	63.8	43.9	21.8	66.7
OBoW† [13]	66.4	42.3	17.5	82.2	64.9	40.7	16.4	77.6	53.8	34.0	12.0	91.1	64.8	45.4	22.9	67.9
BarlowT† [34]	68.2	45.5	20.5	77.7	67.7	45.1	20.7	73.0	64.7	47.2	22.2	88.0	64.8	45.7	24.9	66.7
Clustering based methods																
SeLa [32]	67.6	44.8	19.4	50.7	68.2	45.7	21.2	52.6	**71.8**	**53.9**	**29.7**	80.8	68.9	46.6	24.6	67.1
SCAN [28]	72.4	52.3	29.2	83.7	71.3	50.8	27.8	80.0	65.2	49.4	25.3	**92.5**	70.0	53.6	33.4	74.4
Self-Classifier	**73.6**	**54.1**	**30.7**	**84.4**	**72.9**	**53.4**	**29.8**	**81.0**	67.2	51.8	26.4	90.8	**72.2**	**57.0**	**36.8**	**76.7**

ity. Each dataset contains a specific subpopulation of ImageNet, such as 'Entities', 'Living' things and 'Non-living' things, allowing for a more fine-grained evaluation of hierarchical alignment of self-supervised classification predictions. Again, we see consistent improvement of *Self-Classifier* over previous work and self-supervised representation baselines.

6.2 Image Classification with Linear Models

We evaluate the quality of our unsupervised features using the standard linear classification protocol. Following the self-supervised pre-training stage, we freeze the features and train on top of it a supervised linear classifier (a single fully-connected layer). This classifier operates on the global average pooling features of a ResNet. Table 4 summarizes the results and comparison to the state-of-the-art methods for various number of training budgets (100 to 800 epochs).

In addition to good results for unsupervised classification (Sect. 6.1), *Self-Classifier* additionally achieves results comparable to state of the art for linear classification evaluation using ImageNet. Specifically, as detailed in Table 4, it is one of the top-3 result for 3 out of 4 of the training budgets reported, and top-1 in the 100 epochs category.

6.3 Transfer Learning

We further evaluate the quality of our unsupervised features by transferring them to other tasks - object detection and instance segmentation. Table 5 reports results for VOC07+12 [12] and COCO [22] datasets. We fine-tune our pre-trained model end-to-end in the target datasets using the public codebase from MoCo [18]. We obtain significant (–2%) improvements in the more challenging COCO det/seg over all the self-supervised baselines.

Table 4. ImageNet linear classification using ResNet-50. Top-1 accuracy vs. number of training epochs. Top-3 best methods per-category are underlined

Method	Number of training epochs			
	100	200	400	800
Supervised	76.5	–	–	–
SimCLR [5]	66.5	68.3	69.8	70.4
MoCoV2 [7]	67.4	67.5	71.0	71.1
SimSiam [8]	68.1	70.0	70.8	71.3
SimCLRv2 [6]	–	–	–	71.7
InfoMin [27]	–	–	–	73.0
BarlowT [34]	–	–	–	73.2
OBoW [13]	–	73.8	–	–
BYOL [15]	66.5	70.6	73.2	74.3
NNCLR [11]	69.4	70.7	74.2	74.9
DINO [3]	–	–	–	**75.3**
SwAV [2]	72.1	**73.9**	**74.6**	**75.3**
Self-Classifier	**72.4**	73.5	74.2	74.1

Table 5. Transfer learning: object detection and instance segmentation. Results for other methods are taken from [34]

Method	VOC07+12 det			COCO det			COCO seg		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Supervised	53.5	81.3	58.8	38.2	58.2	41.2	33.3	54.7	35.2
MoCo-v2[7]	**57.4**	82.5	**64.0**	39.3	58.9	42.5	34.4	55.8	36.5
SwAV[2]	56.1	**82.6**	62.7	38.4	58.6	41.3	33.8	55.2	35.9
SimSiam[8]	57.0	82.4	63.7	39.2	59.3	42.1	34.4	56.0	36.7
BarlowT[34]	56.8	**82.6**	63.4	39.2	59.0	42.5	34.3	56.0	36.5
Self-Classifier	56.6	82.4	62.6	**41.5**	**61.3**	**45.0**	**36.1**	**58.1**	**38.7**

6.4 Qualitative Results

In Supplementary, we visualize and analyse a subset of high/low accuracy classes predicted by *Self-Classifier* on **unseen data** (ImageNet validation).

7 Ablation Study

In this section, we evaluate the impact of the design choices of *Self-Classifier*. Namely, the loss function, the number of classes (C), number of classification heads, fixed vs learnable classifier, MLP architecture, Softmax temperatures (row and column), batch-size, some of the augmentations choices, and NN queue length. We evaluate the different models after 100 self-supervised epochs and

Table 6. Ablation: loss function generality. For column definitions see Table 1. SCAN + Eq. (5) is SCAN with clustering step loss replaced with ours.

Method	NMI	AMI	ARI	ACC
SCAN [28]	72.0	51.2	27.5	39.9
SCAN + our loss (Eq. (5))	**72.7**	**52.2**	**29.0**	**40.4**

report results on ImageNet validation set. We report both the K-NN (K=20) classifier accuracy (evaluating the learned representations) and the unsupervised clustering accuracy (evaluating unsupervised classification performance).

Loss Function. For both illustrating the generality of our proposed loss function and making more direct comparison with the unsupervised classification state-of-the-art (SCAN [28]), in Table 6 we report the results of running SCAN official code, while replacing their loss function (in the clustering step) with ours (Eq. (5)) and keeping everything else (e.g. classification heads and augmentations) same as in SCAN. As we can see, our proposed loss generalizes well and improves SCAN result (e.g. by 1.5% ARI and 0.5% ACC). Further results improvements are obtained using our full method (as shown in Table 1).

Number of Classes and Classification Heads. Table 7a reports the results for various number of classes and classification heads. Very interestingly, and somewhat contrary to the intuition of previous unsupervised classification works [28,32] who used the same number of classes for all heads, we found that using a different number of classes for each head while still keeping the total number of parameters constant (e.g. 15×1k vs. 1k+2k+4k+8k) improves results on both metrics. We believe that such a learning objective forces the model to learn a representation that is more invariant to the number of classes, thus improving its generalization performance.

Fixed/Learnable Classifier. As expected, we found that a learnable classifier performs better than a fixed one (Table 7d).

MLP Architecture. Table 7c reports the results for various sizes of hidden/output layers. Surprisingly, we found that decreasing the number of hidden layers and their size improves both metrics. As a result, our best model (4096/128 MLP) has 30% less parameters than the model used in SCAN [28] (that used 2048 sized input to its cls. heads). In addition, we verified there is no peak performance difference between ReLU and leaky-ReLU activation in the MLP.

Softmax Temperature. Table 7b reports the results for a range of Row/-Column softmax temperatures. We found that the ratio between the two temperatures is important for performance (specifically clustering accuracy). The model is robust to ratios (row over column) in the range of 2.0 - 3.5.

Batch Size. Table 7f reports the results for a range of batch size values (256 to 4096). Similarly to previous self-supervised work (and specifically clustering-based), performance improves as we increase batch size.

Table 7. Ablation study. After 100 epochs, reporting performance for ImageNet as accuracy of <'k-NN' | 'unsupervised clustering'> in each experiment.

(a) Classification heads. $^{(2k)}$ $^{(4k)}$ $^{(8k)}$: 2k, 4k and 8k over-clustering accuracy.

	1 × 1k	5 × 1k	10 × 1k	15 × 1k	1 × 2k	1 × 4k	1k+2k+4k+8k
Acc. (%)	59.6|34.1	58.7|34.0	58.6|33.5	58.8|33.9	59.3|38.8$^{(2k)}$	57.0|42.9$^{(4k)}$	**61.7|37.3**,40.6$^{(2k)}$,44.2$^{(4k)}$,48.0$^{(8k)}$

(b) Softmax temperature

τ_{column} \ τ_{row}	0.07	0.1
0.03	59.9|36.9	59.2|36.9
0.05	58.9|29.2	**61.7|37.3**

(l) MLP architecture

MLP hidden layer(s)	MLP output layer 128	256
1 × 4096	**61.7|37.3**	61.3|33.5
2 × 4096	60.9|36.4	60.4|33.6
2 × 8192	60.0|36.9	59.6|36.7

(d) Fixed/Learnable classifier

	Fixed	Learnable
Acc. (%)	57.6|32.2	**61.7|37.3**

(e) Nearest neighbor queue length

Queue len.	128K	256K	512K	1M
Acc. (%)	59.2|36.8	**61.7|37.3**	60.3|36.9	56.8|35.5

(f) Batch size

Batch size	256	512	1024	2048	4096
Acc. (%)	49.0|20.9	52.2|23.1	54.5|26.8	57.0|35.1	**61.7|37.3**

Multi-crop and Nearest Neighbor Augmentations. Table 8 reports the impact of removing multi-crop [2] and nearest neighbor augmentations [11] on linear classification accuracy and compares to other state-of-the-art methods.

Nearest Neighbor Queue Length. The model is somewhat robust to a queue length in the range of 128K - 512K (Table 7e), while increasing it further decreases performance. Most likely due to stale embeddings (as noted by [11] as well).

8 Comparative Analysis

A common and critical element of all self-supervised learning methods is collapse prevention. In this section, we discuss the various approaches of state-of-the-art models for preventing collapse. The approaches can be categorized into two categories: 1) negative samples; and 2) stop-grad operation. Where in practice, stop-grad operation includes two more sub-categories: 2.a) external clustering; and 2.b) momentum encoder. In this paper, we propose a third and completely new approach for collapse prevention - a non-collapsing loss function, i.e., a loss function without degenerate optimal solutions.

Negative Samples. SimCLR [5] and Moco [18] prevent collapse by utilizing negative pairs to explicitly force dissimilarity.

Table 8. Performance without multi-crop and without nearest neighbor augmentations. ImageNet Top-1 linear classification accuracy after 100 epochs. [2,5,7,15] are taken from [8]

	SimCLR [5]	BYOL [15]	SwAV [2]	MoCoV2 [7]	SimSiam [8]	**Ours**
Acc. (%)	66.5	66.5	66.5	67.4	**68.1**	**68.1**

External Clustering. SwAV [2], SeLa [32] and SCAN [28] prevent collapse by utilizing external clustering algorithm such as K-Means (SCAN) or Sinkhorn-Knopp (SwAV/SeLa) for generating pseudo-labels.

Momentum Encoder. MoCo [18], BYOL [15] and DINO [3] prevent collapse by utilizing the momentum encoder proposed by MoCo. The momentum encoder generates a different yet fixed pseudo target in every iteration.

Stop-grad Operation. SimSiam [8] prevent collapse by applying a stop-grad operation on one of the views, which acts as a fixed pseudo label. In fact, except for SimCLR, all of the above methods can be simply differentiated by where exactly a stop-grad operation is used. SwAV/SeLa/SCAN apply a stop-grad operation on the clustering phase, while MoCo/BYOL/DINO apply a stop-grad operation on a second network that is used for generating assignments.

Non-Collapsing Loss Function. In contrast, we show mathematically (Sec. 4) and empirically (Sect. 6) that *Self-Classifier* prevents collapse with a novel loss function (4) and without the use of external clustering, pseudo-labels, momentum encoder, stop-grad nor negative pairs. More specifically, a collapsing solution is simply not in the set of optimal solutions of our proposed loss, which makes it possible to train *Self-Classifier* using just a single network and a simple SGD.

9 Conclusions and Limitations

We introduced *Self-Classifier*, a new approach for unsupervised end-to-end classification and self-supervised representation learning. Our approach is mathematically justified and simple to implement. It sets a new state-of-the-art performance for unsupervised classification on ImageNet and achieves comparable to state of the art results for unsupervised representation learning. We provide a thorough investigation of our method in a series of ablation studies. Furthermore, we propose a new hierarchical alignment quantitative metric for self-supervised classification establishing baseline performance for a wide range of methods and showing advantages of our proposed approach in this new task. *Limitations* of this paper include: (i) our method relies on knowledge of the number of classes, but in some cases it might not be optimal as the true number of classes should really be dictated by the data itself. In this paper we relax this potential weakness by introducing the notion of multiple classification heads, but we believe further investigation would be an interesting future work direction; (ii) one of

the most common sources of error we observed is merging of nearby classes (e.g. different breeds of cat), introducing additional regularization for reducing this artifact is also an interesting direction of future work.

References

1. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 132–149 (2018)
2. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Adv. Neural Inf. Process. Syst. **33**, 9912–9924 (2020)
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 9650–9660 (2021)
4. Chang, J., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep adaptive image clustering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5879–5887 (2017)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, ICML, pp. 1597–1607 (2020)
6. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. Adv. Neural Inf. Process. Syst.**33**, 22243–22255 (2020)
7. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. In: arXiv preprint arXiv:2003.04297 (2020)
8. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758 (2021)
9. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1422–1430 (2015)
10. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 766–774 (2014)
11. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9588–9597 (2021)
12. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
13. Gidaris, S., Bursuc, A., Puy, G., Komodakis, N., Cord, M., Perez, P.: Obow: online bag-of-visual-words generation for self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6830–6840 (2021)
14. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: International Conference on Learning Representations, ICLR (2018)

15. Grill, J.B., et al.: Bootstrap your own latent - a new approach to self-supervised learning. Adv. Neural Inf. Process. Syst. **33**, 21271–21284 (2020)
16. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 297–304 (2010)
17. Haeusser, P., Plapp, J., Golkov, V., Aljalbout, E., Cremers, D.: Associative deep clustering: training a classification network with no labels. In: Brox, T., Bruhn, A., Fritz, M. (eds.) GCPR 2018. LNCS, vol. 11269, pp. 18–32. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12939-2_2
18. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
20. Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9865–9874 (2019)
21. Li, J., Zhou, P., Xiong, C., Hoi, S.: Prototypical contrastive learning of unsupervised representations. In: International Conference on Learning Representations, pp. 9475–9484 (2021)
22. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
23. Misra, I., Maaten, L.V.D.: Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6707–6717 (2020)
24. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 69–84. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_5
25. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2536–2544 (2016)
26. Santurkar, S., Tsipras, D., Madry, A.: Breeds: Benchmarks for subpopulation shift. arXiv preprint arXiv:2008.04859 (2020)
27. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? Adv. Neural Inf. Process. Syst. (2020)
28. Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., Van Gool, L.: SCAN: learning to classify images without labels. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12355, pp. 268–285. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58607-2_16
29. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3733–3742 (2018)
30. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487. PMLR (2016)

31. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5147–5156 (2016)
32. YM., A., C., R., A., V.: Self-labelling via simultaneous clustering and representation learning. In: International Conference on Learning Representations (2020)
33. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
34. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 12310–12320. PMLR (2021)
35. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40
36. Zheltonozhskii, E., Baskin, C., Bronstein, A.M., Mendelson, A.: Self-supervised learning for large-scale unsupervised image clustering. arXiv preprint arXiv:2008.10312 (2020)