Deep Learning for Computer Vision

CNNs for Segmentation

Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Homework

Exercises

- Given two bounding boxes in an image: an upper-left box which is 2×2, and a lower-right box which is 2×3 and an overlapping region of 1×1, what is the IoU between the two boxes? $1/9$

- Consider using YOLO object detector on a 19×19 grid, on a detection problem with 20 classes, and with 5 anchor boxes. During training, for each image, you will need to construct an output volume y as the target value for the neural network; this corresponds to the last layer of the neural network. (y may include background). What is the dimension of this output volume? $19 \times 19 \times (5 \times 5 + 20) = 19 \times 19 \times 45$

Recall: Image Segmentation

- Image segmentation methods: Watershed, Graph Cut, Normalized Cut, Mean Shift, etc

NPTEL

Recall: Image Segmentation

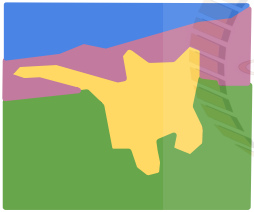
- Image segmentation methods: Watershed, Graph Cut, Normalized Cut, Mean Shift, etc
- Classical segmentation methods inspired early versions of deep learning based methods for object detection; R-CNN used a version of min-cut segmentation method known as CPMC (Constrained Parametric Min Cuts) to generate region proposals for foreground segments

NPTEL

Recall: Image Segmentation

- Image segmentation methods: Watershed, Graph Cut, Normalized Cut, Mean Shift, etc
- Classical segmentation methods inspired early versions of deep learning based methods for object detection; R-CNN used a version of min-cut segmentation method known as CPMC (Constrained Parametric Min Cuts) to generate region proposals for foreground segments
- Moving on to deep learning for segmentation now...

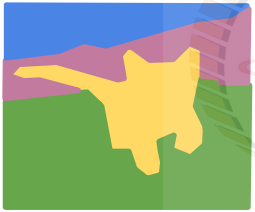
Semantic Segmentation



- Task of grouping together similar (in semantic content) pixels in an image. How to formulate this using DNNs?

NPTEL

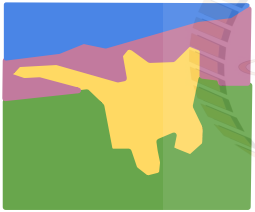
Semantic Segmentation



- Task of grouping together similar (in semantic content) pixels in an image. How to formulate this using DNNs? **Cast as a pixel classification problem!**
- For each training image, each pixel is labeled with a semantic category. Quite annotation-intensive!

NPTEL

Semantic Segmentation



- Task of grouping together similar (in semantic content) pixels in an image. How to formulate this using DNNs? **Cast as a pixel classification problem!**
- For each training image, each pixel is labeled with a semantic category. Quite annotation-intensive!
- Various architectures in recent years: FCN, SegNet, U-Net, PSP-Net, DeepLab and Mask R-CNN

Credit: Fei-Fei Li et al, CS231n, Stanford Univ

Fully Convolutional Networks for Semantic Segmentation (FCN)[1]

- Adapts various classification networks (VGG net, GoogLeNet) into fully convolutional networks by converting FC layers into 1×1 conv layers

NPTEL

[1] Shelhamer et al, Fully Convolutional Networks for Semantic Segmentation, TPAMI 2016

Fully Convolutional Networks for Semantic Segmentation (FCN)[1]

- Adapts various classification networks (VGG net, GoogLeNet) into fully convolutional networks by converting FC layers into 1×1 conv layers
- To obtain classification for each pixel, another 1×1 conv layer is appended with channel dimension $C + 1$ where C is number of classes.

NPTEL

[1] Shelhamer et al, Fully Convolutional Networks for Semantic Segmentation, TPAMI 2016

Fully Convolutional Networks for Semantic Segmentation (FCN)[1]

- Adapts various classification networks (VGG net, GoogLeNet) into fully convolutional networks by converting FC layers into 1×1 conv layers
- To obtain classification for each pixel, another 1×1 conv layer is appended with channel dimension $C + 1$ where C is number of classes.

NPTEL

[1] Shelhamer et al, Fully Convolutional Networks for Semantic Segmentation, TPAMI 2016

Fully Convolutional Networks for Semantic Segmentation (FCN)[1]

- Adapts various classification networks (VGG net, GoogLeNet) into fully convolutional networks by converting FC layers into 1×1 conv layers
- To obtain classification for each pixel, another 1×1 conv layer is appended with channel dimension $C + 1$ where C is number of classes. Do you see any problem?

NPTEL

[1] Shelhamer et al, Fully Convolutional Networks for Semantic Segmentation, TPAMI 2016

Fully Convolutional Networks for Semantic Segmentation (FCN)[1]

- Adapts various classification networks (VGG net, GoogLeNet) into fully convolutional networks by converting FC layers into 1×1 conv layers
- To obtain classification for each pixel, another 1×1 conv layer is appended with channel dimension $C + 1$ where C is number of classes. Do you see any problem?
- Image classification architectures perform downsampling as they go deeper $\implies$ fully convolutional architecture will have lower resolution than input. What to do?

[1] Shelhamer et al, Fully Convolutional Networks for Semantic Segmentation, TPAMI 2016

Fully Convolutional Networks for Semantic Segmentation (FCN)[1]

- Adapts various classification networks (VGG net, GoogLeNet) into fully convolutional networks by converting FC layers into 1×1 conv layers
- To obtain classification for each pixel, another 1×1 conv layer is appended with channel dimension $C + 1$ where C is number of classes. Do you see any problem?
- Image classification architectures perform downsampling as they go deeper $\implies$ fully convolutional architecture will have lower resolution than input. What to do?
- Perform upsampling to get back to original resolution. How to do?

[1] Shelhamer et al, Fully Convolutional Networks for Semantic Segmentation, TPAMI 2016

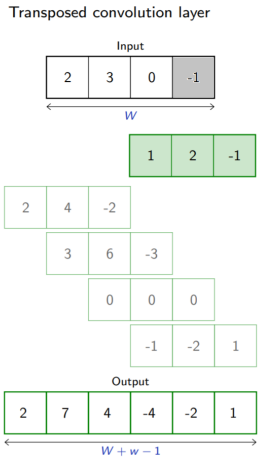
Fully Convolutional Networks for Semantic Segmentation (FCN)[1]

- Adapts various classification networks (VGG net, GoogLeNet) into fully convolutional networks by converting FC layers into 1×1 conv layers
- To obtain classification for each pixel, another 1×1 conv layer is appended with channel dimension $C + 1$ where C is number of classes. Do you see any problem?
- Image classification architectures perform downsampling as they go deeper $\implies$ fully convolutional architecture will have lower resolution than input. What to do?
- Perform upsampling to get back to original resolution. How to do?
- Learnable upsampling done through **Transpose Convolution**

[1] Shelhamer et al, Fully Convolutional Networks for Semantic Segmentation, TPAMI 2016

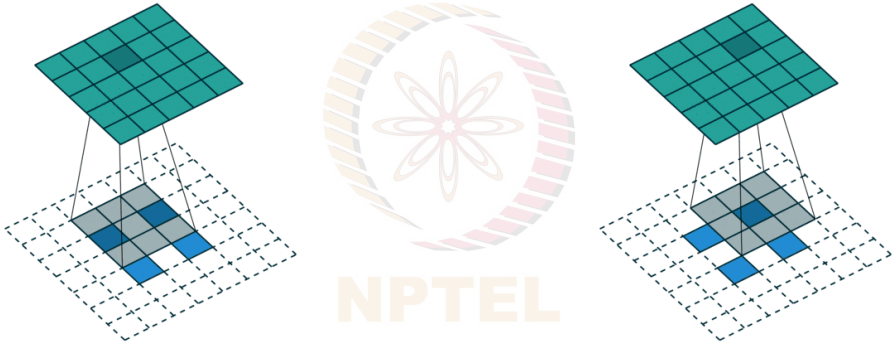
Recall: Transpose Convolution

- Allows for learnable upsampling
- Also known as Deconvolution (bad) or Upconvolution
- Traditionally, we could achieve upsampling through interpolation or similar rules
- Why not allow the network to learn the rules by itself?
- Let us see a 1D example



Credit: Francois Fleuret

Recall: Transpose Convolution

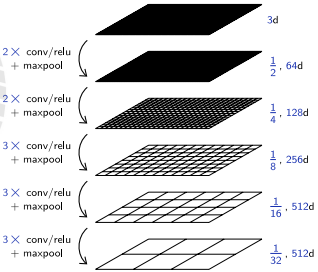


Upsampling 2×2 input to a 5×5 output

Credit: Vincent Dumoulin

FCN with VGG-16 Backbone

- Remove fully connected layers

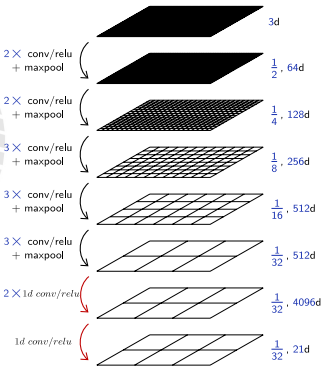


Credit: Francois Fleuret

NPTEL

FCN with VGG-16 Backbone

- Remove fully connected layers
- Replace three FC layers with 1D conv layers; last layer has $C + 1$ filters which give class probabilities (note that spatial size is downsampled because of maxpool operations)

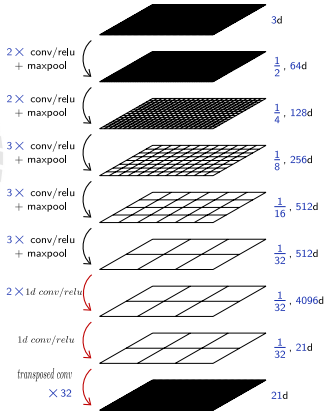


Credit: Francois Fleuret

NPTEL

FCN with VGG-16 Backbone

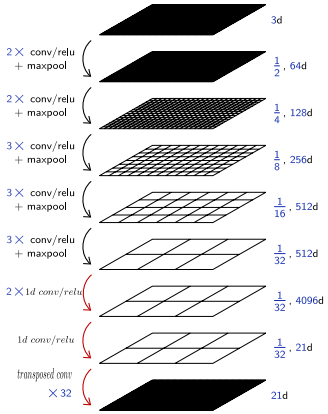
- Remove fully connected layers
- Replace three FC layers with 1D conv layers; last layer has $C + 1$ filters which give class probabilities (note that spatial size is downsampled because of maxpool operations)
- One final upsampling layer to get back to original size



Credit: Francois Fleuret

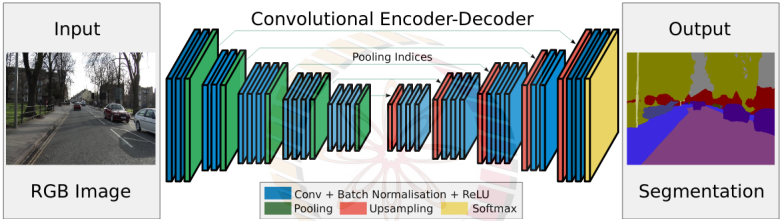
FCN with VGG-16 Backbone

- Remove fully connected layers
- Replace three FC layers with 1D conv layers; last layer has $C + 1$ filters which give class probabilities (note that spatial size is downsampled because of maxpool operations)
- One final upsampling layer to get back to original size
- Instead of one final upsampling layer, can have skip connections from earlier layers (we will see later how) for better boundaries



Credit: Francois Fleuret

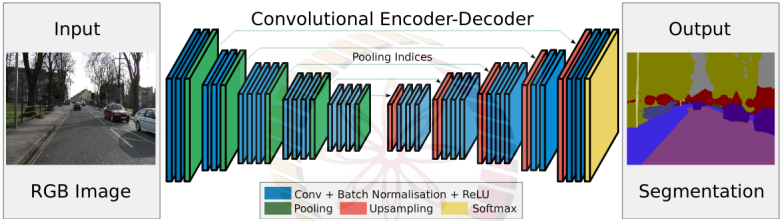
SegNet[2]



- Fully convolutional encoder-decoder architecture

[2]Badrinarayanan et al, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, TPAMI 2017

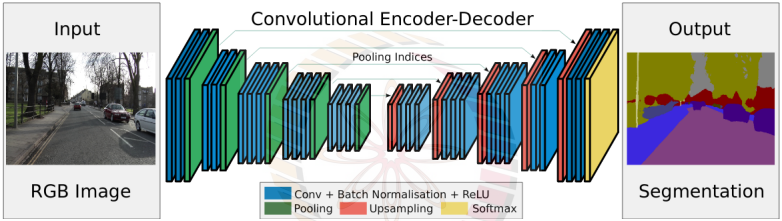
SegNet[2]



- Fully convolutional encoder-decoder architecture
- Encoder is VGG-16 without FC layers

[2]Badrinarayanan et al, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, TPAMI 2017

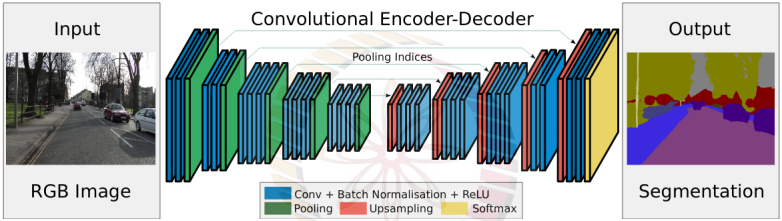
SegNet[2]



- Fully convolutional encoder-decoder architecture
- Encoder is VGG-16 without FC layers
- Decoder maps low-resolution encoder feature maps to input resolution

[2]Badrinarayanan et al, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, TPAMI 2017

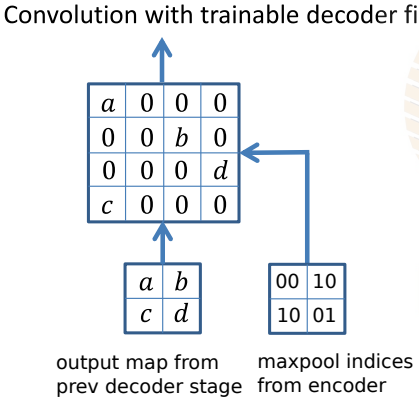
SegNet[2]



- Fully convolutional encoder-decoder architecture
- Encoder is VGG-16 without FC layers
- Decoder maps low-resolution encoder feature maps to input resolution
- Decoder uses pooling indices of corresponding max-pooling steps to perform upsampling! These sparse upsampled maps are followed by conv layers to produce dense feature maps

[2]Badrinarayanan et al, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, TPAMI 2017

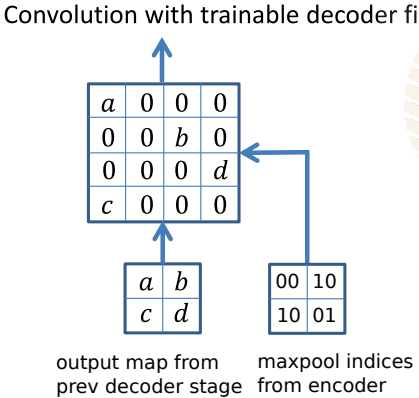
Upsampling in SegNet



- **Max pool indices:** location of maximum feature value for each pooling window for each encoder feature map

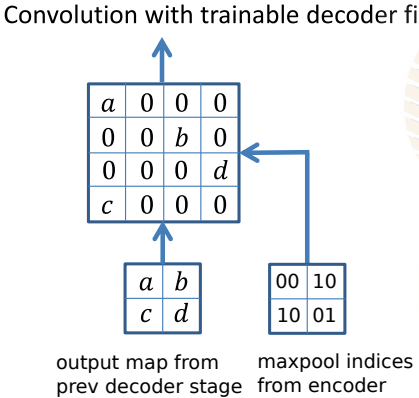
NPTEL

Upsampling in SegNet



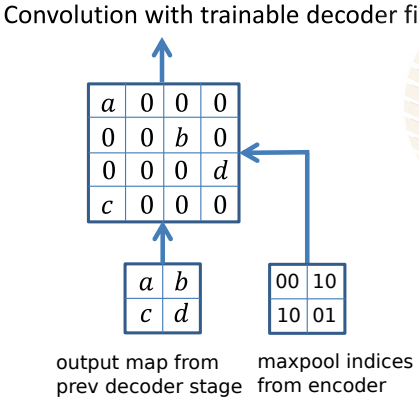
- **Max pool indices:** location of maximum feature value for each pooling window for each encoder feature map
- Storing maxpool indices is memory-efficient (instead of storing full feature map)

Upsampling in SegNet



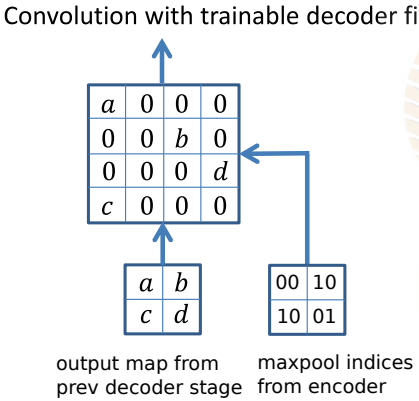
- **Max pool indices:** location of maximum feature value for each pooling window for each encoder feature map
- Storing maxpool indices is memory-efficient (instead of storing full feature map)
- In each stage of decoder, max pooling indices from corresponding encoder stage used to produce sparse upsampled feature maps

Upsampling in SegNet



- **Max pool indices:** location of maximum feature value for each pooling window for each encoder feature map
- Storing maxpool indices is memory-efficient (instead of storing full feature map)
- In each stage of decoder, max pooling indices from corresponding encoder stage used to produce sparse upsampled feature maps
- What is the loss function in all these methods?

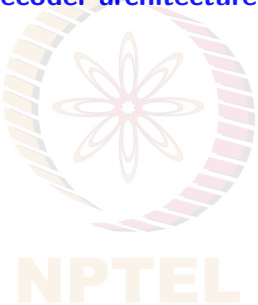
Upsampling in SegNet



- **Max pool indices:** location of maximum feature value for each pooling window for each encoder feature map
- Storing maxpool indices is memory-efficient (instead of storing full feature map)
- In each stage of decoder, max pooling indices from corresponding encoder stage used to produce sparse upsampled feature maps
- What is the loss function in all these methods? **Sum of pixel-wise cross-entropy losses**

U-Net[3]

- **Fully convolutional encoder-decoder architecture with skip connections** (an extension of FCN)



[3]Ronneberger et al, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

U-Net[3]

- **Fully convolutional encoder-decoder architecture with skip connections** (an extension of FCN)
- Contracting path is an existing classification network with FC layers removed

NPTEL

[3]Ronneberger et al, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

U-Net[3]

- **Fully convolutional encoder-decoder architecture with skip connections** (an extension of FCN)
- Contracting path is an existing classification network with FC layers removed
- Each step in expanding path consists of upsampling of feature maps, concatenated with corresponding feature maps of contracting path followed by further conv layers

NPTEL

[3]Ronneberger et al, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

U-Net[3]

- **Fully convolutional encoder-decoder architecture with skip connections** (an extension of FCN)
- Contracting path is an existing classification network with FC layers removed
- Each step in expanding path consists of upsampling of feature maps, concatenated with corresponding feature maps of contracting path followed by further conv layers
- Final layer is 1×1 conv with $C + 1$ channels, C being number of classes

NPTEL

[3]Ronneberger et al, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

U-Net[3]

- **Fully convolutional encoder-decoder architecture with skip connections** (an extension of FCN)
- Contracting path is an existing classification network with FC layers removed
- Each step in expanding path consists of upsampling of feature maps, concatenated with corresponding feature maps of contracting path followed by further conv layers
- Final layer is 1×1 conv with $C + 1$ channels, C being number of classes
- Upsampling performed by 2×2 transpose conv with $s = 2, p = 0$ with number of feature channels being halved each time (this operation doubles input map dimensions)

[3] Ronneberger et al, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

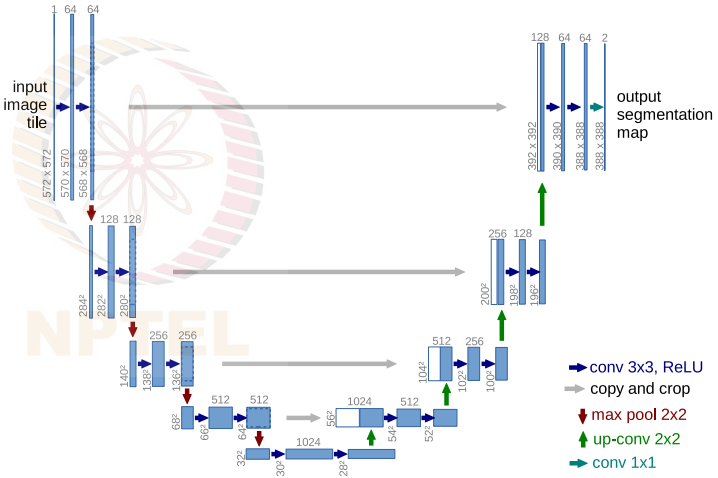
U-Net[3]

- **Fully convolutional encoder-decoder architecture with skip connections** (an extension of FCN)
- Contracting path is an existing classification network with FC layers removed
- Each step in expanding path consists of upsampling of feature maps, concatenated with corresponding feature maps of contracting path followed by further conv layers
- Final layer is 1×1 conv with $C + 1$ channels, C being number of classes
- Upsampling performed by 2×2 transpose conv with $s = 2, p = 0$ with number of feature channels being halved each time (this operation doubles input map dimensions)
- In original architecture, unpadded convolutions are performed, making output segmentation map smaller than input by a constant border width

[3] Ronneberger et al, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

U-Net Architecture

- One modification from FCN is: upsampling part has large number of feature channels
- Concatenation of feature maps from encoder gives localization information to decoder
- Other ways of concatenation include simply summing up feature maps



U-Net: Other Features and Extensions

- Designed for biomedical image segmentation; How does this affect the model learning?

NPTEL

U-Net: Other Features and Extensions

- Designed for biomedical image segmentation; How does this affect the model learning?
- Very few training images available, hence excessive data augmentation is performed by applying elastic deformations to available training images

NPTEL

U-Net: Other Features and Extensions

- Designed for biomedical image segmentation; How does this affect the model learning?
- Very few training images available, hence excessive data augmentation is performed by applying elastic deformations to available training images
- Another challenge in medical domain: separation of touching objects of same class; hence, U-Net proposes a weighted loss to penalize pixels closer to edges

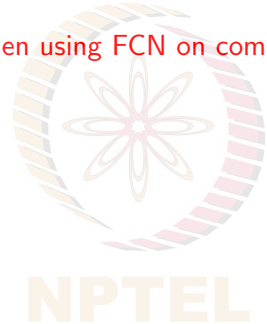
NPTEL

U-Net: Other Features and Extensions

- Designed for biomedical image segmentation; How does this affect the model learning?
- Very few training images available, hence excessive data augmentation is performed by applying elastic deformations to available training images
- Another challenge in medical domain: separation of touching objects of same class; hence, U-Net proposes a weighted loss to penalize pixels closer to edges
- Variants of U-Net: Same principles but with different kinds of blocks (dense blocks instead of conv blocks), depths, etc

PSPNet: Pyramid Scene Parsing Network[4]

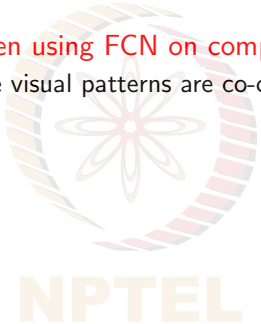
- What challenges could arise when using FCN on complex scenes?



[4] Zhao et al, Pyramid Scene Parsing Network, CVPR 2017

PSPNet: Pyramid Scene Parsing Network[4]

- What challenges could arise when using FCN on complex scenes?
  - FCN may not learn that some visual patterns are co-occurent; E.g. cars are on roads, while boats are over rivers



[4] Zhao et al, Pyramid Scene Parsing Network, CVPR 2017

PSPNet: Pyramid Scene Parsing Network[4]

- What challenges could arise when using FCN on complex scenes?
  - FCN may not learn that some visual patterns are co-occurent; E.g. cars are on roads, while boats are over rivers
  - FCN may predict parts of an object as different categories; E.g. parts of a skyscraper may be mis-classified as different buildings

NPTEL

[4] Zhao et al, Pyramid Scene Parsing Network, CVPR 2017

PSPNet: Pyramid Scene Parsing Network[4]

- What challenges could arise when using FCN on complex scenes?
  - FCN may not learn that some visual patterns are co-occurent; E.g. cars are on roads, while boats are over rivers
  - FCN may predict parts of an object as different categories; E.g. parts of a skyscraper may be mis-classified as different buildings
  - FCN may fail to correctly segment small objects and big objects (relative to its receptive field)

NPTEL

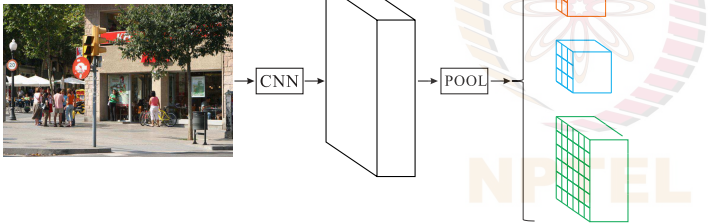
[4] Zhao et al, Pyramid Scene Parsing Network, CVPR 2017

PSPNet: Pyramid Scene Parsing Network[4]

- What challenges could arise when using FCN on complex scenes?
  - FCN may not learn that some visual patterns are co-occurent; E.g. cars are on roads, while boats are over rivers
  - FCN may predict parts of an object as different categories; E.g. parts of a skyscraper may be mis-classified as different buildings
  - FCN may fail to correctly segment small objects and big objects (relative to its receptive field)
- **Hypothesis:** Using global context information can lead to better segmentation; a network with suitable global scene-level information at different scales can improve segmentation

[4] Zhao et al, Pyramid Scene Parsing Network, CVPR 2017

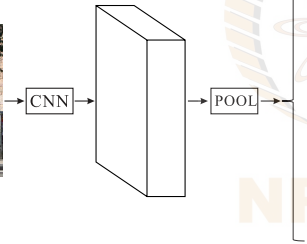
PSPNet: Pyramid Pooling Module



Average pooling done for four different scales

- Take final layer feature map of a deep network like ResNet

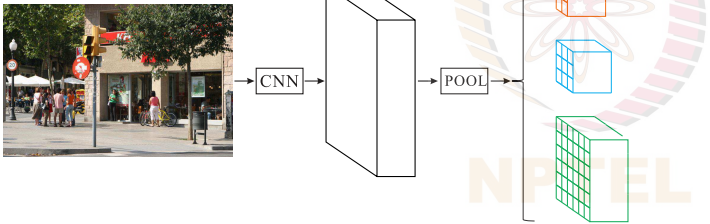
PSPNet: Pyramid Pooling Module



Average pooling done for four different scales

- Take final layer feature map of a deep network like ResNet
- **Pyramid Pooling Module** combines features in four different scales:

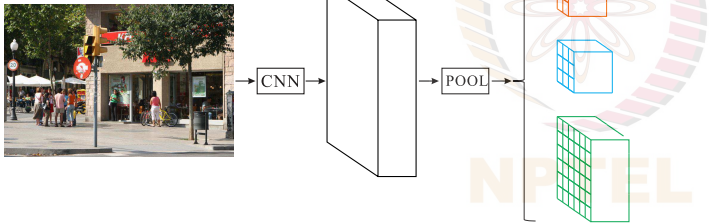
PSPNet: Pyramid Pooling Module



Average pooling done for four different scales

- Take final layer feature map of a deep network like ResNet
- **Pyramid Pooling Module** combines features in four different scales:
  - Coarsest level is simply global average pooling

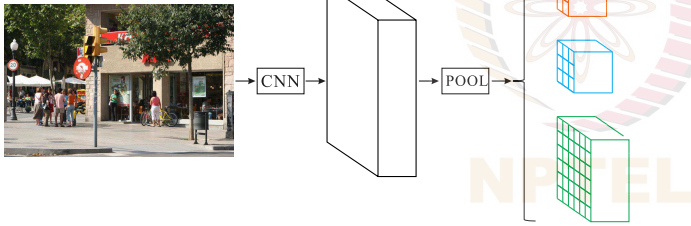
PSPNet: Pyramid Pooling Module



Average pooling done for four different scales

- Take final layer feature map of a deep network like ResNet
- **Pyramid Pooling Module** combines features in four different scales:
  - Coarsest level is simply global average pooling
  - Each successive pooling level gives increased localization information

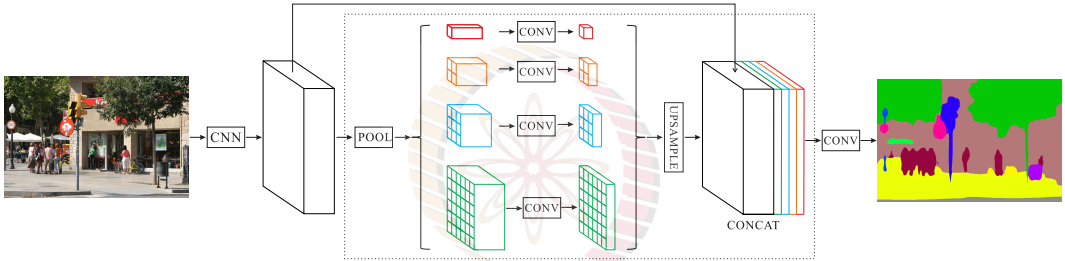
PSPNet: Pyramid Pooling Module



Average pooling done for four different scales

- Take final layer feature map of a deep network like ResNet
- **Pyramid Pooling Module** combines features in four different scales:
  - Coarsest level is simply global average pooling
  - Each successive pooling level gives increased localization information
- Average pooled outputs are thus $1 \times 1 \times c$, $2 \times 2 \times c$, $3 \times 3 \times c$, $6 \times 6 \times c$ where c is number of input channels

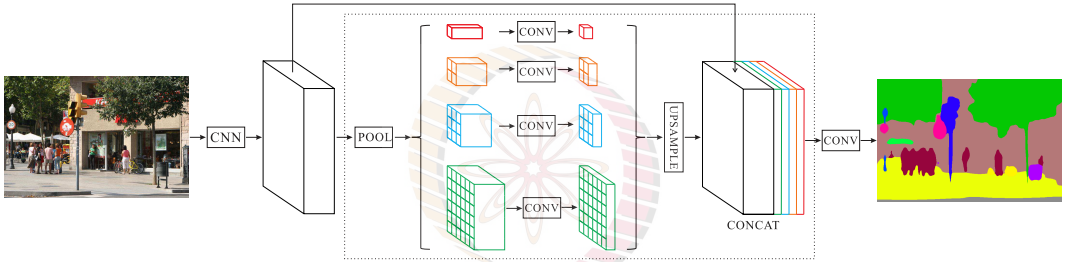
Pyramid Pooling Module



(a) Input Image (b) Feature Map (c) Pyramid Pooling Module (d) Final Prediction

- Following average pooling, a 1×1 conv layer is used to reduce number of channels at each scale; 4 scales used $\implies$ each level's channels reduced to $c/4$

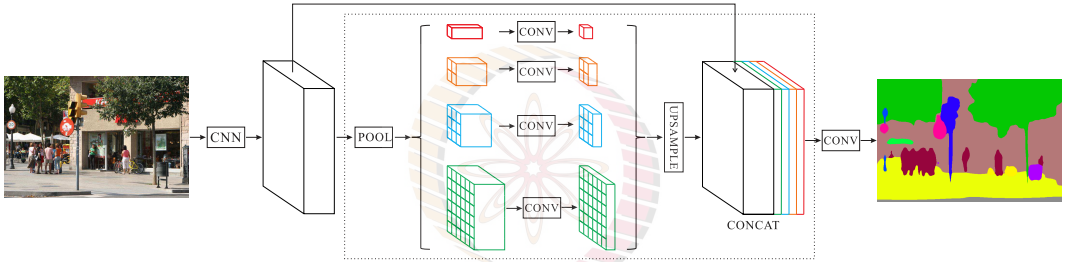
Pyramid Pooling Module



(a) Input Image (b) Feature Map (c) Pyramid Pooling Module (d) Final Prediction

- Following average pooling, a 1×1 conv layer is used to reduce number of channels at each scale; 4 scales used $\implies$ each level's channels reduced to $c/4$
- Low-dimension pooled maps are upsampled (can use bilinear interpolation) to same size as original

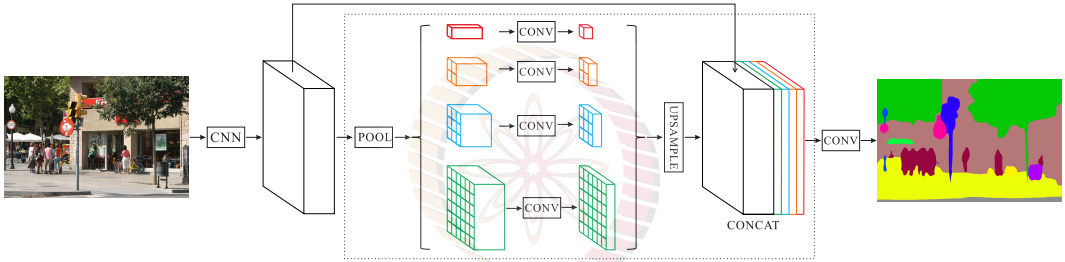
Pyramid Pooling Module



(a) Input Image (b) Feature Map (c) Pyramid Pooling Module (d) Final Prediction

- Following average pooling, a 1×1 conv layer is used to reduce number of channels at each scale; 4 scales used $\implies$ each level's channels reduced to $c/4$
- Low-dimension pooled maps are upsampled (can use bilinear interpolation) to same size as original
- Features then concatenated with original feature map

Pyramid Pooling Module



(a) Input Image (b) Feature Map (c) Pyramid Pooling Module (d) Final Prediction

- Following average pooling, a 1×1 conv layer is used to reduce number of channels at each scale; 4 scales used $\implies$ each level's channels reduced to $c/4$
- Low-dimension pooled maps are upsampled (can use bilinear interpolation) to same size as original
- Features then concatenated with original feature map
- Concatenated feature maps passed through upsampling and conv layers to generate

DeepLab[5]

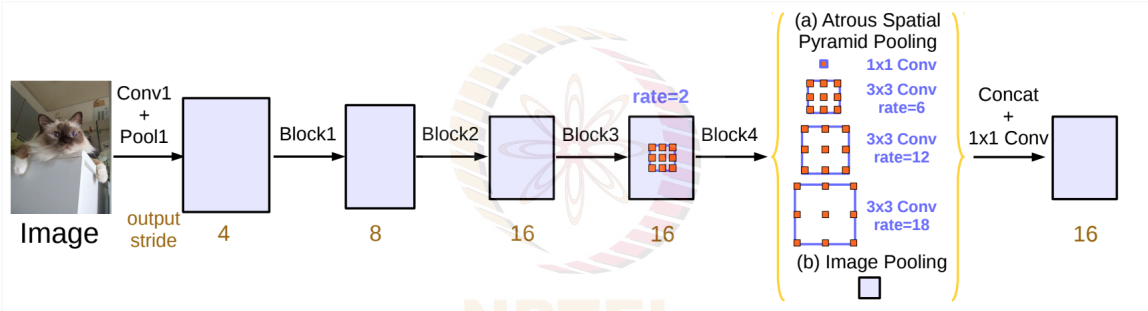
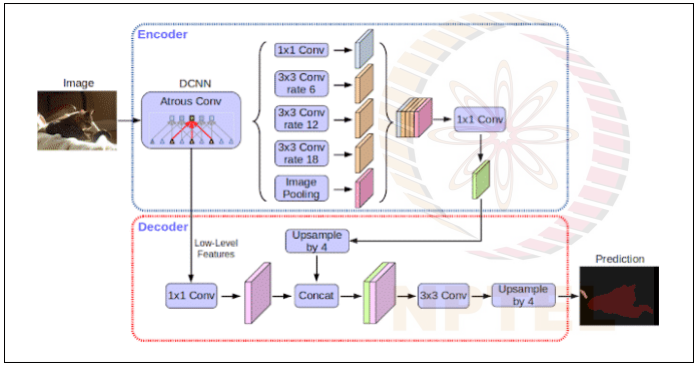


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

Note that atrous convolution = dilated convolution = transpose convolution = fractionally strided convolution

[5]Chen et al, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, TPAMI 2017

DeepLab V3[6]

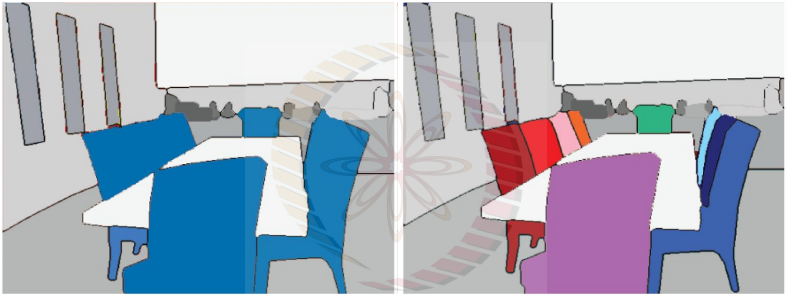


Adds image-level features to ASPP, batch normalization for easier training, decoder module to refine segmentation results

Credit: Kevin Windholm, Novatec-GMBH

[6] Chen et al, Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, ECCV 2018

Instance Segmentation

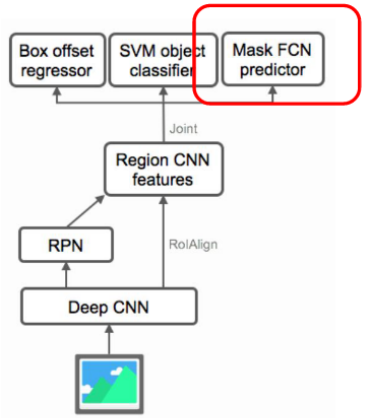


Semantic Segmentation

Semantic Instance Segmentation

Credit: Soroush, StackOverFlow

Mask R-CNN[7]

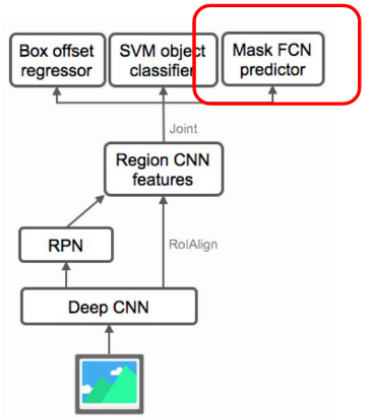


- Aims to tackle instance segmentation (where each pixel is given a class label, as well as an object ID)

Credit: Lilian Weng, Github.io

[7] He, Mask R-CNN, ICCV 2017

Mask R-CNN[7]

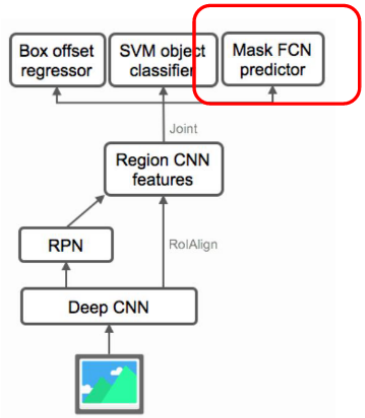


- Aims to tackle instance segmentation (where each pixel is given a class label, as well as an object ID)
- Mask R-CNN = Faster R-CNN with FCN on ROIs

Credit: Lilian Weng, Github.io

[7] He, Mask R-CNN, ICCV 2017

Mask R-CNN[7]



- Aims to tackle instance segmentation (where each pixel is given a class label, as well as an object ID)
- Mask R-CNN = Faster R-CNN with FCN on ROIs
- Adds a parallel head to predict masks along with existing branches for classification and localization

Credit: Lilian Weng, Github.io

[7] He, Mask R-CNN, ICCV 2017

Mask R-CNN: RoIAlign

- An improvement of RoIPool operation (used in detection frameworks)

NPTEL

Mask R-CNN: RoIAlign

- An improvement of RoIPool operation (used in detection frameworks)
- There is loss of information due to quantization when moving from object proposals in image space to proposals in feature space; further loss in RoIPool operation. Why does this matter?

NPTEL

Mask R-CNN: RoIAlign

- An improvement of RoIPool operation (used in detection frameworks)
- There is loss of information due to quantization when moving from object proposals in image space to proposals in feature space; further loss in RoIPool operation. Why does this matter?
- This is significant because one pixel in feature space is equivalent to many pixels on image

NPTEL

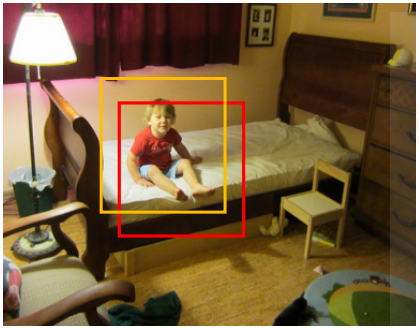
Mask R-CNN: RoIAlign

- An improvement of RoIPool operation (used in detection frameworks)
- There is loss of information due to quantization when moving from object proposals in image space to proposals in feature space; further loss in RoIPool operation. Why does this matter?
- This is significant because one pixel in feature space is equivalent to many pixels on image
- **RoIAlign** performs bilinear interpolation for the exact coordinate

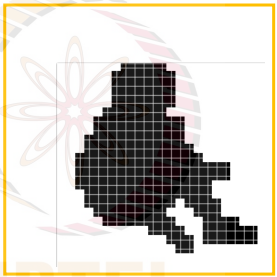
Mask R-CNN: RoIAlign

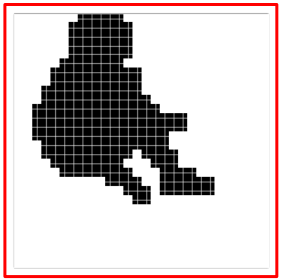
- An improvement of RoIPool operation (used in detection frameworks)
- There is loss of information due to quantization when moving from object proposals in image space to proposals in feature space; further loss in RoIPool operation. Why does this matter?
- This is significant because one pixel in feature space is equivalent to many pixels on image
- **RoIAlign** performs bilinear interpolation for the exact coordinate
- This preserves translation-equivariance of masks

Mask R-CNN: RoIAlign



target masks on RoIs

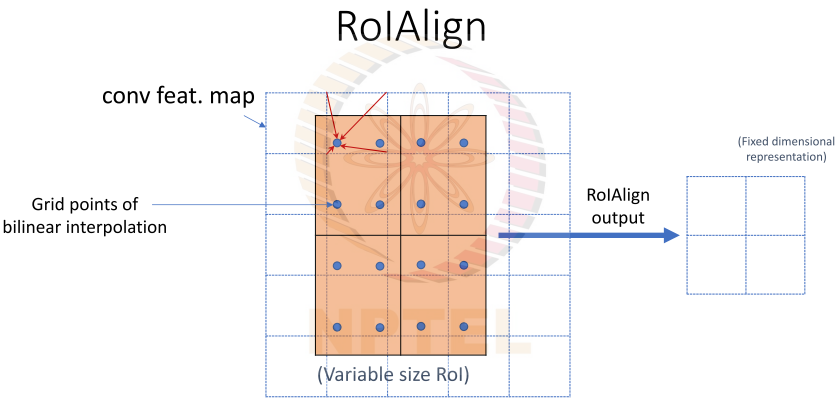




Translation of object in RoI => Same translation of mask in RoI

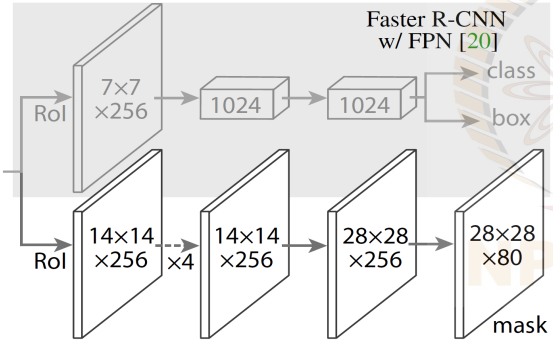
Credit: *Kaiming he, ICCV 2017 Tutorial*

Mask R-CNN: RoIAlign



Credit: *Kaiming he, ICCV 2017 Tutorial*

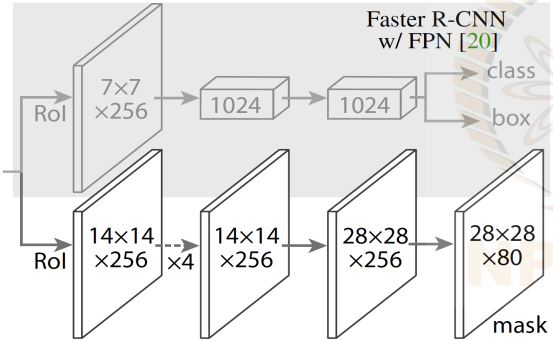
Mask R-CNN: FCN Mask Head



- FCN branch generates mask for each proposal

Credit: *Kaiming he, ICCV 2017 Tutorial*

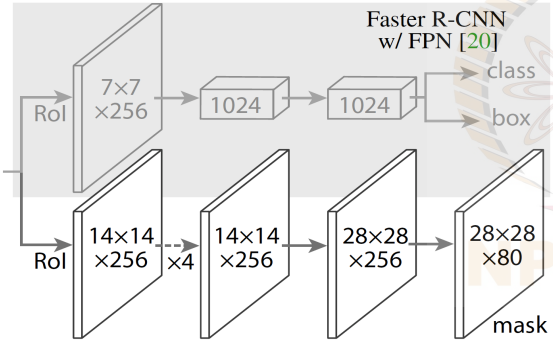
Mask R-CNN: FCN Mask Head



- FCN branch generates mask for each proposal
- Mask is 28×28 in size during training

Credit: *Kaiming he, ICCV 2017 Tutorial*

Mask R-CNN: FCN Mask Head

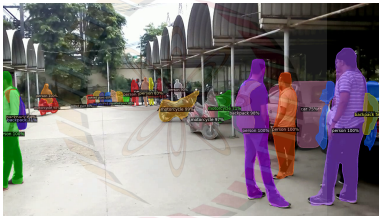


- FCN branch generates mask for each proposal
- Mask is 28×28 in size during training
- Rescaled to bounding box size and overlaid on image during inference

Credit: *Kaiming he, ICCV 2017 Tutorial*

Panoptic Segmentation[8]

Semantic Segmentation



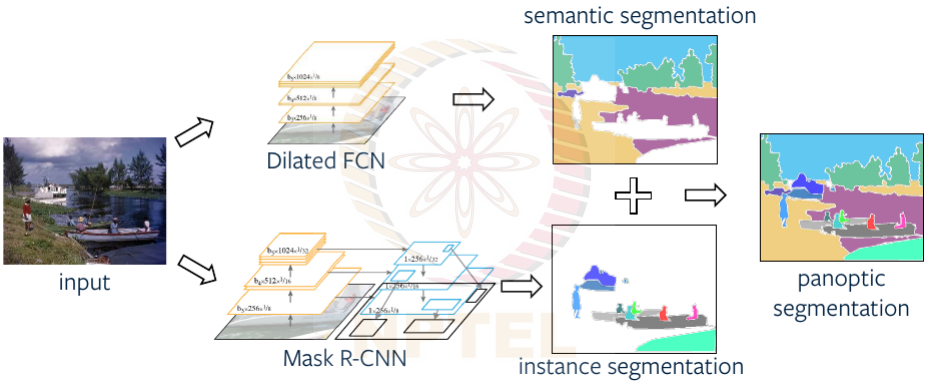
Instance segmentation

Panoptic segmentation

Credit: *Harshit Kumar, Github.io*

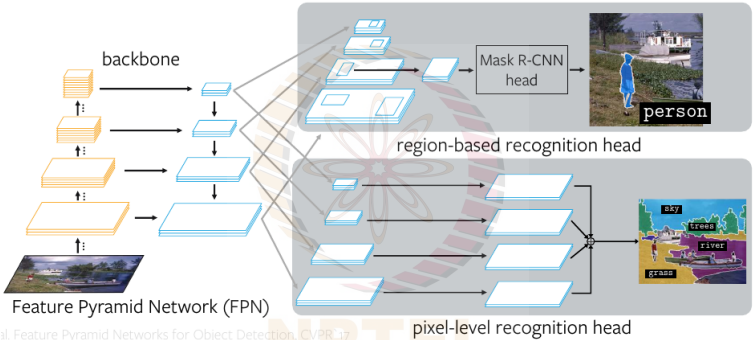
[8] Kirillov et al, Panoptic Segmentation, CVPR 2019

Panoptic Segmentation: Possible Architectures

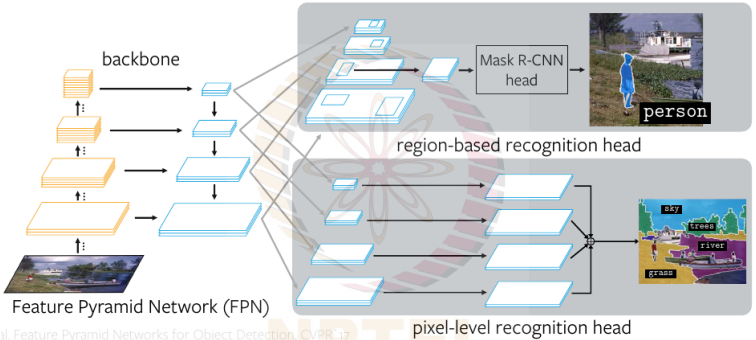


Credit: *Harshit Kumar, Github.io*

Panoptic Segmentation: Possible Architectures

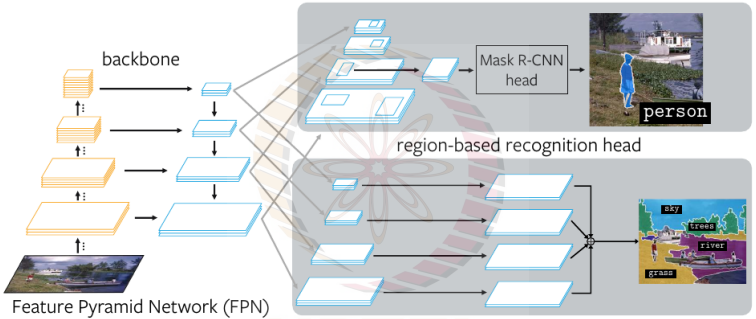


Panoptic Segmentation: Possible Architectures



Loss Function?

Panoptic Segmentation: Possible Architectures



Loss Function? $L = \lambda_i(L_{cls} + L_{bbox} + L_{mask}) + \lambda_s L_s$, where:

- Instance segmentation branch has L_{cls} = classification loss; L_{bbox} = bounding-box loss; L_{mask} = mask loss
- Semantic segmentation branch has L_s = pixel-wise cross-entropy loss

Panoptic Segmentation: New Evaluation Metric

Why do we need a new evaluation metric?

- In semantic segmentation, we use **IoU** and **per-pixel accuracy**
- In instance segmentation, we use **average precision over different IoU thresholds**
- Why can't we use these for panoptic segmentation?

NPTEL

Panoptic Segmentation: New Evaluation Metric

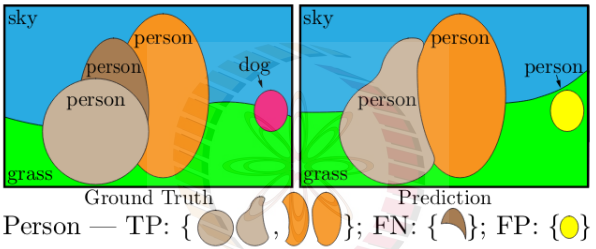
Why do we need a new evaluation metric?

- In semantic segmentation, we use **IoU** and **per-pixel accuracy**
- In instance segmentation, we use **average precision over different IoU thresholds**
- Why can't we use these for panoptic segmentation?
  Can cause asymmetry for classes with or without instance-level annotations

NPTEL

Credit: Harshit Kumar, Github.io

Panoptic Segmentation: PQ Metric



For a ground truth segment g, and for predicted segment p, PQ is calculated as:

$$PQ = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} = \underbrace{\frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality (RQ)}}$$

Credit: Harshit Kumar, Github.io

Homework

Readings

- Semantic Segmentation Overview by Nanonets
- Overview of Deep Lab, AnalyticsVidhya
- Introduction to Panoptic Segmentation