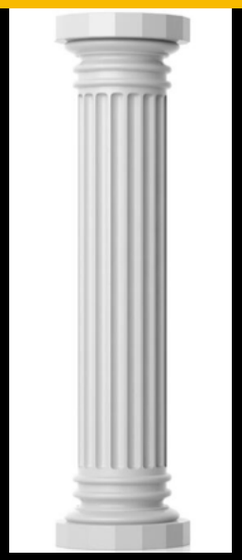
Supervised Learning: Segmentation

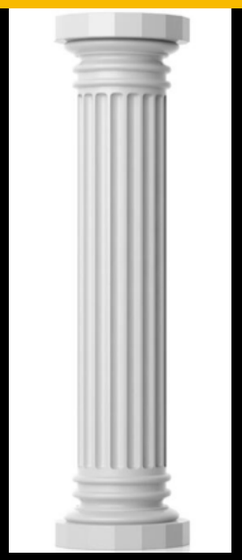
Earl Wong

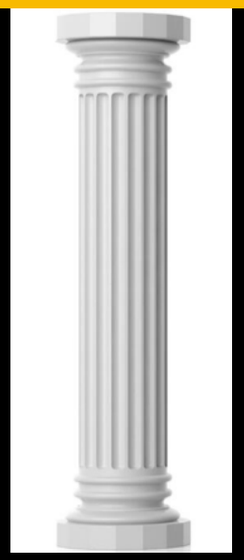
Labeled Dataset

Algorithm

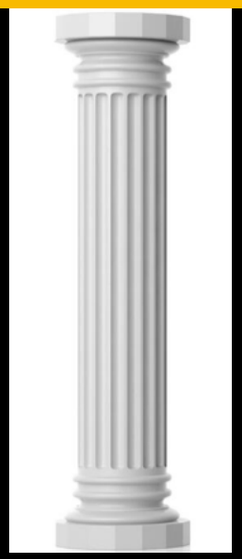
Loss Function







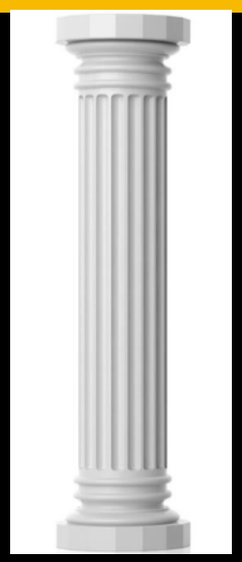
Labeled Dataset



Labeled Dataset

Dataset	Pascal VOC	Pascal Context	CityScapes	COCO/Stuff	ADE
Object Classes	21	60	30	81/171	150
Images	~2.9K	10K	25K	120K/10K	25K

Algorithm



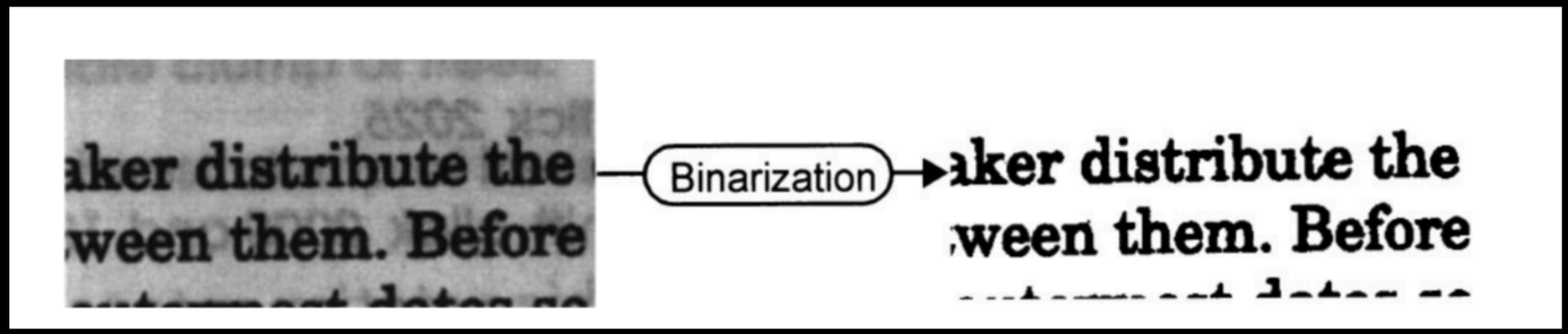
Historical

Image segmentation has a very, very rich history.

We will briefly discuss a practical application, applied to text.

In addition, we will look at a representative method describing how the problem was approached, prior to the deep network era.

Adaptive Image Binarization



Binarization of textural components was achieved by varying the threshold in a moving window that was slid over the image:

$$T(x, y) = m(x, y) \cdot \left[1 + k \cdot \left(\frac{s(x, y)}{R} - 1\right)\right]$$

m(x,y) denoted the local mean for a given window size.

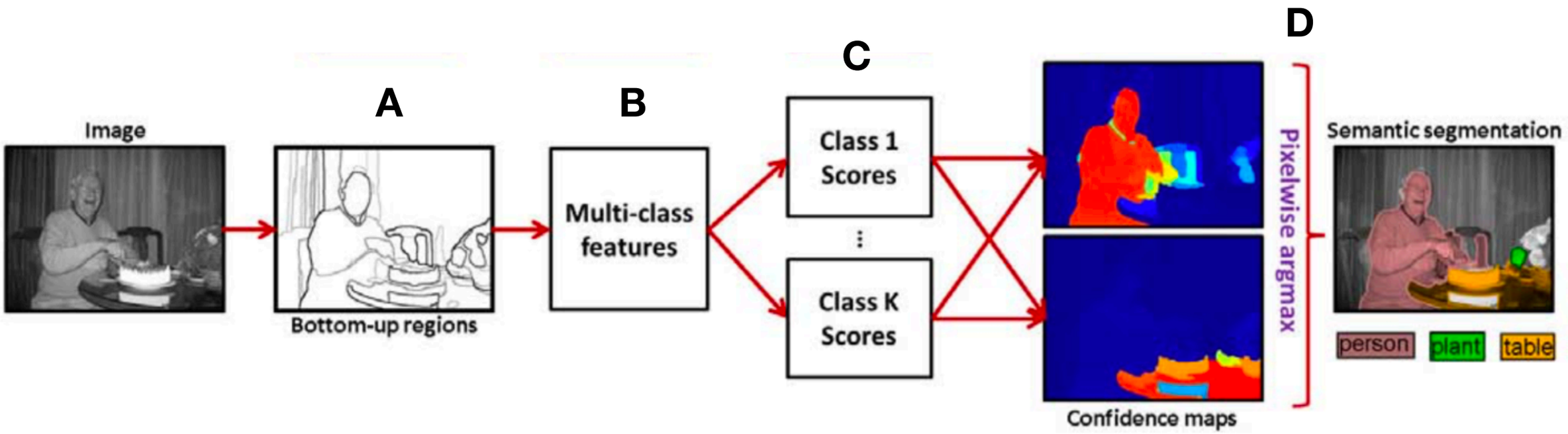
s(x,y) denoted the local standard deviation for a given window size.

R represented the dynamic range of an image, and was typically set to 128.

K was a user defined tuning parameter.

This approach was highly effective, for appropriately chosen window sizes.

Regions and Parts



This paper provides a generic “flavor”, regarding the complexity of semantic segmentation prior to the deep learning era.

A) First, a bottoms up approach was used to generate object candidate regions. The bottoms up approach exploited uniformity in brightness, color or texture.

B) Next, a 856 dimensional multi-class feature vector was generated, using various feature analysis methods and a poselets parts model, applied to the object candidate regions.

C) Twenty category specific classifiers were then trained to output category scores associated with the 856 dimensional vector.

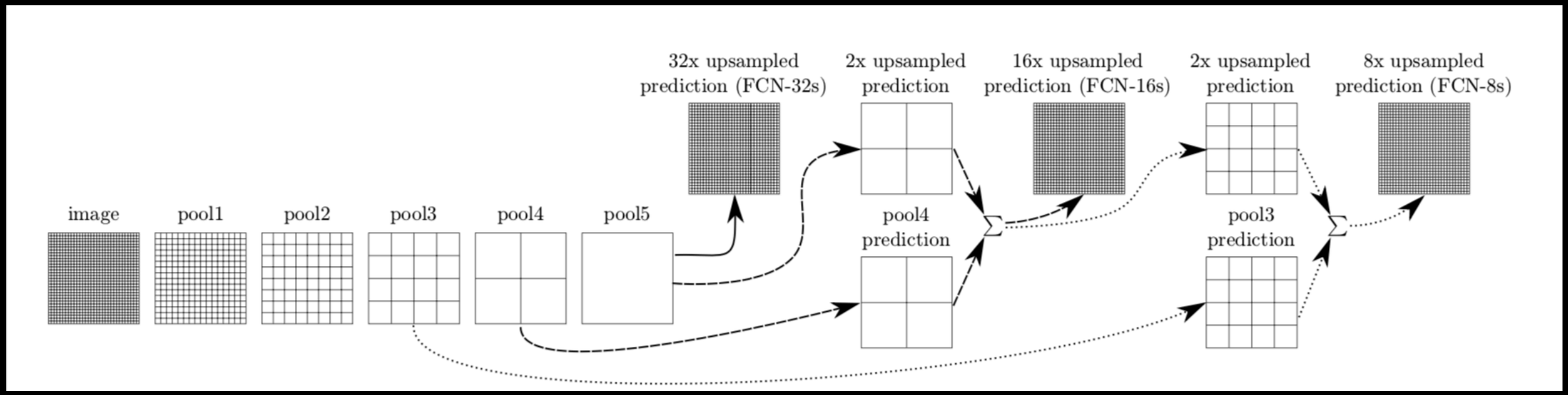
D) Finally, an additional eighty classifiers were trained, to project the category scores onto a single pixel location.

Deep Network Era

- Fully Convolutional Network
- Learning Deconvolution Network
- U-Net & V-Net
- DeepLab
- Attention to Scale & EMANet
- Adaptive Context Network
- Object Contextual Representations
- Mask R-CNN

The above papers were chosen based on historical merit, fundamental concepts and / or current SOTA.

Fully Convolutional Network (FCN)



This was the first work to perform end to end training on a fully convolutional network to produce a pixelwise prediction map.

The network consisted of two parts.

The first part was the standard encoder network used in classification, less the fully connected layers.

The second part was an upsampling process.

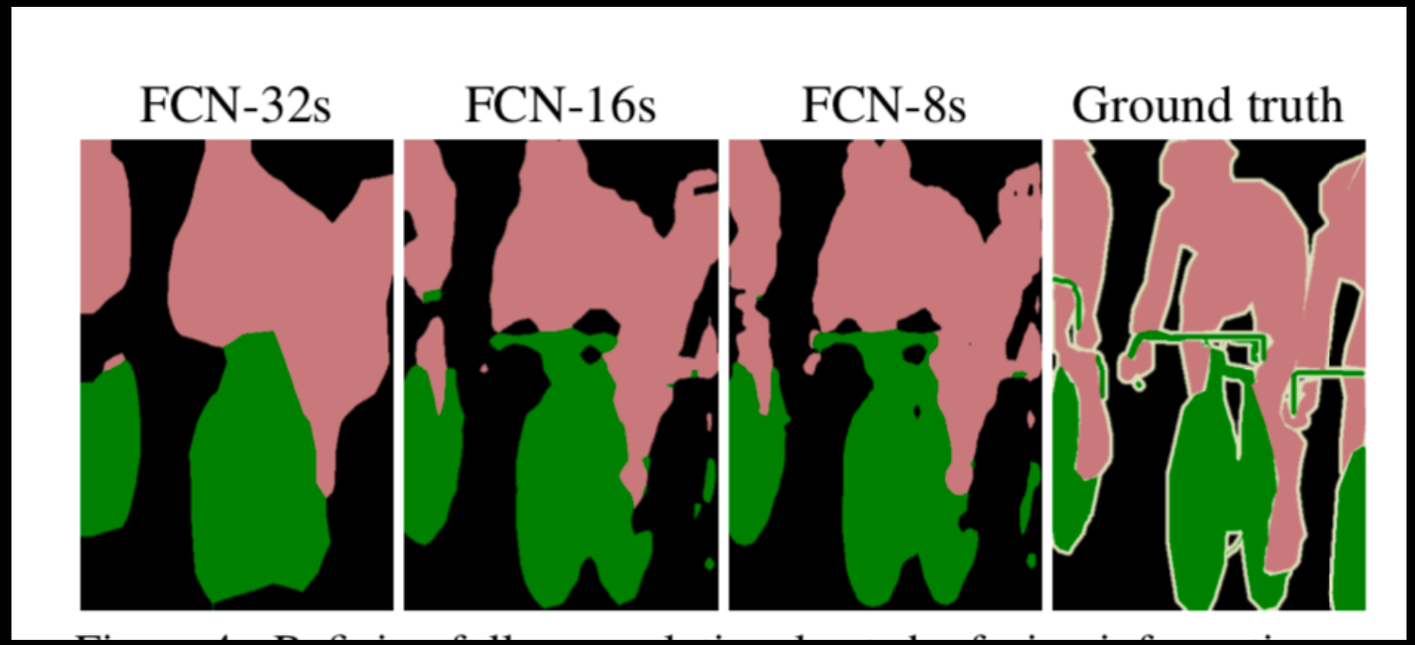
Fully Convolutional Network (FCN)

A pixel prediction map resulted from either:

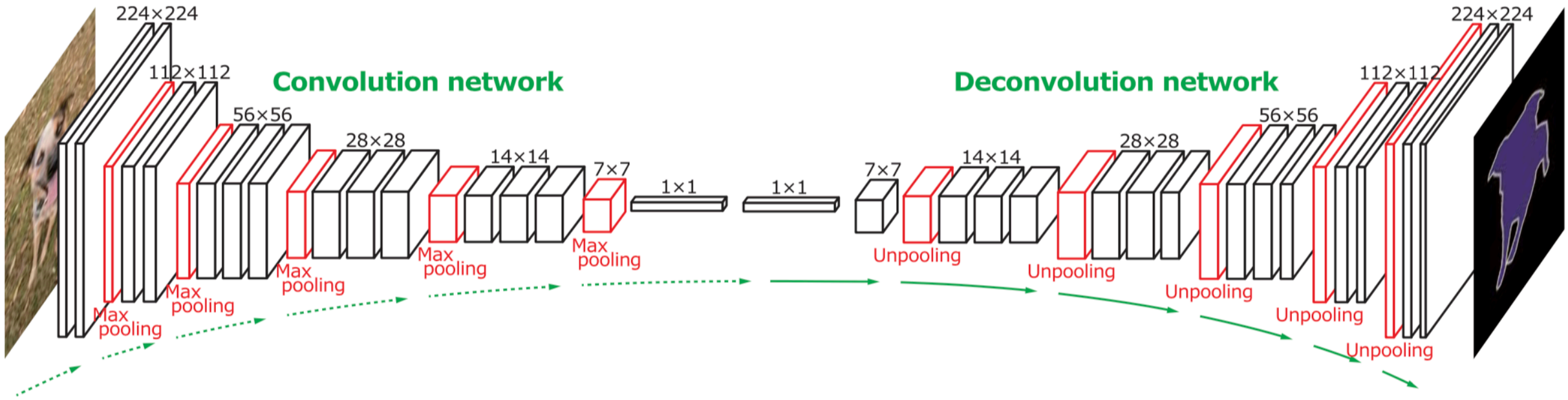
1) Upsampling a pooling layer in the encoder network to the appropriate resolution
2) Combining two pooling layers of different resolutions (by upsampling the lower resolution pooling layer by 2x) and then upsampling the combined result to the appropriate resolution

The network was trained by adjusting the weights in the encoder network using backpropagation.

Using different “skip” architectures from different combinations of 1) and 2), different segmentation results were produced. (See the previous slide for FCN-xx definitions.)



Learning Deconvolution Network

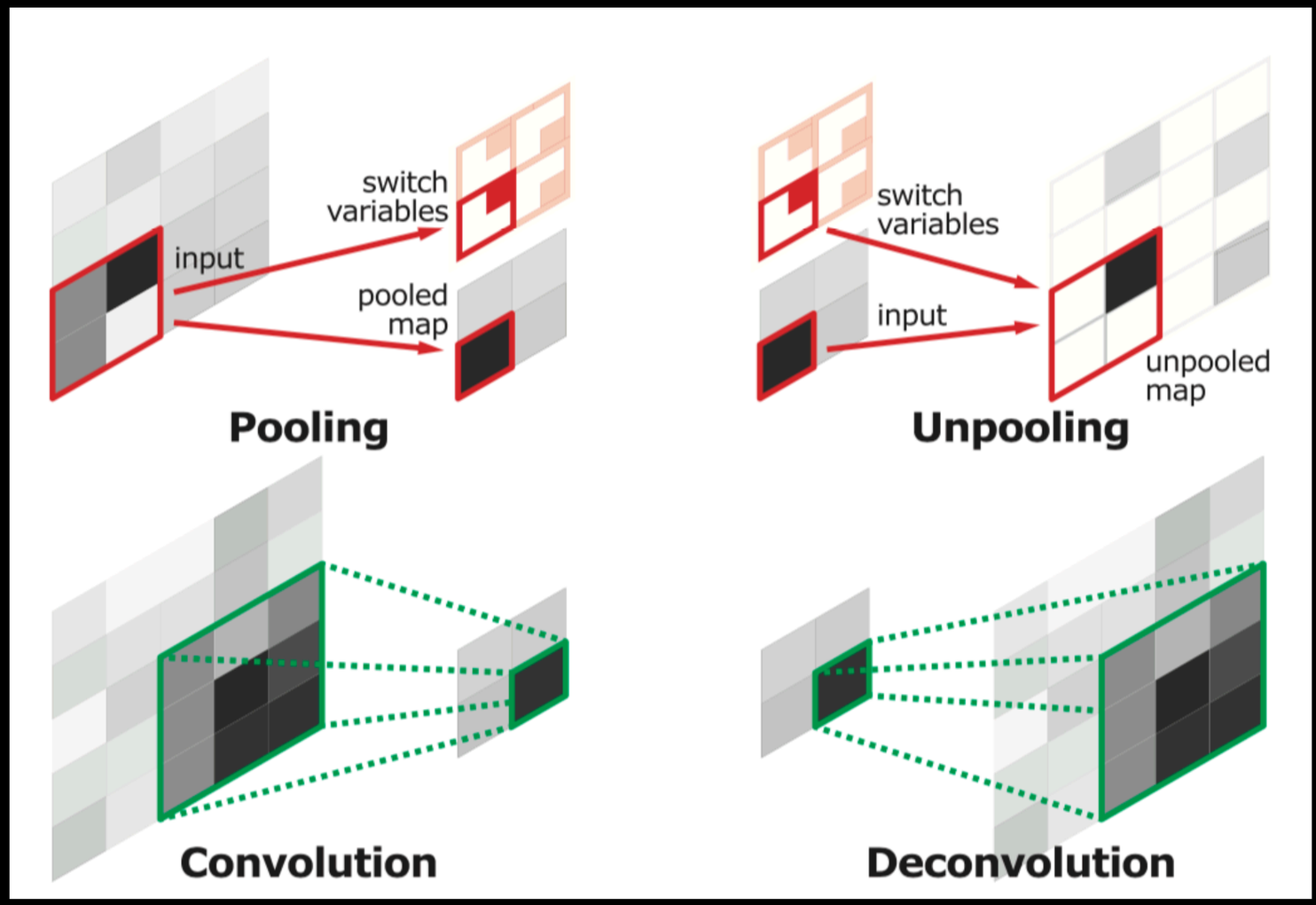


Although FCN's were fully convolutional, FCN's did not contain a true decoder based upsampling network (deconvolution network) with transpose convolution filters.

The above network introduced the latter, resulting in an end to end encoder-decoder network (convolution-deconvolution network) containing 252M parameters.

By construction, this network provided a new method for upsampling, by aggregating information from coarse to fine, to produce the pixelwise prediction map.

Learning Deconvolution Network



The transpose convolution layers in the decoder network consisted of learned transpose convolution filters and unpooling operations.

The learned transpose convolution filters provided an extra degree of freedom, for producing an accurate pixelwise prediction map.

As shown above, the unpooling operations reversed the spatial pooling operations in the encoder stage through the use of a switch variable.

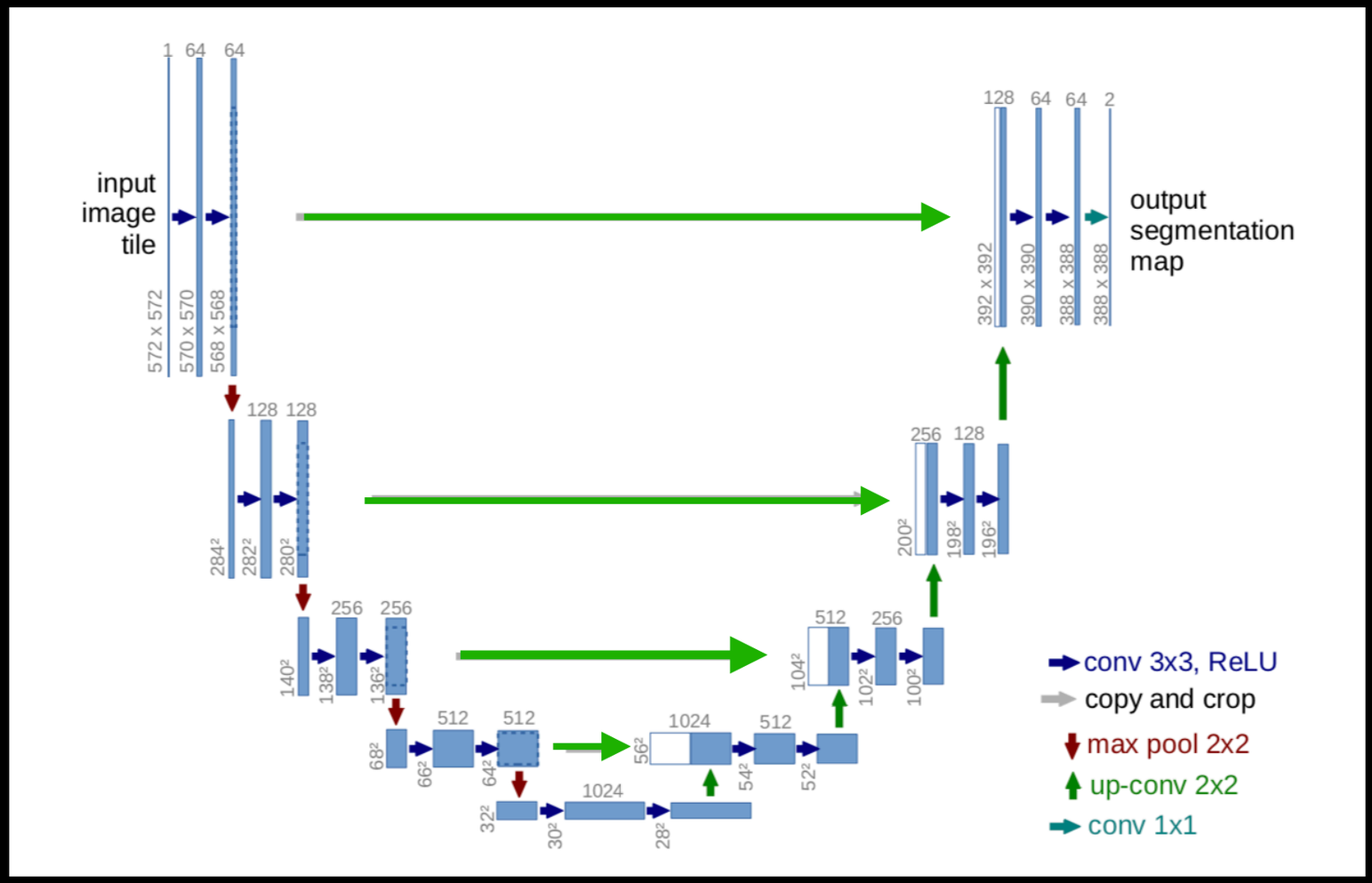
Comments

GT **FCN** **LDN** **Ensemble**

After various investigations, the authors concluded that an ensemble network (consisting of both FCN and the Learned Deconvolution Network) produced the best results.

In the next paper, we will see how the elements from both networks can be combined into a single, and more powerful architecture, eliminating the need for an ensemble network.

U-Net



This network was initially designed to segment microscope images.

Like the Learned Deconvolutional Network, the U-Net was designed to be a completely symmetrical encoder-decoder network (convolutional-deconvolutional network).

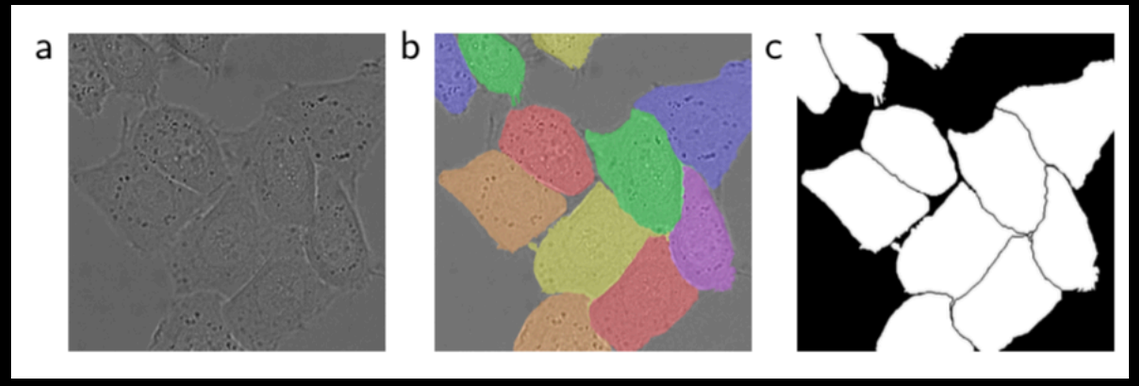
The contracting (downsampling) path in the U-Net captured context, while the expanding (upsampling) path in the U-Net captured localization.

In the original Learned Deconvolution Network, the bottleneck (between the encoder-decoder network) restricted the flow of information between the encoder and the decoder.

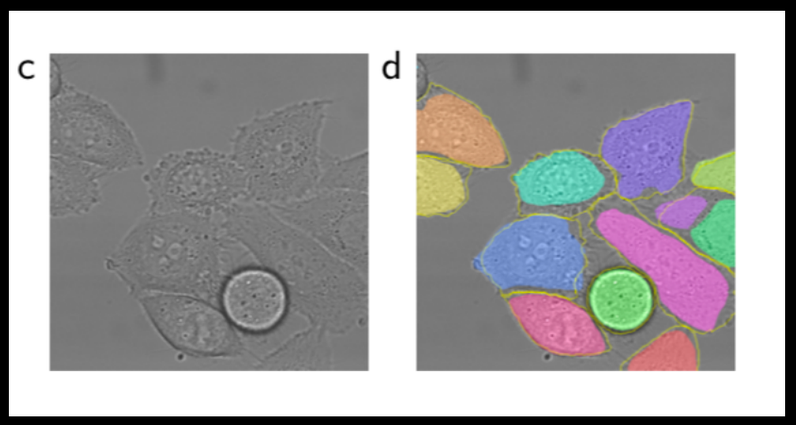
To remedy this, **feature maps in the decoder (downsampling paths) were directly copied to the encoder (upsampling paths), allowing the direct exchange of information** from the decoder network.

This tremendously improved the localization properties in the resulting segmentation map.

Sample results using the U-Net architecture for microscope images are shown below.

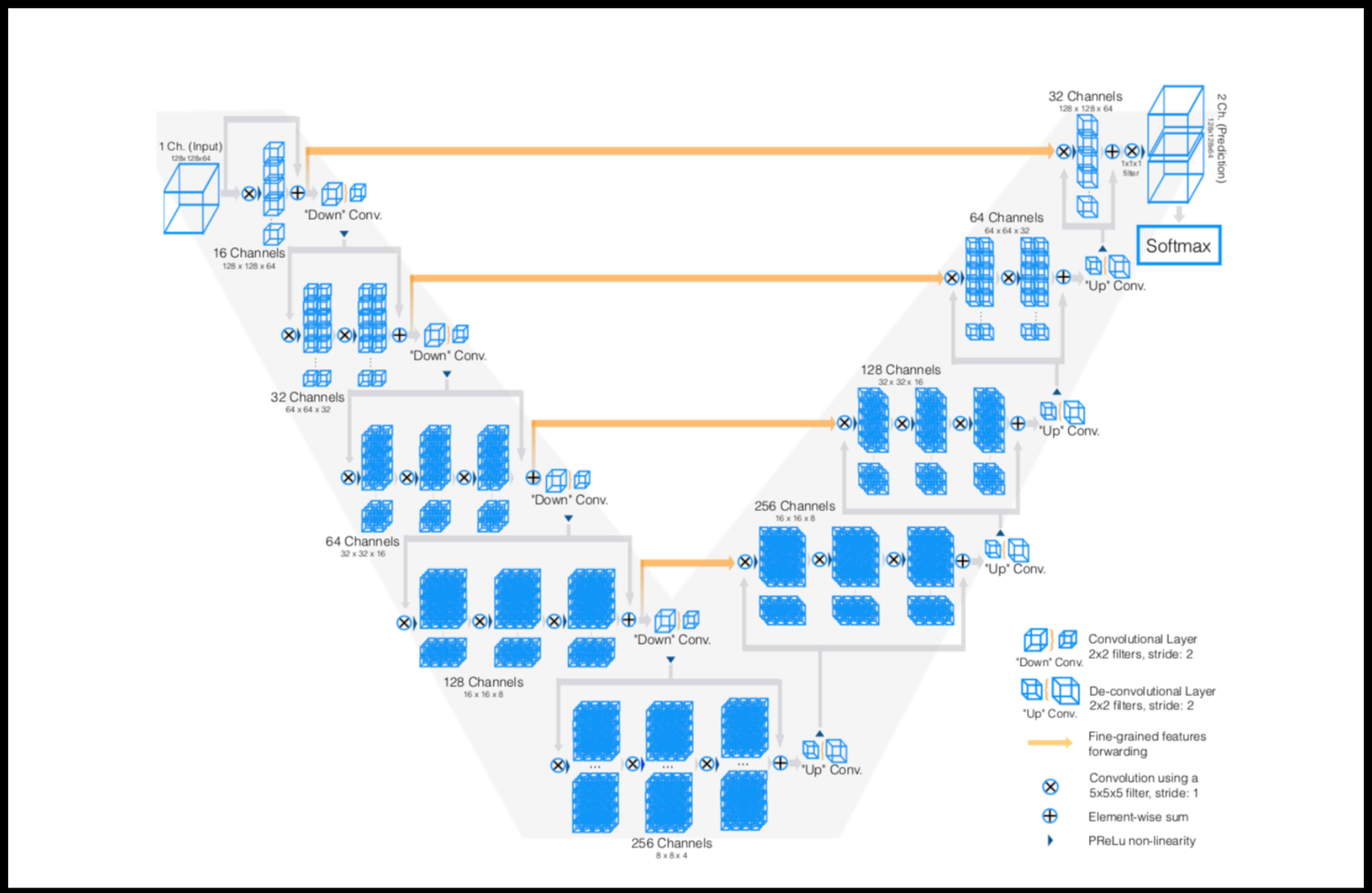


Sample result: Input image, GT, U-Net Binary Segmentation Mask



Sample result: Input image, U-Net Segmentation with GT Colors Super Imposed

V-Net



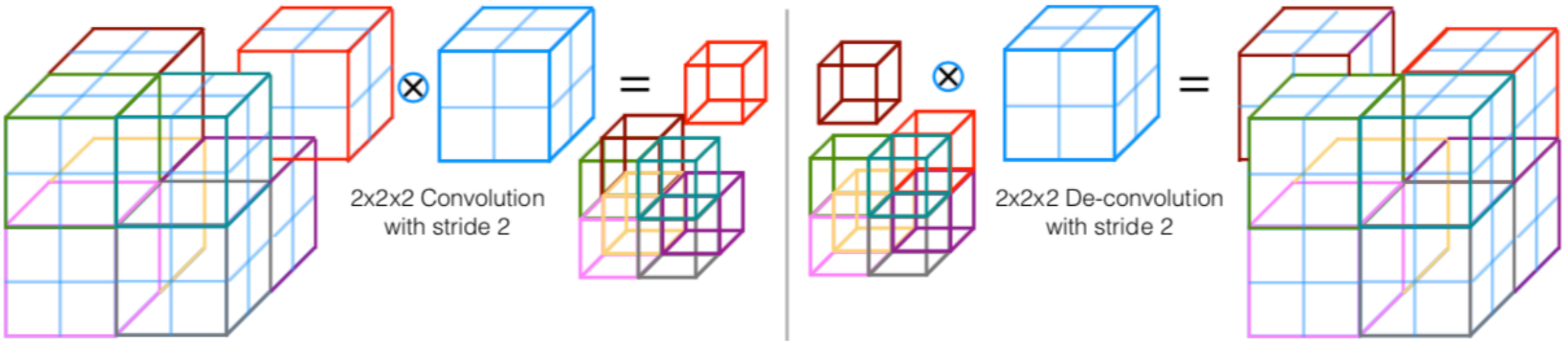
V-Nets were 3D variants of U-Nets, allowing for efficient 3D volumetric image data segmentation.

The first V-Nets were trained end to end on MRI prostate volumes.

Like the U-Net, the V-Net was symmetrical.

Like the U-Net, skip connnections were added between the downsampling and upsampling paths.

V-Net

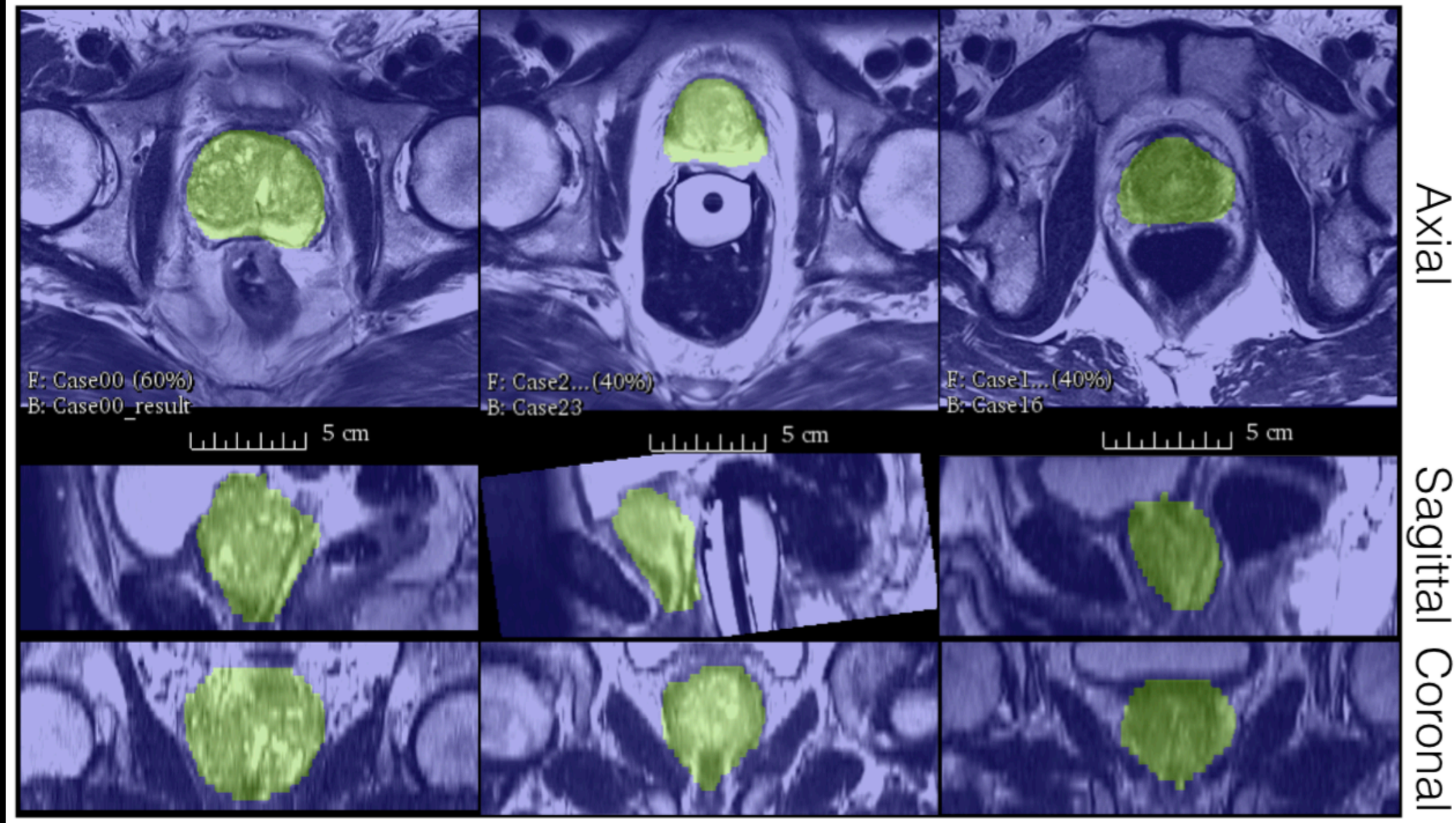


Unlike the U-Net, each convolutional layer in a V-Net also learned a residual function (shown in gray on the previous slide).

Unlike the U-Net, 2x2x2 strided convolutions and deconvolutions (shown above) were used in place of the 2x2 max pooling and the 2x2 upsampling found in the U-Net.

Unlike the U-Net, volumetric convolutions were used in place of spatial convolutions.

In addition, a novel objective function was also employed, to account for the significant imbalance between the number of background and foreground pixels in the prostate segmentation task.

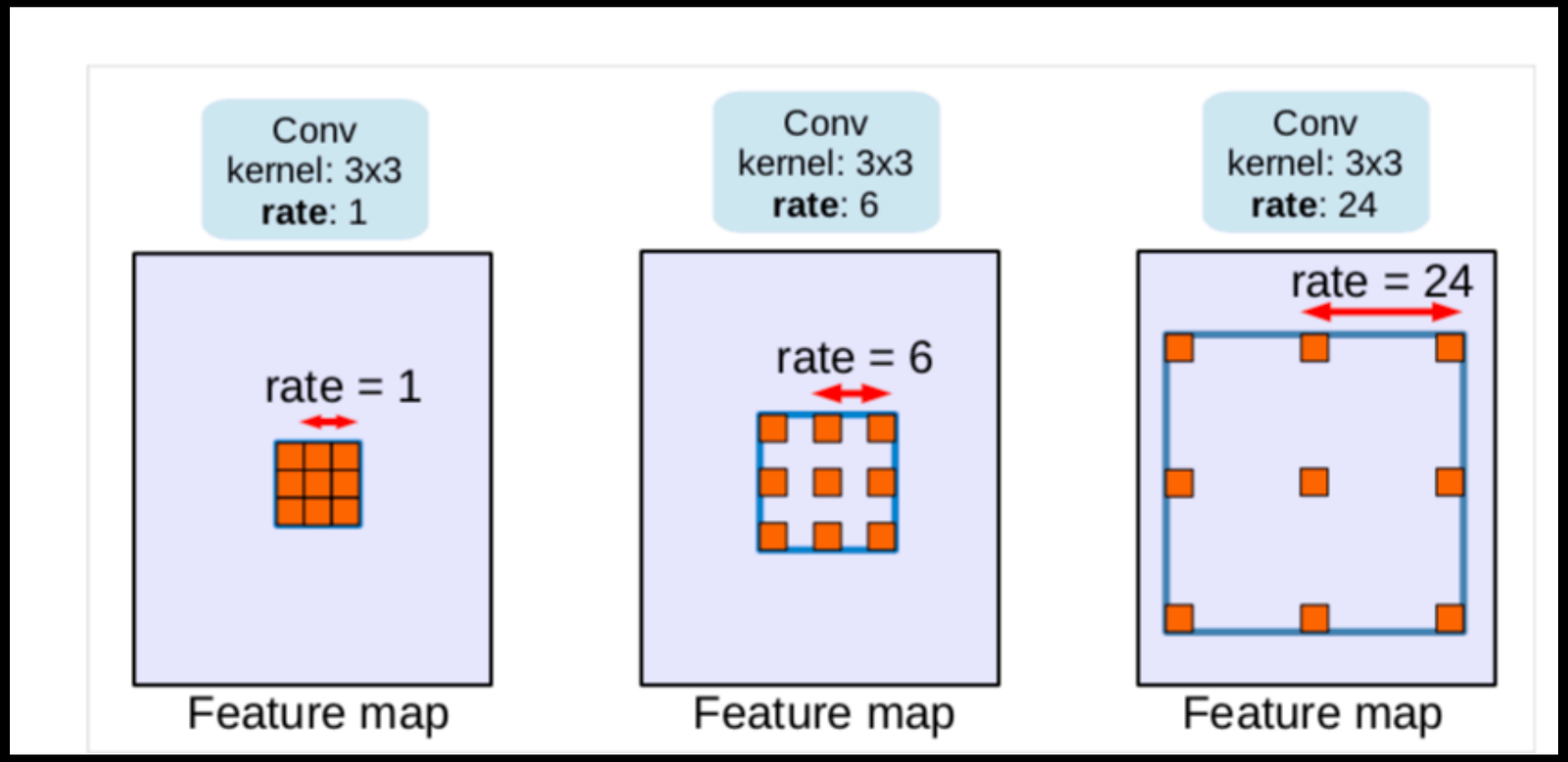


Sample result from the Promise 2012 dataset on the prostate segmentation task.

DeepLab

- DeepLab was an evolving architecture that yielded high semantic segmentation performance.
- In it’s first instatiation, DeepLab made use of atrous (dialated) convolutions coupled with conditional random fields (CRFs).
- In it’s second instantiation, multiscale information was infused, using spatial pyramid pooling (SPP).
- In it’s third instantiation, an coarse encoder-decoder network was incorporated into the existing architecture, to refine the segmentation results along object boundaries.

Atrous (Dilated) Convolution



Atrous = with holes

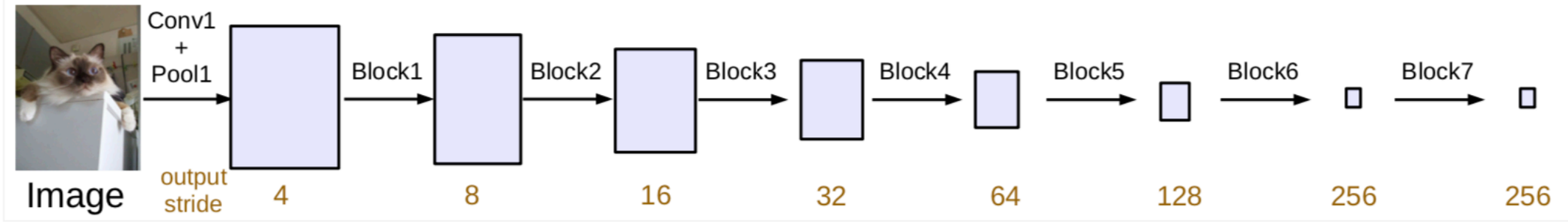
To create an atrous (dilated) convolution, the weights from a non-dilated kernel (left image) are incorporated into a higher order kernel (middle and right images).

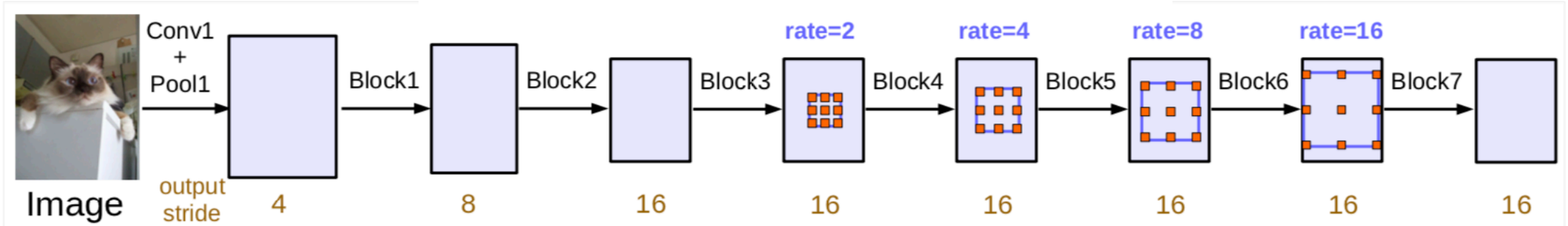
All other locations in the higher order kernel are assigned weights of value zero.

Atrous convolutions allowed the receptive field of the original convolution kernel to increase, without adding any additional weight parameters.

Atrous convolutions also allowed input tensors to be processed at different scales.

V1





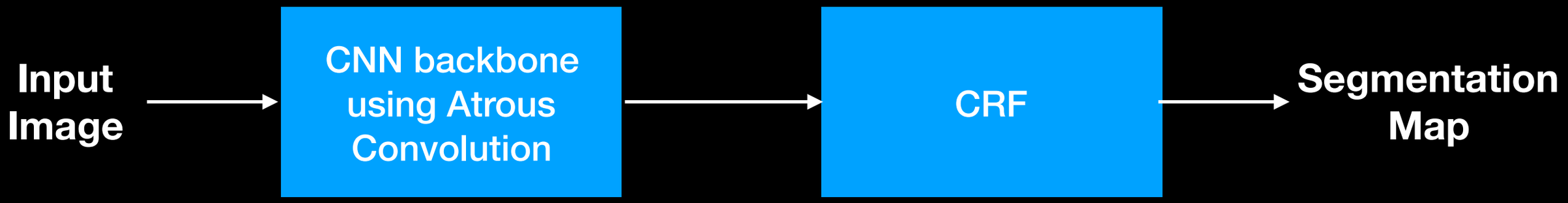
The top diagram illustrates the typical architecture associated with an image recognition task.

Here, semantic information is represented in the higher order blocks of low spatial resolution.

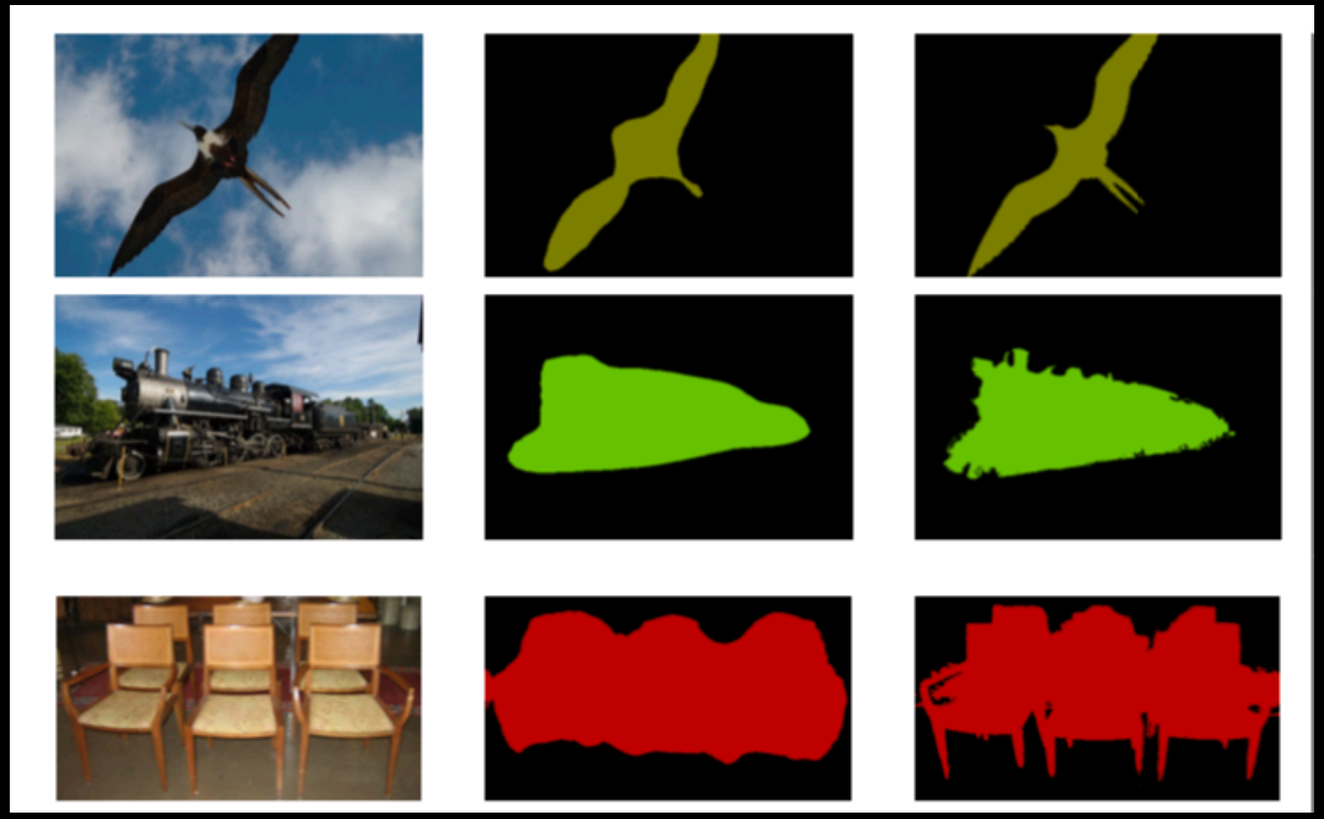
However, this was the opposite of what was required for semantic segmentation.

In the bottom diagram, atrous convolutions were introduced to prevent the continual reduction of spatial resolution.

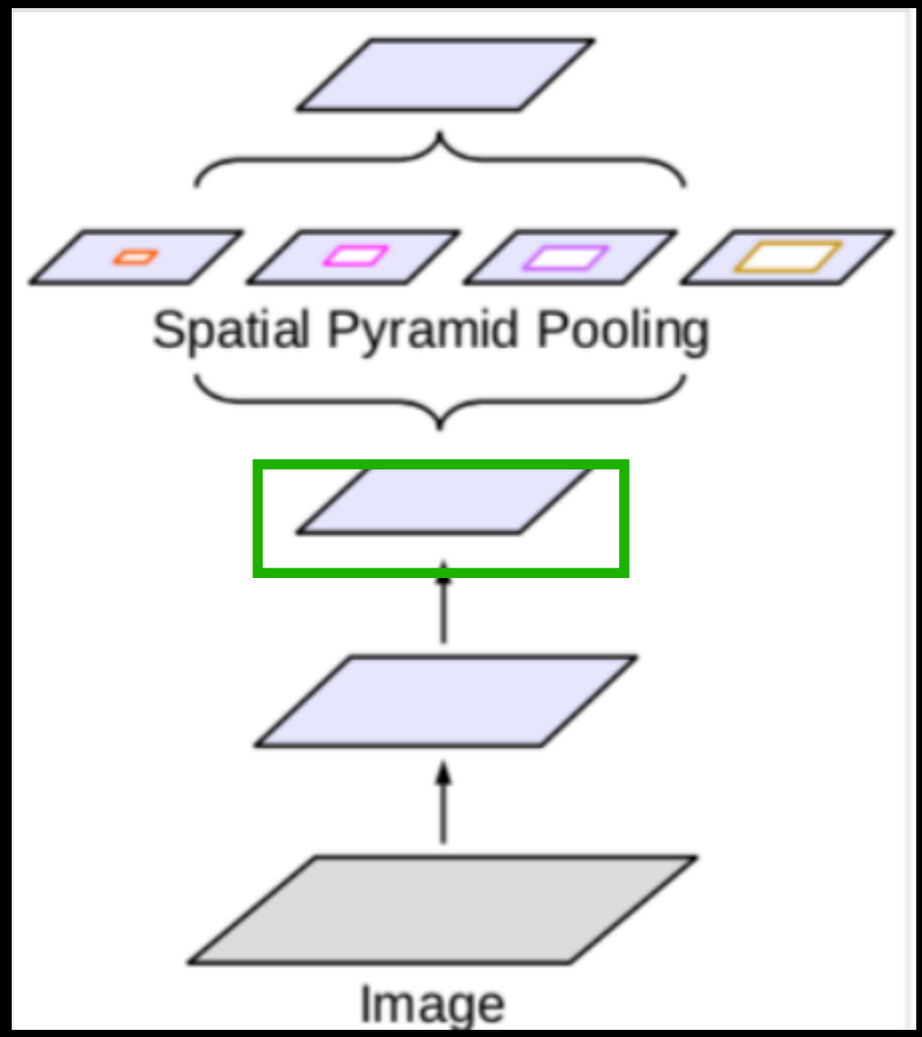
V1



A second Conditional Random Field (CRF) post processing module was then added, to generate an improved sementation maps.



Spatial Pyramid Pooling

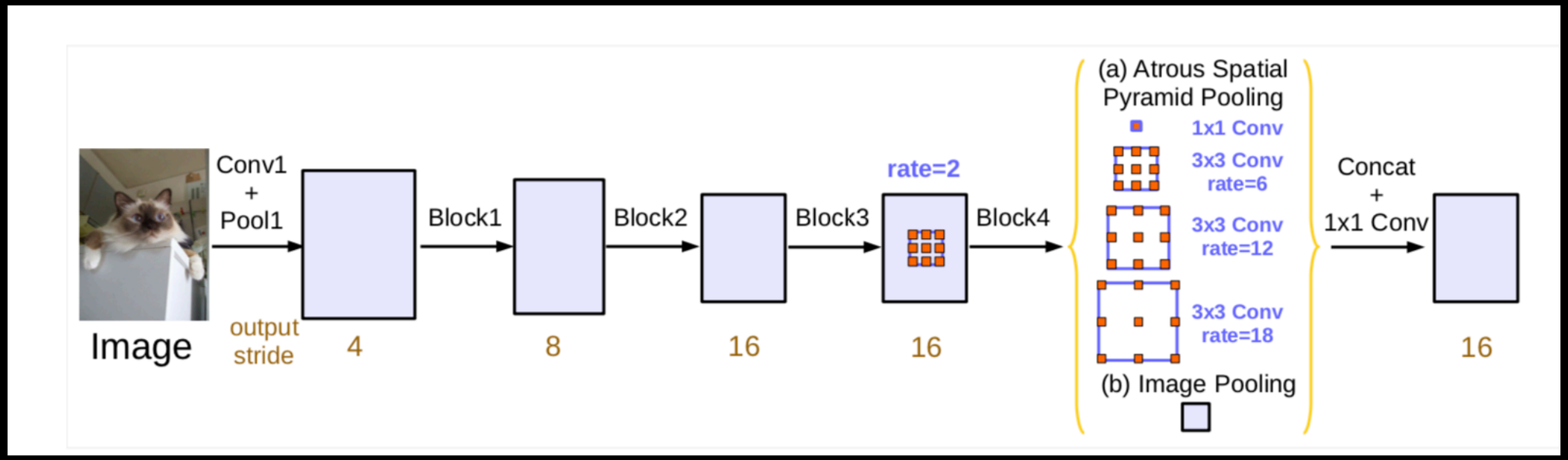


Spatial pyramid pooling computed and aggregated new representations for an input image or feature map.

In the figure shown above, atrous convolutions were applied to the final output feature map, producing a set of new maps containing multi-scale representations.

This technique was known as atrous spatial pyramid pooling (ASSP).

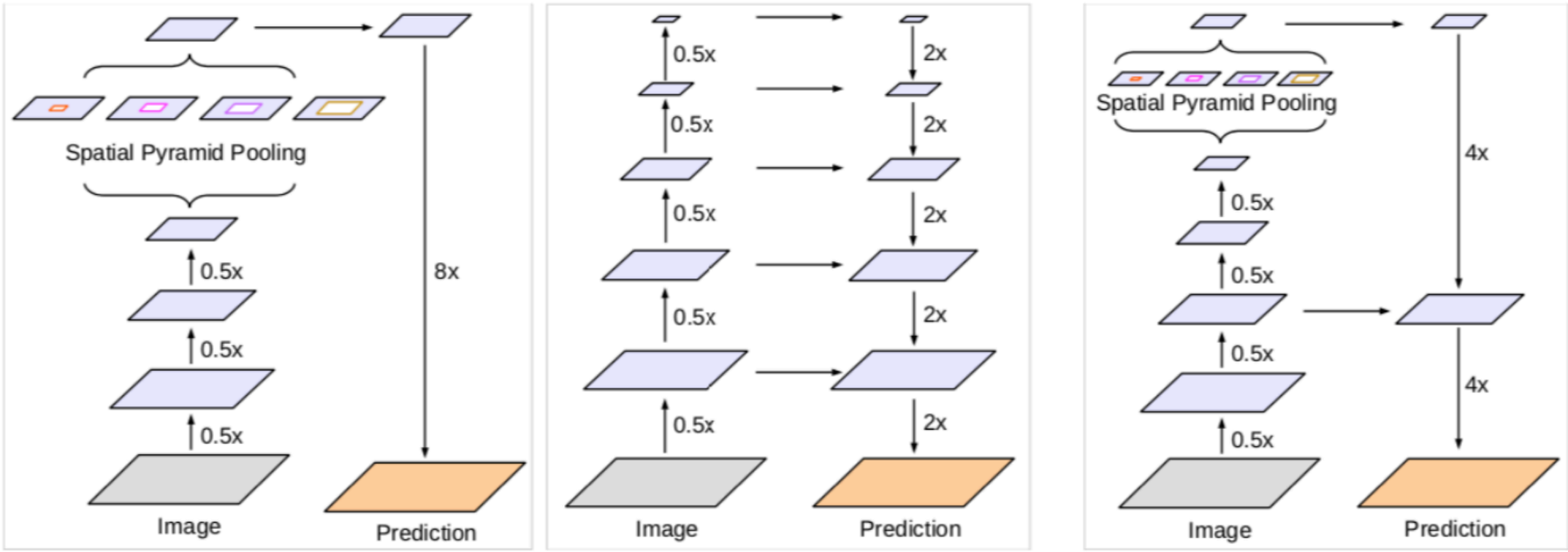
V2



V2: 1) atrous convolutions were employed in the backbone and 2) ASPP replaced the CRF module.

This new architecture outperformed the original V1 architecture.

V3

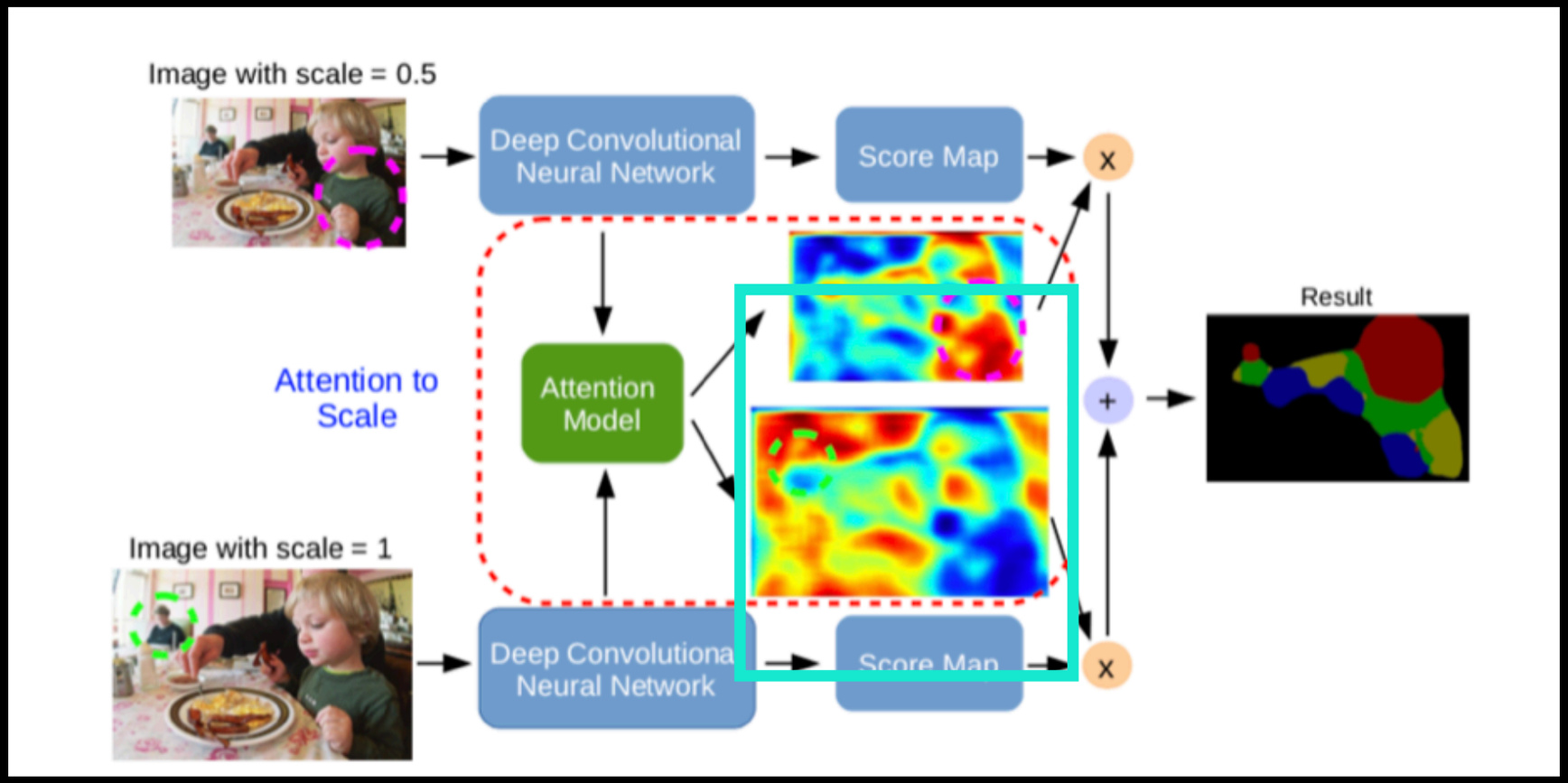


The authors then borrowed from the popular encoder-decoder architecture with skip connections (middle image).

Instead of directly interpolating the ASPP output (left image) as done in V2, the authors combined information from various encoder layers using skip connections (right image).

Based on the improved results, the power of the U-Net architecture (=skip connections) became clearly evident.

Attention To Scale



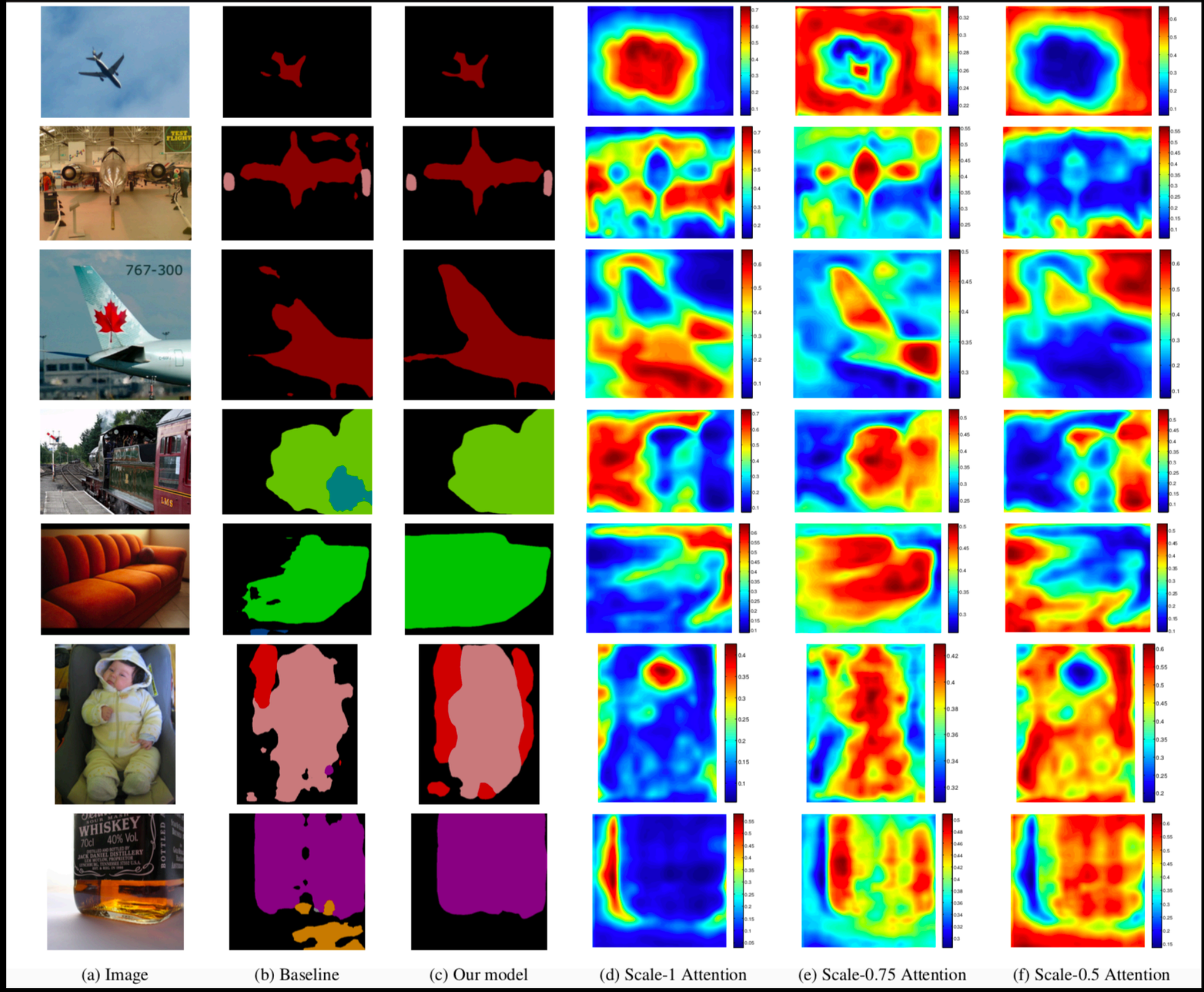
Attention based approaches focused attention to specific areas of the input signal.

For images, attention was focused spatially, temporally or on scale, if a multi-scale representation existed.

In the algorithm above, a multi-scale representation consisting of 2 score maps was generated.

An attention module then determined the appropriate weight map(s), for each corresponding score map(s).

Each score map was then multiplied (pixelwise) by the appropriate weight map and aggregated (via pixelwise summation) to produce a semantic segmentation map.



Baseline shows the original segmentation map without the attention module.

A multi-scale representation consisting of 3 score maps was used in the attention based approach.

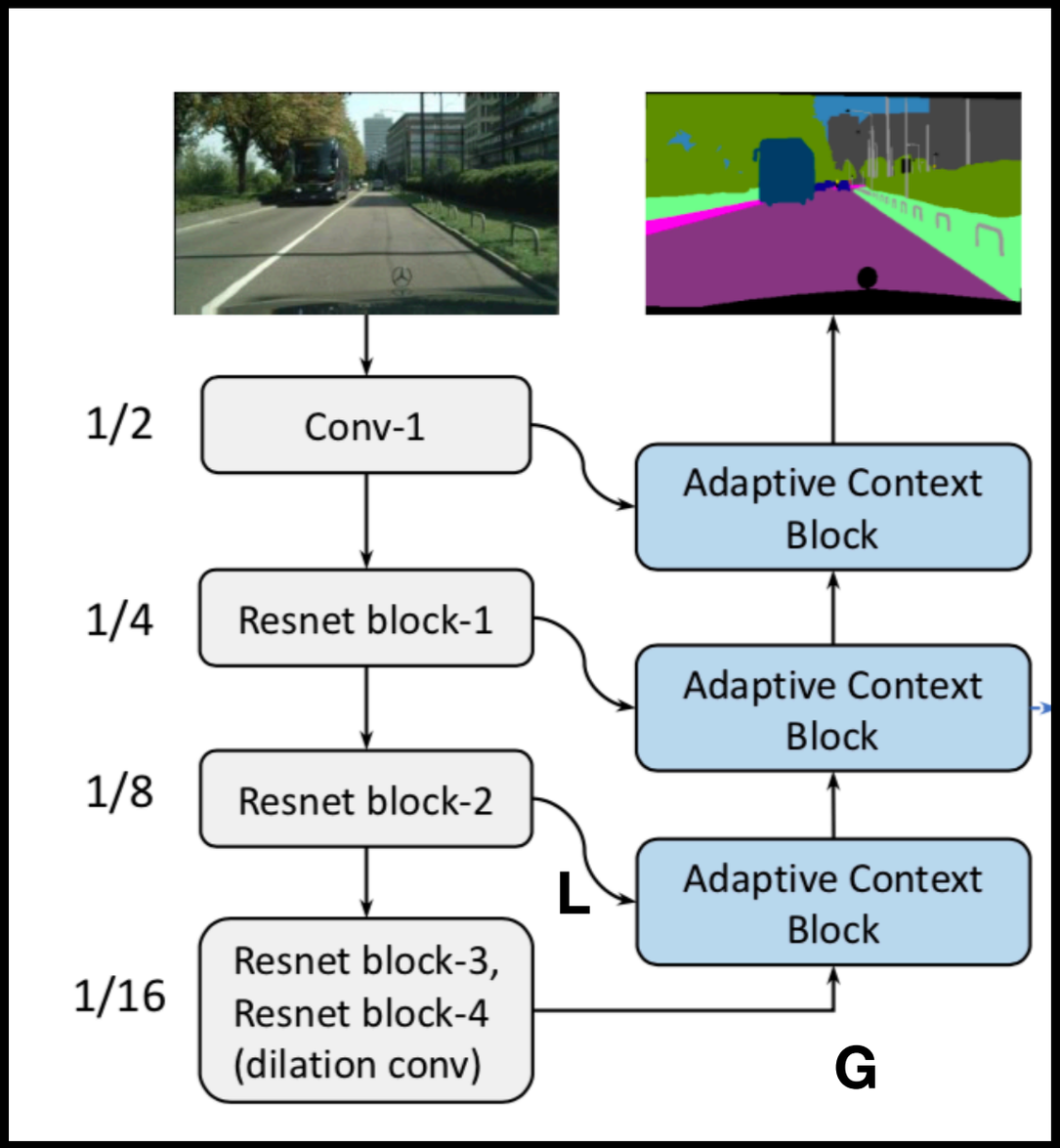
Comments

- The original DeepLab employed a Conditional Random Field (CRF) module to improve the segmentation map.

- In addition, many additional papers employed CRF's or Markov Random Fields (MRF) for the same purpose.

- However, as full end to end techniques began to produce improved segmentation maps, the benefits of including a CRF module disappeared.

Adaptive Context Network



The adaptive context network repeatedly ask the following question at each spatial location in a given feature map: “How close is the current activation value to the average activation value for this feature map?”

A new output map was then generated, that judiciously incorporated the answer to this question.

i.e. The output nap was produced by the Adaptive Context Block shown on the left.

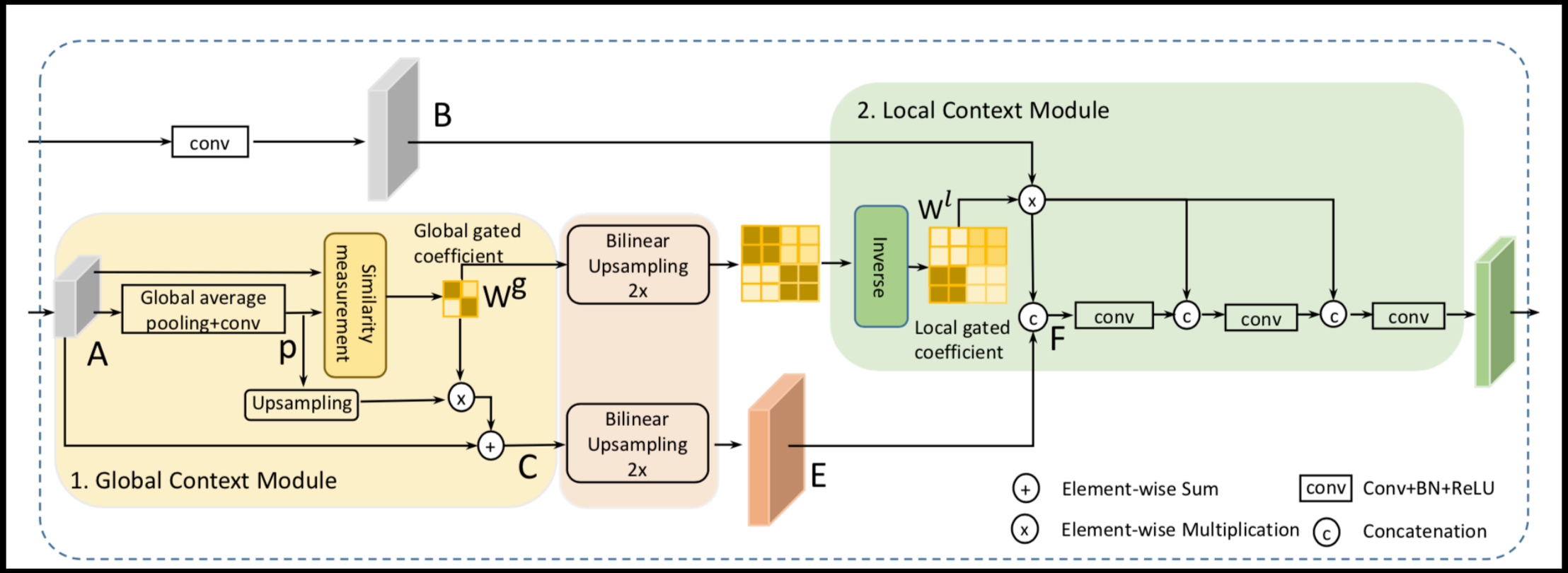
When applied over scale, improved segmentation maps were produced.

What was specifically happening in the Adaptive Context Block at the pixel level?

Adaptive Context Block

L

G



If the activation value at a specific pixel location was very similar to the average activation for the feature map, an output of 1 was assigned to that location. (If the value was very different, an output of 0 was assigned.)

From this, a weight matrix W^g, and it's compliment W^l = (1-W), were computed.

A weight matrix containing a lot of 1's denoted a very smooth feature map=slow varying.

In addition, locations in the weight matrix with values of 0 denoted locations of large "disagreement"=high frequency detail.

Adaptive Context Block

The weight matrices were then used to “scale” the two input tensors to the adaptive context block.

The subsequent information was then aggregated, to produce a new output map.

This process was then repeated for feature maps of increasing resolutions.

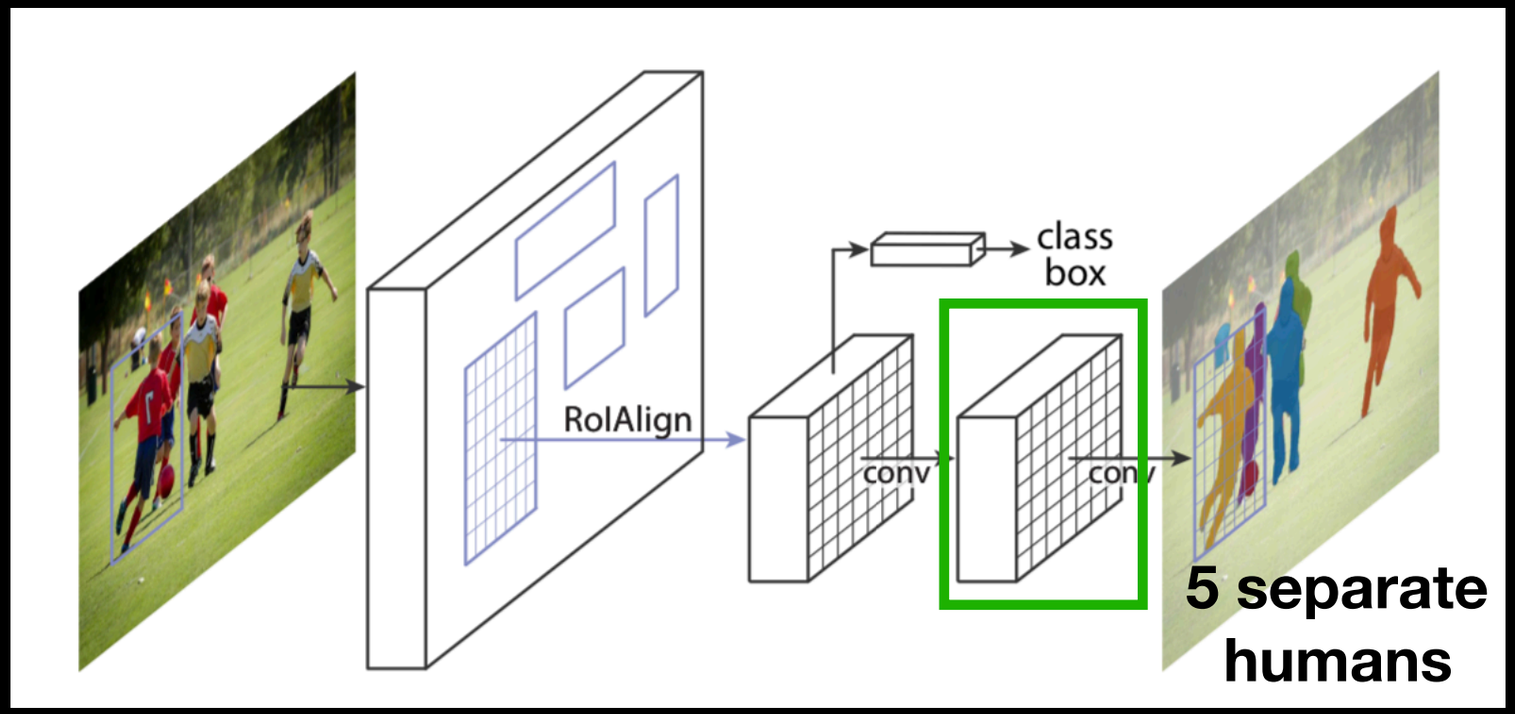
Hence, the entire algorithm could be viewed as a process for adding refinements of increasing acuity, as the algorithm progressed from low resolution feature maps to high resolution feature maps.

The end result, was a high accuracy segmentation map.

ACNet

GT

Mask R-CNN



Whereas the previous methods focused on semantic segmentation (i.e. 5 humans were segmented as a single “human object” instead of 5 separate humans), Mask R-CNN performed instance segmentation.

Mask R-CNN was built on the Faster R-CNN architecture for object detection, by **adding an additional output head** associated with instance segmentation.

To see a thorough writeup on Faster R-CNN, please refer to the SL_ObjectDetection slides.

Algorithm (Backbone)

- FCN - VGG backbone
- Deconvolution Network - VGG backbone (with mirror architecture for the decoder)
- U-Net, V-Net - “VGG/CNN like” backbone (with mirror architecture for the decoder)
- DeepLab - “VGG like” backbone with Atrous Convolutions (and Atrous Spatial Pyramid Pooling)
- Attention - DeepLab (with Attention Module)
- Context - ResNet (with Adaptive Context Block)
- Mask R-CNN - ResNeXt FPN (with Segmentation Head)

Algorithm

- FCN - Segmentation outputs were obtained by combining neighboring feature map outputs that differed in resolution by 2x.

- Deconvolution Networks - Provided a more structured approach for combining muti-resolution information, by employing a decoder network.

- U-Net, V-Net - Improved upon deconvolution networks, by adding skips connection from the encoder network to the decoder network, producing segmentation maps with improved localization.

- DeepLab - (Initially) provided a different to look at the problem, by introducing multi-resolution capabilities via spatial pyramid pooling and atrous convolutions. In the end, DeepLabV3+ possessed a strong resemblence to a U-Net, using atrous convolutions.

- Attention - Focused “attention” on applying the appropriate weights to the different feature maps at various scales. The weights were learned from the data.

- Context - Used an “ad hoc” formula for computing the weights. Context then divided the problem into a base component and a residual component. With increasing resolution, improved segmentation maps resulted as more residual details were added. Internal convolutional layers provided additional “flexibility / memorization” for aggregating the low frequency and high frequency information.

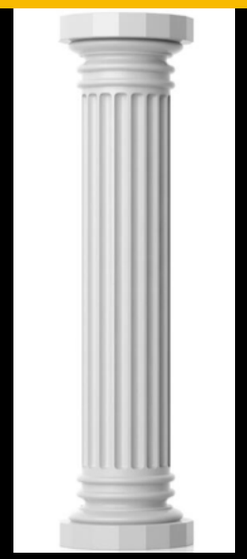
Comments

- GANS have also been used to create segmentation maps. However, they have been less successful operating in the segmentation space.

- RNN's, LSTM, GRU's, etc. have also been used to create segmentation maps. However, they have also been less succcessful operating in the segmentation space.

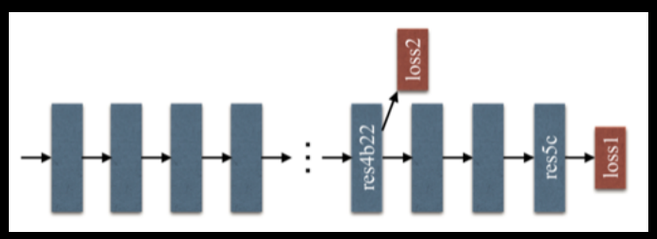
- As mentioned earlier, CRF's and MRF's were also initially used to improve segmentation maps.

Loss Function



Loss Function

- FCN - Cross Entropy Loss
- Deconvolution Network - Cross Entropy Loss
- U-Net, V-Net - Cross Entropy Loss, with sample reweighting
- DeepLab - Cross Entropy Loss
- Attention - (1 + S) Cross Entropy Loss Functions (a loss function at the output and a loss function at each scale)
- Context - Cross Entropy Loss (loss1) + Auxilary Loss (loss2)



- Mask R-CNN - A 4 term Loss Function used in Faster R-CNN consisting of Cross Entropy and Regression Losses

References

- Adaptive Document Image Binarization, Savoula and Pietkainen, Pattern Recognition 2000
- Semantic Segmentation using Regions and Parts, Arbelaez, et. al., CVPR 2012
- Fully Convolutional Networks for Semantic Segmentation, Long, et. al., CVPR 2015
- Learning Deconvolution Network for Semantic Segmentation, Noh, et. al., CVPR 2015
- U-Net: Convolutional Network for Biological Image Segmentation, Ronneberger, et. al., Int. Conference on Medical Image Computing, 2015
- V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, Fourth Int. Conference on 3D Vision, 2016
- Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, ECCV 2018
- Attention to Scale: Scale Aware Semantic Image Segmentation, Chen, et. al., CVPR 2016
- Expectation Maximization Attention Networks for Semantic Segmentation, Li, et. al., CVPR 2019
- Adaptive Context Network for Scene Parsing (AC Net) Fu, et. al., CVPR 2019
- Object Contextual Representations (OCR), Yuan, et. al., ECCV 2020
- Mask R-CNN, He, et. al., CVPR 2017