ScatSimCLR: self-supervised contrastive learning with pretext task regularization for small-scale datasets

Vitaliy Kinakh Olga Taran Svyatoslav Voloshynovskiy*
Department of Computer Science, University of Geneva, Switzerland
{vitaliy.kinakh, olga.taran, svolos}@unige.ch

Abstract

In this paper, we consider a problem of self-supervised learning for small-scale datasets based on contrastive loss between multiple views of the data, which demonstrates the state-of-the-art performance in classification task. Despite the reported results, such factors as the complexity of training requiring complex architectures, the needed number of views produced by data augmentation, and their impact on the classification accuracy are understudied problems. To establish the role of these factors, we consider an architecture of contrastive loss system such as SimCLR, where baseline model is replaced by geometrically invariant "hand-crafted" network ScatNet with small trainable adapter network and argue that the number of parameters of the whole system and the number of views can be considerably reduced while practically preserving the same classification accuracy. In addition, we investigate the impact of regularization strategies using pretext task learning based on an estimation of parameters of augmentation transform such as rotation and jigsaw permutation for both traditional baseline models and ScatNet based models. Finally, we demonstrate that the proposed architecture with pretext task learning regularization achieves the state-of-the-art classification performance with a smaller number of trainable parameters and with reduced number of views. Code: https://github.com/vkinakh/scatsimclr

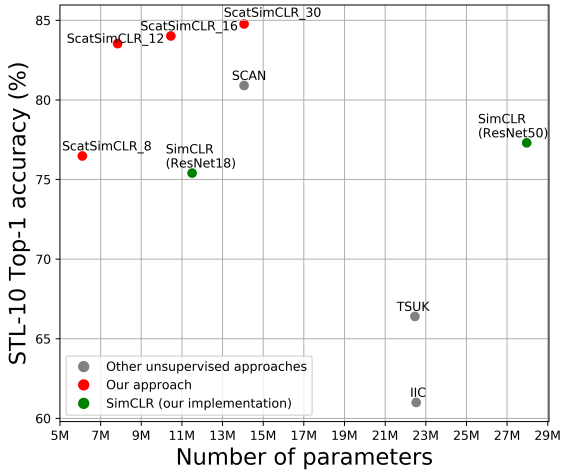

Figure 1. STL-10 [1] Top-1 accuracy of self-supervised methods. Gray dots indicate other self-supervised methods. Our method, ScatSimCLR, is shown in red. Our implementation of SimCLR is shown in green. The results are obtained with models trained for 1000 epochs.

1. Introduction

Self-supervised learning refers to the learning of data representations that are not based on labeled data. The recent techniques of self-supervised learning such as SimCLR [10], SwAV [8], SeLa [3] and BYOL [14] demonstrate a classification performance close to their supervised counterparts. The main common idea behind these self-supervised approaches is to learn an embedding that produces an invariant representation under various data augmentations ranging from image filtering to geometrical transformations. In most cases, some powerful neural network such as for example ResNet [10] is used to implement this embedding. It is demonstrated [10] that the classification accuracy of these systems increases with the increase of the complexity of ResNet represented by the larger number of parameters capable of producing the invariance of visual representation under the broad family of augmentations. Typically the number of parameters of such networks ranges from 5M to 500M that makes their training quite a complex and time consuming task and requires a lot of training data.

At the time, in many practical applications it is infeasible to collect a lot of training data. Moreover, in many cases the amount of labeled data is limited. We refer to these cases as a "small dataset" problem. These restrictions lead to the overfitting of large scale models such as ResNet

*S. Voloshynovskiy is a corresponding author.

and result in their poor generalization. Therefore, to benefit from the recent advancements of self-supervised learning, which performance is generally demonstrated on the large scale datasets such as ImageNet [11], it is important to develop efficient representation learning techniques adapted to the small dataset problem.

In this paper, we try to address the problem of self-supervised learning based on contrastive loss in the application to the small dataset problem by replacing complex ResNet network by networks with a smaller number of parameters. More particularly, we investigate a question whether such complex networks as ResNet are really needed to achieve the targeted representation invariance assuming that the invariance to some families of augmentations can be achieved by a hand-crafted embedding. One candidate for such an invariant hand-crafted embedding is ScatNet [2, 6], which is known to produce stable embeddings under the deformations in terms of Lipschitz continuity property. As a by-product of such an invariance, one might assume that the number of augmentations needed for the training of invariant embedding can be reduced accordingly. Finally, the overall complexity of training might also be lower. To investigate these questions, we propose a ScatSimCLR architecture where the complex ResNet is replaced by ScatNet followed by a simple adapter network. We demonstrate that ScatSimCLR with a reduced number of training parameters and a reduced number of used augmentations can achieve similar performance and in some cases even outperform SimCLR. Furthermore, we demonstrate that the introduction of pretext task learning regularization, yet another popular technique of self-supervised learning, is beneficial for representation learning both for basic neural networks like SimCLR as well as for the proposed architectures.

Main contributions are:

1. We propose a model with the reduced number of parameters of the embedding network while preserving the same classification performance. This is achieved due to the usage of the geometrically invariant network ScatNet. Figures 1 and 8 demonstrate the performance of ScatSimCLR on STL-10 and CIFAR100-20 [1] [1] as a function of the number of parameters with respect to the other state-of-the-art methods. The ScatSimCLR outperforms the state-of-the-art SCAN [40] and RUC [34] methods known to produce the top result for STL-10 and CIFAR100-20 datasets, while using even lower complexity networks.

2. We investigate the impact of pretext task regularization on the classification performance. This includes the regularization based on the estimation of parameters of applied augmentation transform such as the rotation angle and jigsaw permutation.

3. We investigate the impact of the ScatNet and pretext task regularization on several datasets such as STL-10 and CIFAR100-20 and establish that the ScatSimCLR achieves state-of-the-art performance even with the smaller number of parameters.

4. We investigate the role of augmentations in the context of representation learning based on the geometrically invariant ScatNet.

5. We demonstrate that the data agnostic ScatNet is applicable to the datasets with different statistics and labels and does not require extensive training as in the case of ResNet used for SimCLR contrastive learning.

6. Finally, we demonstrate that individual contributions of ScatNet and pretext tasks improves the performance of the model on classification tasks.

[1] CIFAR100-20 is CIFAR100 dataset with 20 superclasses.

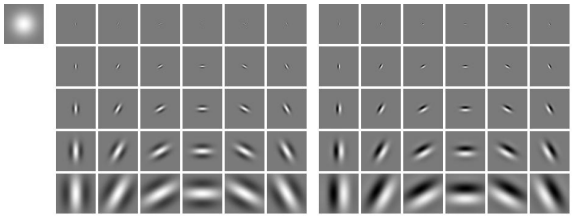

Figure 2. ScatNet [2, 6] filter bank for J = 5 (number of scales) and L = 6 (number of rotations). The top left image corresponds to a low-pass filter. The first left half image corresponds to the real parts of ScatNet filters arranged according to the scales (rows) and orientations (columns). The right half image corresponds to the imaginary part of ScatNet filters.

2. Related work

We briefly summarize the related work to the concepts used in this paper.

Contrastive learning is considered among the state-of-the-art techniques for self-supervised learning [32, 19, 41, 38, 37, 10]. The contrastive learning is based on a parameterized encoding or embedding that produces a low-dimensional data representation such that minimizes some distance between similar (positive) data pairs and maximizes for dissimilar (negative) ones. One of the central questions in contrastive learning is the generation or selection of positive and negative examples without labels. It is a common practice to generate positive examples by a data augmentation when multiple "views" for a given image are created by applying different crops [35, 41, 10, 4, 17, 42, 26], various geometrical transformations of affine or projective families [13], jigsaw image permutations [29, 12], splitting image into luminance and chrominance components

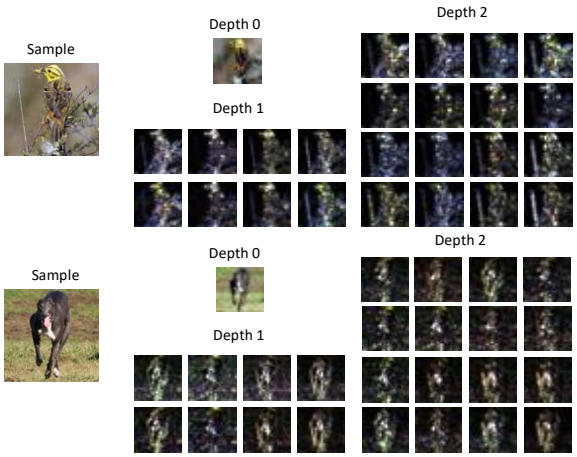

Figure 3. Examples of ScatNet [2] feature vectors for $L = 4$ and $J = 2$ for STL-10 [1] images. ScatNet transform is applied to each color channel separately, then each channel is normalized and merged into a RGB image for better visualization.

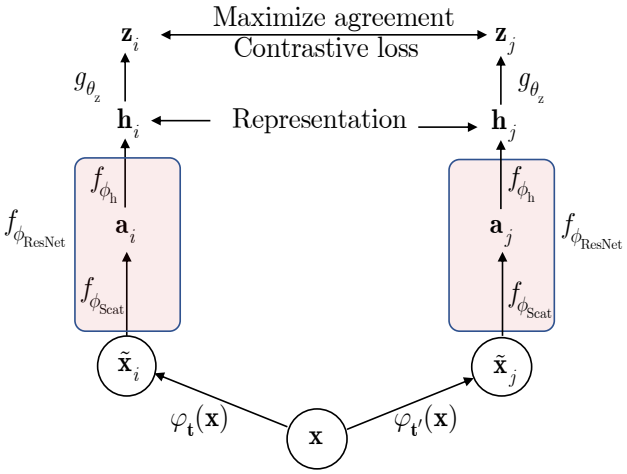

Figure 4. Contrastive learning of visual representation according to SimCLR architecture. In this work, an encoding network $f_{\phi_{\text{ResNet}}}$ producing a representation $\mathbf{h}$ is replaced by ScatNet network $f_{\phi_{\text{Scat}}}$ and adapter network $f_{\phi_{\text{h}}}$. In the rest, the architecture remains the same as for SimCLR.

[38], applying low-pass and high-pass filtering [10, 14], predicting one view from another [44], etc. The overall idea is to create a sort of "associations" between different parts of the same object or scene via a common latent space representation. The negative pairs are generally considered as images or parts of images randomly sampled from unlabeled data. The recent study [39] demonstrates the role of positive and negative example selection and generation and their impact on the overall classification accuracy.

Hand-crafted geometrically invariant transform - ScatNet[2] is a class of Convolutional Neural Networks (CNNs) designed with fixed weights [6] that has a set of important properties. (1) *Deformation stability*: in contrast to the Fourier transformation that is generally unstable to deformations at high frequencies[3], ScatNet is stable to deformations in terms of Lipschitz continuity property. The stability is gained due to the use of non-linearity and average pooling. (2) *Hand-crafted design*: ScatNet is considered as a deep convolution network with fixed filters in a form of wavelet basis functions independent of a specific dataset that at the same time provides (3) *sparse representation*. (4) *Interpretable representation*: in contrast to the most deep convolutional networks that output only the last layer, ScatNet outputs all layers representing the different signal scales. Figure 2 shows typical ScatNet filters for the depth $J = 5$ and number of orientations $L = 6$. A set of features produced by ScatNet for the STL-10 [1] samples is shown in Figure 3.

Hand-crafted pretext task and clustering based pseudo-labeling are used to compensate for the lack of labeled data. The hand-crafted pretext task is considered as a sort of self-supervised learning when the input data are manipulated to extract a supervised signal in the form of a pretext task learning. The hand-crafted pretext task has been widely used in various settings to predict the patch context [12, 28], solve jigsaw puzzles from the same [29] and different images [31], colorize images [43, 25], predict noise [5], count [30], estimate parameters of rotations [13], inpaint patches [35], spot artifacts [20], generate images [36] as well as for predictive coding [32, 18] and instance discrimination [41, 17, 10, 38, 27]. We refer the reader to [22] for the details of these methods. At the same time, clustering based pseudo-labeling can be used as pseudo-labels to learn visual representations [7]. Recent work [8] extends this idea to soft cluster assignment in contrast to hard-assignment. In this work we only consider pretext task learning based on rotation and jigsaw parameters' estimation.

[2]The efficient GPUs' implementation are provided in [33, 2]

[3]The Fourier transform is invariant to translation.

3. ScatSimCLR

The proposed architecture of self-supervised representation learning is shown in Figure 4 and it is based on the SimCLR framework.

For the batch size N, given $\{\mathbf{x}_k\}_{k=1}^{N}$ in the batch, SimCLR produces two augmented versions $\tilde{\mathbf{x}}_{2k-1} = \varphi_{\mathbf{t}}(\mathbf{x}_k)$ and $\tilde{\mathbf{x}}_{2k} = \varphi_{\mathbf{t}'}(\mathbf{x}_k)$ of each $\mathbf{x}_k$ using parameterized augmentation transform $\varphi_{\mathbf{t}}$ with parameters $\mathbf{t}$ and $\mathbf{t}'$ for each view. Both augmented images are first processed by the feature extraction network $f_{\boldsymbol{\phi}_{\text{ResNet}}}$ thus producing two representations $\mathbf{h}_{2k-1} = f_{\boldsymbol{\phi}_{\text{ResNet}}}(\tilde{\mathbf{x}}_{2k-1})$ and $\mathbf{h}_{2k} = f_{\boldsymbol{\phi}_{\text{ResNet}}}(\tilde{\mathbf{x}}_{2k})$ and then by the projection network $g_{\boldsymbol{\theta}_{\text{z}}}$ that produces two vectors $\mathbf{z}_{2k-1} = g_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{h}_{2k-1})$ and $\mathbf{z}_{2k} = g_{\boldsymbol{\theta}_{\text{z}}}(\mathbf{h}_{2k})$.

SimCLR contrastive loss is defined as:

$$\mathcal{L}_{\text{SimCLR}}(\boldsymbol{\phi}_{\text{ResNet}}, \boldsymbol{\theta}_{\text{z}}) = \frac{1}{2N}\sum_{k=1}^{N}[\ell(2k-1, 2k) + \ell(2k, 2k-1)], \quad (1)$$

where $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} 1_{[k\neq i]}\exp(s_{i,k}/\tau)}$ with $s_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\| \, \|\mathbf{z}_j\|)$ denotes a pairwise similarity for all pairs $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ and $1_{[k\neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 iff $k \neq i$ and τ denotes a temperature parameter.

SimCLR demonstrates the increasing performance in classification accuracy as shown in Figure 1 with the growth of the number of parameters of ResNet $f_{\boldsymbol{\phi}_{\text{ResNet}}}$ from about 11M to 28M. Thus it is commonly assumed that this increase in performance is due to the increase of $f_{\boldsymbol{\phi}_{\text{ResNet}}}$ network capacity and its ability to learn more complex associations between different parts of objects. Obviously, all the parameters of the network should be trained to efficiently encode these associations.

In contrast to this, we argue that the complex trainable ResNet $f_{\boldsymbol{\phi}_{\text{ResNet}}}$ can be replaced by the hand-crafted non-trainable ScatNet network $f_{\boldsymbol{\phi}_{\text{Scat}}}$ and small capacity trainable adapter network $f_{\boldsymbol{\phi}_{\text{h}}}$. ScatNet network $f_{\boldsymbol{\phi}_{\text{Scat}}}$ is a hand-crafted network with the fixed parameters and it is agnostic to a particular dataset and corresponding inter-object associations. It produces invariant low-level image representation $\mathbf{a}$. At the same time, the low capacity adapter network $f_{\boldsymbol{\phi}_{\text{h}}}$ aggregates the output of ScatNet and produces the visual representation $\mathbf{h}$. Therefore, one should only train the parameters of an adapter network that is just a fraction of ResNet. Similarly to the results presented in Figure 1, one can change the complexity of the adapter network and investigate its impact on the overall system performance. For the fair comparison, we keep the remaining architecture the same as in SimCLR.

To process color images, we simply apply ScatNet to each color channel as shown in Figure 5. We have used RGB representation but YCbCr or YUV spaces might be even more suited due to the properties of Y component reflecting grayscale images.

We will refer to SimCLR network with the replaced ResNet by ScatNet and the adapter network as ScatSimCLR.

4. Additional regularizer as a pretext task self-learning

In this section, we introduce an additional form of regularization that does not require any labeling, pseudo-labeling or mining for positive or negative neighbors as

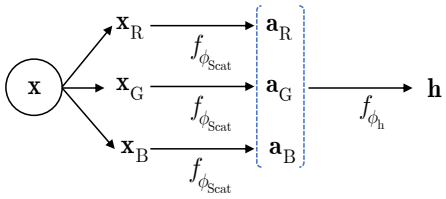

Figure 5. The encoding network for color images. An image $\mathbf{x}$ is represented by three color components $\{\mathbf{x}_{\text{R}}, \mathbf{x}_{\text{G}}, \mathbf{x}_{\text{B}}\}$. Each color component is processed by ScatNet network $f_{\boldsymbol{\phi}_{\text{Scat}}}$ and then the adapter network $f_{\boldsymbol{\phi}_{\text{h}}}$ aggregates the outputs to produce the representation $\mathbf{h}$.

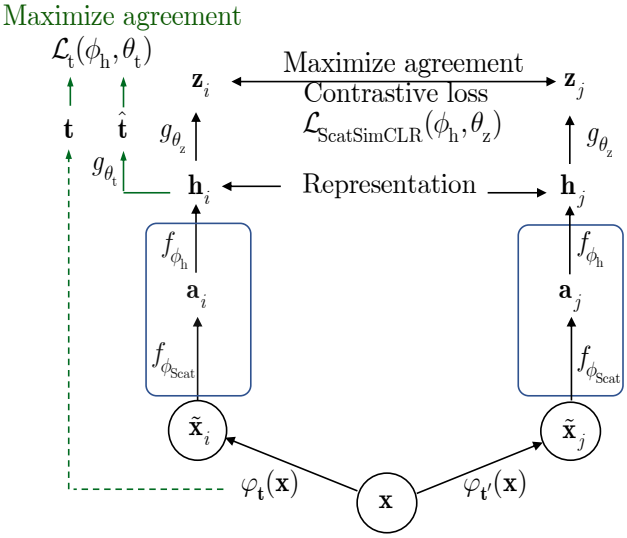

Figure 6. ScatSimCLR with an additional regularization based on the estimation of parameter $\mathbf{t}$ and $\mathbf{t}'$ of augmentation transform φ_t via a network $g_{\boldsymbol{\theta}_{\text{t}}}$ applied to both left and right channels (schematically shown only for the left channel in green).

in [40] that can be surely applied to our framework. Instead for fair comparison with SimCLR we will stay in the scope of the same self-supervised framework and try to explore another direction by investigating the role of latent space regularization via *estimation of parameters of applied augmentation transformation*. The pretext task regularization methods are not new and have been used in a standalone self-supervised architectures as described in Section 2. However, up to our best knowledge these regularization techniques have not been considered in the scope of contrastive representation learning. Thus a hypothesis to verify is whether creating more semantics about the inter-object or inter-scene associations would lead to more meaningful latent space representation.

In our study we define the parameters $\mathbf{t}$ of the augmentation transform $\varphi_{\mathbf{t}}$ under the pretext task estimation to be the rotation or jigsaw permutation. We used 4 rotation angles (0°, 90°, 180°and 270°) and 35 jigsaw permutations. We apply only one augmentation (either rotation or jigsaw

permutation) at time. These parameters are encoded as one-hot-encoding for each augmentation and the corresponding classifier $g_{\boldsymbol{\theta}_\mathrm{t}}$ is used to estimate them from the visual representation $\mathbf{h} = f_{\boldsymbol{\phi}_{\mathrm{Scat}}}(f_{\boldsymbol{\phi}_\mathrm{h}}(\tilde{\mathbf{x}}))$ extracted from the augmented view $\tilde{\mathbf{x}} = \varphi_\mathrm{t}(\tilde{\mathbf{x}})$ as shown in Figure 6.

The pretext task loss is defined as the parameters estimation loss between the applied parameters $\mathbf{t}$ and estimated ones $\hat{\mathbf{t}} = g_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{h})$:

$$\mathcal{L}_\mathrm{t}(\boldsymbol{\phi}_\mathrm{h}, \boldsymbol{\theta}_\mathrm{t}) = \frac{1}{2N} \sum_{i=1}^{2N} d(\mathbf{t}_i, g_{\boldsymbol{\theta}_\mathrm{t}}(\mathbf{h}_i)), \tag{2}$$

where $d(.,.)$ denotes the cross-entropy.

5. Final loss and training

We define ScatSimCLR loss similarly to SimCLR loss (1) with the only difference that instead of $\mathbf{h} = f_{\boldsymbol{\phi}_{\mathrm{ResNet}}}(\tilde{\mathbf{x}})$, we consider $\mathbf{h} = f_{\boldsymbol{\phi}_\mathrm{h}}(f_{\boldsymbol{\phi}_{\mathrm{Scat}}}(\tilde{\mathbf{x}}))$. Thus, the loss of ScatSimCLR is denoted as $\mathcal{L}_{\mathrm{ScatSimCLR}}(\boldsymbol{\phi}_\mathrm{h}, \boldsymbol{\theta}_\mathrm{z})$.

The final loss of ScatSimCLR with the pretext task regularization is defined as:

$$\mathcal{L}(\boldsymbol{\phi}_\mathrm{h}, \boldsymbol{\theta}_\mathrm{z}, \boldsymbol{\theta}_\mathrm{t}) = \mathcal{L}_{\mathrm{ScatSimCLR}}(\boldsymbol{\phi}_\mathrm{h}, \boldsymbol{\theta}_\mathrm{z}) + \lambda \mathcal{L}_\mathrm{t}(\boldsymbol{\phi}_\mathrm{h}, \boldsymbol{\theta}_\mathrm{t}), \tag{3}$$

where λ controls the relative weight of the second loss term.

The parameters estimation is based on the minimization problem:

$$(\hat{\boldsymbol{\phi}}_\mathrm{h}, \hat{\boldsymbol{\theta}}_\mathrm{z}, \hat{\boldsymbol{\theta}}_\mathrm{t}) = \underset{(\boldsymbol{\phi}_\mathrm{h}, \boldsymbol{\theta}_\mathrm{z}, \boldsymbol{\theta}_\mathrm{t})}{\operatorname{argmin}} \mathcal{L}(\boldsymbol{\phi}_\mathrm{h}, \boldsymbol{\theta}_\mathrm{z}, \boldsymbol{\theta}_\mathrm{t}), \tag{4}$$

in practical implementation for the first 40 epochs we assume $\lambda = 0$ in (3) and then $\lambda = 0.3$ for the rest. We have noticed that the network converges better, if it is pre-trained with only contrastive loss at the beginning. The parameter λ is selected to equalize the amplitude of contrastive and cross-entropy losses.

6. Experimental results

In this section, we evaluate ScatSimCLR performance on several datasets in the image classification task. At first, the proposed model is pretrained on a particular dataset based on (4) using unlabeled data and then a logistic one-layer classifier is applied to the learned representation to map it to the class labels encoded based on one-hot-vector encoding.

Datasets. The experimental evaluation is performed on STL-10 [1] and CIFAR100-20 [23] datasets. The experiments aim at investigating the impact of ScatSimCLR architecture and image augmentations on the classification performance. The results are reported as a top-1 result from 5 different runs.

Table 1. Impact of scale J and rotations L parameters of ScatNet on the classification accuracy after 5 epochs.

J	L	Accuracy STL-10	Accuracy CIFAR100-20
1	4	61.90%	36.88%
1	8	62.75%	39.43%
1	12	63.00%	40.56%
1	16	63.70%	41.72%
2	4	63.12%	44.52%
2	8	63.71%	46.09%
2	12	63.34%	46.25%
2	16	**64.03%**	**46.73%**
3	4	60.10%	42.50%
3	8	60.80%	43.85%
3	12	60.90%	44.01%
3	16	61.20%	44.59%
4	4	45.12%	34.39%
4	8	46.58%	35.00%
4	12	48.10%	35.96%
4	16	49.91%	36.74%

6.1. Impact of ScatSimCLR parameters

6.1.1 Impact of scaling and rotation channels

In this section, we investigate the impact of ScatNet parameters on the overall performance of ScatSimCLR. We use two datasets STL-10 and CIFAR100-20 with the images of size 96x96 to fit ScatNet. It should be noted that CIFAR100-20 is up-sampled from the size 32x32 to 96x96 using LANCZOS interpolation [24]. The system is trained with respect to the contrastive loss $\mathcal{L}_{\mathrm{ScatSimCLR}}(\boldsymbol{\phi}_\mathrm{h}, \boldsymbol{\theta}_\mathrm{z})$ and with adapter network fixed to 12 ResBlock layers and fixed depth of ScatNet to be 2. The pretext task loss was not used and the training was performed for the first 5 epochs only to reflect the dynamics of learning.

We have considered a range of ScatNet scaling parameters J from 1 to 4. We experimentally established that for the current architecture of ScatNet applied to the investigated datasets with the images of size 96x96, the best scaling parameter J is 2 as shown in Table 1. It should be pointed out that the increase of the scaling leads to the usage of larger filter sizes. As a consequence, the size of resulting images on the output of ScatNet, representing the feature vector, decreases. In turns, this represents a trade-off between the desirable robustness to the scaling and undesirable loss of details in the produced images. This might explain the optimality of the scaling factor J=2 as opposed to J=4.

Table 1 also demonstrates the impact of rotation parameter L on the classification performance for the considered scale factors J. The investigation of the rotation parameter L was performed in the range from 4 till 16 with the step size equals to 4 for each scale factor. For both considered datasets, the increase of the number of rotations clearly

Table 2. Impact of the number of layers in the adapter network of ScatSimCLR on STL-10 dataset for 1000 epochs.

Num. of layers	Num. of parameters	Accuracy STL-10
8	6.1M	76.47%
12	7.8M	83.53%
16	10.4M	84.01%
30	14.1M	84.76%

leads to the increase of the classification performance that can be explained by the increase of the rotation invariance in the produced feature space. In contrast to the scaling, the increase of the rotation factor L preserves the dimensionality of the produced feature map for a given fixed scaling and only leads to the increase of the number of rotation channels in the network output. This might explain the increase of the rotation parameter leads to overall performance enhancement.

6.1.2 Impact of the number of layers in the adapter network

In this section, we investigate the impact of the adapter network parameters on the classification accuracy. The experiments are performed on the dataset STL-10 with the image size 96x96. The training loss is defined by (3). As the pretext task network we used a classifier consisting of two fully-connected layers followed by the traditional dropout and ReLu activation. The last layer activation is softmax. ScatNet parameters were chosen according to the best results of section 6.1.1, i.e., J=2 and L=16.

We investigate the adapter network with the different number of layers, namely 8, 12, 16 and 30. ScatSimCLR was trained during 1000 epochs for each considered adapter network. The results presented in Table 2 are obtained as the top-1 results on the validation set. The obtained results clearly indicate that the increase of the adapter network complexity increases the performance in the classification task.

6.2. Ablations

6.2.1 Regularization ablations

In this section we investigate the impact of the regularization techniques. We compare the performance of the model trained with and without pretext task based on the estimation of augmentation transform: rotation and jigsaw estimation. We run experiment with all models presented in Table 2 and ScatSimCLR based on ResNet18 to compare the performance in a function of model complexity. ScatNet parameters were chosen according to the best results presented in 6.1.1. We use the STL-10 dataset for our comparison experiments. As the pretext task estimator we used a classifier consisting of two fully-connected layers with ReLU activation and the softmax at the end. We trained each model for 100 epochs. The batch size differs depending on the size of the model.

Table 3. Impact of the pretext task regularization on the classification accuracy on STL-10 dataset.

	Accuracy on STL-10			
Baseline model	Without pretext	With pretext		Num. of paramers
		Rotation	Jigsaw	
ScatSimCLR 8	74.78%	**77.86%**	76.36%	6.1 M
ScatSimCLR 12	76.57%	**78.43%**	77.78%	7.8 M
ScatSimCLR 16	77.03%	**78.5%**	77.91%	10.5 M
ScatSimCLR 30	77.86%	**79.11%**	78.4%	14.1 M
SimCLR (ResNet18)	71.90%	**76.36%**	75.22%	11.5 M

As it is shown in Table 3, the introduction of the pretext task improves the classification accuracy for both considered models: (i) ScatNet based SimCLR and (ii) vanilla SimCLR. For all models, rotation augmentation pretext task provides higher increase in classification performance in comparison to jigsaw. It can be explained by the fact that in the process of jigsaw pretext task, an image is split into 9 patches without an intersection, and then each patch is resized using Lanczos interpolation, so they fit the network input size. Applying interpolation introduces some artifacts. In the considered pretext task based on rotating by 90, 180 and 270 degrees the interpolation is not applied as such.

Therefore the introduction of pretext task regularization improves the classification performance of the models trained with contrastive loss. The proposed ScatSimCLR8 with 6.1 M of parameters outperforms SimCLR (ResNet18) with 11.5 M of parameters for all considered pretext tasks and also without pretext task. This confirms the importance of geometrical invariance of ScatNet.

6.2.2 Ablations of image augmentations

In this section we investigate the impact of image augmentations on the classification performance. We use the STL-10 dataset with image size 96x96. To exclude the impact of batch size and other model hyperparameters, we use the fixed setup with batch size = 256, ScatNet parameters: J=2, L=16 and depth=2. We study *(i) geometric transformations*: random cropping, horizontal flipping and random affine transformations and *(ii) color transformations*: color jitter, Gaussian blur and grayscaling. We tried to investigate the effect of augmentation ablation considering different combinatorics of augmentations.

The obtained results are shown in Figure 7. The baseline system performance is shown by the green bar. The baseline uses all considered augmentation similarly to SimCLR. It is interesting to point out that the removal of affine transformation augmentations leads to the performance enhancement with about 2% with respect to the baseline system with all considered augmentations. This is an important result confirming the invariance of ScatNet to geometrical trans-

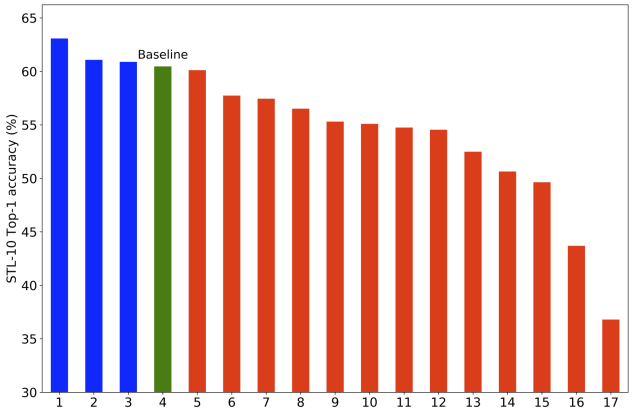

Figure 7. Impact of removing the augmentations on the performance of ScatSimCLR for STL-10: "Baseline" denotes ScatSimCLR trained with all augmentations (cropping, flipping, color, grayscale, Gaussian blur and affine augmentations). The following labels denote: 1 - the baseline without the affine augmentations; 2 - only cropping and color augmentations; 3 - the baseline without the horizontal flipping; 5 - the baseline without Gaussian blur augmentations; 6 - the baseline without cropping and Gaussian blur augmentations; 7 - the baseline without color and Gaussian blur augmentations; 8 - the baseline without grayscale and Gaussian blur augmentations; 9 - the baseline without cropping and grayscale augmentations; 10 - the baseline without color augmentations; 11 - the baseline without cropping augmentations; 12 - the baseline without grayscale augmentations; 13 - only cropping augmentations; 14 - the baseline without color and grayscale augmentations; 15 - only color augmentations; 16 - the baseline without crop and color augmentations; 17 - no augmentations.

formations. Therefore, these augmentations can be further excluded from training. In turns, it might lead to the lower complexity of training under a smaller number of augmentations. The next interesting result is obtained when the only image cropping and color transformations were used as the augmentations. It leads to about 0.5% enhancement over the baseline system. Finally, the same enhancement is observed when the flipping was removed from the baseline augmentations. The performance of ScatSimCLR without any augmentations is about 24% lower with respect to the baseline system.

Summarizing the obtained results, we can conclude that the most important augmentations for ScatSimCLR are cropping and color ones.

6.2.3 Comparison with the state-of-the-art

We compare the results obtained for the proposed ScatSimCLR on STL-10 [1] and CIFAR100-20 [23] with the state-of-the-art results reported in ADC [15], DeepCluster [7], DAC [9], IIC [21], TSUK [16], SCAN [40], RUC [34] and SimCLR[10] on the Figures 1 and 8.

Figures 1 and 8 show the performance of image classification using ScatSimCLR with the linear evaluation layer. We compare the model performance not only in terms of classification accuracy but also in terms of number of trainable parameters. For the STL-10 dataset as shown on Figure 1, we not only achieve SOTA classification accuracy but also our model achieves better performance, compared to previous SOTA [40] with only a half of its parameters. The same tendency is shown for CIFAR100-20 [23] dataset on Figure 8; all proposed ScatSimCLR models achieve SOTA classification accuracy: 58.0% , 59.4% and 63.8%, with 7M, 10.4M and 14M parameters respectively, while previous SOTA RUC [34] achieves 54.3% with 14M trainable parameters.

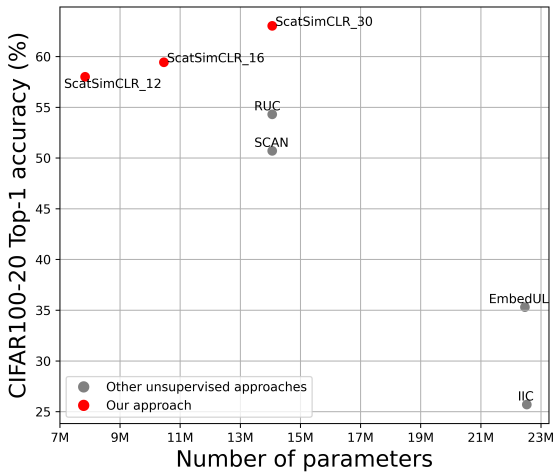

Figure 8. CIFAR100-20 Top-1 accuracy of self-supervised methods. Gray dots indicate other self-supervised methods. ScatSimCLR is shown in red.

7. Conclusion and discussions

In this paper, we address the problem of self-supervised learning for small dataset problems. More particularly, we answer the question whether the complex encoding network used for the contrastive learning can be partially replaced by the simpler hand-crafted network ensuring geometric invariance.

We demonstrate that the proposed model based on geometrically invariant ScatNet with reduced number of trainable parameters can achieve the state-of-the-art performance on STL-10 and CIFAR100-20 datasets.

We demonstrate that introduction of pretext task regularization based on the estimation of augmentation transform improves the performance of the proposed ScatSimCLR models as well as SimCLR with ResNet.

We demonstrate that by using a geometrically invariant ScatNet model, we are able to reduce the great portion of

augmentations used to simulate the geometrical transformations at the training. Also, we confirm that the main benefit in the considered contrastive learning comes from the color and cropping augmentations. This indicates that a promising direction in further reduction of the number of augmentations is to use more efficient color coding schemes and to introduce local windowed encoding in contrast to the whole image encoding considered in the paper.

The performed extensive experiments explain the architectural and design particularities of the considered approach. The obtained results represent the state-of-the-art performance on several datasets among the networks with the same number of parameters.

Acknowledgement

This research was partially funded by the Swiss National Science Foundation SNF project CRSII5_193716.

References

[1] Andrew Ng Adam Coates and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[2] Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andén, Eugene Belilovsky, et al. Kymatio: Scattering transforms in python. *Journal of Machine Learning Research*, 21(60):1–6, 2020.

[3] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545, 2019.

[5] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. *arXiv preprint arXiv:1704.05310*, 2017.

[6] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[7] 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.

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[9] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888, 2017.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[11] 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.

[12] 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.

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

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[15] Haeusser, Plapp, Golkov, Aljalbout, and Cremers. Associative deep clustering: Training a classification network with no labels. In *Pattern Recognition*, pages 18–32. Springer International Publishing, 2019.

[16] Sungwon Han, Sungwon Park, Sungkyu Park, Sundong Kim, and Meeyoung Cha. Mitigating embedding and class assignment mismatch in unsupervised image classification. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 768–784. Springer International Publishing, 2020.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[18] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[19] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[20] Simon Jenni and Paolo Favaro. Self-supervised feature learning by learning to spot artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2733–2742, 2018.

[21] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019.

[22] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[23] Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[24] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of research of the National Bureau of Standards*, 45:255–282, 1950.

[25] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6874–6883, 2017.

[26] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR*, volume 119, 2020.

[27] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[28] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9339–9348, 2018.

[29] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[30] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017.

[31] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9359–9367, 2018.

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[33] Edouard Oyallon, Sergey Zagoruyko, Gabriel Huang, Nikos Komodakis, Simon Lacoste-Julien, Matthew Blaschko, and Eugene Belilovsky. Scattering networks for hybrid representation learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2208–2221, 2018.

[34] Sungwon Park, Sungwon Han, Sundong Kim, Danu Kim, Sungkyu Park, Seunghoon Hong, and Meeyoung Cha. Improving unsupervised image clustering with robust learning. *arXiv preprint arXiv:2012.11150*, 2020.

[35] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[36] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–771, 2018.

[37] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016.

[38] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[39] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[40] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pages 268–285. Springer, 2020.

[41] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[42] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 6210–6219, 2019.

[43] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[44] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.