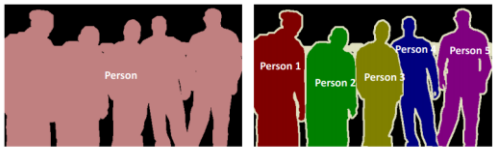
Semantic Segmentation

- Task of assigning a class to every pixel in a given image, also called dense prediction

- Used in
  1) self-driving cars to navigate through routes
  2) portrait mode of google pixel where each pixel is classified as foreground or background and then background pixels are blurred
  3) medical image diagnosis

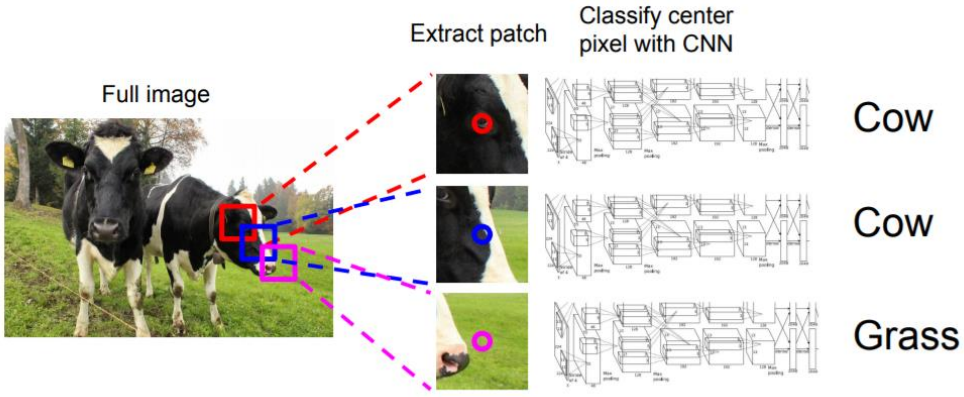
- Don't differentiate instances, only care about pixels. In instance segmentation, different instances of the same class are segmented individually



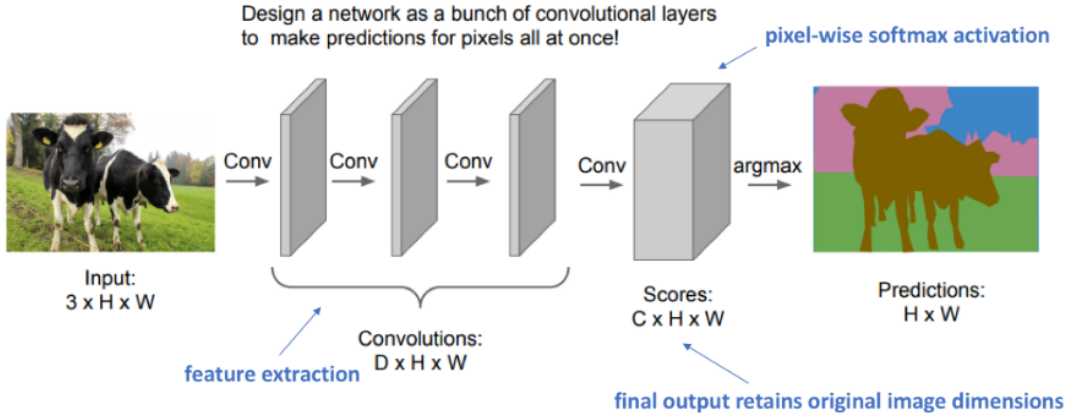
Semantic Segmentation **Instance Segmentation**

- Methods
  - Non-neural network based: watershed, image thresholding, k-means clustering, graph partitioning etc.
  - Neural network based: FCN, U-Net, SegNet, DeepLab etc. [FOCUS HERE]

- First naïve way to segment images was using patch classification where each pixel was separately classified into classes using a patch of image around it. Main reason to use patches was that classification networks usually have full connected layers that require fixed size images.
  - Drawback: very inefficient, not reusing shared features between overlapping patches; requires fixed size patch

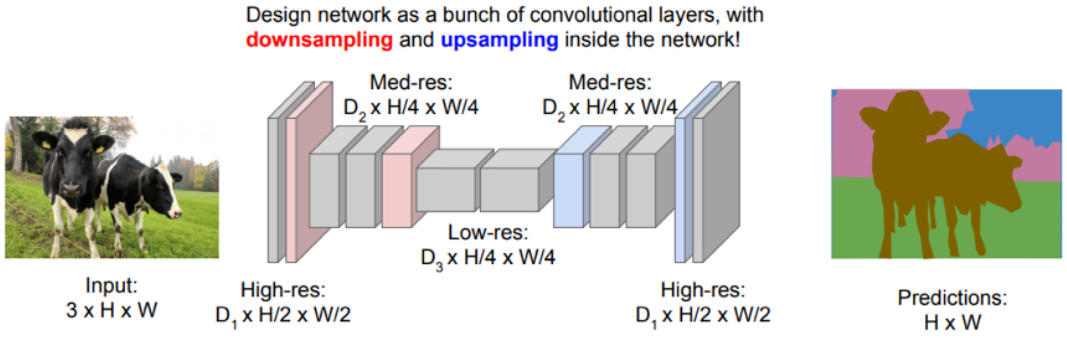


- Second naïve way to segment images is passing an image through a bunch of convolutional layers without padding and max-pooling (to retain pixel location) and output a segmentation map
    - Drawback: computationally expensive and less expressive since we cannot increase the number of features maps at deeper layers to preserve resolution



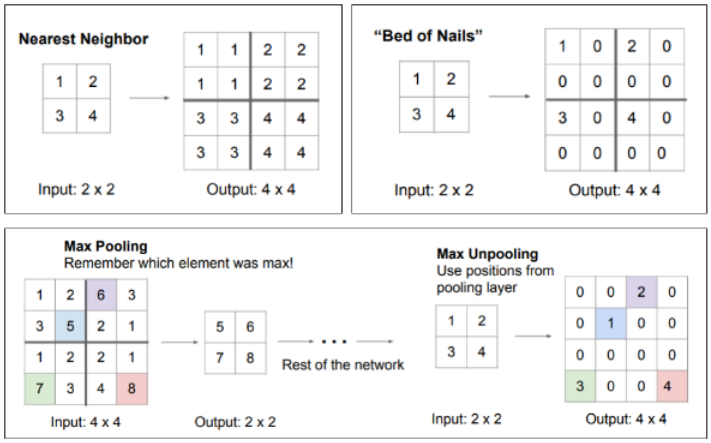
Downside: Preserving image dimensions throughout entire network will be computationally expensive.

- In classification, we are interested whether the image contains that feature or not and not worried about location so downsampling can be used to reduce computational burden. But in segmentation, we are interested in both the *presence and location* of the feature in order to produce a full resolution prediction map.
- An encoder-decoder architecture is suitable for semantic segmentation where we downsample the image to a latent feature space and then upsample the latent feature space to full resolution map. So we can use deeper features in encoder without compromising output resolution.

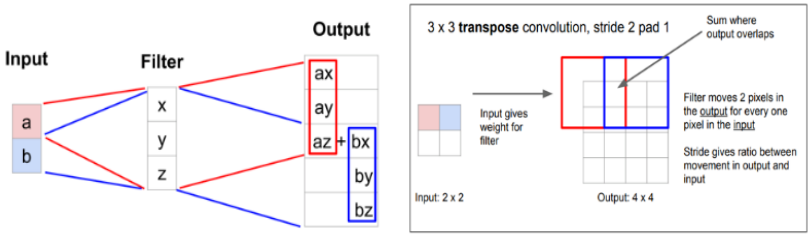


Solution: Make network deep and *work at a lower spatial resolution* for many of the layers.

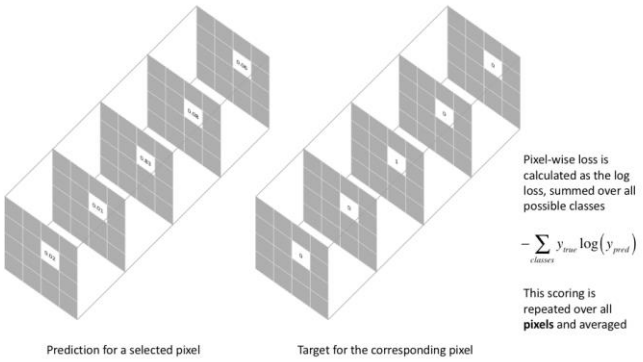
- Downsampling
    - Pooling and strided convolution-> downsamples the resolution by summarizing a local area with a single value
- Upsampling
    - Unpooling -> upsamples the resolution by distributing a single value into a higher resolution
    - Methods: Nearest neighbor, bed-of-nails, max-unpooling, transpose convolution



- Transpose convolutions are the most popular -> learnable upsampling
  - Typically a 3x3 convolution, stride = 1, pad = 1
  - Also known as upconvolution, full convolution, deconvolution or fractionally-strided convolution.
  - Reverse dot operation -> instead of multiplying two vectors to produce a single value, we multiply the single value with weight vector to project those weighted values as an output vector. The overlapping values are summed.



- Loss functions
  - Pixel wise cross-entropy Loss



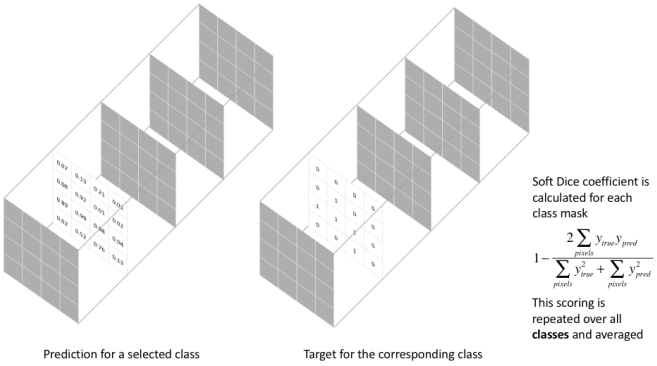
- Problem: in case of unbalanced number of pixels per class, the training can be dominated by majority class
- Solution:
  1) Weighing the loss for each class
  2) Weighing the loss of each pixel => higher loss weight is given to boundary pixels

- Soft Dice Loss
  - Dice coefficient measures an overlap between two samples (masks)

$$Dice = \frac{2|A \cap B|}{|A| + |B|}$$

$$|A \cap B| = \underbrace{\begin{bmatrix} 0.01 & 0.03 & 0.02 & 0.02 \\ 0.05 & 0.12 & 0.09 & 0.07 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix}}_{\text{prediction}} * \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}}_{\text{target}} \xrightarrow{\text{element-wise multiply}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix} \xrightarrow{\text{sum}} 7.41$$

  - Dice coefficient is a sort of ratio of common activations between prediction and target to total number of activations
  - In some cases a squared sum is used in place of a simple sum in denominator. 2 is used to normalize the double counts in the denominator
  - Soft dice loss = 1 - dice
  - Since we normalize with the total activations, soft dice loss does not struggle learning from classes with lesser spatial representation

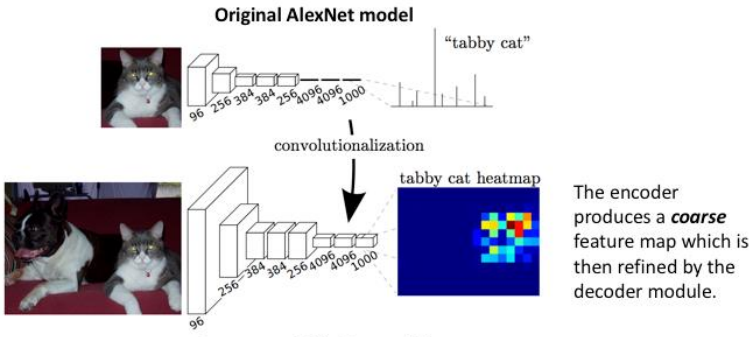


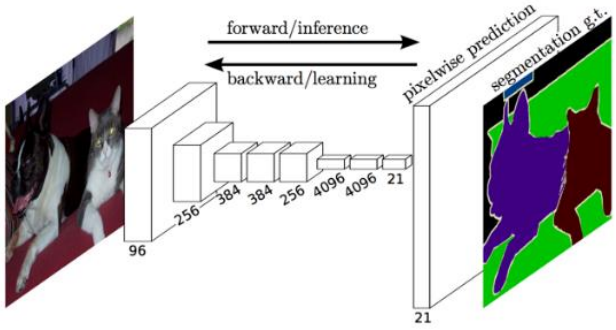
- Condition Random Field post-processing are usually used to improve the segmentation. Smooth the segmentation based on the underlying image intensities, pixel having similar intensities should be labelled same class. CRFs can boost scores by 1-2%.

- Note that predicted segmentation map's size is 1/8th of that of the image in almost all approaches. Thus it is interpolated to generate the full-resolution segmentation map.

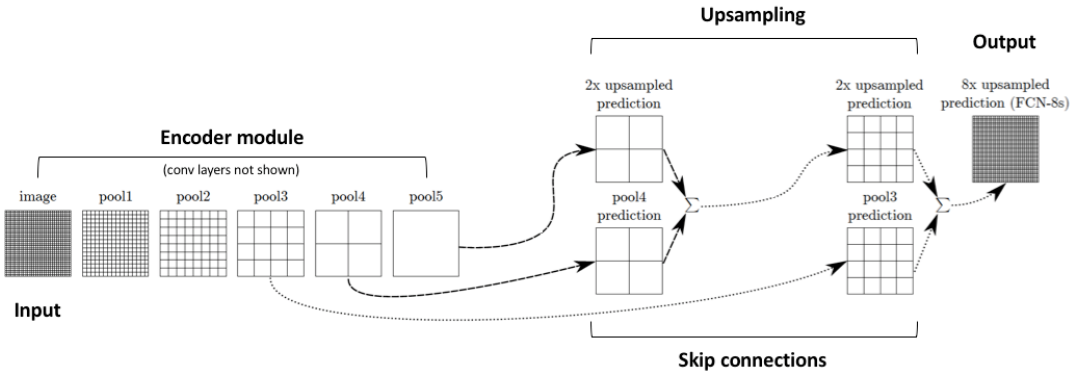
- Chronological sequence of networks:
  1. FCN (2014)
  2. SegNet (2015)
  3. Dilated Convolutions (2015)
  4. UNet (2015)
  5. DeepLab (v1) (2014)
  6. RefineNet (2016)
  7. PSPNet (2016)
  8. DeepLab v2 (2016)
  9. DeepLab v3 (2017)
  10. DeepLab v3+ (2018)

- Fully convolutional networks
  - Proposed by Long et al. in 2014
  - Proposed adapting existing classification architectures like AlexNet for encoder module and transpose convolution layers to upsample coarse feature maps for a decoder module
  - Did not use any fully connected layers so input image of any size can be segmented
  - Convolutionized the FC layers to generate a coarse feature map. How to convolutionize?

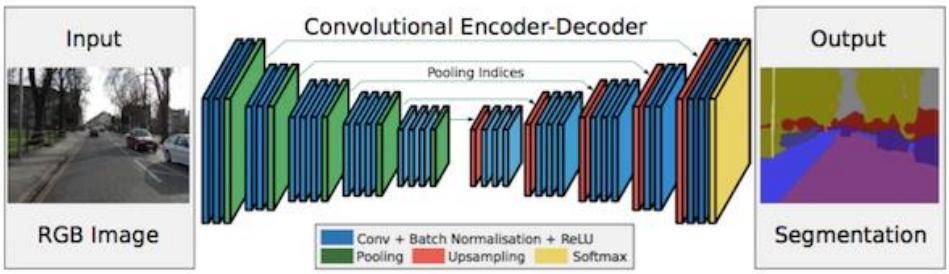




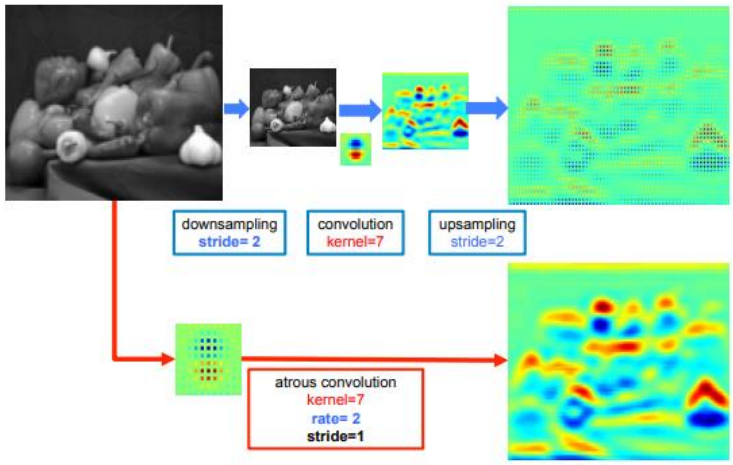
  - Drawback: the encoder module reduces the resolution by a factor of 32 which makes the task of decoder module to locate features very difficult
  - Basically the encoder can learn *what* feature easily but it is difficult for decoder to learn *where the feature is.* The direct predictions of FCN are typically in low resolution, resulting in fuzzy object boundaries.
  - Solution: use 'skip connections' in decoder module which use the features from the previous convolutional layers (high resolution feature maps). This is because the starting layers retain the location information better than the deeper layers so can help to reconstruct shapes of object boundaries accurately.



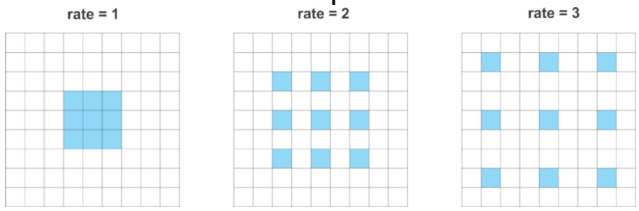
- SegNet
  - Proposed a new way of upsampling
  - Instead of copying encoder features as in FCN using skip connections, indices from the maxpooling layer are copied. This makes SegNet more memory efficient and eliminates the need for learning to upsample.
  - Drawback: Pooling reduces feature map resolution and hence output map resolution. Even if extrapolated to original resolution, lossy image is generated



- Dilated Convolutions/Atrous Convolution
  - a generalized form of convolution which eliminates pooling
  - pooling helps in classification networks because it increases the receptive field
  - However, pooling is not suitable in segmentation since it reduces resolution



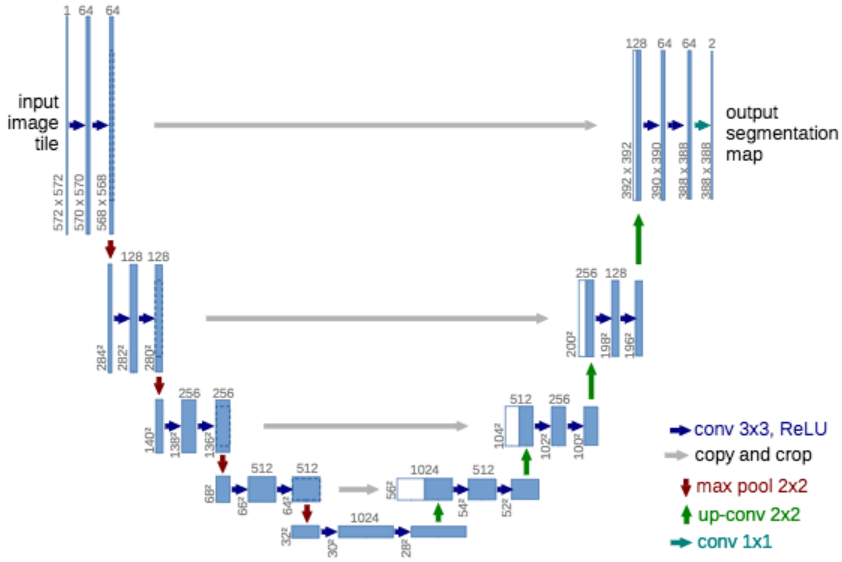
  - Another way to increase the receptive field is to increase the filter size but it is increases the number of parameters
  - So to increase receptive field size without decreasing resolution and without increasing number of parameters, dilated convolution was introduced
  - In dilated convolution, holes are created in the filter. The number of holes are controlled by dilation rate (r).
  - A normal convolution is special case of dilated convolutions with r=1.



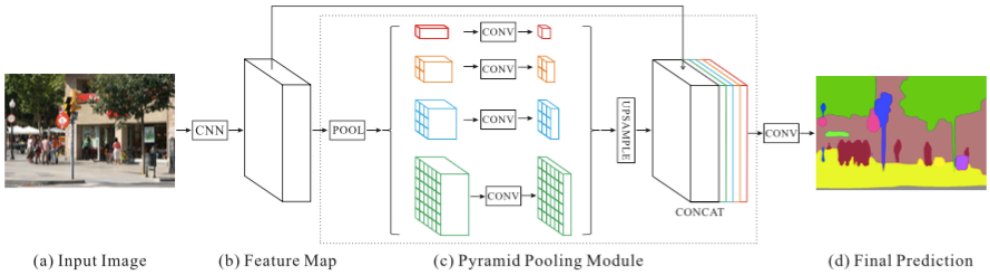
$$y[i] = \sum_k x[i + r \cdot k] w[k]$$

  - Allows user to explicitly control the resolution at which feature maps are computed

- U-Net
  - Extension of FCN for biological microscopy images
  - Composed of two parts: a contracting part to compute features and an expanding part to spatially localize patterns in the image
  - Contracting part (downsampling) has FCN like architecture extracting features with 3x3 convolution
  - Expanding part (upsampling) uses deconvolution reducing the number of feature maps while increasing their height and width
  - Cropped feature maps from the downsampling part of the network are copied to the upsampling part to retain pattern information
  - 1x1 convolution is used to process the feature maps and generate a segmentation map
  - Can be trained with small number of images

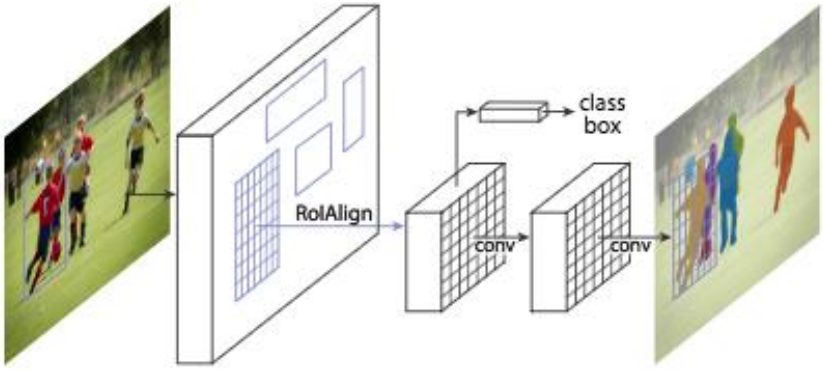


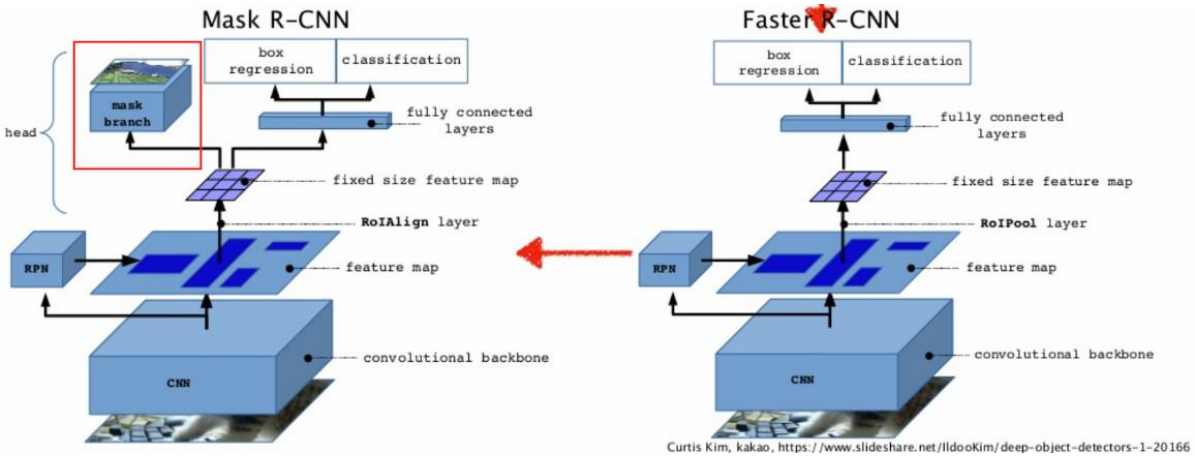
- PSPNet
  - Introduced pyramid pooling module to take into account global contextual information which gives clues about the segmentation class distribution
  - First trained a ResNet CNN with dilated convolution to generate feature maps
  - The feature maps are pooled at different scales, convolved with 1x1 kernel to reduce their size and then upsampled and concatenated with initial feature maps to contain both local and global context information



  - An auxiliary loss additional to the main loss is used which is input to the pyramid pooling module). This is called intermediate supervision??

- Mask RCNN : Region Based Semantic Segmentation]
  - Built on top of Faster R-CNN which is originally designed for object detection
  - Faster R-CNN uses a region proposal network (RPN) to propose bounding box candidates around objects
  - The RPN extracts ROI and an ROIPool layer extracts features from the proposals in order to infer the bounding box coordinates and the class of the object
  - Mask RCNN is a Faster RCNN with 3 outputs: bounding box coordinates, class of object inside the bounding box and binary mask to segment the object
  - The binary mask has a fix size and is generated by the FCN for the given ROI
  - Uses a ROIalign layer instead of ROIpool layer to avoid mis-alignemnets due to quantization of the ROI coordinates. What the heck is this?
  - The unique point of Mask RCNN is its multi-task loss combining the losses of bounding box coordinates, the predicted class and the segmentation mask
  - The model tries to solve complementary tasks leading to better performances on each individual task



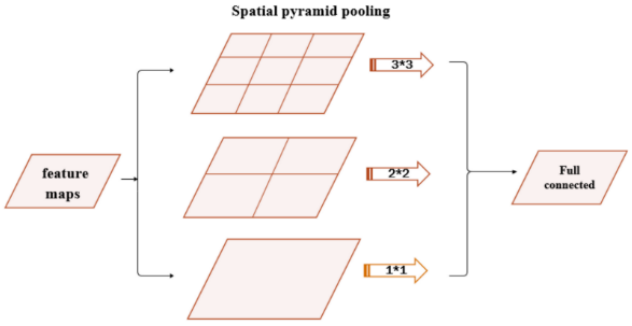


Curtis Kim, kakao, https://www.slideshare.net/IldooKim/deep-object-detectors-1-20166

- DeepLab (v1, v2, v3 and v3+)

  V1 & V2

  - Incorporated multi-scale information to make the model robust to object scale changes (scale invariance)
  - Inspired by feature pyramid network (FPN) for semantic segmentation
  - Proposed Spatial pyramid pooling (SPP) which uses multiple scaled versions of the input feature map for training



  - SPP has high computation complexity since it uses multiple model instances
  - To reduce that complexity and increase receptive field size, dilution convolution (termed as atrous here) convolution is used instead of normal convolution
  - SPP with atrous convolution is called *atrous spatial pyramid pooling.* Four parallel atruous convolution of the same input with different rates are applied to detect spatial patterns on top of the feature map.

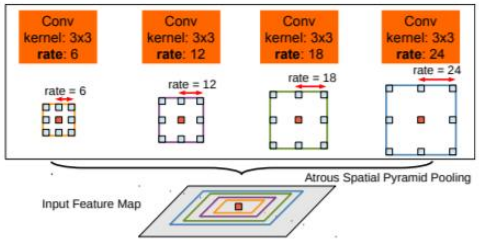


Fig. 4: Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

  - The features maps are processed in separate branches and concatenated using bilinear interpolation to recover the original size of the input. The output feeds a fully connected Conditional Random Field (CRF). What is the role of FC CRF and how it is different from CRF?
  - backbone feature extraction architecture: ResNet-101

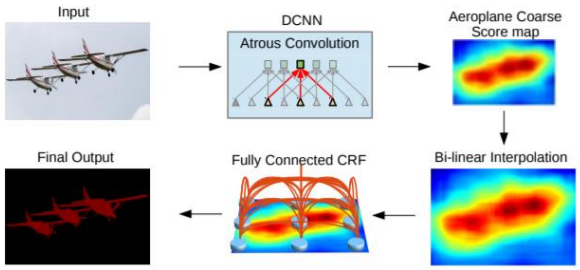
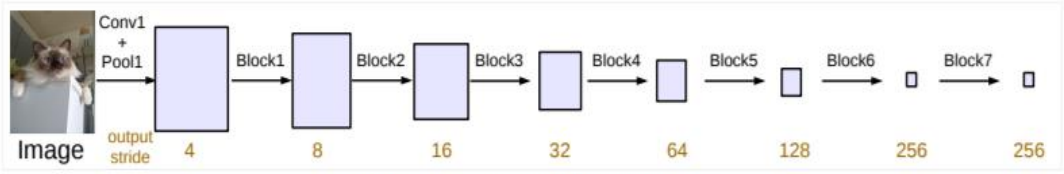


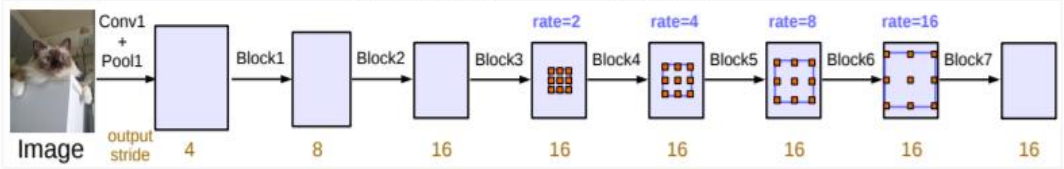
Fig. 1: Model Illustration. A Deep Convolutional Neural Network such as VGG-16 or ResNet-101 is employed in a fully convolutional fashion, using atrous convolution to reduce the degree of signal downsampling (from 32x down 8x). A bilinear interpolation stage enlarges the feature maps to the original image resolution. A fully connected CRF is then applied to refine the segmentation result and better capture the object boundaries.

V3

- ResNet model is modified to use dilated/atrous convolutions as in DeepLabv2 and dilated convolutions. Improved ASPP involves concatenation of image-level features, a 1x1 convolution and three 3x3 atrous convolutions with different rates. Batch normalization is used after each of the parallel convolutional layers.
- Although ASPP with different atruous rates effectively captures multli-scale information in V2. But, as the dilation rate becomes larger, the number of valid filter weights becomes smaller. Example, when applying a 3x3 filter to a 65x65 feature map with different atrous rates, in the extreme case when the rate value is close to feature map size, the 3x3 filter instead of capturing the whole image context, degenrates to a simple 1x1 filter since only the centre filter weight is effective. In short, it cannot take into account the global context.
- To overcome this problem and incorporate global context information to the model, image-level features were concatenated to atrous convolution features.
- Specifically, applied global pooling on the last feature map of the model, feed the resulting image-level features to a 1 × 1 convolution with 256 filters (and batch normalization), and then bilinearly upsample the feature to the desired spatial dimension.
- Improved ASPP consists of
    (a) one 1×1 convolution and three 3 × 3 convolutions with rates = (6, 12, 18) when output stride = 16 (all with 256 filters and batch normalization), and
    (b) second image-level features
- The resulting features from all the branches are then concatenated and pass through another 1 × 1 convolution (also with 256 filters and batch normalization) before the final 1 × 1 convolution which generates the final logits.



(a) Going deeper without atrous convolution.



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

Figure 3. Cascaded modules without and with atrous convolution.

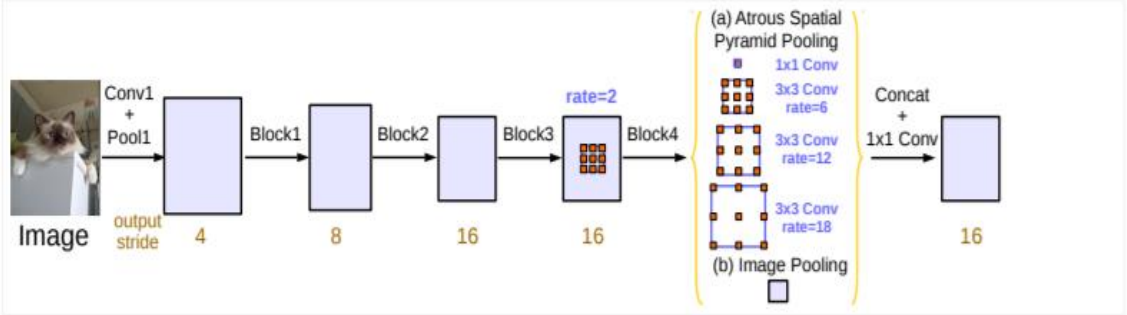
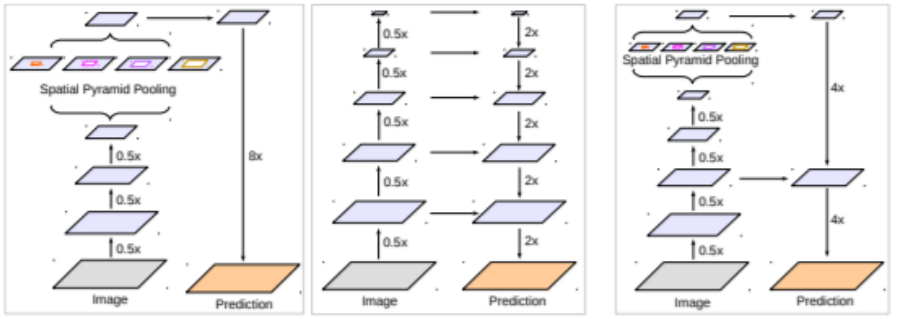


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

V3+

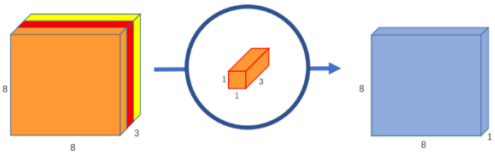
- Combine SPP (encodes multi-scale information) with an encoder-decoder structure (captures sharper object boundaries by recovering the spatial info)
- extends DeepLabv3 by adding a simple yet effective decoder module to refine the segmentation results especially along object boundaries
- adapted the Xception model for segmentation task and applied the depthwise separable convolution to both Atrous Spatial Pyramid Pooling and decoder modules, resulting in a faster and stronger encoder-decoder network



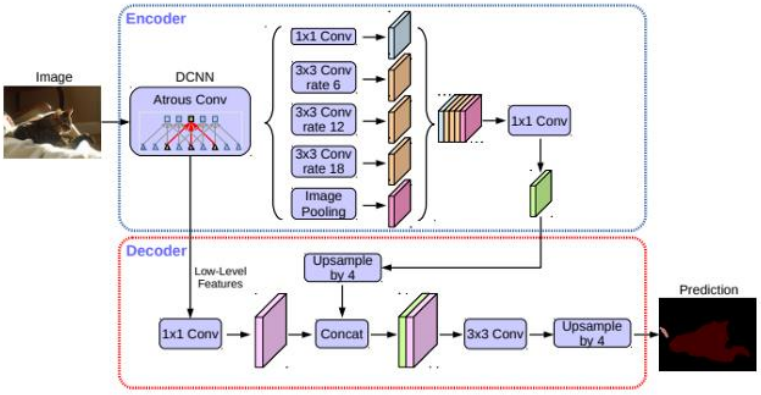
(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Conv

- Depthwise Separable Convolution – method to perform convolutions with less number of computations with similar performance than a standard convolution operation
  - Breaks down convolution operation into two steps- depthwise and pointwise
  - Depthwise convolution

  - Pointwise convolution



  - So basically dividing a 5x5x3 filters into 5x5x1 (depthwise) and 1x1x3 (pointwise)
  - To increase the spatial context -> increase size of depthwise; to increase the number of output channels -> increase size of pointwise.



- Summary of DeepLab models

DeepLab have published four versions so far: 1, 2, 3 and 3+. Below are detailed the main innovations of each version:

1. DeepLabV1: used atrous convolution to explicitly control the resolution at which feature responses are computed within Deep Convolutional Neural Networks.
2. DeepLabV2: used filters at multiple sampling rates and effective fields-of-view, the method is known as atrous spatial pyramid pooling (ASPP) -- reviewed above.
3. DeepLabV3: augmented the ASPP module with image-level features to capture longer range information, and added batch normalization.
4. DeepLabV3+: extended DeepLabV3 to include a simple yet effective encoder-decoder module to refine the segmentation results particularly along object boundaries.

- Evaluation Metrics
  - meanIOU: The IOU is the ratio of overlap between the area of overlap and the area of union between the ground truth and the predicted areas. mIOU is the average between the IOU of the segmented objects over all the images of the test dataset.

- Summary of inventions
  - Encoder-decoder model for feature extraction and localization
  - Skip-connections for precise boundaries
  - Dilated convolution for controlling the resolution and increase receptive field
  - Spatial pyramid pooling for incorporating global context
  - Atrous Spatial pyramid pooling for multi-scale information with less computation
  - Concatenation of image features with ASPP to improve global context when r is high
  - ASPP with encoder-decoder module to take advantages of both

- Comparison of different methods on benchmark datasets

Model	2012 PASCAL VOC (mIoU)	PASCAL-Context (mIoU)	2016 COCO (AP)	2016 COCO (AR)	2017 COCO (AP)	Cityscapes (mIoU)
FCN	62.2	x	x	x	x	x
ParseNet	69.8	40.4	x	x	x	x
Conv & Deconv	72.5	x	x	x	x	x
FPN	x	x	x	48.1	x	x
PSPNet	85.4	x	x	x	x	80.2
Mask R-CNN	x	x	37.1	x	41.8	x
DeepLab	79.7	45.7	x	x	x	70.4
DeepLabv3	86.9	x	x	x	x	81.3
DeepLabv3+	89.0	x	x	x	x	82.1
PANet	x	x	42.0	x	46.7	x
EncNet	85.9	52.6	x	x	x	x

Overview of the scores of the models over the 2012 PASCAL VOC dataset (mIoU), the PASCAL-Context dataset (mIoU), the 2016 / 2017 COCO datasets (AP and AR) and the Cityscapes dataset (mIoU)

References:

1. https://www.analyticsvidhya.com/blog/2019/02/tutorial-semantic-segmentation-google-deeplab/
2. https://www.jeremyjordan.me/semantic-segmentation/
3. https://medium.com/nanonets/how-to-do-image-segmentation-using-deep-learning-c673cc5862ef
4. http://blog.qure.ai/notes/semantic-segmentation-deep-learning-review
5. https://medium.com/@arthur_ouaknine/review-of-deep-learning-algorithms-for-image-semantic-segmentation-509a600f7b57
6. http://www.cs.toronto.edu/~tingwuwang/semantic_segmentation.pdf
7. http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf
8. https://github.com/tensorflow/models/tree/master/research/deeplab
9. Research papers of each network