Supplemental Material

Anonymous Author(s)
Affiliation
Address
email

1 Appendix

Architecture Details of ConvMAE Encoder. The details of our hybrid convolution-transformer encoder is explained below. Given an input image $I \in \mathbb{R}^{3\times H\times W}$, stage 1 of ConvMAE encoder generates a high-resolution token embeddings $E_1 \in \mathbb{R}^{C_1\times \frac{H}{4}\times \frac{W}{4}}$ using non-overlapping 4×4 strided convolution firstly. Then E_1 is feed into stacked convolutional blocks which is repeated L_1 times, where L_1 stands for the number of layers in stage 1. Similar as stage 1, stage 2 further downsamples feature map into token embeddings $E_2 \in \mathbb{R}^{C_2\times \frac{H}{8}\times \frac{W}{8}}$ using non-overlapping 2×2 strided convolution. $E2$ is processed by $L2$ layers of convolutional blocks again. After local information fusion utilized in stage 1 and stage 2, stage 3 perform global feature fusion using transformer block. E_2 is projected into tokens embeddings $E_3 \in \mathbb{R}^{(\frac{H}{16}\times \frac{W}{16})\times C_3}$ using non-overlapping 2×2 strided convolution. E_3 mixing with Intermediate Positional Embedding (IPL) is feed into a pure transformer block with L_3 layers. We denote the number of attention heads in stage 3 as H_a. The mlp-ratios in FFN for different stages is denoted as P_1, P_2 and P_3 in respectively. Stage 1 and stage 2 is designed to capture fine-grained details on high resolution feature map. Stage 3 can perform dynamically global reasoning efficiently on a rather low-resolution feature map. At the same time, stage 3 can enlarge the filed-of-view (FOV) of backbone which benefits a wide range of downstream tasks. The encoder of ConvMAE can seamlessly inherits the merits of convolution and transformer block. The architecture details for small, base and large model is listed in Table 1. ConvMAE small, base, large and huge share similar parameter scale with the encoder of MAE-small, MAE-base, MAE-large and MAE-huge.

Model	$[C_1, C_2, C_3]$	$[L_1, L_2, L_3]$	$[E_1, E_2, E_3]$	$[P_1, P_2, P_3]$	H_a	#Params (M)
ConvMAE-S	[128, 256, 384]	[2, 2, 11]	[56, 28, 14]	[4, 4, 4]	6	22
ConvMAE-B	[256, 384, 768]	[2, 2, 11]	[56, 28,14]	[4, 4, 4]	12	84
ConvMAE-B*	[256, 384, 768]	[2, 2, 11]	[56, 28,14]	[8, 8, 4]	12	88
ConvMAE-L	[384, 768, 1024]	[2, 2, 23]	[56, 28, 14]	[8, 8, 4]	16	322
ConvMAE-H	[768, 1024, 1280]	[2, 2, 31]	[56, 28, 14]	[8, 8, 4]	16	666

Table 1: Architecture details of ConvMAE small, base, large and huge. ConvMAE-B* represents multi-scale encoder with large mlp-ratios in stage 1 and stage 2. $[C_1, C_2, C_3]$, $[L_1, L_2, L_3]$, $[E_1, E_2, E_3]$ and $[P_1, P_2, P_3]$ represents channel dimension, number of layer, spatial resolution and mlp-ratios for each stage 1, stage 2 and stage 3. H_a stands for the number of attention heads in stage 3.

Architecture Details of VideoConvMAE To show exactly how we expand the 2D ConvMAE into VideoConvMAE, we detail the convolution kernel size and output shape in Table 2. Output shape is described in $C\times T\times H\times W$ format, where C is the feature dimension and T, H, W are the time span, height and width, respectively. ρ is the mask ratio, for which we use 0.9 by default. Patch embeds are expanded into cube embeds, performing the same non-overlapping convolution but in 3D, with the kernel size (described in $k_T\times k_H\times k_W$ format) and output channel number specified in the table. Note that we perform temporal downsampling only at data layer and the first cube embed

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Stage	**Blocks**	**Output Shape** $(C \times T \times H \times W)$
data	K400 Sample Rate: 4 SSv2 Sample Rate: 2	$3 \times 16 \times 224 \times 224$
cube embed 1	Kernel Size $2 \times 4 \times 4$ Output Channel 256	$256 \times 8 \times [56 \times 56 \times (1-\rho)]$
conv stage 1	$\begin{bmatrix} \text{DW}_{3\times5\times5}(256) \\ \text{MLP}(1024) \end{bmatrix} \times 2$	$256 \times 8 \times [56 \times 56 \times (1-\rho)]$
cube embed 2	Kernel Size $1 \times 2 \times 2$ Output Channel 384	$384 \times 8 \times [28 \times 28 \times (1-\rho)]$
conv stage 2	$\begin{bmatrix} \text{DW}_{3\times5\times5}(384) \\ \text{MLP}(1536) \end{bmatrix} \times 2$	$384 \times 8 \times [28 \times 28 \times (1-\rho)]$
cube embed 3	Kernel Size $1 \times 2 \times 2, 768$ Output Channel 768	$768 \times 8 \times [14 \times 14 \times (1-\rho)]$
attn stage 3	$\begin{bmatrix} \text{MHA}(768, 12) \\ \text{MLP}(3072) \end{bmatrix} \times 11$	$768 \times 8 \times [14 \times 14 \times (1-\rho)]$
projection	FC(512) concat learnable tokens	$512 \times 8 \times 14 \times 14$
decoder	$\begin{bmatrix} \text{MHA}(512, 16) \\ \text{MLP}(2048) \end{bmatrix} \times 4$	$512 \times 8 \times 14 \times 14$
projection	FC(1536) reshape to input shape	$3 \times 16 \times 224 \times 224$

Table 2: Detailed structure of VideoConvMAE-B with 16 input frames for video recognition. See text for an explanation of the table.

layer. Other blocks are denoted as follows: $\text{DW}_{k_T \times k_T \times k_W}(c)$ is a depthwise convolution block with channel number c, consisting of two linear projections and a depthwise convolution in the middle; $\text{MLP}(c)$ is a two-layer perception with the feature channel expanded to c in the middle; $\text{MHA}(c, h)$ is a multi-head self attention block with channel number c and head number h; $\text{FC}(c)$ is a single linear projection layer with output channel number c used to align the feature dimensions between stages.

Model Scaling up and down. We design ConvMAE of different parameters scales to match those of MAE-small, MAE-base, MAE-large and MAE-huge. Detailed network architectures are in appendix. The finetuning performances are shown in Table 3. Compared with the original MAE of different scales, our ConvMAE of different scales consistently outperform its MAE counterparts on Imagenet finetuning. This suggests that ConvMAE can be an efficient learner for different paramter scales.

Method	P-Epochs	Model size				
		Small	Base	Base*	Large	Huge
MAE	1600	79.5	83.6	N/A	85.9	86.9
ConvMAE	800	82.6	84.6	84.9	86.2	N/A

Table 3: Ablation study of model scales.

Feature Map Visualization. We provide some visualization of multi-scale feature maps generated by MAE and ConvMAE backbone with the Mask R-CNN method in Fig. 1. The masked convolution reveals much more fine-grained features compared with the pure vision transformer architecture of MAE, especially in feature maps with a stride of 4.

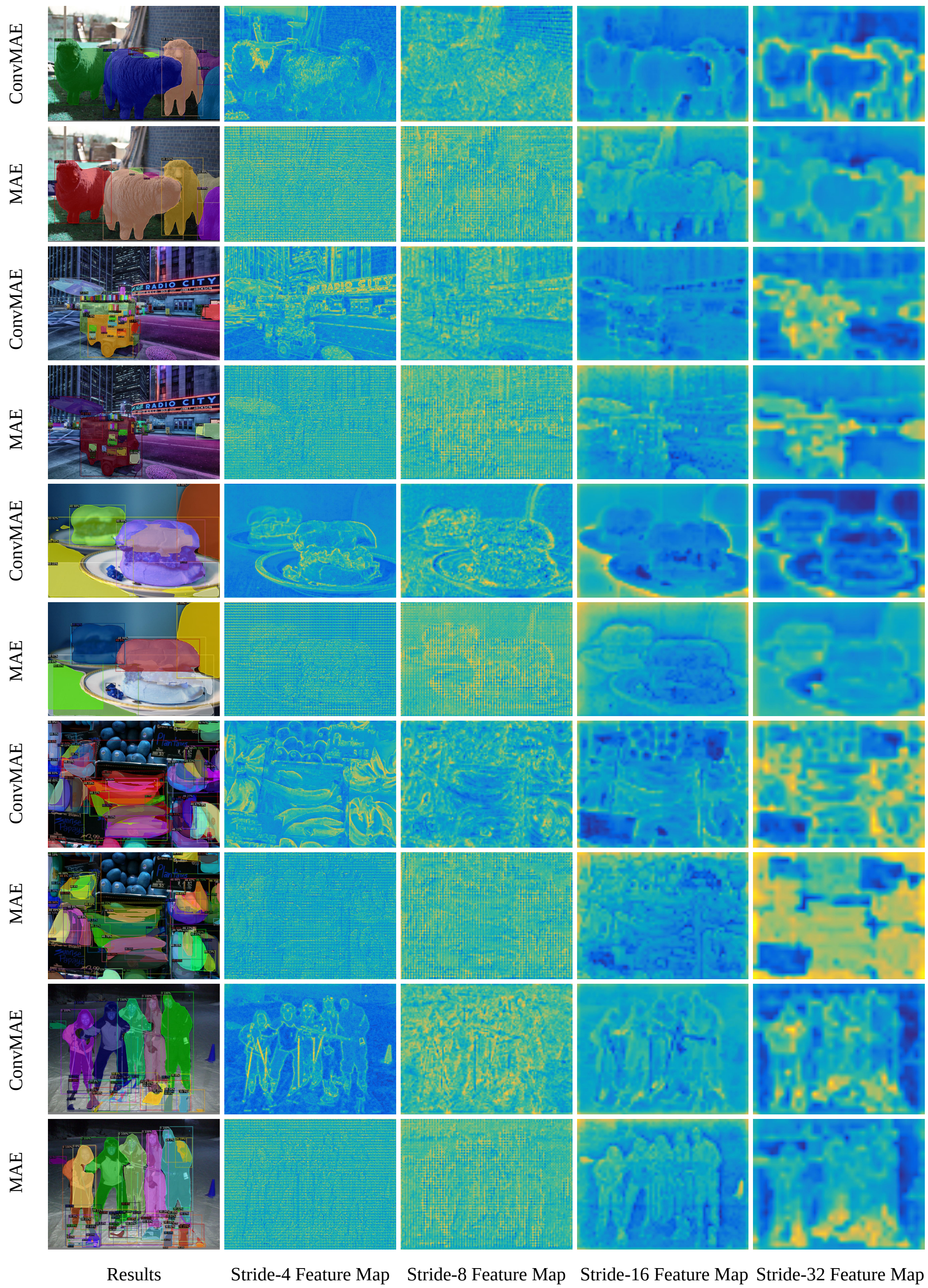

Figure 1: Visualization of feature maps with different strides.