SOTR: Segmenting Objects with Transformers

—— China Agricultural University, arxiv.2021

Fang Zhiyuan

2021.8.27

Prior knowledge

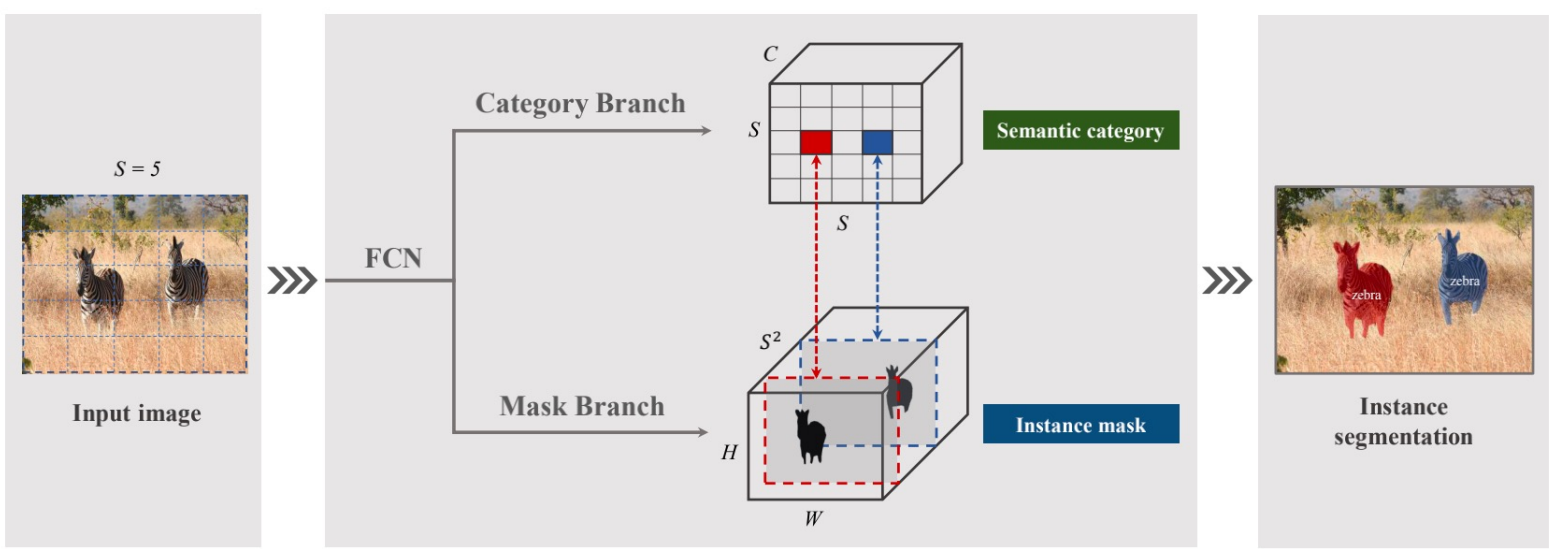
- **Semantic Segmentation & Instance Segmentation**

Semantic Segmentation

Instance Segmentation

Prior knowledge

- **Instance Segmentation**
  - Top-down instance segmentation：Detection + Segmentation
    - Two stage: Mask RCNN
    - One stage: YOLACT
  - Bottom-up instance segmentation：Segmentation + Post process
    - SGN、SSAP
    - SOLO: Split grid – classification – generate mask

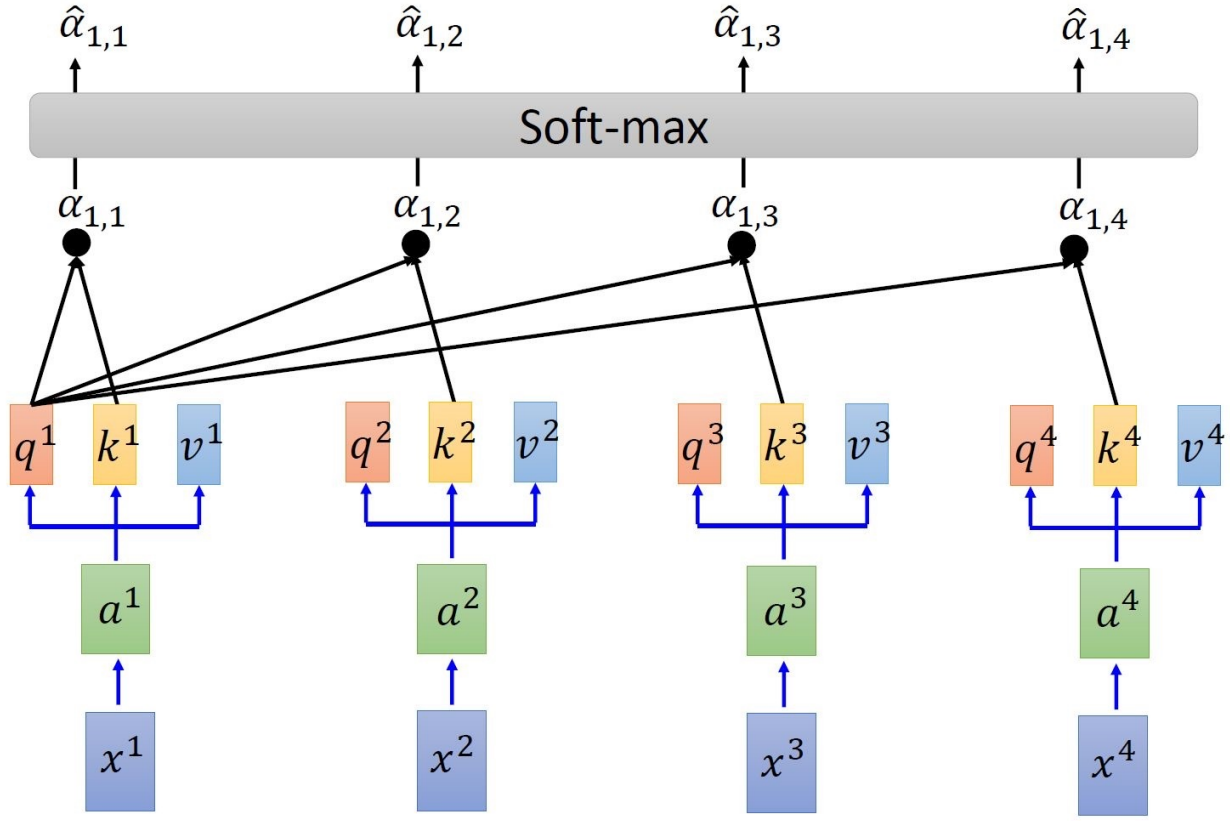


Method

Self-attention

$a_{1,i} = q^1 \cdot k^1 / \sqrt{d}$, d is the dim of q and k

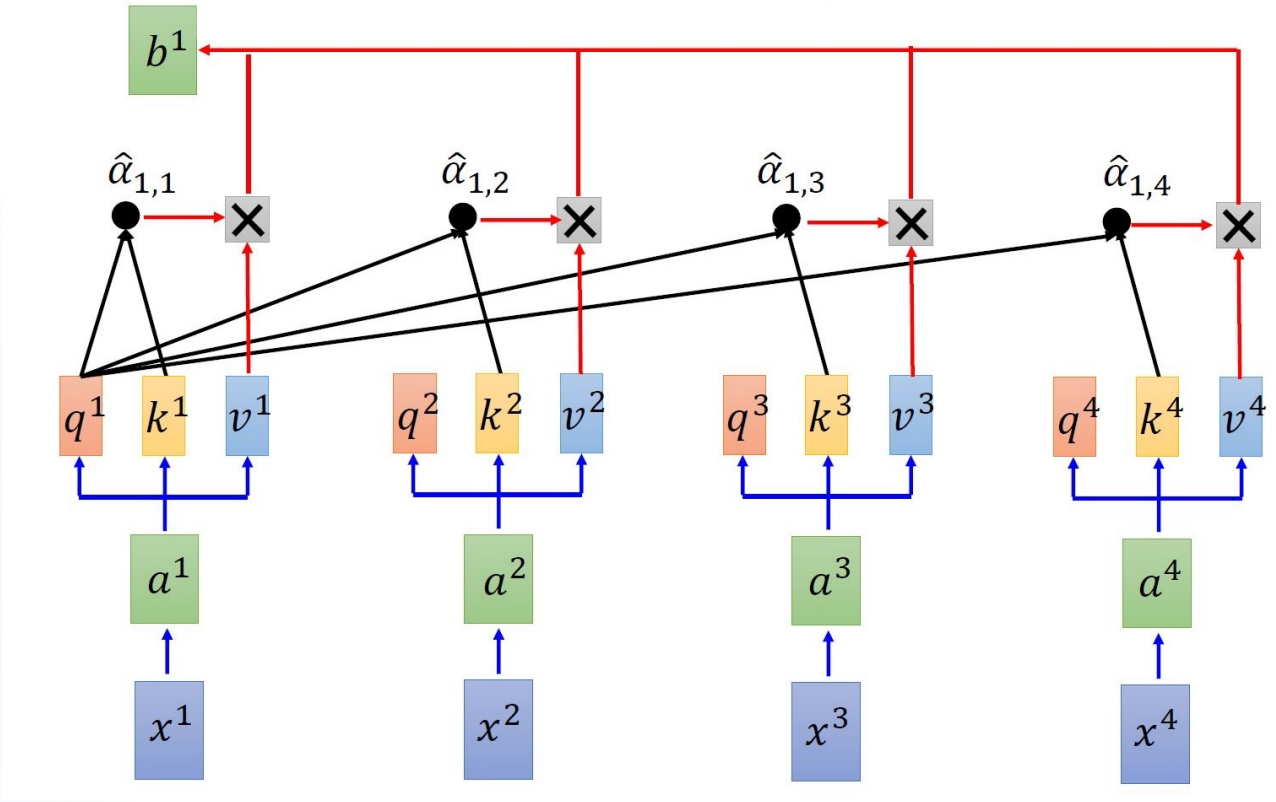
$\hat{a}_{1,i} = \exp(a_{1,i}) / \sum_j \exp(a_{i,j})$



Prior knowledge

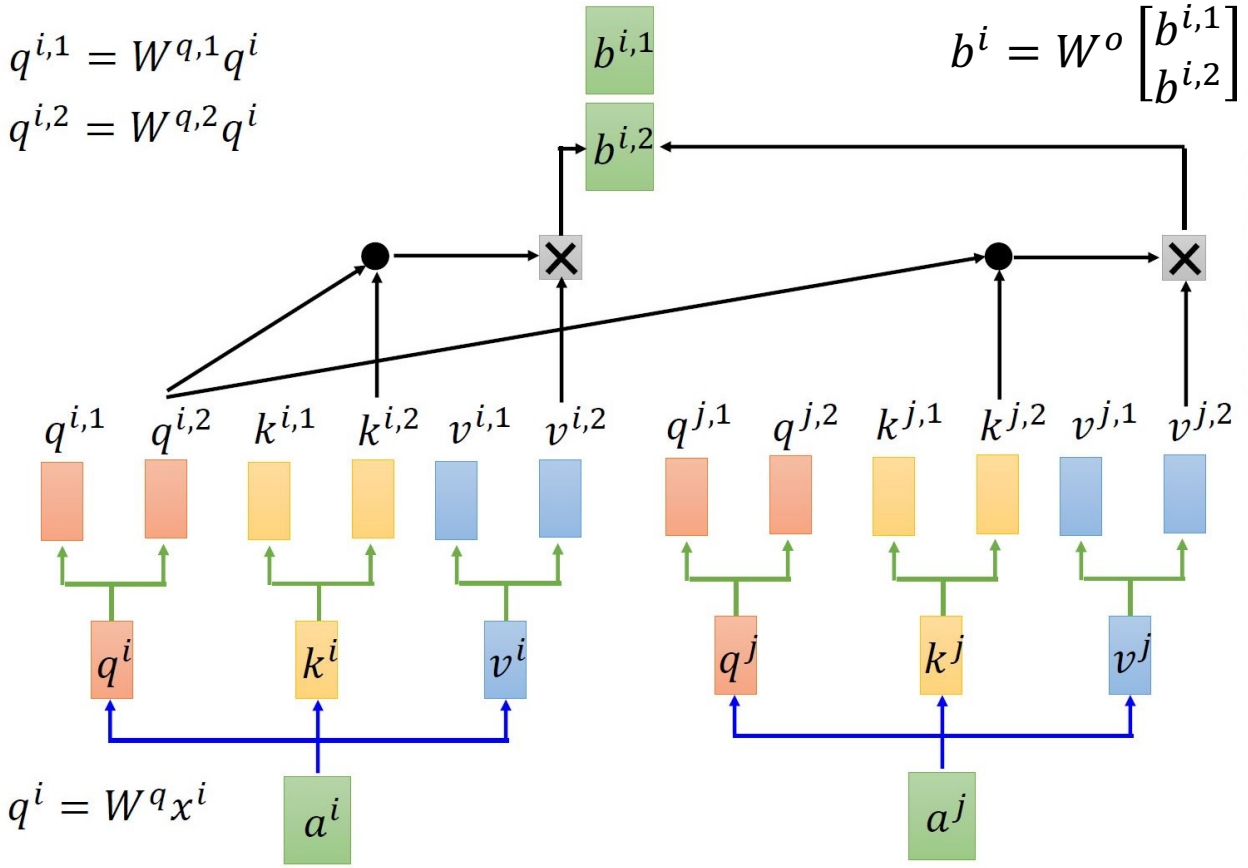
Self-attention

$$b^1 = \sum_i \hat{\alpha}_{1,i}\, v^i$$



Prior knowledge

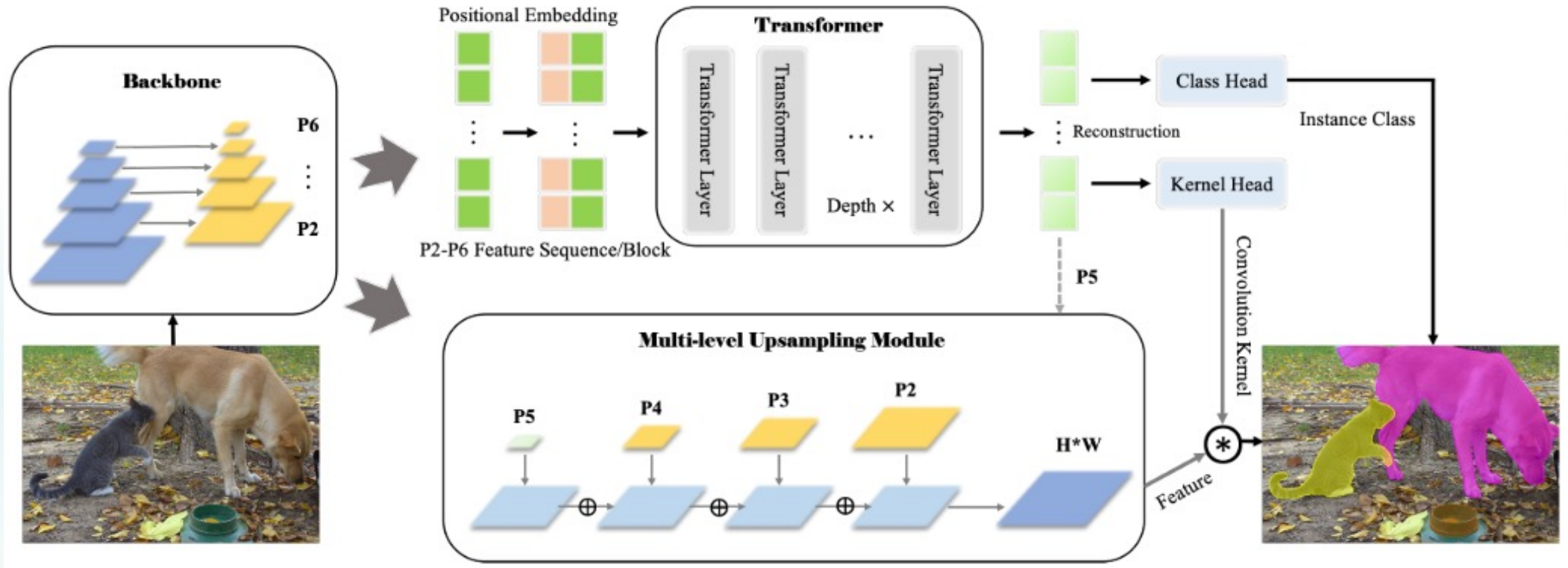
Multi-head Self-attention (2 heads)



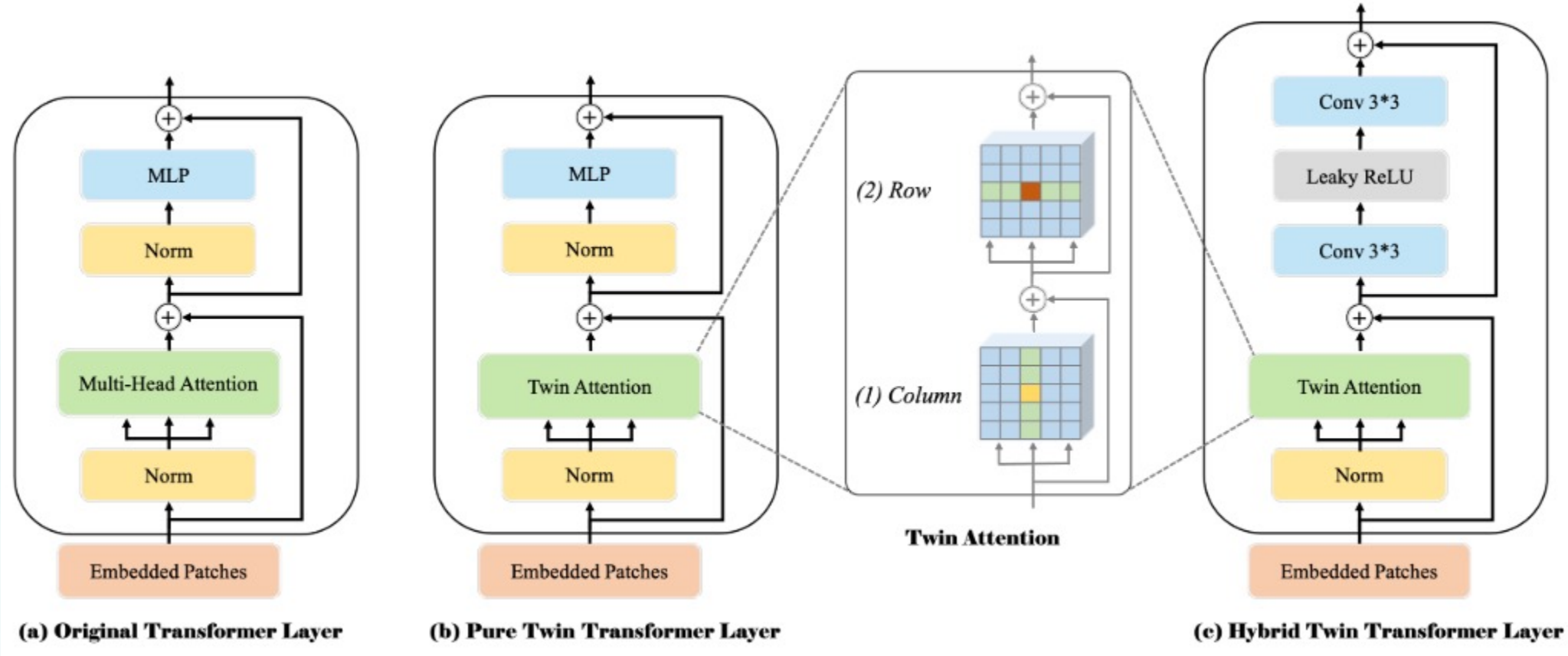
Contribution

- Introduce an innovative **CNN-transformer-hybrid** instance segmentation framework
- Devise the **twin attention**, a new position-sensitive self-attention mechanism
- SOTR **does not need to be pre-trained** on large datasets
- SOTR achieves 40.2% of AP with the ResNet-101-FPN backbone on the MS COCO

Method


Backbone
P6
P2
Positional Embedding
P2-P6 Feature Sequence/Block
Transformer
Transformer Layer
Transformer Layer
Depth ×
Transformer Layer
Reconstruction
Class Head
Instance Class
Kernel Head
Convolution Kernel
P5
Multi-level Upsampling Module
P5
P4
P3
P2
H*W
Feature

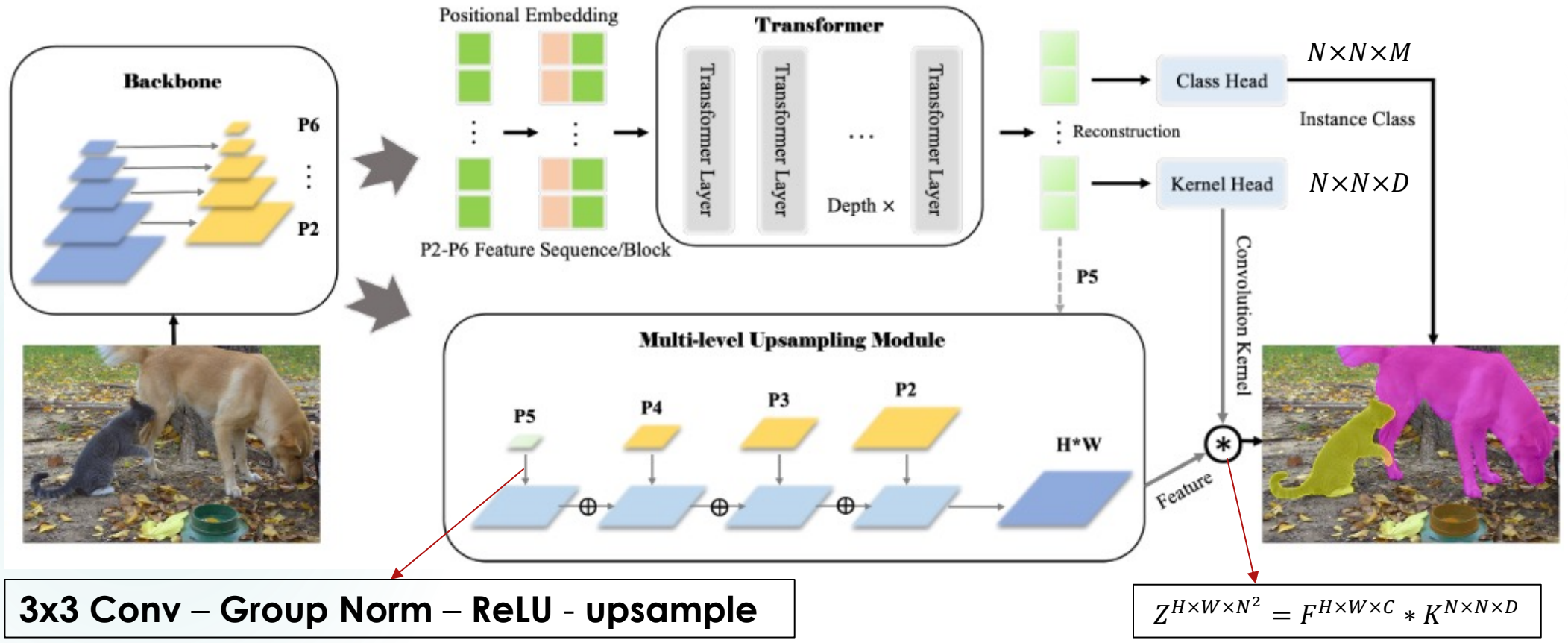
Method



$O((H \times W)^2)$

$O(H \times W^2 + W \times H^2)$

Method


Backbone
P6
P2
Positional Embedding
P2-P6 Feature Sequence/Block
Transformer
Transformer Layer
Transformer Layer
Depth ×
Transformer Layer
Reconstruction
Class Head
N×N×M
Instance Class
Kernel Head
N×N×D
Convolution Kernel
P5
Multi-level Upsampling Module
P5
P4
P3
P2
H*W
Feature
3x3 Conv – Group Norm – ReLU - upsample
$Z^{H\times W\times N^2} = F^{H\times W\times C} * K^{N\times N\times D}$

Experiments

- **Implementation details**
  - Dataset: COCO
  - 300K iterations
  - 4 V100 GPUs of 32G RAM (3-4 days)
  - Batch size 8

Experiments

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FCIS [25]	Res-101-C5	29.5	51.5	30.2	8.0	31.0	49.7
MaskLab+ [7]	Res-101-C4	37.3	59.8	39.6	16.9	39.9	53.5
Mask R-CNN [15]	Res-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN*	Res-101-FPN	37.8	59.8	40.7	**20.5**	40.4	49.3
RetinaMask [13]	Res-101-FPN	34.7	55.4	36.9	14.3	36.7	50.5
MS R-CNN [19]	Res-101-FPN	38.3	58.8	41.5	17.8	40.4	54.4
TensorMask [9]	Res-101-FPN	37.1	59.3	39.4	17.4	39.1	51.6
ShapeMask [23]	Res-101-FPN	37.4	58.1	40.0	16.1	40.1	53.8
YOLACT [3]	Res-101-FPN	31.2	50.6	32.8	12.1	33.3	47.1
YOLACT++ [2]	Res-101-FPN	34.6	53.8	36.9	11.9	36.8	55.1
PolarMask [41]	Res-101-FPN	32.1	53.7	33.1	14.7	33.8	45.3
SOLO [36]	Res-101-FPN	37.8	59.5	40.4	16.4	40.6	54.2
BlendMask [6]	Res-101-FPN	38.4	60.7	41.3	18.2	41.5	53.3
CenterMask [38]	Hourglass-104	34.5	56.1	36.3	16.3	37.4	48.4
MEInst [42]	Res-101-FPN	33.9	56.2	35.4	19.8	36.1	42.3
SOLOv2 [37]	Res-101-FPN	39.7	60.7	42.9	17.3	42.9	57.4
SOTR	Res-101-FPN	**40.2**	**61.2**	**43.4**	10.3	**59.0**	**73.0**
SOLOv2 [37]	Res-DCN-101-FPN	41.7	63.2	45.1	**18.0**	45.0	61.6
SOTR	Res-DCN-101-FPN	**42.1**	**63.3**	**45.5**	11.5	**60.8**	**74.4**

Experiments

SOTR incorporating different transformers on COCO test-dev

Transformer	AP	AP_S	AP_M	AP_L
Original	37.1	9.0	56.1	71.0
Pure Twin	39.7	9.9	**59.1**	**73.6**
Hybrid Twin	**40.2**	**10.3**	59.0	73.0

Feature map substitution on multi-level upsampling process

P4	P5	AP	AP_S	AP_M	AP_L
		38.8	9.7	58.0	72.0
	✓	**40.2**	**10.3**	59.0	73.0
✓	✓	39.9	10.1	**59.1**	**73.7**

Comparison of different depth

Transformer	Depth	AP	Time(ms)	Memory
Original	6	36.2	147	6907M
	12	37.1	199	10696M
Twin	6	37.6	**113**	**3778M**
	12	**40.2**	161	5492M