Pretraining Vision Transformer via Multi-Task Self-Supervision

Haobo Wang
Zhejiang University
wanghaobo@zju.edu.cn

Abstract

This technical report simply shares an idea of a multi-task self-supervised vision transformer, without empirical verification. It would be glad if this model can be implemented and verified by someone.

1 Introduction

Recently, self-supervised learning [10] has demonstrated its ability to get good representation ability. Amongst them, discriminative ones, especially contrastive learning [2, 8, 9], are usually considered to be more effective than generative ones [11] and other handcrafted ones [3, 16, 12]. One of the reasons is that current non-contrastive SSL approaches typically adopt only one self-supervising signal, such as colorizing gray-scale images. Therefore, they fail to capture much richer visual semantics as traditional full-supervising signals do. Moreover, Doersch and Zisserman [4] empirically found that different downstream tasks prefer specific SSL representations. To cope with this issue, they proposed multi-task self-supervised learning (MTSSL), which simultaneously trains the ResNet by multiple SSL tasks like relative position [3], colorization [16], exemplar [6] and motion segmentation [13]. Later on, other works [15, 1] followed this idea. For example, SiT [1] pretrains vision transformer model by multiple SSL tasks with the help of adding pretext tokens.

Despite the intuitiveness of MTSSL, one tricky problem is how to handle the variety of low-level image statistics across tasks. For example, colorization [16] requires to convert the image to gray-scale one, inpainting [14] masks part of an image, and jigsaw-puzzle [12] randomly permutes the image patches. Different pretext tasks usually destroy the global image semantics and thus the transformed image can usually used to fit one specific SSL signal. One potential solution is to generate $k * N$ training examples given k SSL tasks, which involves cross-domain training examples and training can be unstable. There are two strategies to alleviate this problem. The first one, adopted by SiT [1], simply choose some compatible pretext tasks, e.g. reconstruction, contrastive learning, and rotation prediction, whose transformation operations do not conflict with each other. The second one is to *harmonize* the inputs. For instance, in [4], images are converted to Lab, and the a and b channels are discarded. Still, there is no elegant way to simultaneously train different pretext tasks.

To bridge this gap, this technical report introduces an interesting strategy by means of a vision transformer model [5]. Due to some reason, e.g. lacking GPUs, it is impossible for us to empirically validate the effectiveness of this model. However, we believe it is an interesting idea that integrates both discriminative and generative SSL tasks to get better representations.

2 Method

Why multiple SSL tasks cannot be jointly trained? The reason is that many pretext tasks take away part of input as supervising signal and destroy some global image semantics, such as color. Hence, it is intuitive to

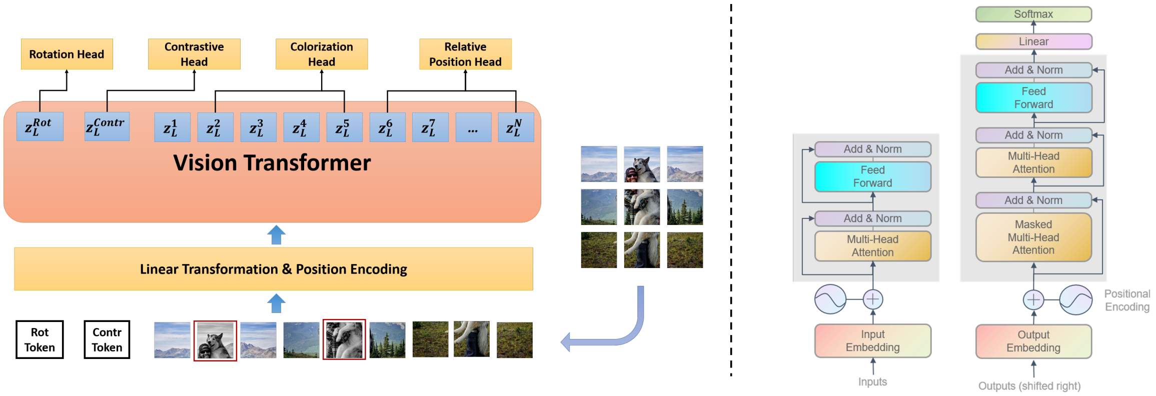

Figure 1: Model architecture. Contrastive head generates features for calculating InfoNCE loss. Rotation head predicts image rotation (0%). Colorization head predicts colors per pixel (or mean color) for patch 2 and 5. The relative position head predicts the relative position of patch 6 and 9.

limit the transformation operation in a patch-level, while preserving the image-level semantics as much as possible. Recent popular vision transformer model (ViT) [5] paves the way to achieve this goal. Similar to Transformer models in NLP, ViT reshapes the image to 16 × 16 patches and then feeds them (with linear transformation) into a standard Transformer model. In ViT, we are now able to get both image-level and patch-level features. Hence, those pretext tasks that require very strong transformation can be performed on a patch level, with global image semantics being preserved.

To achieve this goal, we first categorize some existing SSL tasks into three types.

- **Image-Level** The first one is the image-level ones, which do not destroy the image-level semantics. For instance, rotation is predicted at image level; contrastive signals usually compare augmented/negative images; clustering aggregates many similar images. The transformation operation used typically does not affect other SSL tasks.

- **Single-Patch** Some pretext tasks, however, take away too much from the whole image, e.g. colorization, inpainting. In ViT, these tasks can be incorporated at patch level.

- **Multiple-Patch** Finally, many SSLs, such as jigsaw-puzzle and relative position, predict spatial context between patches. Obviously, they can be achieved by using patch-level representations in ViT.

Now, we are ready to simultaneously combine these SSL tasks via ViT model.

Formally, give an image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$, we reshape it to 2D patches $\boldsymbol{x}_p \in \mathbb{R}^{N \times P^2 \cdot C}$, where $N = HW/P^2$ is the number of patches. Then, the conventional ViT model transforms the inputs to $[\boldsymbol{z}_L^{\text{cls}}, \boldsymbol{z}_L^1, \dots, \boldsymbol{z}_L^N]$, where $\boldsymbol{z}_L^{\text{cls}}$ is the latent representation of [CLS] token and $\boldsymbol{z}_L^i$ $(i = 1, \dots, N)$ is the latent representation i-th patch. In the sequel, we choose three typical tasks, i.e. contrastive learning, colorization and relative-position to elaborate our method.

- For contrastive learning, we follow SiT [1] and use a special token [CT] before the patches to get contrastive representations $\boldsymbol{z}_L^{\text{Contr}}$. Then, we can feed it to a simple MLP model $f_{\text{Contr}}(\cdot)$ and use InfoNCE loss to train this contrastive head network. By using more tokens, we can incorporate more image-level SSL tasks.

- For colorization ones, we first randomly choose k patches and convert them to gray-scale. Assume that $[i_1, i_2, ..., i_k]$ is the patch-index being chosen, we can predict their colors using $f_{\text{Color}}(\boldsymbol{z}_L^{i_j})$ $(j = 1, \ldots, k)$, where $f_{\text{Color}}(\cdot)$ is a shared MLP function. In this way, we can involve more singe-patch tasks, such as reconstructing part of the patch or predicting the whole patch.
- For relative-position tasks, we can randomly choose two patches and then feed their representations $\boldsymbol{z}_L^i$ and $\boldsymbol{z}_L^j$ into a prediction network $f_{\text{RP}}(\cdot, \cdot)$ to solve a simple eight-way classification problem. To avoid trivial solutions, we can slightly augment the used patches such as clipping and color jitting, as suggested in [3]. One can also randomly permute the patches to solve a jigsaw puzzle problem. It seems ViT model can cheat via the position-embeddings. But, we suppose it can benefit those learnable position embedding to strengthen their position-aware ability. Moreover, it may be helpful in alleviating over-smoothing issues in ViT [7], since differences between patches are emphasized.

In summary, the proposed method addresses the input harmonize problem in MTSSL by supervising ViT at both image and patch levels. Since those pretext tasks which destroy the global image semantics are now trained at patch level, we can freely incorporate multiple SSL tasks (both discriminative and generative). Hence, much richer low-level vision semantics can be discovered.

References

[1] Sara Atito Ali Ahmed, Muhammad Awais, and Josef Kittler. Sit: Self-supervised vision transformer. *CoRR*, abs/2104.03602, 2021.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[3] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1422–1430. IEEE Computer Society, 2015.

[4] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2070–2079. IEEE Computer Society, 2017.

[5] 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. *CoRR*, abs/2010.11929, 2020.

[6] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 766–774, 2014.

[7] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Improve vision transformers training by suppressing over-smoothing, 2021.

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE, 2020.

[10] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *CoRR*, abs/1902.06162, 2019.

[11] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[12] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, volume 9910 of *Lecture Notes in Computer Science*, pages 69–84. Springer, 2016.

[13] Deepak Pathak, Ross B. Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6024–6033. IEEE Computer Society, 2017.

[14] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2536–2544. IEEE Computer Society, 2016.

[15] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 762–771. IEEE Computer Society, 2018.

[16] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, pages 649–666. Springer, 2016.