Proc. of the 9th Intl. Conf. of Asian Society for Precision Engg. and Nanotechnology (ASPEN 2022)
15–18 November 2022, Singapore. Edited by Nai Mui Ling Sharon and A. Senthil Kumar
doi://10.3850/978-981-18-6021-8_OR-08-0105

CoTMAE:Hybrid Convolution-Transformer Pyramid Network Meets Masked Autoencoder

Chuanxiang Li[1,2], Shuming Yang[1,2,#], Pengyu Hu[1,2], Huiwen Deng[1,2], Yu Duan[1,2] and Xing Qu[1,2]

1 State Key Laboratory for Manufacturing Systems Engineering, Xi an Jiaotong University, Xi an 710049, China.
2 School of Mechanical Engineering, Xi an Jiaotong University, Xi an 710049, China.
# Corresponding Author / Email: shuming.yang@mail.xjtu.edu.cn

KEYWORDS: Object detection, Self-supervised, Transfer learning, Instance segmentation

Vision Transformer (ViT) has become the most popular architecture for existing vision tasks, but it is difficult to apply to the industrial domain due to its heavy computational cost of its self-attention mechanism. Masked AutoEncoder (MAE) has recently led the trend of self-supervised learning with a simple, scalable, and efficient ViT-based asymmetric encoder-decoder architecture. To mitigate the quadratic complexity of self-attention, we design a hybrid convolution-transformer pyramid network that effectively combines the respective advantages of convolution and self-attention. However, it is still unclear how our convolution-transformer pyramid network can be adopted in MAE pre-training, as it uses the local convolution operation, making it difficult to handle random sequences with only partial visual tokens. In this paper, we present a novel and efficient masked image modeling (MIM) approach, convolutional-contextual transformer masked autoencoder (CoTMAE). The pipeline of CoTMAE consists of: (i) a window masking (WM) strategy that ensures computational efficiency, (ii) an encoder that only takes visible patches as input to our hybrid convolution-transformer network, (iii) a multi-scale fusion module that enhances the output features of the encoder, which allows the decoder to focus on the reconstruction task. (iv) a feature alignment module that handles the distribution of encoded features and masked patches, and (v) a decoder that reconstructs the missing pixels of the masked patches. Specifically, WM directly divides the original image into equal-sized windows, using a random mask strategy within each window. Afterwards, only visible patches are reordered and reorganized into images as input to the hybrid convolution-transformer pyramid network. Our WM significantly improves the training efficiency of hybrid convolution-transformer networks and reduces GPU memory, while maintaining a competitive advantage with supervised training models in downstream tasks. We demonstrate that CoTMAE successfully enables self-supervised pre-training of a hybrid convolution-transformer pyramid network and achieves good fine-tuning performance on instance segmentation datasets. The encoder of CoTMAE is trained on ImageNet-1K dataset classification and fine-tuned on COCO 2017 dataset to achieve 52.9% APbox and 45.8% APmask. On industrial instance segmentation datasets, CoTMAE shows better fine-tuning performance than supervised models.

1. Introduction

Inspired by the great success of Masked Language Modeling (MLM)[1-3] in natural language processing (NLP) and the rapid development of Vision Transformer (ViT)[4] in computer vision (CV), Masked Image Modeling (MIM) has achieved superior results in computer vision. Mask Autoencoders (MAE)[5] is a representative self-supervised approach in MIM, and has gradually become a paradigm of self-supervised pre-training leading the computer field. By using a random masking strategy on the original image, MAE[5] takes only visible image patches as input images and makes predictions for the masked image patches. It expects the encoder network to learn features containing rich semantic information by recovering the pixels of the masked image patches.

In essence, the asymmetric encoder-decoder structure is the optimal design for MAE, in which the encoder operates only on visible patches, and the lightweight decoder aims to recover all patches. On the one hand, this method not only improves the training speed of pre-training and reduces the memory footprint of GPU, but also achieves excellent performance on downstream tasks. On the other hand, ViT[4], as an encoder network for MAE[5], has major obstacles in industrial detection applications due to its heavy computational cost and its huge amount of parameters. The self-attention module possessed by ViT[4] can learn the long-term dependencies of features, which enables ViT[4] to have stronger global context modeling ability than convolutional neural networks (CNNs)[6-11]. In fact, local inductive bias and hierarchical architecture are crucial for boosting the performance of ViT[4]. Many recent works have explored the combination of Convolutional Neural Networks and Transformers. Hybrid convolution-transformer networks[12-16] have demonstrated incredible performance on vision tasks, e.g., image classification, object detection, instance segmentation. However, it still cannot achieve industrial application. Inspired by CoAtNet[17], we propose a hybrid convolution-transformer pyramid backbone network that exploits self-attention to maximize the performance of CNNs. Our backbone network not only achieves good performance on public datasets, but also achieves good performance and computational efficiency on industrial instance segmentation data.

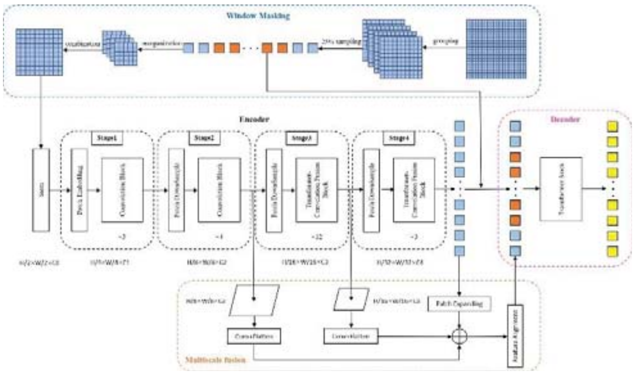
Compared with many self-supervised methods, the masked auto-encoding strategy in MAE[5] has a remarkable effect. Yet, it has limitations due to only supporting the isotropic VIT structure. At present, many methods utilize masked auto-encoding strategies by zero-padded masked patches to restore the entire image. While this works, it only acquires a suboptimal model and also sacrifices training efficiency. We consider whether the mask auto-encoding strategy can be applied to the hybrid convolution-transformer pyramid backbone network for self-supervised learning tasks, so as to further improve the detection performance of industrial data and reduce the time cost.

Our work focuses on extending the masked auto-encoding strategy and the asymmetric encoder-decoder architecture to convolutional transformer pyramid networks. To this end, we present a simple and effective convolutional-contextual transformer masked autoencoder (CoTMAE) approach, illustrated in Fig. 1. Our approach is simple and mainly consists of five components, which are a window masking strategy, an encoder, a decoder, a feature alignment module, and a multi-scale fusion module. The encoder, a hybrid convolution-transformer pyramid network, effectively combines convolution and self-attention, which downscale the input image by 1/4, 1/8, 1/16, and 1/32 in four stages, respectively. The first two stages fully exploit convolution to encode local feature, and the convolution and self-attention fusion modules are used in the latter two stages. The encoder also exploits overlapping windows at the beginning to improve performance and reduce input resolution, which also determines that our framework is not suitable for the masked auto-encoding strategy in MAE[5]. The window masking strategy with sequential reconstruction and alignment operations divides the image into different windows, each of which adopts a random mask strategy. Finally, we extract the visible patch and reassemble it into an image as the input to the encoder. The decoder is a lightweight architecture following the design in MAE[5], which inputs all patches to predict the masked patches. The multi-scale fusion module allows the decoder to focus more on the reconstruction task and improve the quality of representation by fusing the features of different scales of the encoder. Furthermore, we align the multi-scale fused features with the predicted mask patch representations via the feature alignment module.

In summary, we make the following main contributions: (1) We design a simple, effective, and general self-supervised framework CoTMAE, which enables any pyramid network to act as its encoder. (2) We present a window masking strategy that is naturally integrated into our Hybrid Convolutional Transformer Pyramid Network architecture to help achieve masked auto-encoding. (3) Compared with supervised pre-training and other pre-trainings that utilize strategies such as zero-filling of original images, our framework achieves better results in industrial instance segmentation tasks, which not only reduces computational and training costs, but also ensures image segmentation performance.

Fig. 1 Our CoTMAE architecture. Our approach makes visible patches as encoder input through a window masking strategy and enables the decoder to focus on the reconstruction of masked patches through a multi-scale fusion module and a feature alignment module.

2. Related Work



Vision Transformer. Transformers[1-3] have significantly advanced the development of natural language processing (NLP) and computer vision (CV). Vision Transformer (ViT)[4] demonstrates the power of Transformer in the field of computer vision, with performance that outperforms Convolutional Neural Networks (CNNs). ViT[4] splits an image into equal-sized blocks as input to a pure transformer architecture and achieves excellent performance on classification tasks. Yet, it performs poorly on intensive prediction tasks, and the heavy computational cost also becomes a huge obstacle for industrial applications. To this end, a series of works present a ViT-based hierarchical architecture[13,18-22] to reduce the computational complexity and further unleash the potential of ViT as a general model. PVT[14] reduces the complexity of the global self-attention mechanism through non-overlapping spatial reduction windows (SRW). Swin Transformer[13] restrains self-attention operator within non-overlapping, shifted local windows. DaViT[23] presents a dual attention mechanism to achieve global modeling by stacking attention mechanisms in spatial and channel dimensions. Hybrid Convolution-Transformer Network exploits the strong inductive bias in traditional CNNs to address the redundancy and slow convergence of self-attention in shallow features. State-of-the-art performance has been achieved in tasks such as image classification, object detection, semantic segmentation, and video understanding. CoAtNet[17] utilizes a simple and effective method to stack convolutional blocks and self-attention blocks vertically and analyze their respective characteristics in detail. Uniformer[18] seamlessly integrates convolution and self-attention, which solves the redundancy and dependency problems of efficient expression learning. Inspired by hybrid architectures in the visual backbone, the encoder in our CotMAE can better incorporate features learned from convolution and self-attention while balancing efficiency and performance.

Contrastive learning. Contrastive learning[24-33] has been very popular in self-supervised representation learning for vision tasks. The basic idea of contrastive learning is to maximize the consistency between different transformed views of the same image, such as random cropping, random flipping, color transformation, etc., and minimize the consistency between transformed views of different images. In this way, the encoder learns an image-level representation of an image rather than a pixel-level generation.

Masked image modeling. Inspired by BERT[1] for masked language modeling, BEIT[34] solves masked image modeling tasks by predicting pixels or discrete tokens based on the VIT architecture. But there is no explicit encoder-decoder structure. In recent works, MAE[5] presents an asymmetric encoder-decoder structure in which the encoder operates only on visible blocks and predicts mask patches through a lightweight decoder. MaskFeat[35] and PeCo[36] improve the quality of representations for self-supervised learning by studying prediction targets. CAE[37] separates the encoder representation from the prediction task and makes predictions in the latent representation space from visible patches to mask patches. UM-MAE[38] successfully uses quadratic masking strategy to achieve self-supervision in pyramid networks like Swin Transformer[13], PVT[14], etc. ConvMAE[39] presents a simple self-supervised learning framework with a block-wise masking strategy, which demonstrates that multi-scale features from supervised encoders can improve the performance of downstream tasks. The very recent approach Green-MAE[40] is similar to our approach, allowing the hierarchical models to discard masked patches and operate only on the visible ones. Our CoTMAE benefits from the development of hybrid convolutional-transformer pyramid networks and useful experience gained from recent works[34-42].

3. Approach

Our convolutional-contextual transformer masked autoencoder (CoTMAE) pretrains hybrid convolutional-transformer pyramid networks by solving masked image modeling tasks. The architecture, illustrated in Fig. 1, consists of five components: a window masking strategy, a hybrid convolution-transformer backbone network, a multi-scale fusion module, a feature alignment module, and a transformer decoder. We introduce them respectively in the following subsections.

3.1 Window Masking

We present a simple and effective window masking (WM) strategy to support the hybrid convolutional transformer pyramid backbone network. Masked Autoencoders[5] adopt a random masking strategy on the input tokens and only provide visible patches to the encoder. However, the same strategy cannot be directly devoted to our hybrid convolution-transformer pyramid backbone network. Adopting a random mask strategy directly on the original image and zero-padding the masked patches to maintain the original image size is a common approach for pre-training on convolution-transformer pyramid networks, but this seriously reduces training efficiency and affects the performance of downstream tasks. Instead, our Window Masking strategy supports the encoder in CoTMAE to operate only on visible patches.

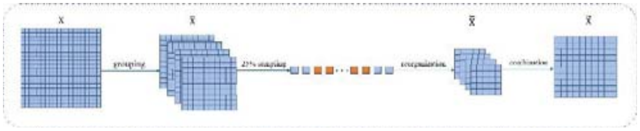


Fig. 2 Illustration of a window masking strategy. In our approach, the input X is divided into equal-sized partial windows, resulting in $\widehat{X}$. A random mask is taken within each window, most of which are invisible. We reorganize the visible patches in each window to form $\overline{X}$. Finally, we recover a complete graph $\widetilde{X}$ from each group.

As shown in Fig. 2, patches of equal size are first directly extracted on the original image. We verify that the extraction process without convolution operation has better performance on downstream tasks. After that, we divide windows of equal size and randomly mask with a fixed 75% mask ratio in each window. Then, the visible blocks in each window are extracted, which are reordered and reassembled into the input image of the encoder. The actual image resolution is half of the original image, which not only improves the training efficiency, i.e. only the visible blocks are trained, but also simply and effectively uses the mask auto-encoding strategy. We also compare random masking strategies without splitting windows and verify that our WM enhances the localization of image location information, allowing convolution to learn finer boundary information.

3.2 Hybrid Convolution-Transformer Pyramid Backbone Network

In the research of visual transformers, a pyramid transformer architecture with a hierarchical structure has been shown to enhance the performance of ViT[4]. In addition, powerful hybrid convolution-transformer structures have also emerged, such as CoAtNet[17], Uniformer[18], etc., which demonstrate the great potential of hybrid convolution-transformer pyramid structures, while achieving good performance on various downstream tasks.

As shown in Fig. 3, our encoder consists of four stages with output spatial resolutions of H/4×W/4, H/8×W/8, H/16×W/16, and H/32× W/32, where H and W are the height and width of the input image. In the first two stages, the resolution is reduced by half using consecutive convolutional layers, where overlapping convolution operations are used to reduce the resolution while improving performance. The output of our second stage is transformed into token Embeddings as the input of the third stage. We propose a Transformer-Convolution Fusion (TCF) module that combines convolutional and self-attention layers simply and efficiently. We also add DW-Conv as an implicit positional encoding in the multi-head self-attention (MHSA) module[43], which can help the transition between attention and convolutional blocks.

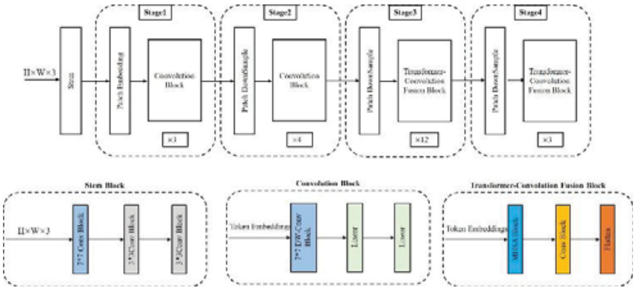


Fig. 3 Illustration of the backbone network of CoTMAE. The patch downsample module uses 3*3 overlapping convolutions, which can improve performance. The convolution block uses a 7*7 convolution kernel to increase the receptive field. It also shows the structure of the transformer-convolution fusion module, which effectively utilizes the features of convolution and self-attention with a simple vertical stacking approach.

3.3 Decoder and Loss Function

The decoder in CoTMAE adopts a structure similar to the decoder in MAE[5], a lightweight transformer layer. The input of the decoder is also added with a positional embedding, which contains the encoder output and the masked patches that need to be predicted. A feature alignment module is added before the decoder, which aligns the features learned by the encoder with the mask patch in space, otherwise, the reconstruction effect will be garbled. Unlike MAE[5], the multi-scale fusion module fuses features from the last three stages, which can enable the decoder to focus on the prediction of masked patches. Due to the need to fuse features of different scales, our multi-scale fusion module consists of a downsampling module, a linear layer, and an upsampling module. The downsampling module uses 2*2 non-overlapping convolutions. The patch-expand module as our upsampling module can achieve better performance than other upsampling modules. The linear layer enables the fused features to maintain the same dimension as the decoder.

$$Pd = Linear\big(Conv(F2,2) + Conv(F3,1) + PatchExpand(F4,2)\big)$$

Among them, Conv(*,k) represents the convolution operation with the size of k in both kernel and stride, and PatchExpand(∗,s) represents the upsampling operation with the magnification of s.

We use mean squared error (MSE) as our loss function. We compute the loss only on masked patches, similar to BERT[1]. Following MAE[5], we also normalize the pixels of the original image as our prediction target, which improves the representation quality.

$$\zeta = 1/m \sum\_(\ i \in m)▒(I(i) - \hat{I}(i))\verb|^|2$$

where m is the set of mask tokens and i is the token index. I(i) is the normalized pixel value of the input image and Î(i) is the decoder output.

4. Experiment

Our experiments focus on fine-tuning accuracy on downstream tasks rather than linear probing. The performance of our self-supervised model is directly compared to supervised models trained on ImageNet-1K[44], COCO datasets[45] using the encoder in CoTMAE. We demonstrate that our approach is more efficient and less time-intensive by validating the performance of both models on an industrial instance segmentation dataset. Furthermore, all ablation experiments are performed on our industrial instance segmentation dataset.

4.1 Setting of Backbone Network

We preliminarily design the number of layers of each stage of the encoder in CoTMAE according to the configuration of ResNet[46]. According to previous works[13,17,47], if an image of size H×W is input, a feature embedding with a resolution of H/2×W/2 and 64 channels can be obtained after the first stem layer. After the four downsampling modules, along with the resolution reduction to H/4×W/4, H/8×W/8, H/16×W/16, and H/32×W/32, the number of channels increased to 96, 192, 384, and 512, respectively. In Stem, we use a 7*7 convolution and two consecutive 3*3 convolutions to extract prior knowledge. The kernel, stride, and padding values for each patch embedding layer are 3, 2, and 1, respectively. The backbone network in self-supervised pre-training is only employed as a module in CoTMAE, which can be replaced by any other CNNs or isotropic backbone network without any modification.

4.2 Downstream Tasks

Pre-training Setup. Self-supervised pre-training is conducted on the industrial instance segmentation dataset with our CoTMAE. We specified the input image to 256*256, which uses simple random augmentation, including random cropping, horizontal flip, and normalization. We trained for a total of 1600 epochs. The number of warm-up epochs is 40. We use the AdamW[48] optimizer with the cosine annealing schedule, which uses a base learning rate of 1.5×10-4, a weight decay of 0.05, and a batch size of 1024.

Instance Segmentation Setup. Our industrial annotation dataset has a total of 30,000 images, of which the test set has 1,500 images. We use the Encoder from CoTMAE as the backbone network of Cascade RCNN[49]. Our architecture uses FPN as detector. We finetune Cascade RCNN on the industrial instance segmentation dataset and report APmask and ARmask on the test dataset. We also use the AdamW[48] optimizer with a learning rate of 1e-4 and a weight decay of 0.05. We train for 36 epochs with a fixed-step weight decay strategy.

Table 1 Instance Segmentation on Industrial Annotation Datasets. All the methods use the hybrid convolution-transformer pyramid network architecture of CoTMAE. All results are based on the same implementation of instance segmentation. COCO-AP^{mask} represents the masked average precision of the supervised model on the COCO 2017 dataset[45]. Segm-* represents the segmentation performance metrics on the industrial instance segmentation dataset

Methods	Segm-AP@0.5	Segm-AR@0.5	COCO-AP^{mask}
No-pretrained model	89.7%	92.0%	-
Supervised model	92.5%	94.0%	45.8%
Ours	92.5%	94.5%	-

Results on Instance Segmentation Dataset. As shown in Table 1, we adopt AP and AR as our evaluation metrics on instance segmentation data. We separately experiment with the performance of the no pretraining model, supervised pretraining model, and self-supervised pretraining model on industrial annotated data. Compared with no pretraining model finetuned for 36 epochs on industrial annotated data, CoTMAE can significantly improve AP and AR by 2.8% and 2.5% with 25 finetuning epochs. Our self-supervised pre-trained model outperforms supervised pre-trained model by 0%/0.5% in terms of AP/AR. Although the performance gain is small, our model greatly improves the pre-training efficiency. It can be contended that our approach is appropriate for hybrid convolution-transformer pyramid networks and achieves better performance than supervised models in downstream tasks.

4.3 Ablation Studies

We conduct some essential ablation experiments on CoTMAE to analyze the effects of different components. We report and analyze the results of the ablation experiments in detail

Comparison results of different masking strategies. Table 2 shows the results of instance segmentation with different strategies. It can be seen that our pretrained model achieves the best segmentation performance. Furthermore, Random Mask Reorganization outperforms Zero-Padding by 0.2%/0.1% in terms of AP and AR. In addition, Zero-Padding operates on all patches, requiring longer training time and larger memory. It can also be seen that the performance of Window Masking （WM） and Random Mask Reorganization is almost the same when only WM is used and that the performance will increase significantly when multi-scale fusion is added. As we analyzed, the fused encoder features allow the decoder to focus on the prediction of mask patch pixels, allowing the encoder to learn more efficient features. This verifies that our multi-scale fusion module is effective for improving downstream instance segmentation performance. In Fig. 4, we also compare the fine-tuning performance variation curves of different masking strategies. Window Masking has higher fine-tuning accuracy throughout training

Table 2 Comparison among different strategies using CoTMAE under 1600 epoch pretraining. Zero-Padding represents filling all mask patches of the original image with zero pixel values. Random Mask Reorganization means reorganizing all unmasked patches into the whole image. Multiscale Fusion indicates that our multiscale fusion module is used.

Mask Strategy (75%)	Image Size	MSE-Loss	epochs	Segm-AP@0.5	Segm-AR@0.5
Zero-Padding	256	0.2014	1600	91.5%	93.5%
Random-Mask-Reorganization	256	0.1989	1600	91.7%	93.6%
Window Masking	256	0.1970	1600	91.8%	93.8%
Window Masking + Multiscale Fusion	256	0.1954	1600	92.5%	94.5%

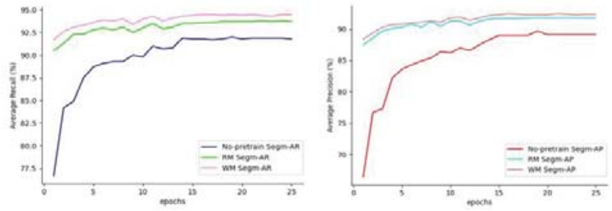


Fig. 4 Fine-tuning results for different masking strategies. Under the settings in Table 2, we can see that our strategy consistently outperforms random mask reorganization.

Pre-training epochs. Based on MAE research, larger pre-training epochs can improve self-supervised fine-tuning performance without overfitting. We test the performance on downstream tasks based on pretrained models trained for 400, 800 and 1600 epochs. The performance results are shown in Table 3.

Table 3 Performance of instance segmentation for different pre-training epochs

Epochs	Segm-AP@0.5	Segm-AR@0.5
400	91.6%	93.7%
800	92.0%	93.9%
1600	92.5%	94.5%

Window size. Previous work has shown that larger window partitions are practical for fine-tuning performance. In the experiments, we divided the reconstituted windows into 4 and 16 windows. To be clear, larger window size add little extra computational cost. Table 4 shows the fine-tuning results for different window sizes, which demonstrate that larger window sizes have better performance. We analyze that smaller window sizes may reduce the difficulty of pre-training.

Table 4 The influence of different window numbers (Wn) on the performance of downstream tasks

Wn	Segm-AP@0.5	Segm-AR@0.5
4	92.5%	94.5%
16	92.3%	94.2%

5. Conclusions

In this work, we present a simple and effective self-supervised pretraining framework named CoTMAE, which uses a novel window masking strategy to allow our hybrid convolution-transformer pyramid network to operate only on visible patches. Compared with existing alternative strategies, it achieves stronger training efficiency and better performance on industrial instance segmentation datasets. We believe that our design scheme is also practical for CNNs to support self-supervised tasks with large amounts of unlabeled data in different industrial scenarios.

REFERENCES

[1] Devlin J, Chang M-W, Lee K, et al. BERT: Pre-training of Deep

Bidirectional Transformers for Language Understanding, 2018: arXiv:1810.04805.

[2] Radford A, Wu J, Child R, et al. Language Models are Unsupervised Multitask Learners[C], 2022.

[3] Brown T B, Mann B, Ryder N, et al. Language Models are Few-Shot Learners[J]. neural information processing systems, 2020.

[4] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2020: arXiv:2010.11929.

[5] He K, Chen X, Xie S, et al. Masked Autoencoders Are Scalable Vision Learners, 2021: arXiv:2111.06377.

[6] Cui J, Zhong Z, Liu S, et al. Parametric Contrastive Learning, 2021: arXiv:2107.12028.

[7] Zhang X, Wang Q, Zhang J, et al. Adversarial AutoAugment, 2019: arXiv:1912.11188.

[8] Li Y, Yu Q, Tan M, et al. Shape-Texture Debiased Neural Network Training, 2020: arXiv:2010.05981.

[9] Chen X, Wang H, Ni B. X-volution: On the unification of convolution and self-attention, 2021: arXiv:2106.02253.

[10] Huang G, Liu Z, Van Der Maaten L, et al. Densely Connected Convolutional Networks, 2016: arXiv:1608.06993.

[11] Li X, Wang W, Hu X, et al. Selective Kernel Networks, 2019: arXiv:1903.06586.

[12] Cao H, Wang Y, Chen J, et al. Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation, 2021: arXiv:2105.05537.

[13] Liu Z, Lin Y, Cao Y, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, 2021: arXiv:2103.14030.

[14] Wang W, Xie E, Li X, et al. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions, 2021: arXiv:2102.12122.

[15] Hassani A, Walton S, Li J, et al. Neighborhood Attention Transformer, 2022: arXiv:2204.07143.

[16] Lee Y, Kim J, Willette J, et al. MPViT: Multi-Path Vision Transformer for Dense Prediction, 2021: arXiv:2112.11010.

[17] Dai Z, Liu H, Le Q V, et al. CoAtNet: Marrying Convolution and Attention for All Data Sizes, 2021: arXiv:2106.04803.

[18] Li K, Wang Y, Gao P, et al. UniFormer: Unified Transformer for Efficient Spatiotemporal Representation Learning, 2022: arXiv:2201.04676.

[19] Chu X, Tian Z, Wang Y, et al. Twins: Revisiting the Design of Spatial Attention in Vision Transformers, 2021: arXiv:2104.13840.

[20] Gao P, Lu J, Li H, et al. Container: Context Aggregation Network[J]. arXiv: Computer Vision and Pattern Recognition, 2021.

[21] Dong X, Bao J, Chen D, et al. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows, 2021: arXiv:2107.00652.

[22] Touvron H, Cord M, El-Nouby A, et al. Augmenting Convolutional networks with attention-based aggregation, 2021: arXiv:2112.13692.

[23] Ding M, Xiao B, Codella N, et al. DaViT: Dual Attention Vision Transformers, 2022: arXiv:2204.03645.

[24] Chen T, Kornblith S, Norouzi M, et al. A Simple Framework for Contrastive Learning of Visual Representations, 2020: arXiv:2002.05709.

[25] Tian Y, Sun C, Poole B, et al. What Makes for Good Views for Contrastive Learning[J]. neural information processing systems, 2020.

[26] Chen X, Xie S, He K. An Empirical Study of Training Self-Supervised Vision Transformers, 2021: arXiv:2104.02057.

[27] Grill J-B, Strub F, Altché F, et al. Bootstrap your own latent: A new approach to self-supervised Learning, 2020: arXiv:2006.07733.

[28] Caron M, Touvron H, Misra I, et al. Emerging Properties in Self-Supervised Vision Transformers, 2021: arXiv:2104.14294.

[29] Chen X, He K. Exploring Simple Siamese Representation Learning, 2020: arXiv:2011.10566.

[30] Caron M, Misra I, Mairal J, et al. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments, 2020: arXiv:2006.09882.

[31] Wu Z, Xiong Y, Yu S, et al. Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination, 2018: arXiv:1805.01978.

[32] He K, Fan H, Wu Y, et al. Momentum Contrast for Unsupervised Visual Representation Learning, 2019: arXiv:1911.05722.

[33] Zhou J, Wei C, Wang H, et al. iBOT: Image BERT Pre-Training with Online Tokenizer, 2021: arXiv:2111.07832.

[34] Bao H, Dong L, Wei F. BEiT: BERT Pre-Training of Image Transformers, 2021: arXiv:2106.08254.

[35] Wei C, Fan H, Xie S, et al. Masked Feature Prediction for Self-Supervised Visual Pre-Training, 2021: arXiv:2112.09133.

[36] Dong X, Bao J, Zhang T, et al. PeCo: Perceptual Codebook for BERT Pre-training of Vision Transformers, 2021: arXiv:2111.12710.

[37] Chen X, Ding M, Wang X, et al. Context Autoencoder for Self-Supervised Representation Learning, 2022: arXiv:2202.03026.

[38] Li X, Wang W, Yang L, et al. Uniform Masking: Enabling MAE Pre-training for Pyramid-based Vision Transformers with Locality, 2022: arXiv:2205.10063.

[39] Gao P, Ma T, Li H, et al. ConvMAE: Masked Convolution Meets Masked Autoencoders, 2022: arXiv:2205.03892.

[40] Huang L, You S, Zheng M, et al. Green Hierarchical Vision Transformer for Masked Image Modeling, 2022: arXiv:2205.13515.

[41] Baevski A, Hsu W-N, Xu Q, et al. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language, 2022: arXiv:2202.03555.

[42] El-Nouby A, Izacard G, Touvron H, et al. Are Large-scale Datasets Necessary for Self-Supervised Pre-training?, 2021: arXiv:2112.10740.

[43] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need, 2017: arXiv:1706.03762.

[44] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database[C]. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009: 248-255.

[45] Lin T-Y, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context, 2014: arXiv:1405.0312.

[46] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition, 2015: arXiv:1512.03385.

[47] Liu Z, Mao H, Wu C-Y, et al. A ConvNet for the 2020s, 2022: arXiv:2201.03545.

[48] Loshchilov I, Hutter F. Decoupled Weight Decay Regularization, 2017: arXiv:1711.05101.

[49] Cai Z, Vasconcelos N. Cascade R-CNN: Delving into High Quality Object Detection, 2017: arXiv:1712.00726.