MA-INF 2221 - Seminar Visual Computing

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

Can Yuce[1]
yuce@uni-bonn.de

[1]Department of Informatics
University of Bonn

July 4, 2017

universitätbonn

Outline

universitätbonn

Semantic Image Segmentation

Semantic segmentation is the task of

- recognizing/delineating objects,
- and classifying each pixel in an image.

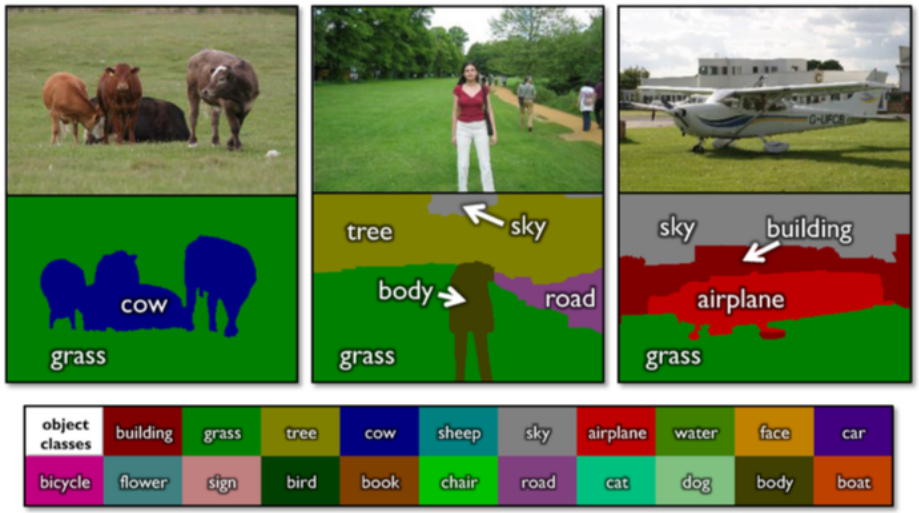


Figure 1: TextonBoost applied to the MSRC 21 Class Database. Example semantic segmentation results[1].

universitätbonn

[1] Shotton, Jamie, et al. "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context."

Why is Semantic Segmentation Important?

Road scene understanding

- Useful for autonomous navigation of cars and drones.

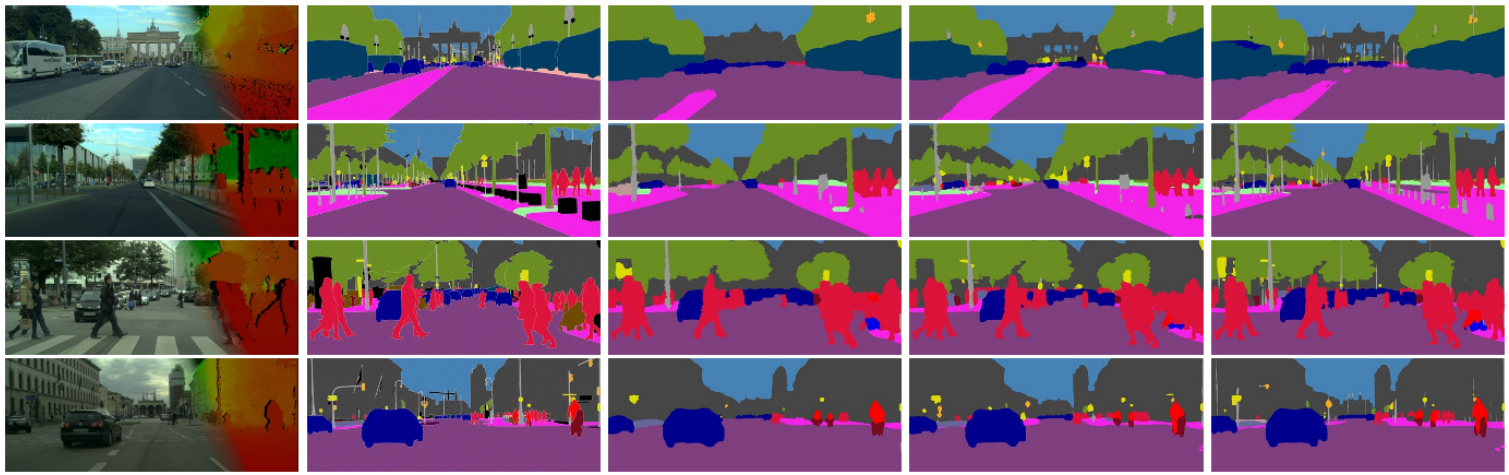


Figure 2: The Cityscapes Dataset for Semantic Urban Scene Understanding[2].

universitätbonn

[2] See more at `https://www.cityscapes-dataset.com`.

Why is Semantic Segmentation Important?

Medical purposes

- Segmenting tumours, organs, dental cavities etc.

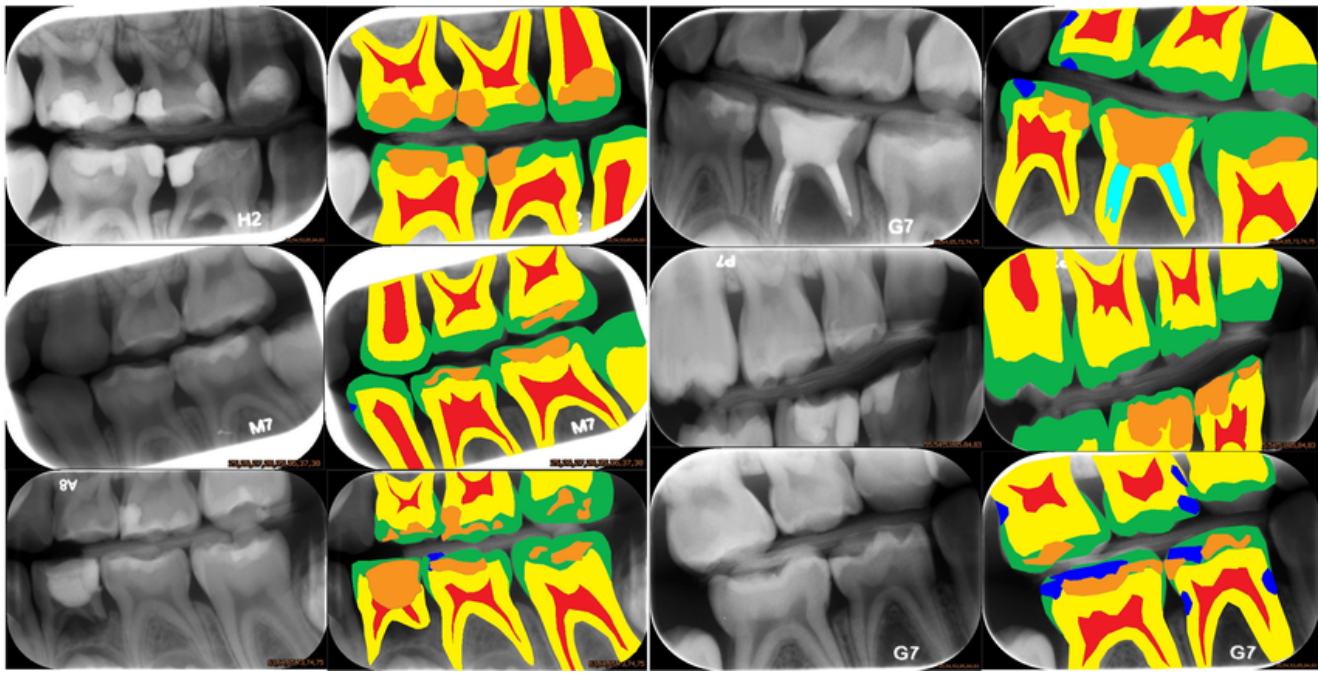


Figure 3: Segmentation of dental cavities[3].

[3]Wang, Ching-Wei, et al. "A benchmark for comparison of dental radiography analysis algorithms."

universitätbonn

Why is Semantic Segmentation Important?

Robot Vision

- To automatically segment objects from its background is important for active agents to interact(i.e. grasping, manipulation) effectively in the real world.

Figure 4: A Vacuum Grapper tries to grasp objects from the shelf [4].

universitätbonn

[4] http://appliedrobotics.blogspot.de/2015_08_01_archive.html

Semantic Segmentation Before Deep Learning

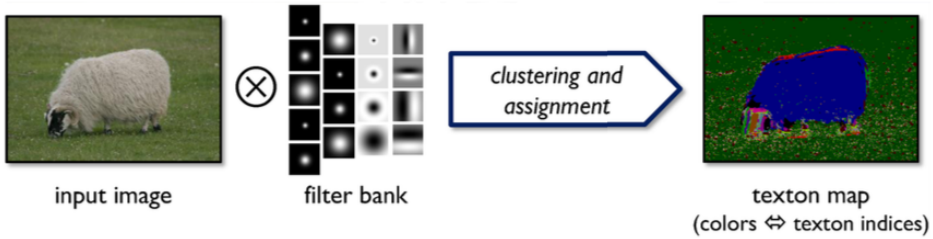


Figure 5: The process of image textonization[1].

Most of the successful semantic segmentation systems used:

- Hand-crafted features,
- Operations on pixels or superpixels,
- Flat classifiers, such as Boosting/Ensemble methods,
- Models incorporating local evidence with interactions between label assignments.
- Hierarchical classifiers incorporating richer information from context.

universitätbonn

Semantic Segmentation Before Deep Learning

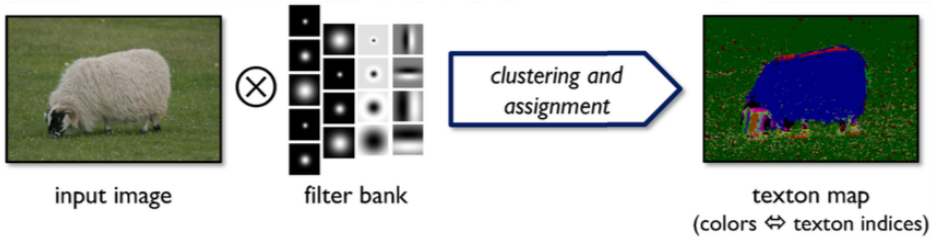


Figure 5: The process of image textonization[1].

Most of the successful semantic segmentation systems used:

- Hand-crafted features,
- Operations on pixels or superpixels,
- Flat classifiers, such as Boosting/Ensemble methods,
- Models incorporating local evidence with interactions between label assignments.
- Hierarchical classifiers incorporating richer information from context.

universitätbonn

Semantic Segmentation Before Deep Learning

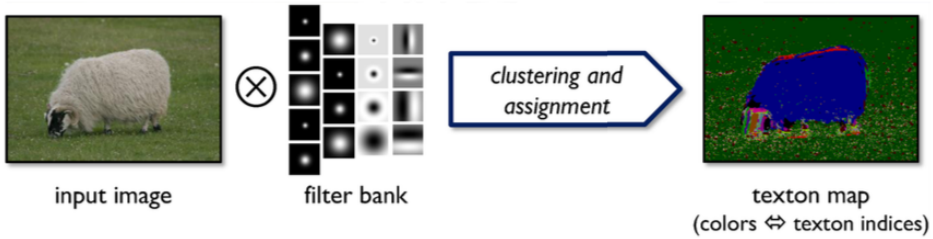


Figure 5: The process of image textonization[1].

Most of the successful semantic segmentation systems used:

- Hand-crafted features,
- Operations on pixels or superpixels,
- Flat classifiers, such as Boosting/Ensemble methods,
- Models incorporating local evidence with interactions between label assignments.
- Hierarchical classifiers incorporating richer information from context.

universitätbonn

Semantic Segmentation Before Deep Learning

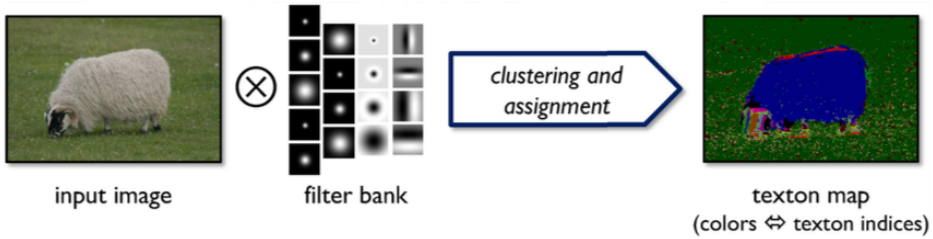


Figure 5: The process of image textonization[1].

Most of the successful semantic segmentation systems used:

- Hand-crafted features,
- Operations on pixels or superpixels,
- Flat classifiers, such as Boosting/Ensemble methods,
- Models incorporating local evidence with interactions between label assignments.
- Hierarchical classifiers incorporating richer information from context.

universitätbonn

Semantic Segmentation Before Deep Learning

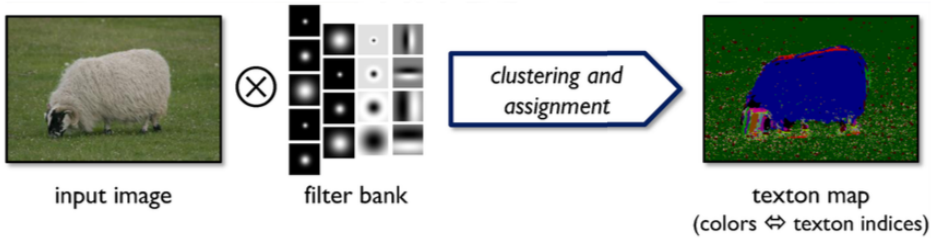


Figure 5: The process of image textonization[1].

Most of the successful semantic segmentation systems used:

- Hand-crafted features,
- Operations on pixels or superpixels,
- Flat classifiers, such as Boosting/Ensemble methods,
- Models incorporating local evidence with interactions between label assignments.
- Hierarchical classifiers incorporating richer information from context.

universitätbonn

Flat Classifiers

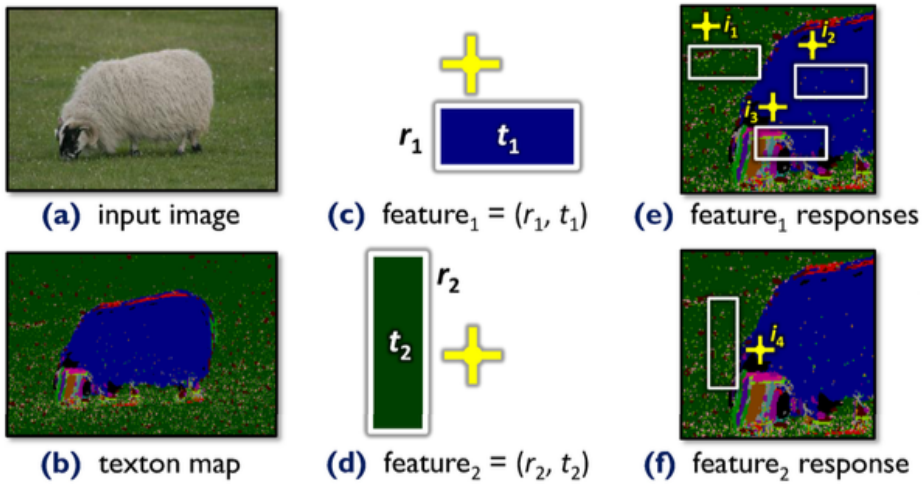


Figure 6: Each weak learner is a decision stump based on the feature response[5].

Boosting/Ensemble Methods,

- Random Forests,
- Support Vector Machines etc.

[5] Shotton, Jamie, et al. "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context."

CRF/MRF based Models

- Conditional/Markov Random Fields (CRF/MRF)are broadly used in semantic segmentation tasks.
- Unary classifiers on single pixel/superpixels,
- Smoothness terms that maximize label agreement between similar pixels.

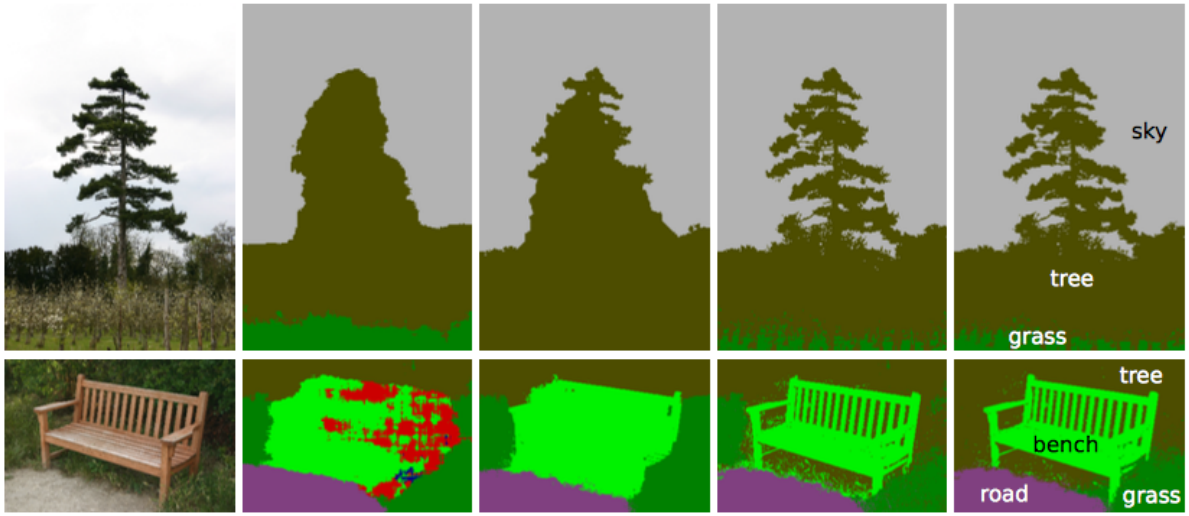


Figure 7: Pixel-level classification with a fully connected CRF.[6].

universitätbonn

[6] Philipp Krähenbühl, Vladlen Koltun. "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials."

Hierarchical Models

These methods benefit from allowing context to be incorporated at multiple levels of quantisation[7]:

- The lowest layer of the image represents the pixel layer,
- The middle layer potentials defined over super-pixels or segments
- The third layer represents our hierarchical terms.

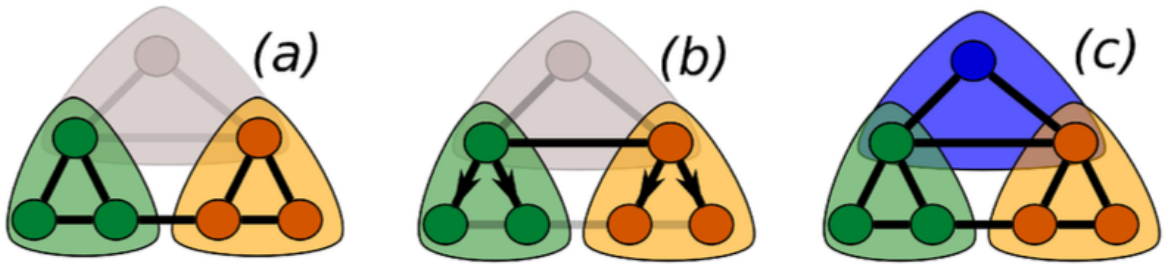


Figure 8: Pixel-level classification with a fully connected CRF.

[7] Russell, Chris, et al. "Associative hierarchical crfs for object class image segmentation."

Deep Learning and Semantic Segmentation

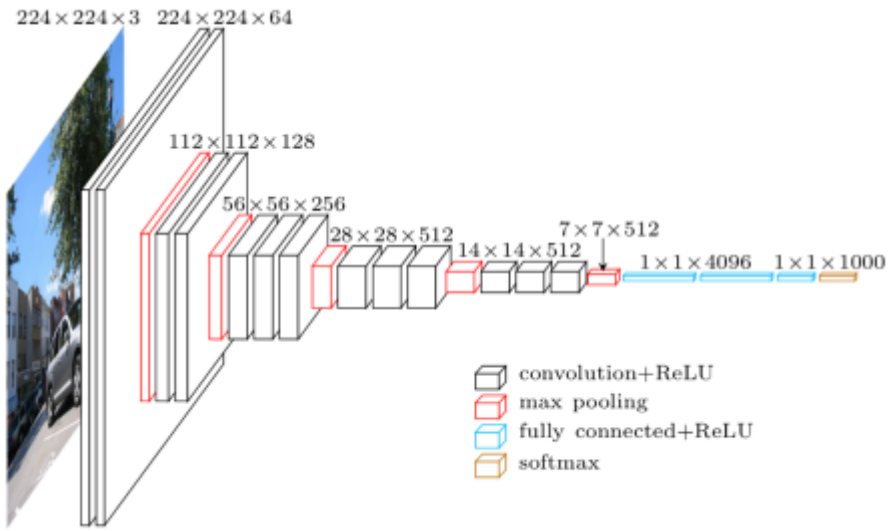


Figure 9: VGG-16 network layer by layer[8].

Deep Convolutional Neural Networks (DCNN) in classification:

- Input: The whole image,
- Output: Probability of each class in image(a car, person, dog etc.)
- Not directly applicable to the Semantic Segmentation problem.

universitätbonn

[8] Simonyan, Karen, et al. "Very deep convolutional networks for large scale image recognition."

Deep Learning and Semantic Segmentation

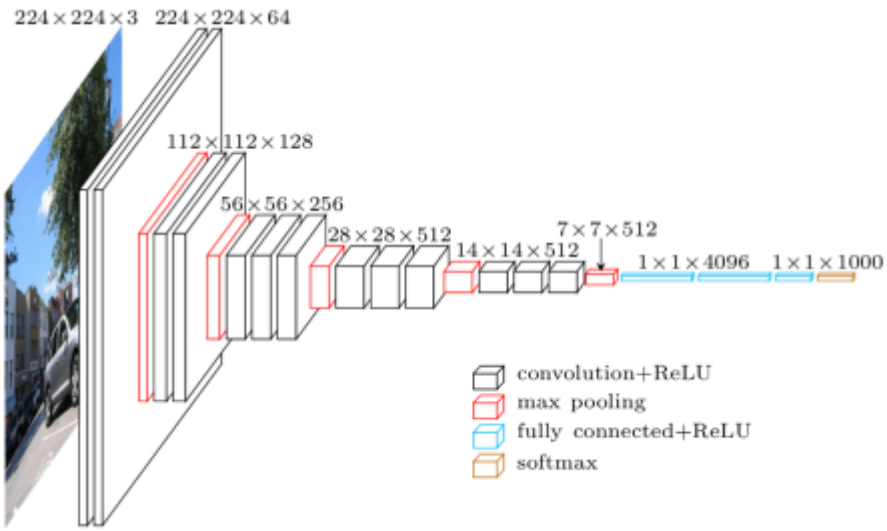


Figure 9: VGG-16 network layer by layer[8].

Deep Convolutional Neural Networks (DCNN) in classification:

- Input: The whole image,
- Output: Probability of each class in image(a car, person, dog etc.)
- Not directly applicable to the Semantic Segmentation problem.

[8] Simonyan, Karen, et al. "Very deep convolutional networks for large scale image recognition."

universitätbonn

DCNN + Region Proposals

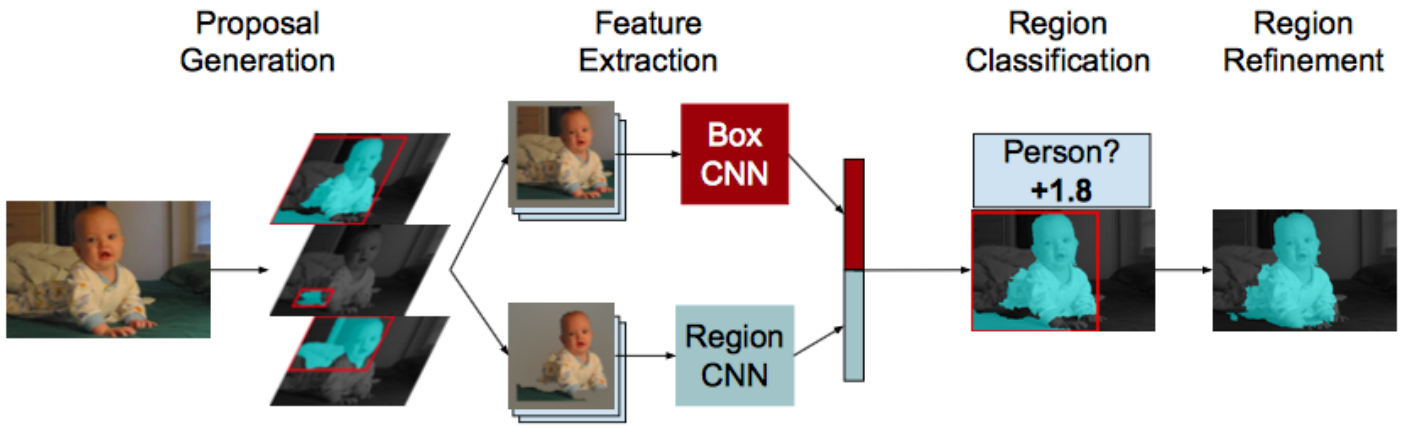


Figure 10: Simultaneous Detection and Segmentation Pipeline.[9].

1. Employing a cascade of bottom-up image segmentation proposals/features,
   - Fail to recover from any of its errors.
2. DCNN-based regional classification,
3. Refining proposals.

universitätbonn

[9] Hariharan, Bharath, et al. "Simultaneous detection and segmentation, 2014."

DCNN + Region Proposals

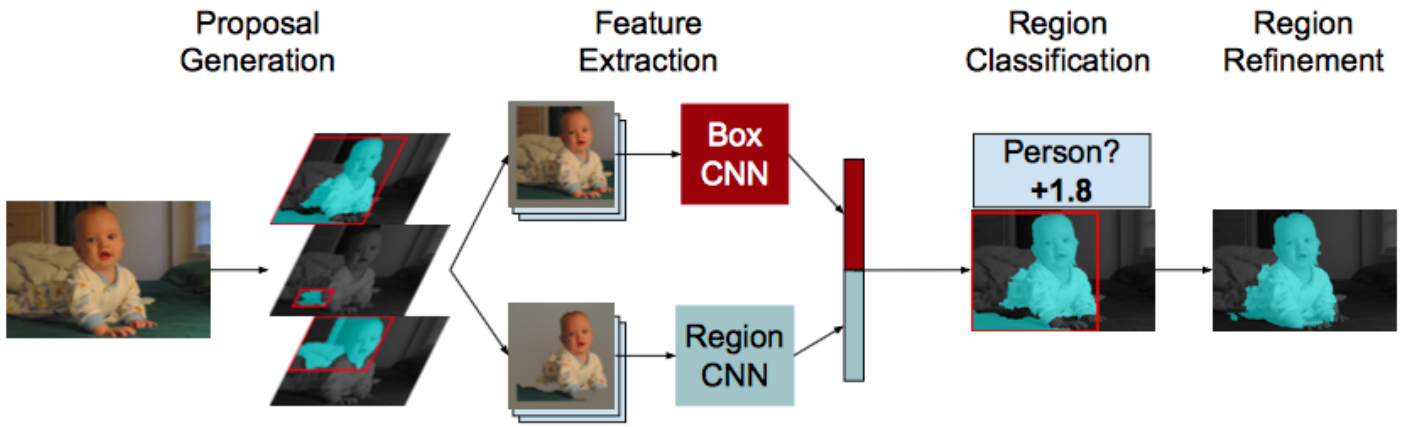


Figure 10: Simultaneous Detection and Segmentation Pipeline.[9].

1. Employing a cascade of bottom-up image segmentation proposals/features,
   - Fail to recover from any of its errors.
2. DCNN-based regional classification,
3. Refining proposals.

universitätbonn

[9] Hariharan, Bharath, et al. "Simultaneous detection and segmentation, 2014."

DCNN + Region Proposals

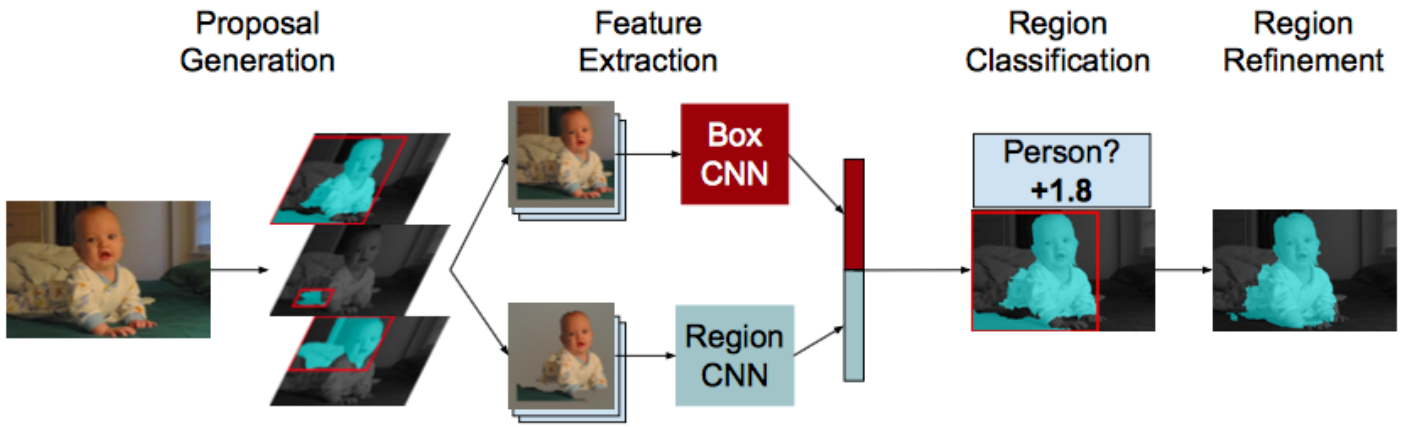


Figure 10: Simultaneous Detection and Segmentation Pipeline.[9].

1. Employing a cascade of bottom-up image segmentation proposals/features,
   - Fail to recover from any of its errors.
2. DCNN-based regional classification,
3. Refining proposals.

universitätbonn

[9] Hariharan, Bharath, et al. "Simultaneous detection and segmentation, 2014."

DCNN Features + Segmentation Proposal

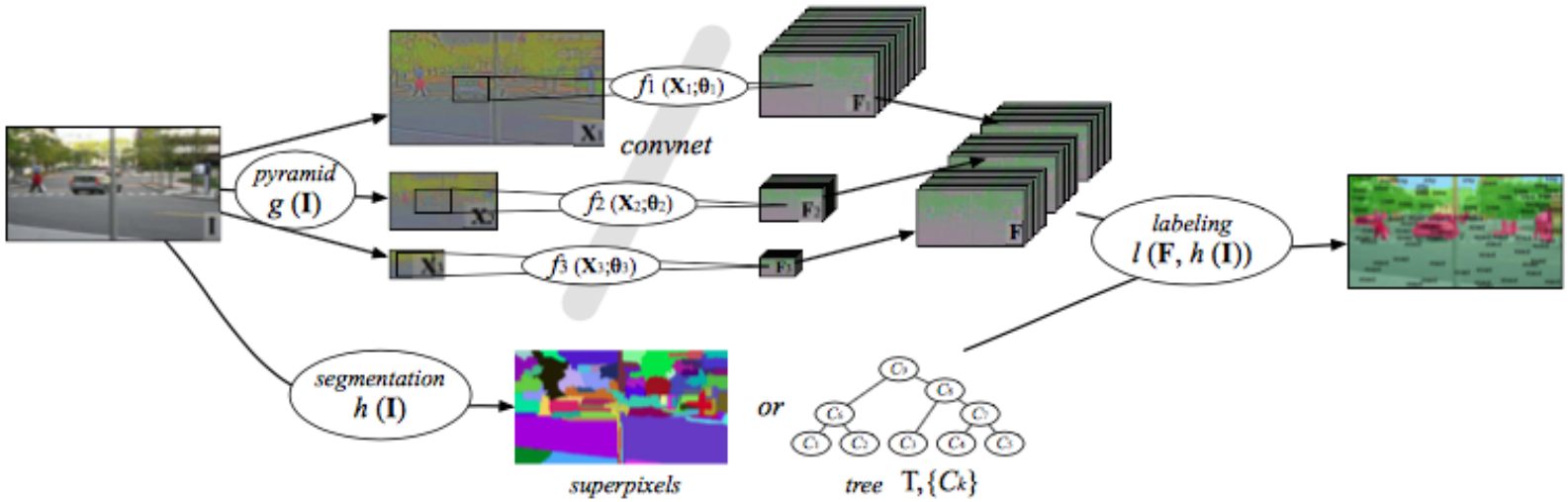


Figure 11: Scene labeling system[10].

1. Employ DCNNs at multiple image resolutions,
2. Construct a segmentation tree to smooth the prediction results,
3. Segmentation algorithms are decoupled from the DCNN classifier's results.
   - Risking commitment to premature decisions.

universitätbonn

[10] Farabet, Clement, et al. "Learning hierarchical features for scene labeling (2013)."

DCNN Features + Segmentation Proposal

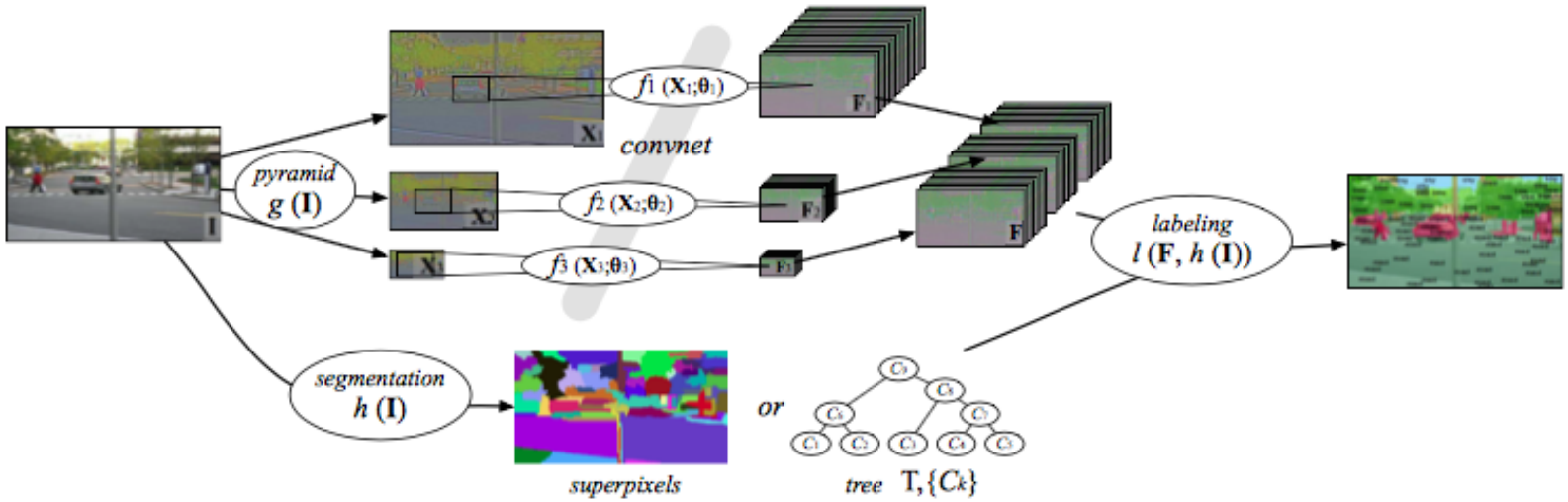


Figure 11: Scene labeling system[10].

1. Employ DCNNs at multiple image resolutions,
2. Construct a segmentation tree to smooth the prediction results,
3. Segmentation algorithms are decoupled from the DCNN classifier's results.
   - Risking commitment to premature decisions.

universitätbonn

[10] Farabet, Clement, et al. "Learning hierarchical features for scene labeling (2013)."

DCNN Features + Segmentation Proposal

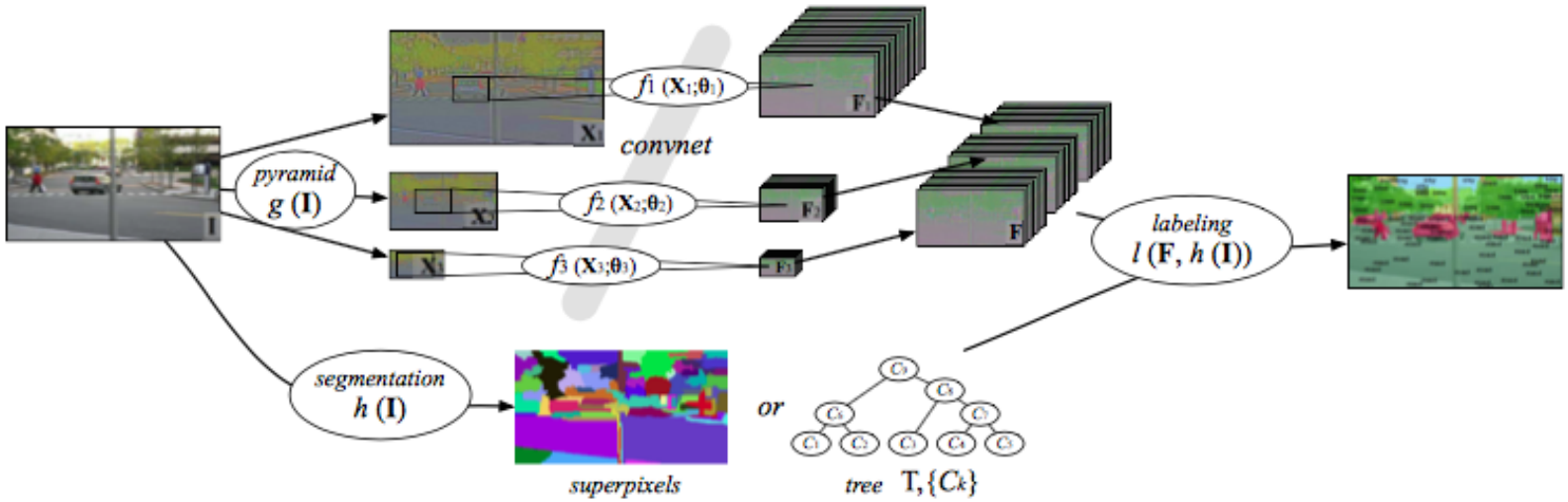


Figure 11: Scene labeling system[10].

1. Employ DCNNs at multiple image resolutions,
2. Construct a segmentation tree to smooth the prediction results,
3. Segmentation algorithms are decoupled from the DCNN classifier's results.
   - Risking commitment to premature decisions.

universitätbonn

[10] Farabet, Clement, et al. "Learning hierarchical features for scene labeling (2013)."

Dense Prediction

- Arbitrary input size: Take input of arbitrary size and produce correspondingly-sized output efficiently.
- Transforming all the fully connected layers to convolutional layers enables a classification net to output a heatmap[11]).
- Transfer learning using pretrained DCNN(i.e. VggNet, Resnet).

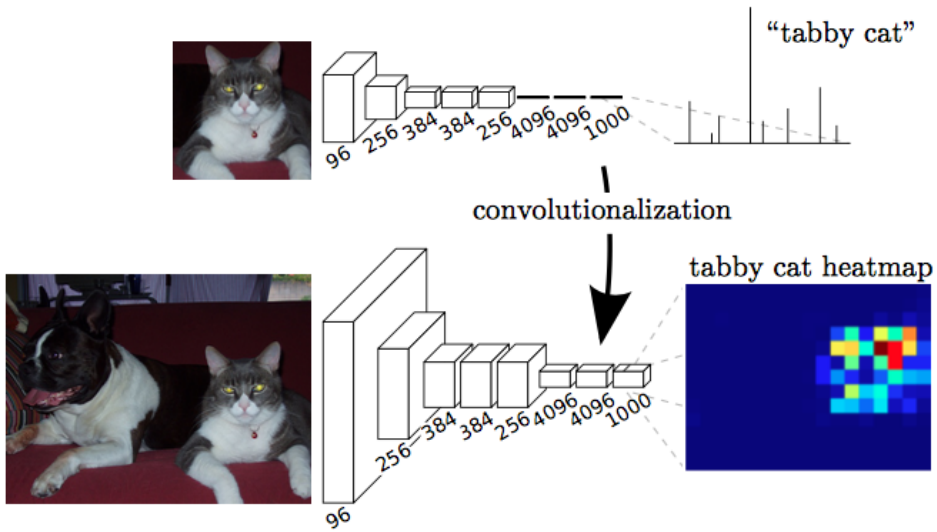


Figure 12: Transforming all the fully connected layers to convolutional layers.

universitätbonn

[11] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

Dense Prediction

- Arbitrary input size: Take input of arbitrary size and produce correspondingly-sized output efficiently.
- Transforming all the fully connected layers to convolutional layers enables a classification net to output a heatmap[11]).
- Transfer learning using pretrained DCNN(i.e. VggNet, Resnet).

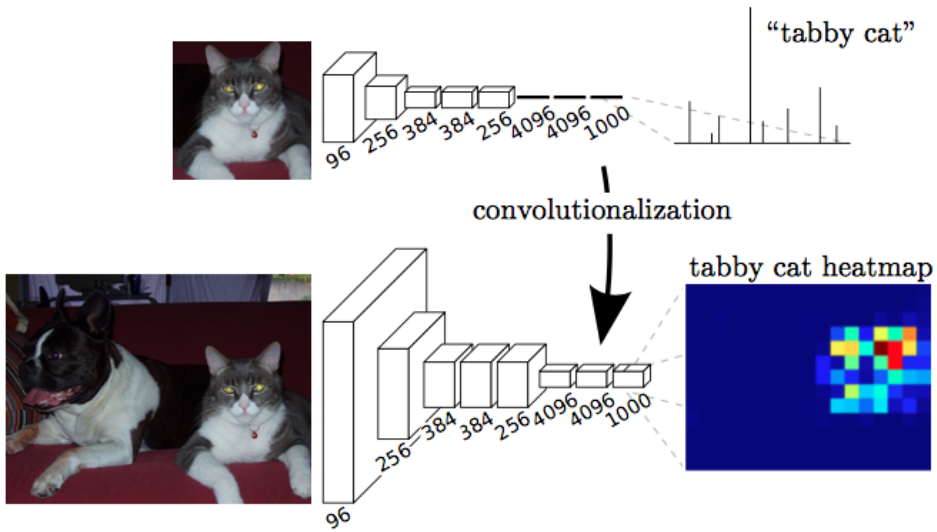


Figure 12: Transforming all the fully connected layers to convolutional layers.

[11] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

Dense Prediction

- Arbitrary input size: Take input of arbitrary size and produce correspondingly-sized output efficiently.
- Transforming all the fully connected layers to convolutional layers enables a classification net to output a heatmap[11]).
- Transfer learning using pretrained DCNN(i.e. VggNet, Resnet).

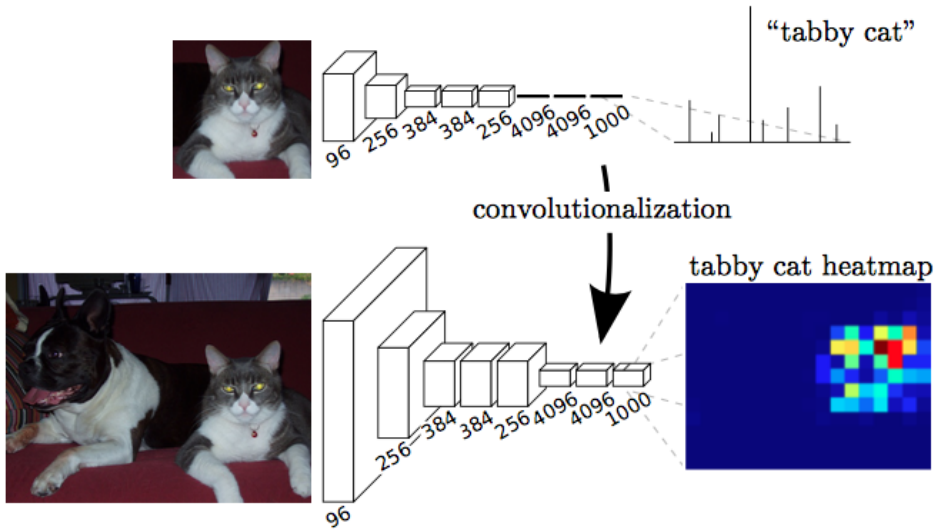


Figure 12: Transforming all the fully connected layers to convolutional layers.

universitätbonn

[11] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

Challenges

Challenges in the application of DCNNs on semantic image segmentation:

1. Reduced feature resolution: Caused by the repeated combination of max-pooling(downsampling) performed at consecutive layers of DCNNs originally designed for image classification.

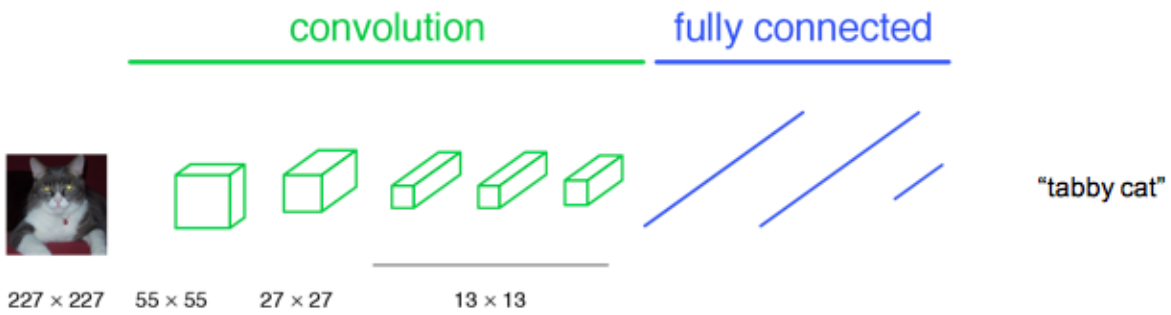


Figure 13: A classification network[12].

universitätbonn

[12] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

Challenges

1. Reduced feature resolution: Caused by the repeated combination of max-pooling(downsampling) performed at consecutive layers of DCNNs originally designed for image classification.
   - Solution #1: Convert coarse outputs to dense pixels using interpolation. i.e. simple bilinear interpolation.
     - \+ Fast.
     - – Not efficient as the operation is not learnable.
   - Solution #2: Convolution with fractional strides. i.e. deconvolution[12].
     - \+ Slow.
     - – Can be learned, thus efficient(fast).
   - Solution #3: DCNN without max-pooling layers.
     - \+ Can be learned.
     - – Discriminative ability owes much of their power to max-pooing layers.

universitätbonn

[12] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

Challenges

1. Reduced feature resolution: Caused by the repeated combination of max-pooling(downsampling) performed at consecutive layers of DCNNs originally designed for image classification.
   - Solution #1: Convert coarse outputs to dense pixels using interpolation. i.e. simple bilinear interpolation.
     - \+ Fast.
     - – Not efficient as the operation is not learnable.
   - Solution #2: Convolution with fractional strides. i.e. deconvolution[12].
     - \+ Slow.
     - – Can be learned, thus efficient(fast).
   - Solution #3: DCNN without max-pooling layers.
     - \+ Can be learned.
     - – Discriminative ability owes much of their power to max-pooing layers.

universitätbonn

[12] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

Challenges

1. Reduced feature resolution: Caused by the repeated combination of max-pooling(downsampling) performed at consecutive layers of DCNNs originally designed for image classification.
   - Solution #1: Convert coarse outputs to dense pixels using interpolation. i.e. simple bilinear interpolation.
     - \+ Fast.
     - – Not efficient as the operation is not learnable.
   - Solution #2: Convolution with fractional strides. i.e. deconvolution[12].
     - \+ Slow.
     - – Can be learned, thus efficient(fast).
   - Solution #3: DCNN without max-pooling layers.
     - \+ Can be learned.
     - – Discriminative ability owes much of their power to max-pooing layers.

universitätbonn

[12] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

Challenges

2. Existence of objects at multiple scales
   - ShareNet: Resizes the input image to several scales and passes each through a shared deep network[13].
   - Define scale-invariant mean squared error[14]:

$$D(y, y^*) = \frac{1}{2n} \sum_{=1}^{n} (\log y_i - \log y_i^* + a(y, y^*))^2$$

$$\text{where, } \alpha(y, y^*) = \frac{1}{n} \sum_{i} (\log y_i^* - \log y_i)$$

[13] Chen, Liang-Chieh, et al. "Attention to scale: Scale-aware semantic image segmentation."

[14] Eigen, David, et al. "Depth map prediction from a single image using a multi-scale deep network."

universitätbonn

Challenges

2. Existence of objects at multiple scales
   - ShareNet: Resizes the input image to several scales and passes each through a shared deep network[13].
   - Define scale-invariant mean squared error[14]:

$$D(y, y^*) = \frac{1}{2n} \sum_{=1}^{n} (\log y_i - \log y_i^* + a(y, y^*))^2$$

$$\text{where, } \alpha(y, y^*) = \frac{1}{n} \sum_{i} (\log y_i^* - \log y_i)$$

[13] Chen, Liang-Chieh, et al. "Attention to scale: Scale-aware semantic image segmentation."

[14] Eigen, David, et al. "Depth map prediction from a single image using a multi-scale deep network."

universitätbonn

Challenges

2. Existence of objects at multiple scales
   - Spatial pyramid pooling[15]: Generates a fixed-length representation regardless of image size/scale

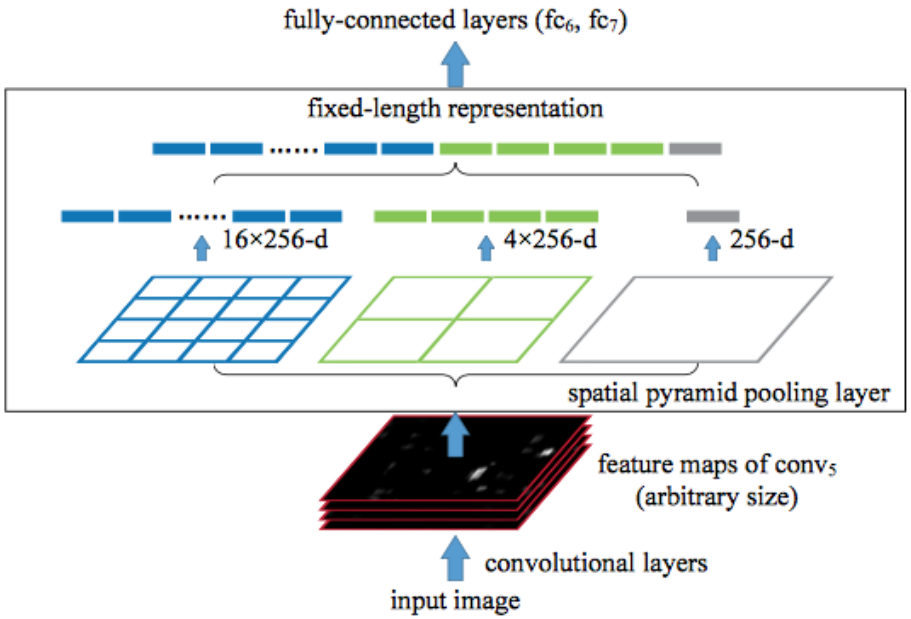


Figure 13: Network structure with a spatial pyramid pooling layer.

[15] He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition (2014)."

Challenges

3. DCNN has limited spatial capacity: Invariance to spatial transformations limits the spatial accuracy of a DCNN.
   - SkipNet: Features from intermediate layers are fused to produce the final output[16].

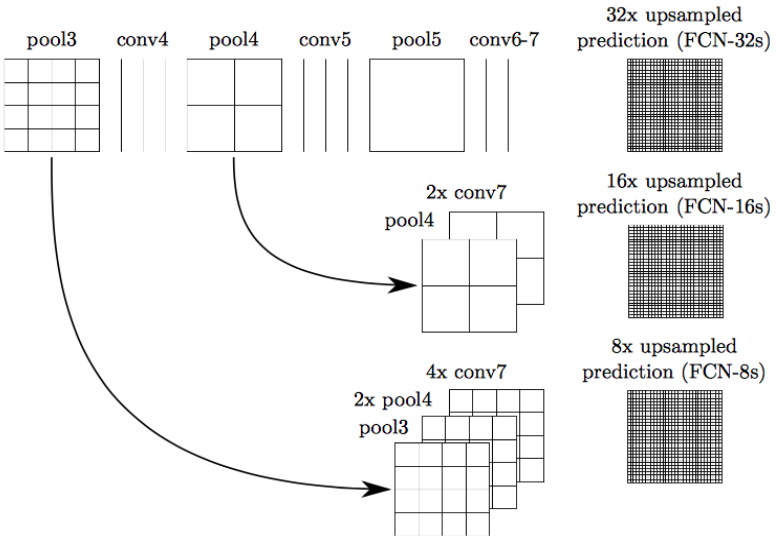


Figure 14: SkipNet Architecture.

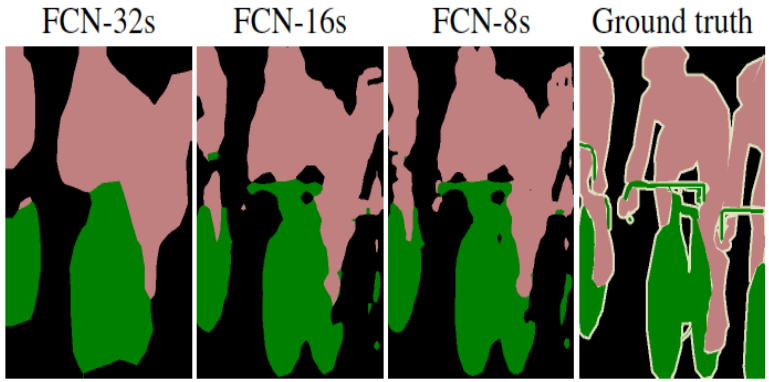


Figure 15: Refining fully convolutional nets by fusing information from layers with different strides[16].

universitätbonn

[16] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

Challenges

3. DCNN has limited spatial capacity: Invariance to spatial transformations limits the spatial accuracy of a DCNN.
   - Hypercolumns: Compute the outputs of all units above the locations at all layers of the CNN, stacked into one vector[17].

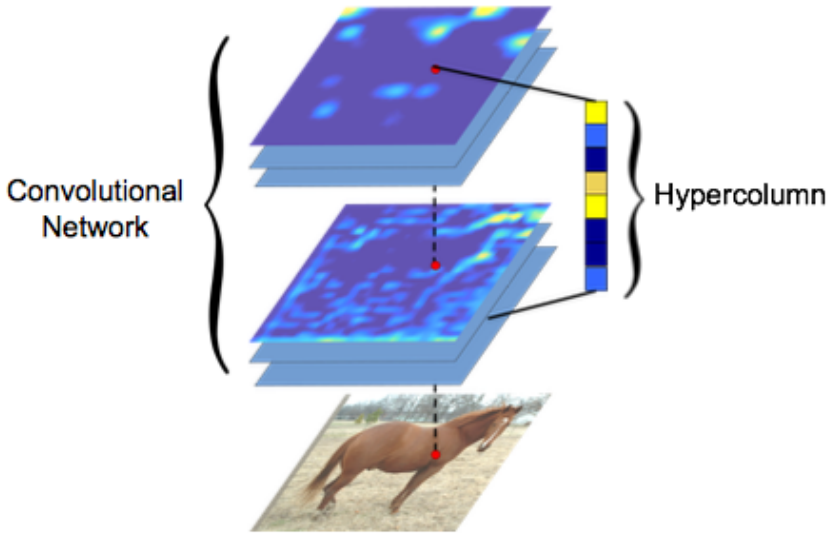


Figure 16: Hypercolumns

universitätbonn

[17] Hariharan, Bharath, et al. "Hypercolumns for object segmentation."

Key Technical Ideas

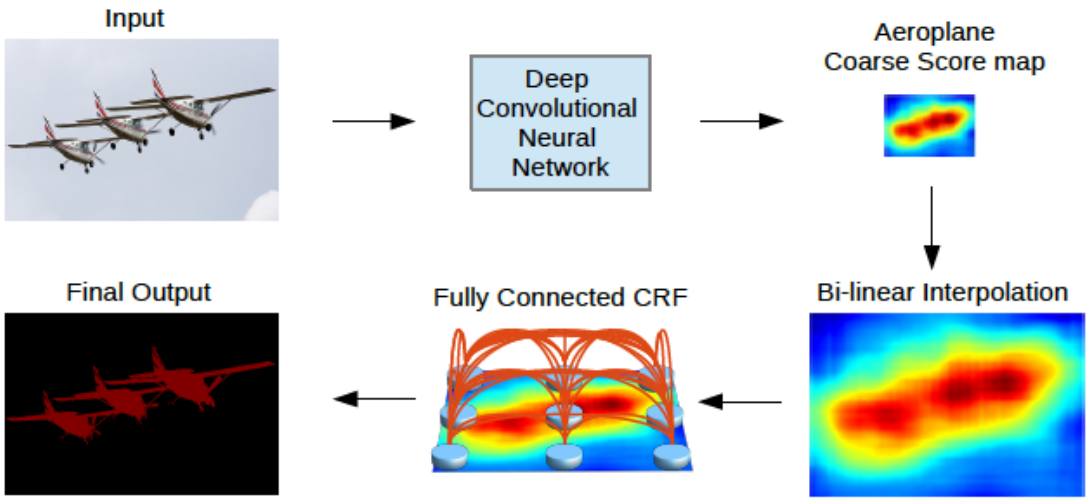


Figure 17: Model Illustration.

- VGG-16/Resnet-101 trained on ImageNet dataset is employed in a fully convolutional fashion,
- Atrous convolution to reduce the degree of signal downsampling (from 32× down to 8×).
- Bilinear interpolation stage to enlarge the feature maps.
- A fully connected CRF applied to refine the segmentation result.

universitätbonn

Key Technical Ideas

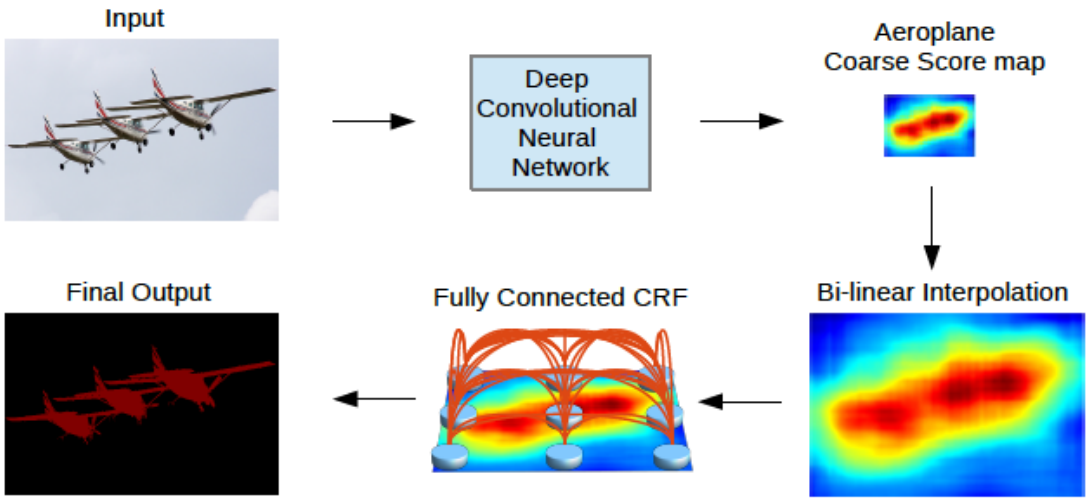


Figure 17: Model Illustration.

- VGG-16/Resnet-101 trained on ImageNet dataset is employed in a fully convolutional fashion,
- Atrous convolution to reduce the degree of signal downsampling (from 32× down to 8×).
- Bilinear interpolation stage to enlarge the feature maps.
- A fully connected CRF applied to refine the segmentation result.

universitätbonn

Key Technical Ideas

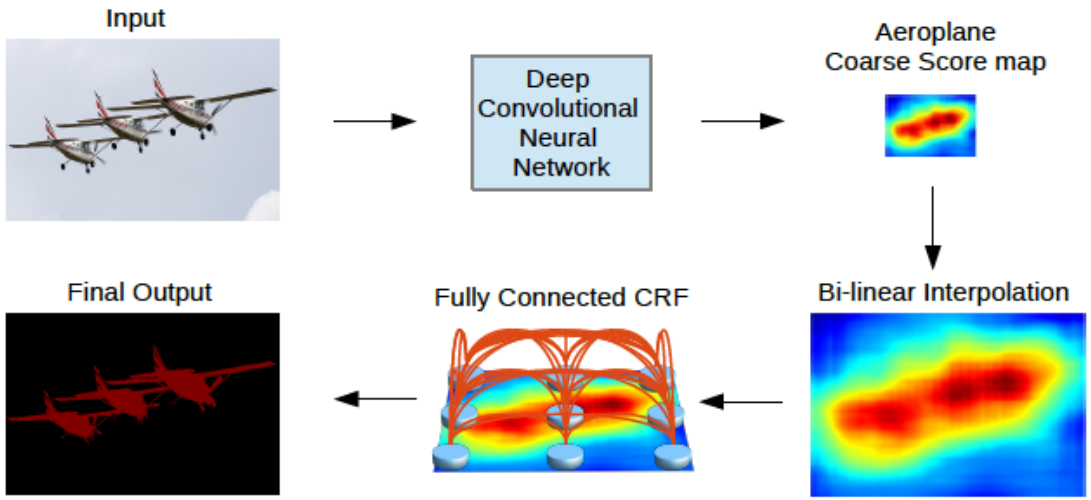


Figure 17: Model Illustration.

- VGG-16/Resnet-101 trained on ImageNet dataset is employed in a fully convolutional fashion,
- Atrous convolution to reduce the degree of signal downsampling (from 32× down to 8×).
- Bilinear interpolation stage to enlarge the feature maps.
- A fully connected CRF applied to refine the segmentation result.

universitätbonn

Key Technical Ideas

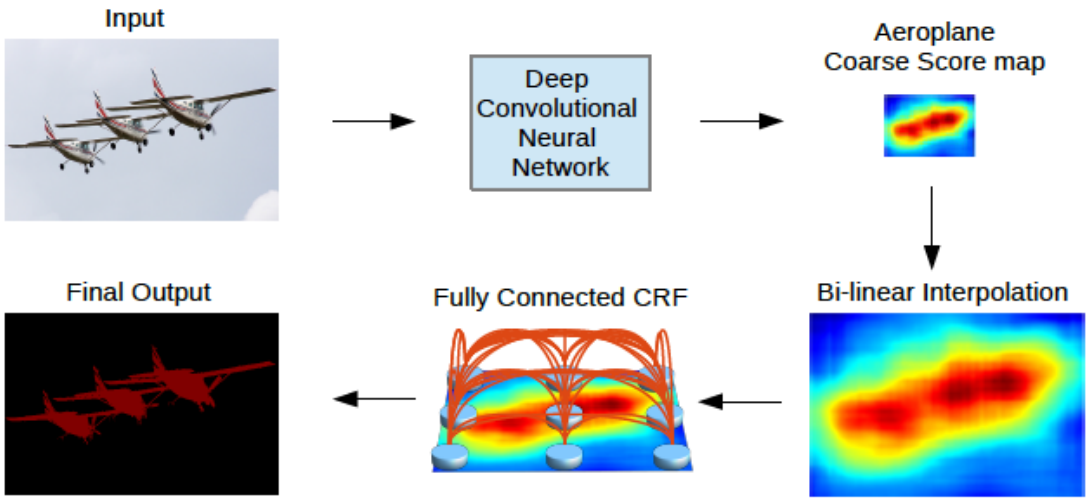


Figure 17: Model Illustration.

- VGG-16/Resnet-101 trained on ImageNet dataset is employed in a fully convolutional fashion,
- Atrous convolution to reduce the degree of signal downsampling (from 32× down to 8×).
- Bilinear interpolation stage to enlarge the feature maps.
- A fully connected CRF applied to refine the segmentation result.

universitätbonn

Atrous Convolution

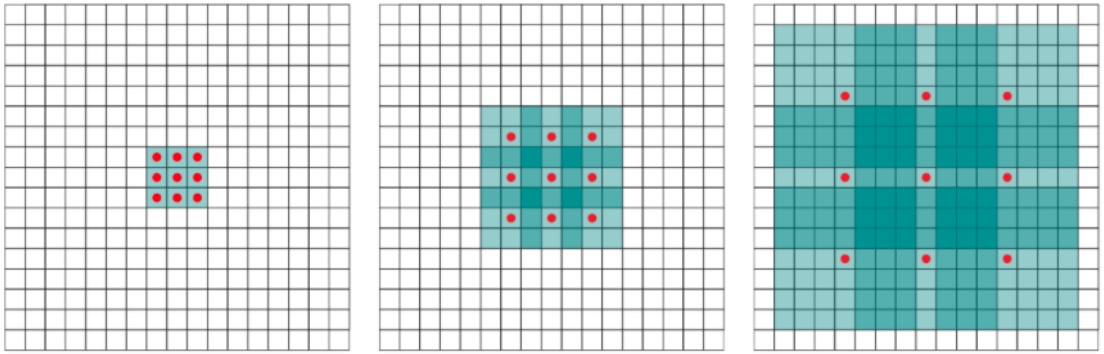


Figure 18: Atrous filtering[18].

- Allows to compute the responses of any layer at any desirable resolution.
- Inspired from the dilating filters used in stationary wavelet transform(*algorithme á trous*[19]).
- The output $y[i]$ of atrous convolution where r is the *rate* parameter:

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k]$$

[18] Yu, F. (2015) Multi-scale context aggregation by dilated convolutions.

[19] Holschneider, Matthias, et al. "A real-time algorithm for signal analysis with the help of the wavelet transform. (1989)"

universitätbonn

Atrous Convolution

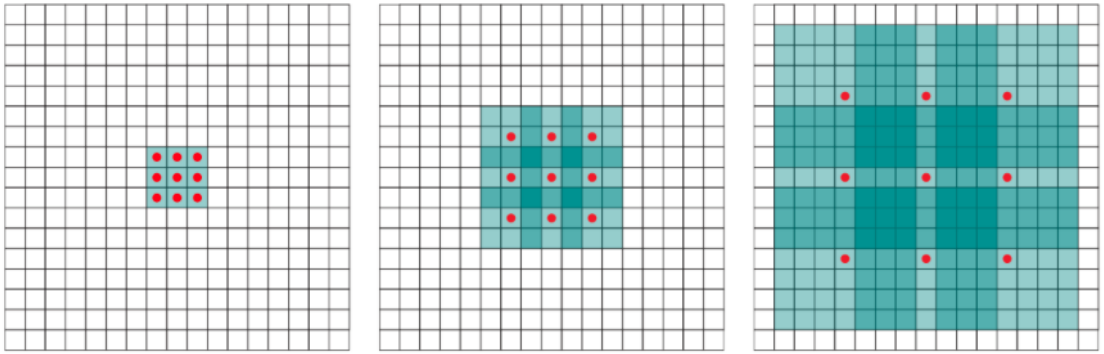


Figure 18: Atrous filtering[18].

- Allows to compute the responses of any layer at any desirable resolution.
- Inspired from the dilating filters used in stationary wavelet transform(*algorithme á trous*[19]).
- The output $y[i]$ of atrous convolution where r is the *rate* parameter:

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k]$$

[18] Yu, F. (2015) Multi-scale context aggregation by dilated convolutions.

[19] Holschneider, Matthias, et al. "A real-time algorithm for signal analysis with the help of the wavelet transform. (1989)"

Atrous Convolution

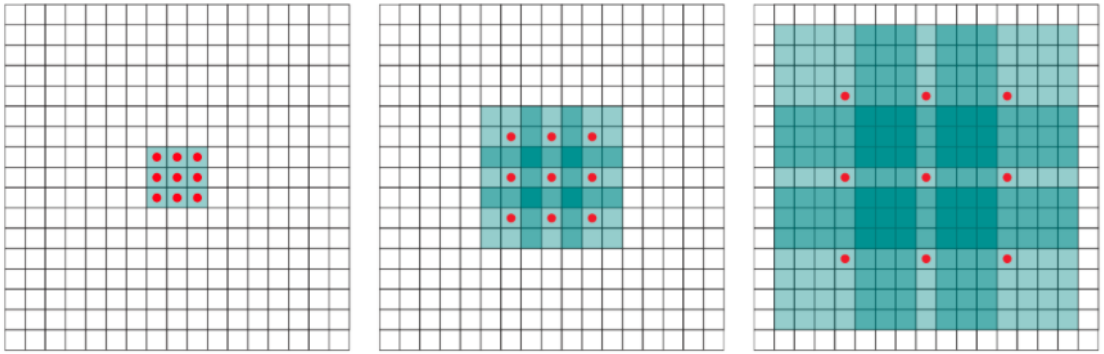


Figure 18: Atrous filtering[18].

- Allows to compute the responses of any layer at any desirable resolution.
- Inspired from the dilating filters used in stationary wavelet transform(*algorithme á trous*[19]).
- The output $y[i]$ of atrous convolution where r is the *rate* parameter:

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k]$$

[18] Yu, F. (2015) Multi-scale context aggregation by dilated convolutions.

[19] Holschneider, Matthias, et al. "A real-time algorithm for signal analysis with the help of the wavelet transform. (1989)"

universitätbonn

Atrous Convolution

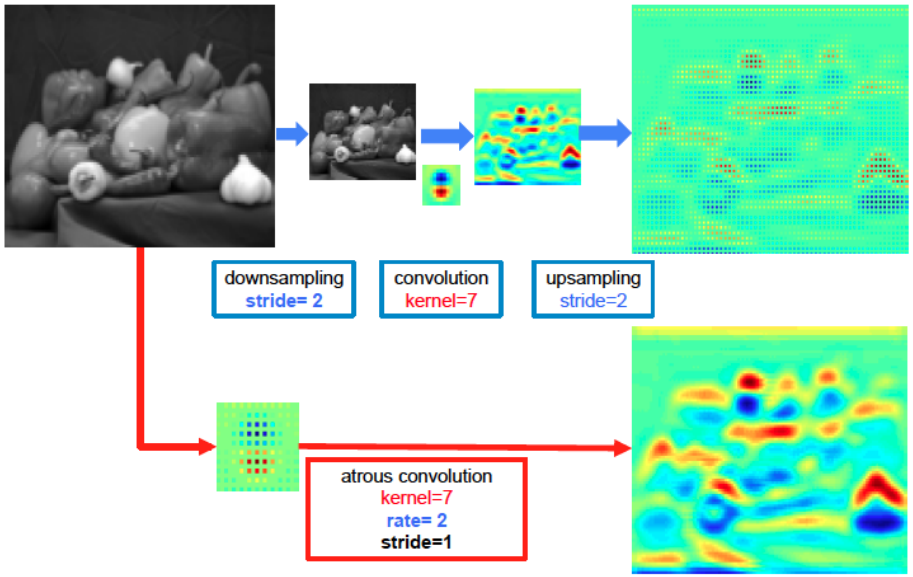


Figure 19: Atrous filtering in 2D

- Trick: Allows to preserve/enlarge image resolution and enlarge FOV at once.
- Subsampling leads to sparse feature extraction. Therefore, upsampled features are sparse as well.
- Number of filter parameters remains the same.
- Learnable and memory/time efficient.

universitätbonn

Atrous Convolution

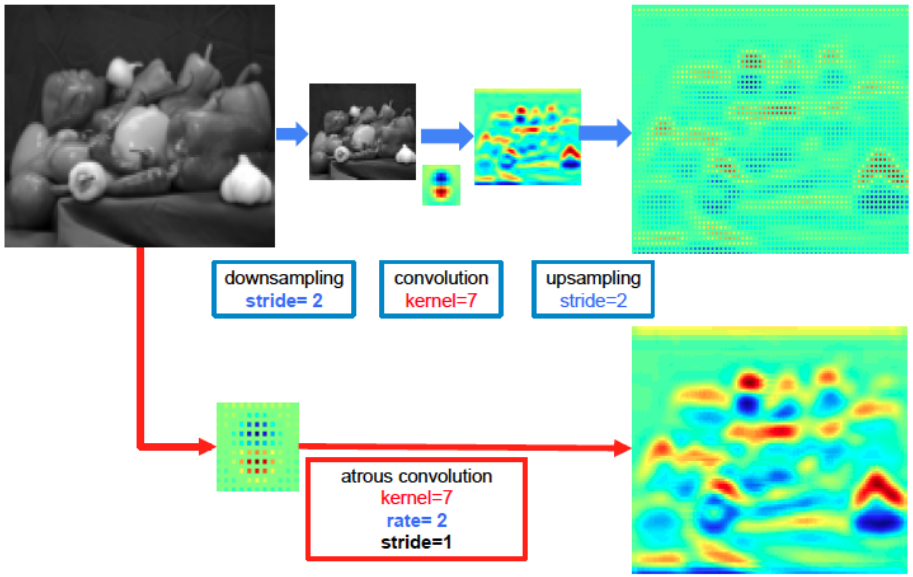


Figure 19: Atrous filtering in 2D

- Trick: Allows to preserve/enlarge image resolution and enlarge FOV at once.
- Subsampling leads to sparse feature extraction. Therefore, upsampled features are sparse as well.
- Number of filter parameters remains the same.
- Learnable and memory/time efficient.

universitätbonn

Atrous Convolution

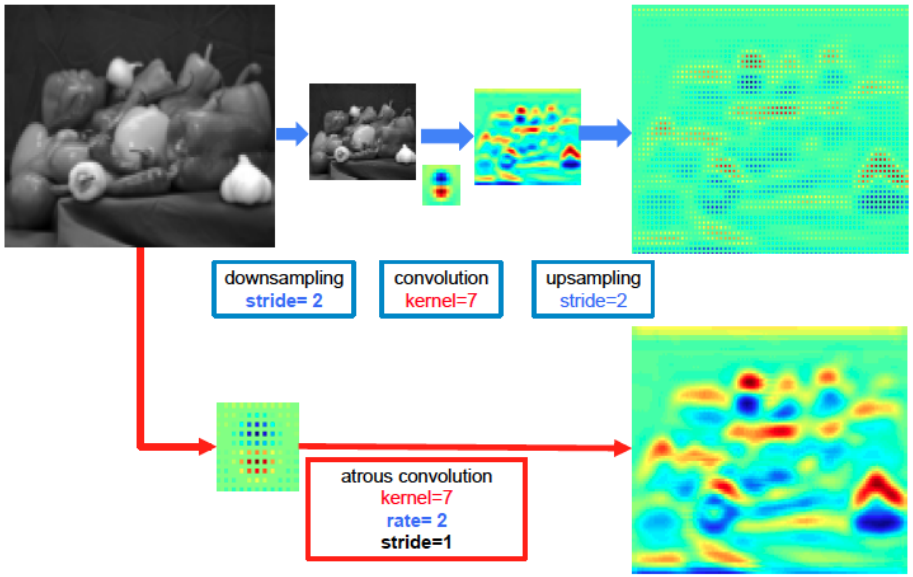


Figure 19: Atrous filtering in 2D

- Trick: Allows to preserve/enlarge image resolution and enlarge FOV at once.
- Subsampling leads to sparse feature extraction. Therefore, upsampled features are sparse as well.
- Number of filter parameters remains the same.
- Learnable and memory/time efficient.

universitätbonn

Atrous Convolution

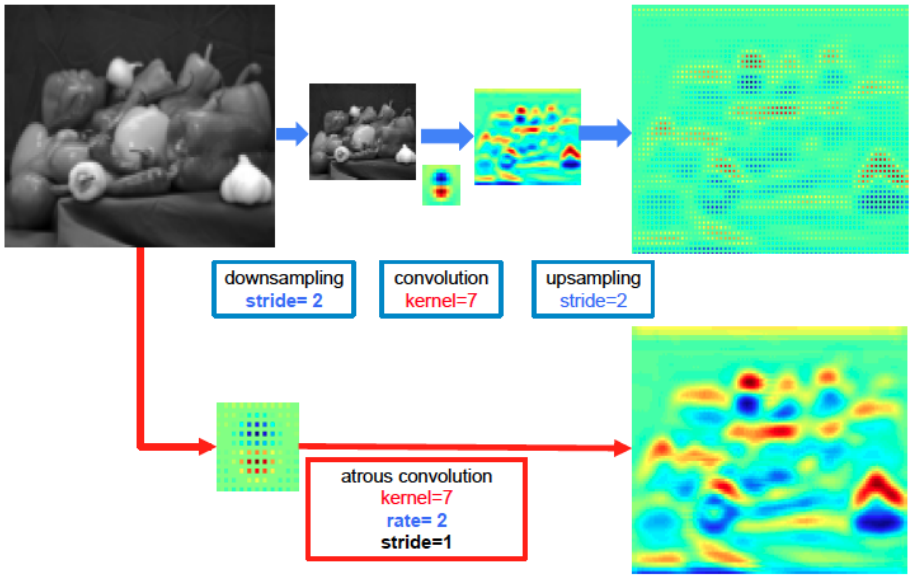


Figure 19: Atrous filtering in 2D

- Trick: Allows to preserve/enlarge image resolution and enlarge FOV at once.
- Subsampling leads to sparse feature extraction. Therefore, upsampled features are sparse as well.
- Number of filter parameters remains the same.
- Learnable and memory/time efficient.

universitätbonn

Atrous Convolution

Table 1: VGG_{16}: 224x224x3 input image; 1x1000 output labels.

	1	2	3	4	5	6	7	8	9	10	11	12
layer	**2xconv**	**max**	**2xconv**	**max**	**3xconv**	**max**	**3xconv**	**max**	**3xconv**	**max**	**2xfc**	**fc**
fs	3-1	2-2	3-1	2-2	3-1	2-2	3-1	2-2	3-1	2-2	-	-
#cha	64	64	128	128	256	256	512	512	512	512	1	1
act	relu	idn	relu	idn	relu	idn	relu	idn	relu	idn	relu	soft
size	224	112	112	56	56	28	28	14	14	7	4096	1000

- The last two max-pooling layers are of 2×2 filters with stride 1.
- Reduce the degree of signal downsampling (from $32\times$ down $8\times$).
- Modify VGG-16 net:
  - $r = 2$ atrous filters in layers: conv5_1, conv5_2 and conv5_3.
  - $r = 4$ atrous filter in the first fully-connected layer(fc6).
  - Bilinear filter($8\times$) in the last fully-connected layer(fc8).

universitätbonn

Atrous Convolution

Kernel	Rate	FOV	**Params**
7x7	4	224	134.3M
4x4	4	128	65.1M
4x4	8	224	65.1M
3x3	12	224	20.5M

Table 2: Effect of kernel size on FOV.

- Direct adaptation of VGG-16 net using 7×7 kernel size and $r = 4$.
- Improved model speed by reducing the kernel size to 4×4.
  - Fails to capture local detail when *rate* is increased.

universitätbonn

Atrous Convolution

Kernel	Rate	FOV	**Params**
7x7	4	224	134.3M
4x4	4	128	65.1M
4x4	8	224	65.1M
3x3	12	224	20.5M

Table 2: Effect of kernel size on FOV.

- Direct adaptation of VGG-16 net using 7×7 kernel size and $r = 4$.
- Improved model speed by reducing the kernel size to 4×4.

– Fails to capture local detail when *rate* is increased.

universitätbonn

Atrous Spatial Pyramid Pooling (ASPP)

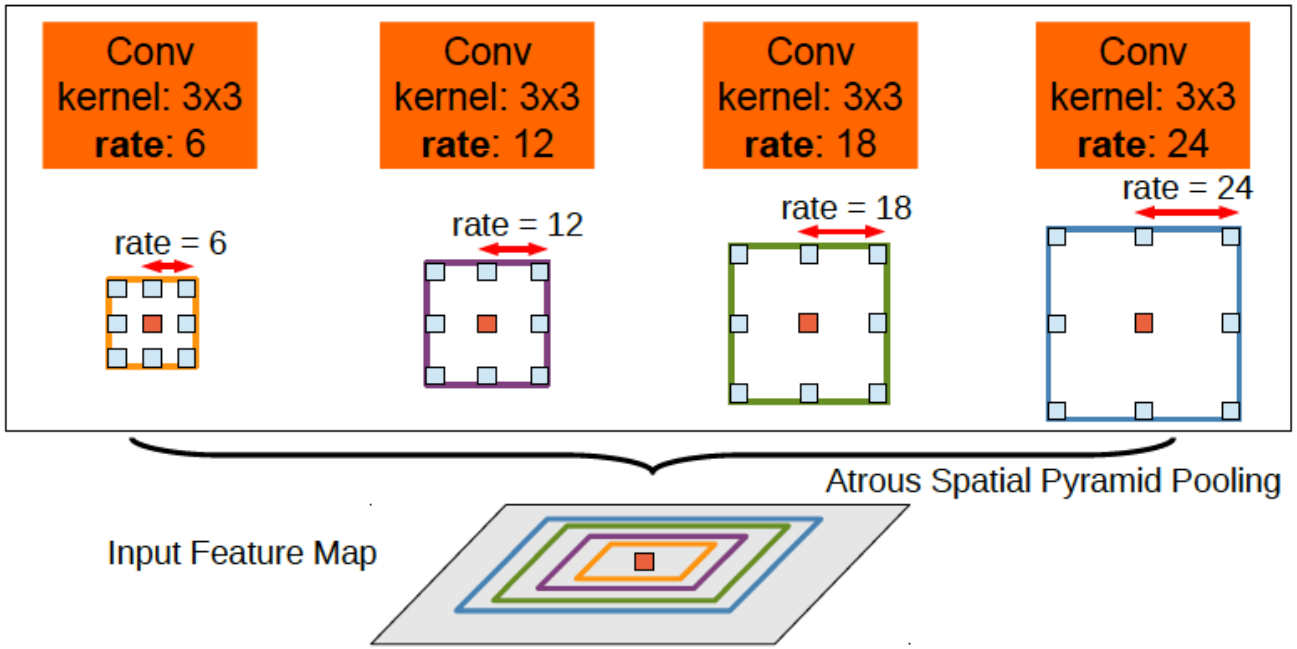


Figure 20: Atrous Spatial Pyramid Pooling (ASPP).

- The network should be able to handle objects at mutiple scale.
- ASPP extracts multi-scale features by employing multiple parallel filters with different rates on the same feature map.

universitätbonn

Atrous Spatial Pyramid Pooling (ASPP)

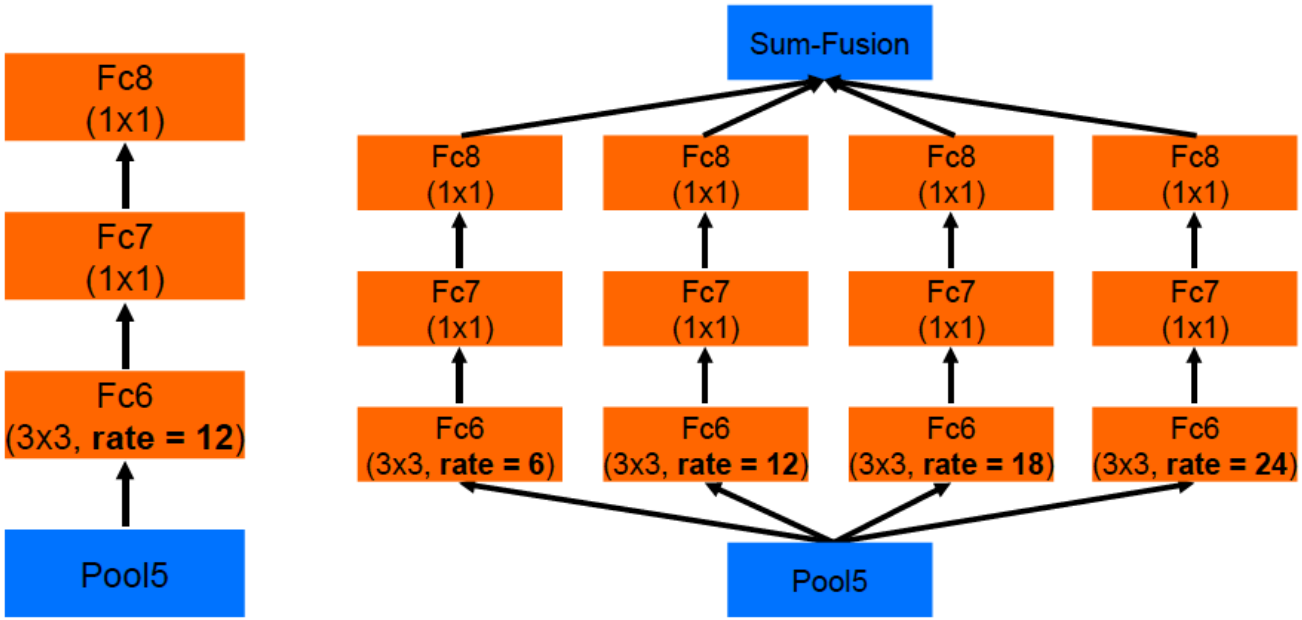


Figure 21: Atrous Spatial Pyramid Pooling (ASPP).

- Inspired from Spatial Pyramid Pooling[20].
- ASPP for VGG-16 employs several parallel fc6-fc7-fc8 branches.
- Four branches and smaller atrous rates $r = \{6, 12, 18, 24\}$.

universitätbonn

[20] He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition (2014)."

Atrous Spatial Pyramid Pooling (ASPP)

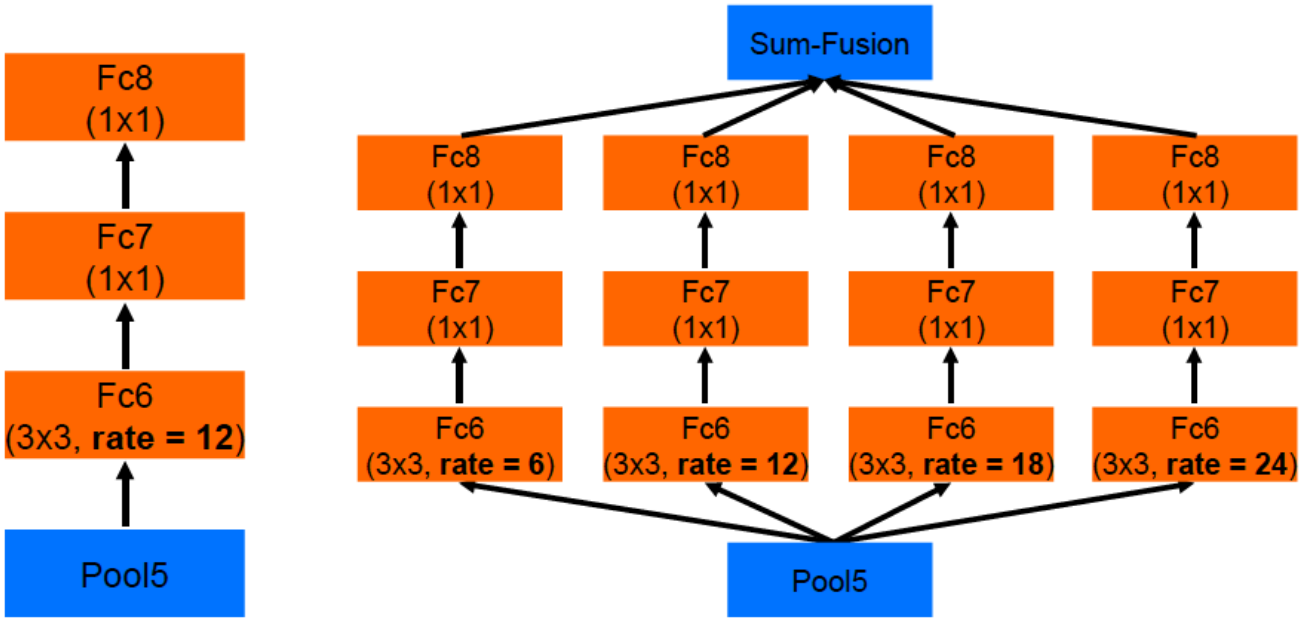


Figure 21: Atrous Spatial Pyramid Pooling (ASPP).

- Inspired from Spatial Pyramid Pooling[20].
- ASPP for VGG-16 employs several parallel fc6-fc7-fc8 branches.
- Four branches and smaller atrous rates $r = \{6, 12, 18, 24\}$.

universitätbonn

[20] He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition (2014)."

Fully Connected CRF

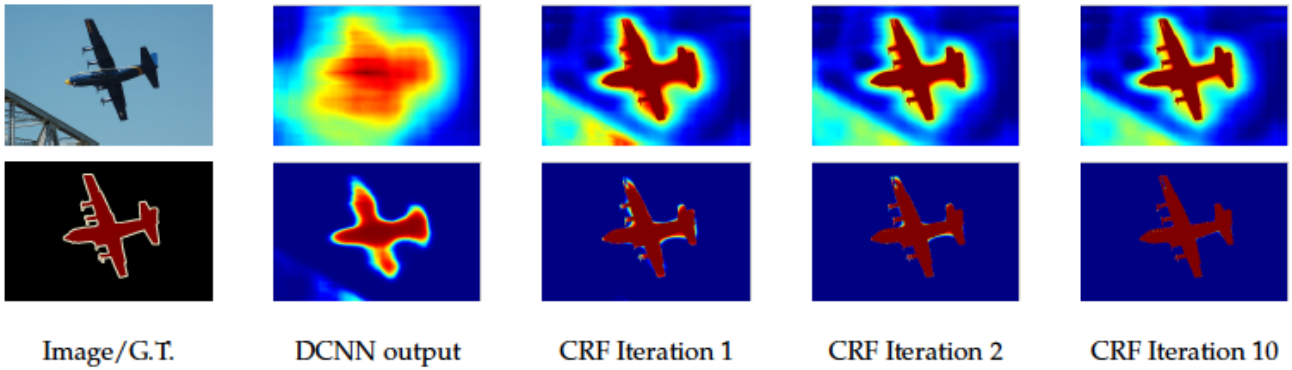


Figure 22: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane.

- Large receptive fields can only yield smooth responses.
- CRF helps us to recover object boundaries at a level of detail.
- Short-range CRF is detrimental: results in excessive smoothing of object boundaries[21].
- To overcome these problems, fully connected CRF is utilized.

[21] Krähenbühl et al. "Efficient inference in fully connected crfs with Gaussian edge potentials."

universitätbonn

Fully Connected CRF

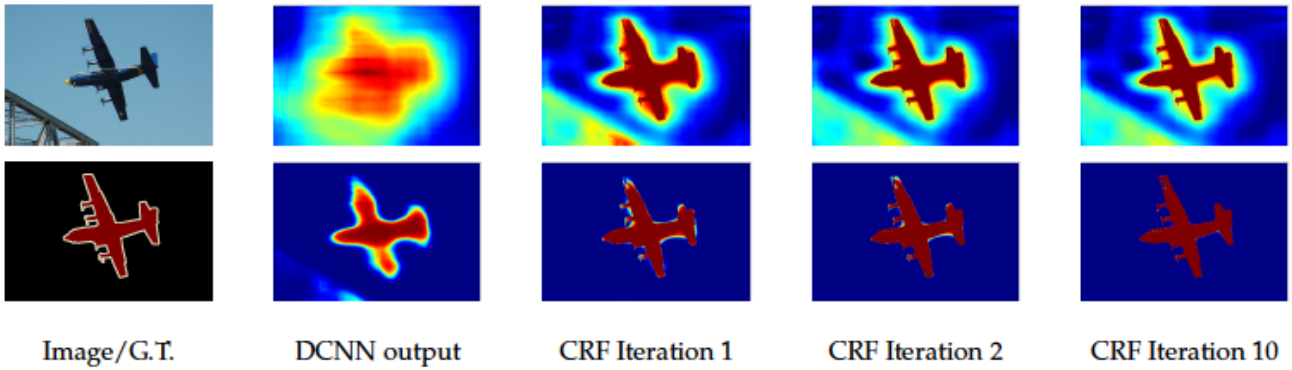


Figure 22: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane.

- Large receptive fields can only yield smooth responses.
- CRF helps us to recover object boundaries at a level of detail.
- Short-range CRF is detrimental: results in excessive smoothing of object boundaries[21].
- To overcome these problems, fully connected CRF is utilized.

universitätbonn

[21] Krähenbühl et al. "Efficient inference in fully connected crfs with Gaussian edge potentials."

Fully Connected CRF

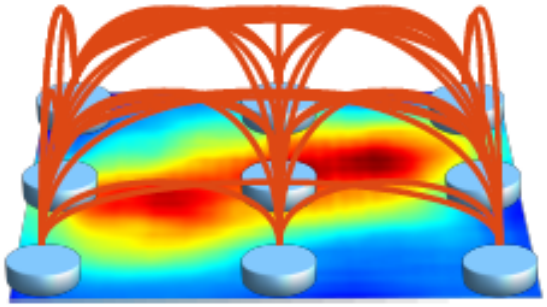


Figure 23: Fully Connected CRF.

- Gibbs energy[22] corresponding to FC-CRF is

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j)$$

where x is the label assignment and ij are pixel pairs.

- Unary potentials are

$$\theta_i(x_i) = -logP(x_i)$$

where $P(x_i)$ is the label assignment probability at pixel i computed by DCNN.

[22] Krähenbühl et al. "Efficient inference in fully connected crfs with Gaussian edge potentials."

universitätbonn

Fully Connected CRF

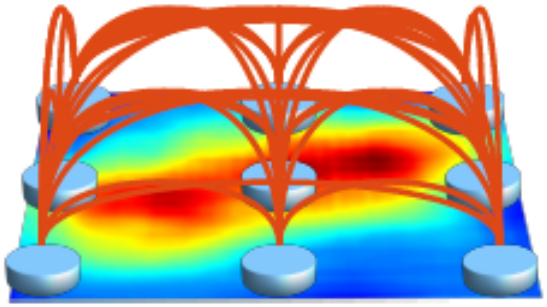


Figure 23: Fully Connected CRF.

- Gibbs energy[22] corresponding to FC-CRF is

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j)$$

where x is the label assignment and ij are pixel pairs.

- Unary potentials are

$$\theta_i(x_i) = -logP(x_i)$$

where $P(x_i)$ is the label assignment probability at pixel i computed by DCNN.

universitätbonn

[22] Krähenbühl et al. "Efficient inference in fully connected crfs with Gaussian edge potentials."

Fully Connected CRF

- MAP labeling by the unary classifiers alone is noisy and inconsistent.
- Pairwise potentials are utilized:

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 exp\left(-\frac{\|\mathbf{p_i} - \mathbf{p_j}\|^2}{2\sigma_\alpha^2} - \frac{\|\mathbf{I_i} - \mathbf{I_j}\|^2}{2\sigma_\beta^2} \right) + w_2 exp\left(-\frac{\|\mathbf{p_i} - \mathbf{p_j}\|^2}{2\sigma_\gamma^2} \right) \right]$$

  where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$ and zero otherwise.
- Gaussian potentials are calculated in different feature spaces.

universitätbonn

Fully Connected CRF

- MAP labeling by the unary classifiers alone is noisy and inconsistent.
- Pairwise potentials are utilized:

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 exp\left(-\frac{\|\mathbf{p_i} - \mathbf{p_j}\|^2}{2\sigma_\alpha^2} - \frac{\|\mathbf{I_i} - \mathbf{I_j}\|^2}{2\sigma_\beta^2} \right) + w_2 exp\left(-\frac{\|\mathbf{p_i} - \mathbf{p_j}\|^2}{2\sigma_\gamma^2} \right) \right]$$

  where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$ and zero otherwise.

- Gaussian potentials are calculated in different feature spaces.

universitätbonn

Fully Connected CRF

- MAP labeling by the unary classifiers alone is noisy and inconsistent.
- Pairwise potentials are utilized:

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 exp\left(-\frac{\|\mathbf{p_i} - \mathbf{p_j}\|^2}{2\sigma_\alpha^2} - \frac{\|\mathbf{I_i} - \mathbf{I_j}\|^2}{2\sigma_\beta^2} \right) + w_2 exp\left(-\frac{\|\mathbf{p_i} - \mathbf{p_j}\|^2}{2\sigma_\gamma^2} \right) \right]$$

  where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$ and zero otherwise.
- Gaussian potentials are calculated in different feature spaces.

universitätbonn

Fully Connected CRF

- Gaussian CRF potentials in the fully connected CRF model can capture long-range dependencies.
- Model is amenable to efficient fast mean field inference[23].
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions[24].
- Key idea: if kernel/pairwise cost function is Gaussian, message passing can be performed via Gaussian filtering in feature space.
- Reduces the complexity of message passing from quadratic to linear. Inference requires < 0.5 secs on a CPU. Graph cut inference in the fully connected models takes > 72 hours.

[23] Krähenbühl et al. "Efficient inference in fully connected crfs with gaussian edge potentials."

[24] Nowozin, Sebastian, and Christoph H. Lampert. "Structured learning and prediction in computer vision."

universitätbonn

Fully Connected CRF

- Gaussian CRF potentials in the fully connected CRF model can capture long-range dependencies.
- Model is amenable to efficient fast mean field inference[23].
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions[24].
- Key idea: if kernel/pairwise cost function is Gaussian, message passing can be performed via Gaussian filtering in feature space.
- Reduces the complexity of message passing from quadratic to linear. Inference requires < 0.5 secs on a CPU. Graph cut inference in the fully connected models takes > 72 hours.

[23] Krähenbühl et al. "Efficient inference in fully connected crfs with gaussian edge potentials."

[24] Nowozin, Sebastian, and Christoph H. Lampert. "Structured learning and prediction in computer vision."

universitätbonn

Fully Connected CRF

- Gaussian CRF potentials in the fully connected CRF model can capture long-range dependencies.
- Model is amenable to efficient fast mean field inference[23].
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions[24].
- Key idea: if kernel/pairwise cost function is Gaussian, message passing can be performed via Gaussian filtering in feature space.
- Reduces the complexity of message passing from quadratic to linear. Inference requires < 0.5 secs on a CPU. Graph cut inference in the fully connected models takes > 72 hours.

[23] Krähenbühl et al. "Efficient inference in fully connected crfs with gaussian edge potentials."

[24] Nowozin, Sebastian, and Christoph H. Lampert. "Structured learning and prediction in computer vision."

universitätbonn

Fully Connected CRF

- Gaussian CRF potentials in the fully connected CRF model can capture long-range dependencies.
- Model is amenable to efficient fast mean field inference[23].
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions[24].
- Key idea: if kernel/pairwise cost function is Gaussian, message passing can be performed via Gaussian filtering in feature space.
- Reduces the complexity of message passing from quadratic to linear. Inference requires < 0.5 secs on a CPU. Graph cut inference in the fully connected models takes > 72 hours.

[23] Krähenbühl et al. "Efficient inference in fully connected crfs with gaussian edge potentials."

[24] Nowozin, Sebastian, and Christoph H. Lampert. "Structured learning and prediction in computer vision."

universitätbonn

Fully Connected CRF

- Gaussian CRF potentials in the fully connected CRF model can capture long-range dependencies.
- Model is amenable to efficient fast mean field inference[23].
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions[24].
- Key idea: if kernel/pairwise cost function is Gaussian, message passing can be performed via Gaussian filtering in feature space.
- Reduces the complexity of message passing from quadratic to linear. Inference requires < 0.5 secs on a CPU. Graph cut inference in the fully connected models takes > 72 hours.

[23] Krähenbühl et al. "Efficient inference in fully connected crfs with gaussian edge potentials."

[24] Nowozin, Sebastian, and Christoph H. Lampert. "Structured learning and prediction in computer vision."

universitätbonn

DeepLab: Experimental Set-Up

- VGG-16 and ResNet-101 trained on ImageNet[25] dataset are employed.
- Evaluation is perfomed on PASCAL VOC 2012, PASCAL-Context, PASCAL-Person-Part, and Cityscapes datasets.
- 1000-way Imagenet classifier in the last layer is replaced with a classifier having the number of semantic classes.
- DCNN and CRF training stages are decoupled.

[25] Krizhevsky, Alex et al. "Imagenet classification with deep convolutional neural networks."

universitätbonn

DeepLab: Experimental Set-Up

- VGG-16 and ResNet-101 trained on ImageNet[25] dataset are employed.
- Evaluation is perfomed on PASCAL VOC 2012, PASCAL-Context, PASCAL-Person-Part, and Cityscapes datasets.
- 1000-way Imagenet classifier in the last layer is replaced with a classifier having the number of semantic classes.
- DCNN and CRF training stages are decoupled.

universitätbonn

[25] Krizhevsky, Alex et al. "Imagenet classification with deep convolutional neural networks."

DeepLab: Experimental Set-Up

- VGG-16 and ResNet-101 trained on ImageNet[25] dataset are employed.
- Evaluation is perfomed on PASCAL VOC 2012, PASCAL-Context, PASCAL-Person-Part, and Cityscapes datasets.
- 1000-way Imagenet classifier in the last layer is replaced with a classifier having the number of semantic classes.
- DCNN and CRF training stages are decoupled.

[25] Krizhevsky, Alex et al. "Imagenet classification with deep convolutional neural networks."

universitätbonn

DeepLab: Experimental Set-Up

- VGG-16 and ResNet-101 trained on ImageNet[25] dataset are employed.
- Evaluation is perfomed on PASCAL VOC 2012, PASCAL-Context, PASCAL-Person-Part, and Cityscapes datasets.
- 1000-way Imagenet classifier in the last layer is replaced with a classifier having the number of semantic classes.
- DCNN and CRF training stages are decoupled.

universitätbonn

[25] Krizhevsky, Alex et al. "Imagenet classification with deep convolutional neural networks."

DeepLab: Experimental Set-Up

- Loss function is the sum of cross-entropy terms for each pixel in the CNN output map. Cross-entropy error function is,

$$\mathcal{H}(\mathbf{o}, \mathbf{t}) = -\sum_{i=0}^{n} \log(o_i) \cdot t_i$$

  where, o_i is the predicted probability distribution and t_i is the true probability of class i.

- CNN output map and GT labels are subsampled by 8 compared to the original image.
- Labels are mildly unbalanced[26] (about 3/4 are background). Hence, all labels are equally weighted when loss function is computed.
- Stochastic Gradient Descent method is utilized[27].

[26] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

[27] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." (2010)

universitätbonn

DeepLab: Experimental Set-Up

- Loss function is the sum of cross-entropy terms for each pixel in the CNN output map. Cross-entropy error function is,

$$\mathcal{H}(\mathbf{o}, \mathbf{t}) = -\sum_{i=0}^{n} \log(o_i) \cdot t_i$$

  where, o_i is the predicted probability distribution and t_i is the true probability of class i.
- CNN output map and GT labels are subsampled by 8 compared to the original image.
- Labels are mildly unbalanced[26] (about 3/4 are background). Hence, all labels are equally weighted when loss function is computed.
- Stochastic Gradient Descent method is utilized[27].

[26] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

[27] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." (2010)

universitätbonn

DeepLab: Experimental Set-Up

- Loss function is the sum of cross-entropy terms for each pixel in the CNN output map. Cross-entropy error function is,

$$\mathcal{H}(\mathbf{o}, \mathbf{t}) = -\sum_{i=0}^{n} \log(o_i) \cdot t_i$$

  where, o_i is the predicted probability distribution and t_i is the true probability of class i.
- CNN output map and GT labels are subsampled by 8 compared to the original image.
- Labels are mildly unbalanced[26] (about 3/4 are background). Hence, all labels are equally weighted when loss function is computed.
- Stochastic Gradient Descent method is utilized[27].

[26] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

[27] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." (2010)

universitätbonn

DeepLab: Experimental Set-Up

- Loss function is the sum of cross-entropy terms for each pixel in the CNN output map. Cross-entropy error function is,

$$\mathcal{H}(\mathbf{o}, \mathbf{t}) = -\sum_{i=0}^{n} \log(o_i) \cdot t_i$$

  where, o_i is the predicted probability distribution and t_i is the true probability of class i.
- CNN output map and GT labels are subsampled by 8 compared to the original image.
- Labels are mildly unbalanced[26] (about 3/4 are background). Hence, all labels are equally weighted when loss function is computed.
- Stochastic Gradient Descent method is utilized[27].

[26] Long, Jonathan, et al. "Fully convolutional networks for semantic segmentation."

[27] Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." (2010)

universitätbonn

DeepLab: Experimental Set-Up

Main contributions implemented and tested:

- Atrous Filter: Single branch, (kernel size 4×4, $r = 4$).
- LargeFOV: Single branch, (kernel size 3×3, $r = 12$) atrous filter.
- ASPP-S: Four branches, $r = \{2, 4, 8, 12\}$.
- ASPP/ASPP-L: Four branches, $r = \{6, 12, 18, 24\}$.
- CRF: Fully Connected CRF.

Other methods:

- MSC: multi-scale inputs are fed to the DCNN images at scale $= \{0.5, 0.75, 1\}$ with max-fusion.
- COCO: models pretrained on MS-COCO dataset.
- Aug: randomly scaled input images from 0.5 to 1.5.
- ResNet-101[28]: deeper network trained on ImageNet dataset.

universitätbonn

[28] He, Kaiming, et al. "Deep residual learning for image recognition. (2016)"

DeepLab: Experimental Set-Up

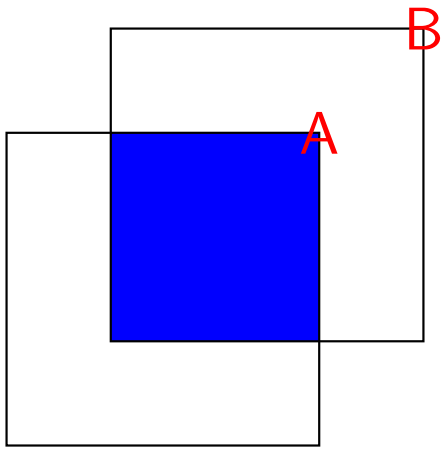


Figure 24: Intersection-over-union(IOU)

- Evaluation metric is pixel intersection-over-union (IOU) averaged across the classes.

$$\text{score} = 100 \times \left[\frac{\text{area of overlap}(A \cap B)}{\text{area of union}(A \cup B)} \right]$$

universitätbonn

Pascal VOC 2012: Dataset

Figure 25: Pascal VOC 2012 Dataset.

- 20 foreground object classes and one background class[29].
- 1,464 (*train*), 10,582 (*trainaug*), 1,449 (*val*), and 1,456 (*test*) pixel-level labeled images.
- Example of segmentation ground truth is shown in Figure 25.

[29] Everingham, M., et al. "The pascal visual object classes challenge 2012 (voc2012) results (2012)."

Pascal VOC 2012: Evaluation

- VGG-16 and ResNet-101 networks pre-trained on Imagenet are used.
- Momentum and weight decay are set to 0.9 and 0.0005 respectively.
- 10 mean field iterations.
- The number of filters in fc6/fc7 layers reduced from 4096 to 1024.
  - \+ Enables 3.36 times faster training speed.
- DeepLab-CRF-LargeFOV (VGG-16) model achieves 70.3% mean IOU performance on the official test set.
- DeepLab-CRF (ResNet-101) achieves 79.7% mean IOU, advancing state-of-the-art.

universitätbonn

Pascal VOC 2012: Evaluation

- VGG-16 and ResNet-101 networks pre-trained on Imagenet are used.
- Momentum and weight decay are set to 0.9 and 0.0005 respectively.
- 10 mean field iterations.
- The number of filters in fc6/fc7 layers reduced from 4096 to 1024.
  - \+ Enables 3.36 times faster training speed.
- DeepLab-CRF-LargeFOV (VGG-16) model achieves 70.3% mean IOU performance on the official test set.
- DeepLab-CRF (ResNet-101) achieves 79.7% mean IOU, advancing state-of-the-art.

universitätbonn

Pascal VOC 2012: Evaluation

- VGG-16 and ResNet-101 networks pre-trained on Imagenet are used.
- Momentum and weight decay are set to 0.9 and 0.0005 respectively.
- 10 mean field iterations.
- The number of filters in fc6/fc7 layers reduced from 4096 to 1024.
  - \+ Enables 3.36 times faster training speed.
- DeepLab-CRF-LargeFOV (VGG-16) model achieves 70.3% mean IOU performance on the official test set.
- DeepLab-CRF (ResNet-101) achieves 79.7% mean IOU, advancing state-of-the-art.

universitätbonn

Pascal VOC 2012: Evaluation

- VGG-16 and ResNet-101 networks pre-trained on Imagenet are used.
- Momentum and weight decay are set to 0.9 and 0.0005 respectively.
- 10 mean field iterations.
- The number of filters in fc6/fc7 layers reduced from 4096 to 1024.
  - \+ Enables 3.36 times faster training speed.
- DeepLab-CRF-LargeFOV (VGG-16) model achieves 70.3% mean IOU performance on the official test set.
- DeepLab-CRF (ResNet-101) achieves 79.7% mean IOU, advancing state-of-the-art.

universitätbonn

Pascal VOC 2012: Evaluation

- VGG-16 and ResNet-101 networks pre-trained on Imagenet are used.
- Momentum and weight decay are set to 0.9 and 0.0005 respectively.
- 10 mean field iterations.
- The number of filters in fc6/fc7 layers reduced from 4096 to 1024.
  - \+ Enables 3.36 times faster training speed.
- DeepLab-CRF-LargeFOV (VGG-16) model achieves 70.3% mean IOU performance on the official test set.
- DeepLab-CRF (ResNet-101) achieves 79.7% mean IOU, advancing state-of-the-art.

universitätbonn

Pascal VOC 2012: Evaluation

- VGG-16 and ResNet-101 networks pre-trained on Imagenet are used.
- Momentum and weight decay are set to 0.9 and 0.0005 respectively.
- 10 mean field iterations.
- The number of filters in fc6/fc7 layers reduced from 4096 to 1024.
  - \+ Enables 3.36 times faster training speed.
- DeepLab-CRF-LargeFOV (VGG-16) model achieves 70.3% mean IOU performance on the official test set.
- DeepLab-CRF (ResNet-101) achieves 79.7% mean IOU, advancing state-of-the-art.

universitätbonn

Pascal VOC 2012: Evaluation

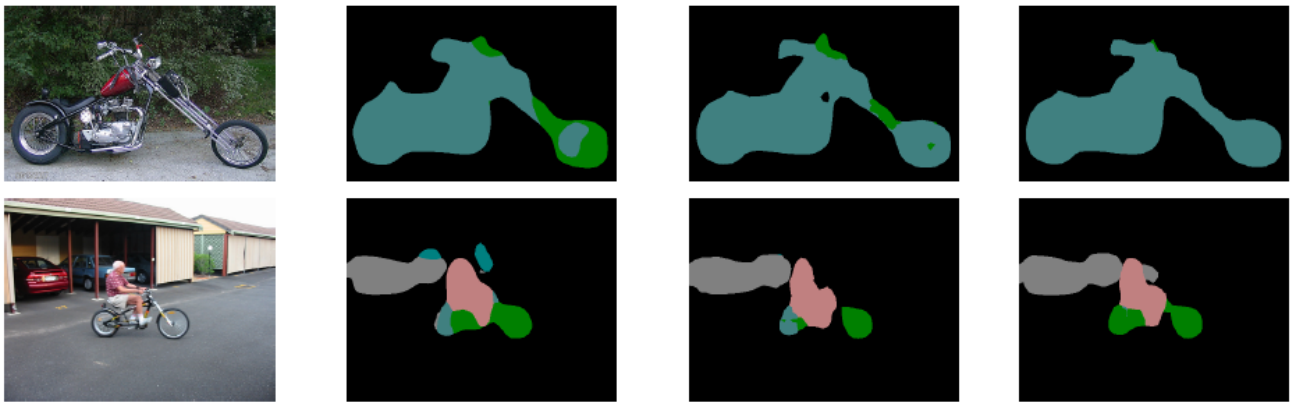
LFOV	ASPP-S	ASPP-L	CRF	**mIOU**
✓				65.76
✓			✓	69.84
	✓			66.98
	✓		✓	69.73
		✓		68.96
		✓	✓	▷ 71.57 ◁

Table 3: PASCAL VOC 2012 VAL set performance (mean IOU) for VGG-16 based DeepLab model. LargeFOV: single branch, r = 12. ASPP-S: four branches, $r = \{2, 4, 8, 12\}$. ASPP-L: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

Pascal VOC 2012: Qualitative Results

- The **ASPP-L** model, employing multiple large FOVs can successfully capture multi-scale objects and image context.



(a) Image (b) LargeFOV (c) ASPP-S (d) ASPP-L

Figure 26: Qualitative segmentation results of ASPP compared to the baseline LargeFOV model. LargeFOV: single branch, r = 12. ASPP-S: four branches, $r = \{2, 4, 8, 12\}$, ASPP-L: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

Pascal VOC 2012: Qualitative Results

- Employing the CRF further improves the performance by removing false positives and refining object boundaries.

Figure 27: Qualitative segmentation results before/after CRF.

universitätbonn

Pascal VOC 2012: Adopting Resnet-101

- Adopting ResNet-101 instead of VGG-16 significantly improves DeepLab performance.

MSC	COCO	Aug	LFOV	ASPP	CRF	**mIOU**
						68.72
✓						71.27
✓	✓					73.28
✓	✓	✓	✓			74.87
✓	✓	✓	✓			75.54
✓	✓	✓		✓		76.35
✓	✓	✓		✓	✓	▷ 77.69 ◁

Table 4: Deeplab results on PASCAL VOC 2012 VAL set.
MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset,
Aug: Randomly Scaled Input, LFOV: single branch,
ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

Pascal VOC 2012: Qualitative Results

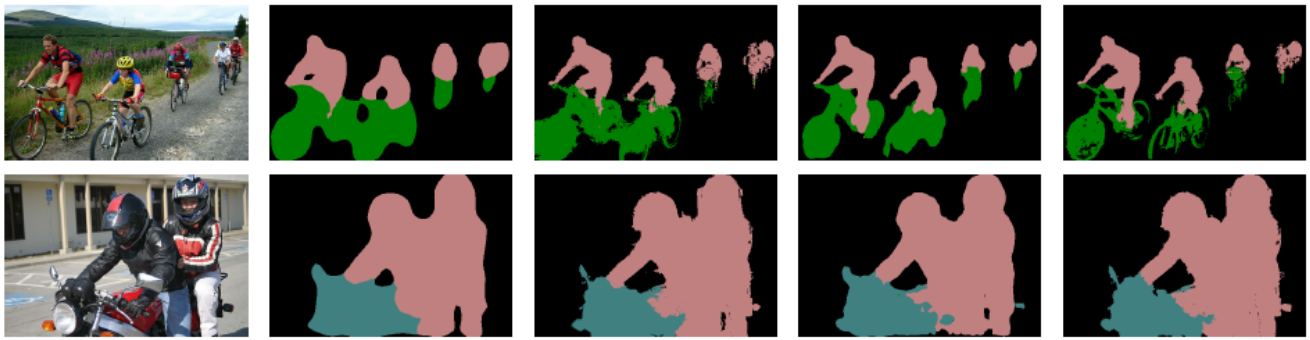


Figure 28: DeepLab results.

- Employing ResNet-101 without CRF has almost the same accuracy along object boundaries as employing VGG-16 with CRF procesing.
- Residual connections in ResNet-101 has similar effect as hyper-column features.

universitätbonn

Pascal VOC 2012: Qualitative Results

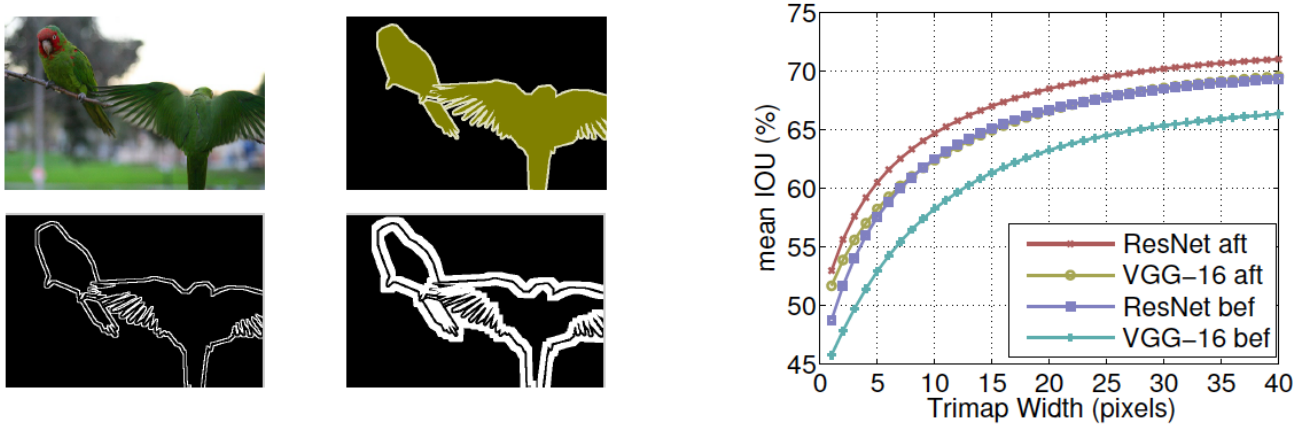


Figure 29: Trimap examples(Left) and pixel IOU(right) as a function of trimap width.

- Using trimaps, one can evaluate segmentation accuracy around boundaries[30].
- Post-processing the ResNet-101 result with a CRF further improves the accuracy result.

[30] Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

Pascal VOC 2012: Benchmark

Method	mIOU
MERL_DEEP_GCRF	73.2
CRF-RNN	74.7
POSTECH_DeconvNet_CRF_VOC	74.8
BoxSup	75.2
Context + CRF-RNN	75.3
$\mathcal{QO}_4^{mres}$	75.5
DeepLab-CRF-Attention	75.7
CentraleSuperBoundaries++	76.0
DeepLab-CRF-Attention-DT	76.3
H-ReNet + DenseCRF	76.8
LRR_4x_COCO	76.8
DPN	77.5
Adelaide_Context	77.8
Oxford_TVG_HO_CRF	77.9
Context CRF + Guidance CRF	78.1
Adelaide_VeryDeep_FCN_VOC	79.1
DeepLab-ASPP-L-CRF	72.6
DeepLab-CRF-LargeFOV-COCO	72.7
DeepLab-CRF (ResNet-101)	▷ 79.7 ◁

Table 5: Benchmark on PASCAL VOC 2012 test set.

universitätbonn

PASCAL-Context: Dataset

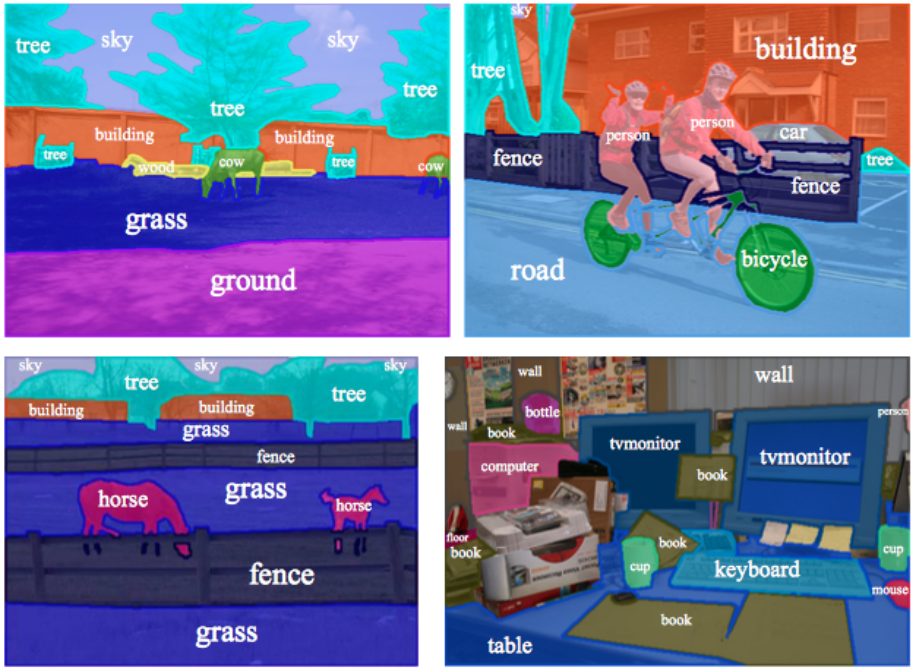


Figure 30: Sample annotations.

- PASCAL-Context[31] dataset provides semantic labels for the whole scene.
- Includes 4998 training, 5105 validation and 9637 test images.
- Out of 400+ classes, results are reported on the the most frequent 59.

[31] Mottaghi, Roozbeh, et al. "The role of context for object detection and semantic segmentation in the wild."

PASCAL-Context: Evaluation

MSC	COCO	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16						
			✓			▷ 37.6 ◁
			✓		✓	39.6
ResNet-101						
						39.6
✓		✓				41.4
✓	✓	✓				42.9
✓	✓	✓	✓			43.5
✓	✓	✓		✓		44.7
✓	✓	✓		✓	✓	45.7

Table 6: Deeplab results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Context: Evaluation

MSC	COCO	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16						
			✓			37.6
			✓		✓	▷ 39.6 ◁
ResNet-101						
						39.6
✓		✓				41.4
✓	✓	✓				42.9
✓	✓	✓	✓			43.5
✓	✓	✓		✓		44.7
✓	✓	✓		✓	✓	45.7

Table 6: Deeplab results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Context: Evaluation

MSC	COCO	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16						
			✓			37.6
			✓		✓	39.6
ResNet-101						
						▷ 39.6 ◁
✓		✓				41.4
✓	✓	✓				42.9
✓	✓	✓	✓			43.5
✓	✓	✓		✓		44.7
✓	✓	✓		✓	✓	45.7

Table 6: Deeplab results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Context: Evaluation

MSC	COCO	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16						
			✓			37.6
			✓		✓	39.6
ResNet-101						
						39.6
✓		✓				▷ 41.4 ◁
✓	✓	✓				42.9
✓	✓	✓	✓			43.5
✓	✓	✓		✓		44.7
✓	✓	✓		✓	✓	45.7

Table 6: Deeplab results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Context: Evaluation

MSC	COCO	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16						
			✓			37.6
			✓		✓	39.6
ResNet-101						
						39.6
✓		✓				41.4
✓	✓	✓				▷ 42.9 ◁
✓	✓	✓	✓			43.5
✓	✓	✓		✓		44.7
✓	✓	✓		✓	✓	45.7

Table 6: Deeplab results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Context: Evaluation

MSC	COCO	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16						
			✓			37.6
			✓		✓	39.6
ResNet-101						
						39.6
✓		✓				41.4
✓	✓	✓				42.9
✓	✓	✓	✓			▷ 43.5 ◁
✓	✓	✓		✓		44.7
✓	✓	✓		✓	✓	45.7

Table 6: Deeplab results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Context: Evaluation

MSC	COCO	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16						
			✓			37.6
			✓		✓	39.6
ResNet-101						
						39.6
✓		✓				41.4
✓	✓	✓				42.9
✓	✓	✓	✓			43.5
✓	✓	✓		✓		▷ 44.7 ◁
✓	✓	✓		✓	✓	45.7

Table 6: Deeplab results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Context: Evaluation

MSC	COCO	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16						
			✓			37.6
			✓		✓	39.6
ResNet-101						
						39.6
✓		✓				41.4
✓	✓	✓				42.9
✓	✓	✓	✓			43.5
✓	✓	✓		✓		44.7
✓	✓	✓		✓	✓	▷ 45.7 ◁

Table 6: Deeplab results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

Pascal-Context: Benchmark

- Final model yields 45.7%, outperforming the current state-of-art method.

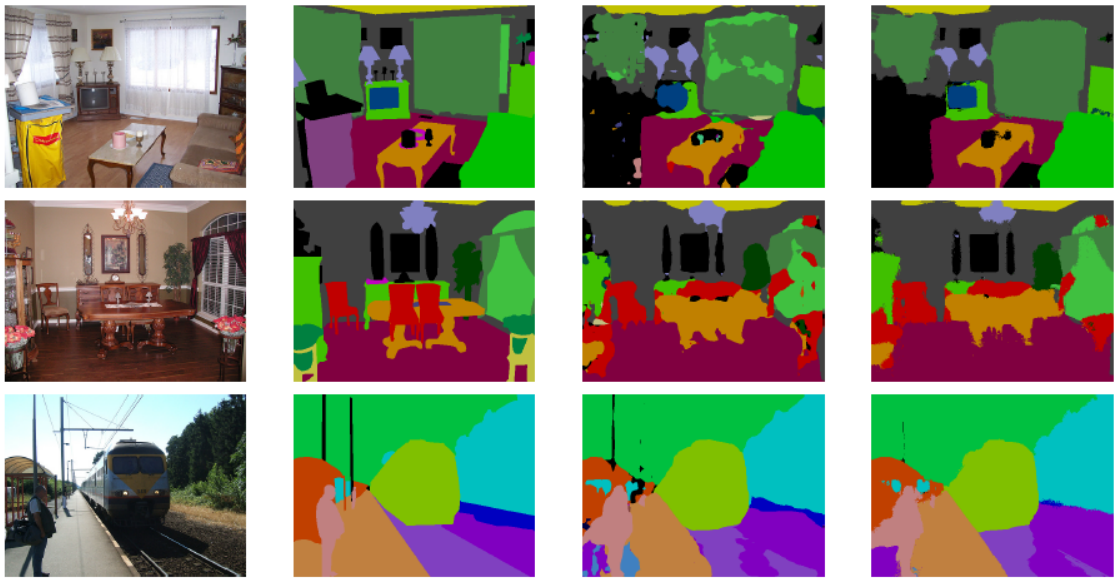
Method	**mIOU**
O2P	18.1
CFM	34.4
FCN-8s	37.8
CRF-RNN	39.3
ParseNet	40.4
BoxSup	40.5
HO_CRF	41.3
Context	43.3
VeryDeep	44.5
DeepLab-MSC-COCO-Aug-ASPP-CRF (VGG-16)	▷ 45.7 ◁

Table 7: Benchmark results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Context: Qualitative Results

- CRF helps improve the model by removing false positives and improving the prediction along object boundaries(Figure 31).



(a) Image (b) G.T. (c) Before CRF (d) After CRF

Figure 31: PASCAL-Context results.

universitätbonn

PASCAL-Person-Part: Dataset

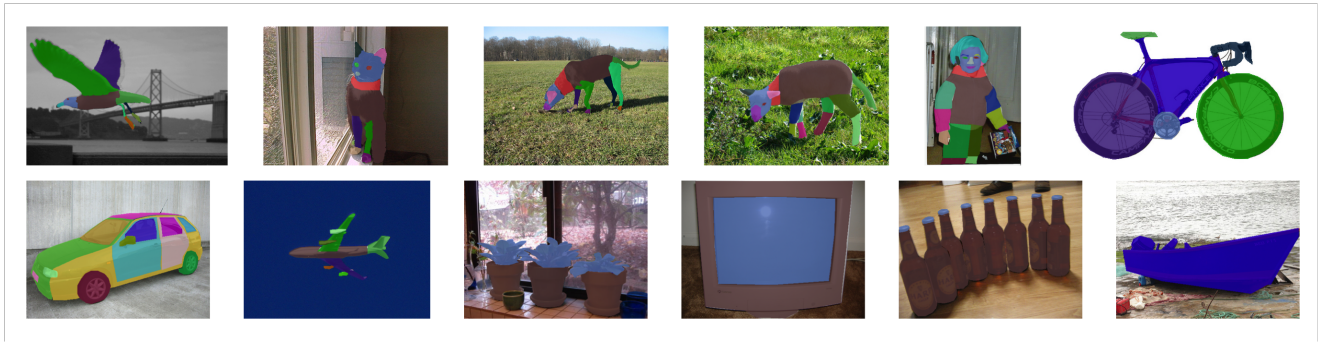


Figure 32: PASCAL-Part-Dataset[32].

- PASCAL-Part-Dataset[33] provides segmentation masks for each body part of the object.
- The dataset contains detailed part annotations for every person *e.g.* eyes, nose etc.
- Annotations are merged to be *Head, Torso, Upper/Lower Arms and Upper/Lower Legs*.

[32] http://www.stat.ucla.edu/pascal_part_dataset/pascal_part.html

[33] Chen, Xianjie, et al. "Detect what you can: Detecting and representing objects using holistic models and body parts. (2014)"

universitätbonn

PASCAL-Person-Part: Evaluation

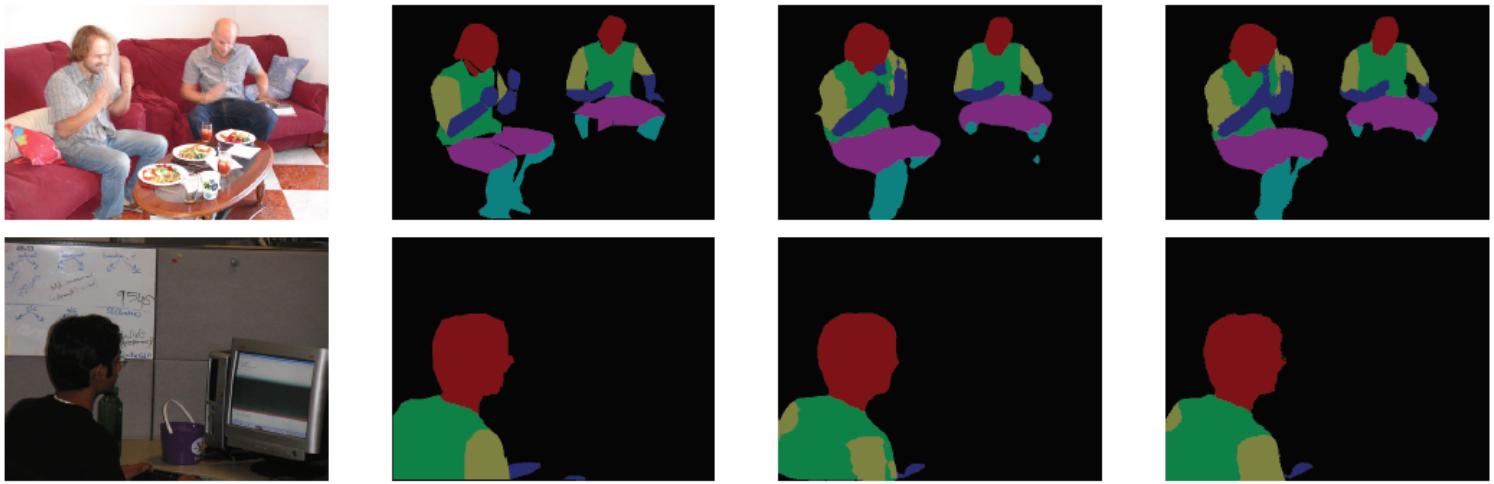
Method	MSC	COCO	Aug	LFOV	ASPP	CRF	**mIOU**
ResNet-101							
DeepLab							58.90
DeepLab	✓		✓				63.10
DeepLab	✓	✓	✓				64.40
DeepLab	✓	✓	✓			✓	▷ 64.94 ◁
VGG-16							
DeepLab	✓	✓	✓	✓			62.18
DeepLab	✓	✓	✓		✓		62.76
Attention							56.39
HAZN							57.54
LG-LSTM							57.97
Graph LSTM							60.16

Table 8: DeepLab results on PASCAL-Person-Part VAL set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

PASCAL-Person-Part: Qualitative Results

- Dense CRF is used to post process final output, advancing the state-of-the-art on the PASCAL-Person-Part dataset.



(a) Image (b) G.T. (c) Before CRF (d) After CRF

Figure 33: Qualitative results on PASCAL-Person-Part dataset.

universitätbonn

Cityscapes: Dataset

Figure 34: Cityscapes dataset[34].

- 5000 images collected in street scenes from 50 different cities[34].
- 2975, 500, and 1525 training, validation, and test images respectively.
- Out of 30, 19 most frequent semantic labels (belonging to 7 super categories: ground, construction, object, nature, sky, human, and vehicle) are used for evaluation.

universitätbonn

[34] MLA Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding. (2016)"

Cityscapes: Evaluation

- Model trained with subsampled images or each image is split into overlapped regions(full).

Full	Aug	LargeFOV	ASPP	CRF	**mIOU**
VGG-16					
		✓			62.97
		✓		✓	64.18
✓		✓			64.89
✓		✓		✓	65.94
Resnet-101					
✓					66.6
✓		✓			69.2
✓			✓		70.4
✓	✓		✓		71.0
✓	✓		✓	✓	71.4

Table 9: DeepLab results on Cityscapes validation set. Full: Model trained with full resolution images(2048×1024). Aug: Randomly Scaled Input, LargeFOV: single branch, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

universitätbonn

Cityscapes: Benchmark

Table 10: Benchmark results on PASCAL Context test set. MSC: DCNN images at scale $= \{0.5, 0.75, 1\}$, COCO: MS-COCO dataset, Aug: Randomly Scaled Input, ASPP: four branches, $r = \{6, 12, 18, 24\}$.

Method	**mIOU**
pre-release version of dataset	
Adelaide_Context	66.4
FCN-8s	65.3
DeepLab-CRF-LargeFOV-StrongWeak	64.8
DeepLab-CRF-LargeFOV	63.1
CRF-RNN	62.5
DPN	59.1
Segnet basic	57.0
Segnet extended	56.1
official version	
Adelaide_Context	71.6
Dilation10	67.1
DPN	66.8
Pixel-level Encoding	64.3
DeepLab-CRF (ResNet-101)	70.4

universitätbonn

Cityscapes: Qualitative Results

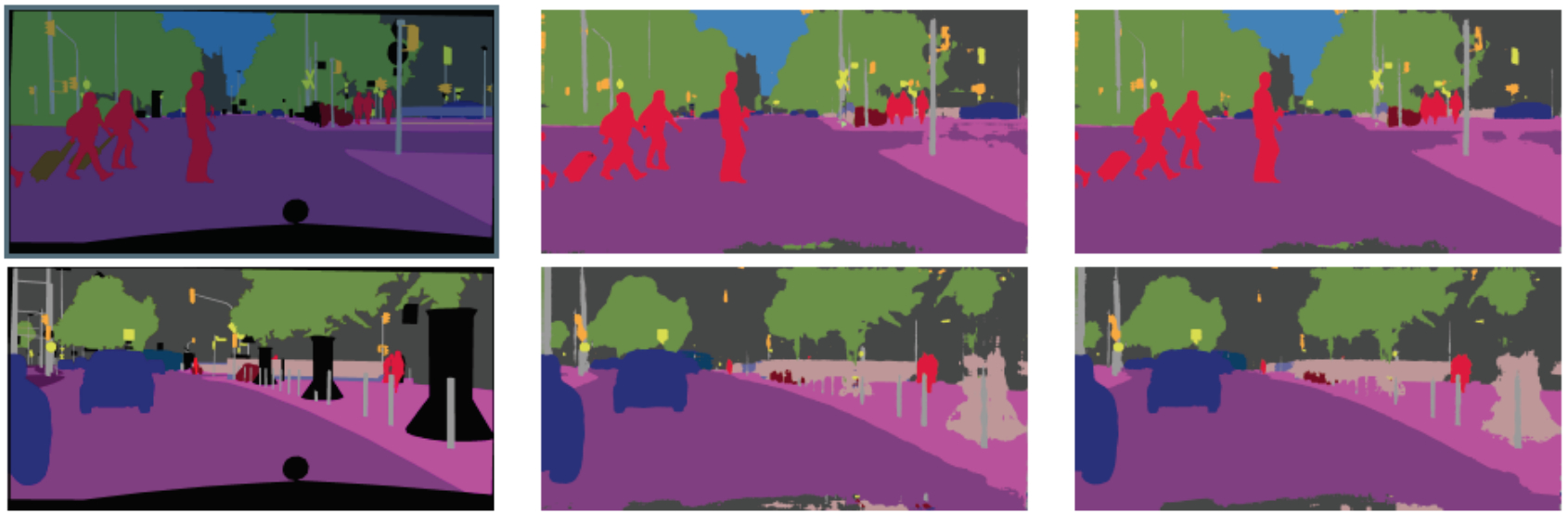
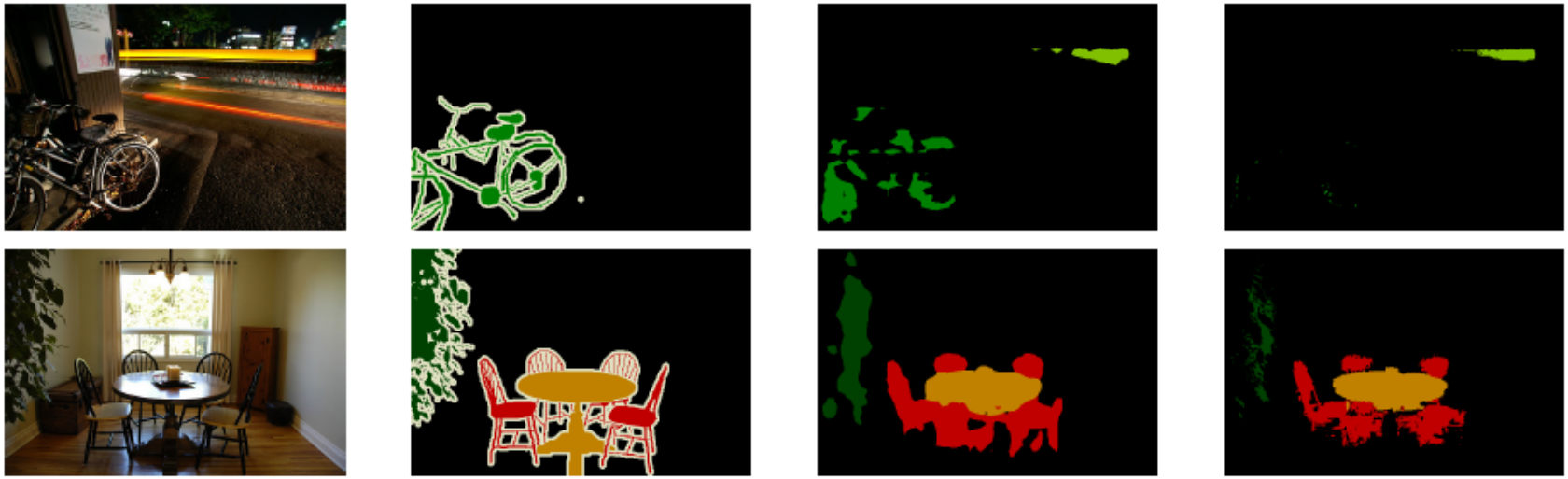


Figure 35: Qualitative results on Cityscapes dataset.

universitätbonn

Failure Modes



(a) Image (b) G.T. (c) Before CRF (d) After CRF

Figure 36: Failure modes.

- Deeplab fails to capture the delicate boundaries of objects, such as bicycle and chair as shown in Fig. 36.
- CRF post processing fails since the *unary term* is not confident enough.

universitätbonn

Future Direction

- Weaker supervision[35] has been conducted in a number of papers, not requiring that pixel-level semantic annotations are available for the whole training set.
- End-to-end CRF+DCNN learning methods could be investigated.
- DeepLab can be applied on other end-to-end regression tasks, i.e depth/optical flow prediction, super-resolution estimation, denoising, demosaicing, bottom-up saliency, keypoint detection etc.
- To capture delicate boundaries, fine-grained features can be incorporated.

universitätbonn

[35] Papandreou, George, et al. "Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. (2015)"

Future Direction

- Weaker supervision[35] has been conducted in a number of papers, not requiring that pixel-level semantic annotations are available for the whole training set.
- End-to-end CRF+DCNN learning methods could be investigated.
- DeepLab can be applied on other end-to-end regression tasks, i.e depth/optical flow prediction, super-resolution estimation, denoising, demosaicing, bottom-up saliency, keypoint detection etc.
- To capture delicate boundaries, fine-grained features can be incorporated.

universitätbonn

[35] Papandreou, George, et al. "Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. (2015)"

Future Direction

- Weaker supervision[35] has been conducted in a number of papers, not requiring that pixel-level semantic annotations are available for the whole training set.
- End-to-end CRF+DCNN learning methods could be investigated.
- DeepLab can be applied on other end-to-end regression tasks, i.e depth/optical flow prediction, super-resolution estimation, denoising, demosaicing, bottom-up saliency, keypoint detection etc.
- To capture delicate boundaries, fine-grained features can be incorporated.

universitätbonn

[35] Papandreou, George, et al. "Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. (2015)"

Future Direction

- Weaker supervision[35] has been conducted in a number of papers, not requiring that pixel-level semantic annotations are available for the whole training set.
- End-to-end CRF+DCNN learning methods could be investigated.
- DeepLab can be applied on other end-to-end regression tasks, i.e depth/optical flow prediction, super-resolution estimation, denoising, demosaicing, bottom-up saliency, keypoint detection etc.
- To capture delicate boundaries, fine-grained features can be incorporated.

universitätbonn

[35] Papandreou, George, et al. "Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. (2015)"

Appendix: Mean Field Approximation

- Key idea[36]: Instead of computing the exact distribution, find a distribution $\mathcal{Q}(\mathbf{X})$ that minimizes the KL-divergence $\mathbf{D}(\mathcal{Q}\|P)$.
- Distribution $\mathcal{Q}$ can be expressed as a product of independent marginals, $\mathcal{Q}(\mathbf{X}) = \prod_i \mathcal{Q}_i(\mathbf{X_i})$

Algorithm 1 Mean field for FC-CRF

Initialize $\mathcal{Q}$ $\quad \triangleright\ \mathcal{Q}_i(x_i) \leftarrow \frac{1}{Z_i} \exp\{-\theta_i(x_i)\}$

while not converged **do**

[36] Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

universitätbonn

Appendix: Mean Field Approximation

- Key idea[36]: Instead of computing the exact distribution, find a distribution $\mathcal{Q}(\mathbf{X})$ that minimizes the KL-divergence $\mathbf{D}(\mathcal{Q}\|P)$.
- Distribution $\mathcal{Q}$ can be expressed as a product of independent marginals, $\mathcal{Q}(\mathbf{X}) = \prod_i \mathcal{Q}_i(\mathbf{X_i})$

Algorithm 1 Mean field for FC-CRF

Initialize $\mathcal{Q}$ $\quad \triangleright\ \mathcal{Q}_i(x_i) \leftarrow \frac{1}{Z_i} \exp\{-\theta_i(x_i)\}$

while not converged **do**

$\tilde{\mathcal{Q}}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f_i}, \mathbf{f_j}) \mathcal{Q}_j(l)$ for all m $\quad \triangleright$ **MP** from all X_j to all X_i

[36] Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

universitätbonn

Appendix: Mean Field Approximation

- Key idea[36]: Instead of computing the exact distribution, find a distribution $\mathcal{Q}(\mathbf{X})$ that minimizes the KL-divergence $\mathbf{D}(\mathcal{Q}\|P)$.
- Distribution $\mathcal{Q}$ can be expressed as a product of independent marginals, $\mathcal{Q}(\mathbf{X}) = \prod_i \mathcal{Q}_i(\mathbf{X_i})$

Algorithm 1 Mean field for FC-CRF

Initialize $\mathcal{Q}$ ▷ $\mathcal{Q}_i(x_i) \leftarrow \frac{1}{Z_i}\exp\{-\theta_i(x_i)\}$

while not converged **do**

$\tilde{\mathcal{Q}}_i^{(m)}(l) \leftarrow \sum_{j\neq i} k^{(m)}(\mathbf{f_i}, \mathbf{f_j})\mathcal{Q}_j(l)$ for all m ▷ **MP** from all X_j to all X_i

$\hat{\mathcal{Q}}_i(x_i) \leftarrow \sum_{l\in\mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \tilde{\mathcal{Q}}_i^{(m)}$ ▷ **Compatibility transform**

universitätbonn

[36] Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

Appendix: Mean Field Approximation

- Key idea[36]: Instead of computing the exact distribution, find a distribution $\mathcal{Q}(\mathbf{X})$ that minimizes the KL-divergence $\mathbf{D}(\mathcal{Q}\|P)$.
- Distribution $\mathcal{Q}$ can be expressed as a product of independent marginals, $\mathcal{Q}(\mathbf{X}) = \prod_i \mathcal{Q}_i(\mathbf{X_i})$

Algorithm 1 Mean field for FC-CRF

Initialize $\mathcal{Q}$ ▷ $\mathcal{Q}_i(x_i) \leftarrow \frac{1}{Z_i}\exp\{-\theta_i(x_i)\}$

while not converged **do**

$\tilde{\mathcal{Q}}_i^{(m)}(l) \leftarrow \sum_{j\neq i} k^{(m)}(\mathbf{f_i}, \mathbf{f_j})\mathcal{Q}_j(l)$ for all m ▷ **MP** from all X_j to all X_i

$\hat{\mathcal{Q}}_i(x_i) \leftarrow \sum_{l\in\mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \tilde{\mathcal{Q}}_i^{(m)}$ ▷ **Compatibility transform**

$\mathcal{Q}_i(x_i) \leftarrow \exp\{-\theta_i(x_i) - \hat{\mathcal{Q}}_i(x_i)\}$ ▷ **Local Update**

[36] Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

universitätbonn

Appendix: Mean Field Approximation

- Key idea[36]: Instead of computing the exact distribution, find a distribution $\mathcal{Q}(\mathbf{X})$ that minimizes the KL-divergence $\mathbf{D}(\mathcal{Q}\|P)$.
- Distribution $\mathcal{Q}$ can be expressed as a product of independent marginals, $\mathcal{Q}(\mathbf{X}) = \prod_i \mathcal{Q}_i(\mathbf{X_i})$

Algorithm 1 Mean field for FC-CRF

Initialize $\mathcal{Q}$ ▷ $\mathcal{Q}_i(x_i) \leftarrow \frac{1}{Z_i}\exp\{-\theta_i(x_i)\}$

while not converged **do**

$\tilde{\mathcal{Q}}_i^{(m)}(l) \leftarrow \sum_{j\neq i} k^{(m)}(\mathbf{f_i}, \mathbf{f_j})\mathcal{Q}_j(l)$ for all m ▷ **MP** from all X_j to all X_i

$\hat{\mathcal{Q}}_i(x_i) \leftarrow \sum_{l\in\mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \tilde{\mathcal{Q}}_i^{(m)}$ ▷ **Compatibility transform**

$\mathcal{Q}_i(x_i) \leftarrow \exp\{-\theta_i(x_i) - \hat{\mathcal{Q}}_i(x_i)\}$ ▷ **Local Update**

Normalize $\mathcal{Q}_i(x_i)$

end while

---

[36] Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

universitätbonn

Appendix: Mean Field Approximation

- Qualitative results:

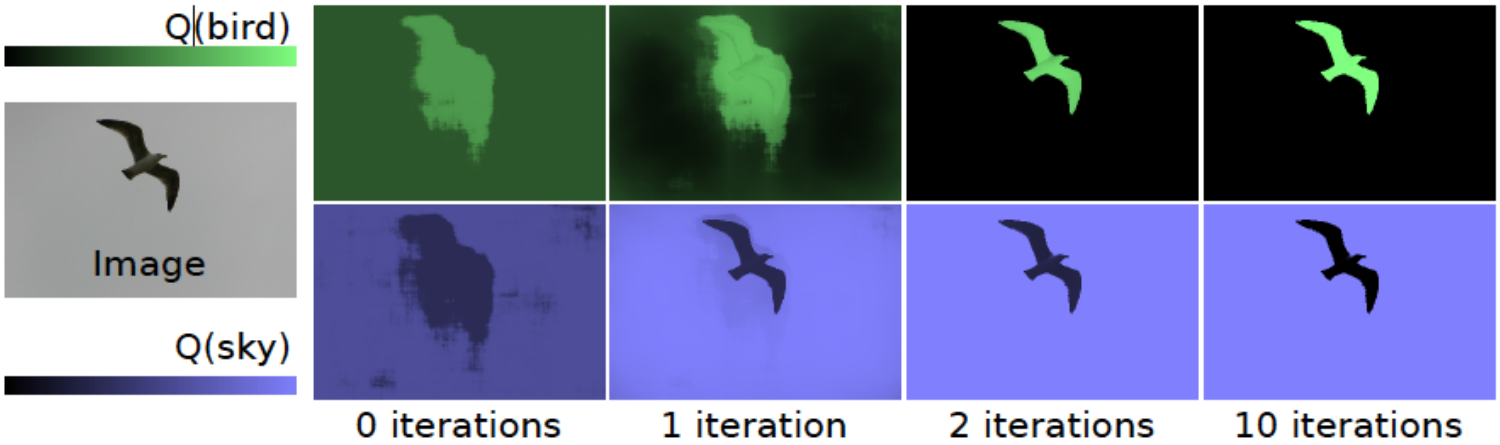


Figure 37: Distributions $\mathcal{Q}(X_i = \text{'bird'})$ (top) and $\mathcal{Q}(X_i = \text{'sky'})$ (bottom)

.

universitätbonn

Appendix: Mean Field Approximation

- *Compatibility transform* and the *local update* run in linear time.
- MP step can be expressed as a convolution with a Gaussian kernel in feature space.
- By the sampling theorem[37], function $\mathcal{Q}(I)$ can be reconstructed from a set of samples whose spacing is proportional to the standard deviation of the filter.
- Use truncated Gaussian, where all values beyond two standard deviations are set to zero.
- Perform the convolution by downsampling.
- Convolution computed at each sample uses values from a constant number of neighbour samples.
- This implies that approx. MP can be performed in $O(N)$.

universitätbonn

[37] S. W. Smith. The scientist and engineer's guide to digital signal processing.