# LocalViT: Bringing Locality to Vision Transformers

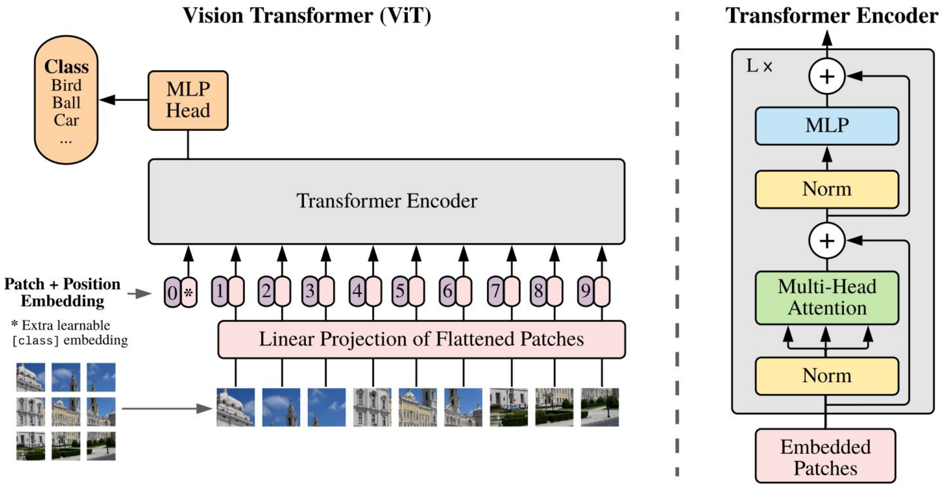
Yawei Li[1] Kai Zhang[1] Jiezhang Cao[1] Radu Timofte[1] Luc Van Gool[1,2]
[1]Computer Vision Lab, ETH Zurich, Switzerland [2]KU Leuven, Belgium

Presenter: Junming CHEN

arXiv:2104.05707v1 [cs.CV] 12 Apr 2021

# Problem in Original ViT

- Lacking a locality mechanism for information exchange within a local region.
    a. Locality is essential for images since it pertains to structures like lines, edges, shapes, and even objects.



# Idea

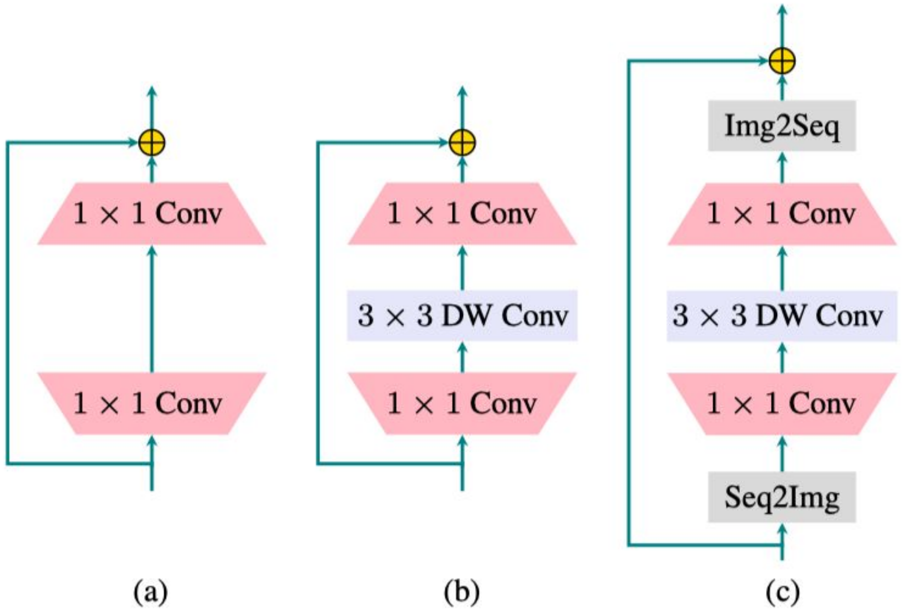
Insert a depthwise convolution between two MLPs in Transformer encoder.

a. Original MLPs in ViT
b. Inverted residual blocks
c. Proposed method

$$\mathbf{Z}^r = \mathrm{Seq2Img}(\mathbf{Z}), \mathbf{Z}^r \in \mathbb{R}^{h \times w \times d},$$
$$\text{where } h = H/p \text{ and } w = W/p.$$

$$\mathbf{Y}^r = f\big(f(\mathbf{Z}^r \circledast \mathbf{W}_1^r) \circledast \mathbf{W}_d\big) \circledast \mathbf{W}_2^r,$$

$$\mathbf{Y} = \mathrm{Img2Seq}(\mathbf{Y}^r),$$



# Idea

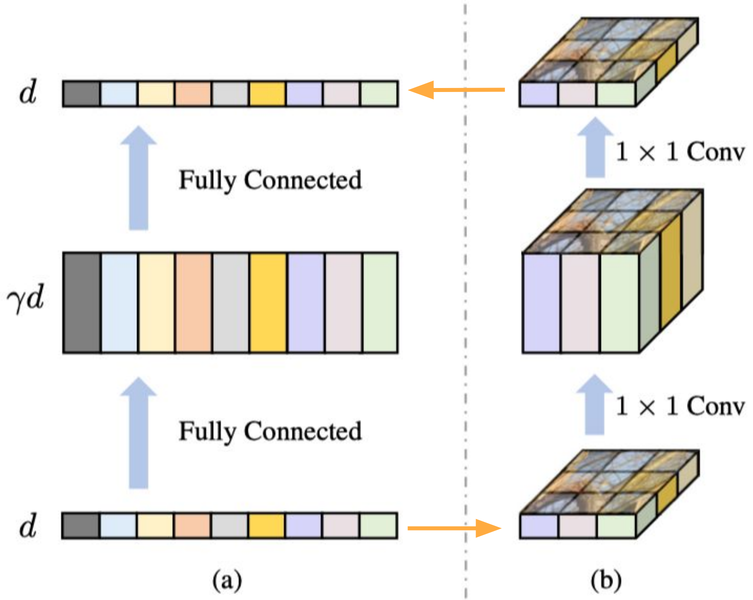
Before apply the convolution: Rearrange

$$\mathbf{Z}^r = \text{Seq2Img}(\mathbf{Z}), \mathbf{Z}^r \in \mathbb{R}^{h \times w \times d},$$

where $h = H/p$ and $w = W/p$.

$$\mathbf{Y}^r = f\big(f(\mathbf{Z}^r \circledast \mathbf{W}_1^r) \circledast \mathbf{W}_d\big) \circledast \mathbf{W}_2^r,$$

$$\mathbf{Y} = \text{Img2Seq}(\mathbf{Y}^r),$$



# Idea

What about classification token? Bypass: split and concatenation

$$(\mathbf{Z}_{cls}, \mathbf{Z}) \leftarrow \mathrm{Split}(\mathbf{Z}).$$

$$\mathbf{Z}^r = \mathrm{Seq2Img}(\mathbf{Z}), \mathbf{Z}^r \in \mathbb{R}^{h \times w \times d},$$

where $h = H/p$ and $w = W/p$.

$$\mathbf{Y}^r = f\big(f(\mathbf{Z}^r \circledast \mathbf{W}_1^r) \circledast \mathbf{W}_d\big) \circledast \mathbf{W}_2^r,$$

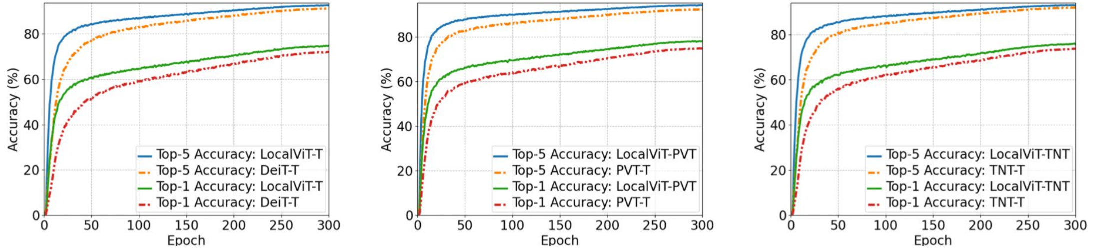
$$\mathbf{Y} = \mathrm{Img2Seq}(\mathbf{Y}^r),$$

$$\mathbf{Y} \leftarrow \mathrm{Concat}(\mathbf{Z}_{cls}, \mathbf{Y}).$$

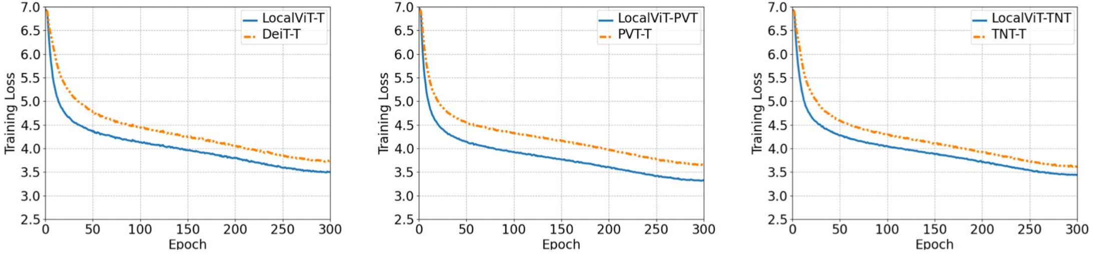
# Results

## Classification on ImageNet 2012:

| Network | $\gamma$ | DW | Params (M) | FLOPs (G) | Top-1 Acc. (%) |
|---|---|---|---|---|---|
| DeiT-T [41] | 4 | No | 5.7 | 1.3 | 72.2 |
| LocalViT-T | 4 | No | 5.7 | 1.3 | 72.5 (0.3↑) |
| LocalViT-T* | 4 | Yes | 5.8 | 1.3 | 73.7 (1.5↑) |
| DeiT-T [41] | 6 | No | 7.5 | 1.6 | 73.1† |
| LocalViT-T | 6 | No | 7.5 | 1.6 | 74.3 (1.2↑) |
| LocalViT-T* | 6 | Yes | 7.7 | 1.6 | 76.1 (3.0↑) |




(a) DeiT-T *vs.* LocalViT-T. Accuracy. (b) PVT-T *vs.* LocalViT-PVT. Accuracy. (c) TNT-T *vs.* LocalViT-TNT. Accuracy.




(d) DeiT-T *vs.* LocalViT-T. Training loss. (e) PVT-T *vs.* LocalViT-PVT. Training loss. (f) TNT-T *vs.* LocalViT-TNT. Training loss.

# Results

Activation functions.

| Activation | Params (M) | FLOPs (G) | Top-1 Acc. (%) |
|---|---|---|---|
| Deit-T [41] | 5.7 | 1.3 | 72.2 |
| ReLU6 | 5.8 | 1.3 | 73.7 (1.5↑) |
| h-swish | 5.8 | 1.3 | 74.4 (2.2↑) |
| h-swish + ECA | 5.8 | 1.3 | 74.5 (2.3↑) |
| h-swish + SE-192 | 5.9 | 1.3 | 74.8 (2.6↑) |
| h-swish + SE-96 | 6.0 | 1.3 | 74.8 (2.6↑) |
| h-swish + SE-48 | 6.1 | 1.3 | 75.0 (2.8↑) |
| h-swish + SE-4 | 9.4 | 1.3 | 75.8 (3.6↑) |

# Results

- Placement of locality.
  - Locality is important in lower layers.

| DW Placement | Layer | Params (M) | FLOPs (G) | Top-1 Acc. (%) |
|---|---|---|---|---|
| High | 9∼12 | 5.78 | 1.26 | 69.1 |
| Mid | 5∼8 | 5.78 | 1.26 | 72.1 |
| Low | 1∼4 | 5.78 | 1.26 | 73.1 |
| Low | 1∼8 | 5.84 | 1.27 | 74.0 |
| All | 1∼12 | 5.91 | 1.28 | 74.8 |

# Takeaway

- Global and local interaction are both significant.
- Convolution can improve the performance of the baseline transformer.
- A better activation function after convolution can result in a significant performance gain.
- Locality is more important for lower layers.
- Expanding the hidden dimension of the feed-forward network leads to a larger model capacity and a higher classification accuracy.

arXiv:2103.13413v1 [cs.CV] 24 Mar 2021

# Vision Transformers for Dense Prediction

René Ranftl Alexey Bochkovskiy Vladlen Koltun

Intel Labs

rene.ranftl@intel.com

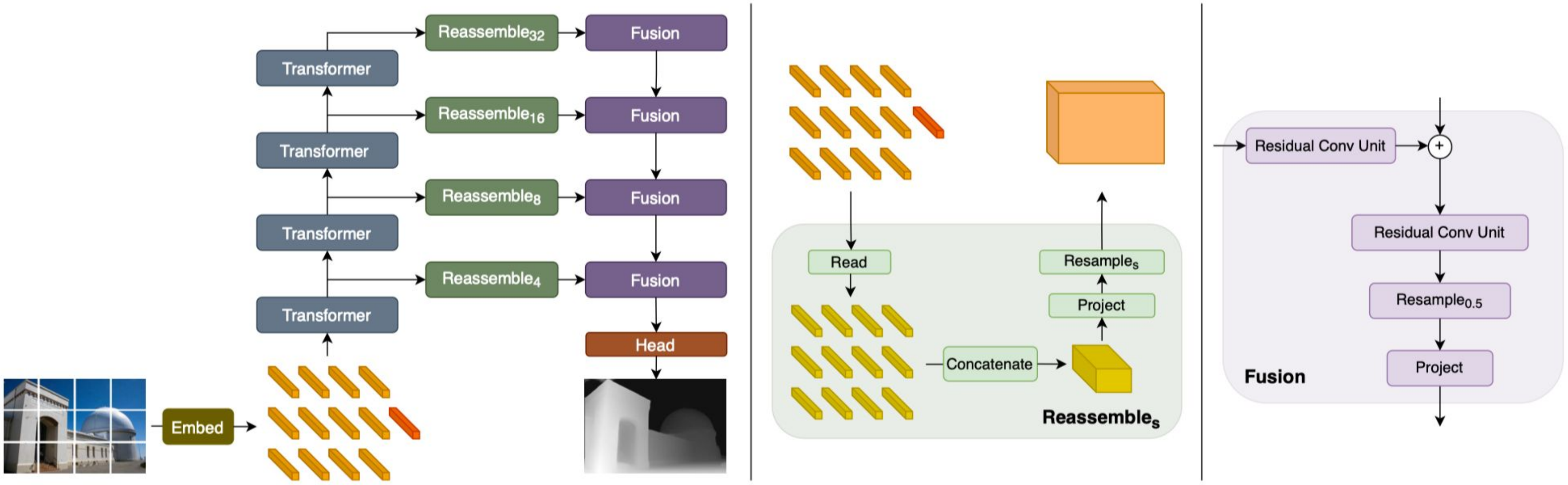
Presenter: Junming CHEN

# Motivation

- Shortcomes of convolutional encoder
  - Convolutional backbones progressively downsample the input image to extract features at multiple scales.
  - Feature resolution and granularity are lost in the deeper stages of the model and can thus be hard to recover in the decoder.
  - However, feature resolution and granularity are critical for dense prediction.
- Why use Transformer for dense prediction?
  - It processes representations at a constant and relatively high resolution. Therefore it should have higher feature resolution and granularity.
  - It has a global receptive field at every stage.

# Method - Overall

- Transformer Encoder + Convolutional Decoder



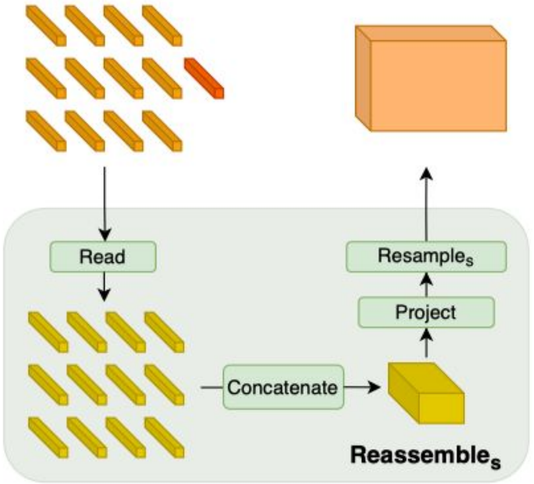
# Method - Reassemble

- Read
  - Ignore $\text{Read}_{ignore}(t) = \{t_1, \ldots, t_{N_p}\}$
  - Add $\text{Read}_{add}(t) = \{t_1 + t_0, \ldots, t_{N_p} + t_0\}$
  - Projection $\text{Read}_{proj}(t) = \{\text{mlp}(\text{cat}(t_1, t_0)), \ldots, \text{mlp}(\text{cat}(t_{N_p}, t_0))\}$
- Concatenate
  - Sequence to lattice. (Patch resolution: p x p)

$$\text{Concatenate} : \mathbb{R}^{N_p \times D} \rightarrow \mathbb{R}^{\frac{H}{p} \times \frac{W}{p} \times D}.$$

- Resample
  - s denotes the output size ratio

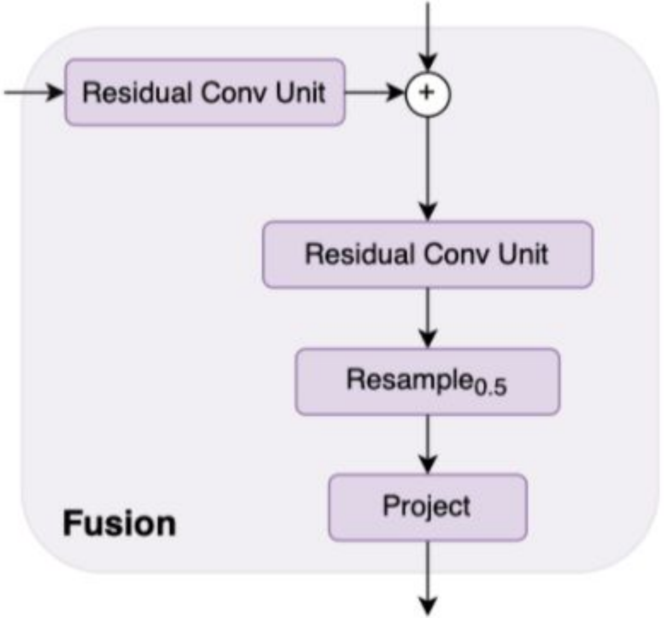
$$\text{Resample}_s : \mathbb{R}^{\frac{H}{p} \times \frac{W}{p} \times D} \rightarrow \mathbb{R}^{\frac{H}{s} \times \frac{W}{s} \times \hat{D}}.$$




$$\text{Reassemble}_s^{\hat{D}}(t) = (\text{Resample}_s \circ \text{Concatenate} \circ \text{Read})(t),$$

# Method - Fusion

- RefineNet-based feature fusion
  - progressively upsample the representation by a factor of two in each fusion stage.




What about positional encoding of various resolution? Interpolation.

# Results - Monocular depth estimation

Set new SOTA.

| | Training set | DIW WHDR | | ETH3D AbsRel | | Sintel AbsRel | | KITTI $\delta$>1.25 | | NYU $\delta$>1.25 | | TUM $\delta$>1.25 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DPT - Large | MIX 6 | **10.82** | (-13.2%) | **0.089** | (-31.2%) | **0.270** | (-17.5%) | **8.46** | (-64.6%) | **8.32** | (-12.9%) | **9.97** | (-30.3%) |
| DPT - Hybrid | MIX 6 | 11.06 | (-11.2%) | 0.093 | (-27.6%) | 0.274 | (-16.2%) | 11.56 | (-51.6%) | 8.69 | (-9.0%) | 10.89 | (-23.2%) |
| MiDaS | MIX 6 | 12.95 | (+3.9%) | 0.116 | (-10.5%) | 0.329 | (+0.5%) | 16.08 | (-32.7%) | 8.71 | (-8.8%) | 12.51 | (-12.5%) |
| MiDaS [30] | MIX 5 | 12.46 | | 0.129 | | 0.327 | | 23.90 | | 9.55 | | 14.29 | |
| Li [22] | MD [22] | 23.15 | | 0.181 | | 0.385 | | 36.29 | | 27.52 | | 29.54 | |
| Li [21] | MC [21] | 26.52 | | 0.183 | | 0.405 | | 47.94 | | 18.57 | | 17.71 | |
| Wang [40] | WS [40] | 19.09 | | 0.205 | | 0.390 | | 31.92 | | 29.57 | | 20.18 | |
| Xian [45] | RW [45] | 14.59 | | 0.186 | | 0.422 | | 34.08 | | 27.00 | | 25.02 | |
| Casser [5] | CS [8] | 32.80 | | 0.235 | | 0.422 | | 21.15 | | 39.58 | | 37.18 | |

Table 1. Comparison to the state of the art on monocular depth estimation. We evaluate zero-shot cross-dataset transfer according to the protocol defined in [30]. Relative performance is computed with respect to the original MiDaS model [30]. Lower is better for all metrics.

# Results - Segmentation

Set new SOTA.

| | Backbone | | pixAcc [%] | mIoU [%] |
|---|---|---|---|---|
| OCNet | ResNet101 | [50] | – | 45.45 |
| ACNet | ResNet101 | [14] | 81.96 | 45.90 |
| DeeplabV3 | ResNeSt-101 | [7, 51] | 82.07 | 46.91 |
| DeeplabV3 | ResNeSt-200 | [7, 51] | 82.45 | 48.36 |
| DPT-Hybrid | ViT-Hybrid | | **83.11** | **49.02** |
| DPT-Large | ViT-Large | | 82.70 | 47.63 |

Table 4. Semantic segmentation results on the ADE20K validation set.

# Results - Ablation

Choice of *Read* operation.

| | HRWSI | BlendedMVS | ReDWeb | Mean |
|---|---|---|---|---|
| Ignore | **0.0793** | 0.0780 | **0.0892** | 0.0822 |
| Add | 0.0799 | 0.0789 | 0.0904 | 0.0831 |
| Project | 0.0797 | **0.0764** | 0.0895 | **0.0819** |

Table 7. Performance of approaches to handle the readout token. Fusing the readout token to the individual input tokens using a projection layer yields the best performance.

# Results - Inference resolution

- Conjecture: global receptive field in every layer makes DPT less dependent on inference resolution. (Training resolution: 384×384 pixels)

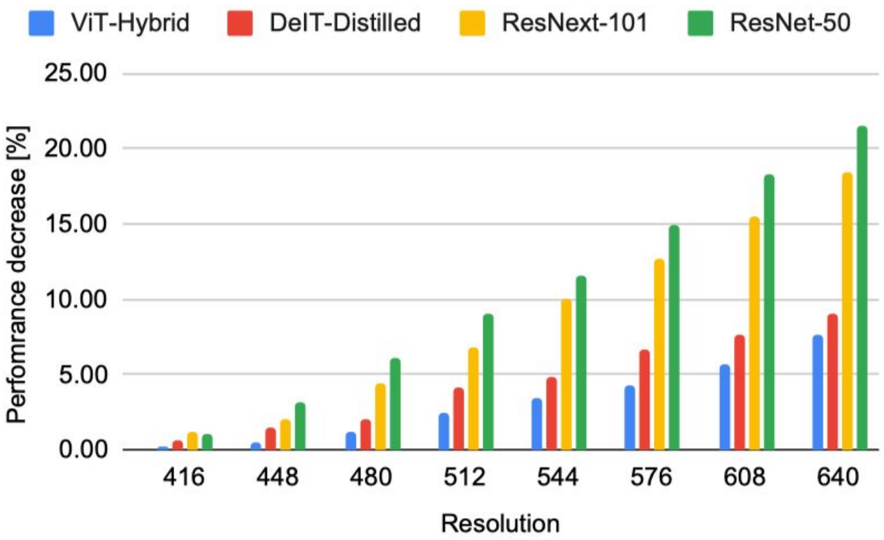



Figure 4. Relative loss in performance for different inference resolutions (lower is better).