Lite Vision Transformer with Enhanced Self-Attention

Chenglin Yang\(^1\), Yilin Wang\(^2\), Jianming Zhang\(^2\), He Zhang\(^2\), Zijun Wei\(^2\), Zhe Lin\(^2\), Alan Yuille\(^1\)
\(^1\)Johns Hopkins University \(^2\)Adobe Inc.
{chenglin.yangw,alan.l.yuille}@gmail.com \{yilwang,jianmzha,hezhan,zwei,zlin\}@adobe.com

Abstract

Despite the impressive representation capacity of vision transformer models, current light-weight vision transformer models still suffer from inconsistent and incorrect dense predictions at local regions. We suspect that the power of their self-attention mechanism is limited in shallower and thinner networks. We propose Lite Vision Transformer (LVT), a novel light-weight transformer network with two enhanced self-attention mechanisms to improve the model performances for mobile deployment. For the low-level features, we introduce Convolutional Self-Attention (CSA). Unlike previous approaches of merging convolution and self-attention, CSA introduces local self-attention into the convolution within a kernel of size $3 \times 3$ to enrich low-level features in the first stage of LVT. For the high-level features, we propose Recursive Atrous Self-Attention (RASA), which utilizes the multi-scale context when calculating the similarity map and a recursive mechanism to increase the representation capability with marginal extra parameter cost. The superiority of LVT is demonstrated on ImageNet recognition, ADE20K semantic segmentation, and COCO panoptic segmentation. The code is made publicly available\(^1\).

1. Introduction

Transformer-based architectures have achieved remarkable success most recently, they demonstrated superior performances on a variety of vision tasks, including visual recognition\([65]\), object detection\([38,56]\), semantic segmentation\([10,60]\) and etc\([32,54,55]\).

Inspired by the success of the self-attention module in the Natural Language Processing (NLP) community\([53]\), Dosovitskiy\([18]\) first propose a transformer-based network for computer vision, where the key idea is to split the image into patches so that it can be linearly embedded with positional embedding. To reduce the computational complexity introduced by\([18]\), Swin-Transformer\([38]\) upgrades the architecture by limiting the computational cost of self-attention with local non-overlapping windows. Additionally, the hierarchical feature representations are introduced to leverage features from different scales for better representation capability. On the other hand, PVT\([56,57]\) proposes spatial-reduction attention (SRA) to reduce the computational cost. It also removes the positional embedding by inserting depth-wise convolution into the feed forward network (FFN) that follows the self-attention layer. Both Swin-Transformer and PVT have demonstrated their effectiveness for downstream vision tasks. However, when scaling down the model to a mobile friendly size, there is also a significant performance degradation.

In this work, we focus on designing a light yet effective vision transformer for mobile applications\([47]\). More specifically, we introduce a Lite Vision Transformer (LVT) backbone with two novel self-attention layers to pursue both...
the performance and compactness. LVT follows a standard four-stage structure [21, 38, 57] but has similar parameter size to existing mobile networks such as MobileNetV2 [47] and PVTv2-B0 [56].

Our first improvement of self-attention is named Convolutional Self-Attention (CSA). The self-attention layers [1, 24, 44, 58, 66] are the essential components in vision transformer, as self-attention captures both short- and long-range visual dependencies. However, identifying the locality is another significant key to success in vision tasks. For example, the convolution layer is a better layer to process low-level features [16]. Prior arts have been proposed to combine convolution and self-attention with the global receptive field [16, 59]. Instead, we introduce the local self-attention into the convolution within the kernel of size $3 \times 3$. CSA is proposed and used in the first stage of LVT. As a result of CSA, LVT has better generalization ability as it enriches the low-level features over existing transformer models. As shown in Fig 1, compared to PVTv2-B0 [56], LVT is able to generate more coherent labels in local regions.

On the other hand, the performances of lite models are still limited by the parameter number and model depth [60]. We further propose to increase the representation capacity of lite transformers by Recursive Atrous Self-Attention (RASA) layers. As shown in Fig 1, LVT results have better semantic correctness due to such effective representations. Specifically, RASA incorporates two components with weight sharing mechanisms. The first one is Atrous Self-Attention (ASA). It utilizes the multi-scale context with a single kernel when calculating the similarities between the query and key. The second one is the recursion pipeline. Following standard recursive network [19, 28], we formalize RASA as a recursive module with ASA as the activation function. It increases the network depth without introducing additional parameters.

Experiments are performed on ImageNet [46] classification, ADE20K [69] semantic segmentation and COCO [36] panoptic segmentation to evaluate the performance of LVT as a generalized vision model backbone. Our main contributions are summarized in the following:

- We propose Convolutional Self-Attention (CSA). Unlike previous methods of merging global self-attention with convolution, CSA integrates local self-attention into the convolution kernel of size $3 \times 3$. It is proposed to process low-level features by including both dynamic kernels and learnable filters.
- We propose Recursive Atrous Self-Attention (RASA). It comprises two parts. The first part is Atrous Self-Attention (ASA) that captures the multi-scale context in the calculation of similarity map in self-attention. The other part is the recursive formulation with ASA as the activation function. RASA is proposed to increase the representation ability with marginal extra parameter cost.
- We propose Lite Vision Transformer (LVT) as a light-weight transformer backbone for vision models. LVT contains four stages and adopts CSA and RASA in the first and last three stages, respectively. The superior performances of LVT are demonstrated in ImageNet recognition, ADE20K semantic segmentation, and COCO panoptic segmentation.

2. Related Work

Vision Transformer. ViT [18] is the first vision transformer that proves that the NLP transformer [53] architecture can be transferred to the image recognition task with excellent performances. After ViT, a series of improving methods are proposed. For training, DeiT [51] introduces the knowledge distillation strategy for transformer. For tokenization, T2TViT [64] recursively aggregates neighboring tokens into one token to enrich local structure modeling. TNT [20] further splits tokens into smaller ones, extracts their features and integrates them with normal token features. For position embedding, CPVT [13] proposes dynamic position encoding that generalizes to images with arbitrary resolutions. For multi-scale processing, Twins [12] investigates the combination of local and global self-attention. CoaT [62] studies the cross-attention among features at various scales from different stages. CrossViT [3] proposes dual-path transformers that process tokens of different scales. For hierarchical design, Swin-Transformer [38] and PVT [57] both adopt four-stage design and gradually downsample the feature maps which is beneficial to the downstream tasks.

Combining Convolution and Self-Attention. There are four categories of methods. The first one is incorporating the position embedding in self-attention with convolution, including CPVT [13] and CoaT [62]. The second one is applying convolution before self-attention, including CvT [59], CoAtNet [16] and BoTNet [49]. The third one is inserting convolution after self-attention, including LocalViT [31] and PVTv2 [56]. The fourth one is to parallel self-attention and convolution, including AA [2] and LESA [63]. Different from all the above methods that merge local convolutions with global self-attention, we propose Convolutional Self-Attention (CSA) that combines self-attention and convolution both with $3 \times 3$ kernels as a powerful layer in the first stage of the model.

Recursive Convolutional Neural Networks. Recursive methods have been exploited on the convolutional neural networks (CNNs) for various vision tasks. It includes image recognition [33], super resolution [29], object detection [33, 37, 42, 48], semantic segmentation [11, 34]. Unlike these methods, we investigate a recursive method in the light-weight vision transformer as a general model backbone. Specifically, we propose a recursive self-attention
layer with the multi-scale query information, which improves the performance of the mobile model effectively.

Unlike the concurrent light-weight model, MobileFormer [8] which mixes the mobile CNN and transformer architecture, our work, LVT, is only based on transformer architecture with enhanced self-attention layers. Transformer architecture is demonstrated to be more parameter efficient and leads to SOTA performances on downstream tasks [9, 32, 38].

3. Lite Vision Transformer

We propose Lite Vision Transformer (LVT), which is shown in Fig 2. As a backbone network for multiple vision tasks, we follow the standard four-stage design [21, 38, 57]. Each stage performs one downsampling operation and consists of a series of building blocks. Their output resolutions are from stride-4 to stride-32 gradually. Unlike previous vision transformers [18, 38, 57, 65], LVT is proposed with limited amount of parameters and two novel self-attention layers. The first one is the Convolutional Self-Attention layer which has a $3 \times 3$ sliding kernel and is adopted in the first stage. The second one is the Recursive Atrous Self-Attention layer which has a global kernel and is adopted in the last three stages.

3.1. Convolutional Self-Attention (CSA)

The global receptive field benefits the self-attention layer in feature extraction. However, convolution is preferred in the early stages of vision models [16] as the locality is more important in processing low-level features. Unlike previous methods of combining convolution and large kernel (global) self-attention [16, 59], we focus on designing a window-based self-attention layer that has a $3 \times 3$ kernel and incorporates the representation of convolution.

Figure 2. Lite Vision Transformer (LVT). The top row represents the overall structure of LVT. The bottom left and right parts visualize the proposed Convolutional Self-Attention (CSA) and Recursive Atrous Self-Attention (RASA). $H, W$ represents the height and width of the image. $C$ is the feature channel. The output resolution of each module is shown. Both the Unfold and Fold operations has a stride of 2. BMM stands for batched matrix multiplication, which corresponds to $W_{i \rightarrow j}x_j$ in Eqn. (1) with the batch dimension being the number of spatial locations in a local window. ASA stands for the proposed Atrous Self-Attention.

Figure 3. Illustration of Convolutional Self-Attention (CSA) in a $3 \times 3$ local window. The outputs of convolution and CSA are $1 \times 1$ and $3 \times 3$, respectively. Mathematically, convolution comprises two procedures: the batched matrix multiplication (BMM) and summation. BMM corresponds to $W_{i \rightarrow j}x_j$ in Eqn. (1) with the batch dimension being the number of spatial locations. CSA has the BMM operation, but has the same summation process as SA. It performs 9 different input-dependent summations with the weights $\alpha$ in Eqn. (2), which process is shown by the colored arrows and patches. Through this design, CSA contains both the learnable filter and dynamic kernel.
Analyzing Convolution. Let $x, y \in \mathbb{R}^d$ be the input and output feature vectors where $d$ represents the channel number. Let $i, j \in \mathbb{R}$ index the spatial locations. Convolution is computed by sliding windows. In each window, we can write the formula of convolution as:

$$y_i = \sum_{j \in N(i)} W_{i \rightarrow j} x_j$$  \hspace{1cm} (1)

where $N(i)$ represents the spatial locations in this local neighborhood that is defined by the kernel centered at location $i$. $|N(i)| = k \times k$ where $k$ is the kernel size. $i \rightarrow j$ represents the relative spatial relationship from $i$ to $j$. $W_{i \rightarrow j} \in \mathbb{R}^{d \times d}$ is the projection matrix. In total, there are $|N(i)|$ $W$s in a kernel. A $3 \times 3$ kernel consists of 9 such matrices $W$s.

Analyzing Self-Attention. Self-Attention needs three projection matrices $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ to compute query, key and value. In this paper, we consider sliding window based self-attention. In each window, we can write the formula of self-attention as

$$y_i = \sum_{j \in N(i)} \alpha_{i \rightarrow j} W_v x_j$$

$$\alpha_{i \rightarrow j} = \frac{e^{(W_q x_i)^T W_k x_j}}{\sum_{z \in N(i)} e^{(W_q x_i)^T W_k x_z}}$$ \hspace{1cm} (2)

where $\alpha_{i \rightarrow j} \in (0, 1)$ is a scalar that controls the contribution of the value in each spatial location in the summation. $\alpha$ is normalized by softmax operation such that $\sum_j \alpha_{i \rightarrow j} = 1$. Compared with convolution with the same kernel size $k$, the number of learnable matrices is three rather than $k^2$. Recently, Outlook Attention [65] is proposed to predict $\alpha$ instead of calculating it by the dot product of query and key, and shows superior performances when the kernel size is small. We employ this calculation, and it can be written as:

$$\alpha_{i \rightarrow j} = \frac{W_{qk} x_i [j]}{\sum_{z \in N(i)} W_{qk} x_i [z]}$$ \hspace{1cm} (3)

where $W_{qk} \in \mathbb{R}^{d \times k^2}$ and $[j]$ means $j$th element of the vector.

Convoluted Self-Attention (CSA). We generalize self-attention and convolution into a unified convolutional self-attention as shown in Fig. 3. Its formulation is shown in the following:

$$y_i = \sum_{j \in N(i)} \alpha_{i \rightarrow j} W_{i \rightarrow j} x_j$$ \hspace{1cm} (4)

Both SA and CSA have the output of size $k \times k$ for a local window. When $\alpha_{i \rightarrow j} = 1$ where all the weights are the same, CSA is the convolution for the output center. When $W_{i \rightarrow j} = W_v$ where all the projection matrices are the same, CSA is self-attention. As we employ the dynamic $\alpha$ predicted by the input, as shown in Eqn. (3), Outlook Attention [65] is a special case of CSA. CSA has a bigger capacity than Outlook Attention. We summarize its property in Tab. 1. By this generalization, CSA has both input-dependent kernel and learnable filter. It is designed for stronger representation capability in first stage of vision transformers.

### 3.2. Recurrent Atrous Self-Attention (RASA)

Light-weight models are more efficient and more suitable for on-device applications [47]. However, their performances are limited by the small number of parameters even with advanced model architecture [60]. For light-weight models, we focus on enhancing their representation capabilities with marginal increase in the number of parameters. Atrous Self-Attention (ASA). Multi-scale features are beneficial in detecting or segmenting the objects [35, 67]. Atrous convolution [5, 6, 41] is proposed to capture the multi-scale context with the same amount of parameters as standard convolution. Weights sharing atrous convolution [42] is also demonstrated in boosting model performances. Unlike convolution, the feature response of self-attention is a weighted sum of the projected input vectors from all spatial locations. These weights are determined by

---

**Table 1. Convolutional Self-Attention: Generalization of convolution and self-attention.**

<table>
<thead>
<tr>
<th>Operations</th>
<th>Input dependent</th>
<th>Learnable filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Self-Attention</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Conv. Self-Attention</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
the similarities between the queries and keys, and represent the strength of the relationship among any pair of feature vectors. Thus we add multi-scale information when generating these weights shown in Fig 4. Specifically, we upgrade the calculation of the query from a $1 \times 1$ convolution to the following operation:

$$Q = \sum_{r \in \{1,3,5\}} \text{SiLU}(\text{Conv}(\hat{Q}, W_{q}^{k=3}, r, g = d))$$

where

$$\hat{Q} = \text{Conv}(X, W_{q}^{k=1}, r = 1, g = 1),$$

$$\text{SiLU}(m) = m \odot \text{sigmoid}(m),$$

$X, Q \in \mathbb{R}^{d \times H \times W}$ are the feature maps, and $W_{q}^{k} \in \mathbb{R}^{k^{2} \times d \times d/g}$ is the kernel weight. $H, W$ are the spatial dimensions. $d$ is the feature channels. $k, r$ and $g$ represent the kernel size, dilation rate, and group number of the convolution. We first use the $1 \times 1$ convolution to apply linear projection. Then we apply three convolutions that have different dilation rates but a shared kernel to capture the multi-scale contexts. The parameter cost is further reduced by setting the group number equal to the feature channel number. The parallel features of different scales are then weighted summed. We employ a self-calibration mechanism that determines the weights for each scale by their activation strength. This can be implemented by the SiLU [22, 45]. By this design, the similarity calculation of the query and key between any pair of spatial locations in self-attention uses the multi-scale information.

**Recursive Atrous Self-Attention (RASA).** For lightweight models, we intend to increase their depths without increasing the parameter usage. Recursive methods have been proposed in many vision tasks for Convolutional Neural Networks (CNNs) including [29, 33, 42, 50]. Unlike these methods, we propose a recursive method for self-attention. The design follows the pipeline of the standard recurrent networks [19, 28]. Together with atrous self-attention (ASA), we propose recursive atrous self-attention (RASA), and its formula can be written in the following:

$$x_{t+1} = \text{ASA}(\mathbf{F}(X_{t}, h_{t-1}))$$

$$h_{t-1} = X_{t-1}$$

$$X_{t} = \text{ASA}(\mathbf{F}(X_{t-1}, h_{t-2}))$$

where $t$ is the step and $h \in \mathbb{R}^{d \times H \times W}$ the hidden state. We take ASA as the non-linear activation function. The initial hidden state $h_{-1} = \mathbf{0}$. $\mathbf{F}(X, h) = W_{F}X + U_{F}h$ is the linear function combining the input and hidden state. $W_{F}, U_{F}$ are the projection weights. However, we empirically find that setting $W_{F} = 1, U_{F} = 1$ provides the best performances and avoids introducing extra parameters. We set the recursion depth as two in order to limit the computation cost.
Table 5. Mobile COCO panoptic segmentation. The FPS is calculated on the 2,000 high-resolution images. They are rescaled such that the maximum length does not exceed 1333 and the minimum length 800. The aspect ratio is kept. It is observed that LVT achieves significant performance improvement over previous SOTA mobile encoders for panoptic segmentation.

3.3. Model Architecture

The architecture of LVT is shown in Tab. 2. We adopt the standard four-stage design [21]. Four Overlapped Patch Embedding layers [60] are employed. The first one downsamples the feature maps to the resolution of stride-8, stride-16, and stride-32. All stages comprise the transformer blocks [18]. Each block contains the self-attention layer followed by an MLP layer. CSA is embedded in the stage-1 while RASA in the other stages. They are enhanced self-attention layers proposed to process local and global features in LVT.

4. Experiments

4.1. ImageNet Classification

Dataset. We perform image recognition experiments on ILSVRC2012 [46], a popular subset of the ImageNet database [17]. The training and validation sets contain 1.3M and 50K images, respectively. There are 1,000 object categories in total. The classes are distributed approximately and strictly uniformly in the training and validation sets.

Settings. The training setting follows previous conventions. We use AdamW as the optimizer [40]. Following previous works [27,51], the learning rate is scaled based on the batch size with the formula being \( \text{lr} = \frac{\text{batch size}}{256} \times \text{lr base} \). We set \( \text{lr base} = 1.6 \times 10^{-3} \). The weight decay of \( 5 \times 10^{-2} \) is adopted. Stochastic depth with drop path rate being 0.1 is employed [26]. In total, there are 300 training epochs. Following [65], we use CutOut [68], RandAug [15], and Token Labeling [27] as the data augmentation methods. Class attention layer [52] is used as the post stage. Both in the training and testing phase, the input resolution is 224 × 224.

Results. The results are shown in Tab. 3. We limit the encoder size less than 3.5M, following MobileNet [47] and PVTv2-B0 [56]. The encoder is our design focus as it is the backbone used by other complex tasks like detection and segmentation. In order to compare LVT with other standard models, we scale LVT to the size of ResNet50 [21], a canonical backbone of vision models. The high performances of LVT for image recognition is demonstrated.

4.2. Mobile ADE20K Semantic Segmentation

Dataset. We perform semantic segmentation task on the challenging ADE20K dataset [69]. There are 150 categories in total, including 35 stuff classes and 115 discrete objects. The training and validation sets contain 20,210 and 2,000 images, respectively.

Settings. Previous conventions are followed. We adopt the Segformer framework [60] and use the MLP decoder. The LVT encoder is pretrained on ImageNet-1K without extra data. The decoder is trained from scratch. We use mmsegmentation [14] as the codebase. The AdamW optimizer [40] with the initial learning rate being 6 × 10^{-5} is used. The weight decay is set as 1 × 10^{-2}. The poly learning rate schedule with power being 1 is employed. There are 160K training iterations in total and the batch size is 16. For data augmentation, we randomly resize the image with ratio 0.5 − 2.0 and then perform random cropping of size 512 × 512. Horizontal flipping with probability 0.5 is applied. During evaluation, we perform single-scale test.

Results. The results are summarized in Tab. 4. The FLOPs
Table 6. Ablation studies. VOLO is used as the base network to add CSA and RASA, because VOLO uses the self-attention with $3 \times 3$ kernel in the first stage. By this comparison, the performance gain from local self-attention to Convolutional Self-Attention (CSA) can be clearly illustrated. It is demonstrated that both CSA and RASA significantly contribute to the performance improvement.

is calculated with the input resolution $512 \times 512$. The FPS is calculated on 2000 images on a single NVIDIA V100 GPU. During inference, the images are resized such that the short side is 512. We only use single-scale testing. The model is compact. Together with the decoder, the parameters are less than 4M. We can observe that LVT demonstrates the best performance among all previous mobile methods for semantic segmentation.

4.3. Mobile COCO Panoptic Segmentation

Dataset. We perform panoptic segmentation on COCO dataset. The 2017 split is employed. It has 118K training images and 5K validation images. On average, each image contains 3.5 categories and 7.7 instances. We choose panoptic segmentation to evaluate our methods as it unifies the object recognition, detection, localization, and segmentation at the same time. We aim to thoroughly evaluate the performance of our model.

Settings. The panoptic FPN [30] framework is adopted. All the models are trained in this framework for fair comparisons. We use mmdetection [4] as the codebase. AdamW [40] optimizer with the initial learning rate $3 \times 10^{-4}$ is used. The weight decay is $1 \times 10^{-4}$. The $3 \times$ schedule is employed. There are 36 training epochs in total, the learning rate is decayed by 10 times after 24 and 33 epochs. We adopt multi-scale training. During training, the images are randomly resized. The maximum length does not exceed 1333. The maximum allowable length of the short side is randomly sampled in the range of 640−800. Random horizontal flipping with probability 0.5 is applied. During testing, we perform single-scale testing.

Results. The results are shown in Tab. 5. The FLOPs is calculated on the input resolution $1200 \times 800$. During the inference, all the images are resized such that the large side is not larger than 1333 and the short side is less than 800. The FLOPs are calculated on 2000 high-resolution images with a single NVIDIA V100 GPU. The whole model including the decoder takes less than 5.5M parameters. We can observe the superiority of LVT compared with the previous SOTA encoders for mobile panoptic segmentation.

5. Ablation Studies

5.1. Recursion Times of RASA

In this section, we investigate the relationship between the recursion times and model performances. The experiments are performed on ImageNet classification. We set the recursion time from 1 to 4. The results are summarised in Tab. 7.
Table 7. Relationship of recursion times and performances on ImageNet Classification. \( R \) means recursion times. The performance increases dramatically with two iterations. Considering the efficiency, we use LVT\(_R2\) in the main experiments.

<table>
<thead>
<tr>
<th>Models</th>
<th>Top-1 (%)</th>
<th>Top-5 (%)</th>
<th>Params (M)</th>
<th>FLOPs (G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVT(_R1)</td>
<td>73.9</td>
<td>92.1</td>
<td>3.4/5.5</td>
<td>0.8</td>
</tr>
<tr>
<td>LVT(_R2)</td>
<td>74.8</td>
<td>92.6</td>
<td>3.4/5.5</td>
<td>0.9</td>
</tr>
<tr>
<td>LVT(_R3)</td>
<td>74.6</td>
<td>92.5</td>
<td>3.4/5.5</td>
<td>1.0</td>
</tr>
<tr>
<td>LVT(_R4)</td>
<td>74.9</td>
<td>92.6</td>
<td>3.4/5.5</td>
<td>1.1</td>
</tr>
</tbody>
</table>

5.2. Contributions of CSA and RASA

Settings. In this section, we study the performance contributions of Convolutional Self-Attention (CSA) and Recursive Atrous Self-Attention. To this end, we build our model via the recently proposed VOLO which employed small kernel self-attention in the first stage. As VOLO is demonstrated as a powerful backbone in image recognition and semantic segmentation, we perform experiments on ImageNet and ADE20K. In order to perform the comparisons in the mobile setting, we scale VOLO to have a parameter size 4.0M. Specifically, we set the layer number of each stage to be 2, and adjust the feature dimensions to be 96, 192, 192, 192. All the other settings are kept unchanged.

Results. The results are reported in Tab. 6. For ImageNet classification, the input resolution in both the training and testing is 224 × 224. For ADE20K semantic segmentation, we insert the VOLO and LVT with the MLP decoder, following the Segformer framework [60]. During testing, the short side of the image is resized to 512. It is observed both CSA and RASA significantly contribute to the performance gain.

6. Conclusion

In this work, we propose a powerful light-weight transformer backbone, Lite Vision Transformer (LVT). It consists of two novel self-attention layers: Convolutional Self-Attention (CSA) and Recursive Atrous Self-Attention (RASA). They are used in the first and the last three stages of LVT to process low and high level features. We demonstrate the strong performances of LVT compared with previous mobile methods in tasks of visual recognition, semantic segmentation, and panoptic segmentation.

Limitations. LVT is a light-weight model. The natural limitation is the weak representation power compared with models with large number of parameters. The focus of this work is on the mobile models. Our future work includes scaling LVT to the large powerful backbones.

Acknowledgements. The authors gratefully acknowledge supports from ONR N00014-21-1-2812.
References


Wenhui Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao.


