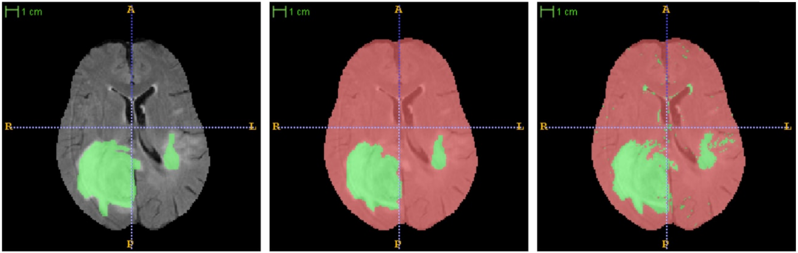
Segmentation

Andrei Nicolicioiu
andrei.nicolicioiu@gmail.com

Segmentation



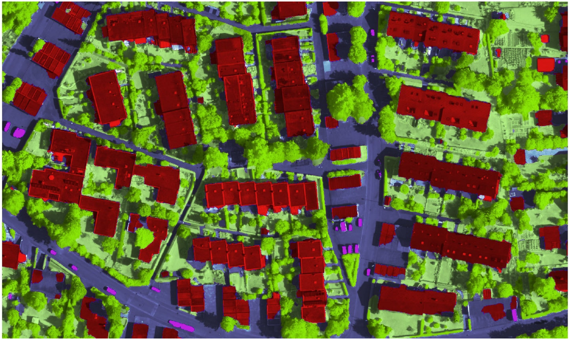


Image sources: Bauer et al. [2011], Marmanis et al. [2016]

Segmentation: definition

- partition an image into regions each of which has a reasonably homogeneous visual appearance or which corresponds to objects or parts of an object [Forsyth and Ponce 2003]

Segmentation: definition

- partition an image into regions each of which has a reasonably homogeneous visual appearance or which corresponds to objects or parts of an object [Forsyth and Ponce 2003]
- **Q:** Semantic Segmentation vs Instance Segmentation?

Segmentation: definition

- partition an image into regions each of which has a reasonably homogeneous visual appearance or which corresponds to objects or parts of an object [Forsyth and Ponce 2003]
- **Q:** Semantic Segmentation vs Instance Segmentation?
  - *Semantic segmentation*: pixels of a certain class
  - *Instance segmentation*: pixels of each individual instance separately

Segmentation: definition

- **Semantic segmentation**

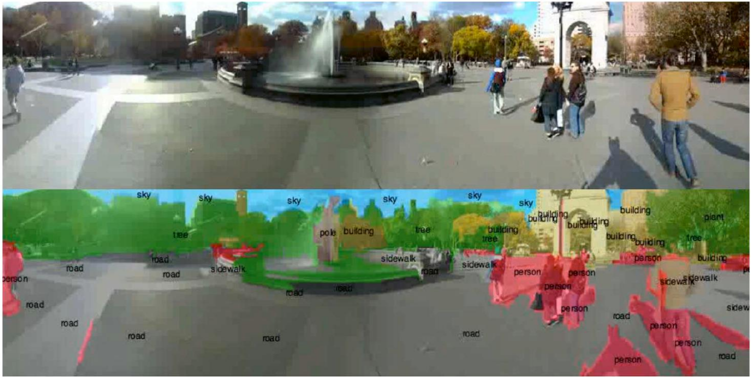


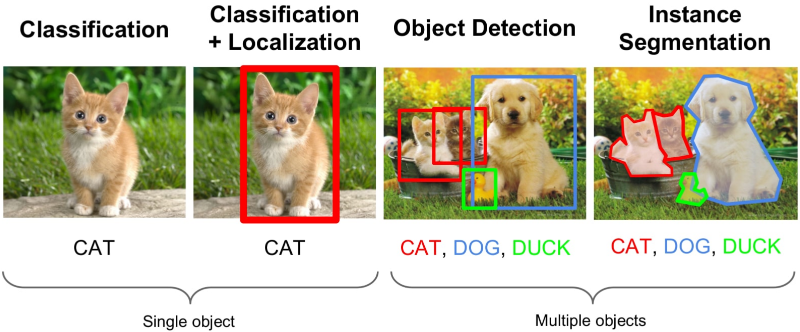
Image source: Farabet et al. [2013]

Segmentation: definition

- **Instance segmentation**

Image source: Pinheiro et al. [2016]

CV tasks


Classification
Classification + Localization
Object Detection
Instance Segmentation
CAT
CAT
CAT, DOG, DUCK
CAT, DOG, DUCK
Single object
Multiple objects

Segmentation

- Carreira et al. [2012] - semantic segmentation using region proposals
  - use classic methods for generating region proposals
  - feature extraction using image descriptors
  - assign each region a score for each class using a SVM based on these features
  - estimate a semantic segmentation by pasting the top scoring masks

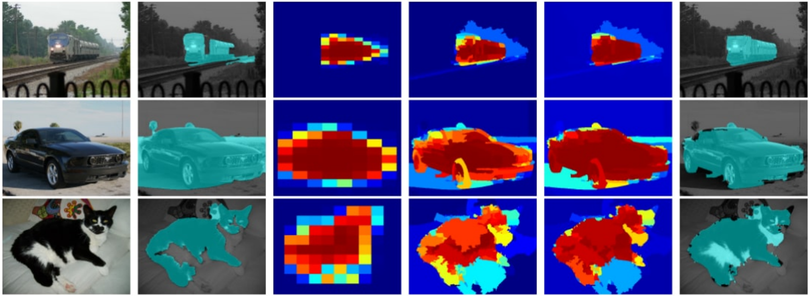
Segmentation

- Hariharan et al. [2014] - segmentation using region proposals
  - use classic methods for generating region proposals: Arbeláez et al. [2014]
  - feature extraction
    - use two CNNs initialised from AlexNet
    - first network is trained on the bounding box around the proposal
    - second network is trained on the bounding box around the proposal with the background masked out
    - the two CNNs are concatenated and fine-tunned jointly
  - assign each region a score for each class using a SVM based on these features

Segmentation

- Hariharan et al. [2014] - segmentation using region proposals
  - Refine the proposed regions by
    - predict **coarse masks** using multiple logistic regressions units (10x10 grid) on features extracted with the CNN from the box around each proposed region
    - project each mask to **superpixels** (Achanta et al. [2012]) by assigning each superpixel the average value of the coarse mask in it's area
    - together with the original proposal predict a **refined segmentation**
  - estimate a semantic segmentation by pasting the top scoring masks

Problems



- we rely on external region proposals, which are not accurate
- the refinement stage still needs improvement
- we would want a end-to-end training pipeline
- slow: 50s for one image

Introduction

Semantic Segmentation
- Fully Convolutional Networks
- Dilated Convolutions

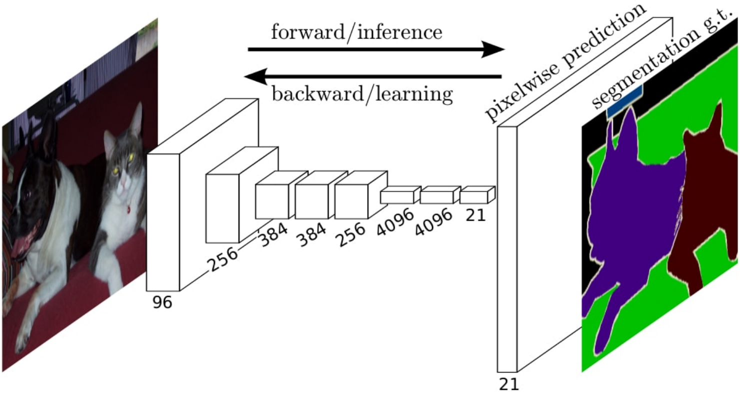
Instance segmentation
- DeepMask
- SharpMask

Fully Convolutional Networks

- J. Long, E. Shelhamer, and T. Darrell. Fully convolutionalnetworks for semantic segmentation. InProceedings of the IEEEConference on Computer Vision and Pattern Recognition, pages34313440, 2015

Fully Convolutional Networks

- Long et al. [2015] Segmantic segmentation



Fully Convolutional Networks (FCNs)

- Use CNN to predict the class of every pixel in the image
- Fine-tune an existing network like AlexNet, GoogLeNet or VGG, replasing only the last fully connected layer
- Apply the operations densely using 'fully convolutional networks'
- Improve the acuracy using 'skip-layers'

Convert CNN to Fully Convolutional Network(FCNs)

- convolution, pooling and activations layers are *local* operations
  - apply them on the whole image
- a *fully-connected* layer could be seen as *convolutional* layer
  - use filters with size of the receptive field of the fully-connected layer

Upsampling - Deconv

- The output will be sparse because of the pooling layers. Posible solutions:
  - Shift-and-stitch approach

[0] Image source: **?**

Upsampling - Deconv

- The output will be sparse because of the pooling layers. Posible solutions:
  - Shift-and-stitch approach
  - Upsample by bilinear interpolation

[0] Image source: **?**

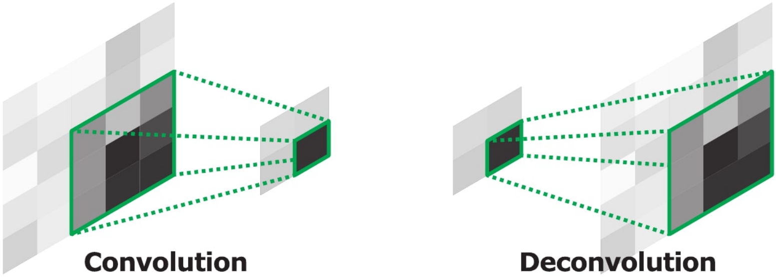
Upsampling - Deconv

- The output will be sparse because of the pooling layers. Posible solutions:
  - Shift-and-stitch approach
  - Upsample by bilinear interpolation
  - Use deconvolution layer to learn an upsampling operation

[0] Image source: **?**

Upsampling - Deconv

- The output will be sparse because of the pooling layers. Posible solutions:
  - Shift-and-stitch approach
  - Upsample by bilinear interpolation
  - Use deconvolution layer to learn an upsampling operation



---

[0] Image source: **?**

Skip Layers

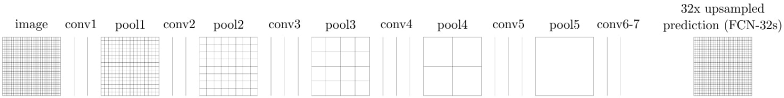
- Combine the outputs of an *upper* and an *intermediate* layer
- **Q:** Why would low-level features help?

Skip Layers

- Combine the outputs of an *upper* and an *intermediate* layer
- **Q:** Why would low-level features help?
  - Useful for aligning the output with image boundaries, or with part of objects

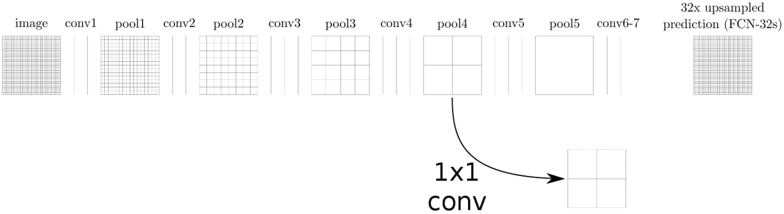
Skip Layers

- Combine the outputs of an *upper* and an *intermediate* layer
- **Q:** Why would low-level features help?
  - Useful for aligning the output with image boundaries, or with part of objects



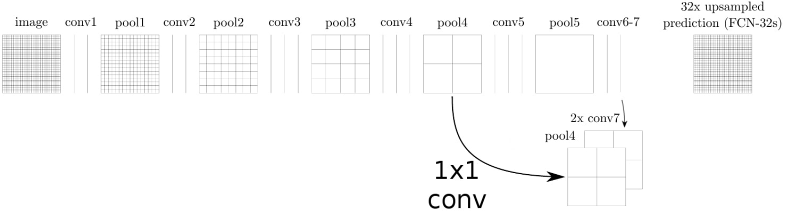
Skip Layers

- Combine the outputs of an *upper* and an *intermediate* layer
- **Q:** Why would low-level features help?
  - Useful for aligning the output with image boundaries, or with part of objects
- Use a 1x1 convolutional layer on features from lower levels to produce class predictions



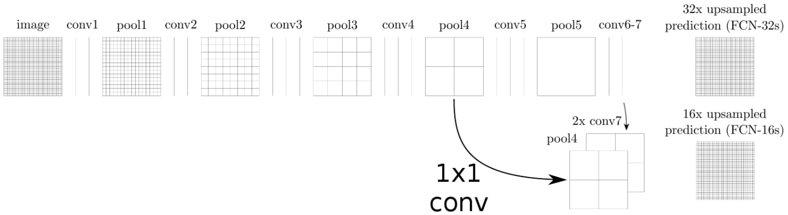
Skip Layers

- Combine the outputs of an *upper* and an *intermediate* layer
- **Q:** Why would low-level features help?
  - Useful for aligning the output with image boundaries, or with part of objects
- Use a 1x1 convolutional layer on features from lower levels to produce class predictions
- Upsample and add prediction scores from different levels then do a softmax



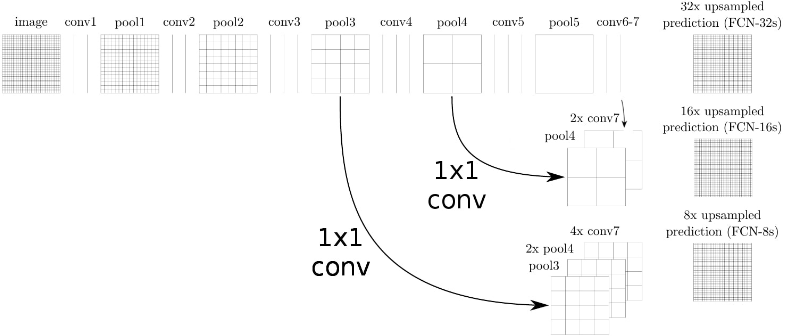
Skip Layers

- Combine the outputs of an *upper* and an *intermediate* layer
- **Q:** Why would low-level features help?
  - Useful for aligning the output with image boundaries, or with part of objects
- Use a 1x1 convolutional layer on features from lower levels to produce class predictions
- Upsample and add prediction scores from different levels then do a softmax

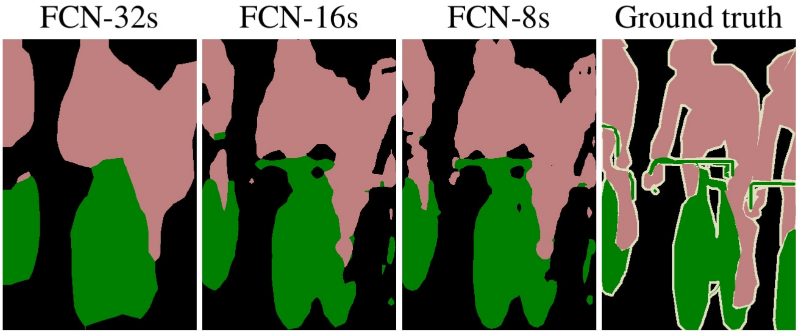


Skip Layers

- Combine the outputs of an *upper* and an *intermediate* layer
- **Q:** Why would low-level features help?
  - Useful for aligning the output with image boundaries, or with part of objects
- Use a 1x1 convolutional layer on features from lower levels to produce class predictions
- Upsample and add prediction scores from different levels then do a softmax



Results



Results

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [10]	47.9	-	-
SDS [15]	52.6	51.6	$\sim$ 50 s
FCN-8s	**62.7**	**62.2**	$\sim$ **175 ms**

Dilated Convolutions

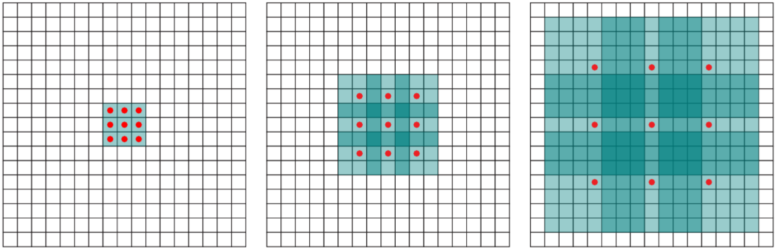
- Yu, Fisher, and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." arXiv preprint arXiv:1511.07122 (2015).

Dilated Convolutions

- Improve the semantic segmentation by using on operation designed for dense prediction
- Aggregate contextual information without losing image resolution

$$(F * k)(p) = \sum_{s+p=t} F(t)k(t) \tag{1}$$

$$(F *_l k)(p) = \sum_{s+l \cdot p=t} F(t)k(t) \tag{2}$$



(a) (b) (c)

Front End

- Adapted from the VGG-16 network
- Remove last 2 pooling, strided layers
- For each pooling eliminated increase the dilation factor by 2
- The parameters can be initialized from VGG
- The output of the network has higher resolution

Context Module

- Increase the receptive field of a feature volume
- Both the input and the output are C feature maps
- The module could be trained individually or it can be plugged into existing architectures
- Use 7 convolutional 3x3 layers with dilation factors: 1, 1, 2, 4, 8, 16, and 1
- The receptive field increases from 3x3 to 67x67

Results

Bitdefender

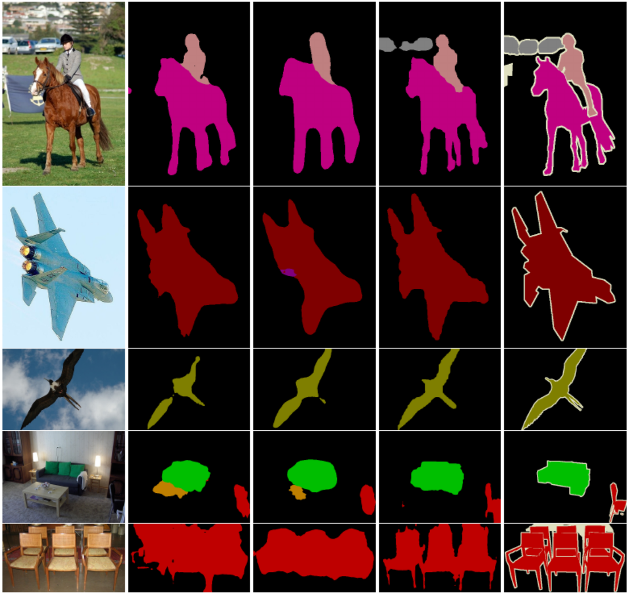
	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	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**

Table 2: Our front-end prediction module is simpler and more accurate than prior models. This table reports accuracy on the VOC-2012 test set.

Results

	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**

Results



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

Introduction

Semantic Segmentation

Fully Convolutional Networks

Dilated Convolutions

Instance segmentation

DeepMask

SharpMask

Segment Object Candidates

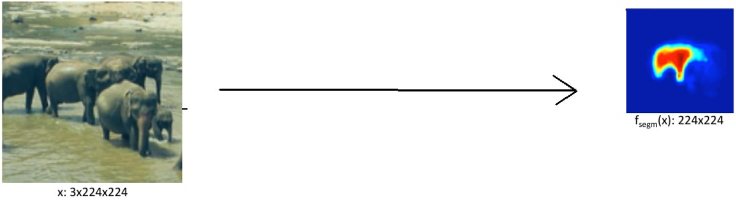
- P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segmentobject candidates. InAdvances in Neural Information ProcessingSystems, pages 19901998, 2015.

Segment Object Candidates

- Goals:
  - Work even when multiple object instances are present
  - Segment each individual object instance in the image
  - Segment categories not seen during training
- Pinheiro et al. [2015]: DeepMask
  - Learn to produce segmentation proposals using CNNs

DeepMask

- Pinheiro et al. [2015]

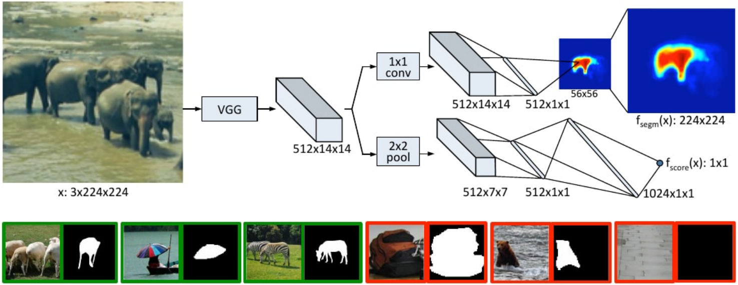


DeepMask: Method

- training is done on 224x224 pixel patches
- predict a segmentation mask and a score for every patch
- at training time the true mask and a label are given
  - the mask contains all the pixel of a centered object
  - the label is 1 if the patch contains a centered object and fully contained in the patch
  - **Q:** Why do they use this score?

DeepMask: Architecture

- Pinheiro et al. [2015]



DeepMask: Architecture

- feature extraction using VGG-A network (Simonyan and Zisserman [2014])
  - eight 3x3 convolution layers (each followed by ReLU) and four 2x2 max-pooling layers
- Segmentation part
  - one 1x1 convolutional layer
  - one fully-connected layer
  - one fully-connected layer with $h^o \times w^o$ outputs representing a low resolution mask
- Score part
  - one pooling layer and two fully-connected layers

Loss

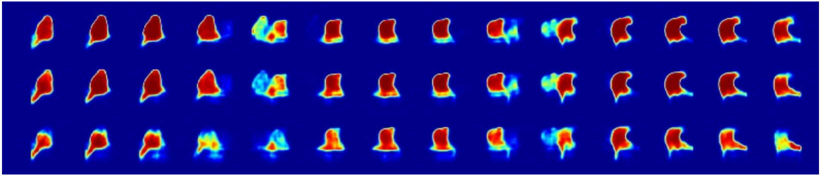
Bitdefender

$$\mathcal{L}(\theta) = \sum_k \left(\frac{1+y_k}{2w^o h^o} \sum_{ij} \log(1 + e^{-m_k^{ij} f_{segm}^{ij}(x_k)}) + \lambda \log(1 + e^{-y_k f_{score}(x_k)}) \right)$$

- score loss
  - the score predicts is an object is centered and fully contained in the input patch.
  - networks learns to segment only whole objects even when multiple objects are present.
  - segments the object at the appropriate scale
- segmentation loss
  - sum over all the 56x56 pixel losses
  - propagate segmentation gradients only for $y_k = 1$. This way the network tries to generate a mask even for unknown objects

Full Scene Inference

- Apply all the computations densely at
  - every pixel location with a stride of 16 pixels
  - multiple scales (scales 2^{-2} to 2^{1} with a step of $2^{1/2}$)
- Produce a segmentation mask and a score for every location and scale

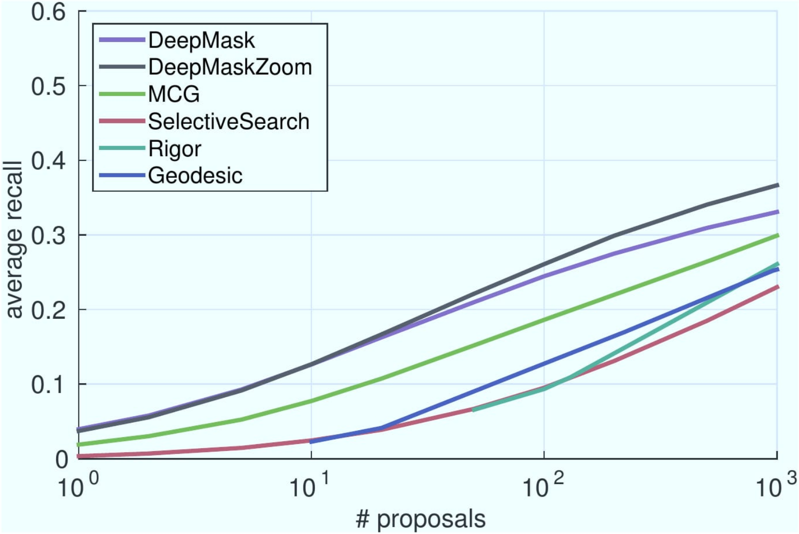


Results

- Train on MS COCO 2014 training set witch contains 80k images and a total of nearly 500k segmented objects
- evaluate on COCO 2014 validation set and on PASCAL VOC 2007 test set
- use Intersection over Union (IoU) metric: area of the intersection between the proposal and the ground-truth divided by their union area
- compute the average recall (AR) using IoU $\in [0.5, 1]$

Results

DeepMask result on COCO dataset



Results

Conclusions

Bitdefender

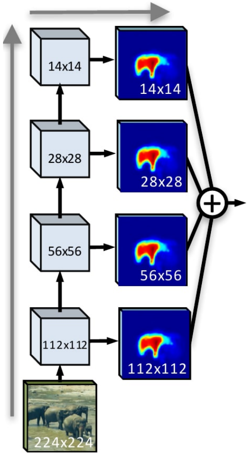
- the masks have good localization, but are poor at defining precise contours
- high level information are necessary to accurately detect an object in its entirely
- low-level information is needed for aligning the segmentation with the true boundaries of the object
- Pinheiro et al. [2016]
  - uses as baseline the preceding architecture
  - produce a refined segmentation in a top-down fashion
  - invert the loss of resolution from the pooling layers

Bitdefender

SharpMask

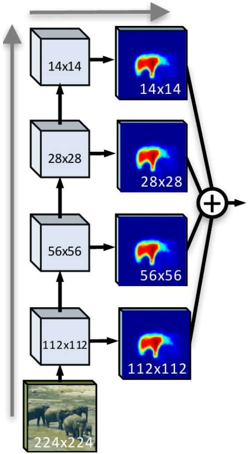
- Pinheiro, Pedro O., Tsung-Yi Lin, Ronan Collobert, and Piotr Dollar. "Learning to refine object segments." In European Conference on Computer Vision, pp. 75-91. Springer International Publishing, 2016..

Skip layers?



- The mask have good localization but are poor at aligning with the precise object shape.
- We have shown that skip layers could improve accuracy.
- Why not using them in this context of instance segmentation?

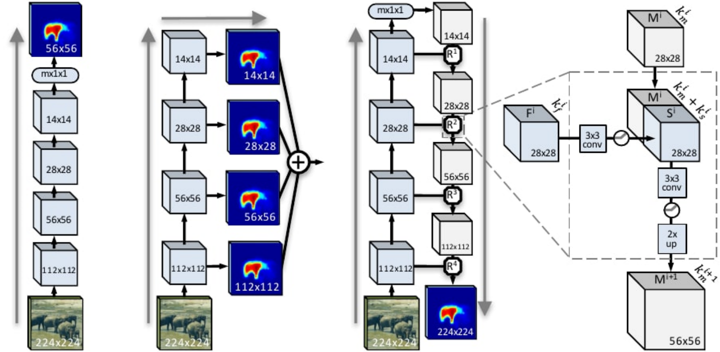
Skip layers?



- skip layers are equivalent to make multiple prediction from different depths then combine all the results
- for object segmentation we have to differentiate between object instances
- high-level information is needed
- "local patches of sheep fur can be labeled as such but without object-level information it can be difficult to determine if they belong to the same animal"

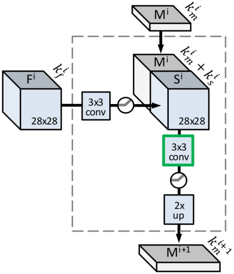
SharpMask architecture

- Pinheiro et al. [2016]



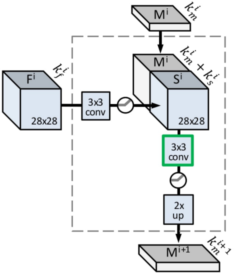
(a) feedforward (b) feedforward + skip (c) proposed network (d) refinement module

Refinement module



- each module is responsible of integrating an upper mask encoding M_i with the features of a lower level F_i
- generate a new mask encoding with double the spatial resolution $M_{i+1} = R_i(M_i, F_i)$
- a problem occurs because of the dimensions of these features: F_i, has the same spacial dimensions as M_i but has more channels, especially in the higher layers.
  - this could be computationally expensive
  - the useful mask encoding information M_i could be lost between the the F_i features

Refinement module



- lower the channel dimension of the F_i features by using a 3x3 convolutional layer and produce new 'skip' features
- concatenate the this new features with M_i and use another 3x3 convolutional layer to produce a new mask encoding
- bilinear upsample the mask encoding to produce the final M_{i+1} output

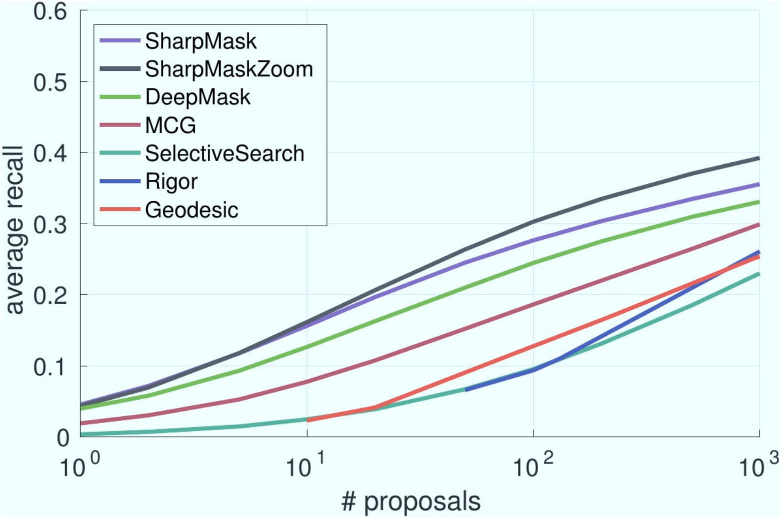
Training and Inference

- use the same loss function as DeepMask
- train in two stages: first train the original DeepMask part, then 'freeze' it and train the refinement modules
  - this leads to faster convergence time
  - the gains of fine-tune the entire network are minimal
  - a coarse mask can still be produce only from the first part
- at inference time most of the computations could be applied densely
- the top $M1$ mask encoding is different at every location
  - the computations in the following refinement module are done independently: **slow**
  - to save computational time they only refine only the top scoring proposals

Results

	Box Proposals				Segmentation Proposals						
	AR^{10}	AR^{100}	AR^{1K}	AUC	AR^{10}	AR^{100}	AR^{1K}	AUC^{S}	AUC^{M}	AUC^{L}	AUC
EdgeBoxes [34]	7.4	17.8	33.8	13.9	—	—	—	—	—	—	—
Geodesic [36]	4.0	18.0	35.9	12.6	2.3	12.3	25.3	1.3	8.6	20.5	8.5
Rigor [37]	—	13.3	33.7	10.1	—	9.4	25.3	2.2	6.0	17.8	7.4
SelectiveSearch [33]	5.2	16.3	35.7	12.6	2.5	9.5	23.0	0.6	5.5	21.4	7.4
MCG [35]	10.1	24.6	39.8	18.0	7.7	18.6	29.9	3.1	12.9	32.4	13.7
RPN [7, 8]	12.8	29.2	42.6	21.4	—	—	—	—	—	—	—
DeepMask [22]	15.3	31.3	44.6	23.3	12.6	24.5	33.1	2.3	26.6	33.6	18.3
DeepMaskZoom [22]	15.0	32.6	48.2	24.2	12.7	26.1	36.6	6.8	26.3	30.8	19.4
DeepMask-ours	18.7	34.9	46.5	26.2	14.4	25.8	33.1	2.2	27.3	37.4	19.4
SharpMask	19.7	36.4	48.2	27.4	15.6	27.6	35.5	2.5	29.1	40.4	20.9
SharpMaskZoom	20.1	39.4	52.8	29.1	16.1	30.3	39.2	6.9	29.7	38.4	22.4
SharpMaskZoom^2	19.2	39.9	55.0	29.2	15.4	30.7	40.8	10.6	27.3	36.0	22.5

Results



SharpMask
SharpMaskZoom
DeepMask
MCG
SelectiveSearch
Rigor
Geodesic
average recall
0.6
0.5
0.4
0.3
0.2
0.1
0
proposals
10^0
10^1
10^2
10^3

Results

Results

(a) DeepMask Output (b) SharpMask Output

Recap

- Fully Convolutional Segmentation
  - semantic segmentation
  - traing and inference efficiently by applying the operations densly
  - use of skip layers
- Dilated Convolution
  - increase receptive fields without losing resolution
- DeepMask
  - instance segmentation
  - inference efficiently
- SharpMask
  - instance segmentation
  - inference efficiently
  - use of top-down reffinemnt modules

References I

Bitdefender

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–335, 2014.

S. Bauer, L.-P. Nolte, and M. Reyes. Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 354–361. Springer, 2011.

References II

J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *European Conference on Computer Vision*, pages 430–443. Springer, 2012.

C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

References III

D. Marmanis, J. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla. Semantic segmentation of aerial images with an ensemble of cnns. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:473–480, 2016.

P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015.

P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.