## ECCV 2018

# DeepLabV3+: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation

## CVPR 2021

# SETR: Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers

## NeurIPS 2021

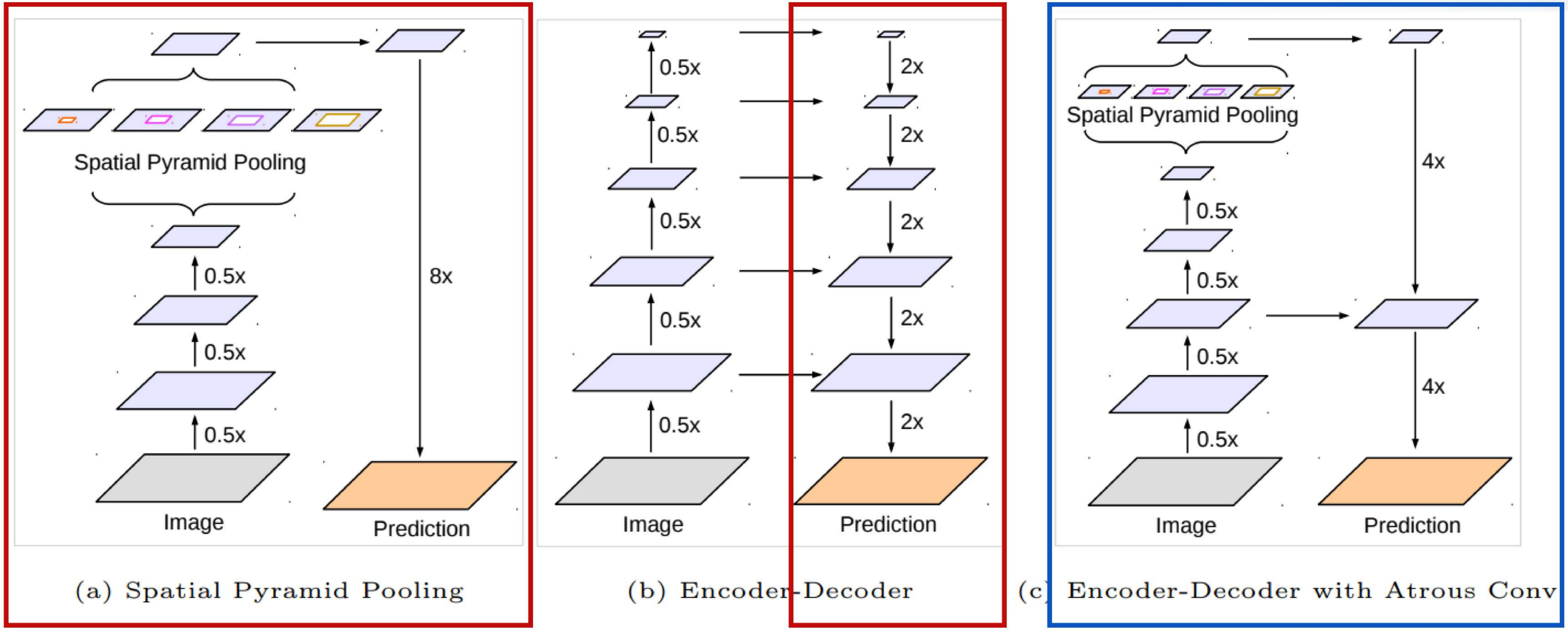
# SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers

고민수

# DeepLabV3+

## Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation

# Summary




**DeepLabV3**

**Decoder module**

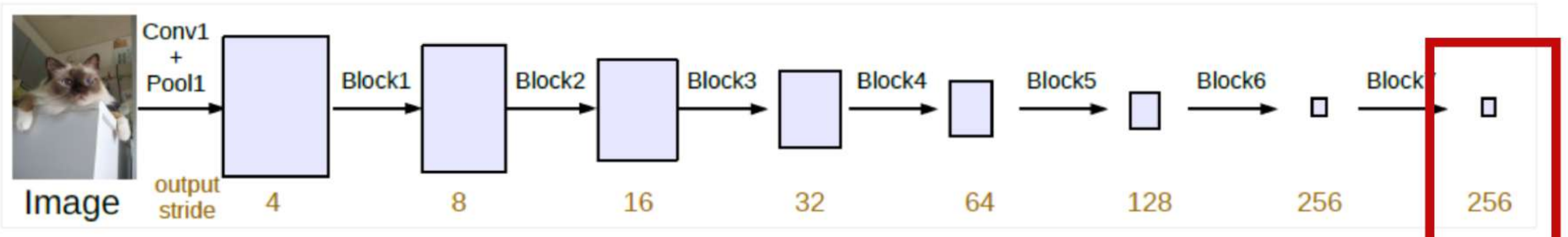
**DeepLabV3+**

exploiting
the multi-scale information

gradually recovers
the spatial information

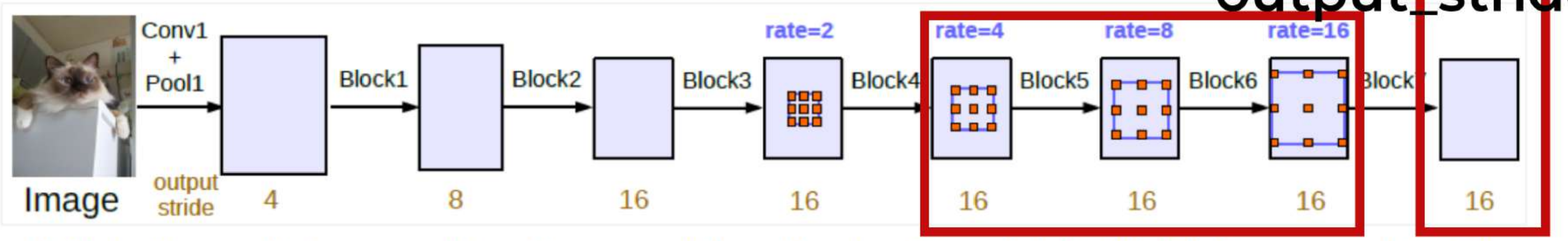
# Encoder-Decoder with Atrous Convolution

## Atrous convolution



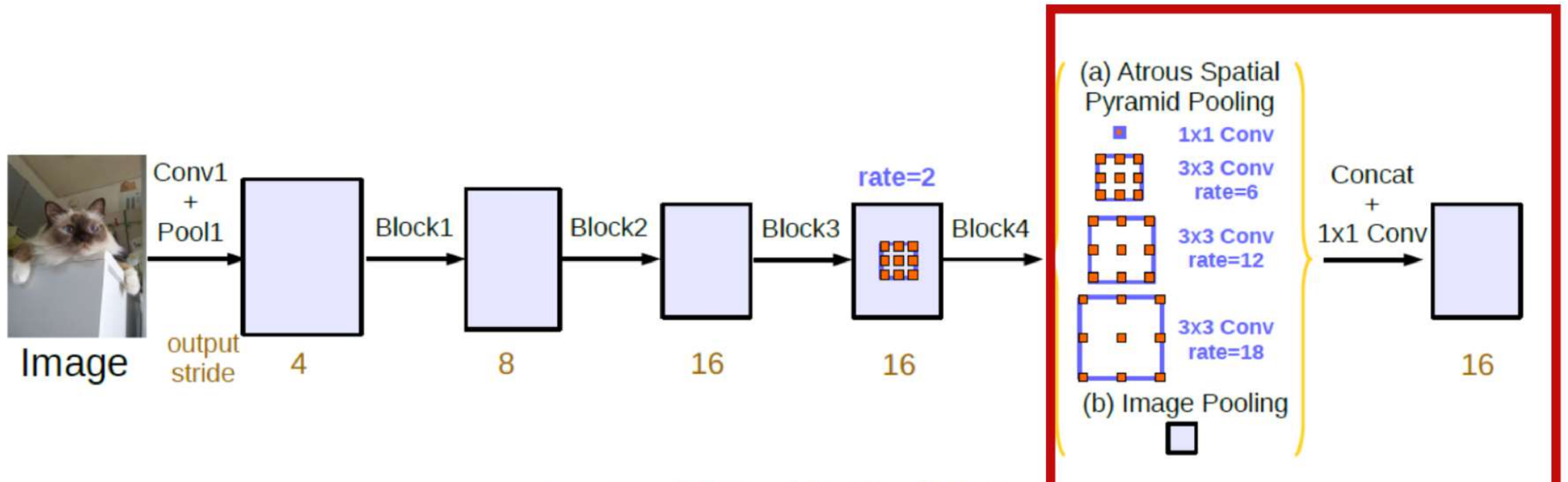

(a) Going deeper without atrous convolution.

output_stride



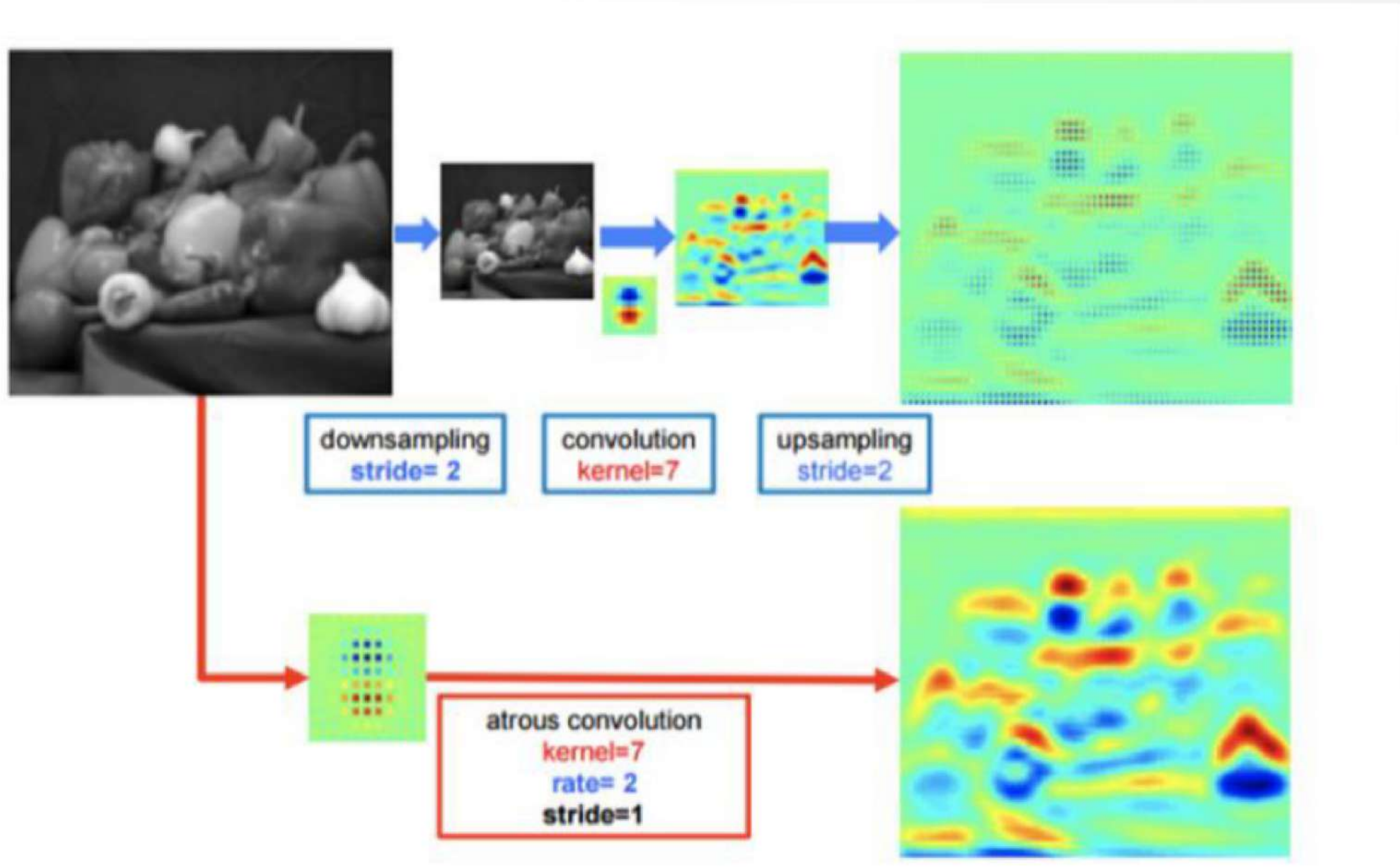

(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

**Smaller output feature map is more efficient in the segmentation task.**




Atrous Spatial Pyramid Pooling (ASPP)

**downsampling - convolution - upsampling**

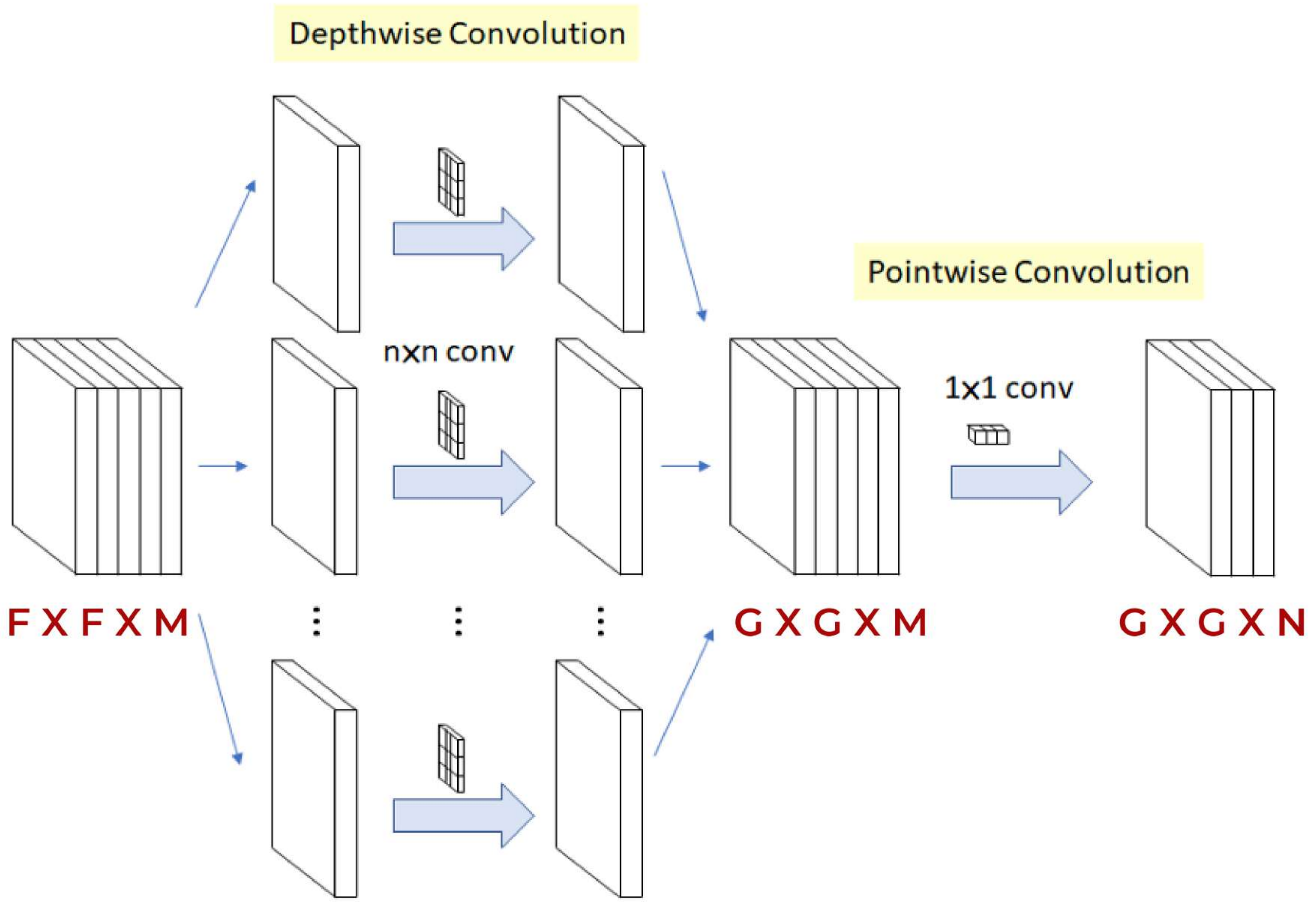


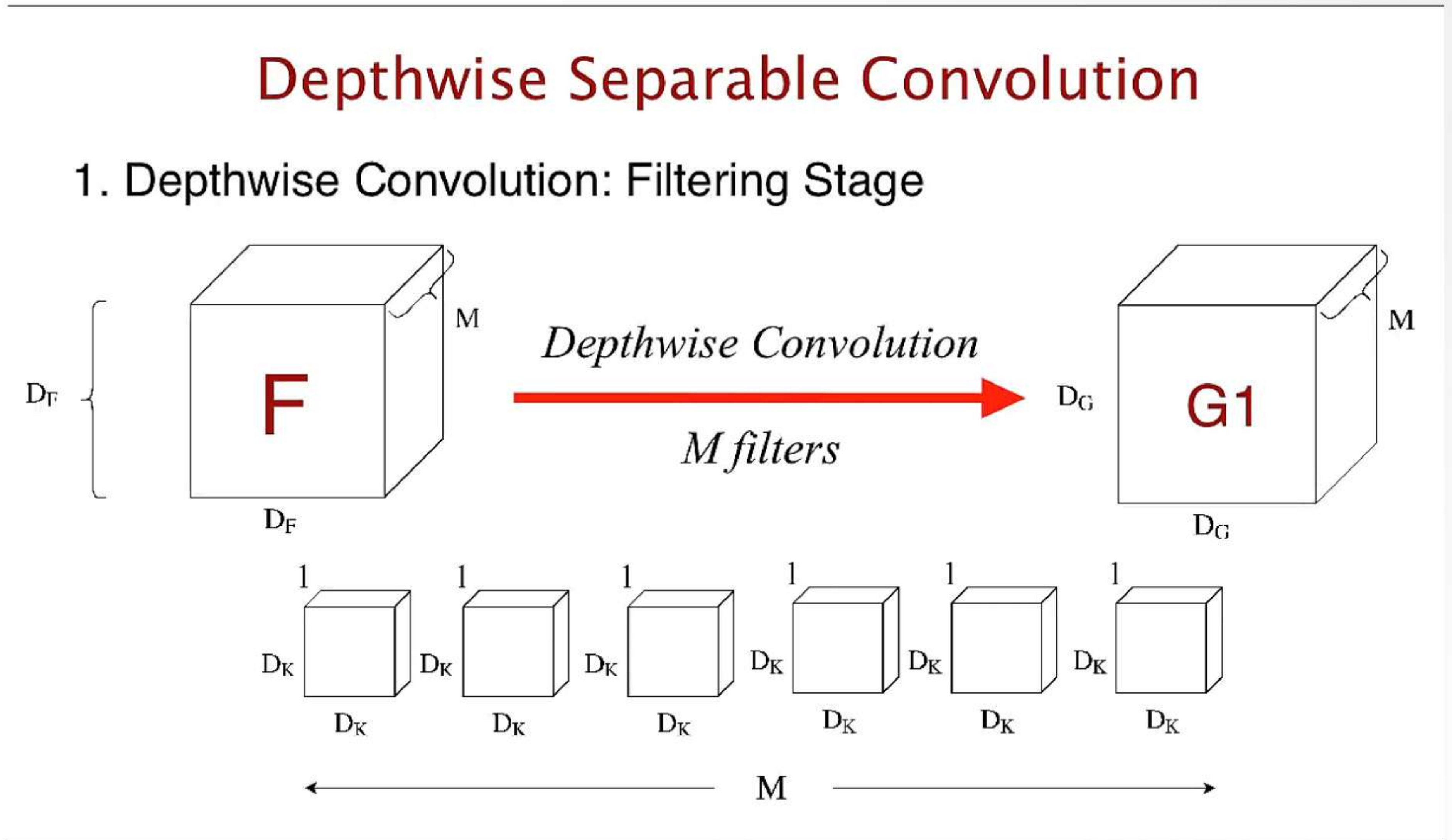

**atrous convolution**

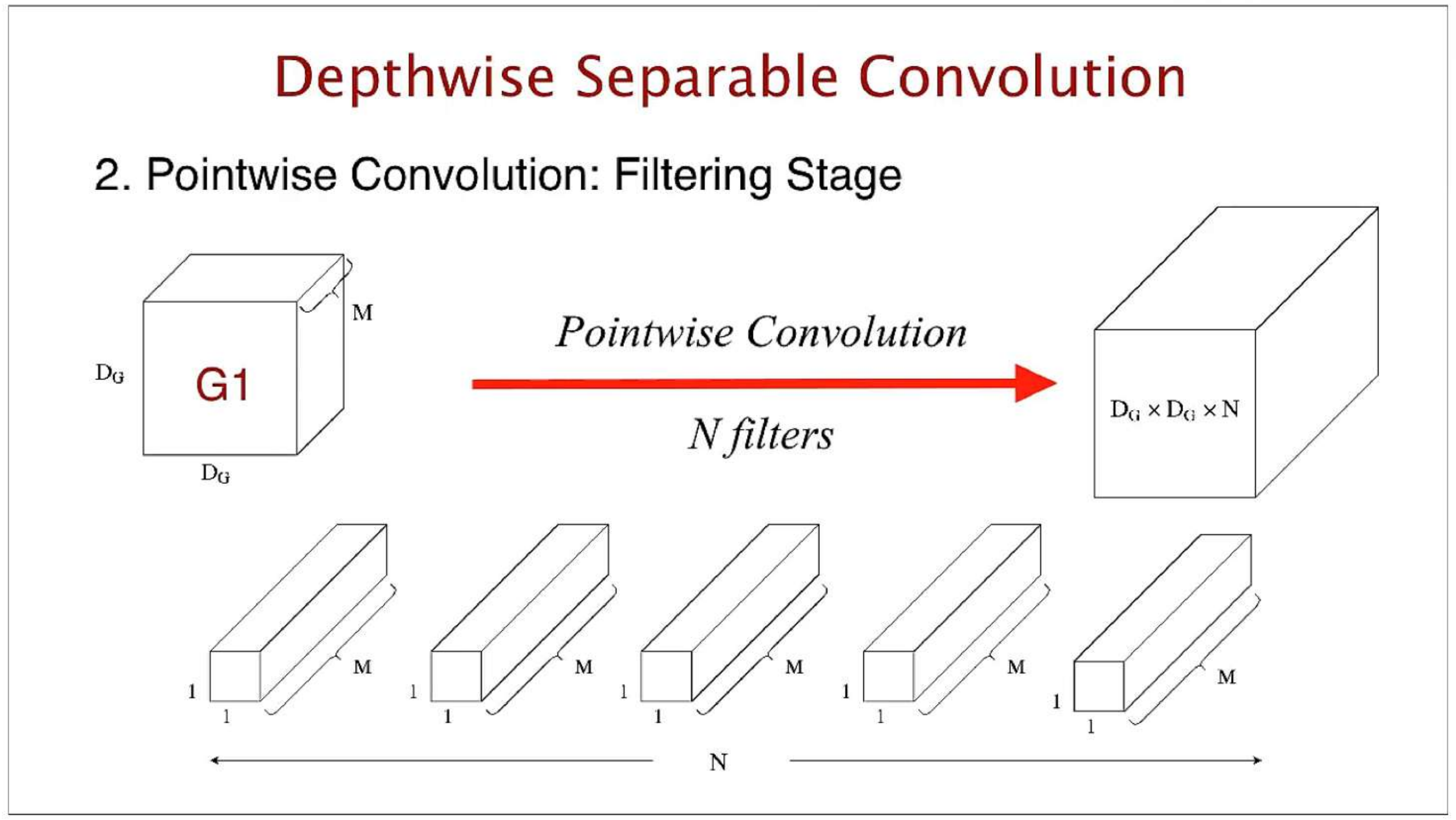
**Using atrous convolution shows better performance than downsampling - convolution - upsampling.**

# Encoder-Decoder with Atrous Convolution

## Depthwise separable convolution



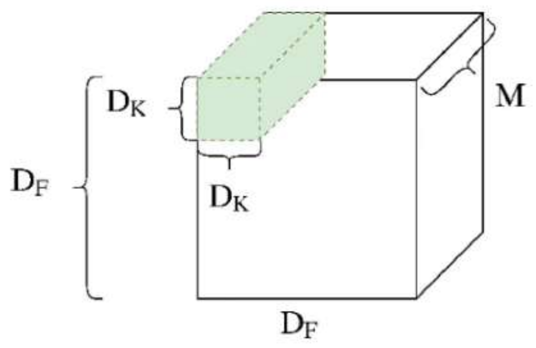



# Encoder-Decoder with Atrous Convolution

## Standard vs Depthwise

### Convolution




Mults once = $D_K^2 \times M$

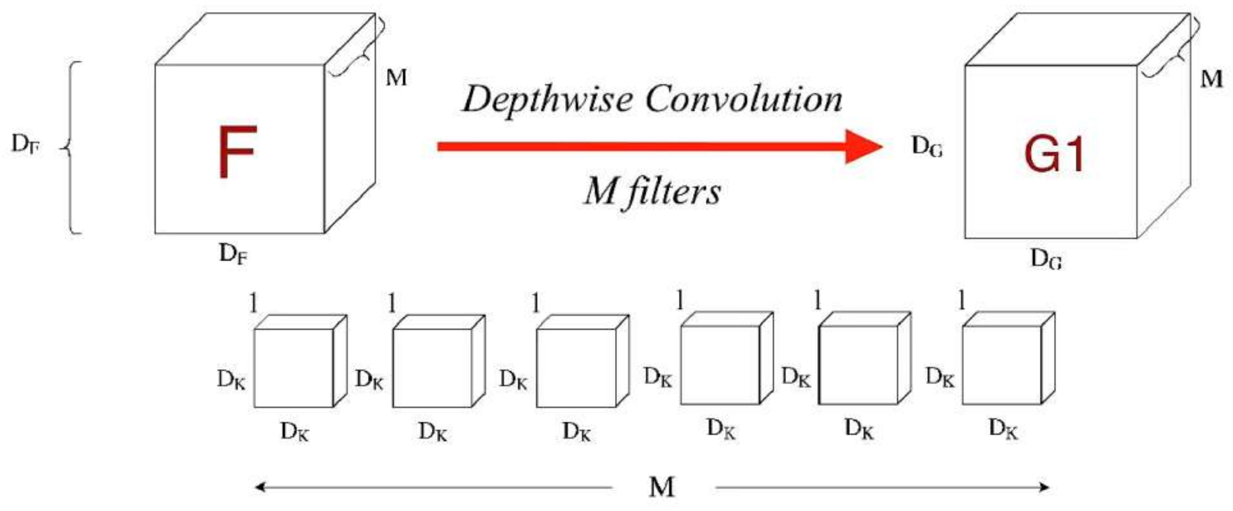
Mults per Kernel = $D_G^2 \times D_K^2 \times M$

Mults N Kernels = $N \times D_G^2 \times D_K^2 \times M$

**N * G² * K² * M**

### Depthwise Separable Convolution

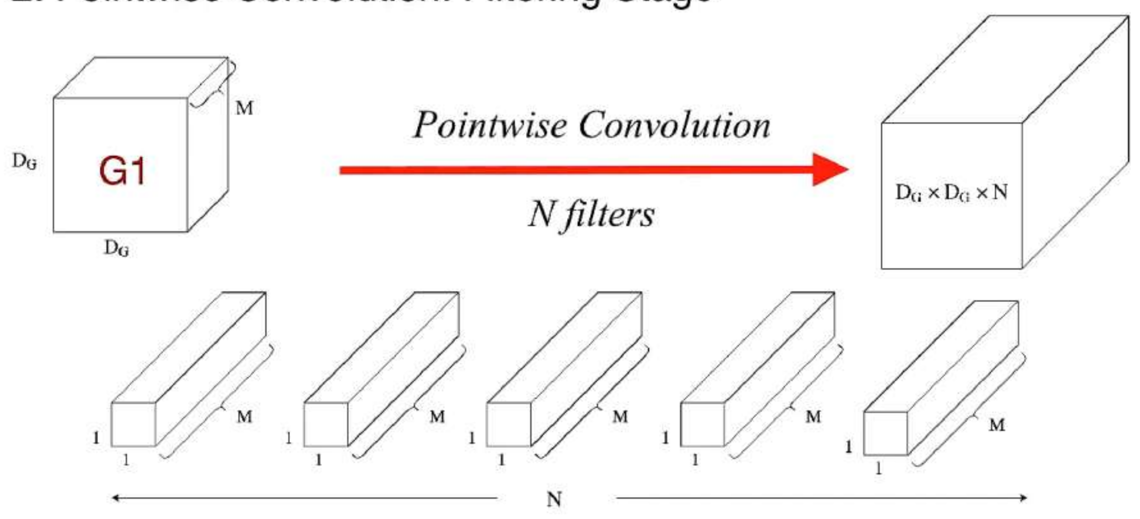
1. Depthwise Convolution: Filtering Stage




**G² * K² * M**

### Depthwise Separable Convolution

2. Pointwise Convolution: Filtering Stage




**N * G² * M**

$$\frac{No.Mults\ in\ Depthwise\ Separable\ Conv}{No.Mults\ in\ Standard\ Conv} = \frac{M \times D_G^2\ (D_K^2 + N)}{N\ \times D_G \times D_G \times D_K \times D_K \times M}$$

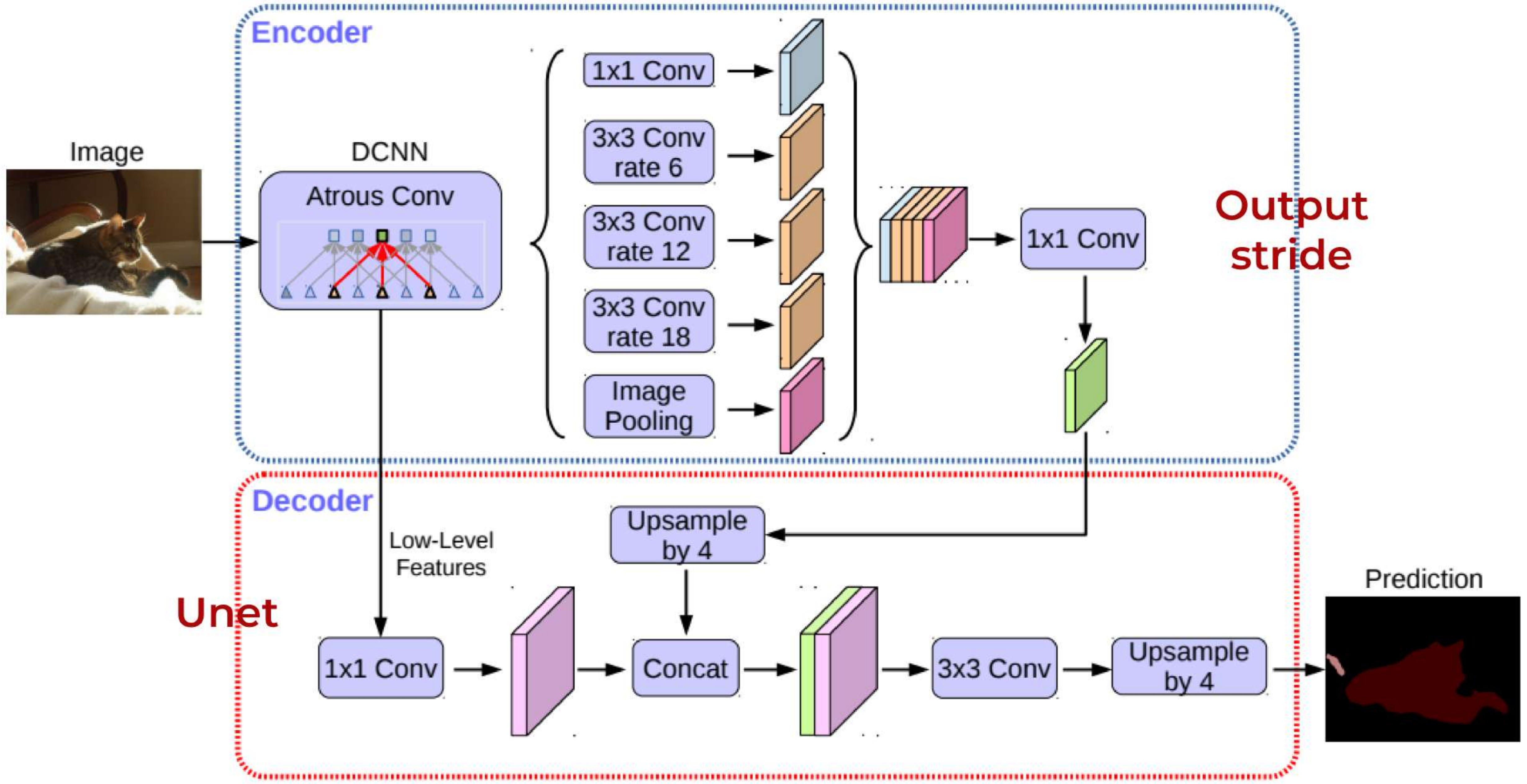
$$\frac{No.\,Mults\ in\ Depthwise\ Separable\ Conv}{No.\,Mults\ in\,Standard\,Conv} = \frac{D_K^2 + N}{(D_K^2 \times N)} = \frac{1}{N} + \frac{1}{D_K^2}$$

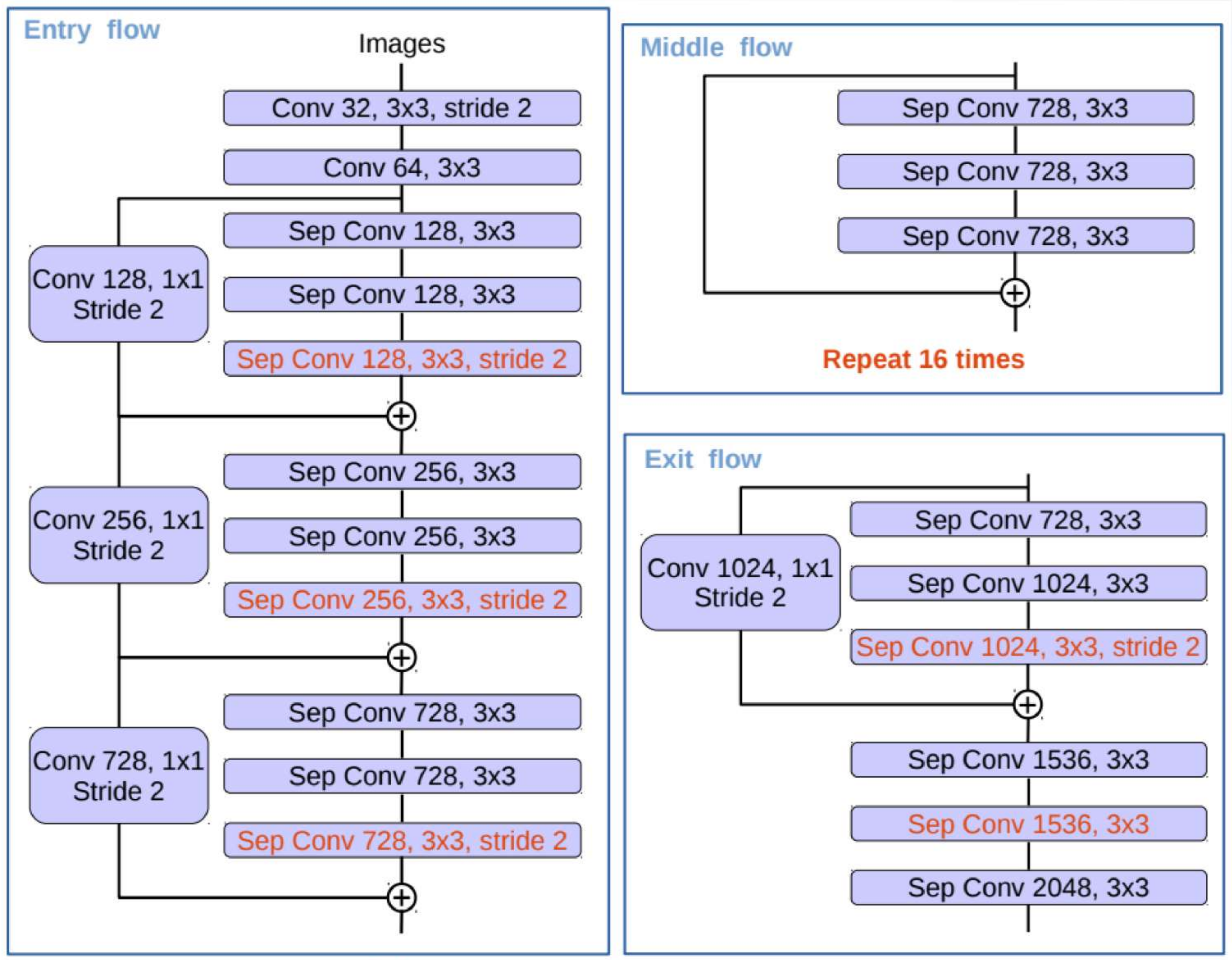
If N=1024, K=3,

$$\frac{No.\,Mults\ in\ Depthwise\ Separable\ Conv}{No.\,Mults\ in\,Standard\,Conv} = \boxed{\frac{1}{1024} + \frac{1}{3^2} = 0.112}$$

# DeepLabV3+

## Structure



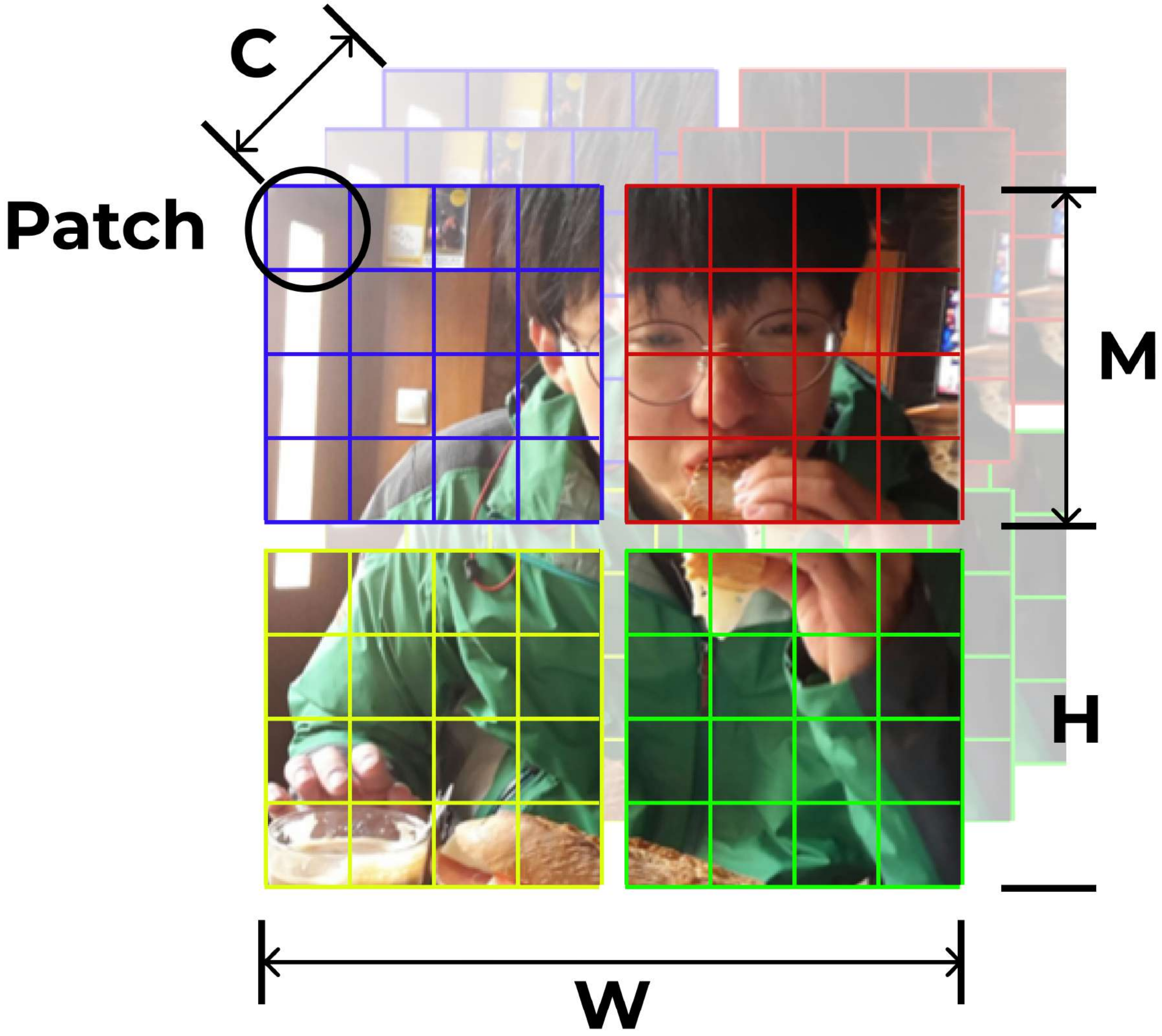

**Backbone - modified Xception**

1. More layer
2. All max pooling operations are replaced by Depthwise separable convolutions
3. Batch normalization and ReLU ard added

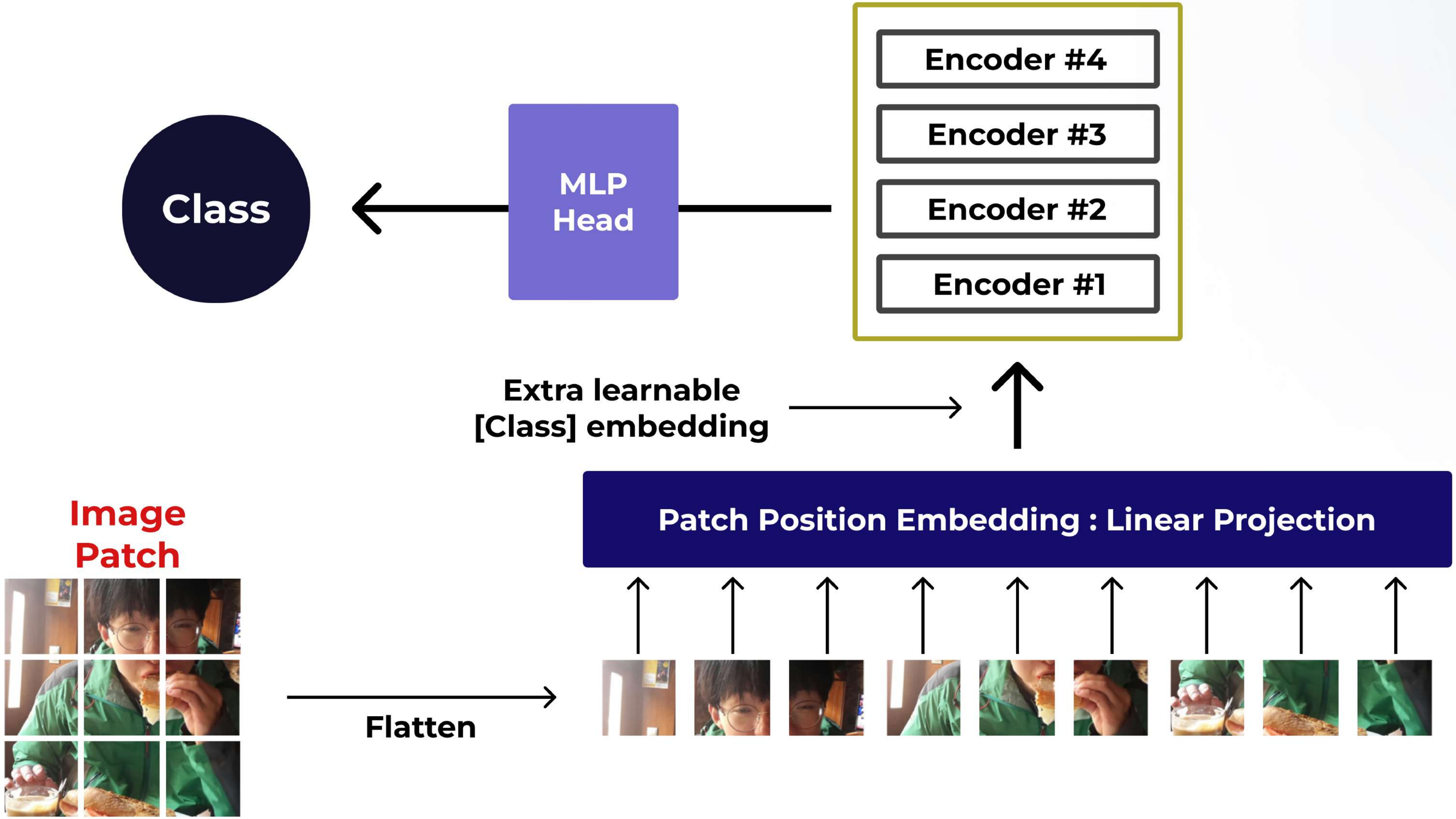
# Vision Transformer

## AN IMAGE IS WORTH 16X16 WORDS:
## TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

# Notice



# Vision Transformer architecture

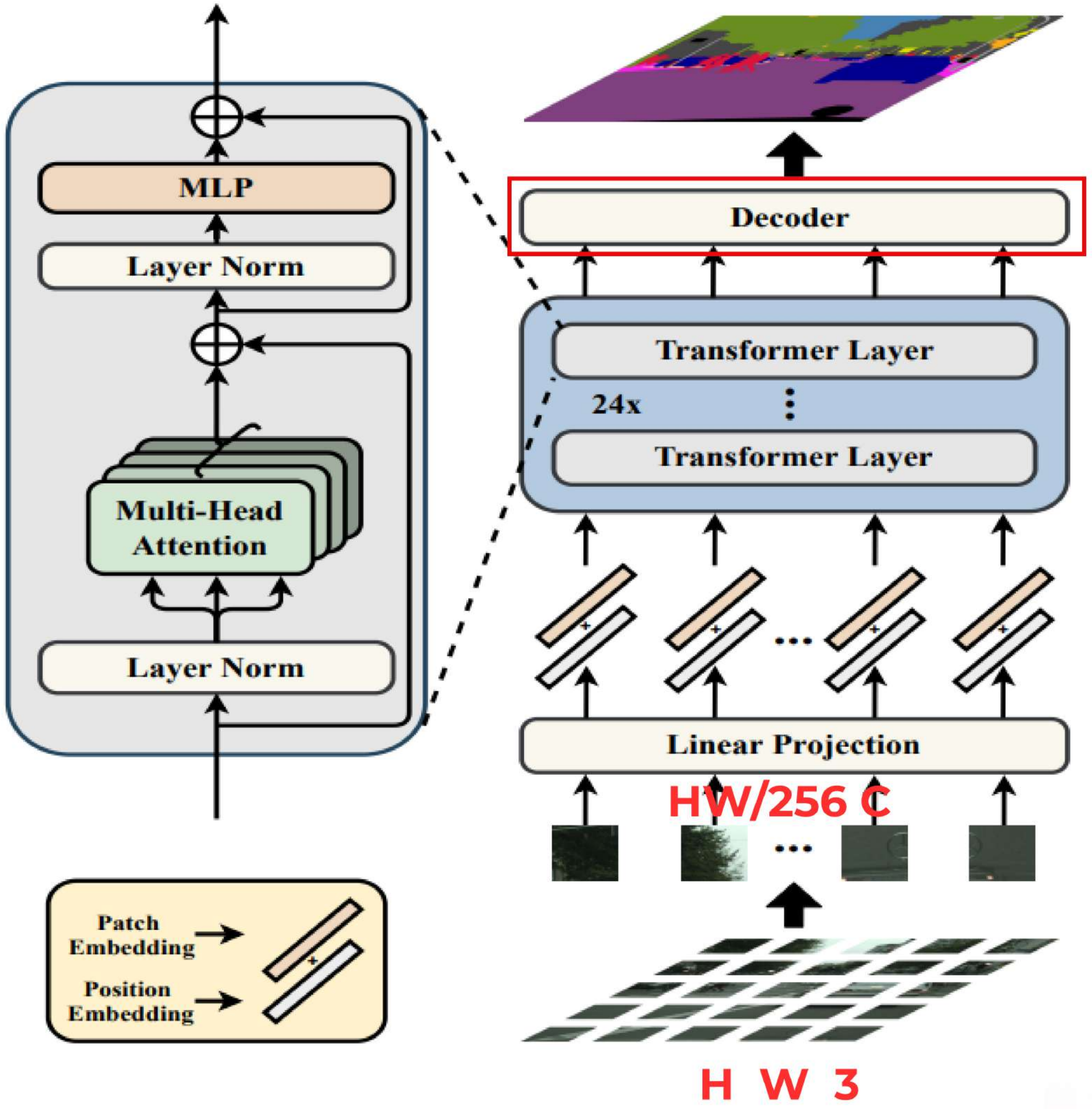


# SETR

## Rethinking Semantic Segmentation
## from a Sequence-to-Sequence Perspective with Transformers

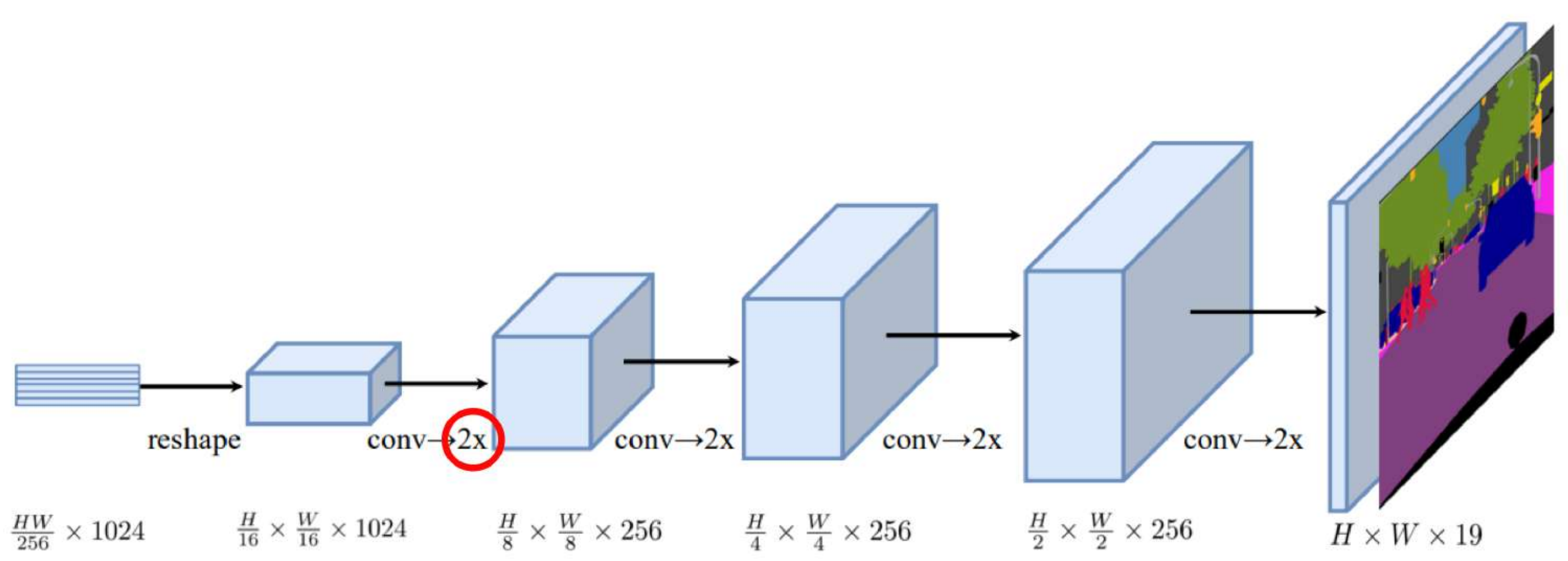
# SETR Summary

## Structure

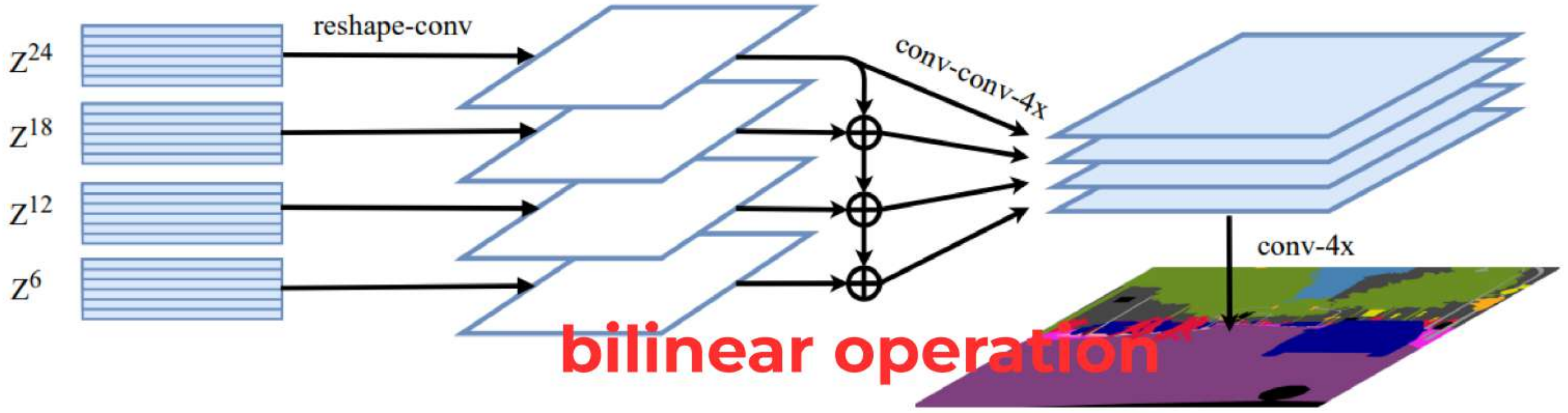



## Decoder

1. Naive - simply bilinearly upsampled
2. PUP - Progressive UPsampling




3. MLA - Multi-Level feature Aggregation

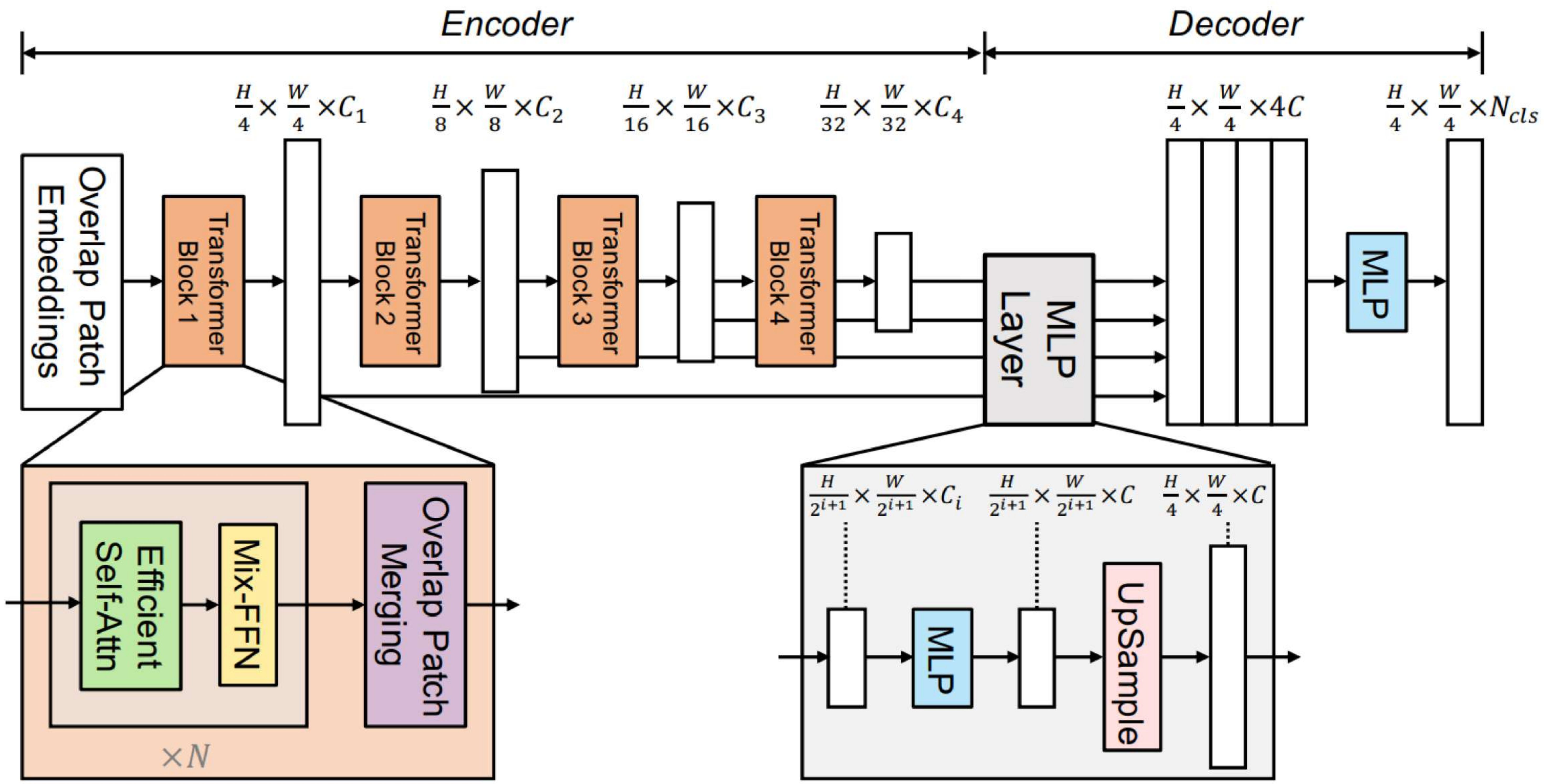


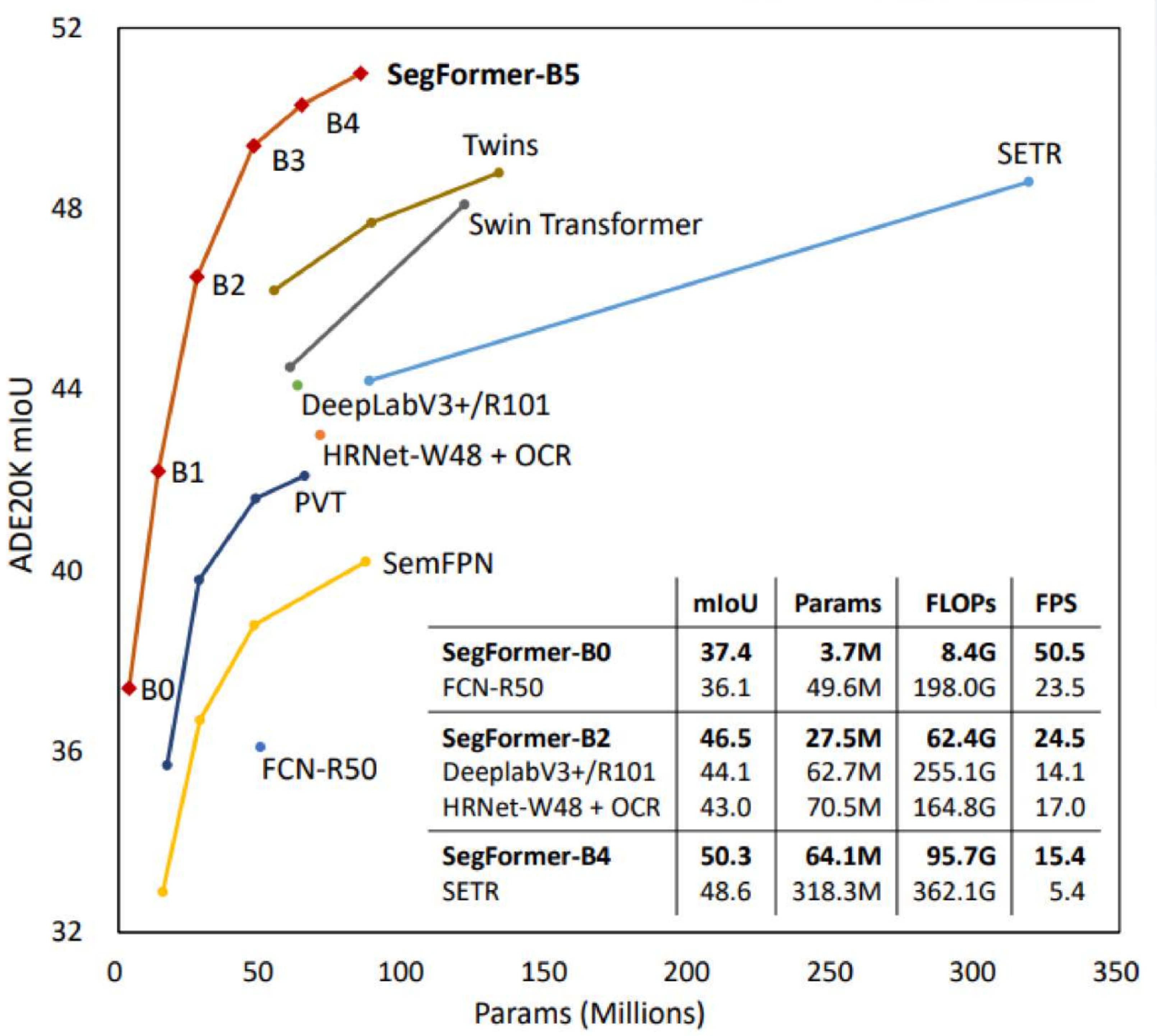
# SegFormer

## Simple and Efficient Design
## for Semantic Segmentation with Transformers

# SegFormer - Summary

## Structure

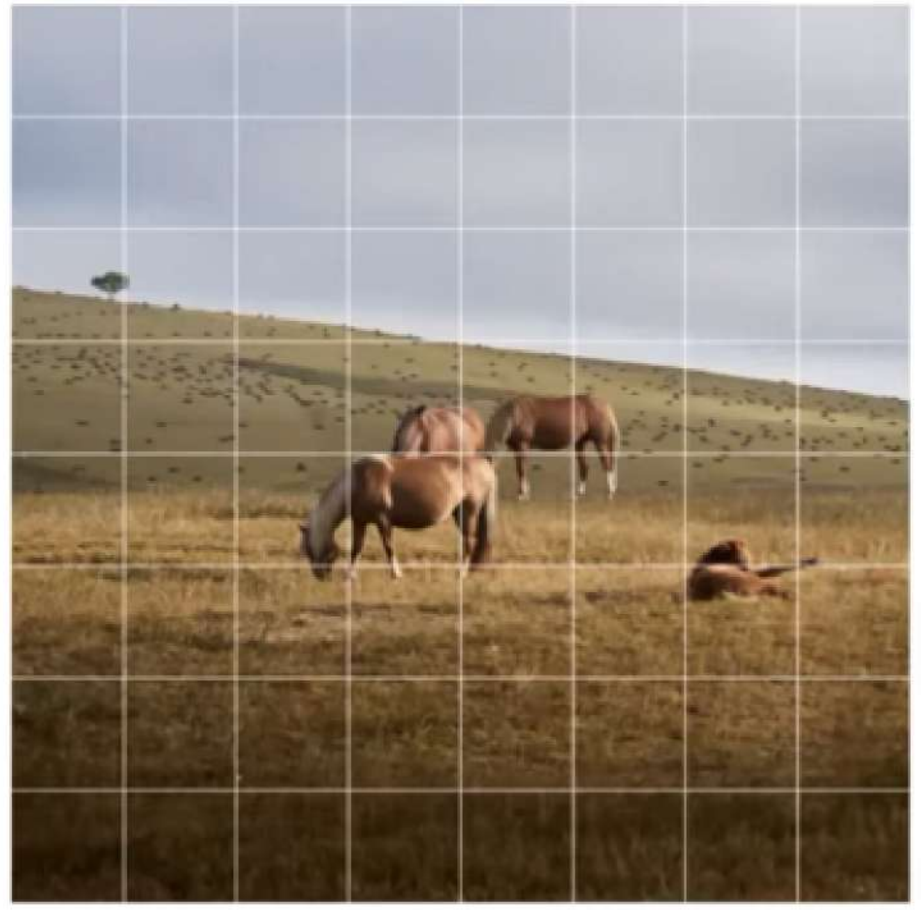


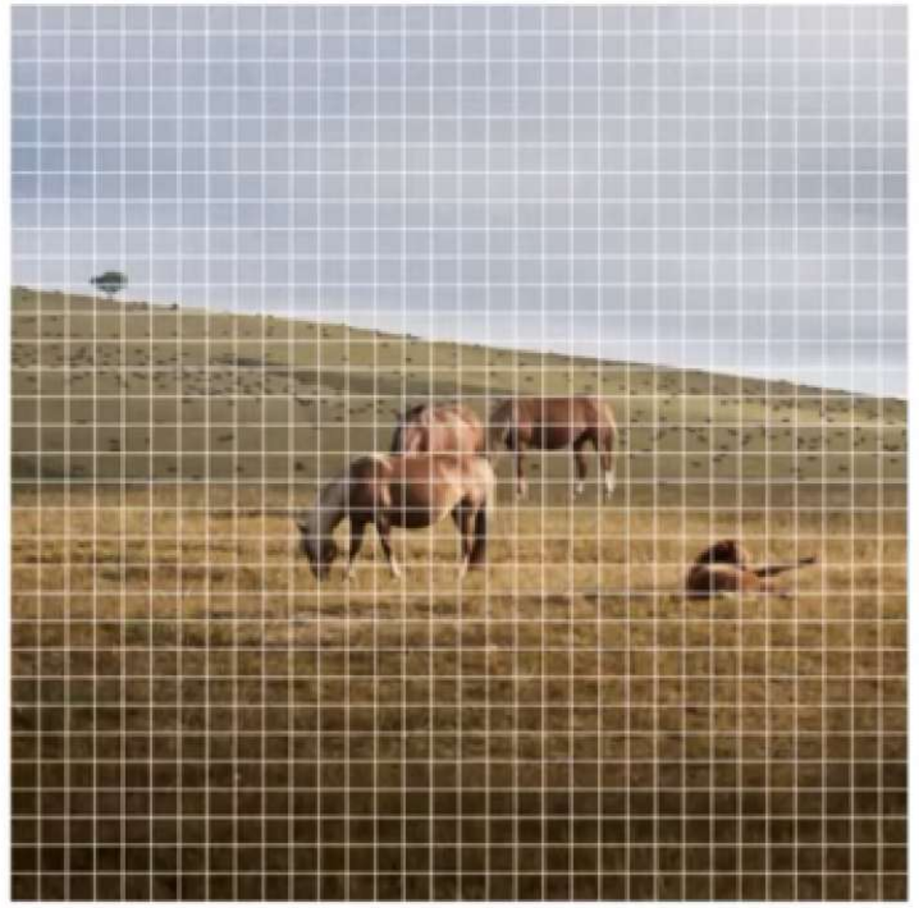

| | mIoU | Params | FLOPs | FPS |
|---|---|---|---|---|
| **SegFormer-B0** | **37.4** | **3.7M** | **8.4G** | **50.5** |
| FCN-R50 | 36.1 | 49.6M | 198.0G | 23.5 |
| **SegFormer-B2** | **46.5** | **27.5M** | **62.4G** | **24.5** |
| DeeplabV3+/R101 | 44.1 | 62.7M | 255.1G | 14.1 |
| HRNet-W48 + OCR | 43.0 | 70.5M | 164.8G | 17.0 |
| **SegFormer-B4** | **50.3** | **64.1M** | **95.7G** | **15.4** |
| SETR | 48.6 | 318.3M | 362.1G | 5.4 |

# Hierarchical Feature Representation

## Hierarchical Feature Representation



ViT
(P=16)

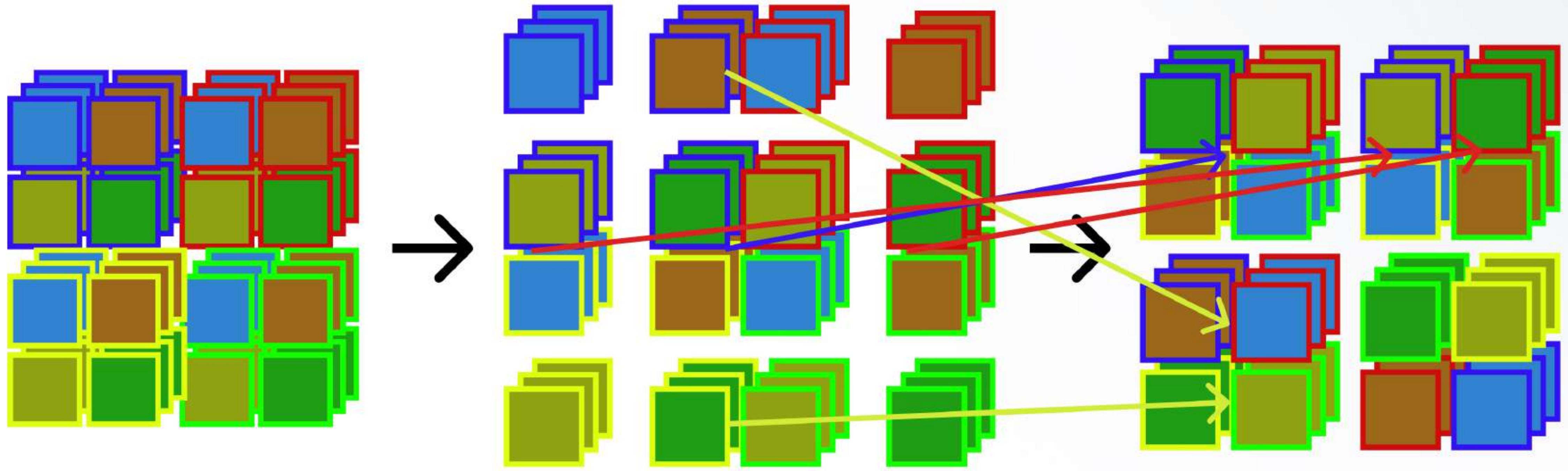


Segformer
(P=4)

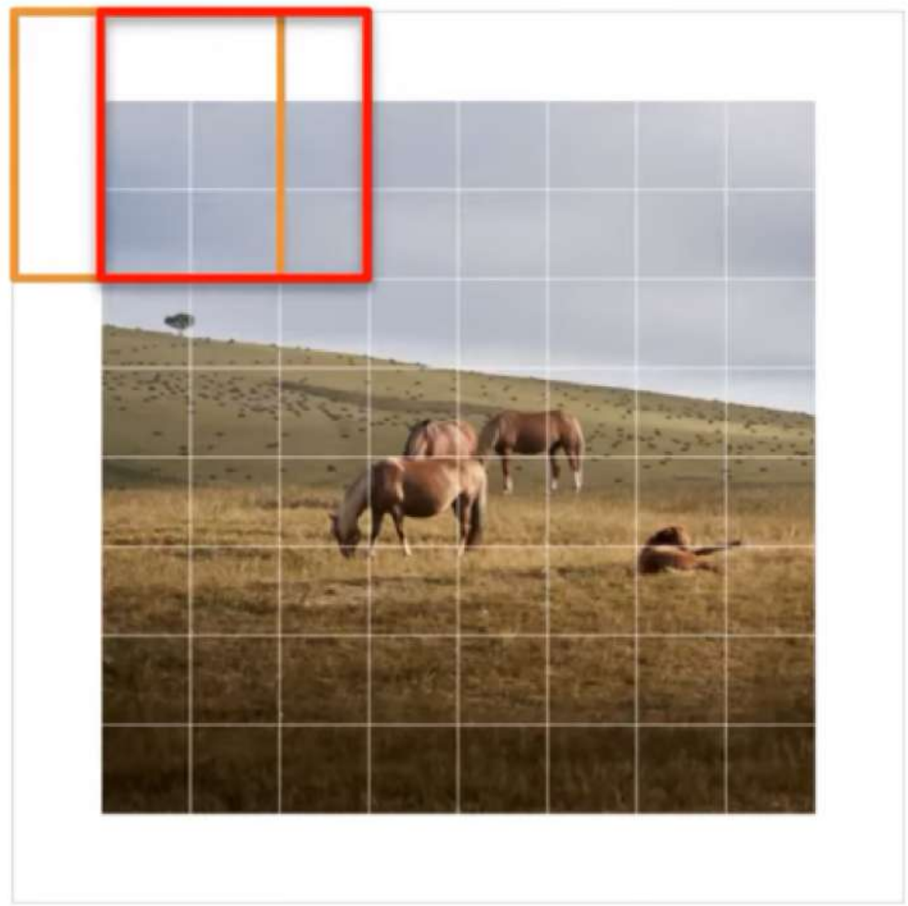
**Self Attention cost ?**

**local continuity ?**

## Shifted Window ( Swin TR )



## Overlapping Patch Window ( SegFormer )



**Similar to how CNNs work**

| | |
|---|---|
| **Stage 1** | **(K=7, S=4, P=3)** |
| **Stage 2, 3, 4** | **(K=3, S=2, P=1)** |

# Hierarchical Feature Representation

## Efficient Self-Attention

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^{\top}}{\sqrt{d_{head}}}\right)V.$$

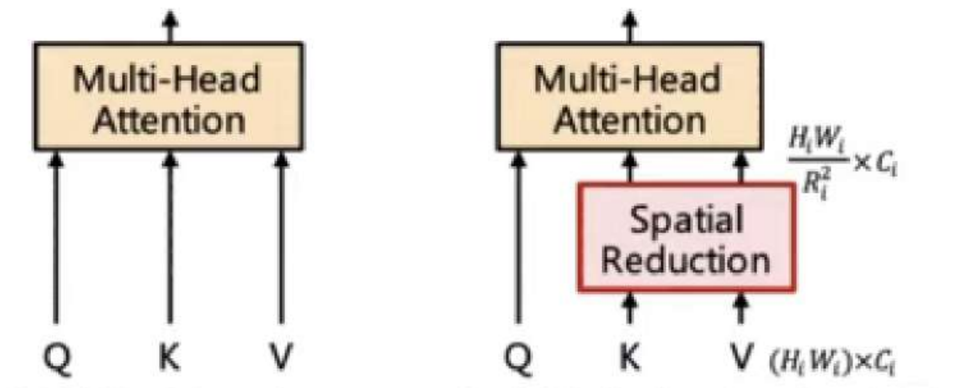
K : (N, C)

K : (N/R, C)

Sequence Reduction Process

$$\hat{K} = \text{Reshape}\left(\frac{N}{R}, C \cdot R\right)(K)$$

$$K = \text{Linear}(C \cdot R, C)(\hat{K}),$$




Multi-Head Attention

Spatial-Reduction Attention (ours)

## Mix-FFN

positional encoding $\longrightarrow$ Fixed input resolution

**The size must be matched through interpolation which causes performance degradation**

$$\mathbf{x}_{out} = \text{MLP}(\text{GELU}(\boxed{\text{Conv}_{3\times3}(\text{MLP}(\mathbf{x}_{in}))})) + \mathbf{x}_{in},$$

Patches

$$\boxed{\text{Conv}_{3\times3}(\text{MLP}(\mathbf{x}_{in}))}$$

- **ConV 3x3 layers use depth-wise convolution**
- **provide location information**

# Lightweight All-MLP Decoder

$$\hat{F}_i = \text{Linear}(C_i, C)(F_i), \forall i$$
$$\hat{F}_i = \text{Upsample}(\frac{W}{4} \times \frac{W}{4})(\hat{F}_i), \forall i$$
$$F = \text{Linear}(4C, C)(\text{Concat}(\hat{F}_i)), \forall i$$
$$M = \text{Linear}(C, N_{cls})(F),$$

**1. All channels of multi-level features are integrated equally.**

**2. Integrate the feature size to 1/4 the size of the original image.**

**3. Concatenate the features and in this process restore the channel that was multiplied by a factor of 4.**

**4. Predict the final segmentation mask. (shape: B(batch) x N(num of classes) x H/4 x W/4)**

1. Manual work and computational effort are not greatly required

2. It can have a larger effective field compared to CNN.

# Lightweight All-MLP Decoder

$$\hat{F}_i = \mathrm{Linear}(C_i, C)(F_i), \forall i$$

$$\hat{F}_i = \mathrm{Upsample}(\frac{W}{4} \times \frac{W}{4})(\hat{F}_i), \forall i$$

$$F = \mathrm{Linear}(4C, C)(\mathrm{Concat}(\hat{F}_i)), \forall i$$

$$M = \mathrm{Linear}(C, N_{cls})(F),$$

**1. All channels of multi-level features are integrated equally.**

**2. Integrate the feature size to 1/4 the size of the original image.**

**3. Concatenate the features and in this process restore the channel that was multiplied by a factor of 4.**

**4. Predict the final segmentation mask. (shape: B(batch) x N(num of classes) x H/4 x W/4)**

1. Manual work and computational effort are not greatly required

2. It can have a larger effective field compared to CNN.