Dense Predictions Using Dilated Convolutions

Najmus Ibrahim

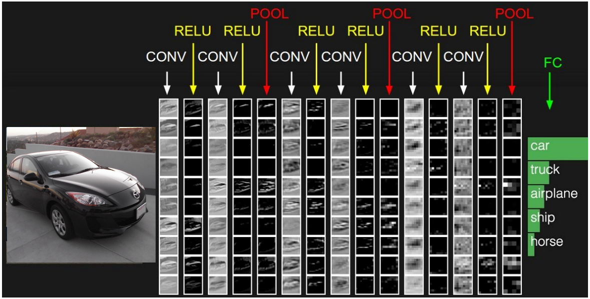
University of Toronto
Institute for Aerospace Studies

January 2018

Introduction

Layers in CNNs for image classification have various modules that control the output volume of subsequent layers (Image Credit: Stanford C321n):

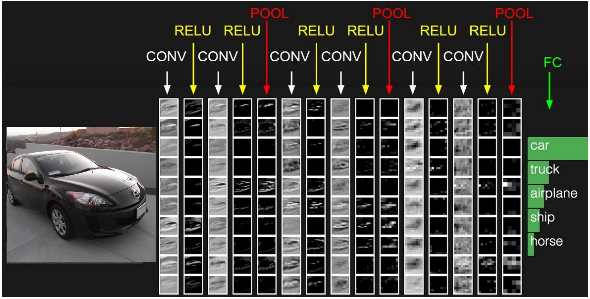
- Convolution Layers
  - Filter Size
  - Stride
  - Padding
- Pooling Layers
- Activation Layers
- FC Layers



Introduction

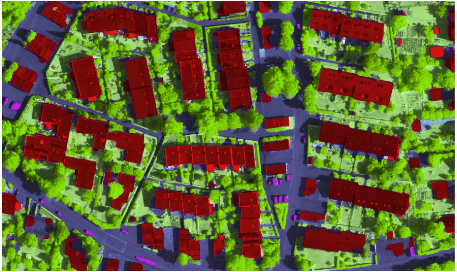
Layers in CNNs for image classification have various modules that control the output volume of subsequent layers (Image Credit: Stanford C321n):

- Convolution Layers
  - Filter Size
  - Stride
  - Padding
- Pooling Layers
- Activation Layers
- FC Layers



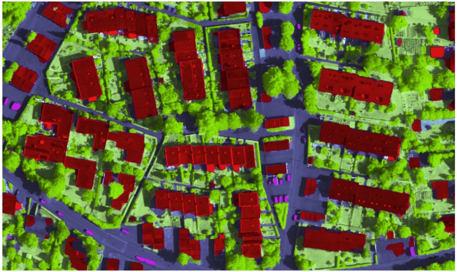
Conventional modules (e.g., pooling/stride) reduce network resolution/coverage between layers and make it challenging to carry out applications that require dense predictions.

- Semantic segmentation: multi-scale contextual reasoning with full-resolution output



Semantic Segmentation of Satellite Imagery (Image Credit: ETH Zurich)

- Semantic segmentation: multi-scale contextual reasoning with full-resolution output



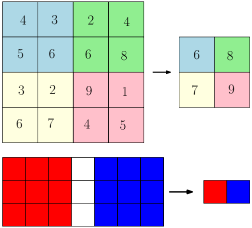
Semantic Segmentation of Satellite Imagery (Image Credit: ETH Zurich)

- Many state-of-the-art models for dense predictions are based on adaptations of CNNs for image classification
- Not all of aspects of image classification are useful for this application

Resolution vs. Coverage

- Resolution: image pixel density
- Pooling: loss of resolution

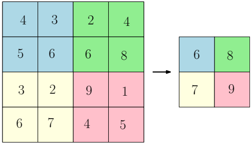
- Coverage: Overlap between adjacent feature maps
- Large Stride: loss of coverage



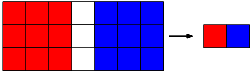
- Recover resolution loss: upsample
- Compensate for coverage loss: use smaller stride

Resolution vs. Coverage

- Resolution: image pixel density
- Pooling: loss of resolution



- Coverage: Overlap between adjacent feature maps
- Large Stride: loss of coverage



- Recover resolution loss: upsample
- Compensate for coverage loss: use smaller stride
- Both increase number of layers/parameters and computation/memory

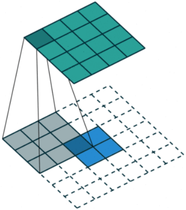
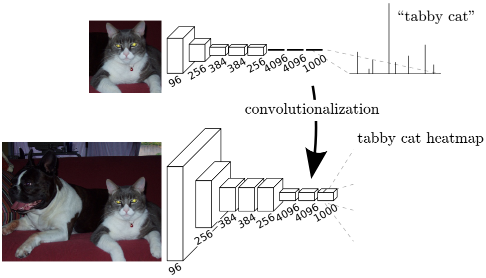


Image Credit:
github.com/vdumoulin/conv_arithmetic/tree/master/gif

Fully Convolutional Network (FCN)

- Conventional semantic segmentation network that uses pooling, stride, upsampling
- Derived from classification architectures that take fixed-size inputs and produce non-spatial outputs
- FC layers considered as convolutions with kernels acting on the entire input region

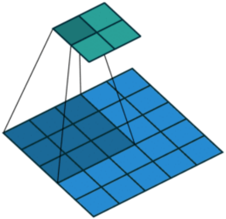


Fully Convolutional Network (Long et al. (2015))

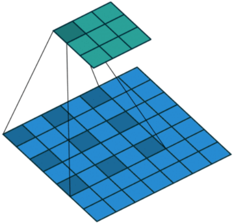
- In-network upsampling and addtional layers to FC output allow pixelwise prediction

Dilated Convolutions

- High resolution operations throughout the network facilitated by dilated convolution
- Sparse filters formed by skipping pixels at regular intervals



(a) 2-Stride

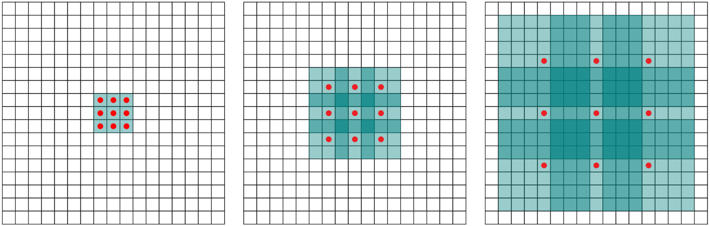


(b) 2-Dilated

- Convention (dark blue squares = non-zero):
  - n-Dilated: $n - 1$ pixels skipped
  - 1-Dilated: 0 pixels skipped
  - 2-Dilated: 1 pixels skipped
  - 4-Dilated: 3 pixels skipped
- 2-Dilated 3×3 Filter = 5×5 Filter (9 non-zero weights)

Dilated Convolutions

- F. Yu, V. Koltun, "Multi-Scale Context Aggregation By Dilated Convolutions"
- Receptive field of an element $\mathbf{x}$ in layer $k+1$ is the set of elements in layer k that influence it



(a) (b) (c)

Consecutive 1-Dilated (left), 2-Dilated (middle), 4-Dilated (right) 3×3 Convolution

- Resulting receptive field of 2^i-Dilated feature map is size $(2^{i+2}-1)^2$
- Receptive field grows exponentially while number of parameters is constant

Multi-Scale Context Aggregation Context Module

- Context module (7 layers) with progressively increasing receptive field without losing resolution
- Has same form of input/output: takes C feature maps in and produces C feature maps out

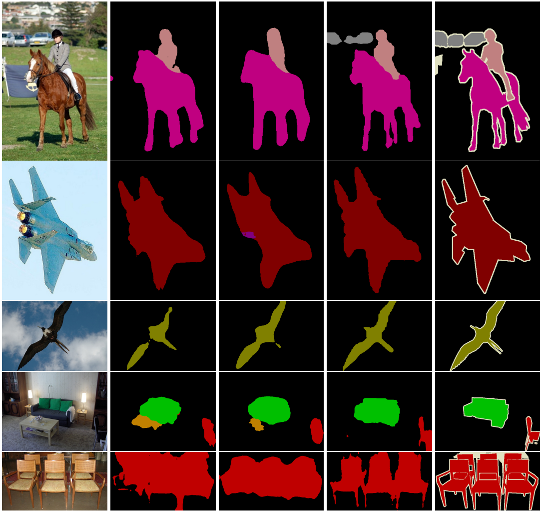
Layer	1	2	3	4	5	6	7	8
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	1	1
Truncation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Receptive field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	67×67
Output channels								
Basic	C	C	C	C	C	C	C	C
Large	$2C$	$2C$	$4C$	$8C$	$16C$	$32C$	$32C$	C

Context Module Using Multi-Layered Dilated Convolutions

- Module can be combined readily with existing dense prediction architectures

Front-End Module

- Simplified image classification CNNs (Simonyan & Zisserman (2015)) by removing layers that are counterproductive for dense prediction
  - Final pooling and striding layers
  - Padding in intermediate feature maps
- Inputs are padded images and outputs are $C = 21$ feature maps at 64×64 resolution
- Training (VOC-2012)
  - Iterations $(n) = 60K$
  - Mini-batch size (p): 14
  - Learning rate (α): 10^{-3}
  - Momentum (β): 0.9
- Test accuracy comparison vs. FCN-8s and DeepLab+



(a) Image (b) FCN-8s (c) DeepLab (d) Our front end (e) Ground truth

Experimentation Results

- Front-end module is both simpler and +5% (mean IoU) more accurate

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	50.8	72.3	42.1	67.9	52.6	62.1
DeepLab-Msc	74.9	34.1	72.6	52.9	61.0	77.9	73.0	73.7	26.4	62.2	49.3	68.4	64.1	74.0	75.0	51.7	72.7	42.5	67.2	55.7	62.9
Our front end	**82.2**	**37.4**	**72.7**	**57.1**	**62.7**	**82.8**	**77.8**	**78.9**	**28**	**70**	**51.6**	**73.1**	**72.8**	**81.5**	**79.1**	**56.6**	**77.1**	**49.9**	**75.3**	**60.9**	**67.6**

VOC-2012 Test Set Accuracy

Experimentation Results

- Front-end module is both simpler and +5% (mean IoU) more accurate

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	50.8	72.3	42.1	67.9	52.6	62.1
DeepLab-Msc	74.9	34.1	72.6	52.9	61.0	77.9	73.0	73.7	26.4	62.2	49.3	68.4	64.1	74.0	75.0	51.7	72.7	42.5	67.2	55.7	62.9
Our front end	**82.2**	**37.4**	**72.7**	**57.1**	**62.7**	**82.8**	**77.8**	**78.9**	**28**	**70**	**51.6**	**73.1**	**72.8**	**81.5**	**79.1**	**56.6**	**77.1**	**49.9**	**75.3**	**60.9**	**67.6**

VOC-2012 Test Set Accuracy

- In anticipation of comparison with high performing systems, two-stage testing done on the front-end module
  - Coarse Tuning: VOC-2012, Microsoft COCO
    - n = 100K, $\alpha = 10^{-3}$
    - n = 40K, $\alpha = 10^{-4}$
  - Fine Tuning: VOC-2012 only
    - n = 50K, $\alpha = 10^{-5}$

Experimentation Results

- Front-end module is both simpler and +5% (mean IoU) more accurate

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	50.8	72.3	42.1	67.9	52.6	62.1
DeepLab-Msc	74.9	34.1	72.6	52.9	61.0	77.9	73.0	73.7	26.4	62.2	49.3	68.4	64.1	74.0	75.0	51.7	72.7	42.5	67.2	55.7	62.9
Our front end	**82.2**	**37.4**	**72.7**	**57.1**	**62.7**	**82.8**	**77.8**	**78.9**	**28**	**70**	**51.6**	**73.1**	**72.8**	**81.5**	**79.1**	**56.6**	**77.1**	**49.9**	**75.3**	**60.9**	**67.6**

VOC-2012 Test Set Accuracy

- In anticipation of comparison with high performing systems, two-stage testing done on the front-end module
  - Coarse Tuning: VOC-2012, Microsoft COCO
    - n = 100K, $\alpha = 10^{-3}$
    - n = 40K, $\alpha = 10^{-4}$
  - Fine Tuning: VOC-2012 only
    - n = 50K, $\alpha = 10^{-5}$
- Mean IoU accuracy of front-end on VOC-2012
  - Test: 71.3%
  - Validation: 69.8%

Experimentation Results

- Front-end module is both simpler and +5% (mean IoU) more accurate

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	50.8	72.3	42.1	67.9	52.6	62.1
DeepLab-Msc	74.9	34.1	72.6	52.9	61.0	77.9	73.0	73.7	26.4	62.2	49.3	68.4	64.1	74.0	75.0	51.7	72.7	42.5	67.2	55.7	62.9
Our front end	**82.2**	**37.4**	**72.7**	**57.1**	**62.7**	**82.8**	**77.8**	**78.9**	**28**	**70**	**51.6**	**73.1**	**72.8**	**81.5**	**79.1**	**56.6**	**77.1**	**49.9**	**75.3**	**60.9**	**67.6**

VOC-2012 Test Set Accuracy

- In anticipation of comparison with high performing systems, two-stage testing done on the front-end module
  - Coarse Tuning: VOC-2012, Microsoft COCO
    - n = 100K, $\alpha = 10^{-3}$
    - n = 40K, $\alpha = 10^{-4}$
  - Fine Tuning: VOC-2012 only
    - n = 50K, $\alpha = 10^{-5}$
- Mean IoU accuracy of front-end on VOC-2012
  - Test: 71.3%
  - Validation: 69.8%
- Controlled experiments performed by inserting Context Module after front-end

Experimentation Results

- Context modules (Basic and Large) added to front-end and then to two different semantic segmentation architectures
  1. CRF (Chen et al. (2015))
  2. CRF-RNN (Zheng et al. (2015))

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
Front end	86.3	38.2	76.8	**66.8**	63.2	87.3	78.7	82	33.7	76.7	53.5	73.7	76	76.6	83	**51.9**	77.8	44	79.9	**66.3**	69.8
Front + Basic	86.4	37.6	78.5	66.3	64.1	89.9	79.9	84.9	**36.1**	79.4	**55.8**	77.6	81.6	79	83.1	51.2	81.3	43.7	82.3	65.7	71.3
Front + Large	**87.3**	**39.2**	**80.3**	65.6	**66.4**	**90.2**	**82.6**	**85.8**	34.8	**81.9**	51.7	**79**	**84.1**	**80.9**	**83.2**	51.2	**83.2**	**44.7**	**83.4**	65.6	**72.1**
Front end + CRF	89.2	38.8	80	**69.8**	63.2	88.8	80	85.2	33.8	80.6	55.5	77.1	80.8	77.3	84.3	**53.1**	80.4	45	80.7	**67.9**	71.6
Front + Basic + CRF	89.1	38.7	81.4	67.4	65	91	81	86.7	**37.5**	81	**57**	79.6	83.6	79.9	**84.6**	52.7	83.3	44.3	82.6	67.2	72.7
Front + Large + CRF	**89.6**	**39.9**	**82.7**	66.7	**67.5**	**91.1**	**83.3**	**87.4**	36	**83.3**	52.5	**80.7**	**85.7**	**81.8**	84.4	52.6	**84.4**	**45.3**	**83.7**	66.7	**73.3**
Front end + RNN	88.8	38.1	80.8	**69.1**	65.6	89.9	79.6	85.7	36.3	83.6	57.3	77.9	83.2	77	**84.6**	**54.7**	82.1	**46.9**	80.9	66.7	72.5
Front + Basic + RNN	89	38.4	82.3	67.9	65.2	91.5	80.4	87.2	**38.4**	82.1	**57.7**	79.9	85	79.6	84.5	53.5	84	45	82.8	66.2	73.1
Front + Large + RNN	**89.3**	**39.2**	**83.6**	67.2	**69**	**92.1**	**83.1**	**88**	**38.4**	**84.8**	55.3	**81.2**	**86.7**	**81.3**	84.3	53.6	**84.4**	45.8	**83.8**	**67**	**73.9**

VOC-2012 Validation Set Accuracy

Experimentation Results

- Context modules (Basic and Large) added to front-end and then to two different semantic segmentation architectures
  1. CRF (Chen et al. (2015))
  2. CRF-RNN (Zheng et al. (2015))

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
Front end	86.3	38.2	76.8	**66.8**	63.2	87.3	78.7	82	33.7	76.7	53.5	73.7	76	76.6	83	**51.9**	77.8	44	79.9	**66.3**	69.8
Front + Basic	86.4	37.6	78.5	66.3	64.1	89.9	79.9	84.9	**36.1**	79.4	**55.8**	77.6	81.6	79	83.1	51.2	81.3	43.7	82.3	65.7	71.3
Front + Large	**87.3**	**39.2**	**80.3**	65.6	**66.4**	**90.2**	**82.6**	**85.8**	34.8	**81.9**	51.7	**79**	**84.1**	**80.9**	**83.2**	51.2	**83.2**	**44.7**	**83.4**	65.6	**72.1**
Front end + CRF	89.2	38.8	80	**69.8**	63.2	88.8	80	85.2	33.8	80.6	55.5	77.1	80.8	77.3	84.3	**53.1**	80.4	45	80.7	**67.9**	71.6
Front + Basic + CRF	89.1	38.7	81.4	67.4	65	91	81	86.7	**37.5**	81	**57**	79.6	83.6	79.9	**84.6**	52.7	83.3	44.3	82.6	67.2	72.7
Front + Large + CRF	**89.6**	**39.9**	**82.7**	66.7	**67.5**	**91.1**	**83.3**	**87.4**	36	**83.3**	52.5	**80.7**	**85.7**	**81.8**	84.4	52.6	**84.4**	**45.3**	**83.7**	66.7	**73.3**
Front end + RNN	88.8	38.1	80.8	**69.1**	65.6	89.9	79.6	85.7	36.3	83.6	57.3	77.9	83.2	77	**84.6**	**54.7**	82.1	**46.9**	80.9	66.7	72.5
Front + Basic + RNN	89	38.4	82.3	67.9	65.2	91.5	80.4	87.2	**38.4**	82.1	**57.7**	79.9	85	79.6	84.5	53.5	84	45	82.8	66.2	73.1
Front + Large + RNN	**89.3**	**39.2**	**83.6**	67.2	**69**	**92.1**	**83.1**	**88**	**38.4**	**84.8**	55.3	**81.2**	**86.7**	**81.3**	84.3	53.6	**84.4**	45.8	**83.8**	**67**	**73.9**

VOC-2012 Validation Set Accuracy

- Addition of Context Module improves accuracy by +0.6% (mean IoU) in all three architectures

Experimentation Results

- Context module (Large) and front-end module compared against other high performing systems
  1. DeepLab variants (Long et al. (2015))
  2. CRF-RNN (Zheng et al. (2015))
  3. Front-end/Context module combinations with CRF-RNN

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
DeepLab++	89.1	38.3	88.1	63.3	69.7	87.1	**83.1**	85	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
DeepLab-MSc++	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	32.6	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	58.3	76.2	67.2	73.9
CRF-RNN	90.4	**55.3**	88.7	**68.4**	69.8	88.3	82.4	85.1	32.6	78.5	**64.4**	79.6	81.9	**86.4**	81.8	**58.6**	82.4	53.5	77.4	**70.1**	74.7
Front end	86.6	37.3	84.9	62.4	67.3	86.2	81.2	82.1	32.6	77.4	58.3	75.9	81	83.6	82.3	54.2	81.5	50.1	77.5	63	71.3
Context	89.1	39.1	86.8	62.6	68.9	88.2	82.6	87.7	33.8	81.2	59.2	81.8	87.2	83.3	83.6	53.6	84.9	53.7	80.5	62.9	73.5
Context + CRF	91.3	39.9	**88.9**	64.3	69.8	88.9	82.6	89.7	34.7	82.7	59.5	83	88.4	84.2	85	55.3	86.7	54.4	**81.9**	63.6	74.7
Context + CRF-RNN	**91.7**	39.6	87.8	63.1	**71.8**	**89.7**	82.9	**89.8**	**37.2**	**84**	63	**83.3**	**89**	83.8	**85.1**	56.8	**87.6**	**56**	80.2	64.7	**75.3**

VOC-2012 Test Set Accuracy

Experimentation Results

- Context module (Large) and front-end module compared against other high performing systems
  1. DeepLab variants (Long et al. (2015))
  2. CRF-RNN (Zheng et al. (2015))
  3. Front-end/Context module combinations with CRF-RNN

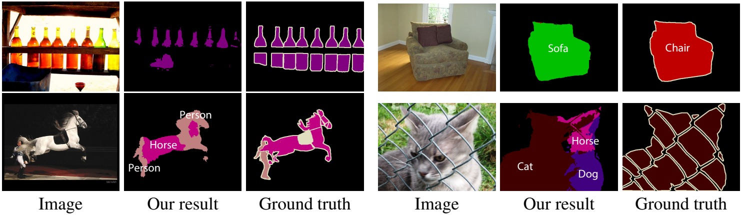
	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
DeepLab++	89.1	38.3	88.1	63.3	69.7	87.1	**83.1**	85	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
DeepLab-MSc++	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	32.6	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	58.3	76.2	67.2	73.9
CRF-RNN	90.4	**55.3**	88.7	**68.4**	69.8	88.3	82.4	85.1	32.6	78.5	**64.4**	79.6	81.9	**86.4**	81.8	**58.6**	82.4	53.5	77.4	**70.1**	74.7
Front end	86.6	37.3	84.9	62.4	67.3	86.2	81.2	82.1	32.6	77.4	58.3	75.9	81	83.6	82.3	54.2	81.5	50.1	77.5	63	71.3
Context	89.1	39.1	86.8	62.6	68.9	88.2	82.6	87.7	33.8	81.2	59.2	81.8	87.2	83.3	83.6	53.6	84.9	53.7	80.5	62.9	73.5
Context + CRF	91.3	39.9	**88.9**	64.3	69.8	88.9	82.6	89.7	34.7	82.7	59.5	83	88.4	84.2	85	55.3	86.7	54.4	**81.9**	63.6	74.7
Context + CRF-RNN	**91.7**	39.6	87.8	63.1	**71.8**	**89.7**	82.9	**89.8**	**37.2**	**84**	63	**83.3**	**89**	83.8	**85.1**	56.8	**87.6**	**56**	80.2	64.7	**75.3**

VOC-2012 Test Set Accuracy

- Context module has +2.2% mean IoU accuracy compared to front end alone
- Context module alone outperforms DeepLab++
- Context module with dense CRF performs on par with CRF-RNN
- Context module combined with CRF-RNN outperforms CRF-RNN by 0.6%

Future Work

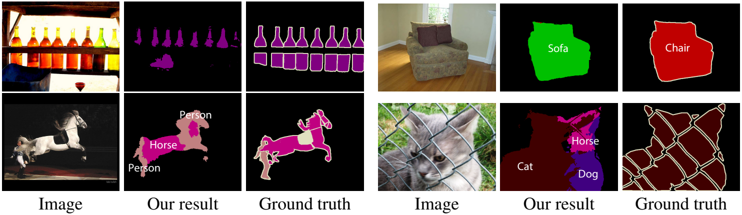
- Accuracies and failure cases leave significant room for future advances



Failure Cases

Future Work

- Accuracies and failure cases leave significant room for future advances

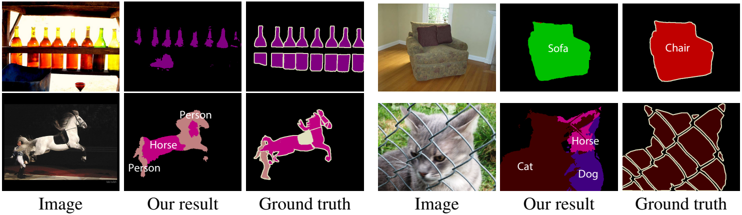


Failure Cases

- Promising results observed for:
  - Dedicated dense prediction architectures without image classification artifacts

Future Work

- Accuracies and failure cases leave significant room for future advances

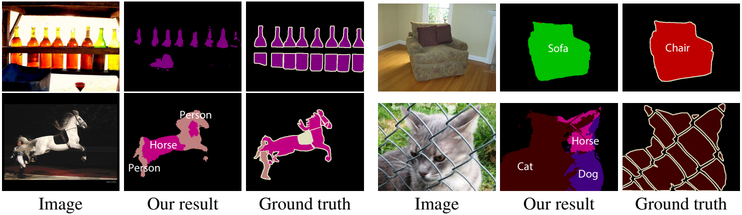


Failure Cases

- Promising results observed for:
  - Dedicated dense prediction architectures without image classification artifacts
  - Removing pre-training by leveraging dilation convolutions and performing end-to-end dense prediction

Future Work

- Accuracies and failure cases leave significant room for future advances



Failure Cases

- Promising results observed for:
  - Dedicated dense prediction architectures without image classification artifacts
  - Removing pre-training by leveraging dilation convolutions and performing end-to-end dense prediction
  - Simplifying and unifying architectures to take inputs and produce outputs at full resolution

Conclusions

- Simplification of adapted image classification systems for semantic segmentation can improve accuracy
- Dilated convolutions support exponential expansion of the receptive field without loss of resolution or coverage
- CNN module with dilated convolutions systematically aggregate multi-scale contextual information without resolution loss
- Context Module increases the accuracy of current state-of-the-art semantic segmentation architectures

For more information:

1. F. Yu, V. Koltun, "Multi-Scale Context Aggregation By Dilated Convolutions", ICLR, 2016
2. J. Long, E. Shelhamer, T. Darrell, "Fully Convolutional Network for Semantic Segmentation", CPVR, 2015
3. S. Zheng et al., "Conditional Random Fields as Recurrent Neural Networks", ICCV, 2015

Thank you.