Vision Transformers for Dense Prediction

공학 딥러닝 논문 리뷰

Intel Labs, René Ranftl, Alexey Bochkovskiy, Vladlen Koltun

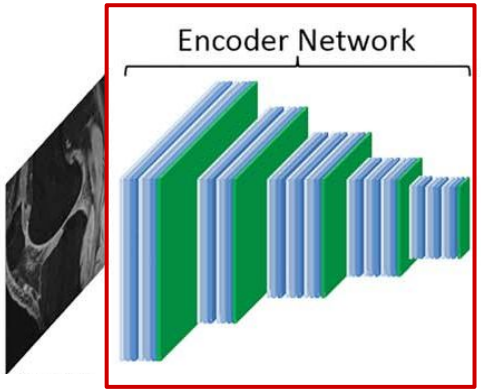
ICCV 2021

발표자: 남유진

울산대학교 UNIVERSITY OF ULSAN
서울아산병원 Asan Medical Center

Introduction

Virtually all existing architectures for **dense prediction** are based on convolutional networks.



→ Enables a progressive increase of the receptive field,
→ grouping of low-level features into abstract high-level features

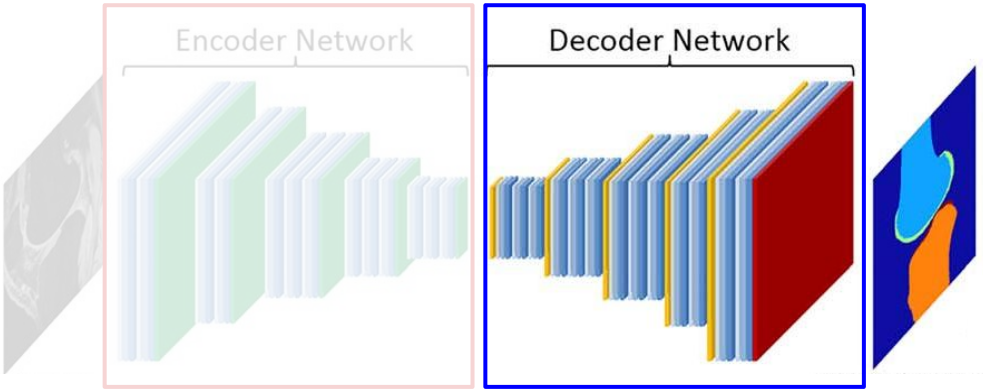
But, **Feature resolution and granularity are lost** in the **deeper stages** of the model and can thus **be hard to recover** in the decoder.

* Called the backbone (frequently based on an image classification network)
* Downsample the input image to extract features at multiple scales.

While it may not matter for some tasks, such as image classification, they are critical for **dense prediction**

Introduction

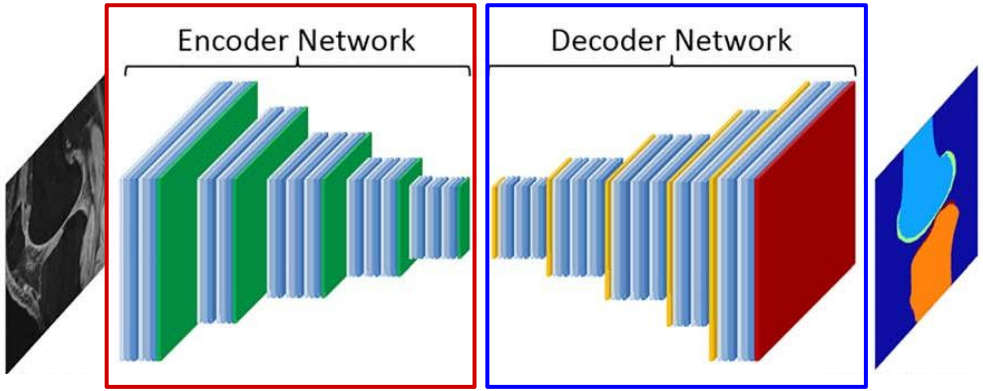
Virtually all existing architectures for **dense prediction** are based on convolutional networks.



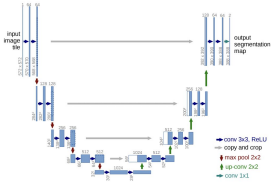
* Aggregate encoded features and convert them to dense prediction

Introduction

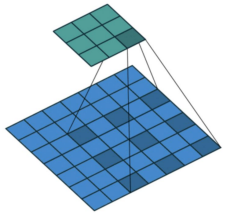
Virtually all existing architectures for **dense prediction** are based on convolutional networks.



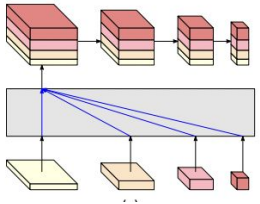
Training at higher input resolution



Unet
Ronneberger et al., 2015



Dilated convolutaion
Yu and Koltun, 2016z



HRNet
Wang et al., 2020

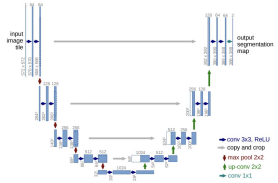
Introduction

Virtually all existing architectures for **dense prediction** are based on convolutional networks.

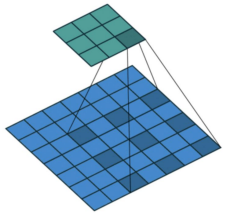
- The networks are still bottlenecked by their fundamental building block: the convolution
- Convolution: limited receptive field > stacking into very deep architecture
- Downsampling is necessary due to keep memory consumption

→ Use **Transformer** for Encoder

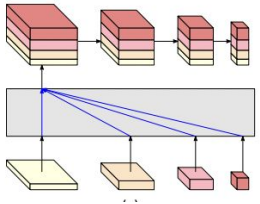
Training at higher input resolution



Unet
Ronneberger et al., 2015



Dilated convolutaion
Yu and Koltun, 2016

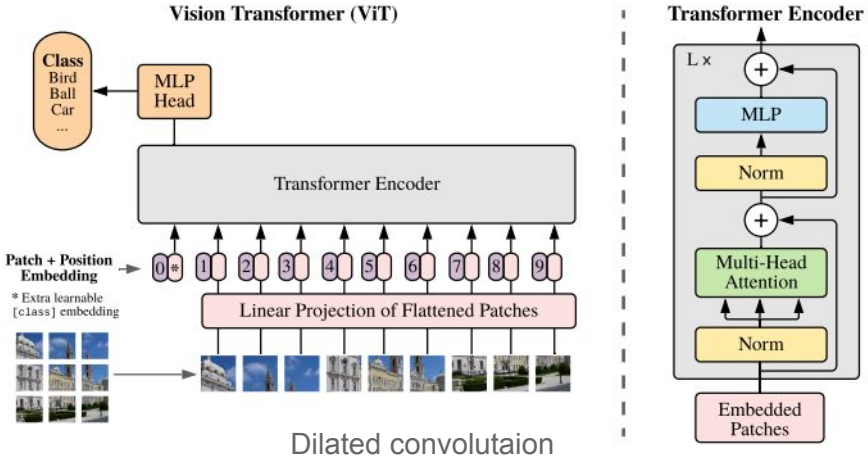


HRNet
Wang et al., 2020

Introduction

DPT is **a dense prediction architecture** that is based on an **encoder-decoder** design that leverages a transformer as the basic computational building block of the encoder.

Encoder: Vision Transformer (Vit) into image-like feature representations at various resolutions, used as a backbone.
Decoder: combine the feature representations into the **final dense prediction** using a convolutional decoder



Dilated convolutaion
Dosovitskiy et al., 2021

→ The Vit backbone **foregoes explicit downsampling operations** after embedding and maintains a representation with constant dimensionality throughout all processing stages.

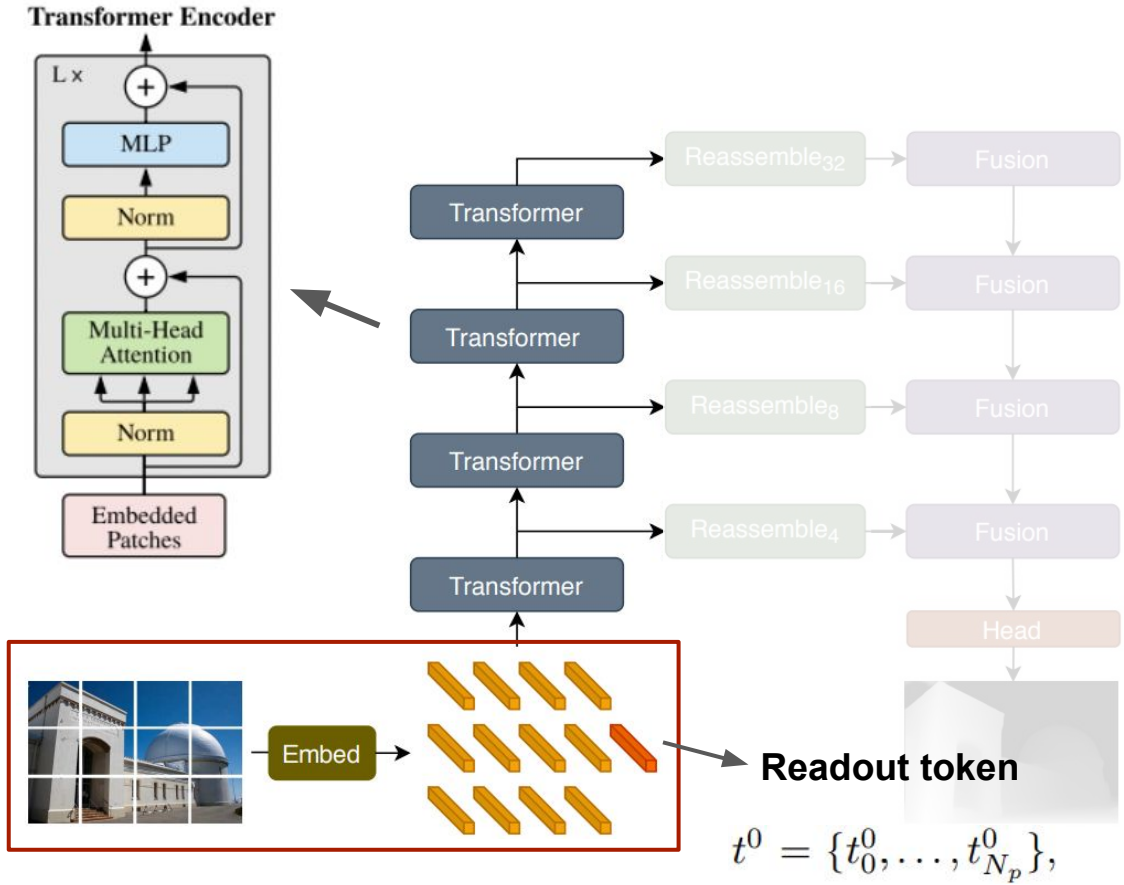
→ It furthermore has a **global receptive field** at every stage: **lead to fine-grained** and globally coherent predictions

Method

Transformer encoder.

1. **Extracts a patch embedding** from the image by processing **all non-overlapping square patches**

   1) The patches are flatten
   2) Individually embedded by linear projection.
   3) Concatenated with a position embedding.
   4) Special token as the **Readout token** to serve as final, global image representation

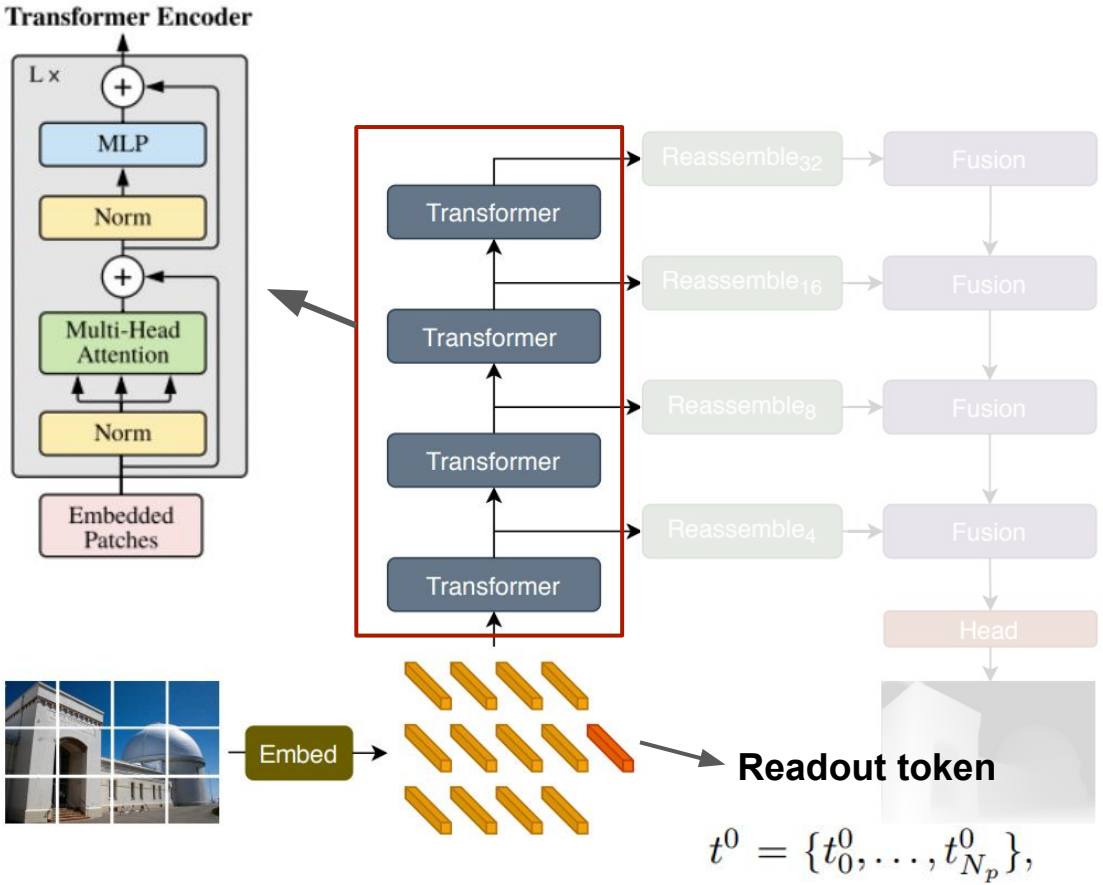


Method

Transformer encoder.

2. **The input tokens are transformed** using L transformer layers.

- **ViT-Base**: 12 layers, D : 768
- **ViT-Large**: 24 layers, D : 1024
- **ViT-Hybrid**: 12 layers, embeds features with ResNet50



Method

Convolutional decoder.

3. **Decoder assembles the set of tokens** into image-like feature representations at various resolutions

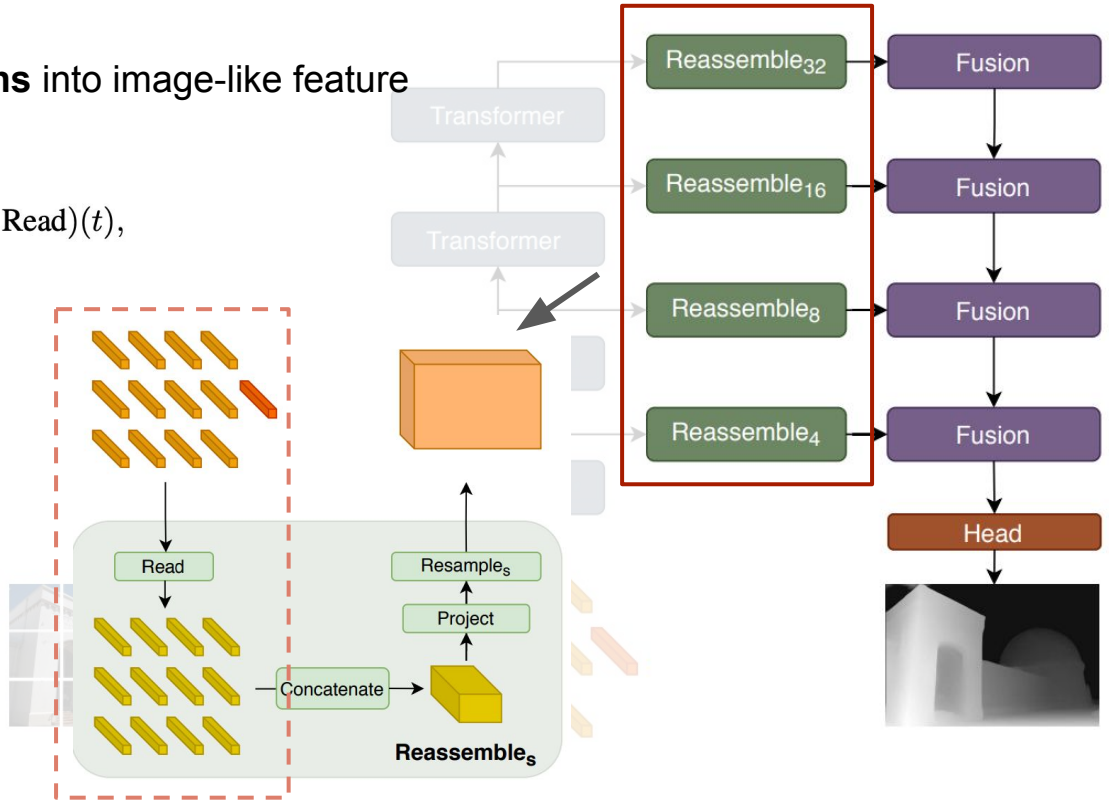
$$\text{Reassemble}_s^{\hat{D}}(t) = (\text{Resample}_s \circ \text{Concatenate} \circ \text{Read})(t),$$

1) First, map the Np + 1 tokens to a set of Np tokens

$$\text{Read}_{ignore}(t) = \{t_1, \ldots, t_{N_p}\}$$

$$\text{Read}_{add}(t) = \{t_1 + t_0, \ldots, t_{N_p} + t_0\}$$

$$\text{Read}_{proj}(t) = \{\text{mlp}(\text{cat}(t_1, t_0)), \ldots,$$



Method

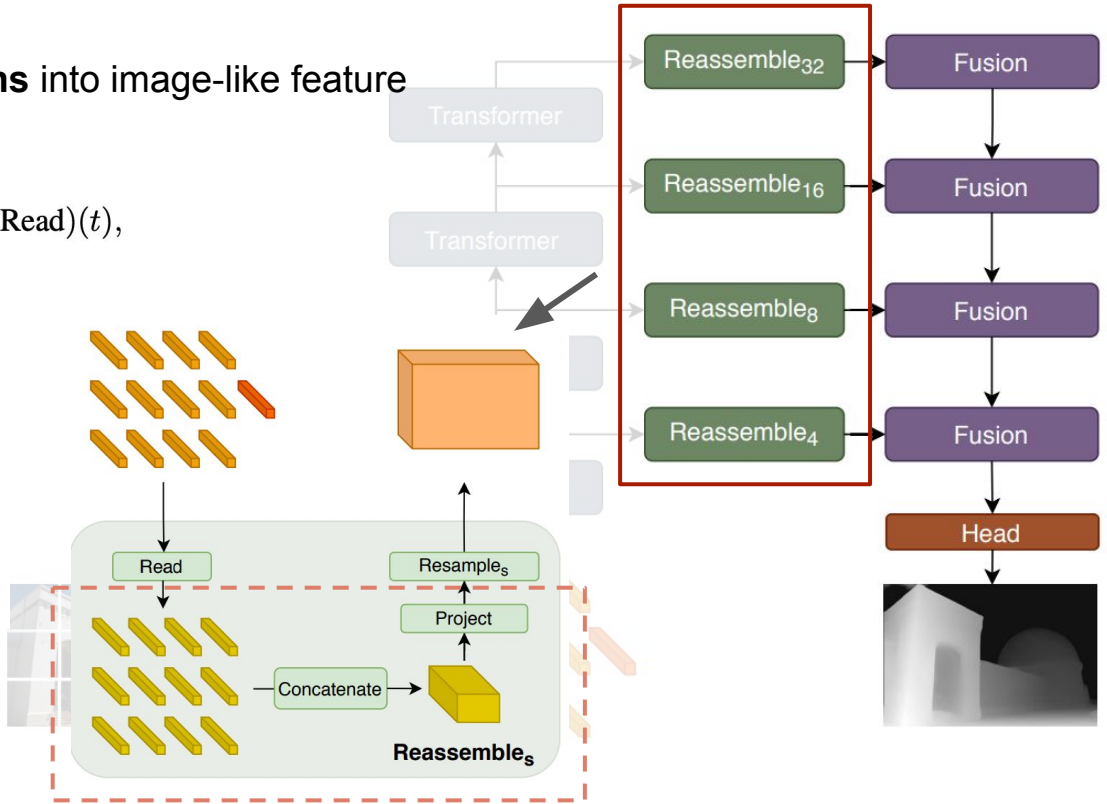
Convolutional decoder.

3. **Decoder assembles the set of tokens** into image-like feature representations at various resolutions

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

2) Np tokens can be reshaped into an image-like representation by placing each token

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



Method

Convolutional decoder.

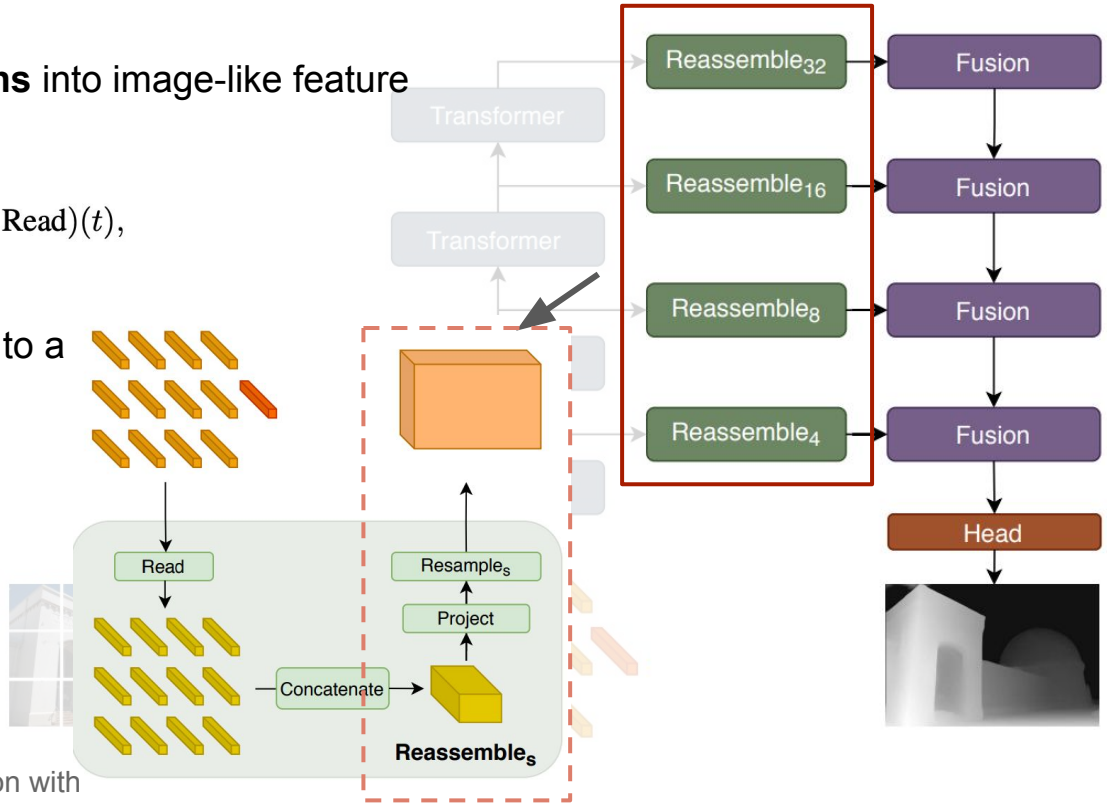
3. **Decoder assembles the set of tokens** into image-like feature representations at various resolutions

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

3) Image-like representation is passed to a spatial resampling layer to scale the representation

1 x 1 conv. for D channels
if s >= p: 3 x 3 conv. for downsampling
if s < p: 3 x 3 strided transpose conv. for upsampling

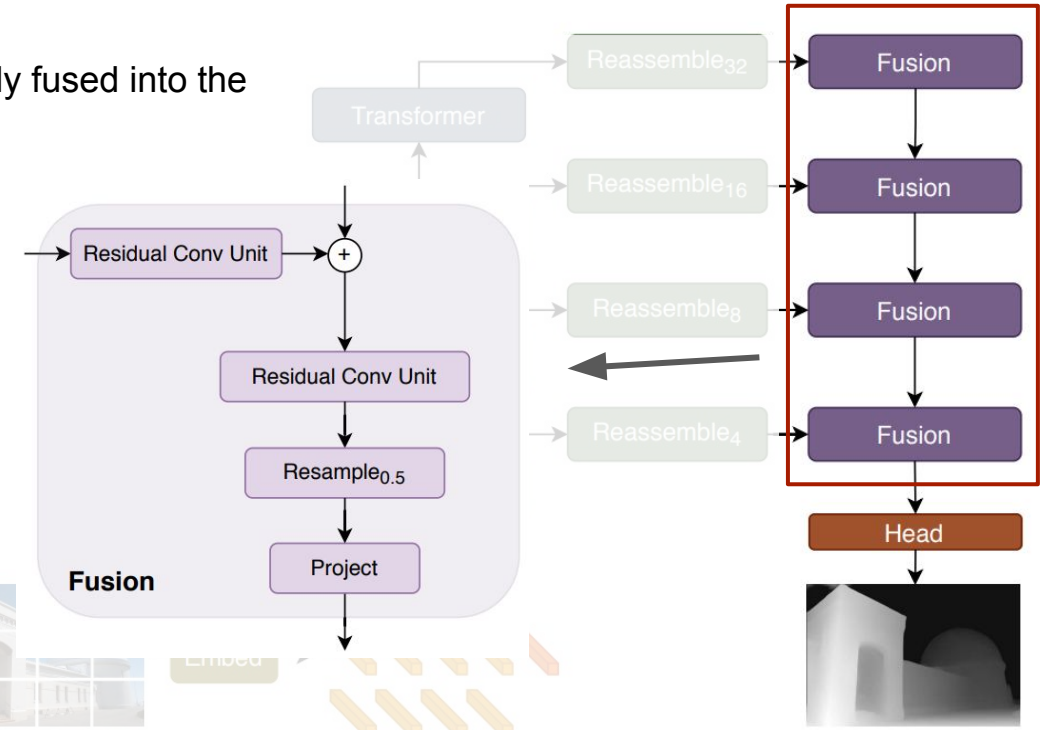
s: the output size ratio of the recovered representation with respect to the input image



Method

Convolutional decoder.

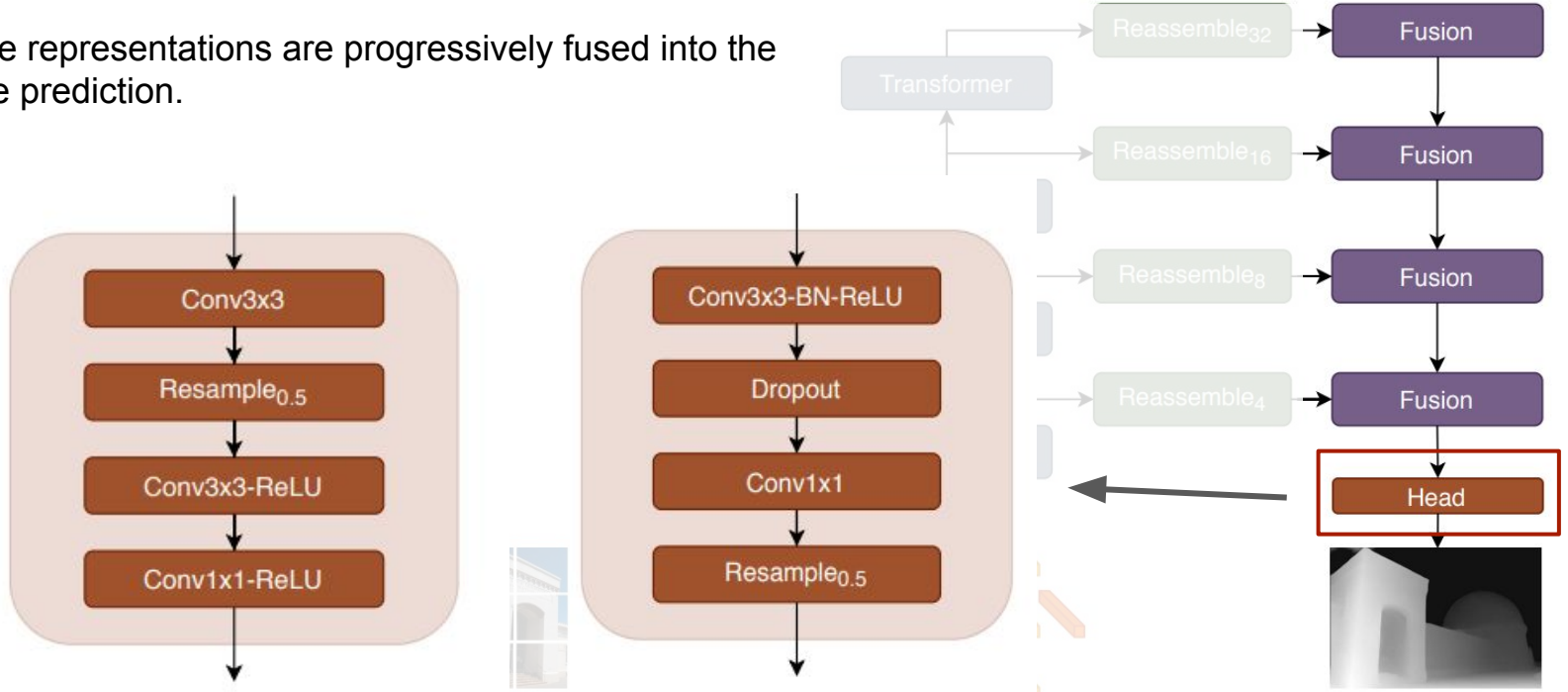
4. The feature representations are progressively fused into the final dense prediction.



Method

Convolutional decoder.

4. The feature representations are progressively fused into the final dense prediction.



(b) Monocular depth estimation head (c) Semantic segmentation head

Experiments

1. Monocular Depth Estimation.

Zero-shot cross-dataset transfer.

Datasets: mixed dataset from MiDaS (Mix5) + additional dataset (Mix6)

	Training set	DIW WHDR	ETH3D AbsRel	Sintel AbsRel	KITTI δ>1.25	NYU δ>1.25	TUM δ>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.

Experiments

1. Monocular Depth Estimation.

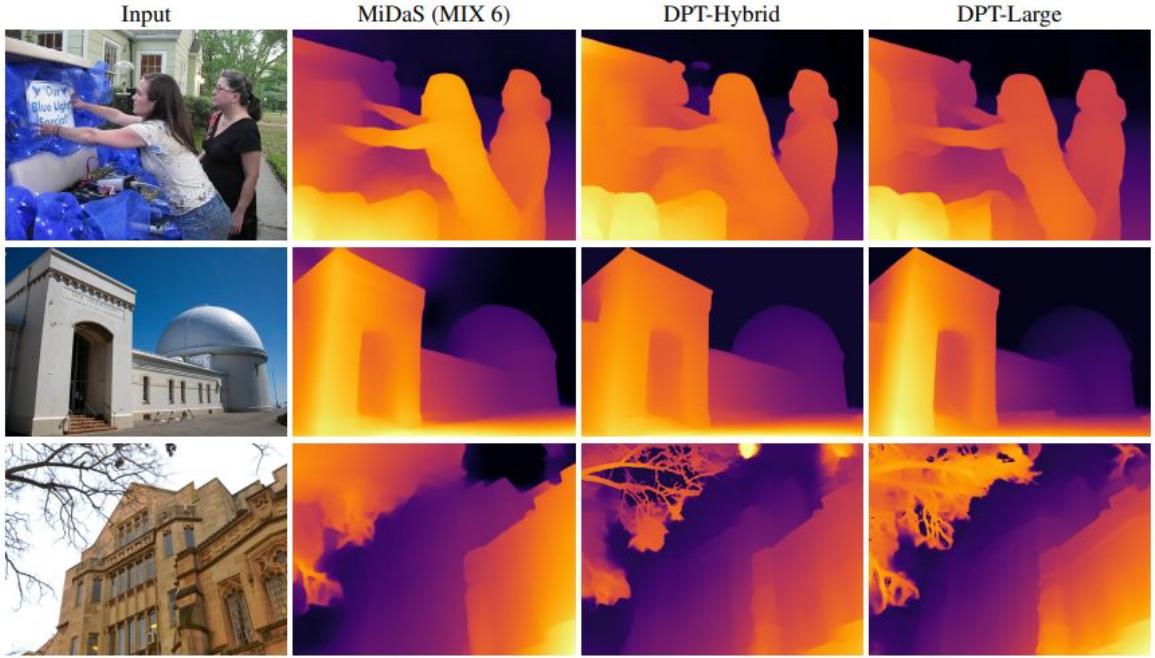


Figure 2. Sample results for monocular depth estimation. Compared to the fully-convolutional network used by MiDaS, DPT shows better global coherence (e.g., sky, second row) and finer-grained details (e.g., tree branches, last row).

Experiments

1. Monocular Depth Estimation.

Fine-tuning on small datasets.

	δ>1.25	δ>1.25^2	δ>1.25^3	AbsRel	RMSE	log10
DORN [13]	0.828	0.965	0.992	0.115	0.509	0.051
VNL [48]	0.875	0.976	0.994	0.111	0.416	0.048
BTS [20]	0.885	0.978	0.994	**0.110**	0.392	0.047
DPT-Hybrid	**0.904**	**0.988**	**0.998**	**0.110**	**0.357**	**0.045**

Table 2. Evaluation on NYUv2 depth.

	δ>1.25	δ>1.25^2	δ>1.25^3	AbsRel	RMSE	RMSE log
DORN [13]	0.932	0.984	0.994	0.072	2.626	0.120
VNL [48]	0.938	0.990	0.998	0.072	3.258	0.117
BTS [20]	0.956	0.993	0.998	**0.059**	2.756	0.096
DPT-Hybrid	**0.959**	**0.995**	**0.999**	0.062	**2.573**	**0.092**

Table 3. Evaluation on KITTI (Eigen split).

Experiments

2. Semantic Segmentation.

Datasets: ADE20K dataset
Fine tuning on smaller datasets: Pascal Context

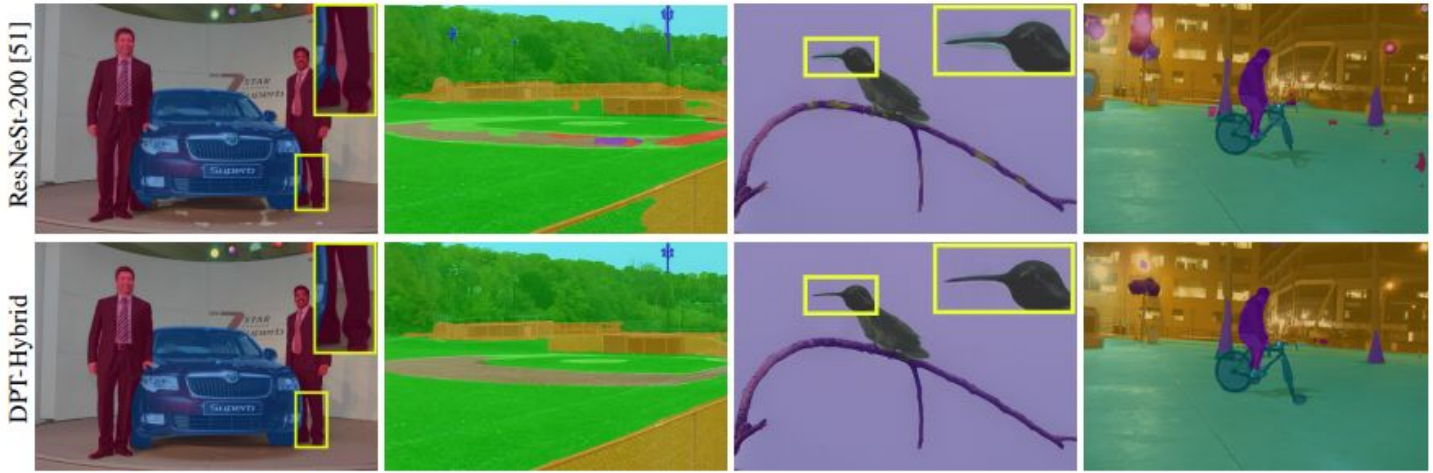


Figure 3. Sample results for semantic segmentation on ADE20K (first and second column) and Pascal Context (third and fourth column). Predictions are frequently better aligned to object edges and less cluttered.

Experiments

3. Ablations.

Ablation study on monocular depth estimation task

- Same hyperparameters used for depth estimation experiment
- Three datasets with high quality ground truth
  (4,1000 images each and 1000 images for validation)

- **Skip connections**

	Layer l	HRWSI	BlendedMVS	ReDWeb	Mean
Base	{3, 6, 9, 12}	0.0793	0.0780	0.0892	0.0822
	{6, 8, 10, 12}	0.0801	0.0789	0.0904	0.0831
	{9, 10, 11, 12}	0.0805	0.0766	0.0912	0.0828
Hybrid	{3, 6, 9, 12}	0.0747	**0.0748**	0.0865	0.0787
	{R0, R1, 9, 12}	**0.0742**	0.0751	**0.0857**	**0.0733**

Table 6. Performance of attaching skip connections to different encoder layers. Best results are achieved with a combination of skip connections from shallow and deep layers.

R0, R1: first and second downsampling stages of ResNet50 embedding

- **Readout token**

	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.

$$\text{Read}_{ignore}(t) = \{t_1, \ldots, t_{N_p}\}$$

$$\text{Read}_{add}(t) = \{t_1 + t_0, \ldots, t_{N_p} + t_0\}$$

$$\text{Read}_{proj}(t) = \{\text{mlp}(\text{cat}(t_1, t_0)), \ldots,$$

Experiments

3. Ablations.

- **Backbones**

	HRWSI	BlendedMVS	ReDWeb	Mean
ResNet50	0.0890	0.0887	0.1029	0.0935
ResNext101-WSL	0.0780	0.0751	0.0886	0.0806
DeIT-Base	0.0798	0.0804	0.0925	0.0842
DeIT-Base-Dist	0.0758	0.0758	0.0871	0.0796
ViT-Base	0.0797	0.0764	0.0895	0.0819
ViT-Large	0.0740	0.0747	**0.0846**	**0.0778**
ViT-Hybrid	**0.0738**	**0.0746**	0.0864	0.0783

Table 8. Ablation of backbones. The hybrid and large backbones consistently outperform the convolutional baselines. The base architecture can outperform the convolutional baseline with better pretraining (DeIT-Base-Dist).

- **Inference and speed**

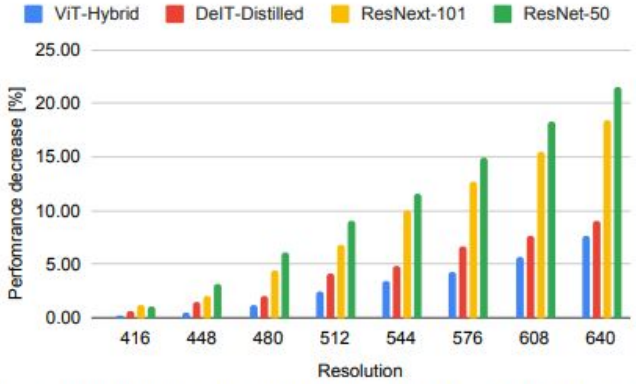


Figure 4. Relative loss in performance for different inference resolutions (lower is better).

	MiDaS	DPT-Base	DPT-Hybrid	DPT-Large
Parameters [million]	105	112	123	343
Time [ms]	32	17	38	35

Table 9. Model statistics. DPT has comparable inference speed to the state of the art.

Conclusions

Introduced dense prediction transformer (DPT)

1. Effectively leverages vision transformers for dense prediction task
2. Outperforms networks with fully-convolutional architecture
3. Shows full potentional when trained on large-scale dataset