

MA-INF 2221 - Seminar Visual Computing

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

Can Yuce¹

yuce@uni-bonn.de

¹Department of Informatics
University of Bonn

July 1, 2017



Outline

- 1 Outline
- 2 Problem Statement and Motivation
- 3 Connections with Other Work
- 4 Challenges
- 5 DeepLab: Key Technical Ideas
- 6 DeepLab: Main Contributions
- 7 DeepLab: Experimental Set-Up
- 8 Future Direction
- 9 Appendix

Semantic Image Segmentation

Semantic segmentation is the task of

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

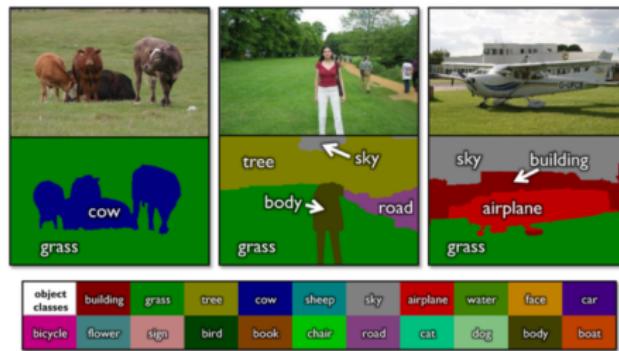


Figure 1: TextronBoost applied to the MSRC 21 Class Database.
Example semantic segmentation results¹.

¹Shotton, Jamie, et al. "Textronboost 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.

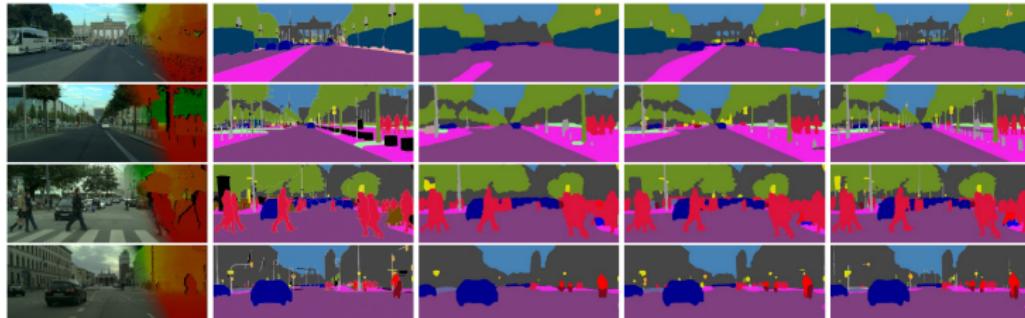


Figure 2: The Cityscapes Dataset for Semantic Urban Scene Understanding².



universität bonn

²See more at <https://www.cityscapes-dataset.com>.

Why is Semantic Segmentation Important?

Medical purposes

- Segmenting tumours, organs, dental cavities etc.

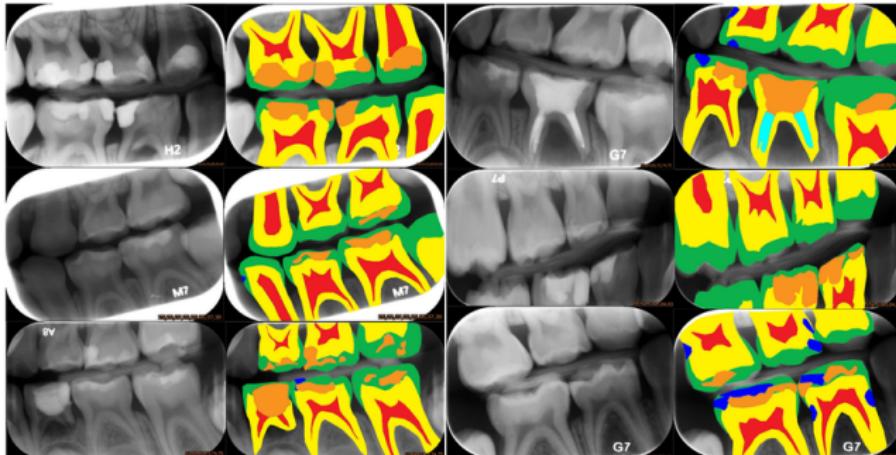


Figure 3: Segmentation of dental cavities³.

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

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.

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

Semantic Segmentation Before Deep Learning

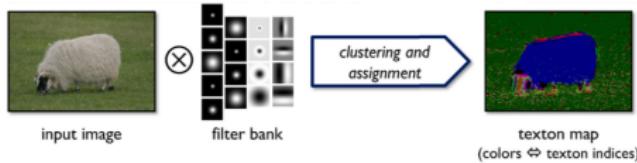


Figure 5: The process of image textonization¹.

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.

Semantic Segmentation Before Deep Learning

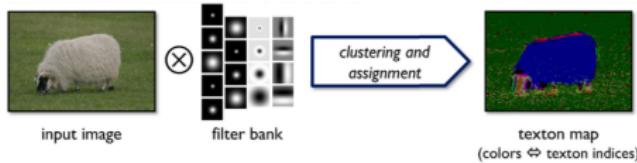


Figure 5: The process of image textonization¹.

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.

Semantic Segmentation Before Deep Learning

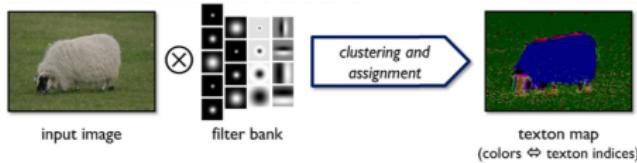


Figure 5: The process of image textonization¹.

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.

Semantic Segmentation Before Deep Learning

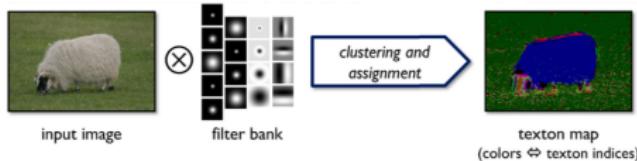


Figure 5: The process of image textonization¹.

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.

Semantic Segmentation Before Deep Learning

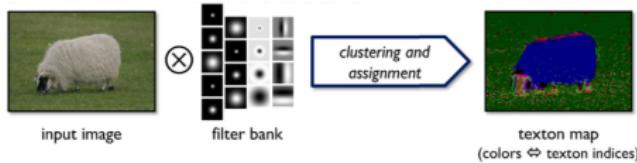


Figure 5: The process of image textonization¹.

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.

Flat Classifiers

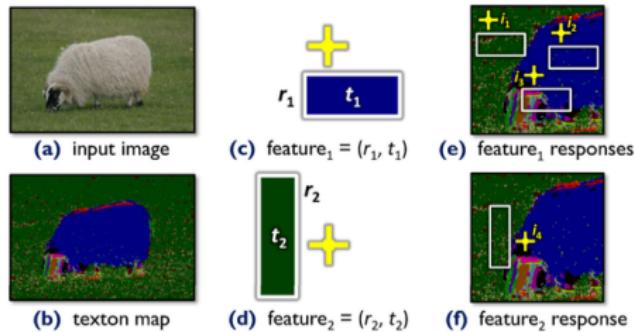


Figure 6: Each weak learner is a decision stump based on the feature response⁵.

Boosting/Ensemble Methods,

- Random Forests,
- Support Vector Machines etc.



⁵Shotton, Jamie, et al. "Textronboost 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.

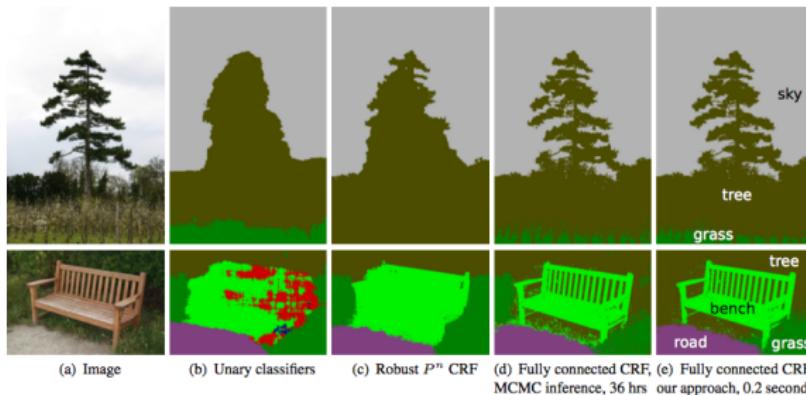


Figure 7: Pixel-level classification with a fully connected CRF.⁶

⁶Philipp 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 quantisation⁷:

- 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.

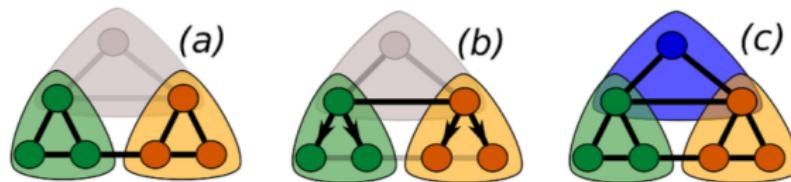


Figure 8: Pixel-level classification with a fully connected CRF.

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

Deep Learning and Semantic Segmentation

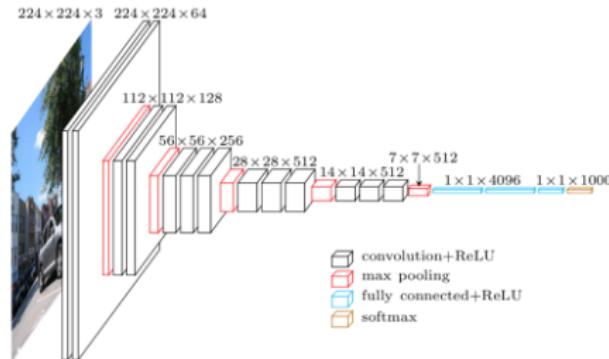


Figure 9: VGG-16 network layer by layer⁸.

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.

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

Deep Learning and Semantic Segmentation

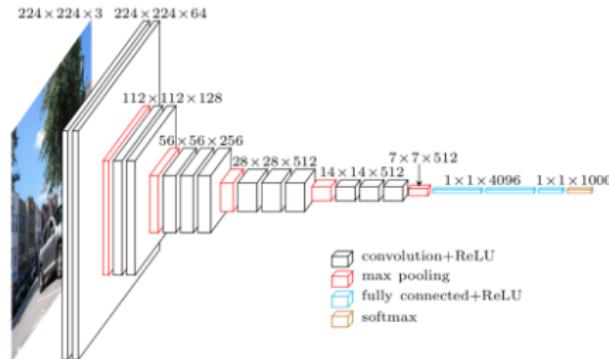


Figure 9: VGG-16 network layer by layer⁸.

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.

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

DCNN + Region Proposals

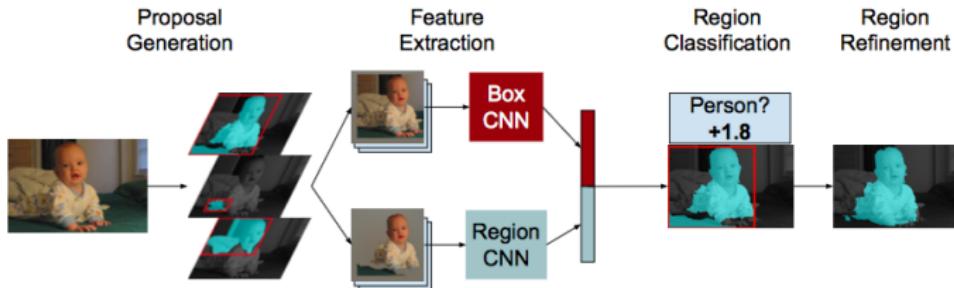


Figure 10: Simultaneous Detection and Segmentation Pipeline.⁹.

- 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.



universität bonn

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

DCNN + Region Proposals

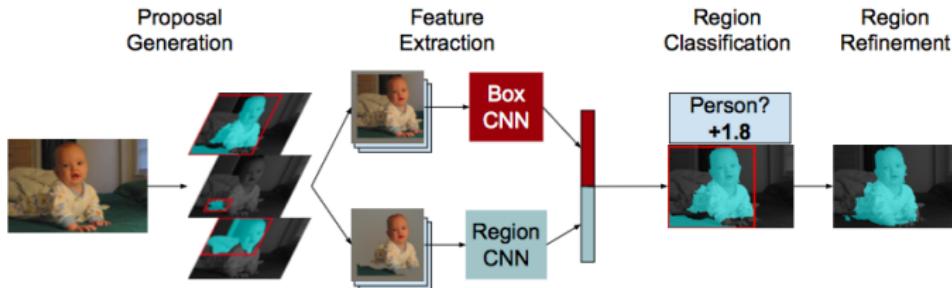


Figure 10: Simultaneous Detection and Segmentation Pipeline.⁹.

- 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.



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

DCNN + Region Proposals

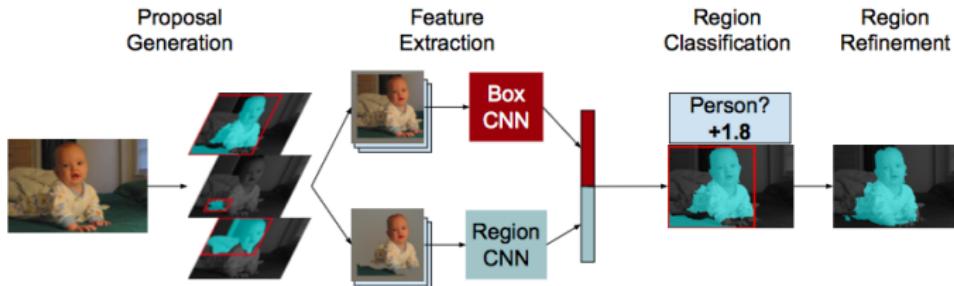


Figure 10: Simultaneous Detection and Segmentation Pipeline.⁹.

- 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.



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

DCNN Features + Segmentation Proposal

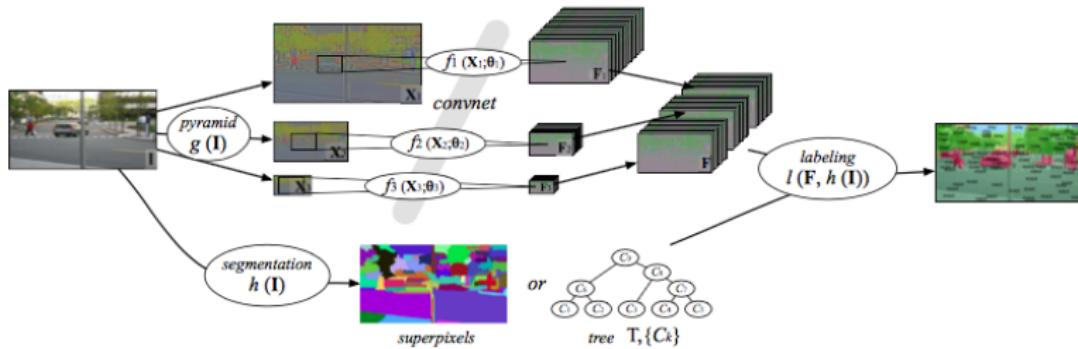


Figure 11: Scene labeling system¹⁰.

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

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

DCNN Features + Segmentation Proposal

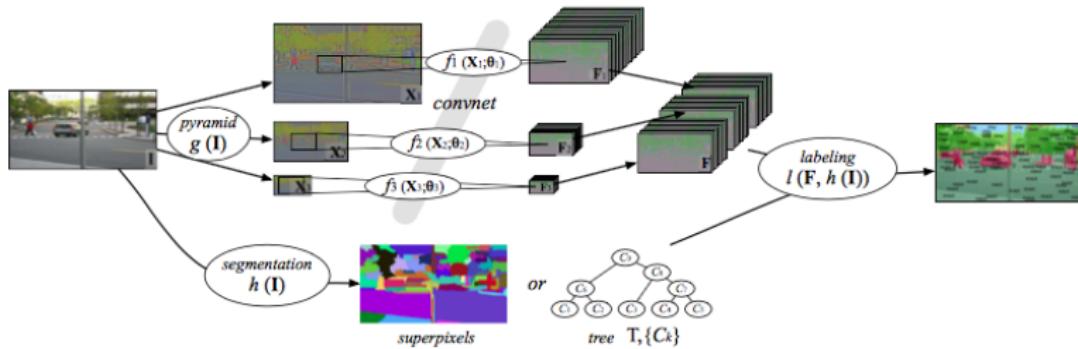


Figure 11: Scene labeling system¹⁰.

- 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.



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

DCNN Features + Segmentation Proposal

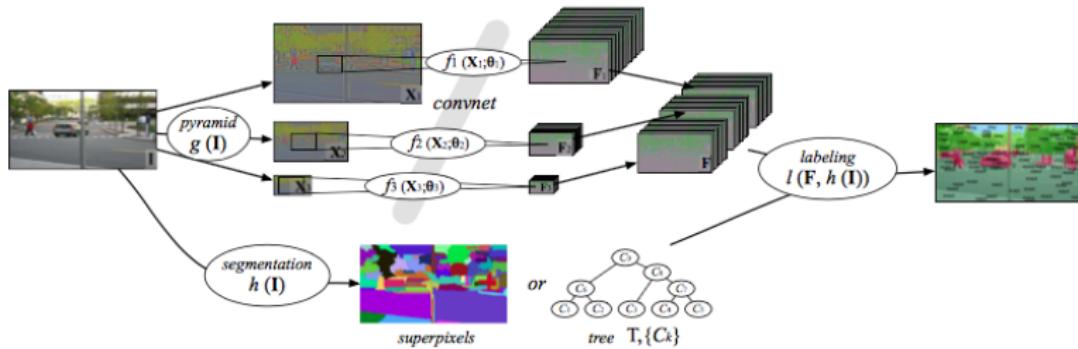


Figure 11: Scene labeling system¹⁰.

- 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.

¹⁰Farabet, 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 heatmap¹¹).
- Transfer learning using pretrained DCNN(i.e. VggNet, Resnet).

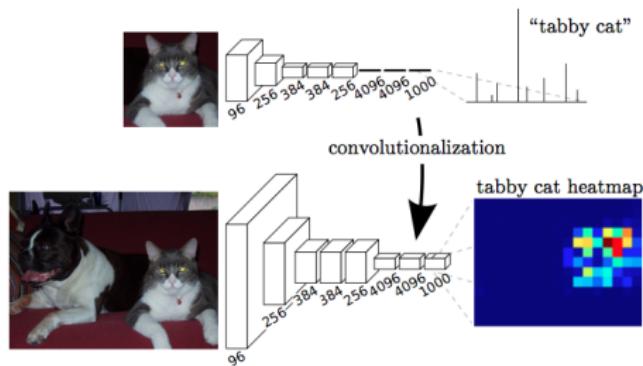


Figure 12: Transforming all the fully connected layers to convolutional layers.



universität bonn

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 heatmap¹¹).
- Transfer learning using pretrained DCNN(i.e. VggNet, Resnet).

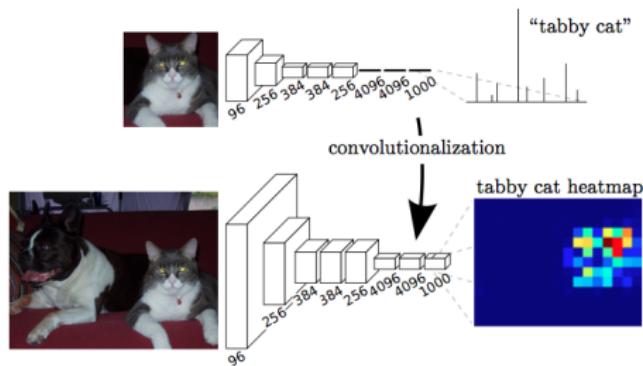


Figure 12: Transforming all the fully connected layers to convolutional layers.



universität bonn

¹¹Long, 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 heatmap¹¹).
- Transfer learning using pretrained DCNN(i.e. VggNet, Resnet).

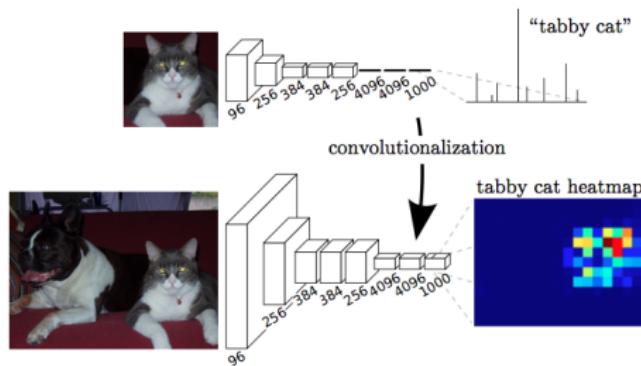


Figure 12: Transforming all the fully connected layers to convolutional layers.



universität bonn

¹¹Long, 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.

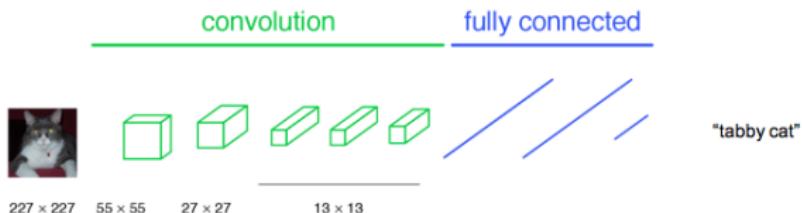


Figure 13: A classification network¹².



¹²Long, 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-pooling layers.



¹²Long, 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-pooling layers.



universität bonn

¹²Long, 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-pooling layers.



¹²Long, 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 network¹³.
- Define scale-invariant mean squared error¹⁴:

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

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

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



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

Challenges

② Existence of objects at multiple scales

- ShareNet: Resizes the input image to several scales and passes each through a shared deep network¹³.
- Define scale-invariant mean squared error¹⁴:

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

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

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



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

Challenges

② Existence of objects at multiple scales

- Spatial pyramid pooling¹⁵: Generates a fixed-length representation regardless of image size/scale

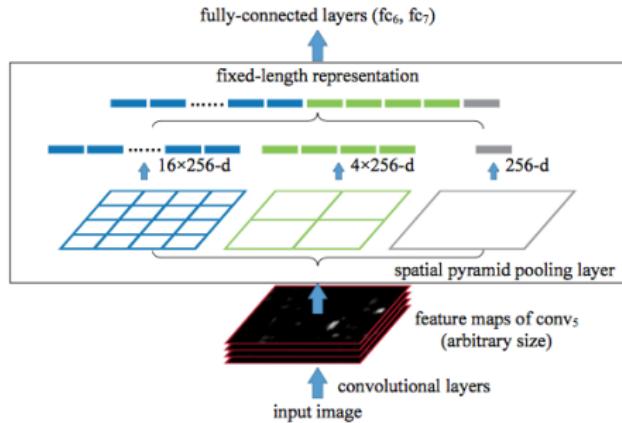


Figure 13: Network structure with a spatial pyramid pooling layer.

¹⁵He, 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 output¹⁶.

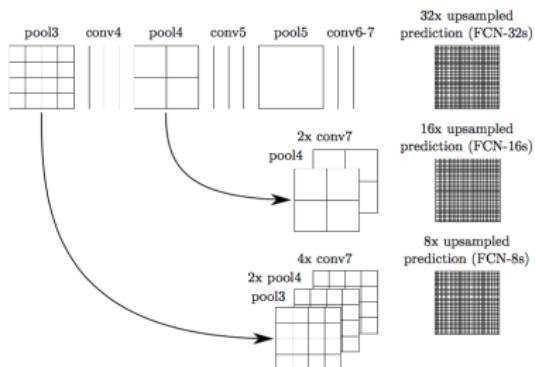


Figure 14: SkipNet Architecture.

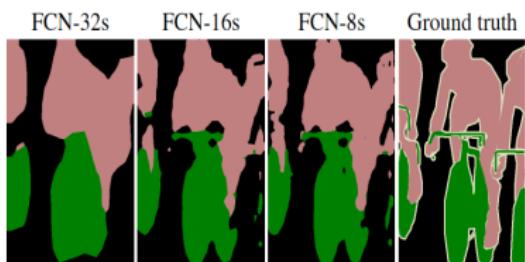


Figure 15: Refining fully convolutional nets by fusing information from layers with different strides¹⁶.



universität bonn

¹⁶Long, 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 vector¹⁷.

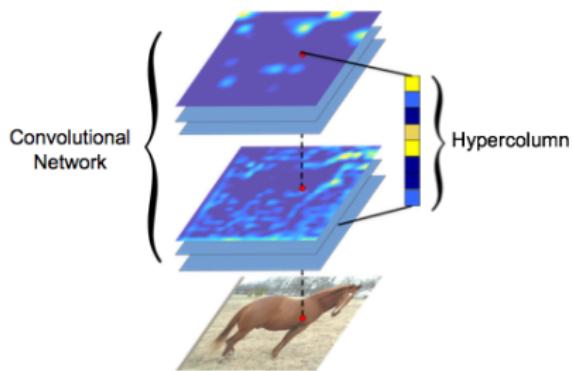


Figure 16: Hypercolumns



universität bonn

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

Key Technical Ideas

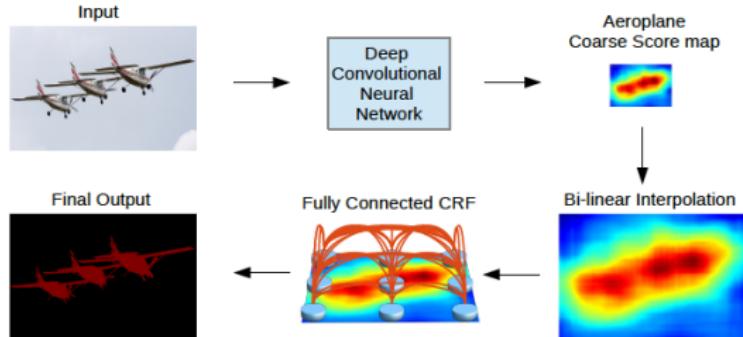


Figure 17: Model Illustration.

- 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 $32\times$ down to $8\times$).
- **Bilinear interpolation** stage to enlarge the feature maps.
- A fully connected CRF applied to refine the segmentation result.



Key Technical Ideas

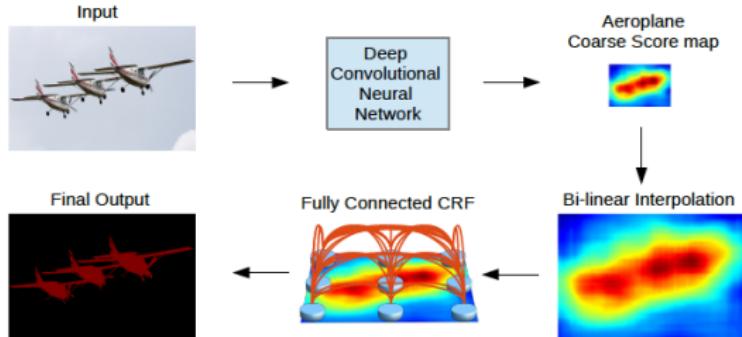


Figure 17: Model Illustration.

- 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 $32\times$ down to $8\times$).
- **Bilinear interpolation** stage to enlarge the feature maps.
- A fully connected CRF applied to refine the segmentation result.

Key Technical Ideas

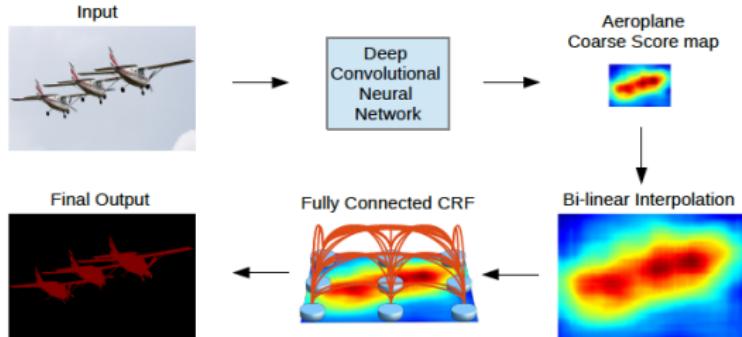


Figure 17: Model Illustration.

- 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 $32\times$ down to $8\times$).
- **Bilinear interpolation** stage to enlarge the feature maps.
- A fully connected CRF applied to refine the segmentation result.

Key Technical Ideas

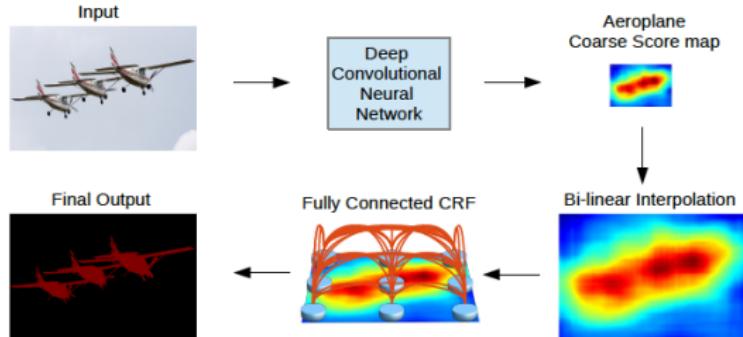


Figure 17: Model Illustration.

- 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 $32\times$ down to $8\times$).
- **Bilinear interpolation** stage to enlarge the feature maps.
- A fully connected CRF applied to refine the segmentation result.

Atrous Convolution

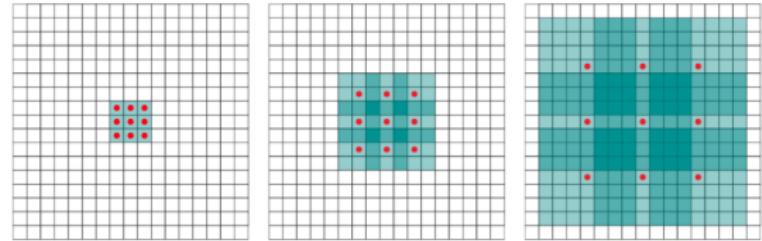


Figure 18: Atrous filtering¹⁸.

- 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:

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

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

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



Atrous Convolution

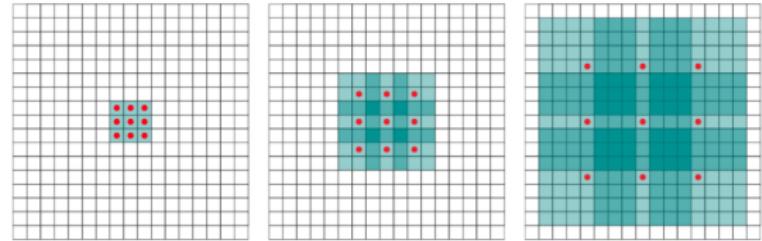


Figure 18: Atrous filtering¹⁸.

- 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:

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

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

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

Atrous Convolution

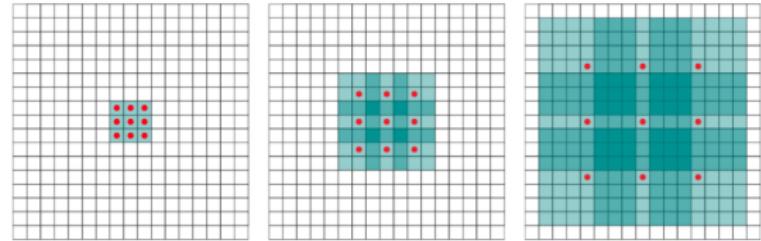


Figure 18: Atrous filtering¹⁸.

- 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:

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

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

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

Atrous Convolution

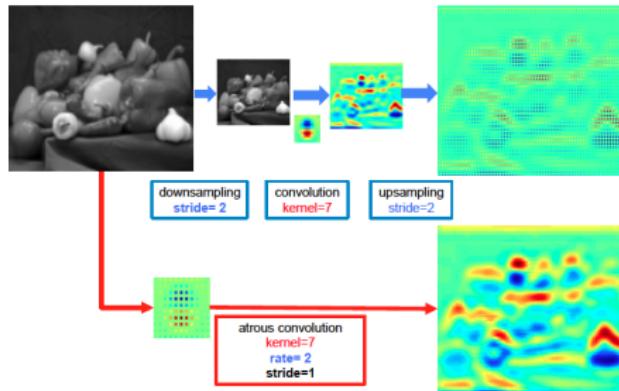


Figure 19: Atrous filtering in 2D

- 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.

Atrous Convolution

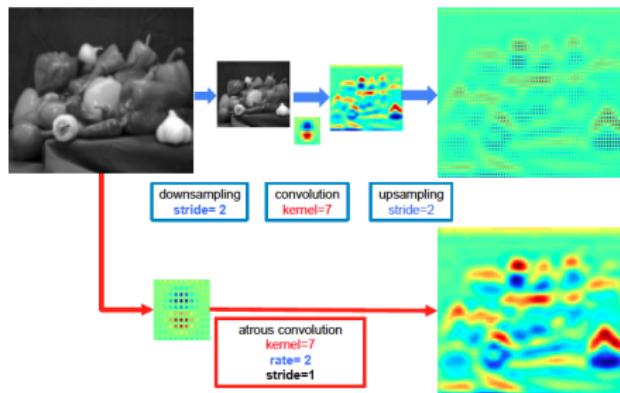


Figure 19: Atrous filtering in 2D

- 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.

Atrous Convolution

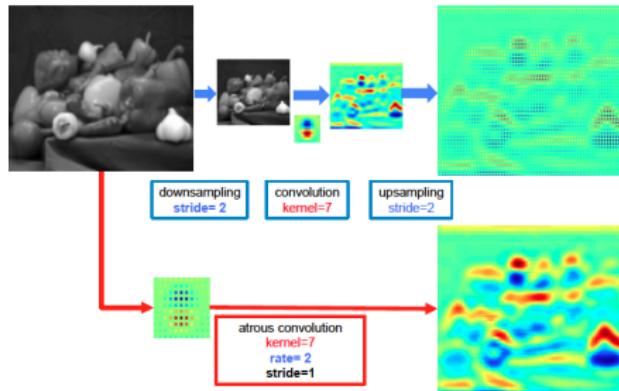


Figure 19: Atrous filtering in 2D

- 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.

Atrous Convolution

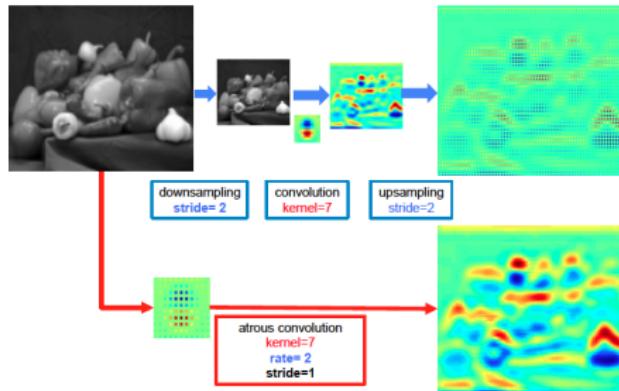


Figure 19: Atrous filtering in 2D

- 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.

Atrous Convolution

Table 1: VGG₁₆: 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	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](#)).



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.



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.



Atrous Spatial Pyramid Pooling (ASPP)

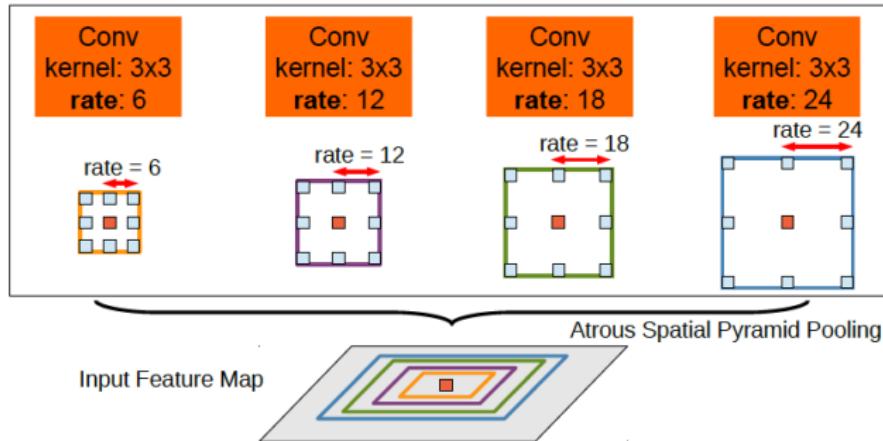


Figure 20: Atrous Spatial Pyramid Pooling (ASPP).

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

Atrous Spatial Pyramid Pooling (ASPP)

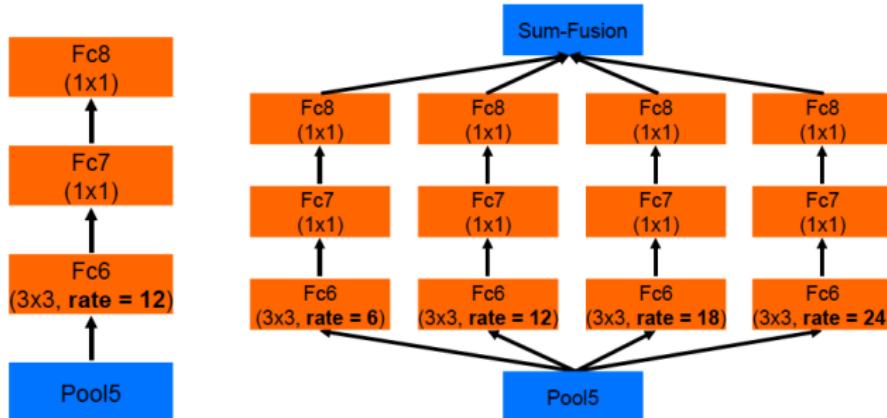


Figure 21: Atrous Spatial Pyramid Pooling (ASPP).

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

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

Atrous Spatial Pyramid Pooling (ASPP)

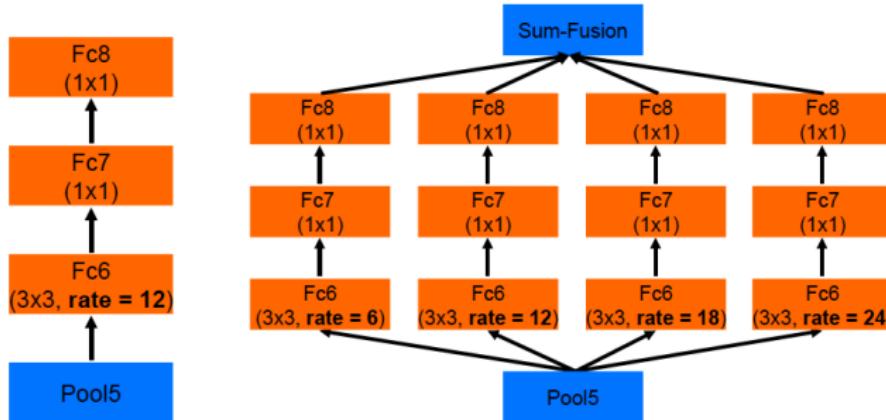


Figure 21: Atrous Spatial Pyramid Pooling (ASPP).

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

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

Fully Connected CRF

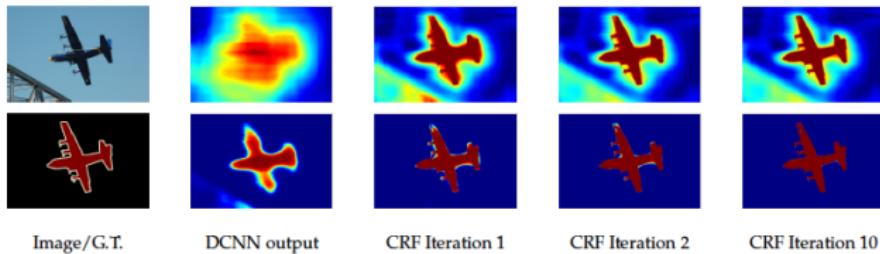


Figure 22: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane.

- 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 boundaries²¹.
- To overcome these problems, **fully connected CRF** is utilized.

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

Fully Connected CRF

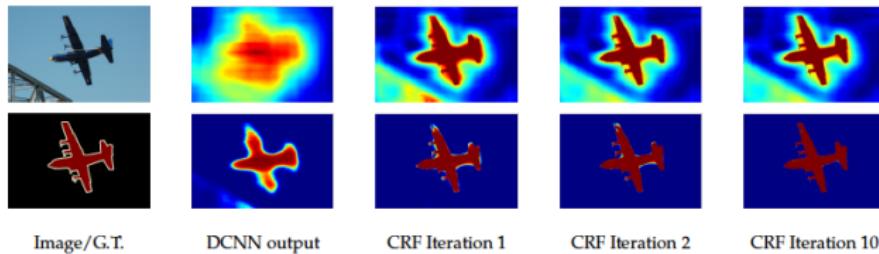


Figure 22: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane.

- 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 boundaries²¹.
- To overcome these problems, **fully connected CRF** is utilized.

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

Fully Connected CRF

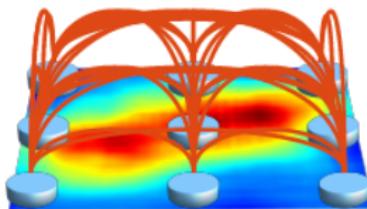


Figure 23: Fully Connected CRF.

- Gibbs energy²² 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

$$\theta_i(x_i) = -\log P(x_i)$$

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

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

Fully Connected CRF

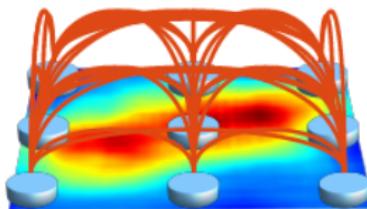


Figure 23: Fully Connected CRF.

- Gibbs energy²² 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

$$\theta_i(x_i) = -\log P(x_i)$$

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

²²Krähenbühl 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:

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 \exp \left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\alpha^2} - \frac{\|\mathbf{l}_i - \mathbf{l}_j\|^2}{2\sigma_\beta^2} \right) + w_2 \exp \left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\gamma^2} \right) \right]$$

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

- Gaussian potentials are calculated in different feature spaces.



universität bonn

Fully Connected CRF

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

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 \exp \left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\alpha^2} - \frac{\|\mathbf{l}_i - \mathbf{l}_j\|^2}{2\sigma_\beta^2} \right) + w_2 \exp \left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\gamma^2} \right) \right]$$

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

- Gaussian potentials are calculated in different feature spaces.



universität bonn

Fully Connected CRF

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

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 \exp \left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\alpha^2} - \frac{\|\mathbf{l}_i - \mathbf{l}_j\|^2}{2\sigma_\beta^2} \right) + w_2 \exp \left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_\gamma^2} \right) \right]$$

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

- Gaussian potentials are calculated in different feature spaces.



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 inference²³.
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions²⁴.
- 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.

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

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

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 inference²³.
 - Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions²⁴.
 - 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.

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

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

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 inference²³.
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions²⁴.
- 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.

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

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

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 inference²³.
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions²⁴.
- 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.

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

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

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 inference²³.
- Mean field methods perform approximate probabilistic inference by searching within a tractable subset of distributions²⁴.
- 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.

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

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

DeepLab: Experimental Set-Up

- VGG-16 and ResNet-101 trained on ImageNet²⁵ dataset are employed.
- Evaluation is performed 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.

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



DeepLab: Experimental Set-Up

- VGG-16 and ResNet-101 trained on ImageNet²⁵ dataset are employed.
- Evaluation is performed 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.

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



DeepLab: Experimental Set-Up

- VGG-16 and ResNet-101 trained on ImageNet²⁵ dataset are employed.
- Evaluation is performed 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.

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



DeepLab: Experimental Set-Up

- VGG-16 and ResNet-101 trained on ImageNet²⁵ dataset are employed.
- Evaluation is performed 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.

²⁵Krizhevsky, 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 spatial position 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 .

- All positions and labels are equally weighted when loss function is computed.
- Stochastic Gradient Descent method is utilized²⁶.
- CNN output map is subsampled by 8 compared to the original image.

²⁶Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." (2010)



DeepLab: Experimental Set-Up

- Loss function is the sum of cross-entropy terms for each spatial position 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 .

- All positions and labels are equally weighted when loss function is computed.
- Stochastic Gradient Descent method is utilized²⁶.
- CNN output map is subsampled by 8 compared to the original image.



universität bonn

²⁶Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." (2010)

DeepLab: Experimental Set-Up

- Loss function is the sum of cross-entropy terms for each spatial position 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 .

- All positions and labels are equally weighted when loss function is computed.
- Stochastic Gradient Descent method is utilized²⁶.
- CNN output map is subsampled by 8 compared to the original image.



universität bonn

²⁶Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." (2010)

DeepLab: Experimental Set-Up

- Loss function is the sum of cross-entropy terms for each spatial position 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 .

- All positions and labels are equally weighted when loss function is computed.
- Stochastic Gradient Descent method is utilized²⁶.
- CNN output map is subsampled by 8 compared to the original image.



universität bonn

²⁶Bottou, Léon. "Large-scale machine learning with stochastic gradient descent." (2010)

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-101²⁷**: deeper network trained on ImageNet dataset.



universität bonn

²⁷He, Kaiming, et al. "Deep residual learning for image recognition. (2016)"

DeepLab: Experimental Set-Up

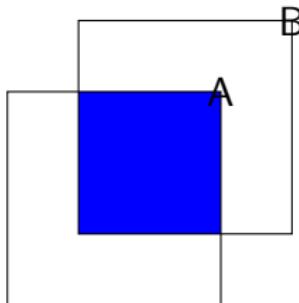


Figure 24: Intersection-over-union(IOU)

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

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



universität bonn

Pascal VOC 2012: Dataset



Figure 25: Pascal VOC 2012 Dataset.

- 20 foreground object classes and one background class.
- 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.



universität bonn

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 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.



universität bonn

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 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.



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 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.



universität bonn

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 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.



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 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.



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 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.



universität bonn

Pascal VOC 2012: Evaluation

LFOV	ASPP-S	ASPP-L	CRF	VAL	TEST	mIOU
✓				✓		▷ 65.76 ◌
✓			✓	✓		69.84
	✓			✓		66.98
	✓		✓	✓		69.73
		✓		✓		68.96
		✓	✓	✓		71.57
	✓	✓			✓	72.60

Table 3: PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Evaluation

LFOV	ASPP-S	ASPP-L	CRF	VAL	TEST	mIOU
✓				✓		65.76
✓			✓	✓		▷ 69.84 ◌
	✓			✓		66.98
	✓		✓	✓		69.73
		✓		✓		68.96
		✓	✓	✓		71.57
	✓	✓			✓	72.60

Table 3: PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Evaluation

LFOV	ASPP-S	ASPP-L	CRF	VAL	TEST	mIOU
✓				✓		65.76
✓			✓	✓		69.84
	✓			✓		▷ 66.98 ◌
	✓		✓	✓		69.73
		✓		✓		68.96
		✓	✓	✓		71.57
	✓	✓			✓	72.60

Table 3: PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



universität bonn

Pascal VOC 2012: Evaluation

LFOV	ASPP-S	ASPP-L	CRF	VAL	TEST	mIOU
✓				✓		65.76
✓			✓	✓		69.84
	✓			✓		66.98
	✓		✓	✓		▷ 69.73 ◁
		✓		✓		68.96
		✓	✓	✓		71.57
	✓	✓			✓	72.60

Table 3: PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Evaluation

LFOV	ASPP-S	ASPP-L	CRF	VAL	TEST	mIOU
✓				✓		65.76
✓			✓	✓		69.84
	✓			✓		66.98
	✓		✓	✓		69.73
		✓		✓		▷ 68.96 ◁
		✓	✓	✓		71.57
	✓	✓			✓	72.60

Table 3: PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Evaluation

LFOV	ASPP-S	ASPP-L	CRF	VAL	TEST	mIOU
✓				✓		65.76
✓			✓	✓		69.84
	✓			✓		66.98
	✓		✓	✓		69.73
		✓		✓		68.96
		✓	✓	✓		▷ 71.57 ◁
	✓	✓		✓		72.60

Table 3: PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Evaluation

LFOV	ASPP-S	ASPP-L	CRF	VAL	TEST	mIOU
✓				✓		65.76
✓			✓	✓		69.84
	✓			✓		66.98
	✓		✓	✓		69.73
		✓		✓		68.96
		✓	✓	✓		71.57
	✓	✓			✓	▷ 72.60 ◁

Table 3: PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Qualitative Results

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

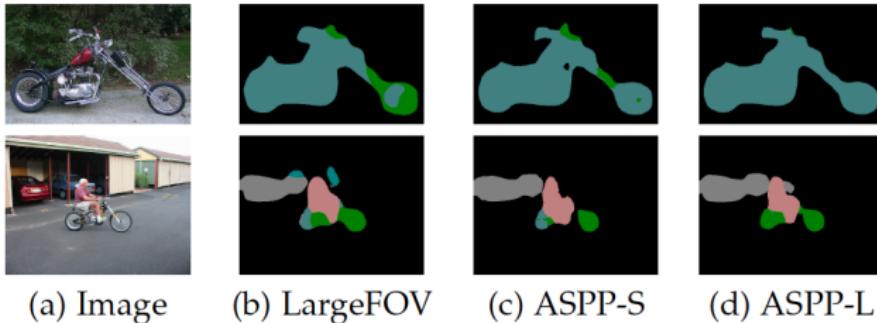


Figure 26: Qualitative segmentation results of ASPP compared to the baseline LargeFOV model. **LargeFOV**: single branch, $r = 12$. **ASPP-S**: four branches, $r = \{2, 4, 8, 12\}$, **ASPP-L**: four branches, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Qualitative Results

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

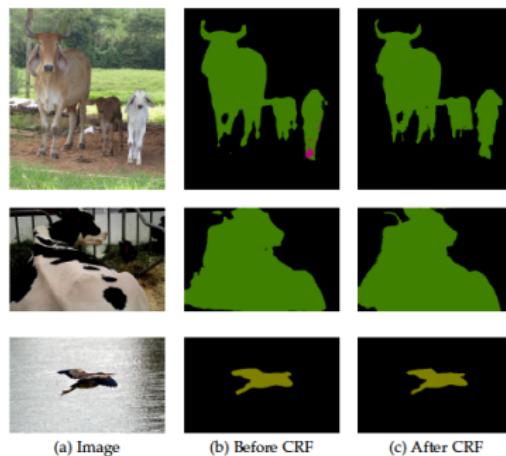


Figure 27: Qualitative segmentation results before/after CRF.

Pascal VOC 2012: Adopting Resnet-101

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

MSC	COCO	Aug	LFOV	ASPP	CRF	VAL	TEST	mIOU
✓						✓		▷ 68.72 ◁
✓	✓					✓		71.27
✓	✓	✓	✓			✓		73.28
✓	✓	✓	✓			✓		74.87
✓	✓	✓	✓			✓		75.54
✓	✓	✓		✓		✓		76.35
✓	✓	✓		✓	✓	✓		77.69
✓	✓	✓		✓	✓	✓	✓	79.70

Table 4: Deeplab results with ResNet-101 on PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Adopting Resnet-101

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

MSC	COCO	Aug	LFOV	ASPP	CRF	VAL	TEST	mIOU
✓						✓		68.72
✓	✓					✓		71.27
✓	✓	✓	✓			✓		73.28
✓	✓	✓	✓			✓		74.87
✓	✓	✓	✓			✓		75.54
✓	✓	✓		✓		✓		76.35
✓	✓	✓		✓	✓	✓		77.69
✓	✓	✓		✓	✓	✓	✓	79.70

Table 4: Deeplab results with ResNet-101 on PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Adopting Resnet-101

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

MSC	COCO	Aug	LFOV	ASPP	CRF	VAL	TEST	mIOU
✓						✓		68.72
✓						✓		71.27
✓	✓					✓		73.28 ▢
✓	✓	✓	✓			✓		74.87
✓	✓	✓	✓			✓		75.54
✓	✓	✓		✓		✓		76.35
✓	✓	✓			✓	✓	✓	77.69
✓	✓	✓			✓	✓	✓	79.70

Table 4: Deeplab results with ResNet-101 on PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Adopting Resnet-101

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

MSC	COCO	Aug	LFOV	ASPP	CRF	VAL	TEST	mIOU
✓						✓		68.72
✓		✓				✓		71.27
✓	✓					✓		73.28
✓	✓	✓	✓			✓		74.87
✓	✓	✓	✓			✓		75.54
✓	✓	✓		✓		✓		76.35
✓	✓	✓			✓	✓	✓	77.69
✓	✓	✓			✓	✓	✓	79.70

Table 4: Deeplab results with ResNet-101 on PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Adopting Resnet-101

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

MSC	COCO	Aug	LFOV	ASPP	CRF	VAL	TEST	mIOU
✓						✓		68.72
✓		✓				✓		71.27
✓	✓					✓		73.28
✓	✓	✓	✓			✓		74.87
✓	✓	✓	✓	✓		✓		75.54
✓	✓	✓		✓		✓		76.35
✓	✓	✓		✓	✓	✓		77.69
✓	✓	✓		✓	✓		✓	79.70

Table 4: Deeplab results with ResNet-101 on PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Adopting Resnet-101

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

MSC	COCO	Aug	LFOV	ASPP	CRF	VAL	TEST	mIOU
✓						✓		68.72
✓		✓				✓		71.27
✓	✓					✓		73.28
✓	✓	✓	✓			✓		74.87
✓	✓	✓	✓			✓		75.54
✓	✓	✓		✓		✓		76.35
✓	✓	✓			✓	✓	✓	77.69
✓	✓	✓			✓	✓	✓	79.70

Table 4: Deeplab results with ResNet-101 on PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Adopting Resnet-101

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

MSC	COCO	Aug	LFOV	ASPP	CRF	VAL	TEST	mIOU
✓						✓		68.72
✓		✓				✓		71.27
✓	✓					✓		73.28
✓	✓	✓	✓			✓		74.87
✓	✓	✓	✓			✓		75.54
✓	✓	✓		✓		✓		76.35
✓	✓	✓		✓	✓	✓	✓	77.69
✓	✓	✓		✓	✓		✓	79.70

Table 4: Deeplab results with ResNet-101 on PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Adopting Resnet-101

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

MSC	COCO	Aug	LFOV	ASPP	CRF	VAL	TEST	mIOU
✓						✓		68.72
✓						✓		71.27
✓	✓					✓		73.28
✓	✓	✓	✓			✓		74.87
✓	✓	✓	✓			✓		75.54
✓	✓	✓		✓		✓		76.35
✓	✓	✓		✓	✓	✓		77.69
✓	✓	✓		✓	✓	✓	✓	▷ 79.70 ▷

Table 4: Deeplab results with ResNet-101 on PASCAL VOC 2012 VAL/TEST 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, $r = \{6, 12, 18, 24\}$.



Pascal VOC 2012: Qualitative Results

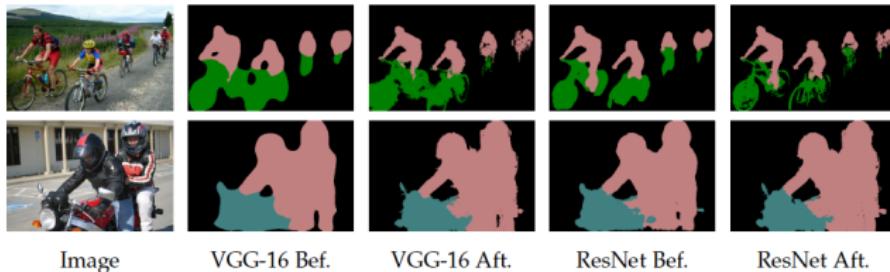


Figure 28: DeepLab results.

- Residual connections in ResNet-101 has similar effect as hyper-column features.



universität bonn

PASCAL-Context: Dataset

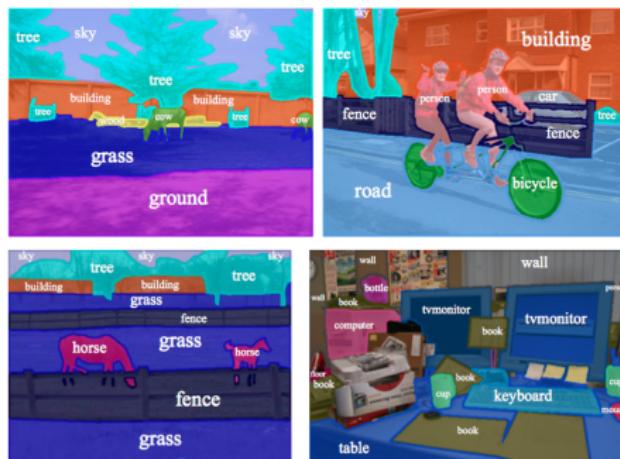


Figure 29: Sample annotations.

- PASCAL-Context²⁸ dataset provides semantic labels(400+) for the whole scene.
- PASCAL-Context is a dataset of additional annotations for PASCAL VOC 2010, including 4998 training, 5105 validation and 9637 testing images.

²⁸Mottaghi, Roozbeh, et al. "The role of context for object detection and semantic segmentation in the wild."

PASCAL-Context: Evaluation

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

Table 5: 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, $r = \{6, 12, 18, 24\}$.



PASCAL-Context: Evaluation

ResNet-101	MSC	COCO	Aug	LargeFOV	ASPP	CRF	mIOU
				✓			37.6
					✓	✓	39.6
✓							39.6
✓		✓		✓			41.4
✓		✓	✓	✓			42.9
✓		✓	✓	✓	✓		43.5
✓		✓	✓	✓		✓	44.7
✓		✓	✓	✓		✓	45.7

Table 5: 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, $r = \{6, 12, 18, 24\}$.



PASCAL-Context: Evaluation

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

Table 5: 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, $r = \{6, 12, 18, 24\}$.



PASCAL-Context: Evaluation

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

Table 5: 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, $r = \{6, 12, 18, 24\}$.



PASCAL-Context: Evaluation

ResNet-101	MSC	COCO	Aug	LargeFOV	ASPP	CRF	mIOU
				✓			37.6
					✓	✓	39.6
✓							39.6
✓		✓		✓			41.4
✓		✓	✓	✓			► 42.9 ◄
✓		✓	✓	✓	✓		43.5
✓		✓	✓	✓		✓	44.7
✓		✓	✓	✓		✓	45.7

Table 5: 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, $r = \{6, 12, 18, 24\}$.



PASCAL-Context: Evaluation

ResNet-101	MSC	COCO	Aug	LargeFOV	ASPP	CRF	mIOU
				✓			37.6
					✓	✓	39.6
✓							39.6
✓		✓		✓			41.4
✓		✓	✓	✓			42.9
✓		✓	✓	✓	✓		► 43.5 ◄
✓		✓	✓	✓		✓	44.7
✓		✓	✓	✓		✓	45.7

Table 5: 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, $r = \{6, 12, 18, 24\}$.



PASCAL-Context: Evaluation

ResNet-101	MSC	COCO	Aug	LargeFOV	ASPP	CRF	mIOU
				✓			37.6
					✓	✓	39.6
✓							39.6
✓		✓		✓			41.4
✓		✓	✓	✓			42.9
✓		✓	✓	✓	✓		43.5
✓		✓	✓	✓		✓	44.7
✓		✓	✓	✓		✓	45.7

Table 5: 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, $r = \{6, 12, 18, 24\}$.



PASCAL-Context: Evaluation

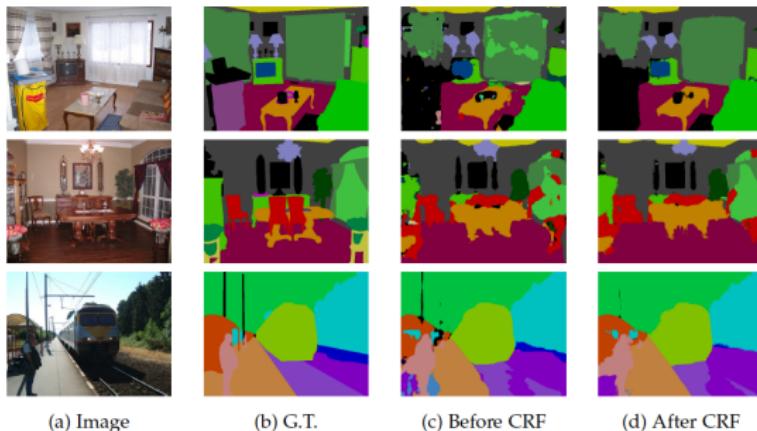
ResNet-101	MSC	COCO	Aug	LargeFOV	ASPP	CRF	mIOU
				✓			37.6
					✓	✓	39.6
✓							39.6
✓		✓		✓			41.4
✓		✓	✓	✓			42.9
✓		✓	✓	✓	✓		43.5
✓		✓	✓	✓		✓	44.7
✓		✓	✓	✓		✓	45.7

Table 5: 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, $r = \{6, 12, 18, 24\}$.



PASCAL-Context: Qualitative/Quantitive Results

- CRF helps improve the model by removing false positives and improving the prediction along object boundaries([Figure 30](#)).
- Final model yields **45.7%**, outperforming the current state-of-art method.



[Figure 30: PASCAL-Context results.](#)

PASCAL-Person-Part: Dataset

- PASCAL-Part-Dataset²⁹ providing segmentation masks for each body part of the object.
- The dataset contains detailed part annotations for every person.
- Annotations are merged to be Head, Torso, Upper/Lower Arms and Upper/Lower Legs.

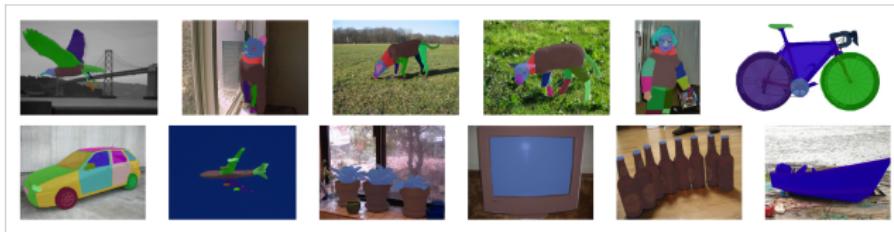


Figure 31: PASCAL-Part-Dataset³⁰.

²⁹Chen, Xianjie, et al. "Detect what you can: Detecting and representing objects using holistic models and body parts. (2014)" 

³⁰http://www.stat.ucla.edu/~xianjie.chen/pascal_part_dataset/pascal_part.html

PASCAL-Person-Part: Evaluation

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

MSC	COCO	Aug	LFOV	ASPP	CRF	mIOU
✓		✓				▷ 58.90 ◁
✓	✓	✓				63.10
✓	✓	✓				64.40
✓	✓	✓			✓	64.94
✓	✓	✓	✓			62.18
✓	✓	✓			✓	62.76

Table 6: DeepLab results with Resnet-101 on PASCAL-Person-Part validation 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, $r = \{6, 12, 18, 24\}$.

PASCAL-Person-Part: Evaluation

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

MSC	COCO	Aug	LFOV	ASPP	CRF	mIOU
✓		✓				58.90
✓	✓	✓				▷ 63.10 ◁
✓	✓	✓			✓	64.40
✓	✓	✓				64.94
✓	✓	✓	✓			62.18
✓	✓	✓		✓		62.76

Table 6: DeepLab results with Resnet-101 on PASCAL-Person-Part validation 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, $r = \{6, 12, 18, 24\}$.

PASCAL-Person-Part: Evaluation

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

MSC	COCO	Aug	LFOV	ASPP	CRF	mIOU
✓		✓				58.90
✓	✓	✓				63.10
✓	✓	✓				▷ 64.40 ◁
✓	✓	✓		✓		64.94
✓	✓	✓	✓			62.18
✓	✓	✓		✓		62.76

Table 6: DeepLab results with Resnet-101 on PASCAL-Person-Part validation 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, $r = \{6, 12, 18, 24\}$.

PASCAL-Person-Part: Evaluation

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

MSC	COCO	Aug	LFOV	ASPP	CRF	mIOU
✓		✓				58.90
✓	✓	✓				63.10
✓	✓	✓				64.40
✓	✓	✓			✓	▷ 64.94 ◌
✓	✓	✓	✓			62.18
✓	✓	✓		✓		62.76

Table 6: DeepLab results with Resnet-101 on PASCAL-Person-Part validation 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, $r = \{6, 12, 18, 24\}$.

PASCAL-Person-Part: Evaluation

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

MSC	COCO	Aug	LFOV	ASPP	CRF	mIOU
✓		✓				58.90
✓	✓	✓				63.10
✓	✓	✓				64.40
✓	✓	✓			✓	64.94
✓	✓	✓	✓			▷ 62.18 ◌
✓	✓	✓		✓		62.76

Table 6: DeepLab results with Resnet-101 on PASCAL-Person-Part validation 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, $r = \{6, 12, 18, 24\}$.

PASCAL-Person-Part: Evaluation

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

MSC	COCO	Aug	LFOV	ASPP	CRF	mIOU
✓		✓				58.90
✓	✓	✓				63.10
✓	✓	✓				64.40
✓	✓	✓			✓	64.94
✓	✓	✓	✓			62.18
✓	✓	✓				▷ 62.76 ◁

Table 6: DeepLab results with Resnet-101 on PASCAL-Person-Part validation 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, $r = \{6, 12, 18, 24\}$.

PASCAL-Person-Part: Qualitative Results

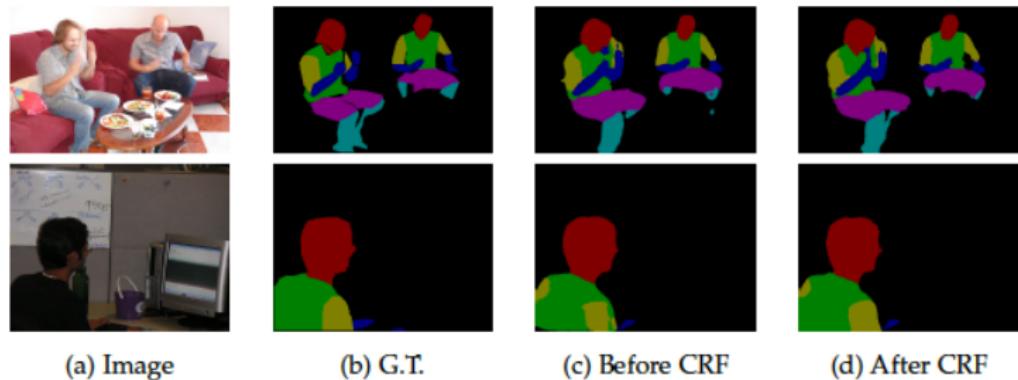


Figure 32: Qualitative results on PASCAL-Person-Part dataset.



universität bonn

Cityscapes: Dataset



Figure 33: Cityscapes dataset³¹.

- 5000 images collected in street scenes from 50 different cities³¹.
- 19 semantic labels (belonging to 7 super categories: ground, construction, object, nature, sky, human, and vehicle).
- 2975, 500, and 1525 training, validation, and test images respectively.

³¹MLA Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding. (2016)"

Cityscapes: Evaluation

- The images of Cityscapes have resolution 2048×1024 .
- Model trained with subsampled images or each image is split into overlapped regions(**full**).
- DeepLab-CRF(ResNet-101) ranked as the second-best model with 70.4% mIOU on the *test* set.

Full	Aug	LargeFOV	ASPP	CRF	mIOU
✓					▷ 66.6 ◌
	✓	✓			69.2
			✓		70.4
✓	✓		✓		71.0
✓	✓		✓	✓	71.4

Table 7: DeepLab results with Resnet-101 on Cityscapes validation set.

Full: Model trained with full resolution images. **Aug:** Randomly Scaled

Input, **LargeFOV**: single branch, **ASPP**: four branches,

$$r = \{6, 12, 18, 24\}.$$



Cityscapes: Evaluation

- The images of Cityscapes have resolution 2048×1024 .
- Model trained with subsampled images or each image is split into overlapped regions(**full**).
- DeepLab-CRF(ResNet-101) ranked as the second-best model with 70.4% mIOU on the *test* set.

Full	Aug	LargeFOV	ASPP	CRF	mIOU
✓					66.6
	✓		✓		▷ 69.2 ◌
✓				✓	70.4
✓	✓			✓	71.0
✓	✓		✓	✓	71.4

Table 7: DeepLab results with Resnet-101 on Cityscapes validation set.

Full: Model trained with full resolution images. **Aug:** Randomly Scaled

Input, **LargeFOV**: single branch, **ASPP**: four branches,

$$r = \{6, 12, 18, 24\}.$$



Cityscapes: Evaluation

- The images of Cityscapes have resolution 2048×1024 .
- Model trained with subsampled images or each image is split into overlapped regions(**full**).
- DeepLab-CRF(ResNet-101) ranked as the second-best model with 70.4% mIOU on the *test* set.

Full	Aug	LargeFOV	ASPP	CRF	mIOU
✓					66.6
✓		✓			69.2
✓			✓		▷ 70.4 ◁
✓	✓		✓		71.0
✓	✓		✓	✓	71.4

Table 7: DeepLab results with Resnet-101 on Cityscapes validation set.

Full: Model trained with full resolution images. **Aug:** Randomly Scaled

Input, **LargeFOV**: single branch, **ASPP**: four branches,

$$r = \{6, 12, 18, 24\}.$$



Cityscapes: Evaluation

- The images of Cityscapes have resolution 2048×1024 .
- Model trained with subsampled images or each image is split into overlapped regions(**full**).
- DeepLab-CRF(ResNet-101) ranked as the second-best model with 70.4% mIOU on the *test* set.

Full	Aug	LargeFOV	ASPP	CRF	mIOU
✓					66.6
✓		✓			69.2
✓			✓		70.4
✓	✓		✓		▷ 71.0 ◷
✓	✓		✓	✓	71.4

Table 7: DeepLab results with Resnet-101 on Cityscapes validation set.

Full: Model trained with full resolution images. **Aug:** Randomly Scaled

Input, **LargeFOV**: single branch, **ASPP**: four branches,

$$r = \{6, 12, 18, 24\}.$$



Cityscapes: Evaluation

- The images of Cityscapes have resolution 2048×1024 .
- Model trained with subsampled images or each image is split into overlapped regions(**full**).
- DeepLab-CRF(ResNet-101) ranked as the second-best model with 70.4% mIOU on the *test* set.

Full	Aug	LargeFOV	ASPP	CRF	mIOU
✓					66.6
✓		✓			69.2
✓			✓		70.4
✓	✓		✓		71.0
✓	✓		✓	✓	▷ 71.4 ▷

Table 7: DeepLab results with Resnet-101 on Cityscapes validation set.

Full: Model trained with full resolution images. **Aug:** Randomly Scaled

Input, **LargeFOV**: single branch, **ASPP**: four branches,

$$r = \{6, 12, 18, 24\}.$$



Cityscapes: Qualitative Results

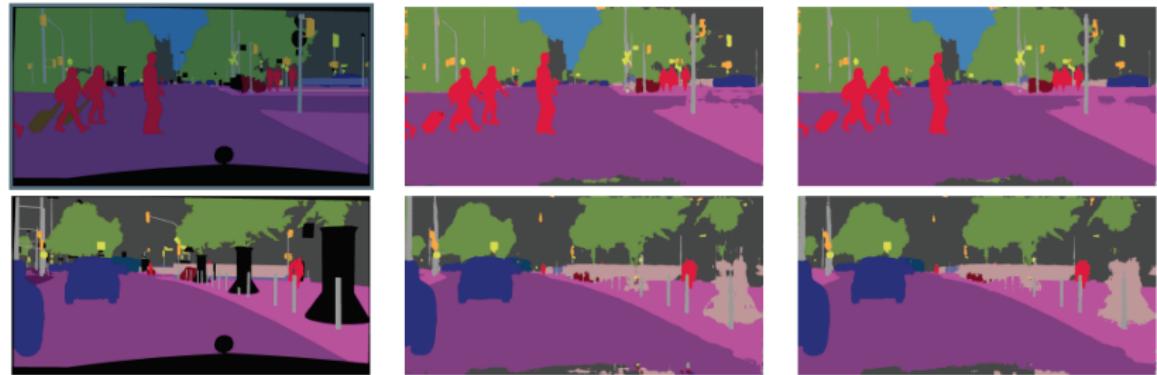


Figure 34: Qualitative results on Cityscapes dataset.



universität bonn

Future Direction

- Weaker supervision³² 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.

³²Papandreou, George, et al. "Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. (2015)"

Future Direction

- Weaker supervision³² 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.

³²Papandreou, George, et al. "Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. (2015)"

Future Direction

- Weaker supervision³² 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.

³²Papandreou, George, et al. "Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. (2015)"



Appendix: Mean Field Approximation

- Key idea³³: 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}

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

while not converged **do**

³³Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

Appendix: Mean Field Approximation

- Key idea³³: 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

³³Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."



Appendix: Mean Field Approximation

- Key idea³³: 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} $\triangleright \mathcal{Q}_i(x_i) \leftarrow \frac{1}{Z_i} \exp\{-\theta_i(x_i)\}$

while not converged **do**

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

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

³³Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

Appendix: Mean Field Approximation

- Key idea³³: 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**

- $\tilde{\mathcal{Q}}_i^{(m)}(I) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) \mathcal{Q}_j(I)$ for all m ▷ **MP** from all X_j to all X_i
- $\hat{\mathcal{Q}}_i(x_i) \leftarrow \sum_{I \in \mathcal{L}} \mu^{(m)}(x_i, I) \sum_m w^{(m)} \tilde{\mathcal{Q}}_i^{(m)}$ ▷ **Compatibility transform**
- $\mathcal{Q}_i(x_i) \leftarrow \exp\{-\theta_i(x_i) - \hat{\mathcal{Q}}_i(x_i)\}$ ▷ **Local Update**

³³Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

Appendix: Mean Field Approximation

- Key idea³³: 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} $\triangleright \mathcal{Q}_i(x_i) \leftarrow \frac{1}{Z_i} \exp\{-\theta_i(x_i)\}$

while not converged **do**

- $\tilde{\mathcal{Q}}_i^{(m)}(I) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) \mathcal{Q}_j(I)$ for all m $\triangleright \mathbf{MP}$ from all X_j to all X_i
- $\hat{\mathcal{Q}}_i(x_i) \leftarrow \sum_{I \in \mathcal{L}} \mu^{(m)}(x_i, I) \sum_m w^{(m)} \tilde{\mathcal{Q}}_i^{(m)}$ $\triangleright \mathbf{Compatibility transform}$
- $\mathcal{Q}_i(x_i) \leftarrow \exp\{-\theta_i(x_i) - \hat{\mathcal{Q}}_i(x_i)\}$ $\triangleright \mathbf{Local Update}$

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

end while

³³Krähenbühl et al. "Efficient inference in fully connected CRFs with gaussian edge potentials."

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 theorem³⁴, function $Q(I)$ 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 $O(N)$.



universität bonn

³⁴S. W. Smith. The scientist and engineer's guide to digital signal processing.