Multi-level Contrastive Learning for Self-Supervised Vision Transformers

Shentong Mo[1], Zhun Sun[2,*], Chao Li[3]
[1]Carnegie Mellon University,[2]Tohoku University
[3]Center for Advanced Intelligence Project (AIP), RIKEN
shentonm@andrew.cmu.edu,zhunsun@gmail.com,chao.li@riken.jp

Abstract

*Recent studies aim to establish contrastive self-supervised learning (CSL) algorithms specialized for the family of Vision Transformers (ViTs) to make them function normally as ordinary convolutional-based backbones in the training progress. Despite obtaining promising performance on related downstream tasks, one compelling property of the ViTs is ignored in those approaches. As previous studies have demonstrated, vision transformers benefit from the early stage global attention mechanics, obtaining feature representations that contain information from distant patches, even in their shallow layers. Motivated by this, we present a simple yet effective framework to facilitate the self-supervised feature learning of transformer based vision architectures, namely, **M**ulti-level **C**ontrastive learning for **V**ision **T**ransformers (**MCVT**). Specifically, we equip the vision transformers with individual-based (InfoNCE) and prototypical-based (ProtoNCE) contrastive loss in different stages of the architecture to capture low-level invariance and high-level invariance between views of samples, respectively. We conduct extensive experiments to demonstrate the effectiveness of the proposed method, using two well-known vision transformer backbones, on several vision downstream tasks, including linear classification, detection, and semantic segmentation.*

1. Introduction

Recently, Transformer [29] has become the new standard module in designing backbone architectures for vision tasks. The family of Vision Transformers (ViTs) [13, 28, 21, 35] have achieved superior performance compared to Convolutional Neural Networks (CNNs) in image classification [13, 28, 21, 35], object detection [3, 11], semantic segmentation [36], *etc*. During the same period, self-supervised learning frameworks [6, 9, 16, 15] have shown their successes in utilizing quantities of unlabeled data. It is a nature idea to combine them together, [10, 33, 5] provide several initial attempts. By introducing ad hoc tricks or specializing in the backbone, they replace the convolutional backbones with the family of ViTs, yet yield in-degraded performance in the downstream tasks.

The global attention mechanics is considered to be the most important property in vision transformers [5], with which they encourage the "local-to-global" correspondence, leading to the effectiveness of self-supervised learning of vision transformers. Meanwhile, a recent study [25] demonstrates that the early-stage global attention employed in the shallow layers could also help the vision transformers obtain feature representations that contain information from distant patches.

Motivated by these results, we explore the potential of learning a vision transformer in the self-supervised approach with low-level feature representations. To achieve this, we start by appending auxiliary InfoNCE loss [26, 31, 24] to the early stages of vision transformers. As a result, we observe consistently improved performances in the downstream classification tasks, which confirms the ability to capture instance-wise low-level feature invariance in the early stages of vision transformers. We then further examine the possibility of using prototypical contrastive losses (ProtoNCE) [19] to impose high-level (semantic) feature invariance. Specifically, we introduce three types of multi-level contrastive vision transformers, with InfoNCE and ProtoNCE attached to different stages of the backbones (See Section 3.4 for details). We empirically find that the variant that captures prototype-wise invariance using features from later stages while preserving instance-wise invariance using early stages features obtains superior performance.

We present our findings as a simple yet effective framework for training the family of vision transformers in self-supervised styles. Namely, **M**ulti-level **C**ontrastive learning for self-supervised **V**ision **T**ransformers (**MCVT**). The overall framework is shown in Figure 1 and Figure 2. Concretely, we project the class token in the early/late stages of the vision transformer onto embedding spaces through multi-layer perceptrons (MLPs). We simply use the global

*Corresponding author.

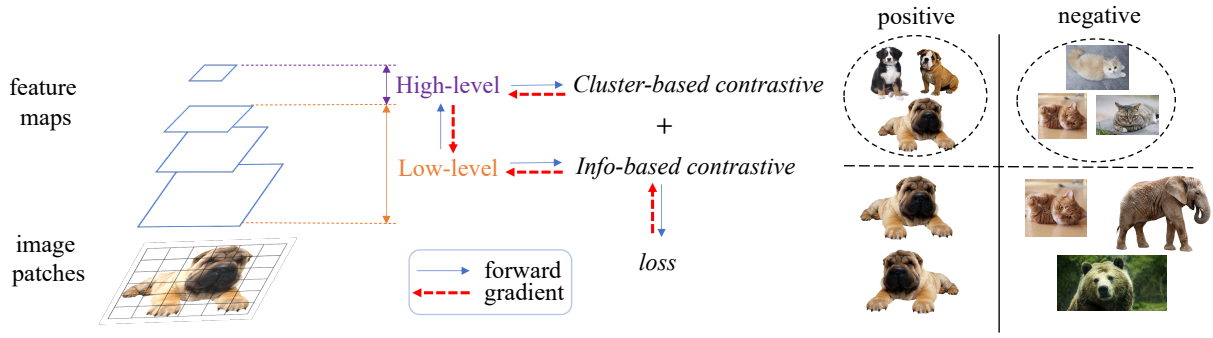

Figure 1. Illustration of our **M**ulti-level **C**ontrastive **V**ision **T**ransformers (MCVT) scheme, where info-based and cluster-based contrastive losses are tailored for low-level and high-level features, respectively. In this manner, low-level and high-level feature in-variances are iteratively captured during pre-training.

average pooling of the feature representations for the backbones that do not employ class tokens. We term the embedded features representations as *low-level* and *high-level* features, respectively. We then apply InfoNCE loss to low-level features and ProtoNCE loss to high-level features, which we term with *low-level* contrastive loss and *high-level* contrastive loss, respectively.

We pre-train our MCVT frameworks with two widely-used vision transformer backbones (ViT [13] and Swin [21]) on the ImageNet100, ImageNet-1K [12], where we evaluate the pre-trained models on the two benchmarks for image classification. We also transfer the models pre-trained on ImageNet-1K to downstream vision tasks, using the MS-COCO and ADE20K benchmark datasets for evaluating their performance on object detection, instance segmentation, and semantic segmentation.

In the ablation studies, we first draw the similarity between low- and high-level representations using the CKA heat map proposed in [25]. Then, we reveal that the Swin [21] transformer pre-trained with our proposed MCVT approach behaves more similarly to a full-supervised optimized one than the MoBY [33] approach. We further investigate several variants of the MCVT by manipulating the attached loss term at each stage. In the end, we vary the crucial hyper-parameters such as batch size and the number of clusters and show that the performance is undegraded within a wide range.

To summarize, in this study, our main contributions are recapped as follows:

- We investigate the effects of low-level features from earlier stages of a vision transformer in the contrastive self-supervised learning algorithm. We utilize the low-level features in both the instance-wise and prototypical manner for the investigation.
- Based on the observation, we propose a simple yet effective framework with multi-level contrastive learning for self-supervised vision transformers, which we term MCVT.
- In the experimental analysis, we show the proposed MCVT framework benefits vision transformers of different architectures in different downstream vision tasks.
- We also show that the representations learned through the MCVT framework is closer to those learned with a fully-supervised style, further revealing the effectiveness brought by utilizing the low-level features.

2. Related Work

2.1. Vision Transformer

In recent years, vision transformers [13, 28, 21, 35, 14] have gained many researcher's interests in various downstream tasks, such as image classification, object detection, and segmentation. Typically, Dosovitskiy *et al.* [13] first applied a pure transformer directly to the sequences of input image patches with dimension 16×16. A teacher-student strategy was further proposed in DeiT [28] to reduce training parameters and costs, in which a distillation token was leveraged to make the student learn from the teacher through attention. More recently, Swin transformer [21] introduced a hierarchical architecture with shifted windows in the attention modules to learn non-overlapping local information and cross-window connection, which achieves state-of-the-art results on various benchmarks. In this work, we mainly focus on self-supervised vision transformers with multi-level contrastive learning to improve the quality of pre-trained representations. Our approach is orthogonal to these vision transformers and can be easily applied to these backbones to learn a better pre-trained model.

2.2. Self-supervised Learning

Self-supervised methods [31, 6, 7, 15, 16, 8, 9, 37, 4, 19, 30, 22, 23] often apply pretext tasks to train a model by mining the internal characteristics of data without any

label. In the early period, the instance-level noise contrastive estimation was proposed in NIPD [31] to deal with the non-parametric classification problems. After that, the instance-wise contrastive learning was widely used in a lot of work [6, 7, 15, 16, 8, 9, 37]. Typically, MoCo [16] was introduced with a momentum encoder to maintain negative samples from a large and consistent dictionary on the fly. A Siamese network and a stop-gradient operator were leveraged in SimSiam [9] to achieve satisfactory results without the momentum encoder and large batch size. On top of instance-level contrastive learning, some work [19, 4, 30] adopt cluster-based contrastive learning to pull representations closer to their assigned prototypes and far from other prototypes. However, in this work, we take advantage of low-level and high-level features from vision transformers for self-supervised learning. Multi-level contrastive learning is proposed to capture low-level and high-level invariances between views from various stages of the vision transformer.

2.3. Self-supervised Vision Transformer

Recently, self-supervised vision transformers [10, 33, 5] have addressed people's attention due to their strong performance on various downstream tasks. Specifically, MoCov3 [10] extended the MoCo [16] method to ViT [13] for minimizing the distance between representations of two augmented views. MoCo v2 and BYOL were applied simultaneously in MOBY [33] to form a self-supervised framework based on the Swin [21] backbone. In DINO [5], knowledge distillation was combined with momentum encoder and multi-crop training for learning the local-to-global correspondence in the vision transformer. However, they only capture the single-scale feature representation from the global view for pre-training.

As proven to be effective in a previous study [25], vision transformers can obtain global representations from shallow layers. Therefore, it is desirable to take into account low-level features from the shallow stage for learning more fine-grained invariances. One concurrent work, MST [20], applied a masked token strategy to the multi-head self-attention map in both the student and teacher network to capture the local context of an image while preserving the global semantic information. Another concurrent work, BEiT [1], proposed a masked image modeling task to recover the original visual tokens based on the corrupted image patches. In this work, we leverage low-level feature invariances from shallow layers and high-level feature invariances from deep layers. We are also the first to simultaneously leverage info-based and cluster-based contrastive learning in self-supervised vision transformers to pre-train better representations.

3. Method

In this section, we propose a simple yet effective framework with Multi-level Contrastive learning for self-supervised Vision Transformers, namely **MCVT**, as shown in Figure 2. First, we begin with the formal problem setup for pre-training a self-supervised vision transformer, and list all notations for easier reading. Then, we elaborate on the process of extracting low-level and high-level features from vision transformers with MLP-based projection heads. Finally, we present the technical details of our MCVT for self-supervised vision transformers, where three types of MCVT variants are introduced.

3.1. Problem Setup

We closely follow the problem setup in previous self-supervised vision transformers [10, 33, 5]. Thus, our work aims to pre-train a vision transformer backbone with more meaningful representations for achieving good performance on downstream tasks. To explain the problem in a unified manner, we define notations as follows.

Notations. Given a set of training examples $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, we apply a vision transformer backbone $f(\cdot)$ to generate global-view representations $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n\}$, *i.e.,* $\mathbf{v}_i = f(\mathbf{x}_i)$. Suppose the vision transformer backbone $f(\cdot)$ is composed of s transformer blocks, that is stages. For example, there are four transformer blocks in the Swin [21] transformer. In this case, $s = 4$. For each training example $\mathbf{x}_i$, we use $f(\cdot)$ to generate low-level representations $\mathcal{U}_i = \{\mathbf{u}_i^1, \mathbf{u}_i^2, \cdots, \mathbf{u}_i^s\}$ from each stage s, where $i \in [1, n]$. Note that the features from the last stage is the global-scale, that is, $\mathbf{u}_i^s = \mathbf{v}_i$. A set of projection heads $\mathcal{G} = \{g_1, g_2, \cdots, g_s\}$ are applied on $\mathcal{U}_i$ to generate low-dimension features $\mathcal{H}_i = \{\mathbf{h}_i^1, \mathbf{h}_i^2, \cdots, \mathbf{h}_i^s\}$ for contrastive learning.

3.2. High-level Feature Invariance

Similar to the non-transformer based pre-training frameworks [16, 8, 6, 7], we extract the high-level features from the last stage of vision transformers. Given a training example $\mathbf{x}_i$, we take two augmented views $\mathbf{x}_i$ and $\mathbf{x}_i'$ for each image $\mathbf{x}_i$ under a set of random data augmentations $\mathcal{T}$. Then two views are fed into two vision transformer backbones $f(\cdot)$ to generate the high-level features $\mathbf{v}_i$ and $\mathbf{v}_i'$, that is, $\mathbf{u}_i^s$ and $(\mathbf{u}_i^s)'$, where s denotes the number of stages in the transformers. Finally, we apply a MLP-based projection head g_s to project $\mathbf{u}_i^s$ and $(\mathbf{u}_i^s)'$ into a low-dimensional embedding $\mathbf{h}_i^s$ and $(\mathbf{h}_i^s)'$. In order to capture the high-level invariance between features $\mathbf{h}_i^s$ and $(\mathbf{h}_i^s)'$ from the final stage, we consider $\mathbf{h}_i^s$ and $(\mathbf{h}_i^s)'$ as the high-level features in this case.

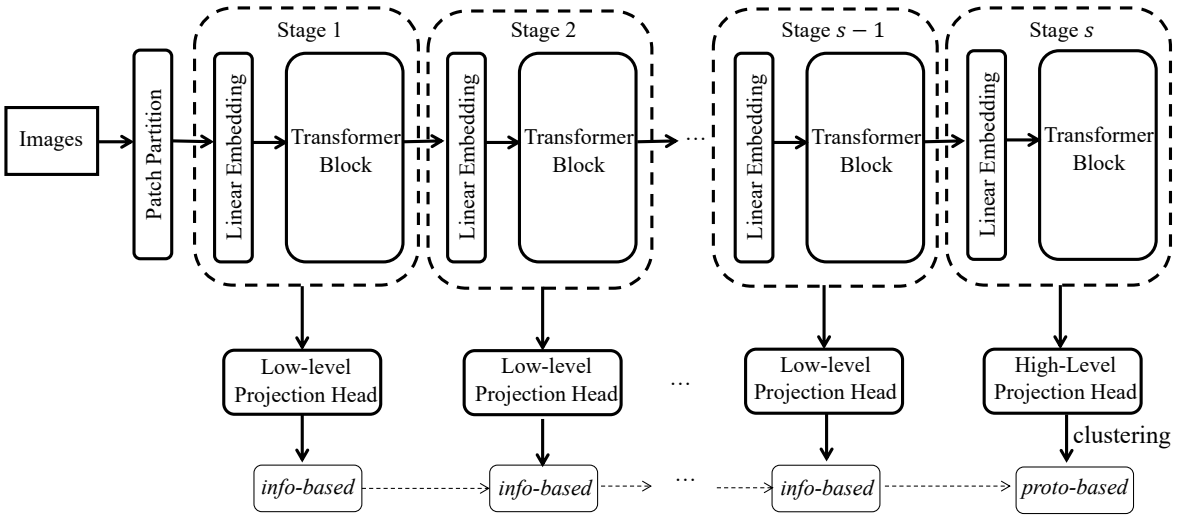

Figure 2. Illustration of our Multi-level Contrastive Vision Transformers (MCVT) scheme. Specifically, we generate early-stage features of image patches from shallow layers and later-stage features from deep layers. Then, early-stage and high-level projection heads composed of multi-layer perceptrons (MLPs) are leveraged to project features to low-dimension embeddings for multi-level contrastive learning in terms of low-level and high-level views. The *low-level* contrastive loss is calculated in terms of the mutual information of low-level features to capture the instance-wise invariance between views, while the *high-level* contrastive loss is employed on the mutual information of high-level features to learn the prototype-wise invariance between global views.

3.3. Low-level Feature Invariance

Motivated by previous study [25] which has shown that vision transformers can learn global representations from shallow layers, we take into account low-level features from the shallow stages for learning more fine-grained invariances. Specifically, we apply a set of projection heads $\mathcal{G} = \{g_1, g_2, \cdots, g_s\}$ on the low-level features $\mathcal{U}_i = \{\mathbf{u}_i^1, \mathbf{u}_i^2, \cdots, \mathbf{u}_i^s\}$ and $\mathcal{U}_i' = \{(\mathbf{u}_i^1)', (\mathbf{u}_i^2)', \cdots, (\mathbf{u}_i^s)'\}$ from the shallow stages for two augmented views $\mathbf{x}$ and $\mathbf{x}'$. For learning the low-level invariance between features $\mathcal{U}_i$ and $\mathcal{U}_i'$ from the shallow stages, we consider $\mathcal{U}_i$ and $\mathcal{U}_i'$ as the low-level feature in this case. It is worth mentioning that the low-level features $\mathbf{u}_i^s$ and $(\mathbf{u}_i^s)'$ from the last stage indeed represent the high-level features.

3.4. Multi-level Contrastive Vision Transformer

In this part, we are inspired by previous non-transformer contrastive learning studies [16, 8, 15] and introduce three types of multi-level contrastive vision transformers. Firstly, we apply the info-based normalized cross-entropy loss on both low-level and high-level features to capture the instance-wise invariance together, which we call *MCVT-info*. Then, we use the cluster-based normalized cross-entropy loss on both low-level and high-level features to learn the proto-wise invariance simultaneously, which we call *MCVT-proto*. Finally, we define the low-level contrastive loss with the mutual information of low-level features to capture the instance-wise invariance between views. Meanwhile, we employ the high-level contrastive loss on the mutual information of high-level features to discriminate the proto-wise invariance between global views. This type of MCVT is denoted as *MCVT-mix*.

MCVT-info. Following previous momentum-based contrastive learning frameworks [16, 8], we input two augmented views $\mathbf{x}_i$ and $\mathbf{x}_i'$ for each image $\mathbf{x}_i$ under a set of random data augmentations $\mathcal{T}$. Then two views are fed into two vision transformer backbones $f(\cdot), f'(\cdot)$ and the set of projection heads $\mathcal{G}$ to generate the query features set $\mathcal{H}_i$ and the critical features set for $\mathcal{H}_i'$ for contrastive learning. The info-based MCVT loss is formulated as:

$$\mathcal{L}_{\text{MCVT-info}} = \sum_{t=1}^{s}\sum_{i=1}^{n} -\log \frac{\exp(\mathbf{h}_i^t \cdot (\mathbf{h}_i^t)'/\tau)}{\sum_{j=1}^{r}\exp(\mathbf{h}_i^k \cdot \mathbf{h}_j^t/\tau)} \quad (1)$$

where $\mathbf{h}_i^t, (\mathbf{h}_i^t)', \mathbf{h}_j^t$ represent the anchor, positive, and negative embedding from the stage index t for each training sample $\mathbf{x}_i$, and τ is a temperature hyper-parameter. r denotes the number of negative samples.

MCVT-proto. Inspired by previous non-transformer based contrastive learning frameworks [19, 4], we apply M times clustering to the representations $\mathcal{H}_i$ during pre-training, with the number of prototypes as $k_m, m \in \{1, 2, \cdots, M\}$. Therefore, we have a set of different number of prototypes $K = \{k_1, k_2, \cdots, k_M\}$. The prototypes of the samples using k_m clusters are marked as $\mathcal{C}^m = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{k_m}\}$. In this way, we define the objective of our *MCVT-proto* as:

Algorithm 1 MCVT-mix main learning algorithm

Input: Data $\mathcal{X}$, $f(\cdot), \mathcal{G}$, sets of augmentation $\mathcal{T}$.
Initialize the parameters of $f(\cdot), \mathcal{G}$
for each epoch **do**
  Obtain two view $\mathbf{x}_i, \mathbf{x}_i'$ with $\mathcal{T}$
  Encode features $\mathcal{U}_i, \mathcal{U}_i'$ with $f(\cdot)$
  Project features to $\mathcal{H}_i, \mathcal{H}_i'$ with $\mathcal{G}$
  for $t \leftarrow 1$ to $s-1$ **do**
    Compute the low-level loss in Eq. 3 w.r.t $\{\mathbf{h}_i^1, \mathbf{h}_i^2, \cdots, \mathbf{h}_i^{s-1}\}$
  end for
  for $m \leftarrow 1$ to M **do**
    Obtain prototypes $\mathbf{c}_{k_m}$ with K-means.
    Compute the high-level loss in Eq. 4 w.r.t $\mathbf{h}_i^s$
  end for
  Compute the overall loss in Eq. 5
end for
Output: $f(\cdot)$

$$\mathcal{L}_{\text{MCVT-proto}} = \sum_{t=1}^{s}\sum_{i=1}^{n} -\log \frac{\exp(\mathbf{h}_i^t \cdot (\mathbf{h}_i^t)'/\tau)}{\sum_{j=1}^{r}\exp(\mathbf{h}_i^k \cdot \mathbf{h}_j^t/\tau)} + \sum_{t=1}^{s}\sum_{i=1}^{n} -\frac{1}{M}\sum_{m=1}^{M}\log \frac{\exp(\mathbf{h}_i^t \cdot \mathbf{c}_p^m/\phi_p^m)}{\sum_{j=1}^{r}\exp(\mathbf{h}_i^t \cdot \mathbf{c}_j^m/\phi_j^m)} \tag{2}$$

where $\mathbf{h}_i^t$ denotes the anchor representation from the stage index t for each training sample i. $\mathbf{c}_s^m, \mathbf{c}_j^m$ are the positive prototype p that the sample i belongs to and the negative prototype j at m step. ϕ_p^m, ϕ_j^m are the concentration estimation indicator for the distribution of representations around the prototype p, j at the m step.

MCVT-mix. To discriminate the low-level and high-level features during pre-training, we propose a self-supervised approach with a multi-level contrastive vision transformer (MCVT) by info-based contrastive leaning of features from shallow layers and cluster-wise contrastive leaning of features from deep layers. Specifically, we calculate the low-level contrastive loss $\mathcal{L}_{low}$ with the info-based normalized cross-entropy loss with respect to the low-level features $\{\mathbf{h}_i^1, \mathbf{h}_i^2, \cdots, \mathbf{h}_i^{s-1}\}$. The high-level contrastive loss $\mathcal{L}_{high}$ is defined with the cluster-based normalized cross-entropy loss in terms of the high-level features $\mathbf{h}_i^s$. Thus, the low-level contrastive loss $\mathcal{L}_{low}$, the high-level contrastive loss $\mathcal{L}_{high}$, and the overall objective of our MCVT-mix are formulated as follows:

$$\mathcal{L}_{low} = \sum_{t=1}^{s-1}\sum_{i=1}^{n} -\log \frac{\exp(\mathbf{h}_i^t \cdot (\mathbf{h}_i^t)'/\tau)}{\sum_{j=1}^{r}\exp(\mathbf{h}_i^k \cdot \mathbf{h}_j^t/\tau)} \tag{3}$$

$$\mathcal{L}_{high} = \sum_{i=1}^{n} -\frac{1}{M}\sum_{m=1}^{M}\log \frac{\exp(\mathbf{h}_i^s \cdot \mathbf{c}_p^m/\phi_p^m)}{\sum_{j=1}^{r}\exp(\mathbf{h}_i^s \cdot \mathbf{c}_j^m/\phi_j^m)} \tag{4}$$

$$\mathcal{L}_{\text{MCVT-mix}} = \mathcal{L}_{low} + \lambda \cdot \mathcal{L}_{high} \tag{5}$$

In this manner, the low-level contrastive loss is applied to capture the fine-grained instance-wise invariance between augmented views, while the high-level contrastive loss is employed to learn cluster-wise invariance between global views. The overall algorithm is summarized as in Algorithm 1.

4. Experiments

4.1. Datasets & Configurations

Following previous methods [10, 33, 5], we use four benchmarks for comparison, including ImageNet-100 [27] and ImageNet-1K [12] for image classification, MS-COCO [34] for object detection, and ADE20K [38, 39] for semantic segmentation. During pre-training, we use data augmentation methods with random resize crop, random color jittering, random horizontal flip, and random grayscale. We train for 300 epochs and apply the first 20 epochs as a warm-up step by only using the InfoNCE loss. The initial learning rate is set to 5e-4, and we use a cosine scheduler to multiply it with a decay rate of 0.1 for every 30 epochs. AdamW optimizer is used with a weight decay of 0.05, a momentum of 0.9, and a batch size of 512. We adopt the faiss-GPU [17] library for k-means clustering during the pre-training.

ImageNet-100. For pre-training, we set number of clusters K = 2500,5000,10000, r = 1024. For linear classification, we train a linear classifier on the frozen backbone weights. We train it for 100 epochs and use the first 5 epochs as a warm-up stage. We apply SGD as our optimizer with a base learning rate of 1.0, a momentum of 0.9, and a weight decay of 0.

ImageNet-1K. For pre-training, we set number of clusters K = 25000,50000,100000, r = 16000. For linear classification, we follow the same setting as ImageNet-100. For end-to-end fine-tuning, we initialize the network with the pre-trained weights and adapt them for fine-tuning.

MS-COCO. We closely follow previous work [10, 33, 5], and adopt the Cascade Mask R-CNN [2] as the detector. The Swin-T [21] backbone weights are pre-trained on ImageNet-1K using our MCVT. Other settings are the same as the implementation in this work [21] except that we use a 1× schedule.

ADE20K. Following the settings in [33, 21], we use the UPerNet approach [32] based on our ImageNet-1K pre-trained Swin-T for evaluation. We fine-tune the detector

Table 1. Comparisons between MoBY and three types of MCVT variants with various transformer architectures (ViT-S and Swin-T) under the linear classification evaluation on the ImageNet-100 dataset.

Method	Arch.	Param.(M)	Batch	Epochs	Top-1 (%)	Top-5 (%)
MoBY	ViT-S	22	512	300	86.28	97.08
MCVT-info	ViT-S	22	512	300	87.79	97.69
MCVT-proto	ViT-S	22	512	300	81.05	95.27
MCVT-mix	ViT-S	22	512	300	**89.31**	**98.72**
MoBY	Swin-T	29	512	300	87.92	97.84
MCVT-info	Swin-T	29	512	300	89.45	98.78
MCVT-proto	Swin-T	29	512	300	82.53	95.82
MCVT-mix	Swin-T	29	512	300	**91.26**	**99.12**

with the same learning rate in [33, 21] for a fair comparison.

Table 2. Comparisons between our MCVT-mix and other methods with various transformer architectures (ViT and Swin) under the end-to-end fine-tuning and linear classification for evaluation on the ImageNet-1K dataset. * denotes that no multi-crop scheme is used.

Method	Arch.	Param.(M)	Batch	Epochs	Top-1 (%)
end-to-end fine-tuning:					
MoCo-v3	ViT-S	21	1024	300	81.4
DINO	ViT-S	21	1024	300	81.5
MCVT-mix	ViT-S	21	512	300	**81.7**
MoCo-v3	ViT-B	85	4096	300	83.2
DINO	ViT-B	85	1024	300	82.8
MCVT-mix	ViT-B	85	512	300	**83.4**
linear classification:					
MoCo v3	ViT-S	21	1024	300	72.5
DINO*	ViT-S	21	1024	300	72.5
MoBY	ViT-S	21	512	300	72.8
MCVT-mix	ViT-S	21	512	300	**73.1**
MoBY	Swin-T	29	512	100	70.9
MCVT-mix	Swin-T	29	512	100	71.6
MoBY	Swin-T	29	512	300	75.0
MCVT-mix	Swin-T	29	512	300	**75.3**

4.2. Experimental Results

In this part, we conduct extensive experiments by transferring our MCVT pre-trained backbone to various downstream tasks, including image classification, object detection, instance segmentation, and semantic segmentation for comprehensive analysis. To demonstrate the advantage of our approach, we compare it with existing self-supervised vision transformers, such as MoCo v3 [10], MoBY [33], and DINO [5].

ImageNet-100. Table 1 reports the comparison results between MoBY [33] and three types of MCVT variants using ViT-S [13] and Swin-T [21] in terms of linear classification. As can be seen, all our MCVT-info frameworks with ViT-S and Swin-T architectures achieve better performance than MoBY, which demonstrates the effectiveness of using early-stage features in the info-based low-level contrastive loss. Furthermore, applying our MCVT-mix to ViT-S outperforms the baseline by 3.03% and 1.64% in terms of top-1 and top-5 accuracy under the same setting of model size and pre-training epochs. In particular, our MCVT-mix with Swin-T achieves the best result, outperforming MoBY [33] by 3.34% and 1.28% in terms of top-1 and top-5 accuracy. This further shows the state-of-the-art advantage of our MCVT-mix frameworks for self-supervised vision transformers.

ImageNet-1K. We compare our MCVT-mix framework with previous self-supervised vision transformers [10, 5, 33] in Table 2 by using ViT and Swin architectures under the end-to-end fine-tuning and linear classification for comprehensive evaluation. We can observe that our MCVT-mix frameworks outperform previous methods in terms of all architectures with various model sizes. The performance gain (+0.6%) achieved by our MCVT-mix under the linear classification is more significant than the gain (+0.3%) under the setting of end-to-end fine-tuning. This demonstrates the effectiveness of our method in learning better representations during pre-training. In the meanwhile, compared to MoBY [33], pre-training for 100 epochs achieves a better performance gain (+0.7%) than the gain (+0.3%) of pre-training 300 for epochs. This is because with low-level and high-level invariances learned in our approach, we achieve faster convergence speed and perform better at the first 100 epochs for linear probing evaluation.

MS-COCO. In Table 3, we report the comparison results of object detection and instance segmentation by fine-tuning Cascade Mask R-CNN [2] based on Swin-T pre-trained by three types of our MCVT frameworks. In terms of object detection, our MCVT-info method consistently performs better than baselines due to the self-supervision of low-level invariances involved in the early stage of vision transformers. Besides, our MCVT-mix framework achieves even better performance than the supervised baseline, which shows the effectiveness of our approach in pre-training meaningful representations. Also, when transferred to instance segmentation, our MCVT-mix framework achieves better re-

sults than the supervised baseline and MoBY [33]. This further verifies the superiority of our MCVT methods in self-supervised vision transformers.

Table 3. Comparison results of object detection and instance segmentation fine-tuned on COCO with Cascade Mask R-CNN based on Swin-T. AP^b and AP^m denote the metrics for the bounding box and the mask, respectively. Bold numbers indicate the first place.

Method	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Supervised	48.1	67.1	52.2	41.7	64.4	45.0
MoBY	48.1	67.1	52.1	41.5	64.0	44.7
MCVT-info	48.2	67.1	52.3	41.7	64.1	44.9
MCVT-proto	47.3	66.3	51.2	40.7	63.1	43.9
MCVT-mix	**48.6**	**67.6**	**52.5**	**42.1**	**64.5**	**45.3**

ADE20K. Table 4 compares our MCVT variants with MoBY [33] under the same setting by fine-tuning our pre-trained Swin-T on the ADE20K benchmark, where the mIoU metric is reported. As can be seen, our MCVT-mix framework achieves better performance than the self-supervised baseline, which shows the advantage of our approach to self-supervised vision transformers. Furthermore, we have a smaller gap between ours and the supervised baseline than MoBY. This also validates the effectiveness of using low-level and high-level invariances as self-supervision.

Table 4. Comparison results of semantic segmentation fine-tuned on ADE20K. mIoU denotes the mean mean intersection-over-union averaged across classes for the ADE20K validation set. Bold and underline denote the first and second place.

Method	Backbone	Schedule	mIoU
Supervised	Swin-T	160K	**45.81**
MoBY	Swin-T	160K	45.58
MCVT-info	Swin-T	160K	45.62
MCVT-proto	Swin-T	160K	45.01
MCVT-mix	Swin-T	160K	<u>45.76</u>

Visualizations of representation similarity To verify the effectiveness of our MCVT pre-trained model, we quantitatively evaluate the representation structure within and across different stages, where the Centered kernel alignment (CKA) [18] is applied to calculate the similarity of all pairs of layer representations. Figure 3 shows the heatmap between all layers across the model structures pre-trained with self-supervised learning constrained on only final stage output and our MCVT pre-trained model architecture. We can observe that the CKA heatmap between all layers across our MCVT pre-trained model is similar to the full-supervised model. This further demonstrates the effectiveness of our multi-level contrastive learning for self-supervised vision transformers.

5. Ablation Study

In this section, we explore the effect of each stage, batch size, and clustering on the final performance of our approach. Unless specified, all experiments for ablation studies are conducted on the ImageNet-100 dataset with the Swin-T architecture. We evaluate linear classification with our MCVT pre-trained Swin-T framework on the ImageNet-100 benchmark.

Table 5. Comparison of performance of top-1, top-5 accuracy by ablating each stage on ImageNet-100. ★, ✓, and ✗ denote the cluster-based normalized cross-entropy, info-based normalized cross-entropy, and no loss.

stage 4	stage 3	stage 2	stage 1	Top-1 (%)	Top-5 (%)
✓	✗	✗	✗	87.53	97.64
✓	✓	✗	✗	88.16	98.05
✓	✓	✓	✗	88.63	98.26
✓	✓	✓	✓	89.45	98.78
★	✓	✓	✓	**91.36**	**99.12**
★	✗	✗	✗	89.17	98.58

Effect of each stage. We analyze the effect of each stage on the final performance of our MCVT framework in Table 5. Specifically, we apply the info-based normalized cross-entropy loss from stage 4 to stage 1. As can be seen, both the top-1 and top-5 accuracy of our MCVT framework increases with the number of stages using the info-based normalized cross-entropy loss, which demonstrates the importance of early-stage features as self-supervision for pre-training vision transformers. Adding the cluster-based normalized cross-entropy loss to the final stage boosts the performance. This also shows the effectiveness of combining low-level info-based invariance and high-level cluster-wise invariance in our MCVT framework. In the meanwhile, removing the info-based cross-entropy loss from the early stage deteriorates the performance of our approach, which verifies the importance of early-stage supervision for self-supervised vision transformers.

Effect of batch size. Table 6 explores the effect of the batch size on the performance of linear classification with our MCVT-mix framework. Specifically, we vary the batch size from 32, 64, 128, 256, 512, and 1024. With the increase of the batch size to 512, our MCVT-mix framework achieves upward performance consistently in terms of top-1 and top-5 accuracy. However, when set the batch size to 1024, we did not observe the rising tendency. Therefore, we set 512 in all our experiments for the best performance.

Effect of clustering. To explore the effect of clustering on the final performance of our MCVT-mix framework, we vary K, the numbers of used prototypes, from (1250,2500,5000), (2500,5000,10000), (5000,10000,20000), and (10000,20000,40000) given negative prototypes of a fixed size, 1024. The experimental re-

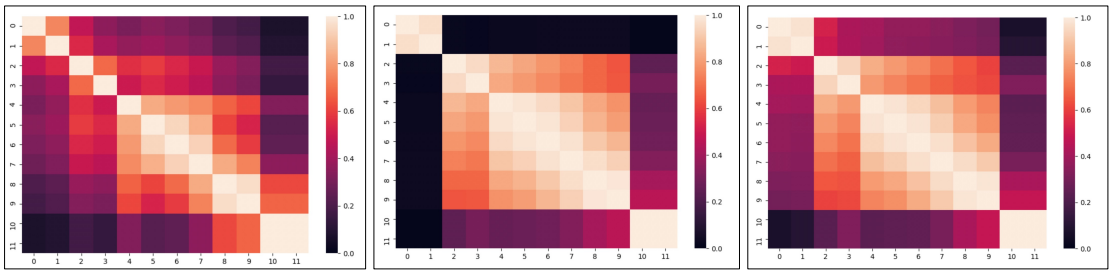

Figure 3. The CKA heatmap between all layers across the model structures pre-trained with self-supervised learning constrained on only final stage output and our MCVT pre-trained model architecture.

Table 6. Comparison of performance of top-1, top-5 accuracy by ablating batch size on ImageNet-100.

batch size	Top-1 (%)	Top-5 (%)
32	89.49	98.71
64	89.76	98.83
128	89.96	98.93
256	91.12	99.03
512	**91.36**	**99.12**
1024	91.32	**99.12**

Table 7. Comparison of performance of top-1, top-5 accuracy by ablating the K and r on ImageNet-100.

K	r	Top-1 (%)	Top-5 (%)
$10000, 20000, 40000$	1024	88.96	98.42
$5000, 10000, 20000$	1024	91.22	99.08
$2500, 5000, 10000$	1024	**91.36**	**99.12**
$1250, 2500, 5000$	1024	89.88	98.91
$2500, 5000, 10000$	2048	91.25	99.09
$2500, 5000, 10000$	512	91.17	99.06
$2500, 5000, 10000$	256	90.25	98.97

sults are reported in Table 7. When the numbers of used prototypes are set to (2500,5000,10000), our MCVT-mix achieves the best performance in terms of top-1 and top-5 accuracy. This demonstrates the importance of clustering in our MCVT-mix approach to learn more meaningful representations. Furthermore, we vary the number of negative prototypes from 256, 512, 1024, and 2048 given prototypes of (2500,5000,10000) to explore the effect of using cluster-wise invariance. As can be seen in Table 7, the performance of our MCVT-mix framework drops with the decrease of negative prototypes, which shows the effectiveness of learning the cluster-wise invariance from features of the final stage. However, introducing more negative prototypes deteriorates the performance of our MCVT-mix framework. This is because some false negative clusters are introduced during pre-training to damage the cluster-based normalized cross-entropy loss.

6. Conclusion

Summary In this work, we propose MCVT, a simple yet effective self-supervised framework with multi-level contrastive learning for vision transformers. Specifically, the low-level info-based contrastive loss is leveraged to capture the fine-grained invariance between local views, and the high-level cluster-based contrastive loss is applied to discriminate the coarse-grained invariance between global views. Furthermore, we comprehensively analyze three various multi-level contrastive learning frameworks to show the superiority of our MCVT for self-supervised transformers. Extensive experiments and ablation studies also demonstrate the state-of-the-art advantage of our method against baselines.

Limitation First, there are a lot of hyper-parameters that need to be tuned to achieve the best performance. Particularly, the best hyper-parameters employed in the prototypical contrastive head may change significantly with different datasets and downstream tasks, which is also discussed in the original prototypical contrastive learning [19] paper. We consider modifying this loss term to make it more suitable for the vision transformers in future work. Second, we are aware of the phenomenon discussed in [25]: when datasets much larger than the ImageNet-1K are employed for self-supervised learning, the representations in lower layers attend to be both locally and globally. Due to the limitation of computational resources, we do not conduct experiments on larger datasets. Therefore, we are unsure about the effectiveness of our approach *w.r.t.* larger datasets.

Broad Impact This work provides a promising direction for applying multi-level contrastive learning on self-supervised vision transformers with info-based and cluster-based contrastive losses. Furthermore, introducing more supervision signals in the early-stage pre-training process of self-supervised vision transformers indeed boosts the performance of downstream tasks, such as image classification and semantic segmentation.

Acknowledgement

This work was partially supported by JSPS KAKENHI (Grant No. 20H04249, 20H04208) and the National Natural Science Foundation of China (Grant No. 62006045).

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: BERT pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6154–6162, 2018.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Proceedings of European Conference on Computer Vision (ECCV)*, pages 213–229, 2020.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of International Conference on Machine Learning (ICML)*, 2020.

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[11] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2988–2997, 2021.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia. Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of International Conference on Learning Representations*, 2021.

[14] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Xcit: Cross-covariance image transformers. *arXiv preprint arXiv:2106.09681*, 2021.

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

[17] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[18] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[19] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[20] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, and Jinqiao Wang. MST: masked self-supervised transformer for visual representation. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[22] Shentong Mo, Zhun Sun, and Chao Li. Siamese prototypical contrastive learning. In *BMVC*, 2021.

[23] Shentong Mo, Zhun Sun, and Chao Li. Rethinking prototypical contrastive learning through alignment, uniformity and correlation. In *BMVC*, 2022.

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

[25] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *arXiv preprint arXiv:2108.08810*, 2021.

[26] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[27] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[28] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[30] Xudong Wang, Ziwei Liu, and Stella X Yu. CLD: unsupervised feature learning by cross-level instance-group discrimination. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[31] 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 (CVPR)*, 2018.

[32] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 432–448, 2018.

[33] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021.

[34] Tsung yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[35] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[36] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. *arXiv preprint arXiv:2110.09408*, 2021.

[37] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of International Conference on Machine Learning (ICML)*, 2021.

[38] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, 2017.

[39] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision (IJCV)*, 127:302–321, 2018.