Integrally Pre-Trained Transformer Pyramid Networks

Yunjie Tian¹, Lingxi Xie², Zhaohui Wang¹, Longhui Wei², Xiaopeng Zhang², Jianbin Jiao¹, Yaowei Wang¹, Qi Tian², Qixiang Ye¹,³*  
¹UCAS ²Huawei Inc. ³Pengcheng Lab.
tianyunjie19@mails.ucas.ac.cn 198808xc@gmail.com wangzaozhi22@mails.ucas.ac.cn weilonghui1@huawei.com zxphistory@gmail.com yaoweiwang@bit.edu.cn jiaojb@ucas.ac.cn tian.qil@huawei.com qxye@ucas.ac.cn

Abstract

In this paper, we present an integral pre-training framework based on masked image modeling (MIM). We advocate for pre-training the backbone and neck jointly so that the transfer gap between MIM and downstream recognition tasks is minimal. We make two technical contributions. First, we unify the reconstruction and recognition necks by inserting a feature pyramid into the pre-training stage. Second, we complement mask image modeling (MIM) with masked feature modeling (MFM) that offers multi-stage supervision to the feature pyramid. The pre-trained models, termed integrally pre-trained transformer pyramid networks (iTPNs), serve as powerful foundation models for visual recognition. In particular, the base/large-level iTPN achieves an 86.2%/87.8% top-1 accuracy on ImageNet-1K, a 53.2%/55.6% box AP on COCO object detection with 1x training schedule using Mask-RCNN, and a 54.7%/57.7% mIoU on ADE20K semantic segmentation using UPerHead – all these results set new records. Our work inspires the community to work on unifying upstream pre-training and downstream fine-tuning tasks. Code is available at github.com/sunsmarterjie/iTPN.

1. Introduction

Recent years have witnessed two major progresses in visual recognition, namely, the vision transformer architecture [22] as network backbone and masked image modeling (MIM) [3, 28, 68] for visual pre-training. Combining these two techniques yields a generalized pipeline that achieves state-of-the-arts in a wide range of visual recognition tasks, including image classification, object detection, and instance/semantic segmentation.

One of the key issues of the above pipeline is the transfer gap between upstream pre-training and downstream fine-

*Corresponding author.
features. However, most existing pre-training tasks (e.g., BEiT [3] and MAE [28]) were built upon plain vision transformers. Even if hierarchical vision transformers have been used (e.g., in SimMIM [68], ConvMAE [25], and Green-MIM [33]), the pre-training task only affects the backbone but leaves the neck (e.g., a feature pyramid) un-trained. This brings extra risks to downstream fine-tuning as the optimization starts with a randomized initialized neck which is not guaranteed to cooperate with the pre-trained backbone.

In this paper, we present an integral pre-training framework to alleviate the risk. We establish the baseline with HiViT [74], an MIM-friendly hierarchical vision transformer, and equip it with a feature pyramid. To jointly optimize the backbone (HiViT) and neck (feature pyramid), we make two-fold technical contributions. First, we unify the upstream and downstream necks by inserting a feature pyramid into the pre-training stage (for reconstruction) and reusing the weights in the fine-tuning stage (for recognition). Second, to better pre-train the feature pyramid, we propose a new masked feature modeling (MFM) task that (i) computes intermediate targets by feeding the original image into a moving-averaged backbone, and (ii) uses the output of each pyramid stage to reconstruct the intermediate targets. MFM is complementary to MIM and improves the accuracy of both reconstruction and recognition. MFM can also be adapted to absorb knowledge from a pre-trained teacher (e.g., CLIP [52]) towards better performance.

The obtained models are named integrally pre-trained pyramid transformer networks (iTPNs). We evaluate them on standard visual recognition benchmarks. As highlighted in Figure 1, the iTPN series report the best known downstream recognition accuracy. On COCO and ADE20K, iTPN largely benefits from the pre-trained feature pyramid. For example, the base/large-level iTPN reports a 53.2%/55.6% box AP on COCO (1× schedule, Mask R-CNN) and a 54.7%/57.7% mIoU on ADE20K (UPerNet), surpassing all existing methods by large margins. On ImageNet-1K, iTPN also shows significant advantages, implying that the backbone itself becomes stronger during the joint optimization with neck. For example, the base/large-level iTPN reports an 86.2%/87.8% top-1 classification accuracy, beating the previous best record by 0.7%/0.5%, which is not small as it seems in such a fierce competition. In diagnostic experiments, we show that iTPN enjoys both (i) a lower reconstruction error in MIM pre-training and (ii) a faster convergence speed in downstream fine-tuning – this validates that shrinking the transfer gap benefits both upstream and downstream parts.

Overall, the key contribution of this paper lies in the integral pre-training framework that, beyond setting new state-of-the-arts, enlightens an important future research direction – unifying upstream pre-training and downstream fine-tuning to shrink the transfer gap between them.

2. Related Work

In the deep learning era [36], visual recognition algorithms are mostly built upon deep neural networks. There are two important network backbones in the past decade, namely, the convolutional neural networks [31,35,53] and the vision transformers [23,45,59,74]. This paper focuses on the vision transformers which were transplanted from the natural language processing field [58]. The core idea is to extract visual features by treating each image patch as a token and computing self-attentions among them.

The vanilla vision transformers appeared in a plain form [11,23,41,70,75] where, throughout the backbone, the number of tokens keeps a constant and the attention among these tokens are totally symmetric. To compensate the inductive priors in computer vision, the community designed hierarchical vision transformers [16,45,59,65,74] that allow the number of tokens to gradually decrease throughout the backbone, i.e., similar as in convolutional neural networks. Other design principles were also inherited, such as introducing convolution into the transformer architecture so that the relationship between neighborhood tokens is better formulated [16,24,40,47,59,60], interacting between hybrid information [51], using window [20,45,74] or local [71] self-attentions to replace global self-attentions, adjusting the geometry for local-global interaction [69], decomposing self-attentions [57], and so on. It was shown that hierarchical vision transformers can offer high-quality, multi-level visual features that easily cooperate with a neck module (for feature aggregation, e.g., a feature pyramid [42]) and benefit downstream visual recognition tasks.

The continuous growth of vision data calls for visual pre-training, in particular, self-supervised learning that learns generic visual representations from unlabeled vision data. At the core of self-supervised learning lies a pretext task, i.e., an unsupervised learning objective that the model pursues. The community started with preliminary pretext tasks such as geometry-based tasks (e.g., determining the relative position between image patches [19,48,62] or the transformation applied to an image [26]), and generation-based tasks (e.g., recovering the removed contents [49] or attributes [55,56,72] of an image), but these methods suffer unsatisfying accuracy (i.e., trailed by fully-supervised pre-training significantly) when transferred to downstream recognition tasks. The situation was changed when new pretext tasks were introduced, in particular, contrastive learning [7,8,12,27,29,74], and masked image modeling (MIM) [1,3,5,15,28,37,38,68], where the latter is yet another type of generation-based learning objective.

This paper focuses on MIM, which takes the advantage of vision transformers that formulate each image patch into a token. Hence, the tokens can be arbitrarily masked (discarded from the input data) and the learning objective is to recover the masked contents at the pixel level [28,55,68],
the feature level [3, 61], or in the frequency space [44]. MIM has shown an important property named scalability, i.e., augmenting the amount of pre-training data (e.g., from ImageNet-1K to ImageNet-22K) and/or increasing the model size (e.g., from the base level to the large or huge level) can boost the downstream performance [13, 28], which aligns with the observations in language modeling [5, 18].

Most existing MIM methods worked on the plain vision transformers, yet the hierarchical vision transformers have higher potentials in visual recognition. The first work which tried to bridge the gap was SimMIM [68], but the overall pre-training overhead was largely increased because the entire image (with dummy masked patches) were fed to the encoder. This issue was later alleviated by reforming the hierarchical vision transformers [33, 74] to fit MIM better. This paper inherits the design and goes one step further by integrating the neck (e.g., a feature pyramid) into the pre-training phase, constructing the integrally pre-trained transformer pyramid network.

3. The Proposed Approach

3.1. Motivation: Integral Pre-Training

We first establish a notation system. The pre-training stage is built upon a dataset \( D^{\text{pt}} = \{x_n^{\text{pt}}\}_{n=1}^N \), where \( N \) is the number of samples. Note that these samples are not equipped with labels. The fine-tuning phase involves another dataset \( D^{\text{ft}} = \{x_m^{\text{ft}}, y_m^{\text{ft}}\}_{m=1}^M \), where \( M \) is the number of samples and \( y_m^{\text{ft}} \) is the semantic label of \( x_m^{\text{ft}} \). Let the target deep neural network be composed of backbone, neck, and head\(^1\), denoted as \( f(\cdot; \theta), g(\cdot; \phi), h(\cdot; \psi) \), respectively, where \( \theta, \phi, \psi \) are learnable parameters and can be omitted for simplicity. \( f(\cdot) \) directly takes \( x \) as input, while \( g(\cdot) \) and \( h(\cdot) \) works on the outputs of \( f(\cdot) \) and \( g(\cdot) \), i.e., the entire function is \( h(g(f(x; \theta); \phi); \psi) \).

Throughout this paper, the pre-training task is masked image modeling (MIM) and the fine-tuning tasks can be image classification, object detection, and instance/semantic segmentation. Existing approaches assumed that they share the same backbone, but need different necks and heads. Mathematically, the pre-training and fine-tuning objectives are written as:

\[
\begin{align*}
\min_{\theta, \phi, \psi} & \mathbb{E}_{D^{\text{pt}}}[\|x_n^{\text{pt}} - h^{\text{pt}}(f(x_n^{\text{pt}}; \theta), \phi^{\text{pt}}), \psi^{\text{pt}})\|], \\
\min_{\theta, \phi, \psi} & \mathbb{E}_{D^{\text{ft}}}[\|y_m^{\text{ft}} - h^{\text{ft}}(f(x_m^{\text{ft}}; \theta), \phi^{\text{ft}}), \psi^{\text{ft}})\|],
\end{align*}
\]

(1)

where parameters are not shared between \( \phi^{\text{pt}}, \phi^{\text{ft}} \) and \( \psi^{\text{pt}}, \psi^{\text{ft}} \). We argue that such a pipeline leads to a significant transfer gap between pre-training and fine-tuning, and thus brings two-fold drawbacks. First, the backbone parameters, \( \theta \), are not optimized towards being used for multi-level feature extraction. Second, the fine-tuning phase starts with optimizing a randomly initialized \( \phi^{\text{ft}} \) and \( \psi^{\text{ft}} \), which may slow down the training procedure and lead to unsatisfying recognition results. To alleviate the gap, we advocate for an integral framework that unifies \( g^{\text{pt}}(\cdot) \) and \( g^{\text{ft}}(\cdot) \), so that the pre-trained \( \phi^{\text{pt}} \) is easily reused to be an initialization of \( \phi^{\text{ft}} \), and thus only \( \psi^{\text{ft}} \) is randomly initialized.

The overall framework is illustrated in Figure 2.

\(^1\text{We follow the conventional definition that the neck is used for multi-stage feature aggregation (e.g., a feature pyramid [42]) while the head is used for final prediction (e.g., a linear classifier).}\)
3.2. Unifying Reconstruction and Recognition

Let a hierarchical vision transformer contain \( S \) stages and each stage has several transformer blocks. Most often, the backbone \( \text{a.k.a. encoder} \) gradually down-samples the input signal and produces \( S + 1 \) feature maps:

\[
f(x; \theta) = U^0, U^1, \ldots, U^S,
\]

where \( U^0 \) denotes the direct embedding of input, and a smaller superscript index indicates a stage closer to the input layer. Each feature map is composed of a set of tokens (feature vectors), \( i.e., U^s = \{u^s_1, u^s_2, \ldots, u^s_{K^s}\} \), where \( K^s \) is the number of tokens in the \( s \)-th feature map.

We show that \( g^{\text{pt}}(\cdot) \) and \( g^{\text{pt}}(\cdot) \) can share the same architecture and parameters because both of them start with \( U^S \) and gradually aggregate it with lower-level features. Thus, we write the neck part as follows:

\[
\begin{align*}
V^S &= U^S, \\
V^s &= U^s + g^s(V^{s+1}; \phi^s), \quad 1 \leq s < S,
\end{align*}
\]

where \( g^s(\cdot) \) up-samples \( V^{s+1} \) to fit the resolution of \( V^s \). Note that the learnable parameters, \( \phi \), are composed of a layer-wise parameter set, \( \{\phi^s\} \). With these parameters being reused in fine-tuning, we largely shrink the transfer gap: the only modules that remain individual between pre-training and fine-tuning are the heads \( e.g., \) the decoder for MIM vs. the Mask R-CNN head for detection.

Before entering the next part that discusses the loss terms, we remind the readers that other differences exist between pre-training and fine-tuning, while they do not impact the overall design of network architectures.

- MIM samples a random mask \( M \) and applies it to \( x \), \( i.e., \) \( x \) is replaced by \( x \odot M \). Consequently, all the backbone outputs, \( U^0, \ldots, U^S \), do not contain the tokens with indices in \( M \), and so are \( V^1, \ldots, V^S \). At the start of decoder, \( V = \sum_{s=1}^{S} V^s \) is complemented by adding dummy tokens to the masked indices, and then fed into a decoder for image reconstruction.

- The downstream fine-tuning procedure makes use of specific outputs of decoder for different tasks. For image classification, \( V^S \) is used. For detection and segmentation, all of \( V^1, \ldots, V^S \) are used.

3.3. Masked Feature Modeling

We first inherit the reconstruction loss from MIM that minimizes \( \|x - h^{\text{pt},0}(V; \psi^{\text{pt},0})\|_2 \), where \( h^{\text{pt},0}(\cdot) \) involves a few transformer blocks that reconstruct the original image from \( V = \sum_{s=1}^{S} V^s \). To acquire the ability of capturing multi-stage features, we add a reconstruction head to each stage, termed \( h^{\text{pt},s}(\cdot; \psi^{\text{pt},s}) \), and optimize the following multi-stage loss:

\[
\mathcal{L} = \|x - h^{\text{pt},0}(V)\|_2 + \lambda \sum_{s=1}^{S} \|x^s - h^{\text{pt},s}(V^s)\|_2,
\]

where \( x^s \) is the expected output at the \( s \)-th decoder stage, and \( \lambda = 0.3 \) is determined in a held-out validation set. Since the goal is to recover the masked features, we name the second term as the masked feature modeling (MFM) loss that complements the first term, the masked image modeling (MIM) loss. We illustrate MFM in Figure 2.

The remaining issue is to determine the intermediate reconstruction target, \( i.e., \) \( x^1, \ldots, x^S \). We borrow the idea from knowledge distillation [32] that makes use of a teacher backbone \( f^{\text{back}}(\cdot) \) to generate the intermediate targets, \( i.e., \) \( \hat{f}(x; \theta) = x^1, \ldots, x^S \). The teacher model is chosen to be the moving-averaged [54] encoder (in this case, no external knowledge is introduced) or another pre-trained model \( e.g., \) CLIP [52], as used by [63, 64], that was pre-trained on a large dataset of image-text pairs. In the former case, we only feed the masked patches \( x \odot M, \) not the entire image, to the teacher model for acceleration. In the latter case, we follow BEiT [3] to feed the entire image to the pre-trained CLIP model.

3.4. Technical Details

We build the system beyond HiViT [74], a recently proposed, hierarchical vision transformer. HiViT simplified the Swin transformers [45] by (i) replacing early-stage shifted-window attentions with channel-wise multi-layer perceptrons (C-MLPs) and (ii) removing the \( 7 \times 7 \) stage so that global attentions are computed on the \( 14 \times 14 \) stage. With these improvements, when applied to MIM, HiViT allows the masked tokens to be directly discarded from input (by contrast, with Swin as the backbone, SimMIM [68] required the entire image to be used as input), saving 30%-50% computational costs and leading to better performance.

Table 1 summarizes the configuration of iTPN. We follow the convention to use \( 224 \times 224 \) images during the pre-training. HiViT produces three stages (\( S = 3 \)) with spatial resolutions of \( 56 \times 56, 28 \times 28 \), and \( 14 \times 14 \), respectively. An \( S \)-stage feature pyramid is built upon the backbone. We replace all convolutions in the feature pyramid with C-MLPs to avoid leaking information from visible patches to invisible patches. As we shall see in ablation (Section 4.4), using C-MLP in the feature pyramid leads to consistent accuracy gain in various visual recognition tasks, and the improvement is complementary to that brought by MFM.

Regarding MFM, we investigate two choices of the teacher model. (i) The first option involves computing the exponential moving average (EMA) of the online target
model with a coefficient of 0.996. We extract the supervision from the last layer of each stage, so that for any $s$, $x^s$ has the same spatial resolution as $V^2$, and thus $h^s(\cdot)$ is a linear projection working on each token individually. (ii) The second option directly inherits a CLIP pre-trained model. Note that CLIP offers standard ViTs that do not produce multi-resolution feature maps. In this scenario, we unify the $S$ MFM terms into one by down-sampling all the feature maps to the lowest spatial resolution $(14 \times 14)$, adding them together, and comparing the sum to the last-layer output of the CLIP model.

4. Experiments

4.1. Settings and Implementation Details

We pre-train iTPN on the ImageNet-1K dataset [17], a subset of ImageNet that contains 1.28M training images of 1,000 classes. The class labels are not used during the pre-training stage. Each training image is pre-processed into $224 \times 224$ and partitioned into $14 \times 14$ patches sized $16 \times 16$ pixels. Following MAE [28], a random subset of 75% patches are masked from input, and the normalized pixels are preserved for reconstruction.

We use an AdamW optimizer [46] with an initial learning rate of $1.5 \times 10^{-4}$, a weight decay of 0.05, and batch size of 4,096. The learning rate follows a cosine annealing schedule and the number of warm-up epochs is set to be 40. The numbers of pre-training epochs are 400 and 1,600 in the former scenario, or 300 and 800 in the latter scenario\(^2\). We train all these models using 64 NVIDIA Tesla-V100 GPUs. For the $base$-level models, one pixel-supervised epoch and one CLIP-supervised epoch take about 2.7 and 4.7 minutes, respectively. For the $large$-level models, the numbers are 4.2 and 12.0 minutes, respectively. That said, a 1600-epoch pixel-supervised pre-training of iTPN-$base$/$large$ takes around 75/115 hours.

4.2. Image Classification

Fine-tuning We report results of ImageNet-1K classification. The number of epochs is 100 for $base$-level models and 50 for $large$-level models. We use the AdamW optimizer, with the initial learning rate being $5 \times 10^{-4}$ and $1 \times 10^{-3}$ for $base$-level and $large$-level models, respectively. The weight decay is 0.05 and the batch size is 1,024. The number of warm-up epochs is 5. The layer decay is set to be 0.55 and 0.50 for $base$-level and $large$-level models.

Results are summarized in Table 2. One can see that iTPN achieves higher accuracy than existing methods on all tracks, i.e., using $base$-level or $large$-level backbones, with or without external supervision (i.e., CLIP [52]). For example, using the $base$-level backbone, iTPN achieves an 85.1% accuracy with only 400 pre-training epochs, surpassing MAE [28] and HiViT [74] with 1,600 epochs. The accuracy of iTPN continues growing to 85.5% with 1,600 pre-training epochs, which is on par with BEiT-v2 [50] that distilled knowledge from CLIP-B [52] (1,600 epochs), yet iTPN reports an 86.2% accuracy with the supervision of CLIP (800 epochs). Similar situations occur when we use the $large$-level backbone, where the advantage of iTPN is a bit smaller due to the higher baseline. The best practice appears when an iTPN-L/14 model (i.e., patch size is adjusted to $14 \times 14$) is supervised by a CLIP-L teacher—the classification accuracy, 88.0%, is the highest to date under fair comparisons.

Linear probing We then evaluate the pre-trained models using the linear probing. Following the convention, we train the models for 90 epochs using the LARS optimizer [34] with a batch size of 16,384 and a learning rate of 0.1. Specifically, the iTPN-B (pixel) model with 1,600 pre-training epochs reports a 71.6% accuracy, surpassing 1,600-epoch MAE [28] by a significant margin of 3.8%, as well as 400-epoch BEiT, 800-epoch SimMIM, and 1,600-epoch CAE by 22.2%, 14.9%, and 1.2%, respectively. With CLIP supervision, iTPN with 300 epochs of pre-training reports a 77.3% accuracy, surpassing MVP [63] with the same setting by 1.9%.

Insights Note that image classification experiments do not involve transferring the pre-trained neck, i.e., the feature pyramid. That said, iTPN achieves higher classification accuracy with the pre-trained backbone alone. This implies that (i) a joint optimization of the backbone and neck leads to a stronger backbone, and hence, (ii) the derived backbone can be directly transferred for various vision tasks, extending iTPN to more application scenarios.

\(^2\)By using CLIP as supervision, each pre-training epoch takes longer time but the pre-training converges faster. So, we adjust the number of pre-training epochs according to the computational budget.
4.3. Detection and Segmentation

COCO: object detection & instance segmentation

We follow the configuration provided by [13] to evaluate the pre-trained models on the COCO [43] dataset. We use Mask R-CNN [30] implemented by MMDetection [10]. We use the AdamW optimizer [46] with a weight decay of 0.05. The standard 1 × (12 epochs) and 3 × schedules are applied, where the initial learning rate is 3 × 10⁻⁴ and it decays by a factor of 10 after 3/4 and 11/12 of fine-tuning epochs. The layer-wise decay rate is set to be 0.90. We also try a 3 × Cascade Mask R-CNN [6] towards higher accuracy.

Results are summarized in Table 4. Compared to image classification, the advantages of iTPN become more significant because the pre-trained neck is reused so that the fine-tuning stage only needs to initialize a task-specific head. For example, using a pixel-supervised base-level backbone, the 1 × Mask R-CNN produces 53.0% box AP, surpassing all other methods significantly (e.g., +4.6% over MAE [28] and +3.0% over CAE [13]). Compared to HiViT that did not pre-train the feature pyramid, iTPN claims a +1.7% gain in box AP. With stronger heads, iTPN reports stronger numbers, e.g., the box/mask AP is 56.0%/48.5% using 3 × Cascade Mask R-CNN, setting a new record with base-level models. Later, we will show that the benefits indeed come from pre-training the feature pyramid and loading it for downstream fine-tuning.

ADE20K: semantic segmentation

We follow BeiT [3] to build a UperHead [66] on top of the pre-trained backbone. We use the AdamW optimizer [46] and the learning rate is fixed as 3 × 10⁻⁵. We fine-tune the model for a total of 160k iterations and the batch size is 16. The input resolution is 512 × 512 and we do not use a multi-scale test. Results are summarized in Table 4. Again, iTPN reports the best accuracy in terms of mIoU. In particular, the pixel-supervised base/large-level models report 53.5%/56.1% mIoUs which surpass all the competitors substantially. Introducing CLIP supervision further improves both numbers by more than 1%, setting solid new records for both base-level and large-level models.
Table 4. Visual recognition results (%) on COCO (object detection and instance segmentation, AP) and ADE20K (semantic segmentation, mIoU). Mask R-CNN (abbr. MR, $1 \times$) and Cascade Mask R-CNN (abbr. CMR, $3 \times$) are used on COCO, and UPerHead with $512 \times 512$ input is used on ADE20K. For the base-level models, each cell of COCO results contains object detection (box) and instance segmentation (mask) APs. For the large-level models, the accuracy of $1 \times$ Mask R-CNN surpasses all existing methods. $\dagger$: ConvMAE used a different setting from all other methods – fine-tuning using ViTDet [39] for 25 epochs. $\ddagger$: More techniques, such as multi-scale test and softnms [4] etc, are used in the test stage.

<table>
<thead>
<tr>
<th>Method</th>
<th>Arch.</th>
<th>Sup.</th>
<th>Eps.</th>
<th>Param. (M)</th>
<th>COCO</th>
<th>ADE20K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MR, 1×</td>
<td>COCO, 3×</td>
<td></td>
</tr>
<tr>
<td>MoCo-v3 [14]</td>
<td>ViT-B</td>
<td>pixel</td>
<td>300</td>
<td>86</td>
<td>45.5/40.5</td>
<td>47.3</td>
</tr>
<tr>
<td>BEiT [3]</td>
<td>ViT-B</td>
<td>DALL-E</td>
<td>400</td>
<td>86</td>
<td>42.1/37.8</td>
<td>47.1</td>
</tr>
<tr>
<td>DINo [9]</td>
<td>ViT-B</td>
<td>pixel</td>
<td>400</td>
<td>86</td>
<td>46.8/41.5</td>
<td>47.2</td>
</tr>
<tr>
<td>iBoT [76]</td>
<td>ViT-B</td>
<td>pixel</td>
<td>1600</td>
<td>86</td>
<td>–</td>
<td>50.0</td>
</tr>
<tr>
<td>CAE [13]</td>
<td>ViT-B</td>
<td>DALL-E</td>
<td>1600</td>
<td>86</td>
<td>50.0/44.0</td>
<td>50.2</td>
</tr>
<tr>
<td>SimMIM [68]</td>
<td>Swin-B</td>
<td>pixel</td>
<td>800</td>
<td>88</td>
<td>52.3/–</td>
<td>52.8</td>
</tr>
<tr>
<td>MAE [28]</td>
<td>ViT-B</td>
<td>pixel</td>
<td>1600</td>
<td>86</td>
<td>48.4/42.6</td>
<td>48.1</td>
</tr>
<tr>
<td>ConvMAE [25]</td>
<td>ConViT-B</td>
<td>pixel</td>
<td>1600</td>
<td>88</td>
<td>53.2/47.1$\dagger$</td>
<td>51.7</td>
</tr>
<tr>
<td>HiViT [28]</td>
<td>HiViT-B</td>
<td>pixel</td>
<td>1600</td>
<td>66</td>
<td>51.3/44.6</td>
<td>52.8</td>
</tr>
<tr>
<td>MVP [63]</td>
<td>ViT-B</td>
<td>CLIP-B</td>
<td>300</td>
<td>86</td>
<td>–</td>
<td>53.5/46.3</td>
</tr>
<tr>
<td>iTPN (ours)</td>
<td>HiViT-B</td>
<td>pixel</td>
<td>1600</td>
<td>79</td>
<td>53.0/46.5</td>
<td>53.5</td>
</tr>
<tr>
<td>iTPN (ours)</td>
<td>HiViT-B</td>
<td>CLIP-B</td>
<td>800</td>
<td>79</td>
<td>53.2/46.6</td>
<td>54.2</td>
</tr>
<tr>
<td>Method</td>
<td>Arch.</td>
<td>Sup.</td>
<td>Eps.</td>
<td>Param. (M)</td>
<td>COCO object det.</td>
<td>COCO instance seg.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>COCO</td>
<td></td>
</tr>
<tr>
<td>MAE [28]</td>
<td>ViT-L</td>
<td>pixel</td>
<td>1600</td>
<td>307</td>
<td>54.0</td>
<td>47.1</td>
</tr>
<tr>
<td>SimMIM [68]</td>
<td>Swin-L</td>
<td>pixel</td>
<td>800</td>
<td>197</td>
<td>53.8</td>
<td>–</td>
</tr>
<tr>
<td>SimMIM [68]</td>
<td>SwinV2-H</td>
<td>pixel</td>
<td>800</td>
<td>658</td>
<td>54.4</td>
<td>–</td>
</tr>
<tr>
<td>CAE [28]</td>
<td>ViT-L</td>
<td>pixel</td>
<td>1600</td>
<td>304</td>
<td>54.5</td>
<td>47.6</td>
</tr>
<tr>
<td>iTPN (ours)</td>
<td>HiViT-L</td>
<td>pixel</td>
<td>1600</td>
<td>288</td>
<td>55.6</td>
<td>48.6</td>
</tr>
<tr>
<td>iTPN (ours)</td>
<td>HiViT-L</td>
<td>CLIP-L</td>
<td>300</td>
<td>288</td>
<td>55.2</td>
<td>48.2</td>
</tr>
<tr>
<td>iTPN (ours)</td>
<td>HiViT-L</td>
<td>pixel</td>
<td>1600</td>
<td>288</td>
<td>58.0$\ddagger$</td>
<td>50.3</td>
</tr>
</tbody>
</table>

Figure 3. A comparison between the attention maps generated by iTPN, the variant without integral pre-training (w/o iPT), and the MIM baseline (MAE [28]). In each case, the red block indicates the query token, and the attention map between the query and other tokens at the corresponding transformer block is shown. We use $224 \times 224$ input images in (a), (b), and $512 \times 512$ images in (c).
Table 5. Ablations on whether the model is integrally pre-trained (iPT) and whether the feature pyramid is loaded for detection and segmentation. Fine-tuning on ImageNet-1K does not involve loading the pyramid. The numbers are in % for classification accuracy, box AP, and mIoU. The models are pre-trained for 400 epochs. For COCO, 1 × Mask R-CNN is used and box AP is reported.

<table>
<thead>
<tr>
<th></th>
<th>ImageNet-1K</th>
<th>COCO</th>
<th>ADE20K</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
<td>84.4</td>
<td>50.6</td>
<td>51.5</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>51.5</td>
<td>51.8</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>85.1</td>
<td>52.1</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
<td>52.2</td>
</tr>
</tbody>
</table>

Table 6. Ablations on C-MLP and MFM. The settings remain the same as in Table 5. The ∗ sign indicates that convolution is used (instead of C-MLP) for both the backbone and feature pyramid, which leads to worse recognition results.

<table>
<thead>
<tr>
<th>C-MLP</th>
<th>MFM</th>
<th>ImageNet-1K</th>
<th>COCO</th>
<th>ADE20K</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
<td>✗</td>
<td>84.3</td>
<td>49.8</td>
<td>50.0</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>84.6</td>
<td>50.8</td>
<td>50.7</td>
</tr>
<tr>
<td>✓</td>
<td>✗</td>
<td>84.9</td>
<td>51.8</td>
<td>51.8</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td><strong>85.1</strong></td>
<td>52.1</td>
<td><strong>52.2</strong></td>
</tr>
</tbody>
</table>

4.4. Analysis

Ablative studies Throughout this part, we use the 400-epoch pixel-supervised model for diagnosis. We first ablate the benefit of integral pre-training. As shown in Table 5, jointly optimizing the backbone and neck leads to higher recognition accuracy on all datasets including ImageNet-1K, COCO, and ADE20K. Beyond this point, loading the pre-trained feature pyramid (neck) further improves the recognition accuracy on COCO and ADE20K. This validates that the backbone itself is strengthened by iTPN, and thus it can be transferred to downstream tasks independently of the neck.

Next, we investigate the technical details of integral pre-training, in particular, using channel-wise multi-layer perceptron (C-MLP) in the feature pyramid and applying masked feature modeling (MFM) for multi-stage supervision. As shown in Table 6, both C-MLP and MFM contribute individually to recognition accuracy, meanwhile, integrating them yields even better recognition performance.

Visualization In Figure 3, we visualize the attention maps generated by iTPN and baseline methods. (1) On the encoder, iTPN shows the advantage of detecting complete objects on ImageNet and concentrating on the chosen object on COCO. Such ability arises because iTPN forces the model to preserve richer visual features (multi-scale feature maps), which facilitates better recognition results in downstream. (2) On the decoder, iTPN can still realize the semantic relationship between tokens, resulting in better reconstruction results (Figure 4). We owe such benefits to the pre-trained neck that aggregates multi-stage visual features.

The benefits brought by more complete attentions can be quantified using two-fold experiments shown in Figure 4. (1) In the left part, we observe that iTPN achieves better reconstruction results (i.e., lower reconstruction loss values). Note that simply using a hierarchical vision transformer (with multi-scale feature maps) does not improve reconstruction, implying that integral pre-training is the major contributor. (2) In the right part, we show that better depiction of objects helps downstream visual recognition tasks (e.g., object detection) converge faster and achieve a higher upper-bound – this aligns with the outstanding accuracy on COCO (see Section 4.3). Integrating these analysis, we conclude that iTPN successfully transfers the benefits from upstream pre-training (reconstruction) to downstream fine-tuning (recognition), completing the entire chain.

5. Conclusions and Future Remarks

In this paper, we present an integral framework for pre-training hierarchical vision transformers. The core contribution lies in a unified formulation that uses a feature pyramid for both reconstruction and recognition, so that the transfer gap between pre-training and fine-tuning is maximally reduced. Besides, a masked feature modeling (MFM) task is designed to complement masked image modeling (MIM) for a better optimization of the feature pyramid. The pre-trained iTPNs report superior recognition in a few popular visual recognition tasks. Our work clearly enlightens a future direction – designing a unified framework for upstream and downstream visual representation learning.

6. Acknowledgement

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 6225208, 62171431 and 61836012, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27000000.
References


[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In IEEE CVPR, pages 16000–16009, 2022. 1, 2, 3, 5, 6, 7


[34] Zhuyuan Huo, Bin Gu, and Heng Huang. Large batch optimization for deep learning using new complete layer-wise adaptive rate scaling. In AAAI, 2021. 5


[38] Xiaotong Li, Yixiao Ge, Kun Yi, Zixuan Hu, Ying Shan, and Yanghao Li, Mao Hanzi Mao, Ross Girshick, and Kaiming He, Bharath Hariharan, and Sergey J. Belongie. Feature pyramid networks for object detection. In IEEE CVPR, pages 936–944, 2017. 2, 3


[40] Hao Liu, Xinghua Jiang, Xin Li, Antai Guo, Deqiang Jiang, and Bo Ren. The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training. CoRR, abs/2204.08227, 2022. 3


[50] Anit Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. NeurIPS, 30, 2017. 4


[54] Hao Liu, Xinghua Jiang, Xin Li, Antai Guo, Deqiang Jiang, and Bo Ren. The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training. CoRR, abs/2204.08227, 2022. 3


[61] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In IEEE CVPR, pages 14668–14678, 2022. 3, 6


[68] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In IEEE CVPR, pages 9653–9663, 2022. 1, 2, 3, 4, 6, 7


[74] Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In International Conference on Learning Representations, 2023. 2, 3, 4, 5, 6
