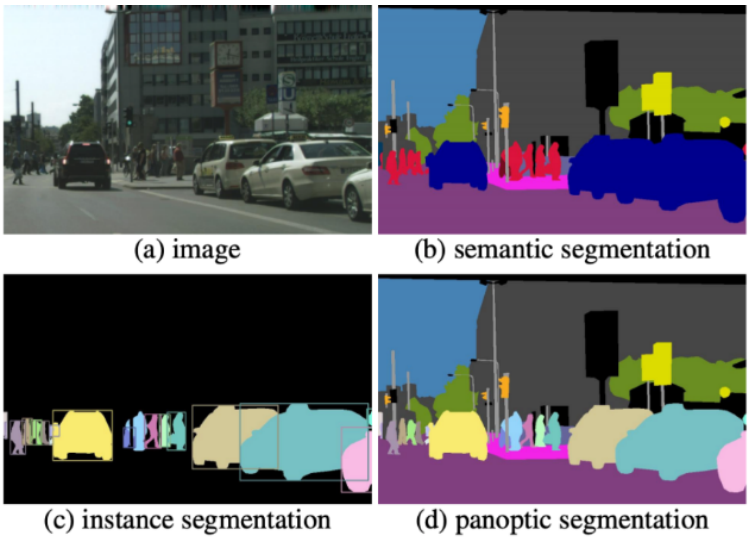
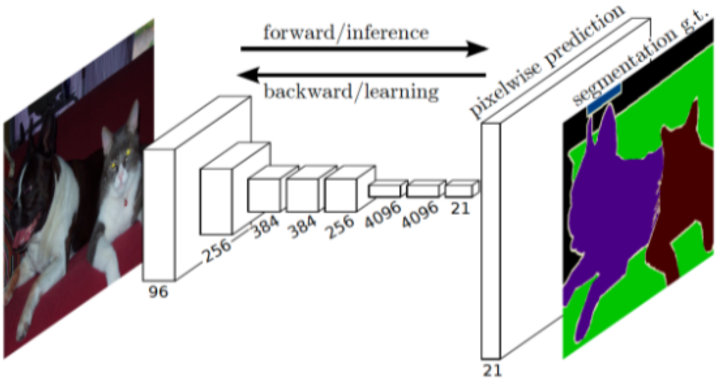
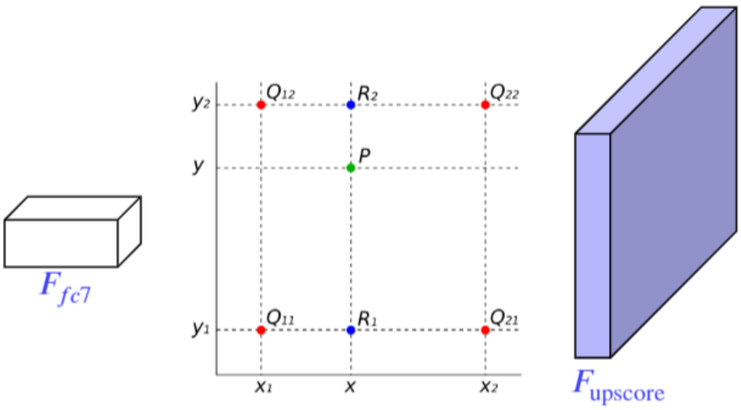
# Image Segmentation

- Brief Introduction
  - semantic segmentation
    - classification of each pixel

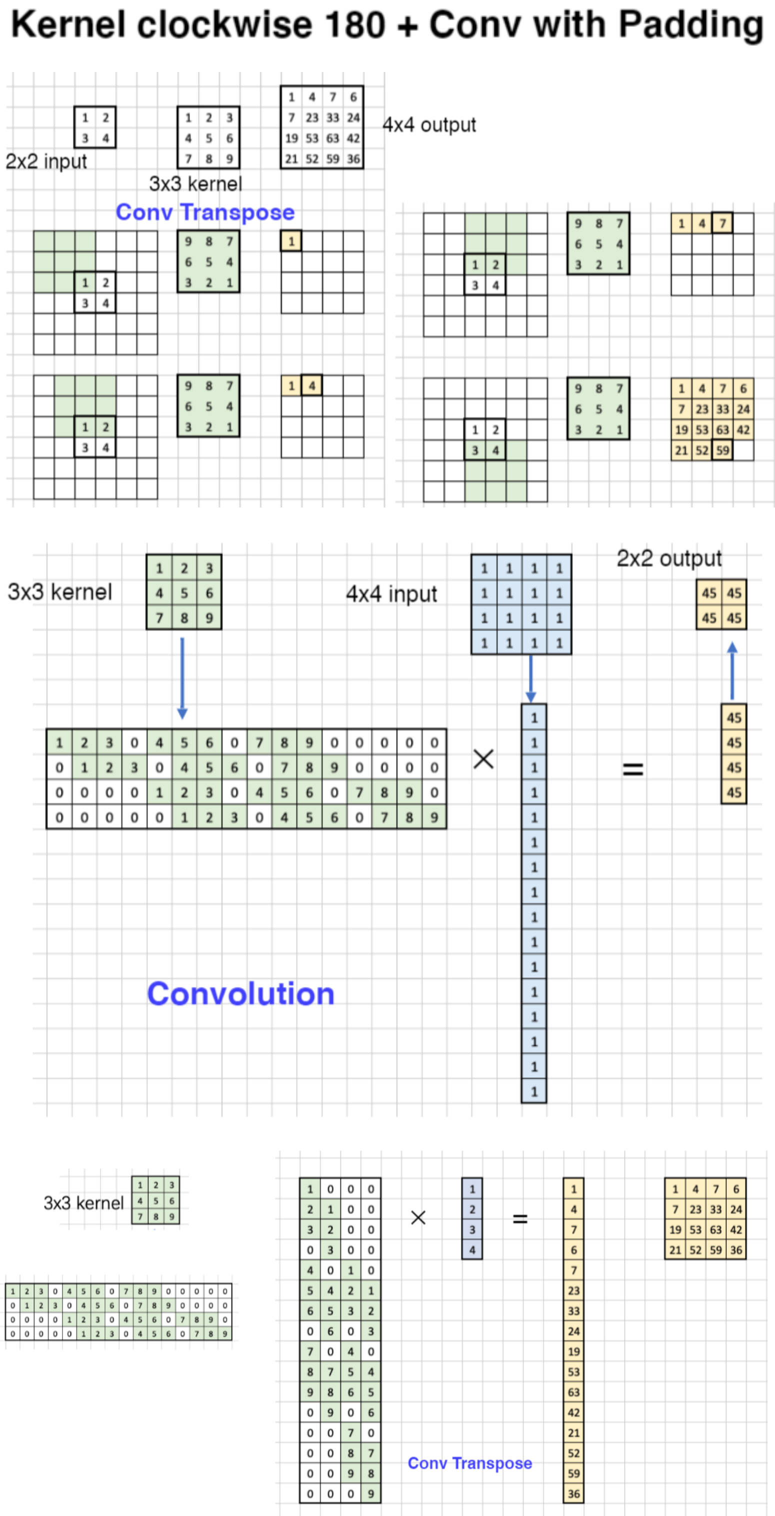
      
      
    - Pipeline
      - Input: Image (RGB)
      - Algorithm: deep learning model
      - Output: classification result (single-channel image consistent with the input size)
      - Training process:
        - input:image+label
        - forward: out=model(image)
        - Calculate the loss: loss = loss_ func (out, label)
        - Bp: Loss. behind()
        - Update weight: optimizer.minimize(loss)
    - Evaluation Method
      - mIoU: mean intersevtion-over union
      - mAcc: mean Accuracy
  - instance segmentation
    - mask each object in bbox
  - panoptic segmentation
    - mask + pixel classification




- FCN Network
  - 

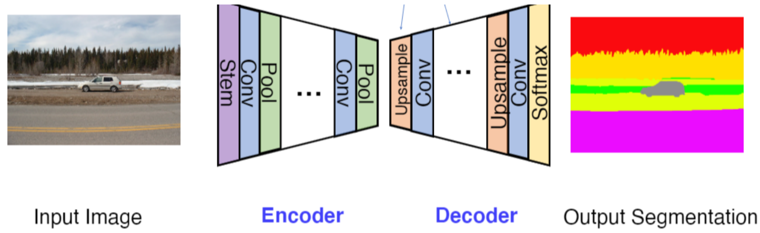
  - input size= output size
  - make feature map bigger
    - 1. up-sampling
      - resize -> bilinear interpolation
      - 

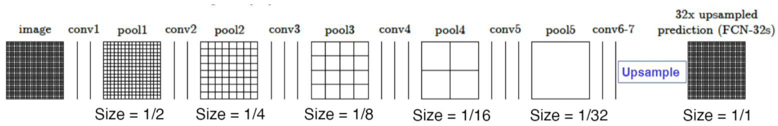

$$
\begin{aligned}
v(x,y) = {} & v(0,0) * (1-x)(1-y) + \\
& v(1,0) * x(1-y) + \\
& v(0,1) * (1-x)y + \\
& v(1,1) * xy
\end{aligned}
$$

- 2. transpose conv
  - 

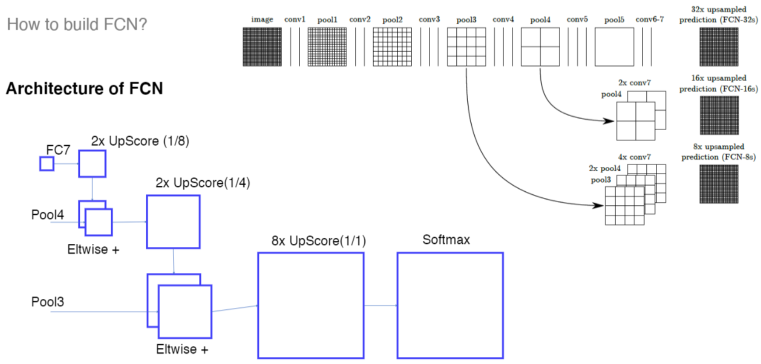
- 3. up-pooling
  - reverse of pooling

- Architecture of FCN : encoder +decoder
  - 



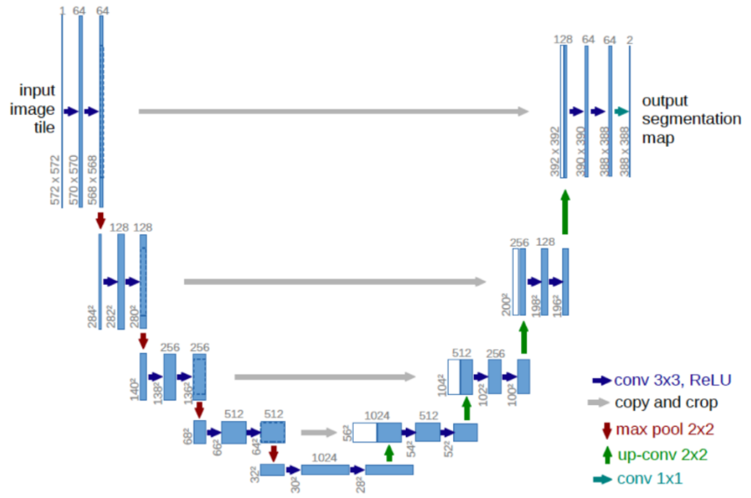



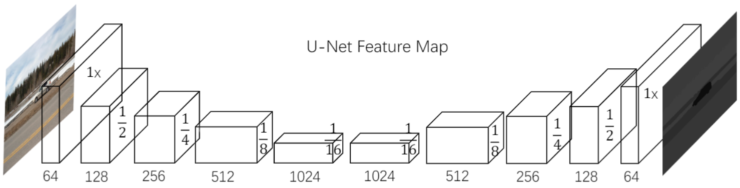

- convolution
- down-sampling
- up-sampling
- element wise add

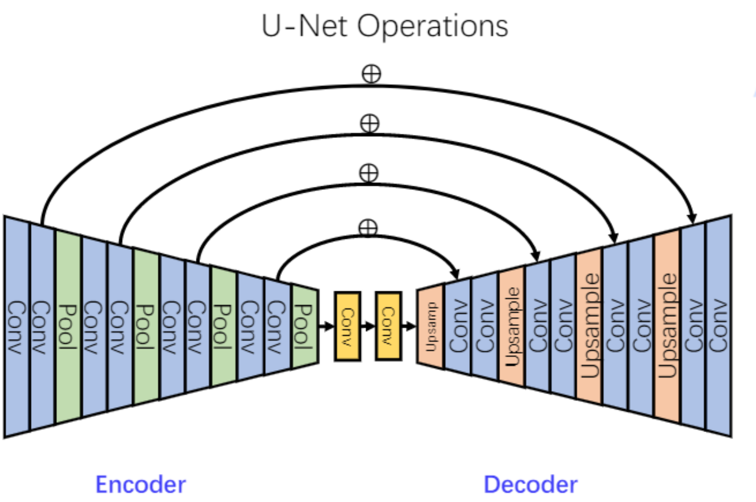



- advantage:
  - Any size input
  - High efficiency (compared to before)
  - Combine shallow information
- shortcoming:
  - The segmentation result is not fine enough
  - Contextual information is not considered

- U-Net
  - encoder+decoder

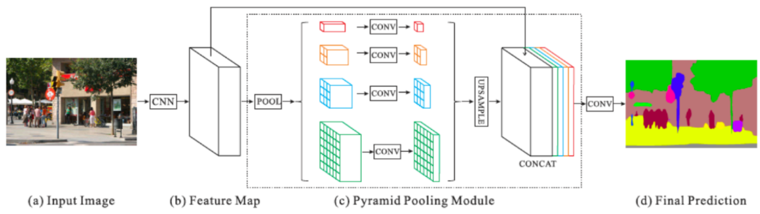
    
    

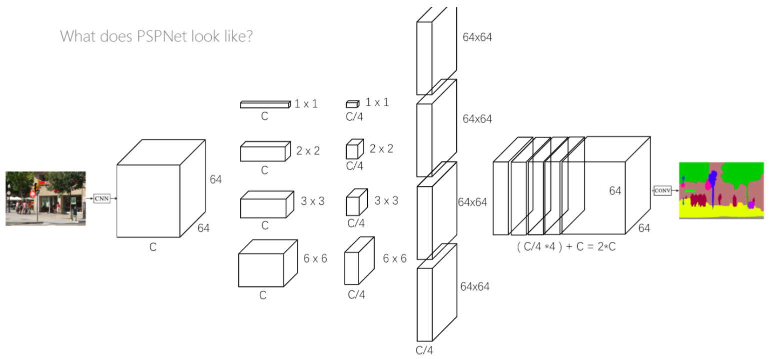
    
    

    -

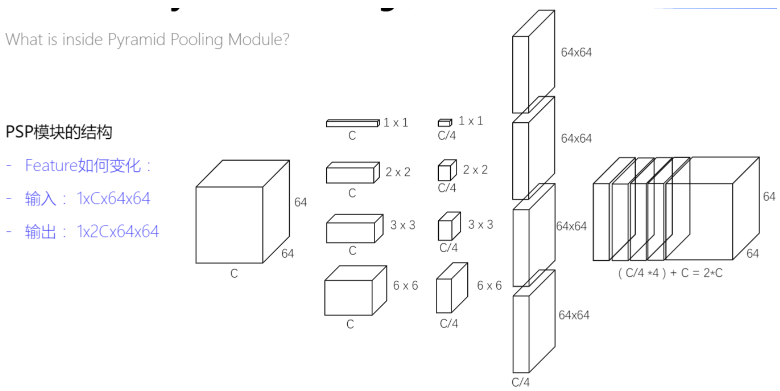
  - skip-connect
    - concatenation + crop + conv
  - main operations
    -
      > Conv 3x3, (with bn, relu)
      > Pool 2D
      > Transpose Conv 2x2
      > Crop, Concat
      > Conv 1x1,
      > SoftMax, argmax, squeeze

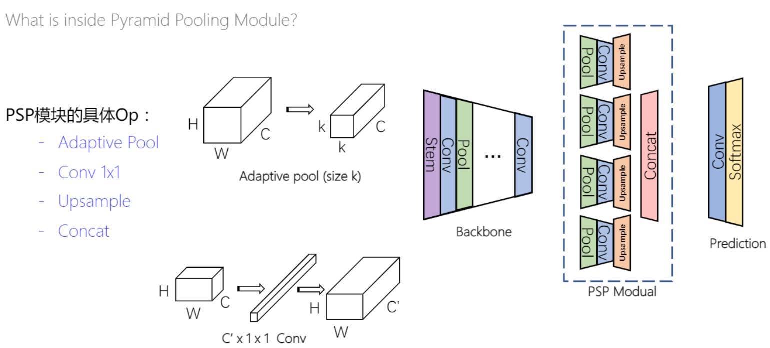
- PSP Net: Pyramid Scene Parsing Network
  - increase feature map -> increase receptive field -> consider Contextual information
  - Architecture



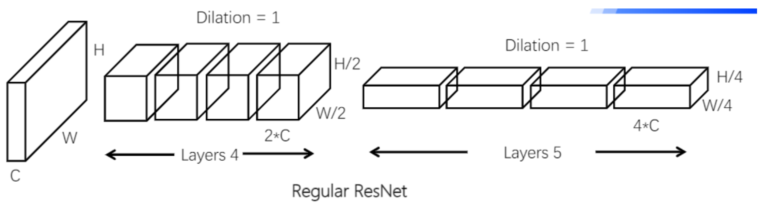



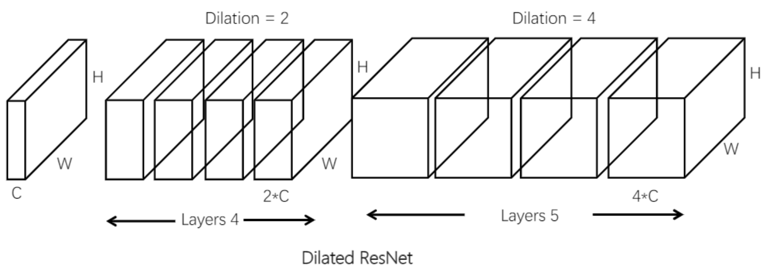

- Pyramid Pooling Module



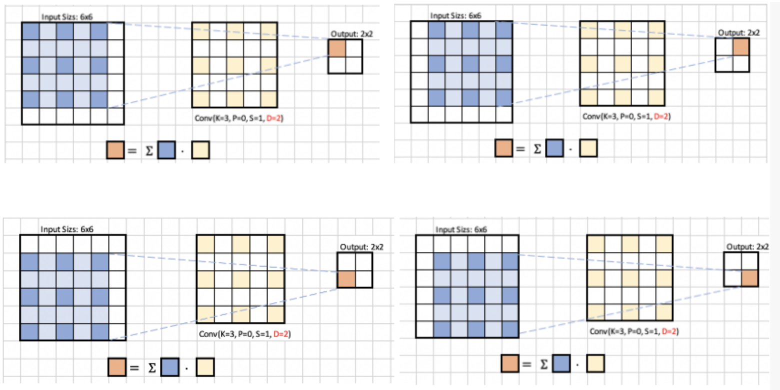




- Backbone: dilated ResNet -> increase receptive field



- dilated conv




- DeepLab
  - development

    DeepLab Series：

    - V1: Semantic image segmentation with deep convolutional nets and fully connected CRFs (ICLR 2015)
    - V2: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs (TPAMI 2018)
    - V3: Rethinking Atrous Convolution for Semantic Image Segmentation
    - V3+: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation (ECCV 2018)

  - Architecture

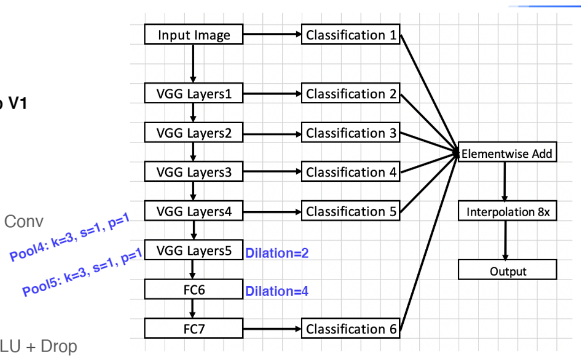
| Network | Backbone | Atrous Conv | MultiScale | Fully-connect CRF |
|---|---|---|---|---|
| DeepLab V1 | VGG-16 | Atrous Block | Training | Yes |
| DeepLab V2 | ResNet | ASPP | Training | Yes |
| DeepLab V3 | MG ResNet | Upgraded ASPP | Inference | No |
| DeepLab V3+ | Xception | ASPP + decoder | Inference | No |

  - DeepLab V1

    What is DeepLab V1?

    **Architecture of DeepLab V1**

    - DeepLab Series：
    - VGG Layers 5: Astrous Conv
    - FC6: Astrous Conv
    - FC7: Conv1x1
    - Classification X:
      - Conv (stride) + ReLU + Drop
      - Conv1x1 + ReLU + Drop
      - Conv1x1 (n_classes)

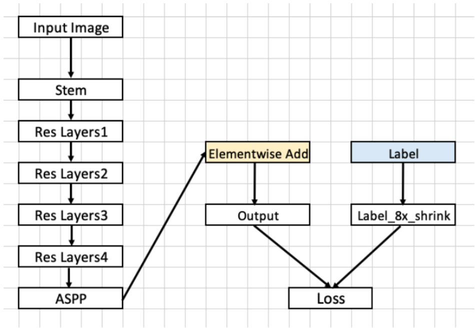



  - DeepLab V2

What is DeepLab V2?

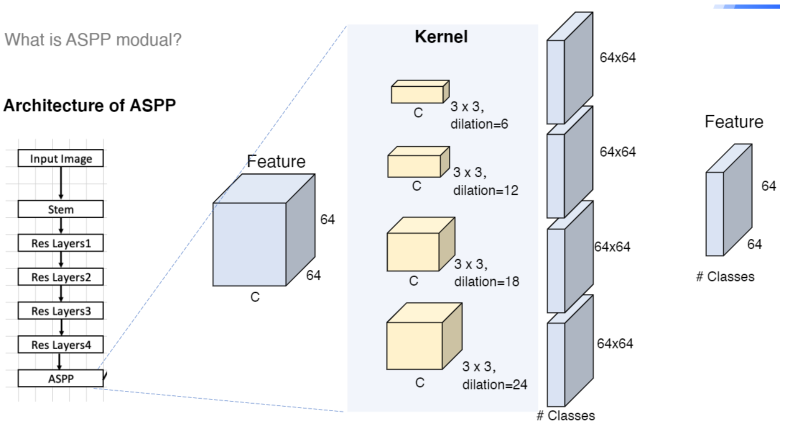
**Architecture of DeepLab V2**

- Backbone: ResNet
- Dilated Conv: ASPP Module
    - Atrous Spatial Pyramid Pooling

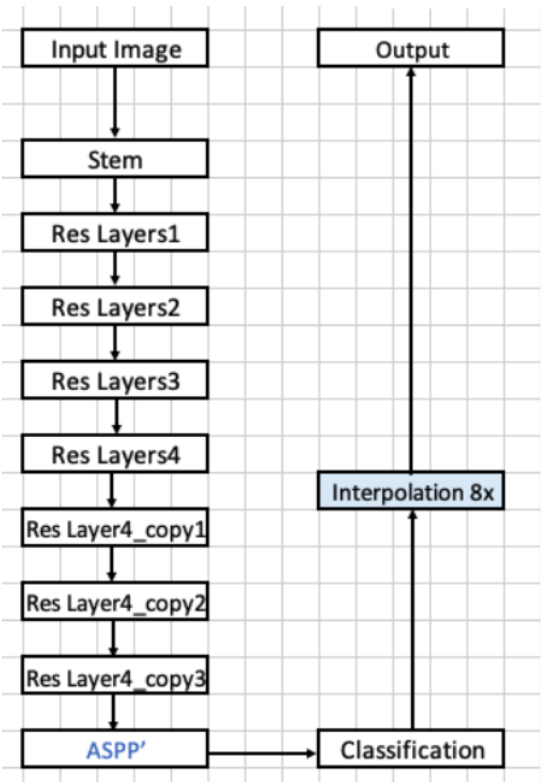



What is ASPP modual?

**Architecture of ASPP**

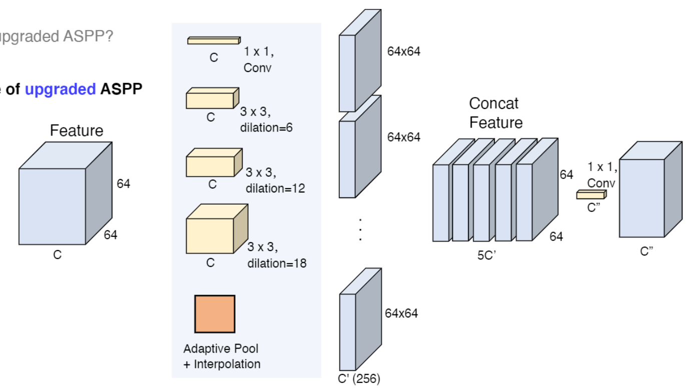



- DeepLab V3

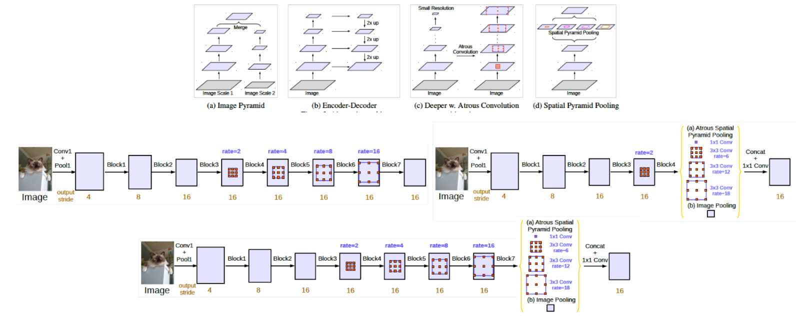


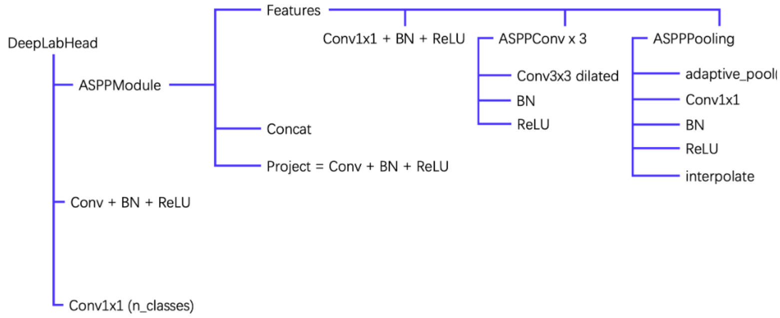

What is upgraded ASPP?

**Architecture of upgraded ASPP**

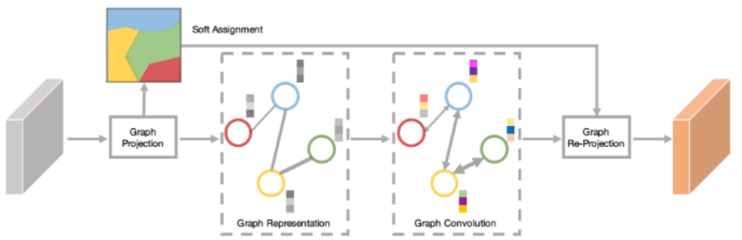


What is DeepLab V3 shown in the paper?






- Graph convolutional network
  - Graph Convolutional Unit (GCU) = Graph Projection + Graph Convolution + Graph Re-projection




    - Graph Projection Gproj: Project the 2-dimensional feature graph X into a graph G = (V, E), assign similar features to the same node, and are aggregated into nodes Characterize Z∈Rdx|V|

**图投影 (Graph Projection)**：分配特征X $= [x_1; x_2; \ldots; x_N] \in R^{N\times d}$到节点集合

计算软分配矩阵$Q \in R^{\mathrm{HW}\times|\mathcal{V}|}$ $$q_{ij}^k = \frac{\exp(-\left\|\frac{x_{ij}-\omega_k}{\sigma_k}\right\|^2/2)}{\sum_k \exp(-\left\|\frac{x_{ij}-\omega_k}{\sigma_k}\right\|^2/2)}$$

计算图表征Z $\in R^{d\times|\mathcal{V}|}$表示 $$z_k = \frac{z_k'}{\|z_k'\|}, \quad z_k' = \frac{1}{\sum_i q_{ij}^k} q_{ij}^k (x_{ij}-\omega_k)/\sigma_k$$

计算邻接矩阵$\mathcal{A} = Z^T Z$

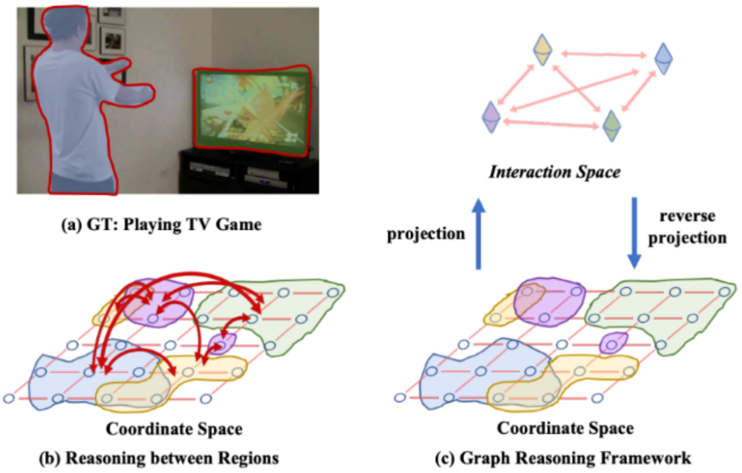
    - Graph Convolution Gcomv: Perform graph convolution on graph G. (feature propagation along the edges of the graph, modeling the global context) to get a new graph sign

$$Z' = f\left(\mathcal{A} Z^T W_g\right)$$

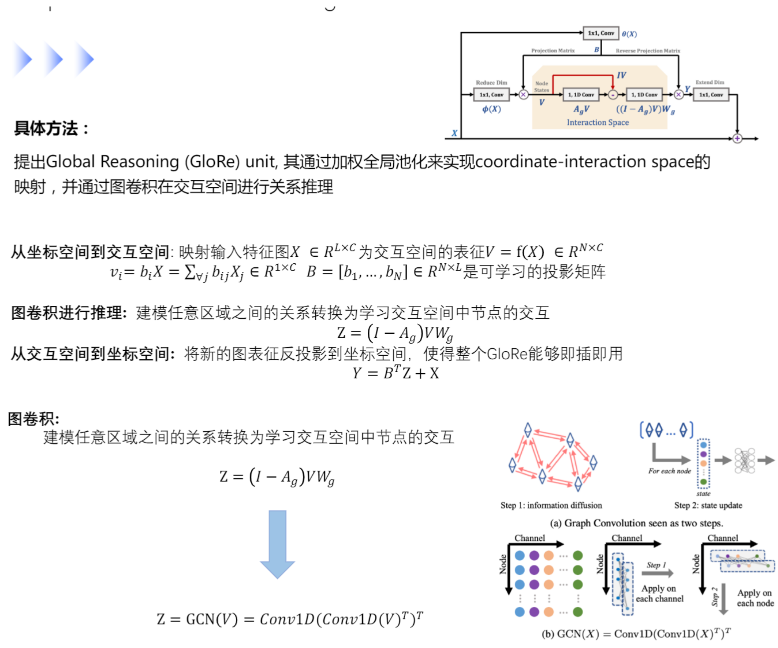
    - Graph Re-projection Greprof: Backproject the new graph representation to a 2-dimensional space, making the entire GCU plug-and-play

$$X' = QZ'^T$$

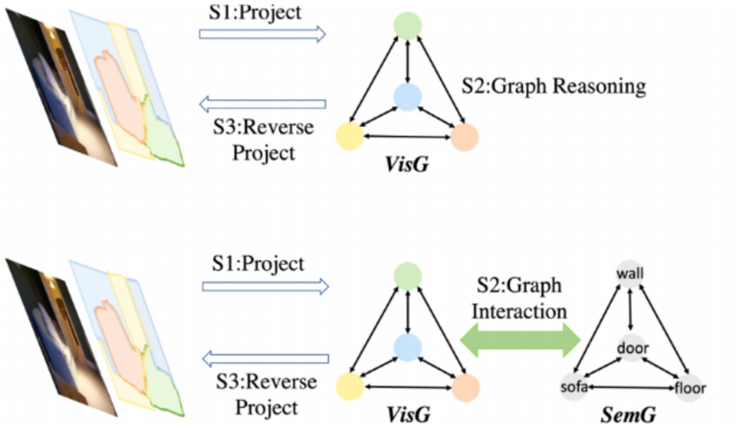
- Graph based global reasoning networks

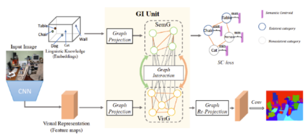



  - The pixel-level features of the coordinate space (Coordinate Space) are aggregated and projected to the Interaction Space (Interaction Space), and then effective relational reasoning is carried out. Finally, the features with relational attributes are back-projected to the original coordinates. space
  -




- GINet: Graph Interaction Network for Scene Parsing
  - A new Graph Interaction Unit (Graph Interaction Unit) is proposed, which uses semantic knowledge based on data sets to further promote the contextualization of visual graph representation.




具体方法：

Graph Interaction Unit由图构建，语义到视觉推理，

视觉到语义推理，单元输出构成

**图构建**: 映射视觉特征$X \in \mathbb{R}^{L\times C}$和每个类别语义特征$l_i \in \mathbb{R}^K(\mathrm{i} = \{0,1,\ldots,M-1\})$为视觉图$P \in \mathbb{R}^{N\times D}$和语义图$\mathrm{S} \in \mathbb{R}^{M\times D}$

视觉图：$P = ZXW$,　　$Z \in \mathbb{R}^{N\times L}$ 是投影矩阵, $W \in \mathbb{R}^{C\times D}$是特征维度变换矩阵

语义图： $\mathrm{s}_i$=MLP($l_i$)

**语义到视觉推理:** 给每个视觉图的节点表征提取对应的语义表示

$$P_o = P + \beta_{s2v} G^{s2v} SW s_{2v}$$

**视觉到语义推理:** 给每个输入样本生成基于样本的语义图表征

$$S_o = S + \beta_{v2sSGv}{}^{2s} PW_{v2s}$$

$\beta_{s2v}$ , $\beta_{v2s} \in \mathbb{R}^N$ 是可学习向量; $G^{s2v}, G^{v2s}$ 是分配矩阵; $W_{s2v}$ , $W_{v2s} \in \mathbb{R}^{\mathrm{D\times D}}$ 是可训练参数

**单元输出:** 将新的视觉图表征$P_o$投影回二维的像素级特征　　$X_o = X + ZT\, P_o W_o$