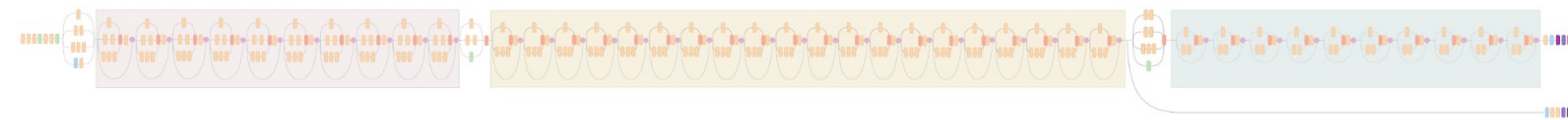
Lecture 23
Deep Learning: Segmentation

COS 429: Computer Vision

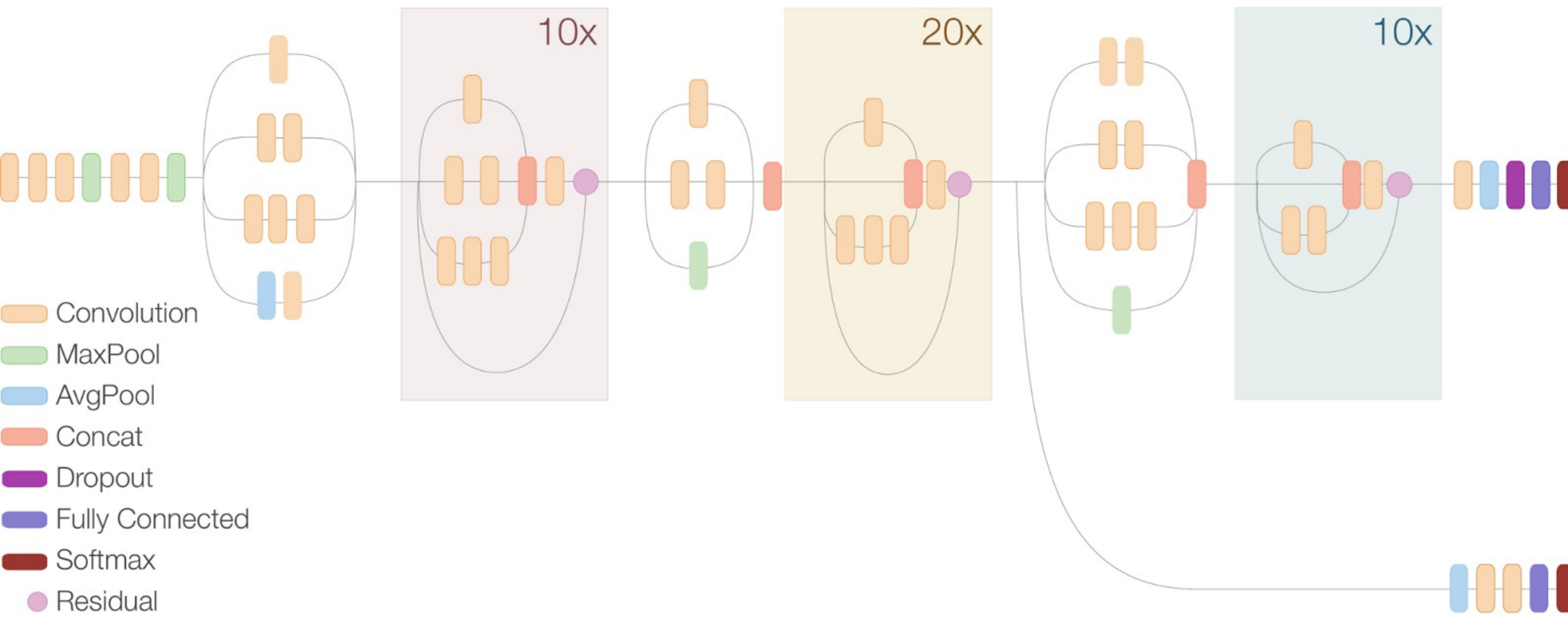
PRINCETON UNIVERSITY

Thanks: most of these slides shamelessly adapted from
Stanford CS231n: Convolutional Neural Networks for Visual Recognition
Fei-Fei Li, Andrej Karpathy, Justin Johnson
http://cs231n.stanford.edu/

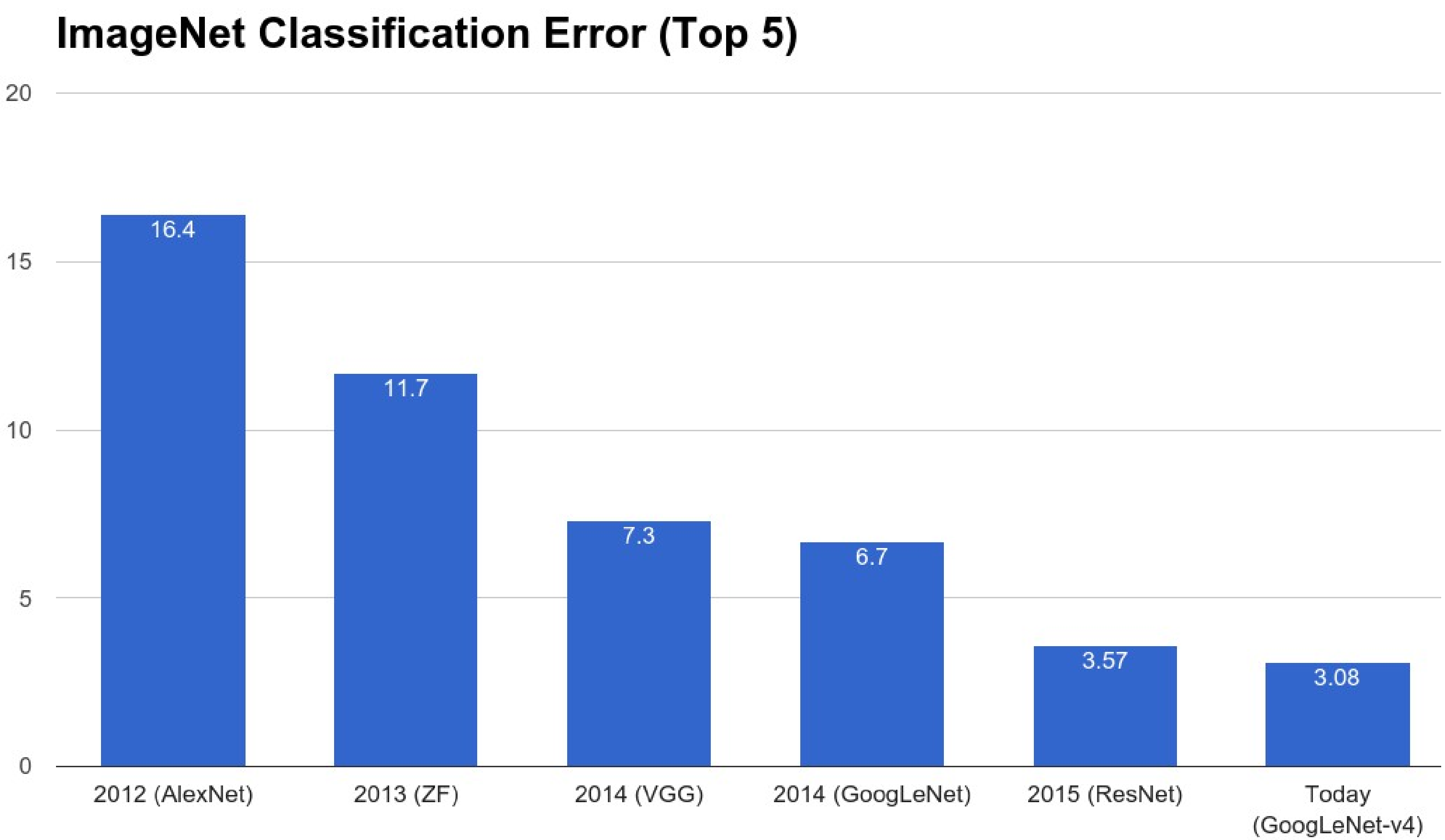
Inception Resnet V2 Network



Compressed View



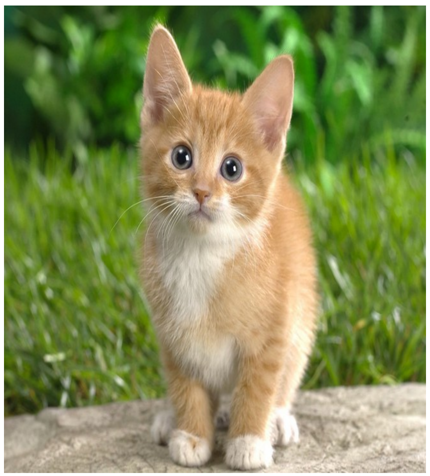
 Slide Credit:



Szegedy et al, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, arXiv 2016

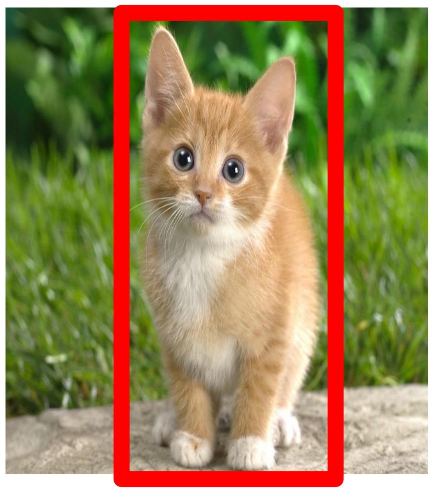
Computer Vision Tasks

Classification



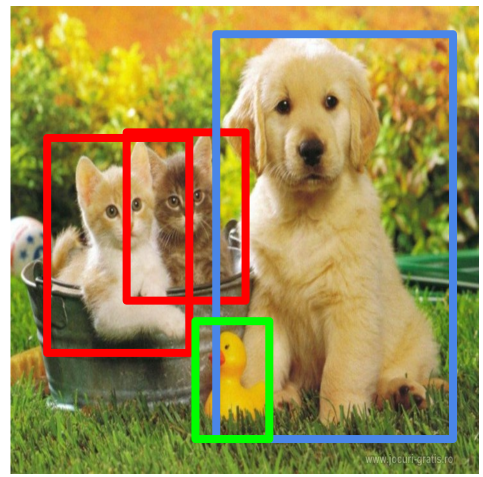
CAT

Classification + Localization



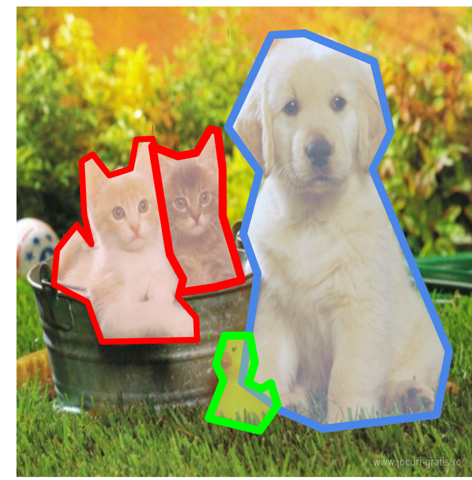
CAT

Object Detection



CAT, DOG, DUCK

Instance Segmentation



CAT, DOG, DUCK

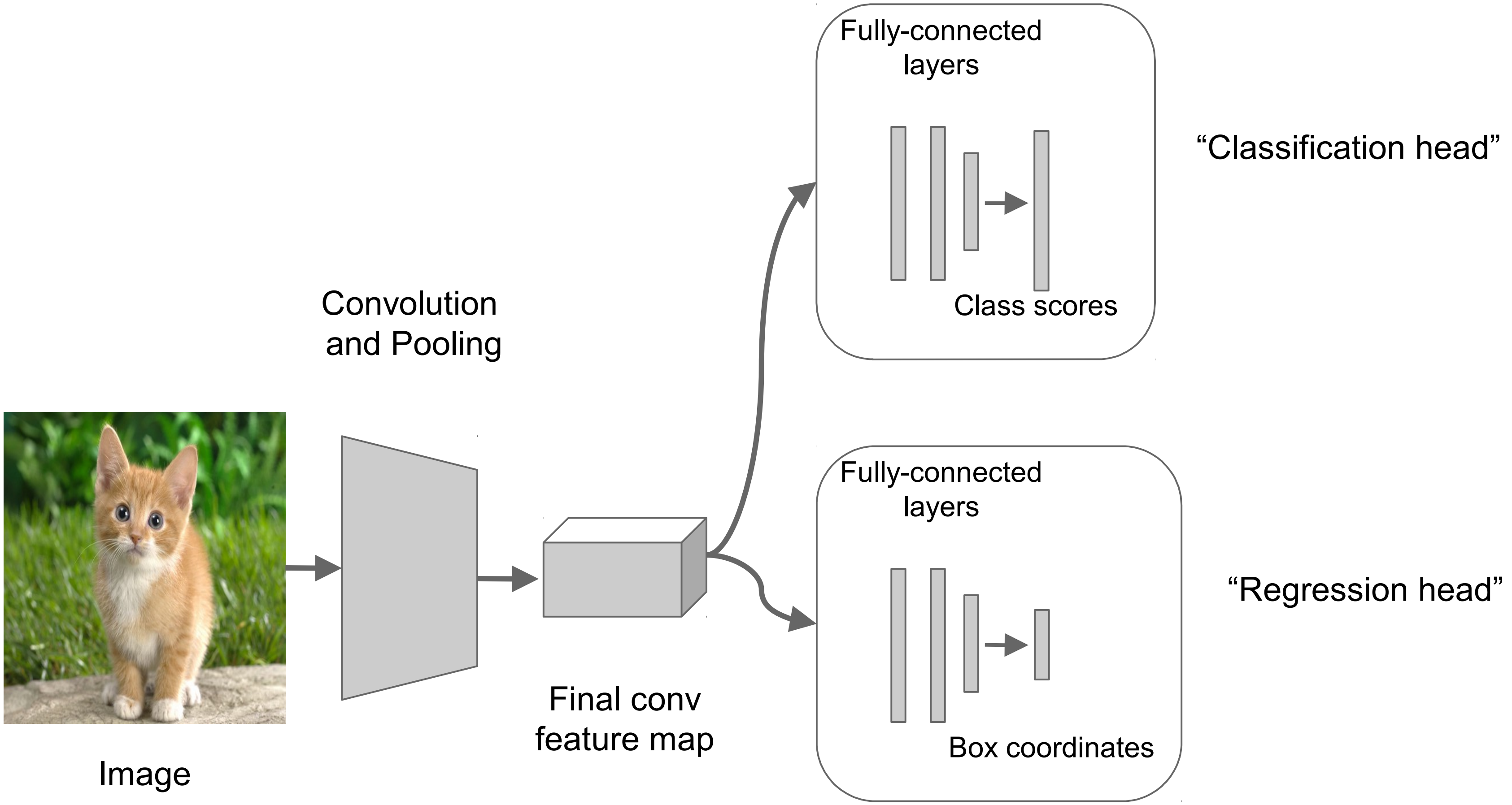
Single object

Multiple objects

Slide Credit:

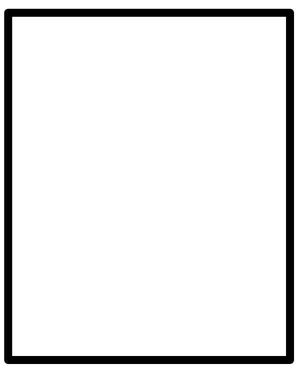
Simple Recipe for Classification + Localization

Step 2: Attach new fully-connected "regression head" to the network

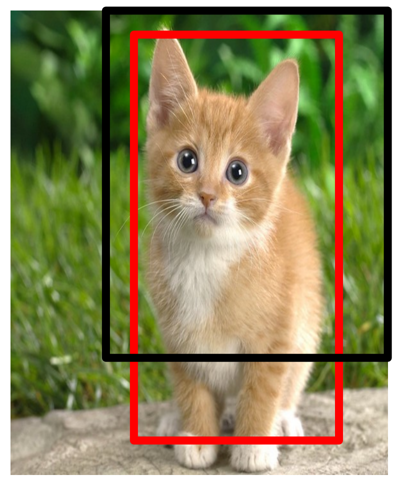


Slide Credit:

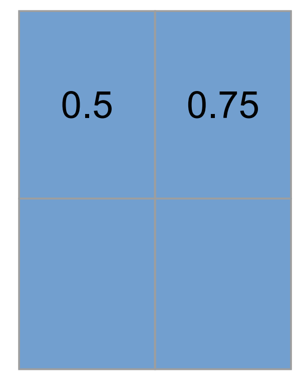
Sliding Window: Overfeat



Network input:
3 x 221 x 221

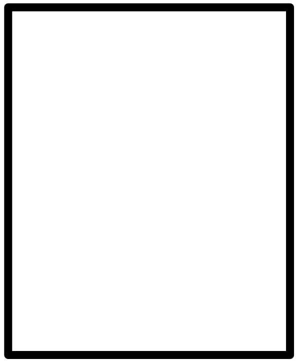


Larger image:
3 x 257 x 257

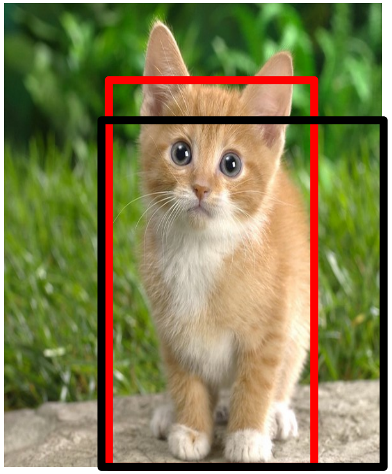


Classification scores:
P(cat)

Sliding Window: Overfeat



Network input:
3 x 221 x 221



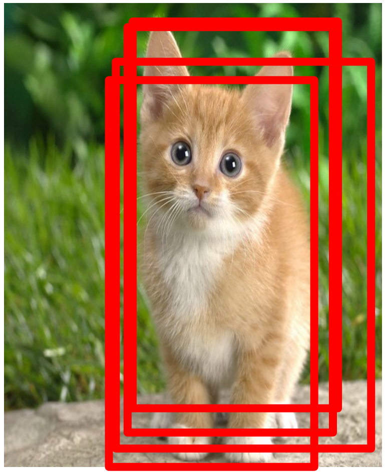
Larger image:
3 x 257 x 257

0.5	0.75
0.6	0.8

Classification scores:
P(cat)

Sliding Window: Overfeat

Network input:
3 x 221 x 221



Larger image:
3 x 257 x 257

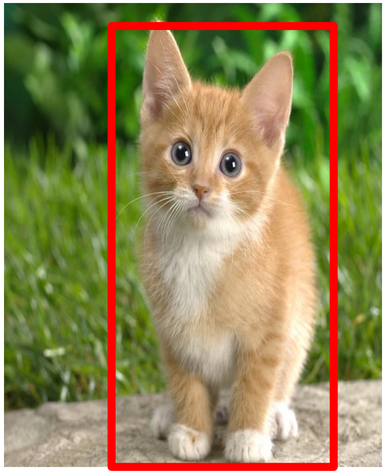
0.5	0.75
0.6	0.8

Classification scores:
P(cat)

Sliding Window: Overfeat

Greedily merge boxes and scores (details in paper)

Network input:
3 x 221 x 221



Larger image:
3 x 257 x 257

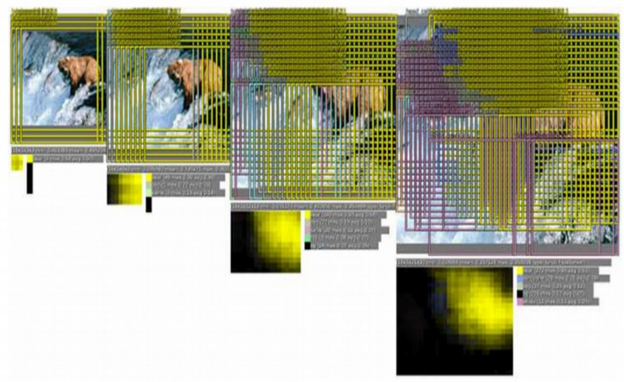
0.8

Classification score:
P(cat)

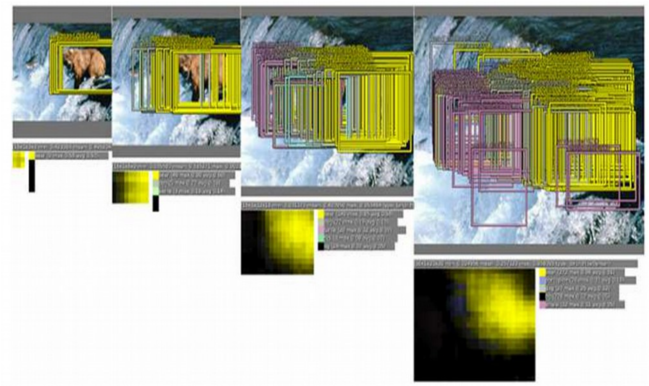
Sliding Window: Overfeat

In practice use many sliding window locations and multiple scales

Window positions + score maps



Box regression outputs

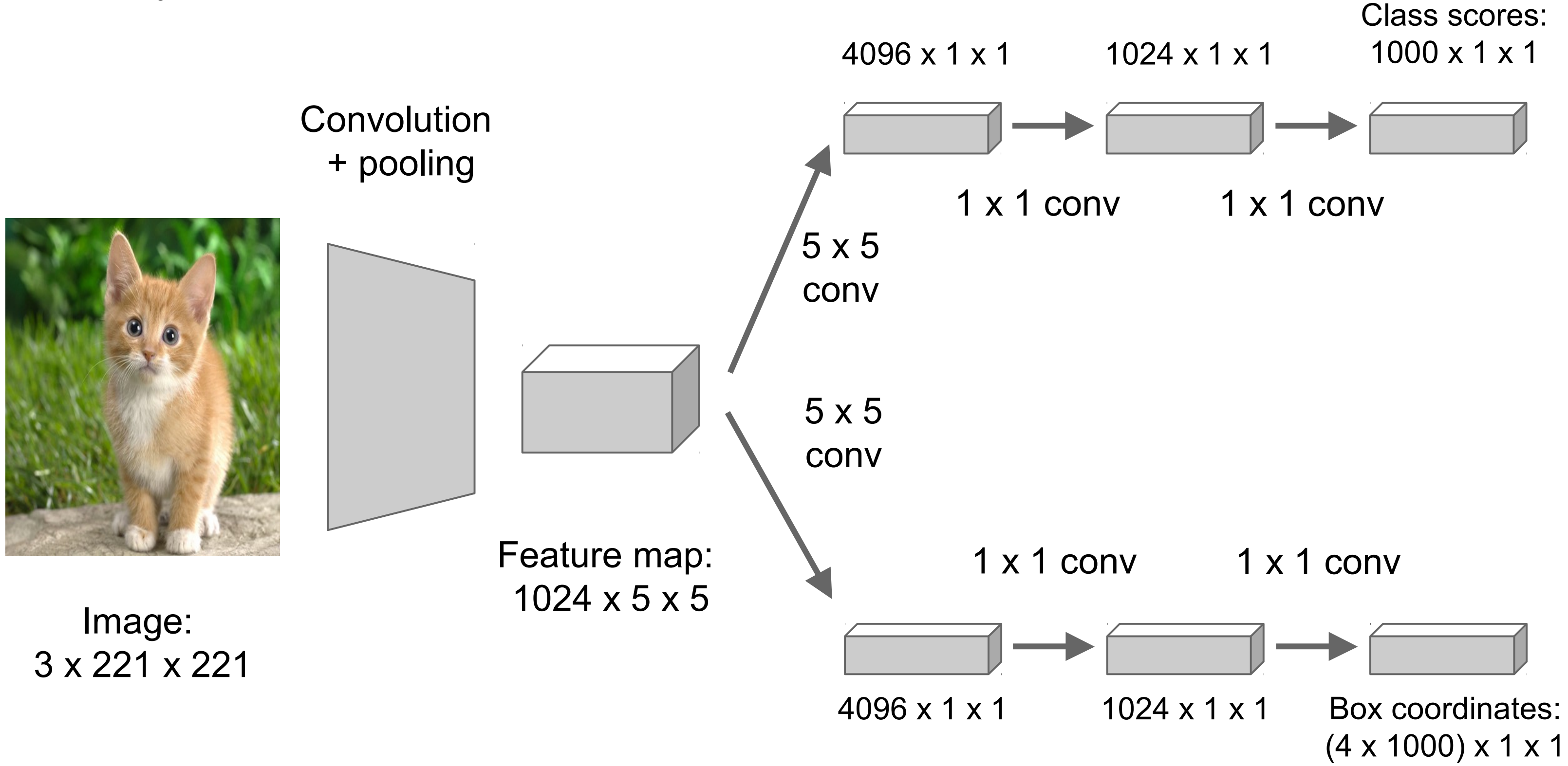


Final Predictions

Sermanet et al, “Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Efficient Sliding Window: Overfeat

Efficient sliding window by converting fully-connected layers into convolutions



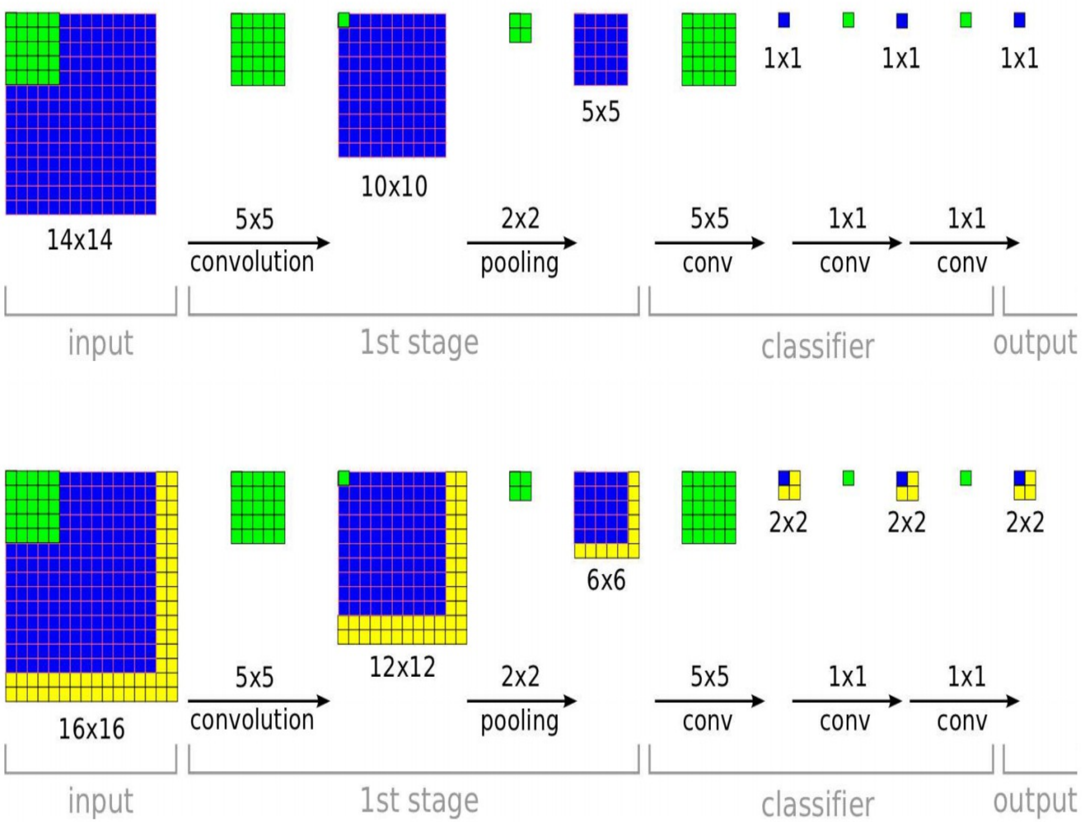
11

Slide Credit:

Efficient Sliding Window: Overfeat

Training time: Small image, 1 x 1 classifier output

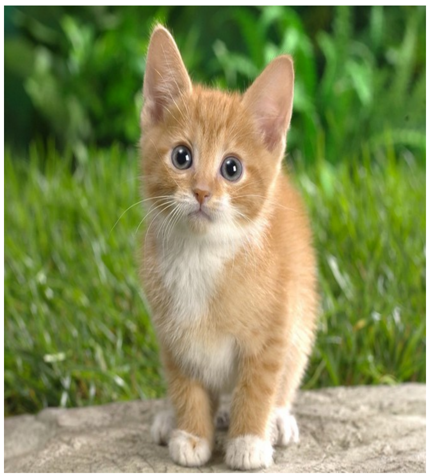
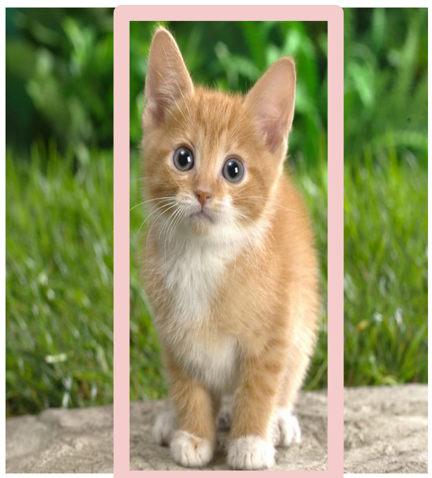
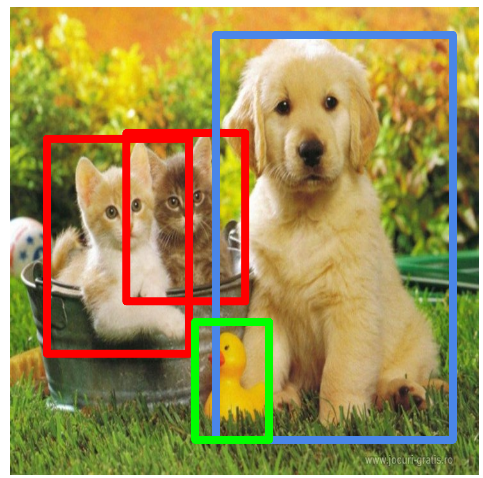
Test time: Larger image, 2 x 2 classifier output, only extra compute at yellow regions



Sermanet et al, "Integrated Recognition, Localization and Detection using Convolutional Networks", ICLR 2014

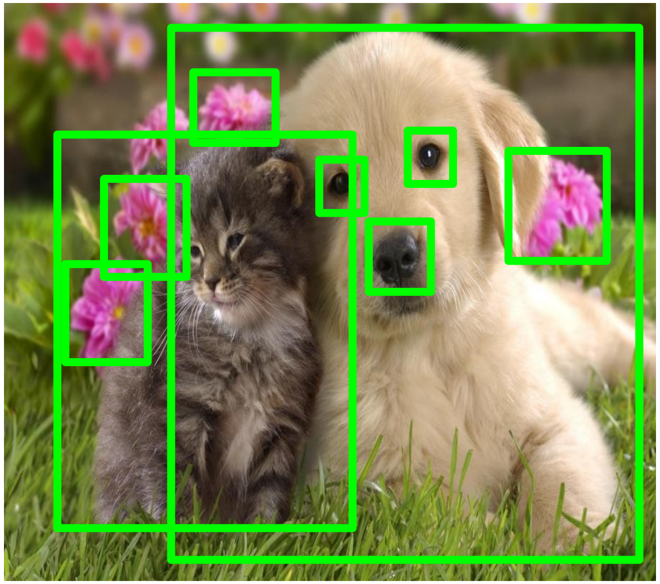
Computer Vision Tasks

Classification	Classification + Localization	**Object Detection**	Instance Segmentation


Region Proposals

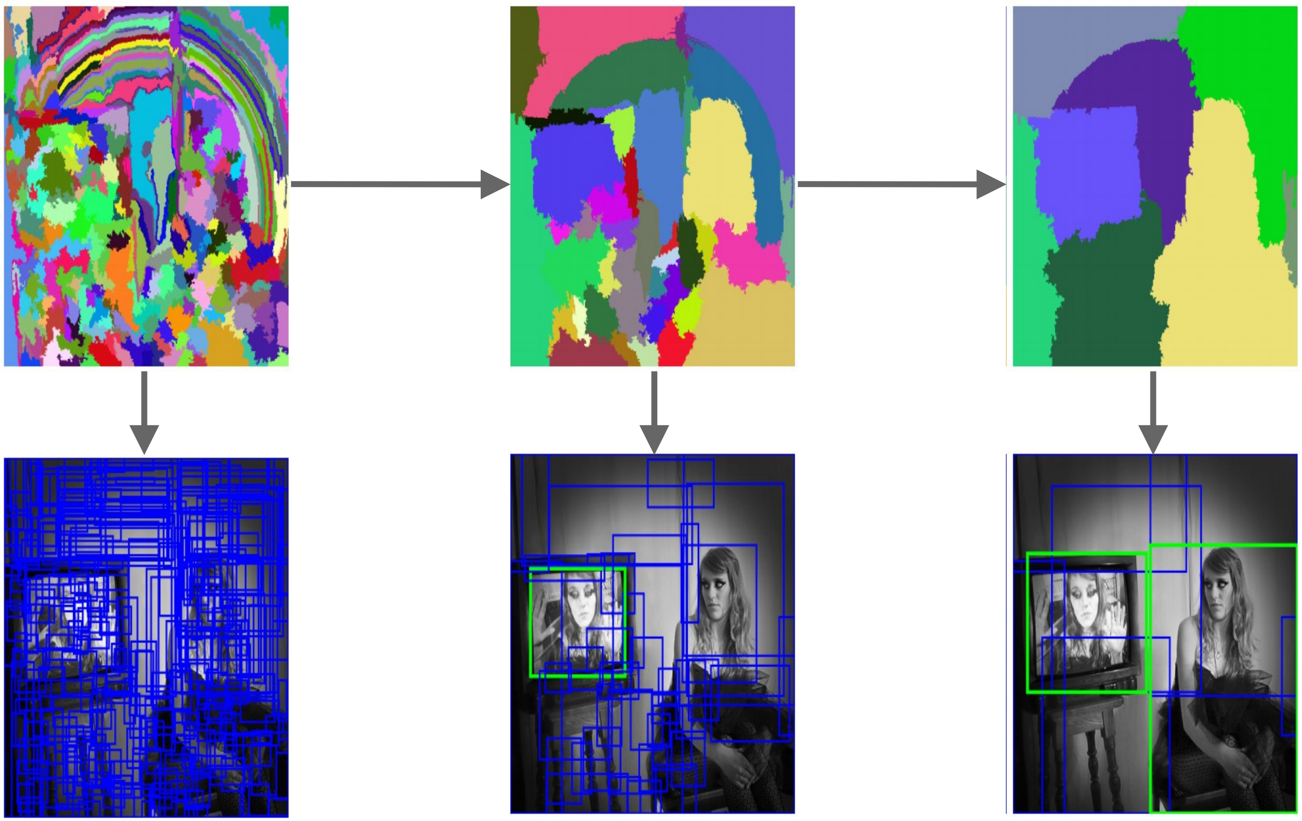
- Find “blobby” image regions that are likely to contain objects
- “Class-agnostic” object detector
- Look for “blob-like” regions



Region Proposals: Selective Search

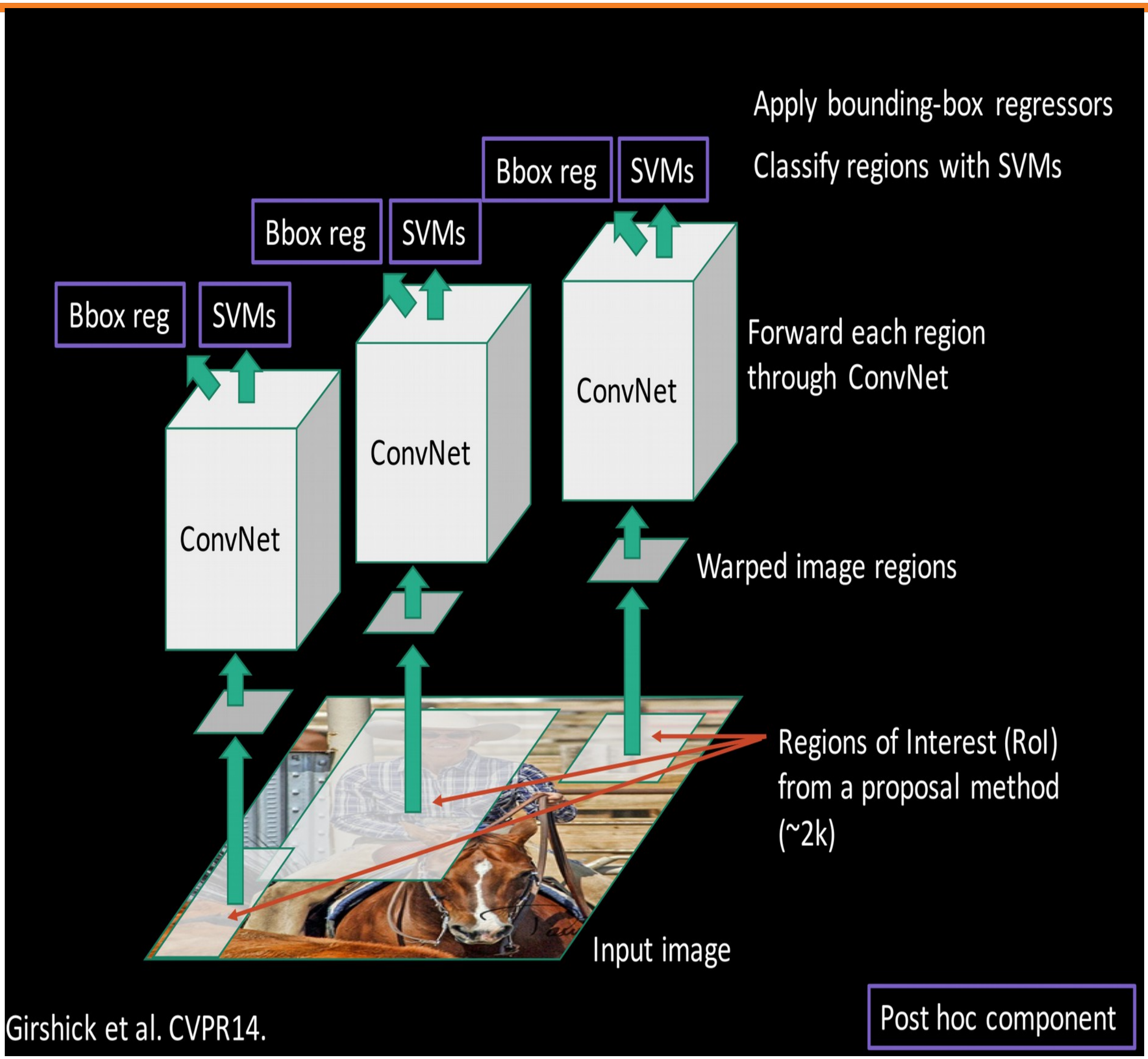
Bottom-up segmentation, merging regions at multiple scales

Convert regions to boxes



Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013

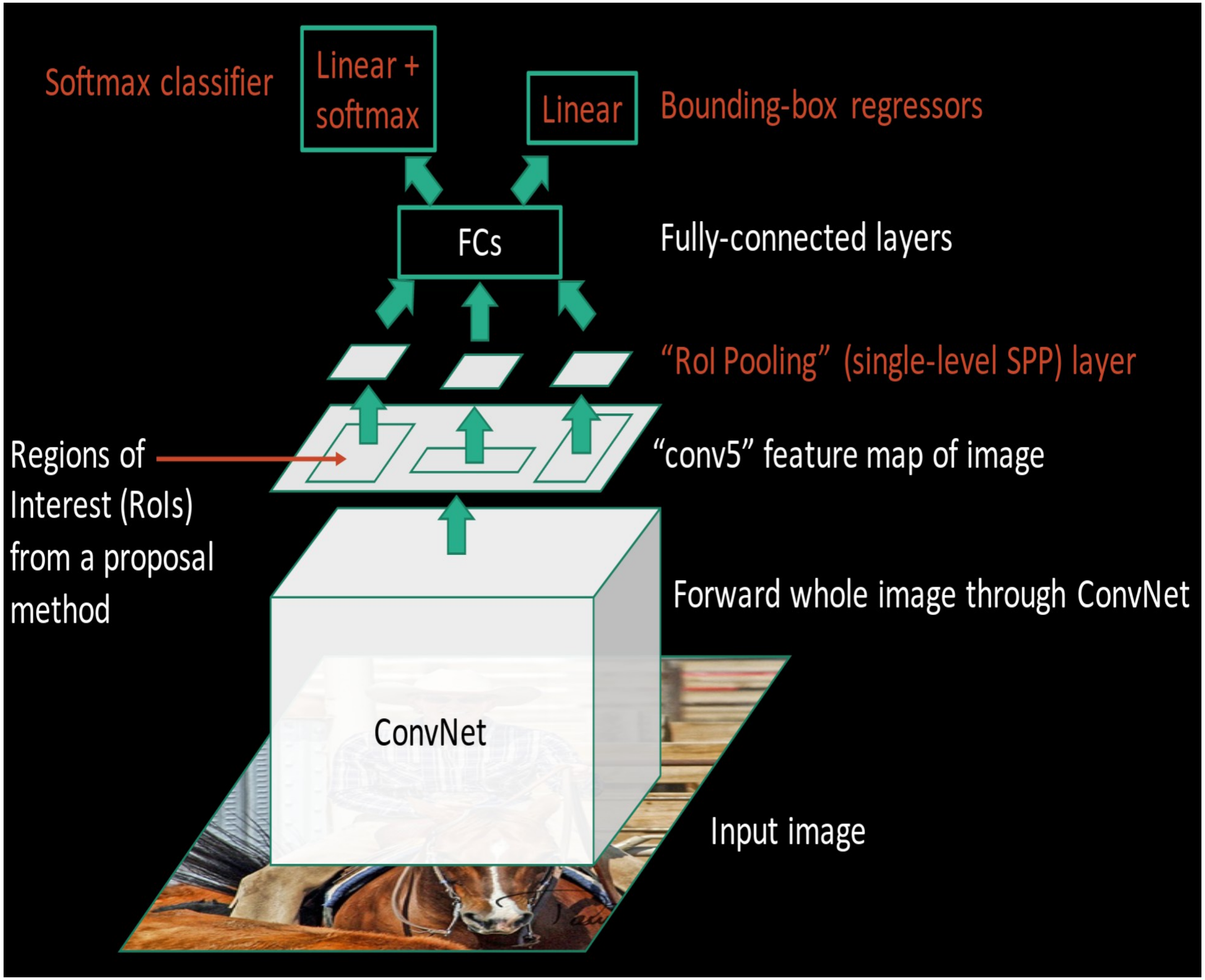
R-CNN



Girschick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014

Slide credit: Ross Girschick

Fast R-CNN



R-CNN Problems:
Slow at test-time due to independent forward passes of the CNN

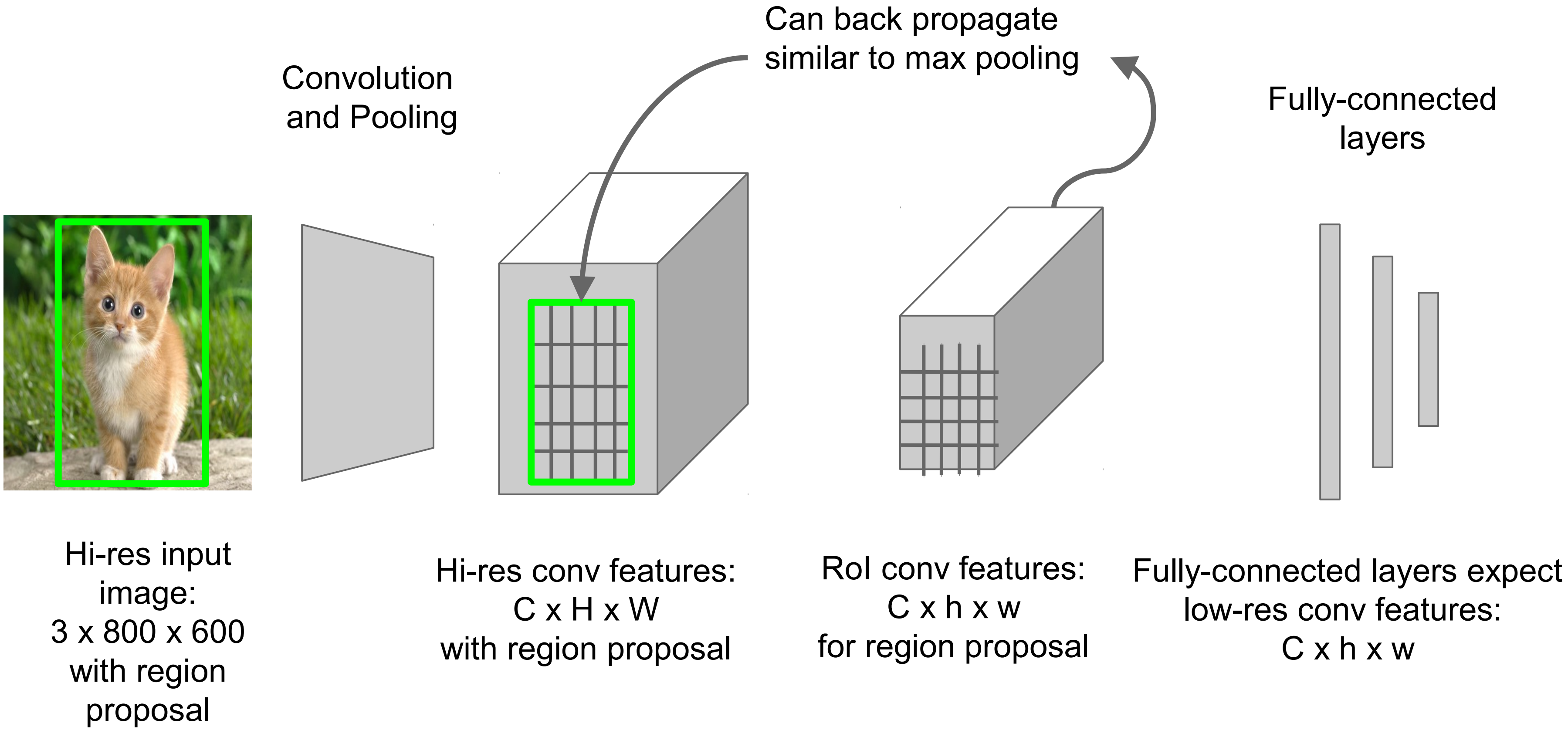
Solution:
Share computation of convolutional layers between proposals for an image

R-CNN Problems:
- Post-hoc training: CNN not updated in response to final classifiers and regressors
- Complex training pipeline

Solution:
Just train the whole system end-to-end all at once!

Slide Credit:

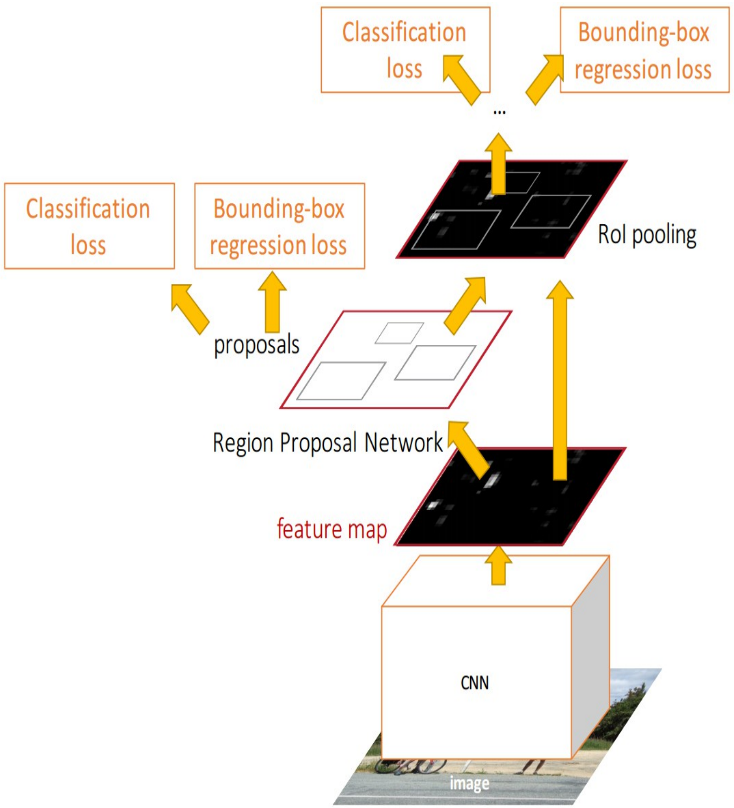
Fast R-CNN: Region of Interest Pooling



Faster R-CNN: Training

In the paper: Ugly pipeline
- Use alternating optimization to train RPN, then Fast R-CNN with RPN proposals, etc.
- More complex than it has to be

Since publication: Joint training!
One network, four losses
- RPN classification (anchor good / bad)
- RPN regression (anchor -> proposal)
- Fast R-CNN classification (over classes)
- Fast R-CNN regression (proposal -> box)



Slide credit: Ross Girschick

Faster R-CNN: Results

	R-CNN	**Fast R-CNN**	**Faster R-CNN**
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

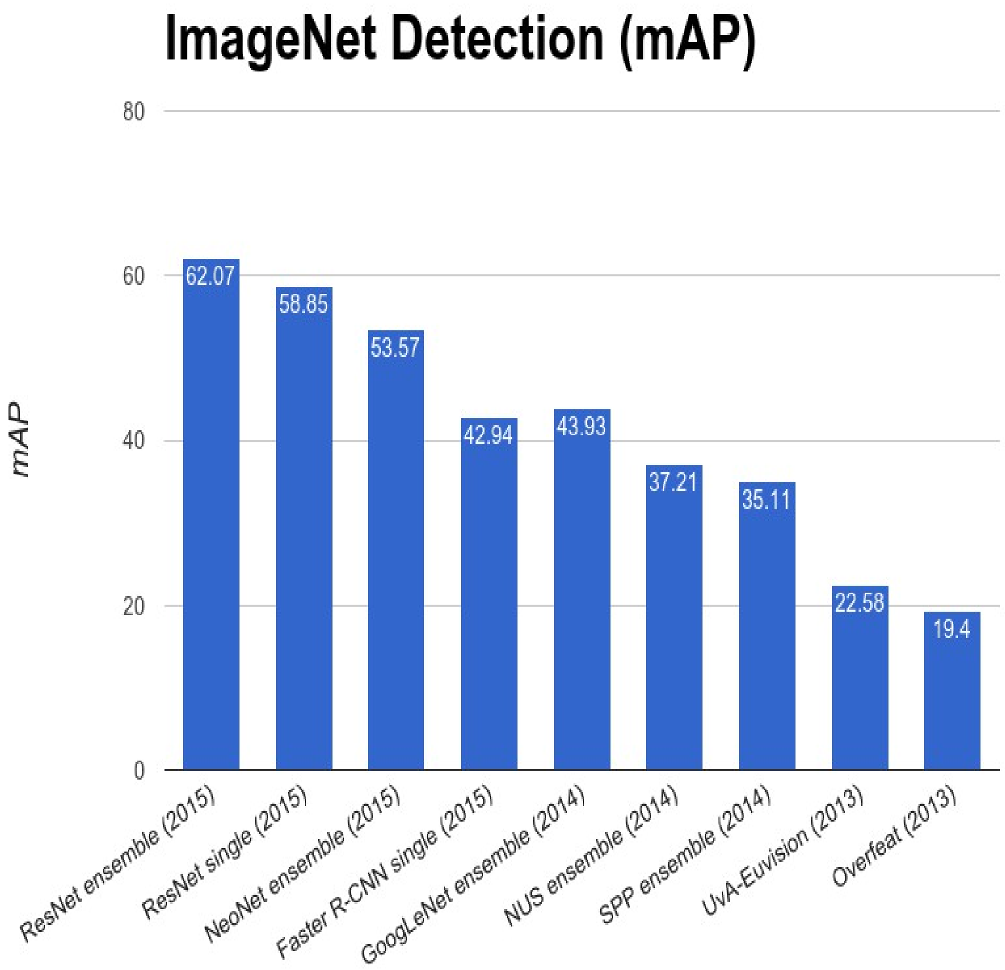
Slide Credit:

Object Detection State-of-the-art: ResNet 101 + Faster R-CNN + some extras

training data	COCO train		COCO trainval	
test data	COCO val		COCO test-dev	
mAP	@.5	@[.5, .95]	@.5	@[.5, .95]
baseline Faster R-CNN (VGG-16)	41.5	21.2		
baseline Faster R-CNN (ResNet-101)	48.4	27.2		
+box refinement	49.9	29.9		
+context	51.1	30.0	53.3	32.2
+multi-scale testing	53.8	32.5	**55.7**	**34.9**
ensemble			**59.0**	**37.4**

He et. al, "Deep Residual Learning for Image Recognition", arXiv 2015

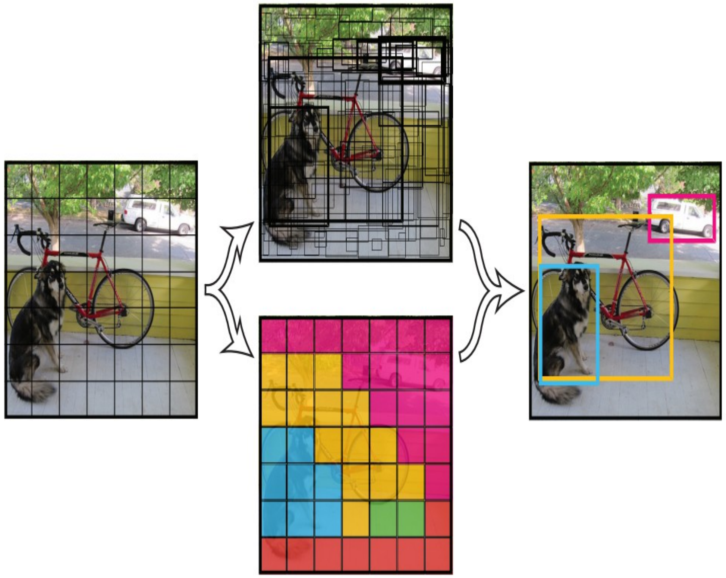
ImageNet Detection 2013 - 2015



Slide Credit:

YOLO: You Only Look Once

Detection as Regression

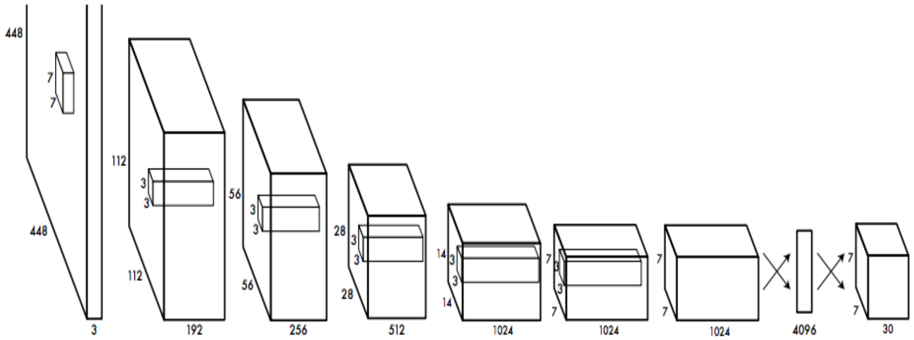


Divide image into S x S grid

Within each grid cell predict:
- B Boxes: 4 coordinates + confidence
- Class scores: C numbers

Regression from image to
7 x 7 x (5 * B + C) tensor

Direct prediction using a CNN



Redmon et al, "You Only Look Once:
Unified, Real-Time Object Detection", arXiv 2015

YOLO: You Only Look Once
Detection as Regression

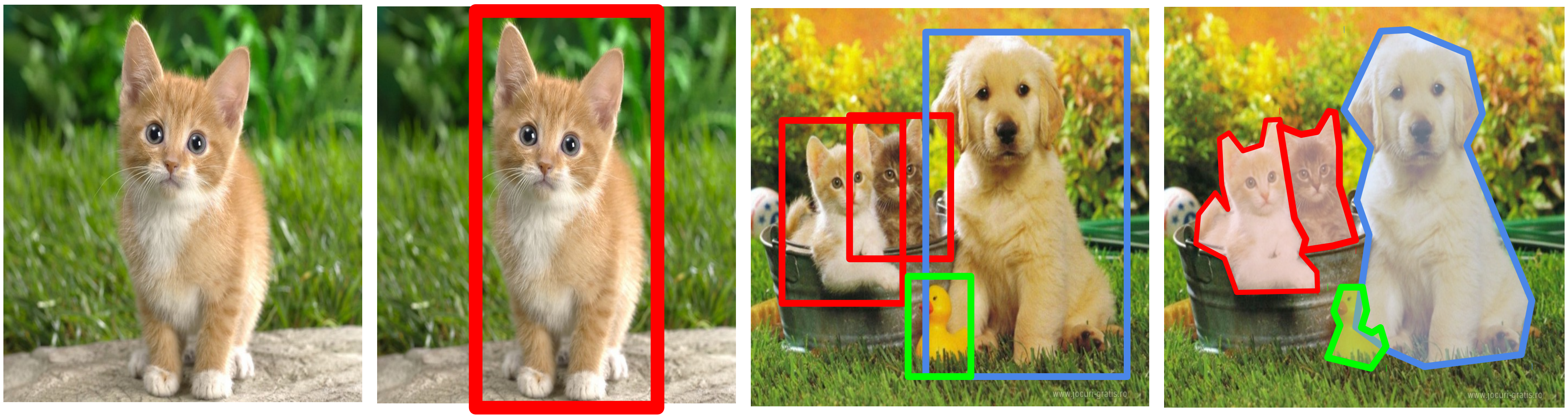
Faster than Faster R-CNN, but not as good

Redmon et al, “You Only Look Once:
Unified, Real-Time Object Detection”, arXiv 2015

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18

Computer Vision Tasks

Classification	Classification + Localization	Object Detection	Segmentation

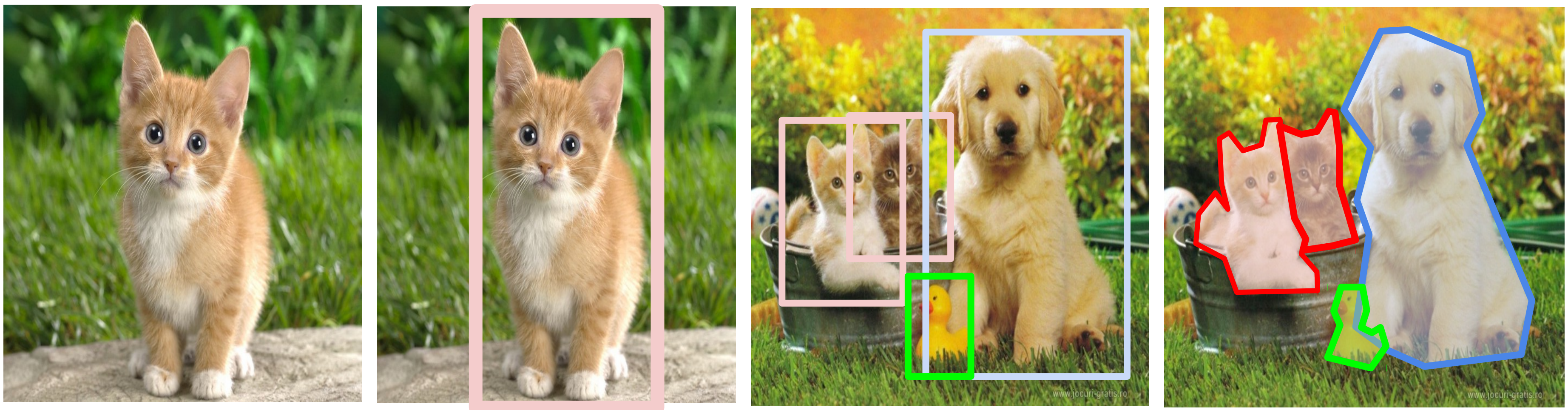


CAT	CAT	CAT, DOG, DUCK	CAT, DOG, DUCK

Single object — Multiple objects

Today

Classification | Classification + Localization | Object Detection | **Segmentation**



Today

Slide Credit:

Semantic Segmentation

Label every pixel!

Don’t differentiate instances (cows)

Classic computer vision problem

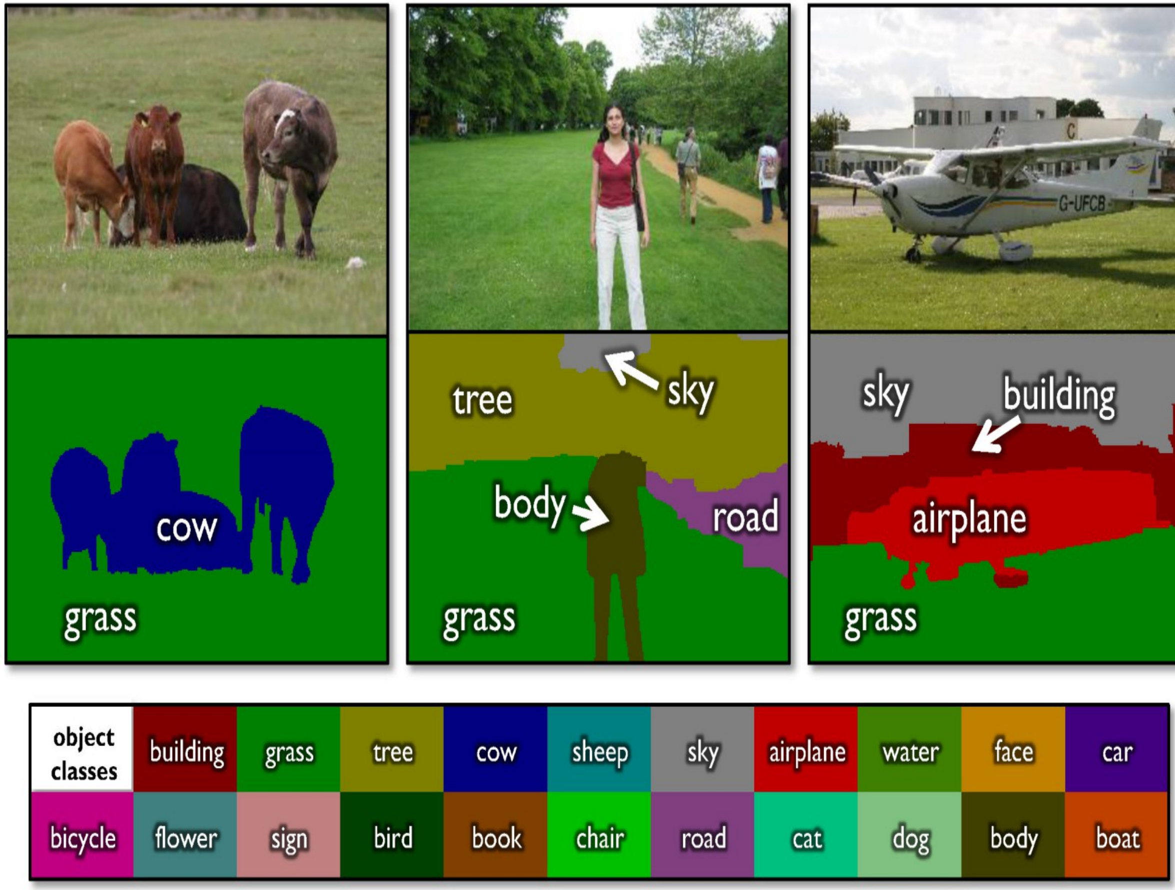


Figure credit: Shotton et al, “TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context”, IJCV 2007

Slide Credit:

Instance Segmentation

Detect instances, give category, label pixels

“simultaneous detection and segmentation” (SDS)

Lots of recent work (MS-COCO)

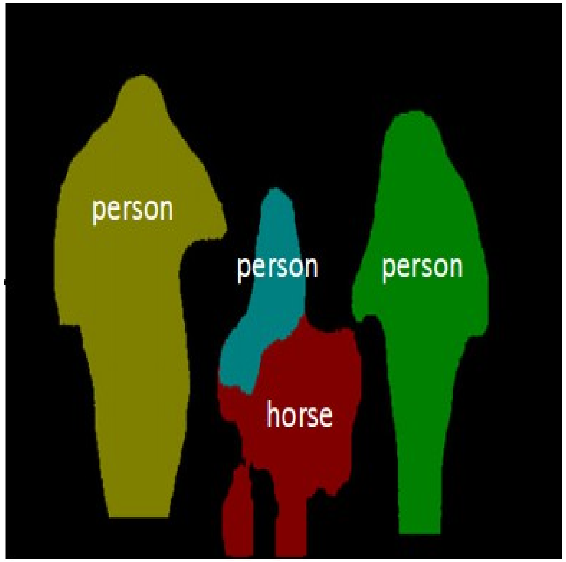
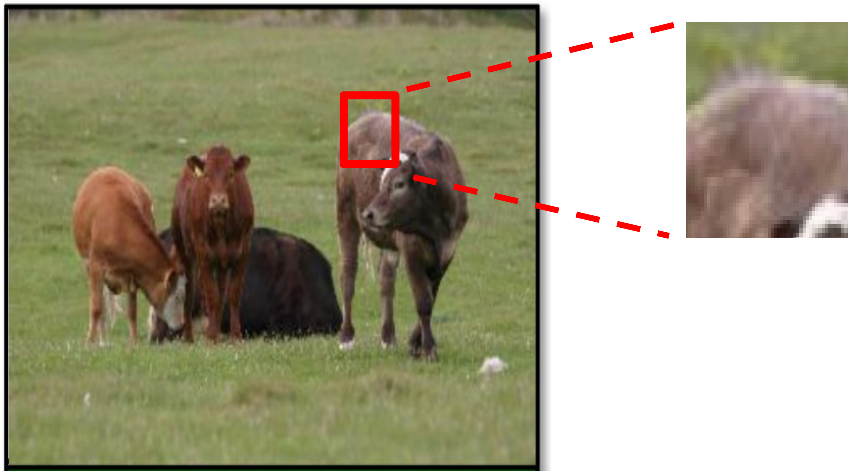


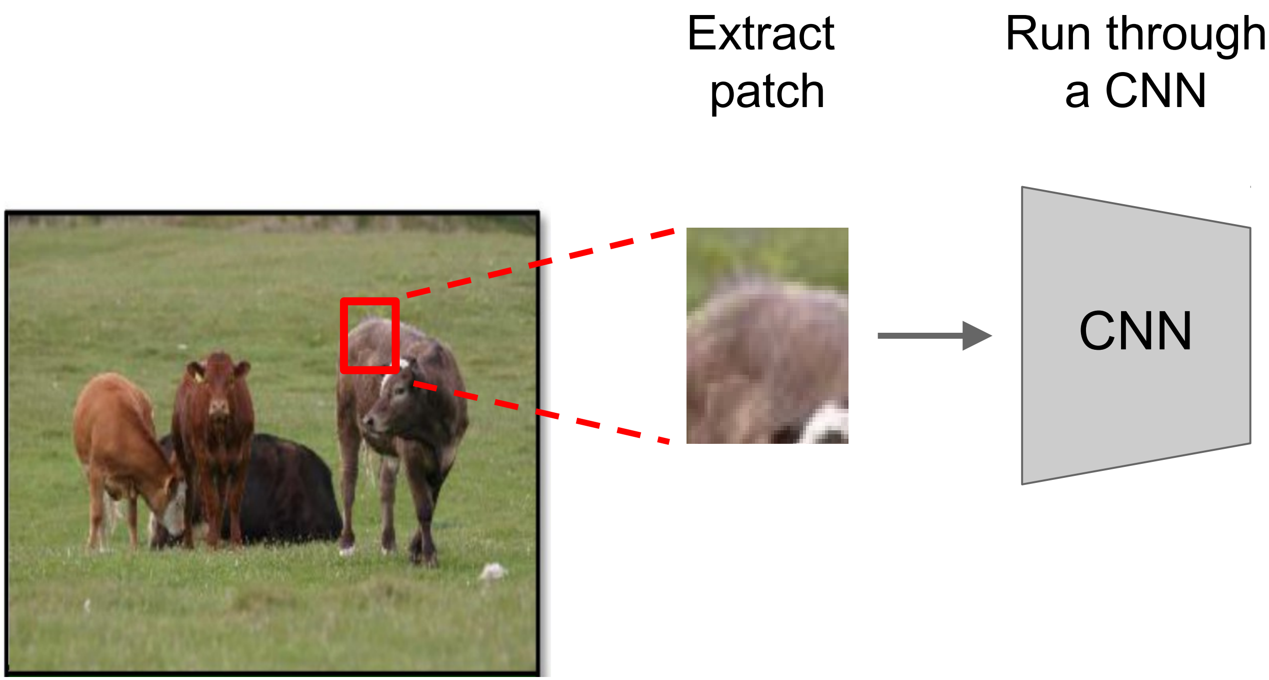
Figure credit: Dai et al, “Instance-aware Semantic Segmentation via Multi-task Network Cascades”, arXiv 2015

Semantic Segmentation

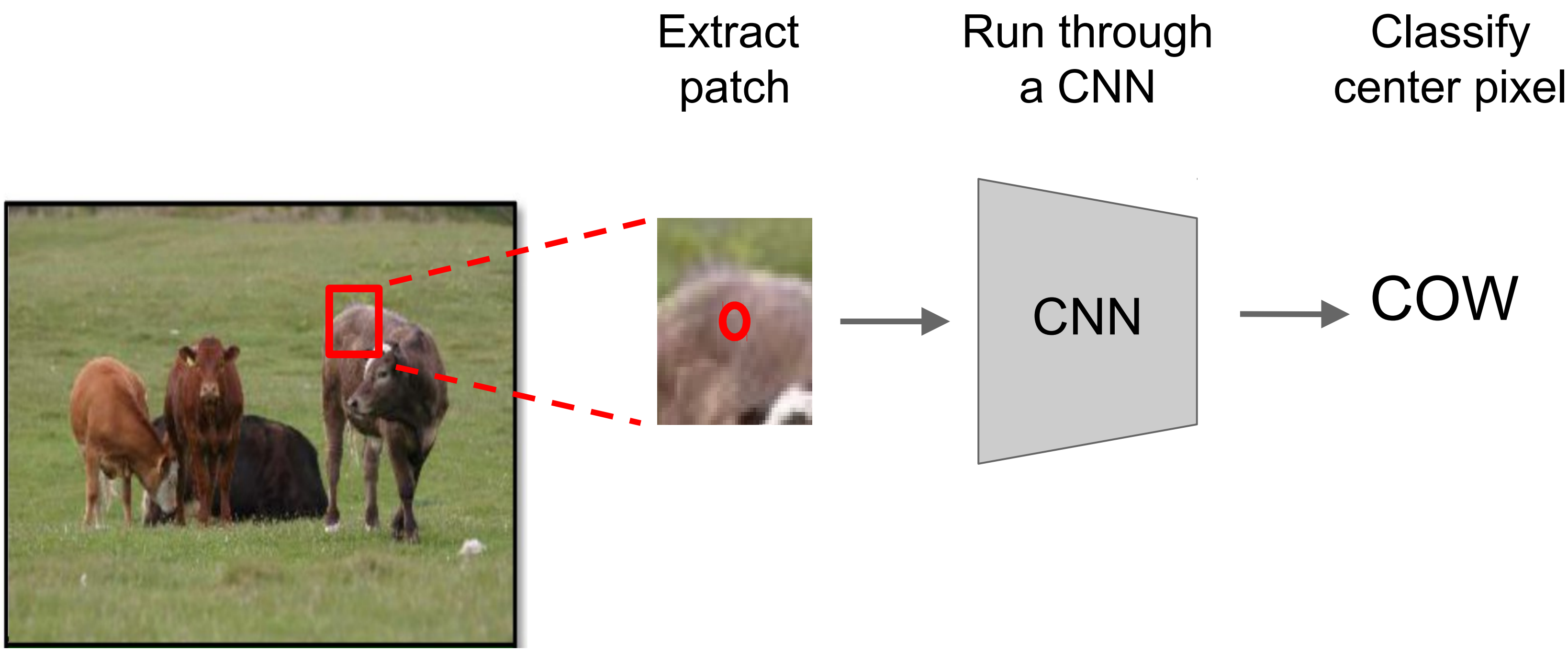
Extract patch



Semantic Segmentation

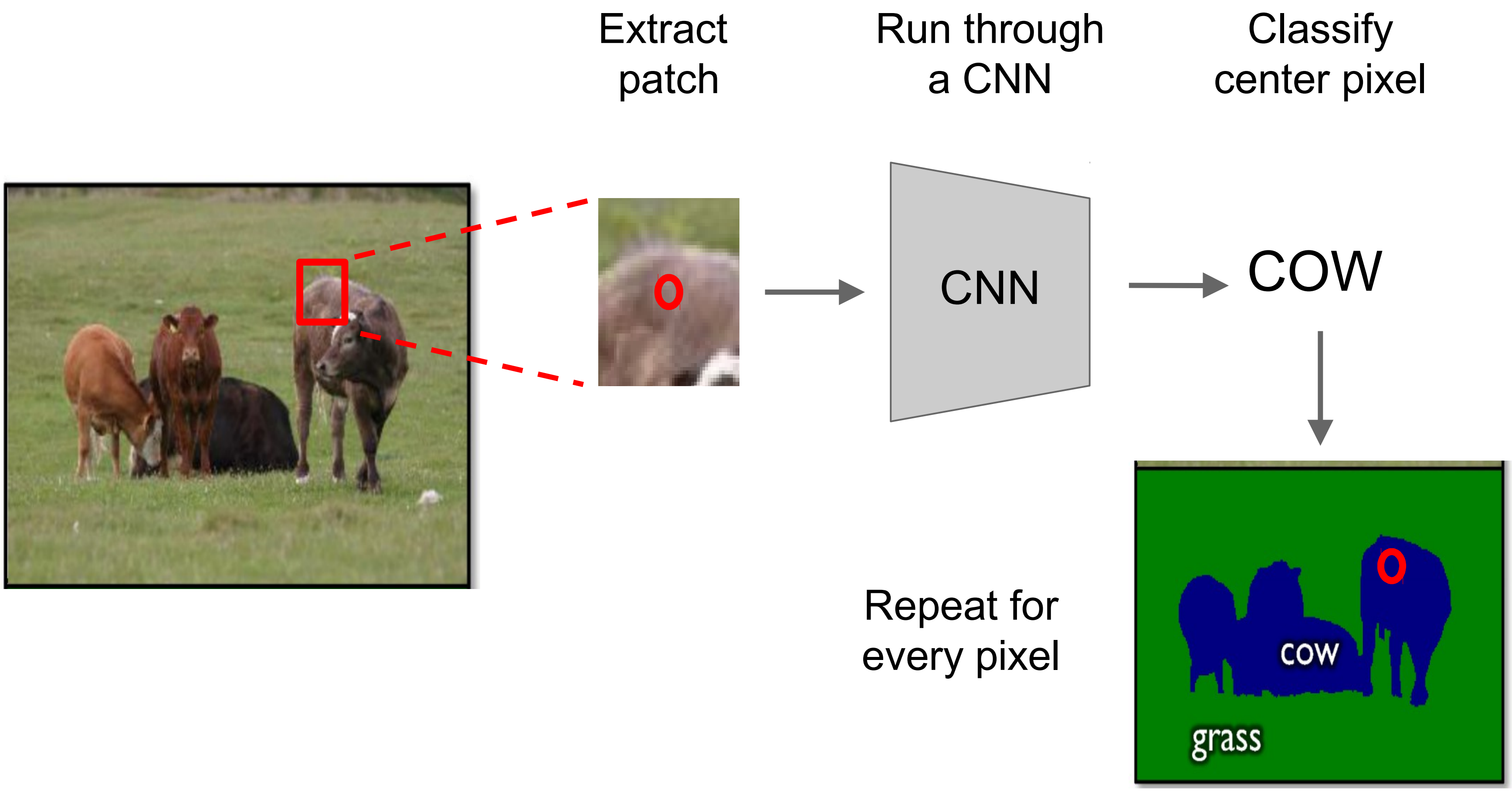


Semantic Segmentation



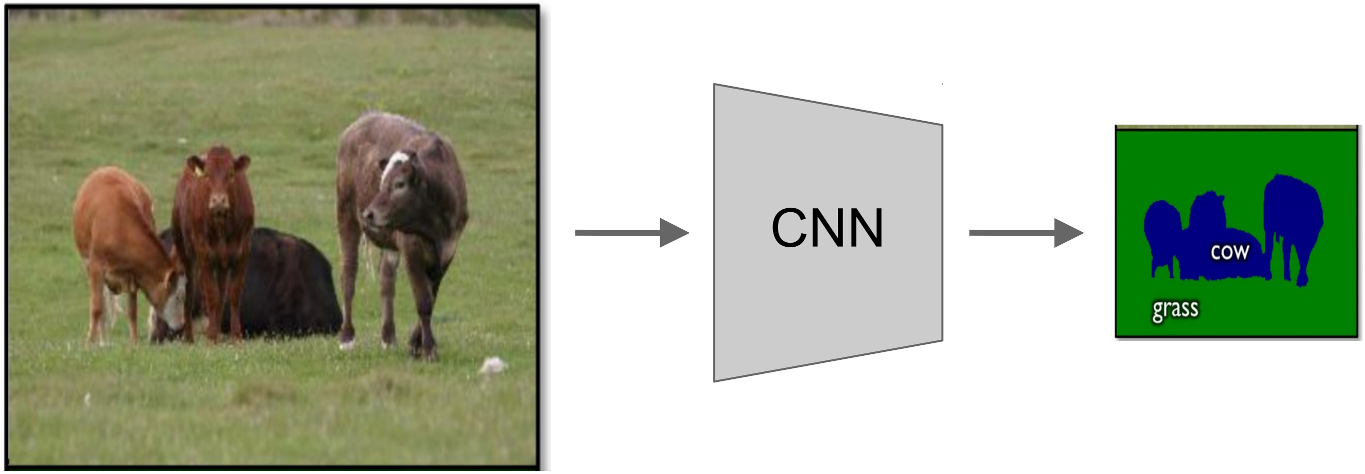
Slide Credit:

Semantic Segmentation



Semantic Segmentation

Run “fully convolutional” network
to get all pixels at once



Smaller output
due to pooling

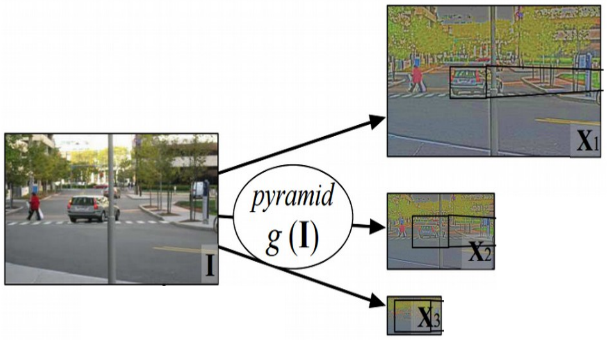
Semantic Segmentation: Multi-Scale

Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013

Slide Credit:

Semantic Segmentation: Multi-Scale

Resize image to
multiple scales

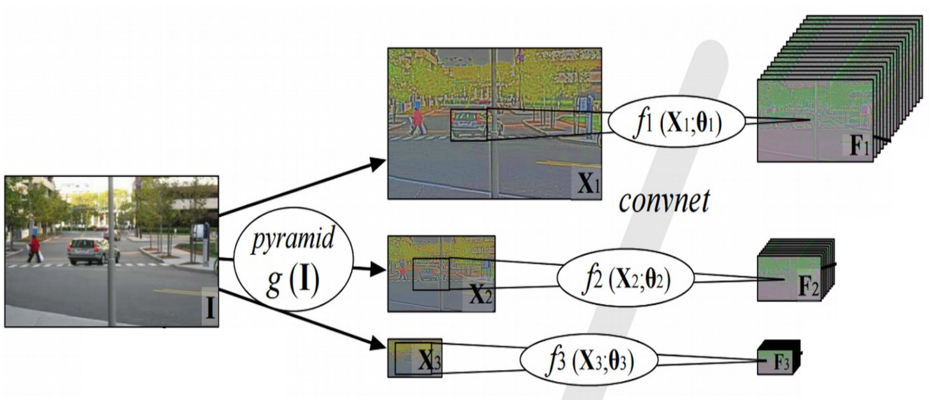


Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013

Semantic Segmentation: Multi-Scale

Resize image to multiple scales

Run one CNN per scale



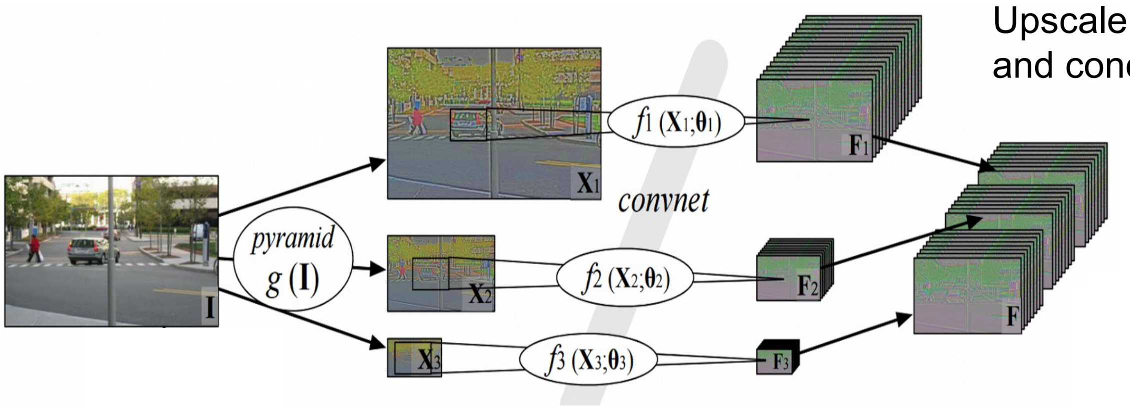
Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013

Semantic Segmentation: Multi-Scale

Resize image to multiple scales

Run one CNN per scale

Upscale outputs and concatenate



Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013

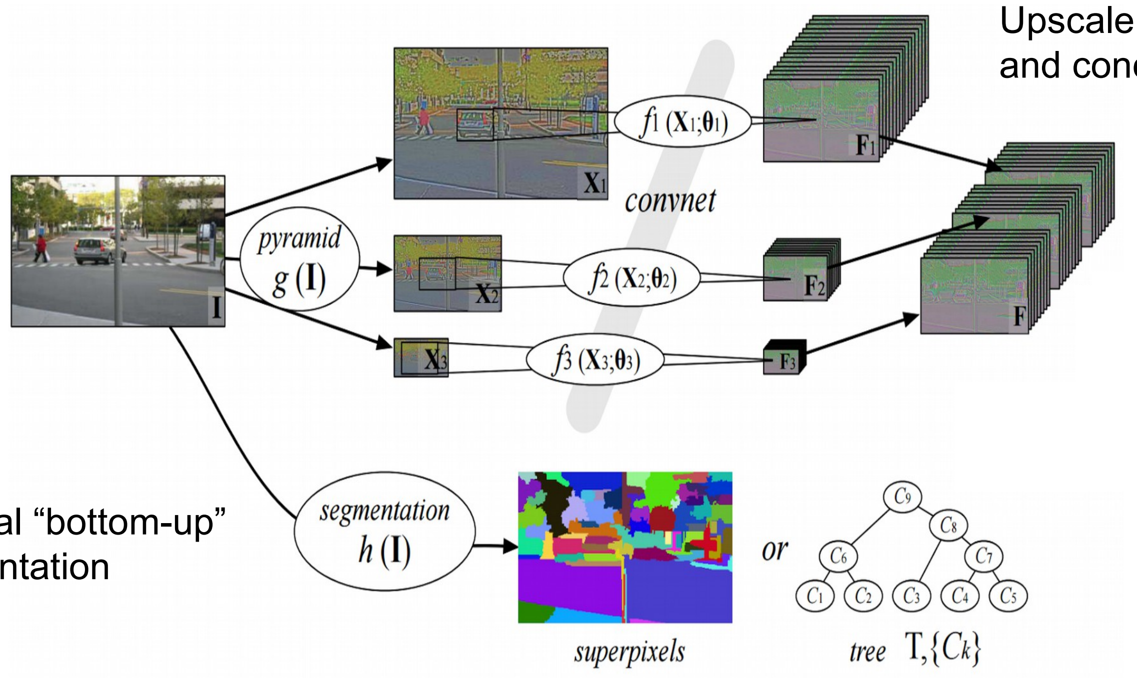
Semantic Segmentation: Multi-Scale

Resize image to multiple scales

Run one CNN per scale

Upscale outputs and concatenate

External “bottom-up” segmentation



Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013

Slide Credit:

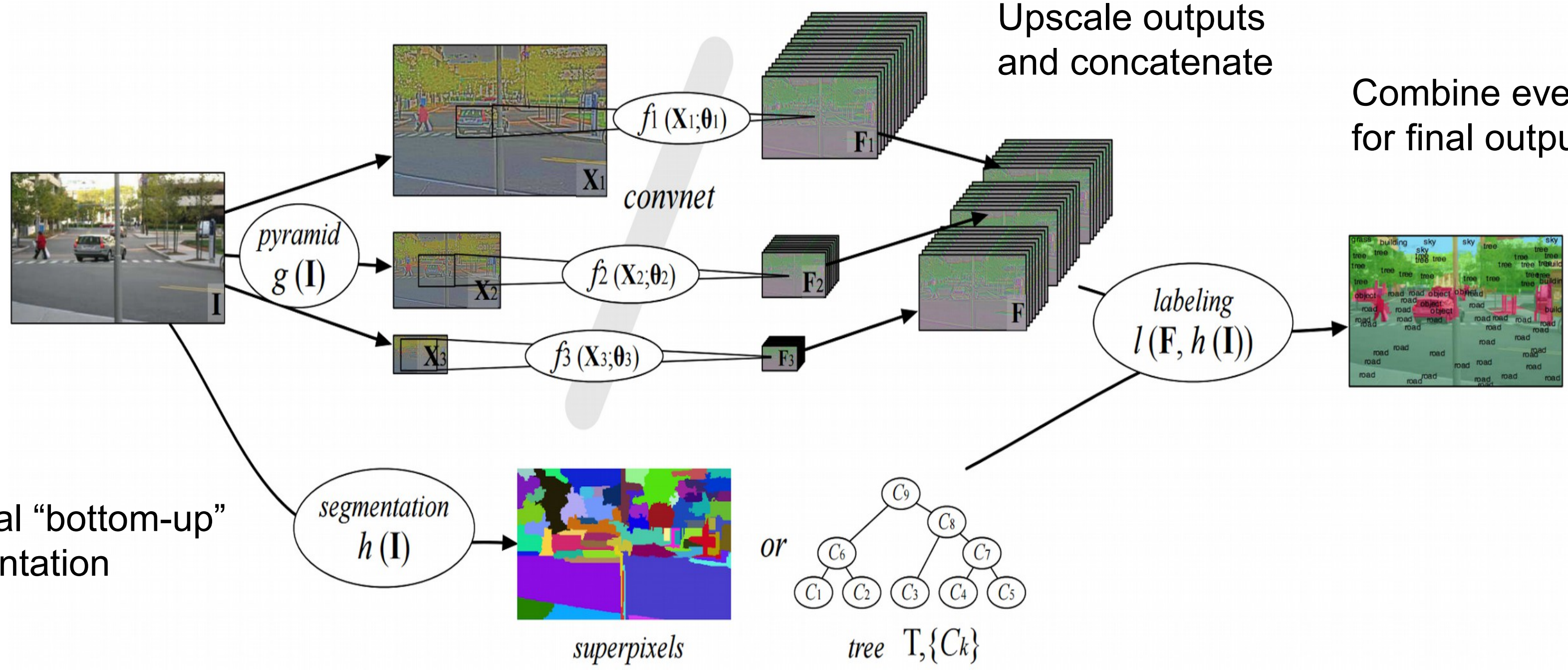
Semantic Segmentation: Multi-Scale

Resize image to multiple scales

Run one CNN per scale

Upscale outputs and concatenate

Combine everything for final outputs



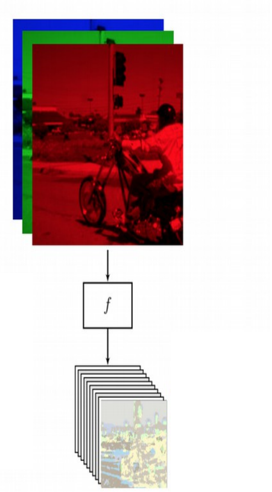
External “bottom-up” segmentation

Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013

Slide Credit:

Semantic Segmentation: Refinement

Apply CNN once to get labels



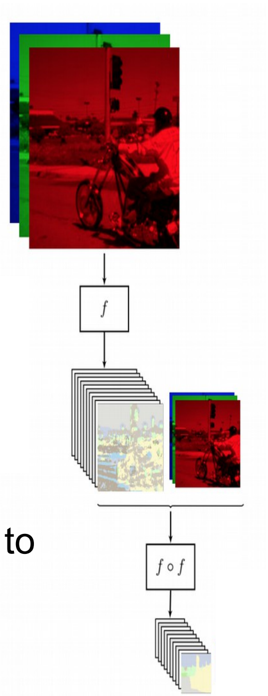
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Slide Credit:

Semantic Segmentation: Refinement

Apply CNN once
to get labels

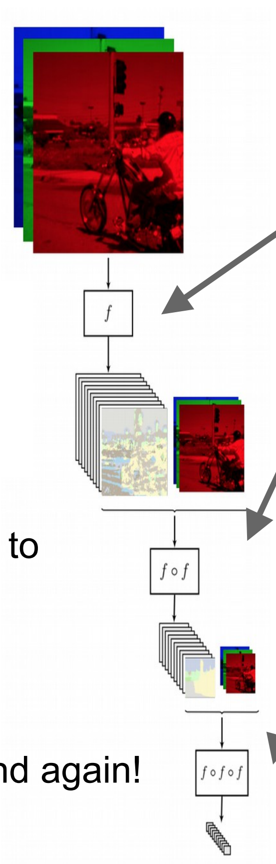
Apply AGAIN to
refine labels



Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Semantic Segmentation: Refinement

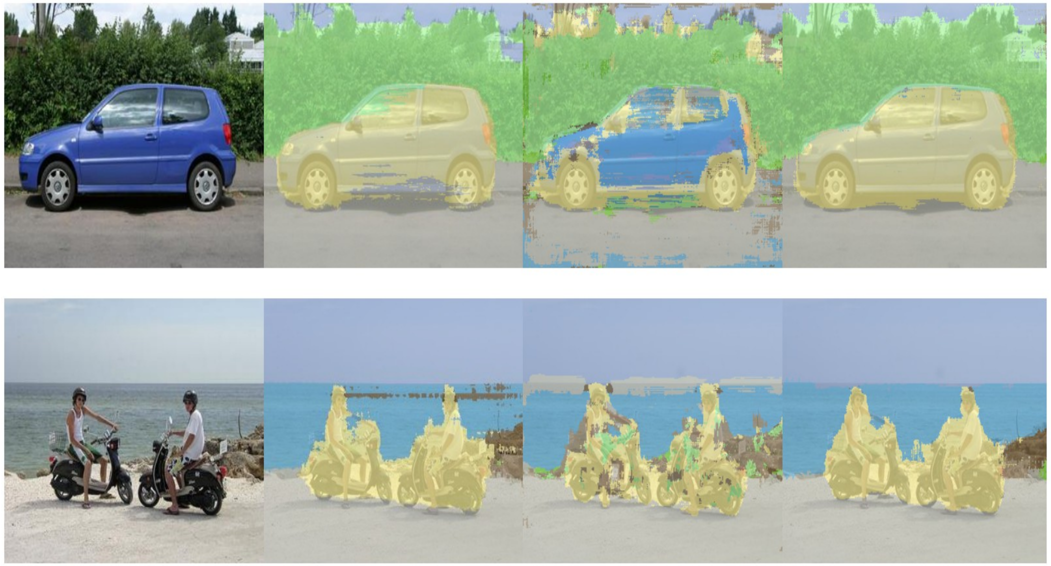
Same CNN weights:
recurrent convolutional network



Apply CNN once
to get labels

Apply AGAIN to
refine labels

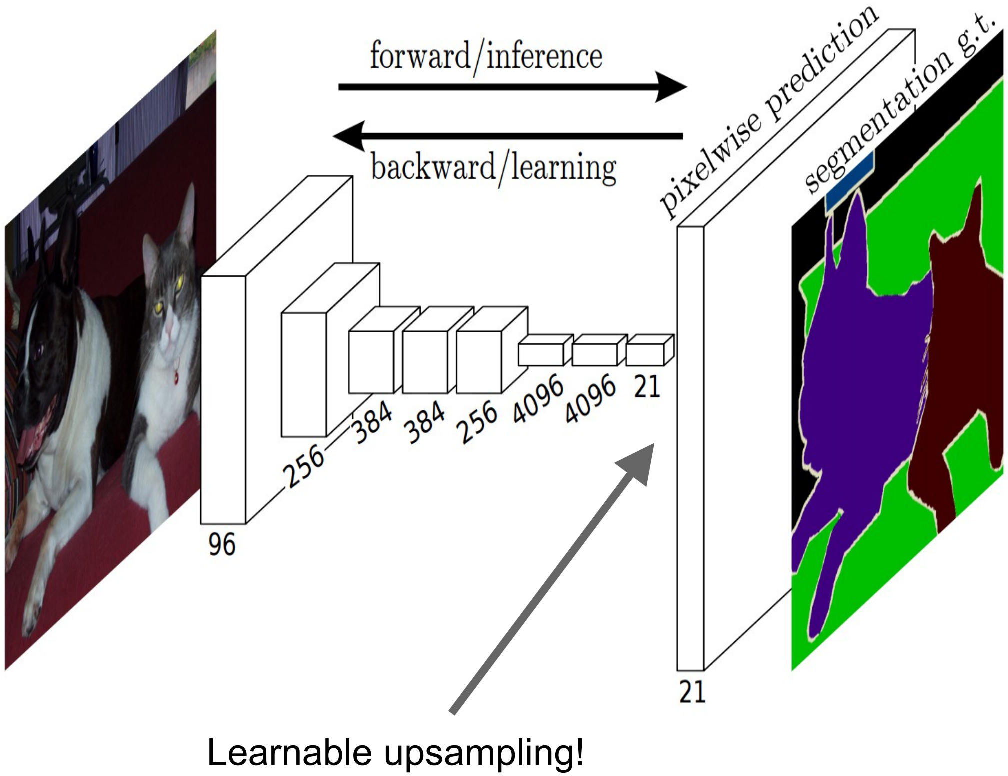
And again!



More iterations improve results

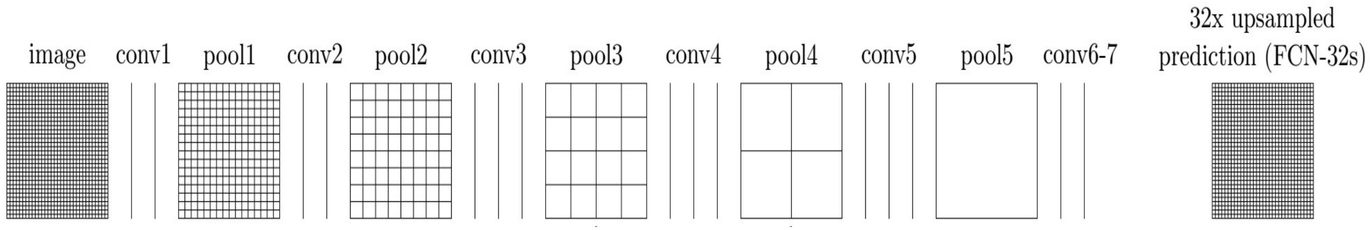
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Semantic Segmentation: Upsampling



Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

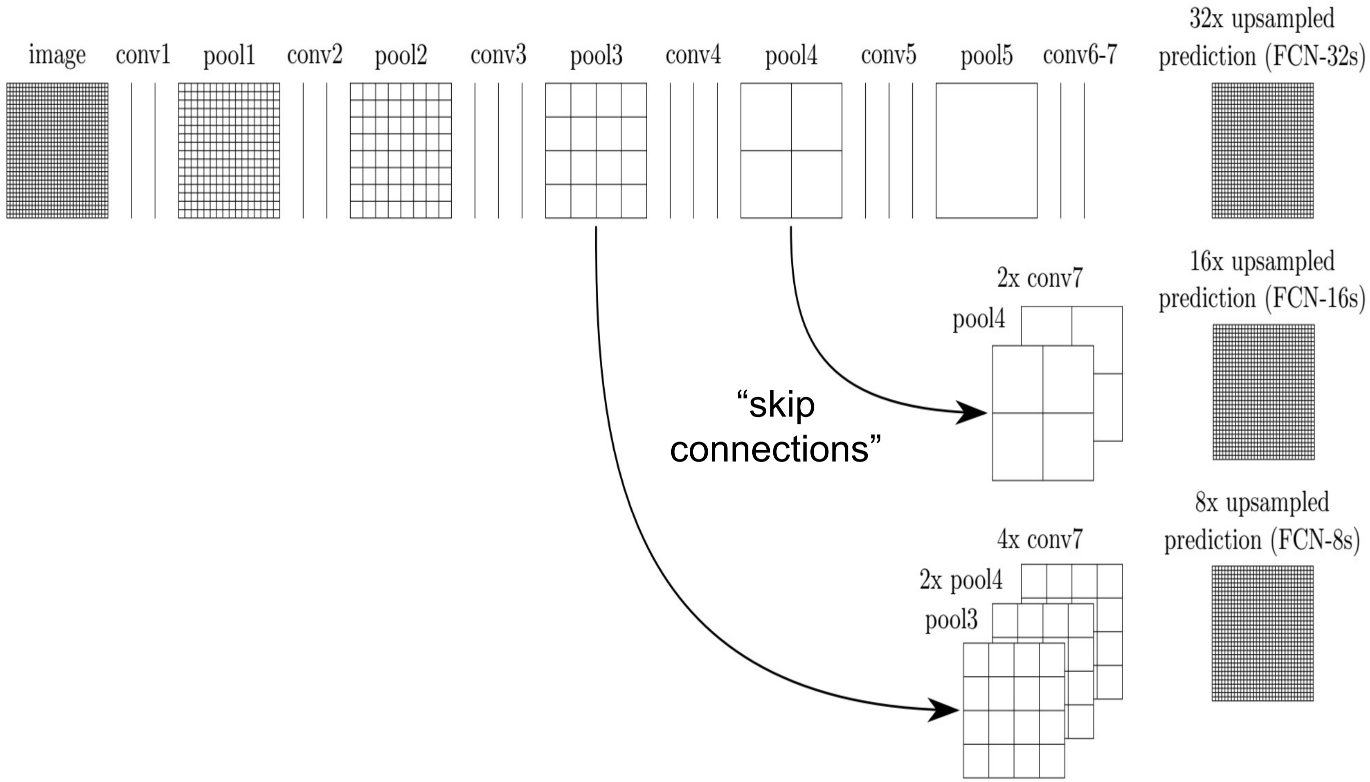
Semantic Segmentation: Upsampling



Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

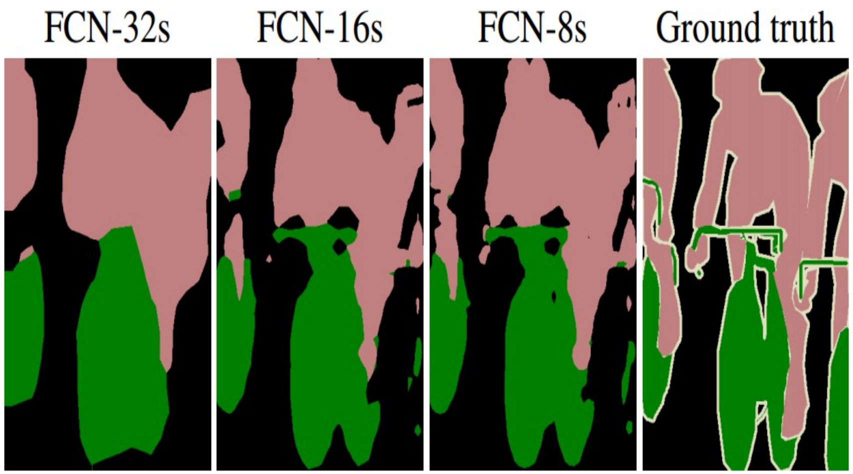
Slide Credit:

Semantic Segmentation: Upsampling

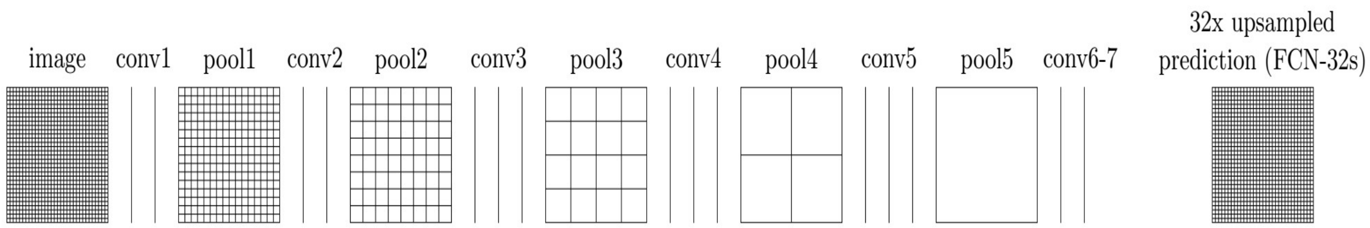


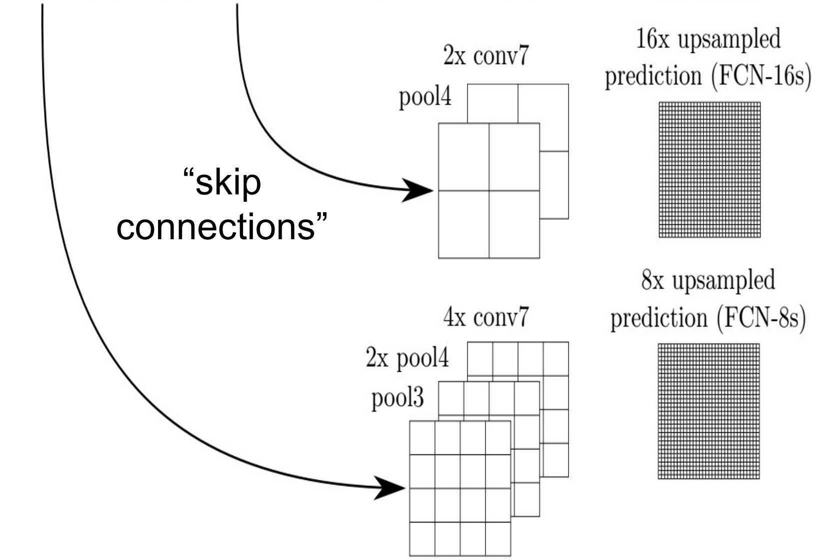
Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

Semantic Segmentation: Upsampling



Skip connections = Better results

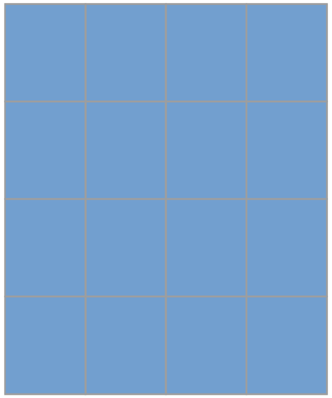




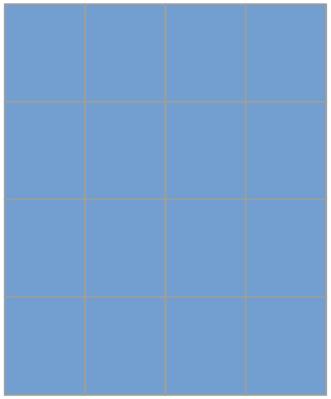
Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 1 pad 1



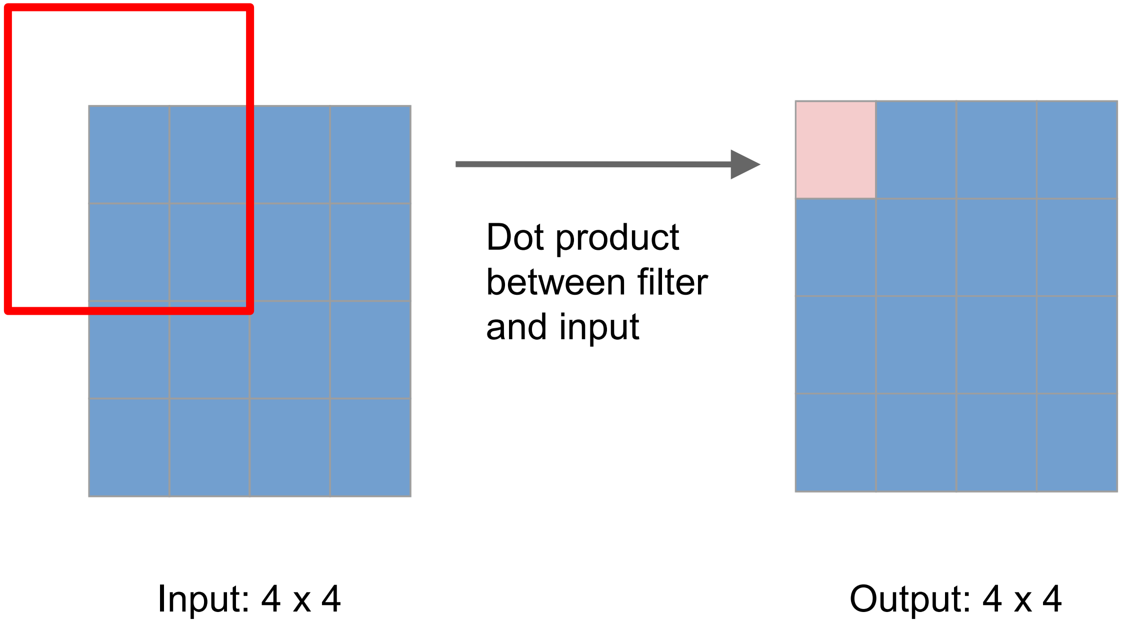
Input: 4 x 4



Output: 4 x 4

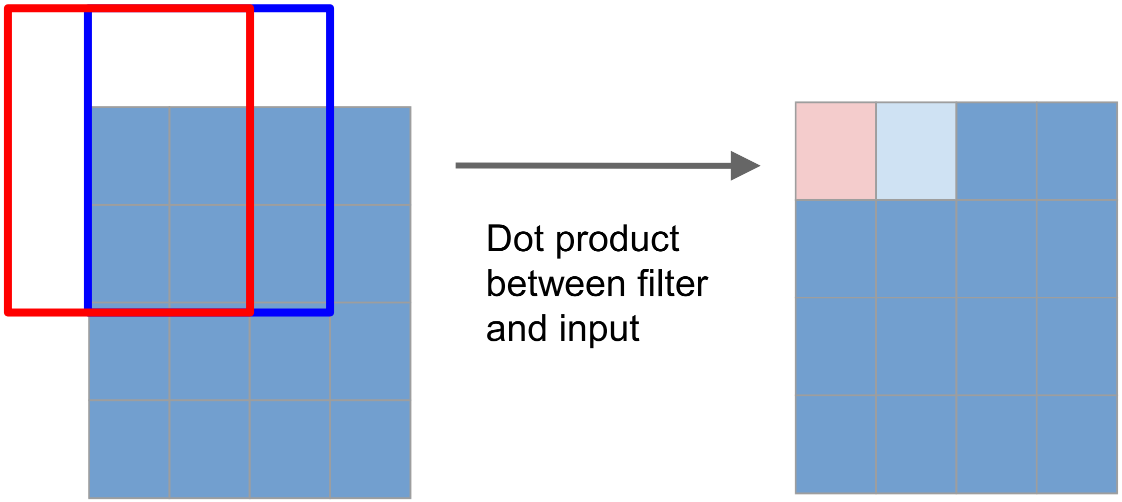
Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 1 pad 1



Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 1 pad 1

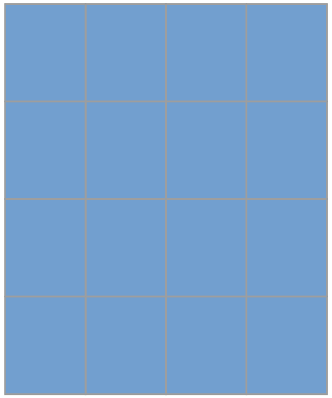


Input: 4 x 4

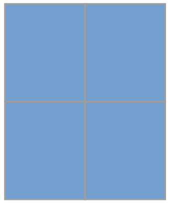
Output: 4 x 4

Learnable Upsampling: "Deconvolution"

Typical 3 x 3 convolution, **stride 2** pad 1



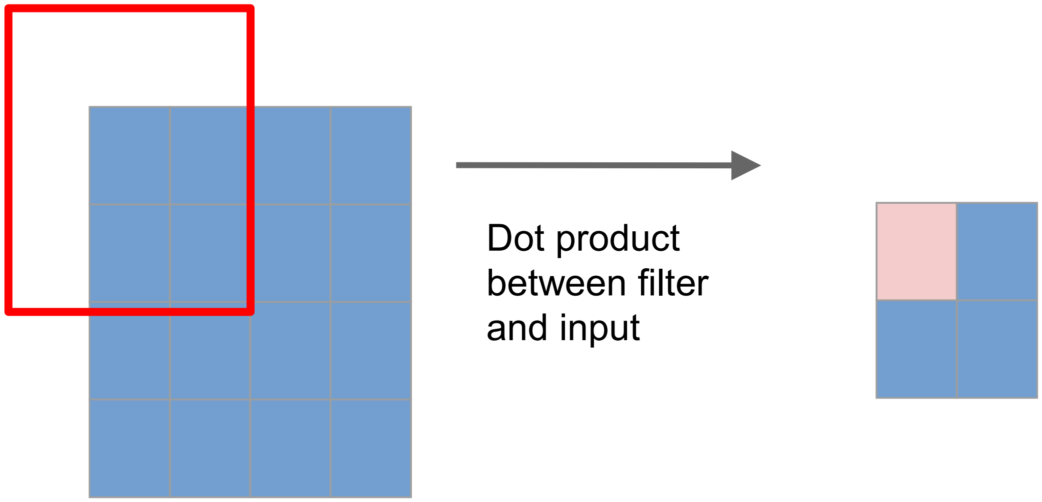
Input: 4 x 4



Output: 2 x 2

Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 2 pad 1



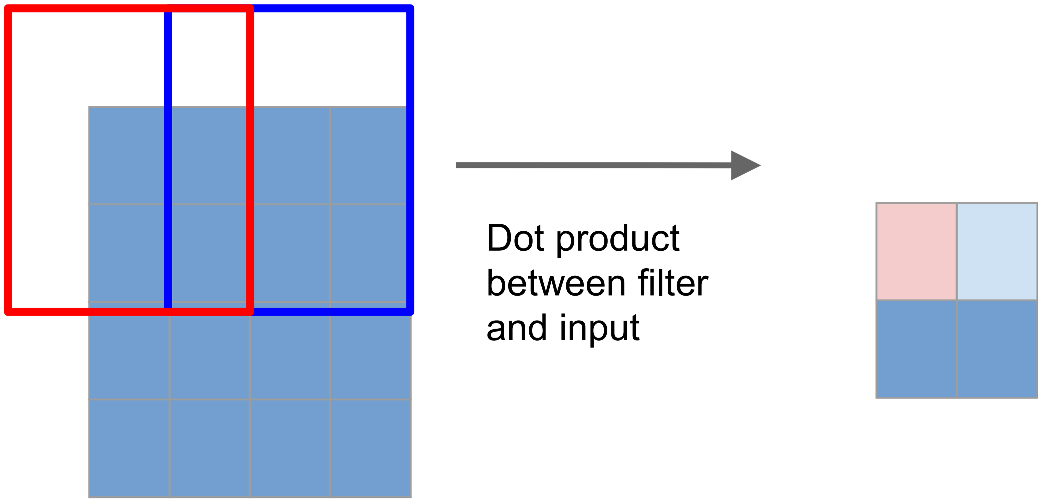
Input: 4 x 4

Output: 2 x 2

Slide Credit:

Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 2 pad 1



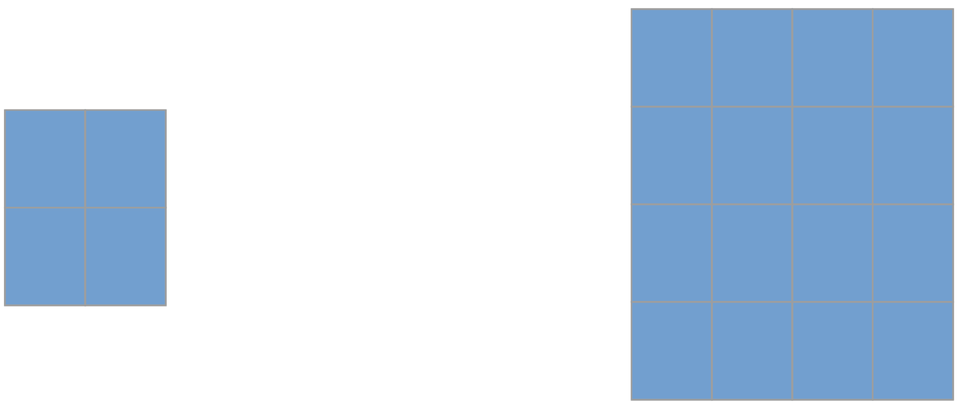
Input: 4 x 4

Output: 2 x 2

Slide Credit:

Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1



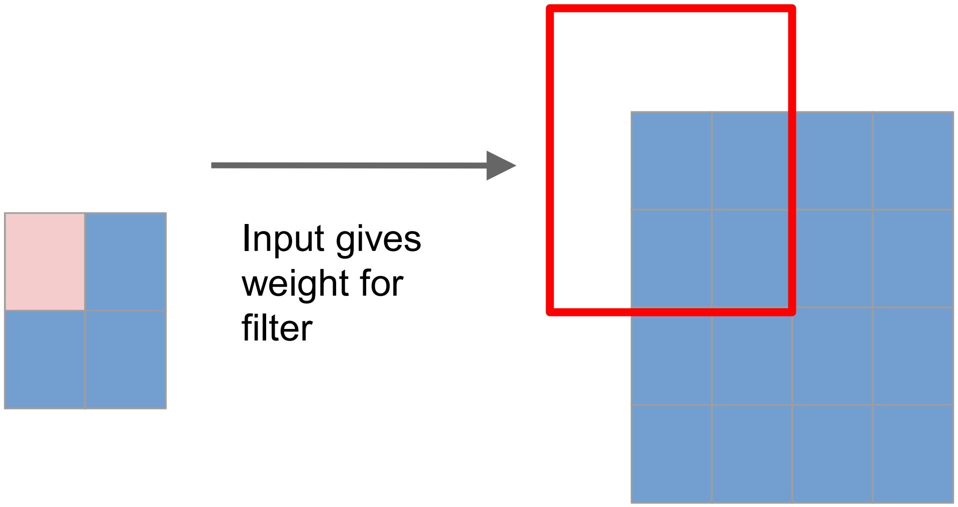
Input: 2 x 2

Output: 4 x 4

Slide Credit:

Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1



Input: 2 x 2

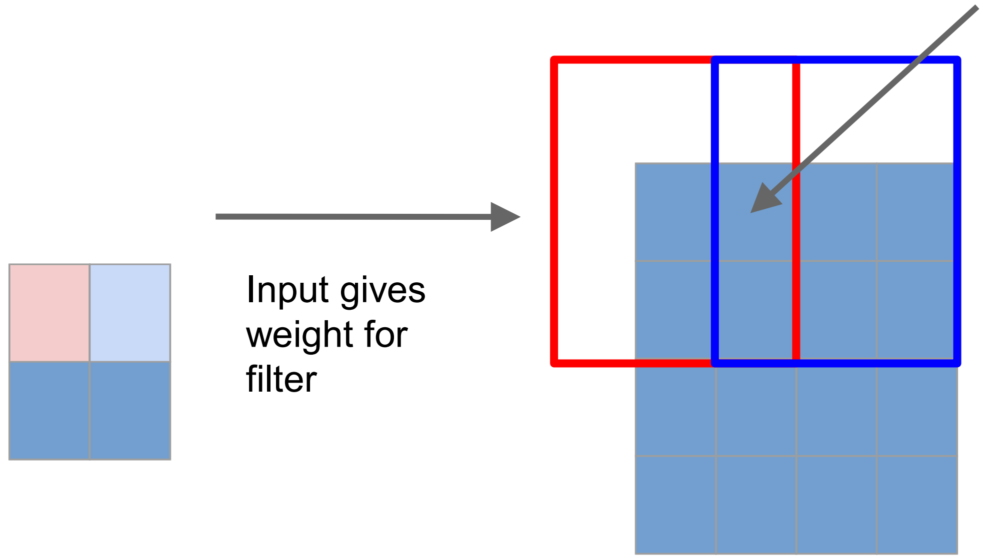
Output: 4 x 4

Slide Credit:

Learnable Upsampling: “Deconvolution”

3 x 3 “deconvolution”, stride 2 pad 1

Sum where
output overlaps



Same as backward pass for normal convolution!

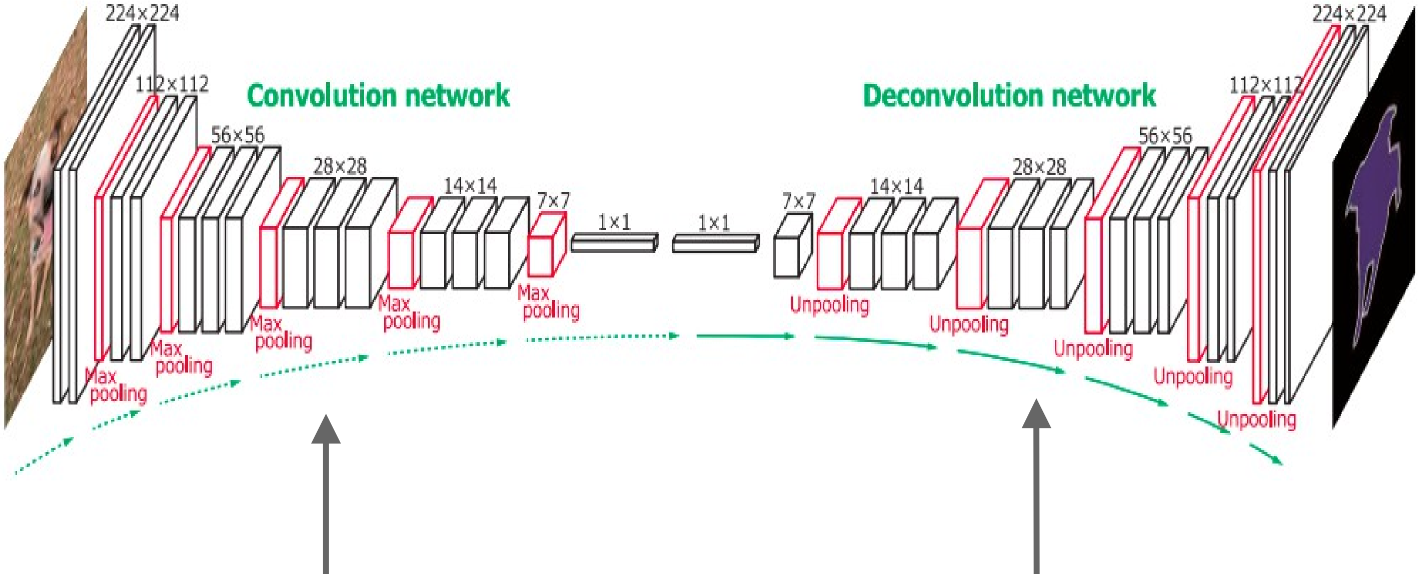
“Deconvolution” is a bad name, already defined as “inverse of convolution”

Better names:
convolution transpose,
backward strided convolution,
1/2 strided convolution,
upconvolution

Input: 2 x 2

Output: 4 x 4

Semantic Segmentation: Upsampling



Normal VGG

“Upside down” VGG

Noh et al, “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

6 days of training on Titan X…

Slide Credit:

Instance Segmentation

Detect instances, give category, label pixels

"simultaneous detection and segmentation" (SDS)

Lots of recent work (MS-COCO)

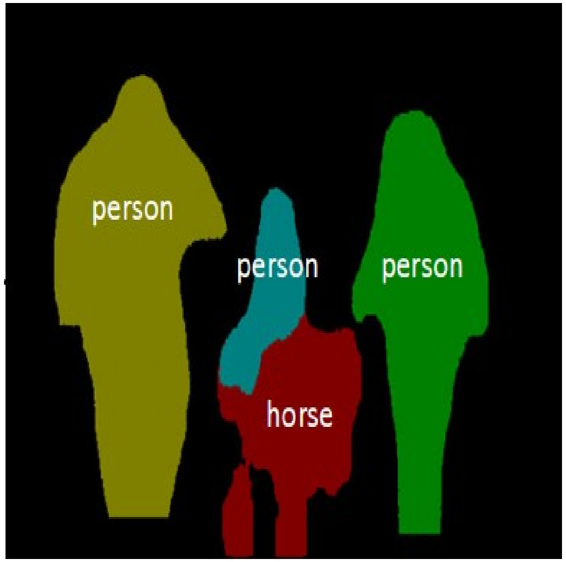


Figure credit: Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Instance Segmentation

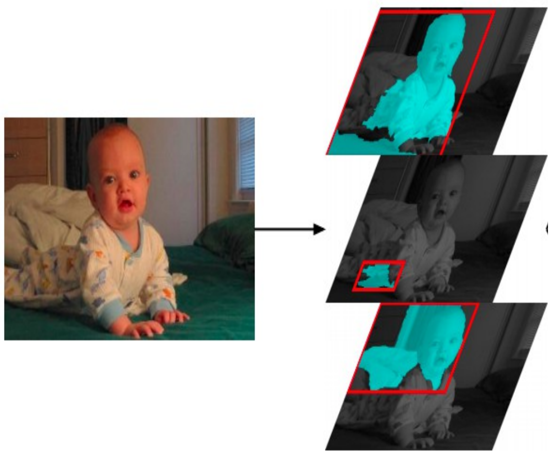
Similar to R-CNN, but with segments

Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

Slide Credit:

Instance Segmentation

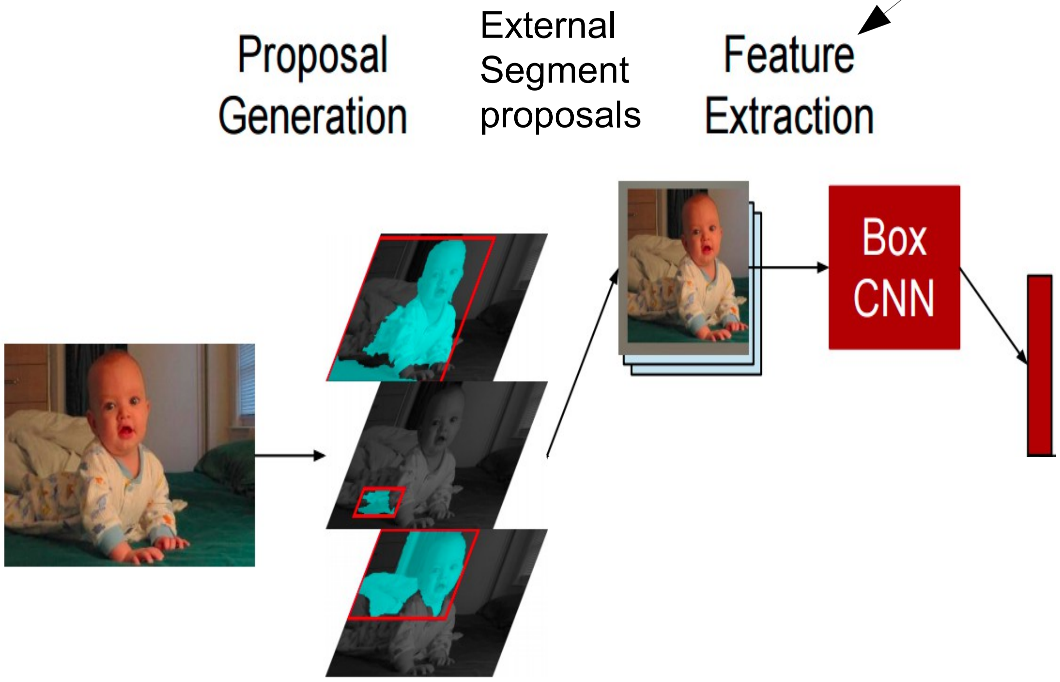
Similar to R-CNN, but with segments



Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

Instance Segmentation

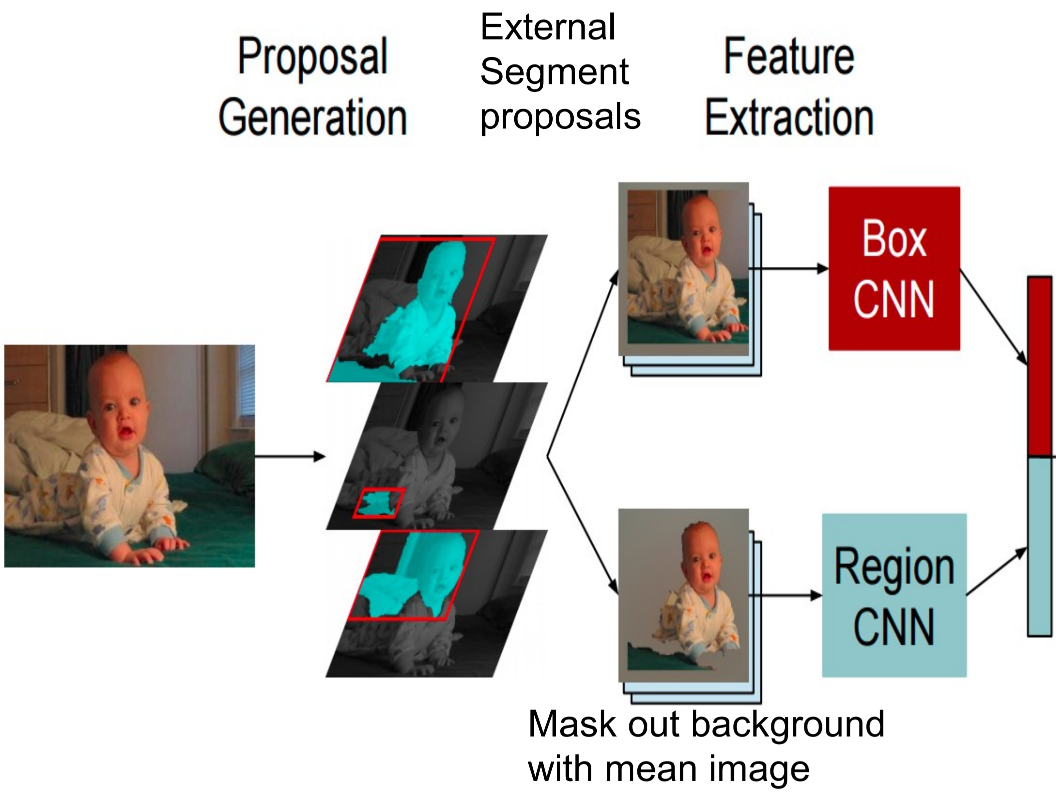
Similar to R-CNN



Hariharan et al, “Simultaneous Detection and Segmentation”, ECCV 2014

Instance Segmentation

Similar to R-CNN, but with segments

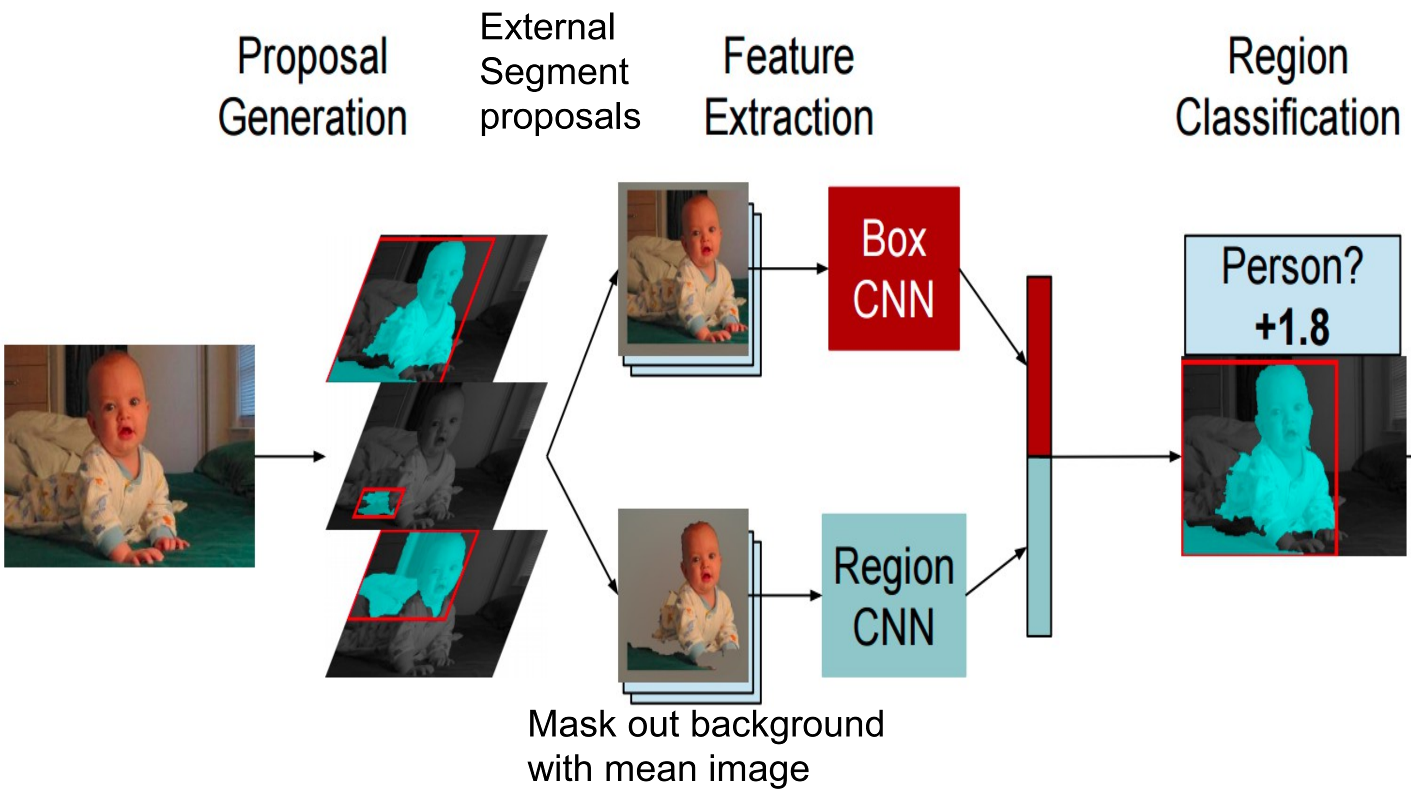


Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

Slide Credit:

Instance Segmentation

Similar to R-CNN, but with segments

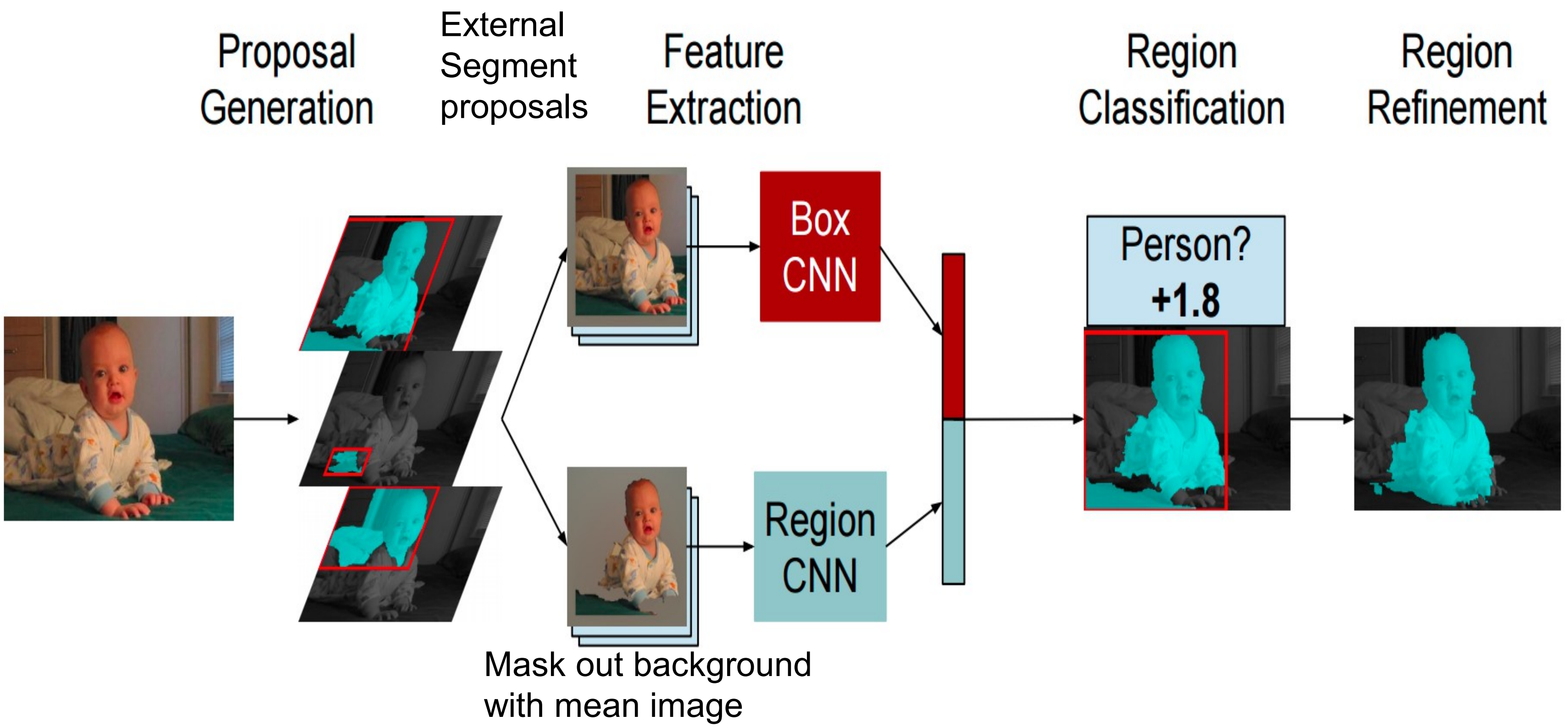


Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

Slide Credit:

Instance Segmentation

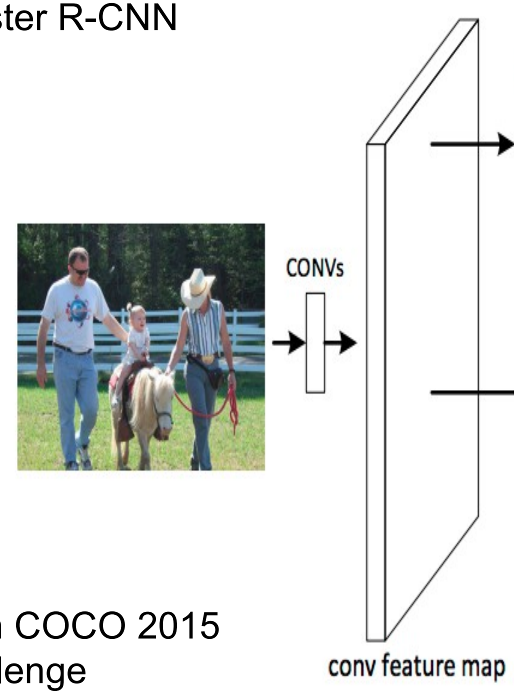
Similar to R-CNN, but with segments



Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

Instance Segmentation: Cascades

Similar to
Faster R-CNN



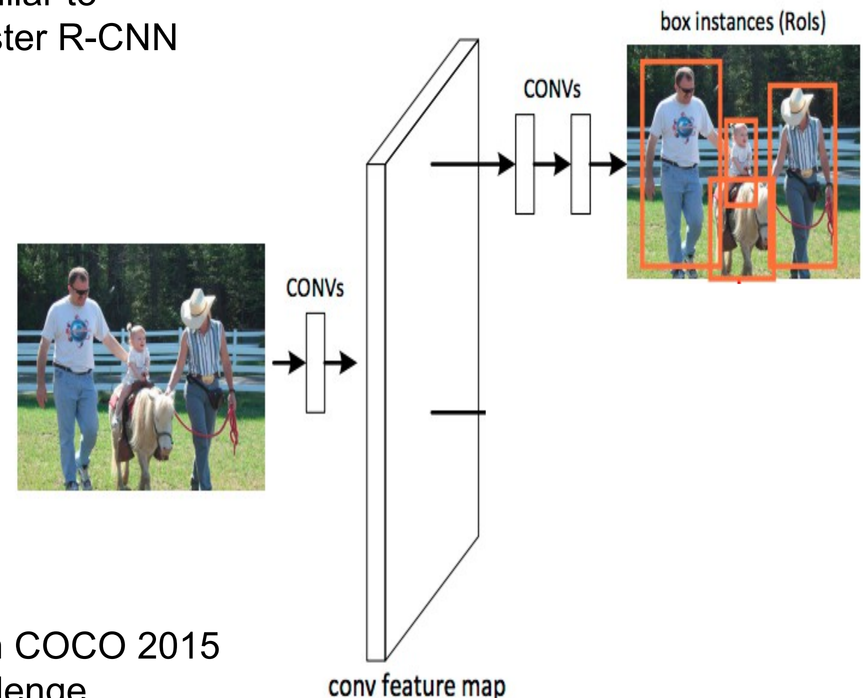
Won COCO 2015
challenge
(with ResNet)

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Instance Segmentation: Cascades

Similar to
Faster R-CNN

Region proposal network (RPN)



Won COCO 2015
challenge
(with ResNet)

Dai et al, “Instance-aware Semantic Segmentation via Multi-task Network Cascades”, arXiv 2015

Slide Credit:

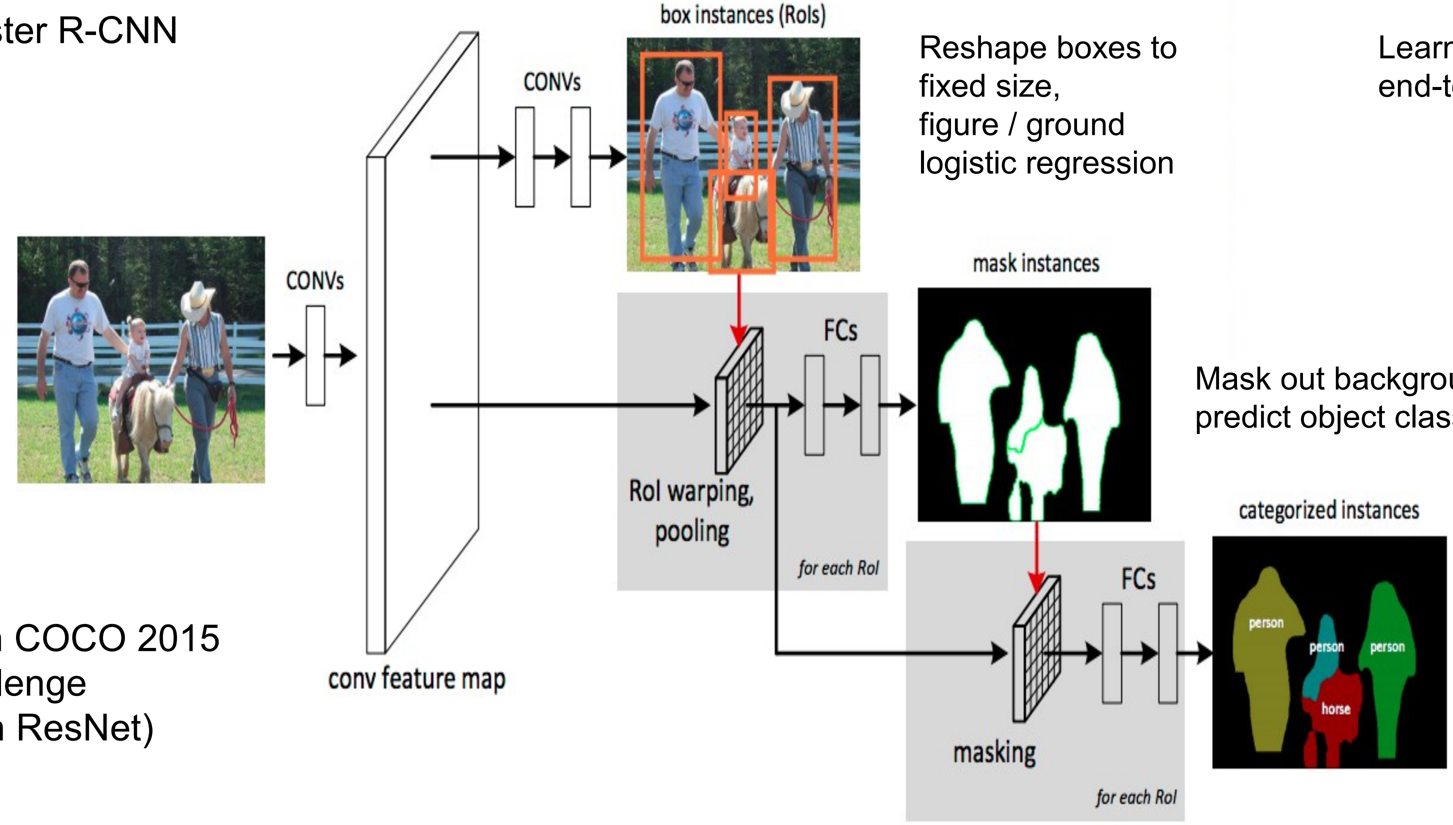
Instance Segmentation: Cascades

Similar to
Faster R-CNN

Region proposal network (RPN)

Reshape boxes to
fixed size,
figure / ground
logistic regression

Learn entire model
end-to-end!



Mask out background,
predict object class

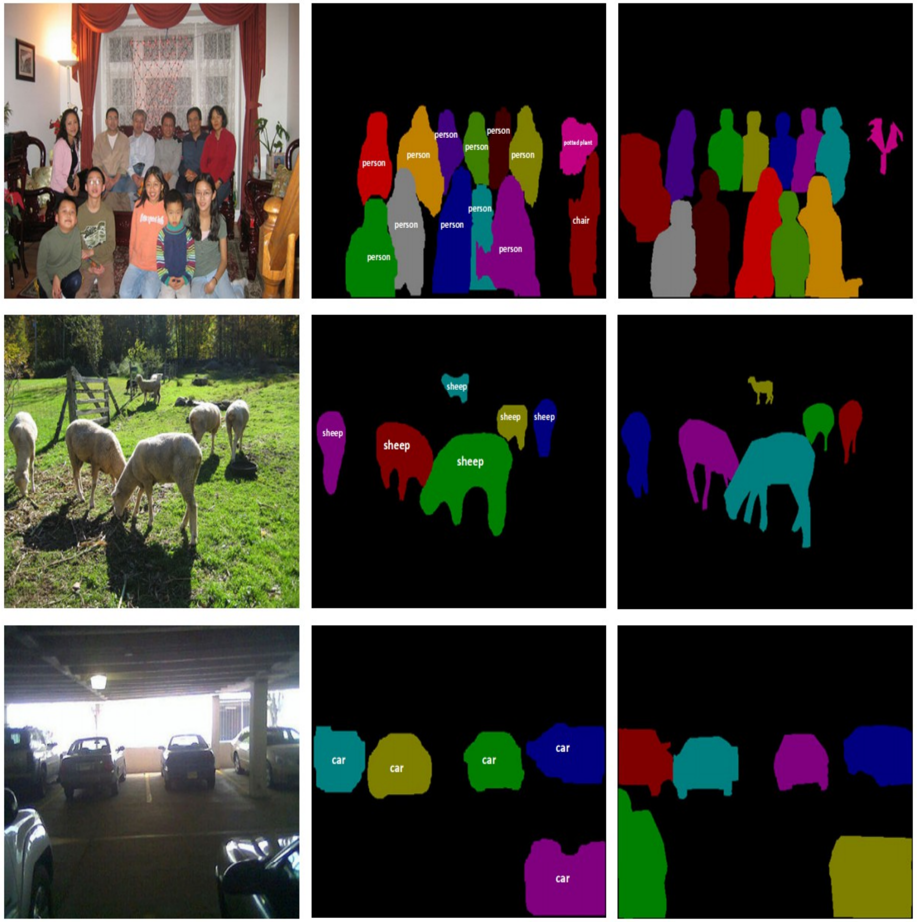
Won COCO 2015
challenge
(with ResNet)

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

66

Slide Credit:

Instance Segmentation: Cascades



Predictions Ground truth

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Slide Credit:

Segmentation Overview

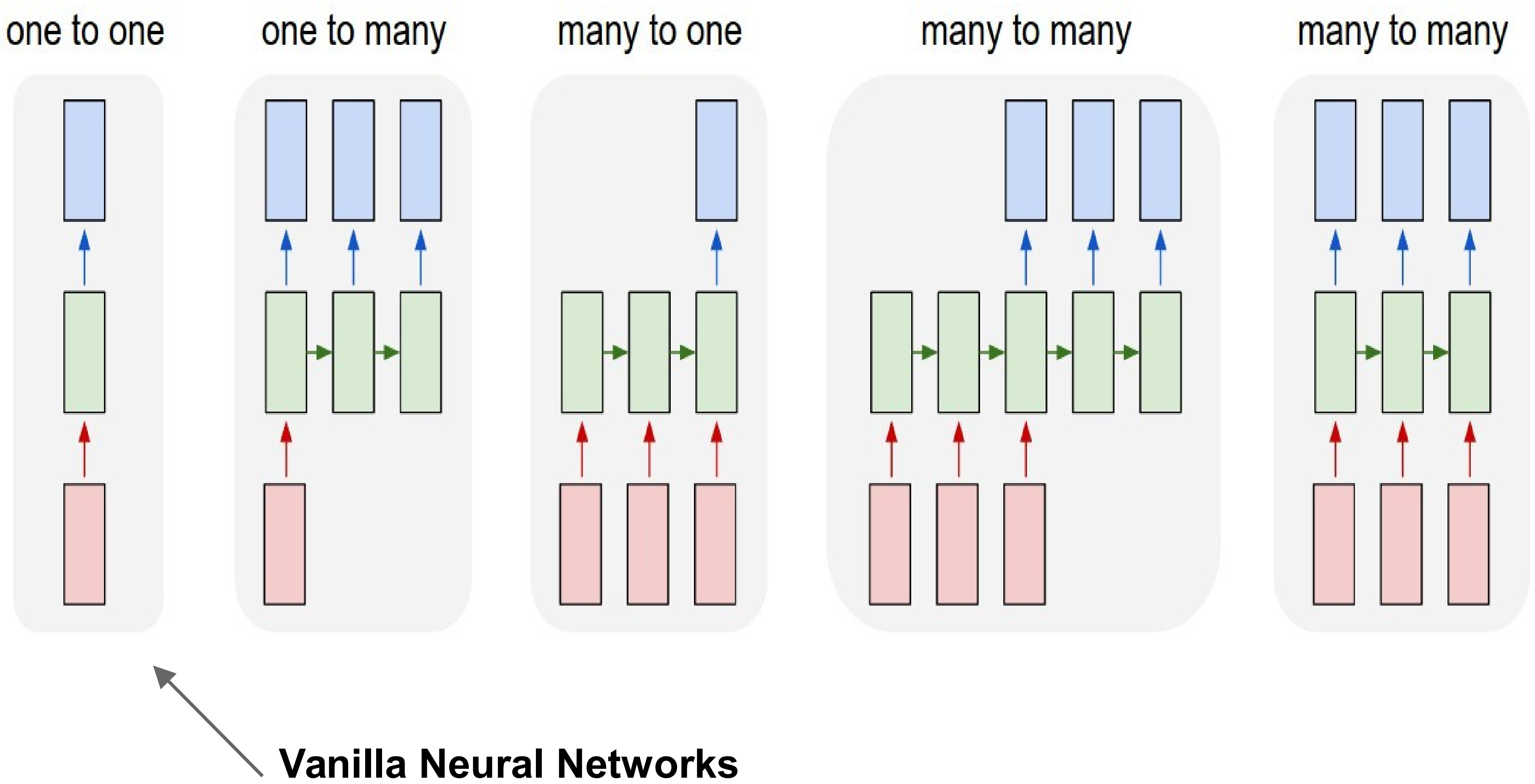
Semantic segmentation
- Classify all pixels
- Fully convolutional models, downsample then upsample
- Learnable upsampling: fractionally strided convolution
- Skip connections can help

Instance Segmentation
- Detect instance, generate mask
- Similar pipelines to object detection

Slide Credit:

Quick overview of Other Topics

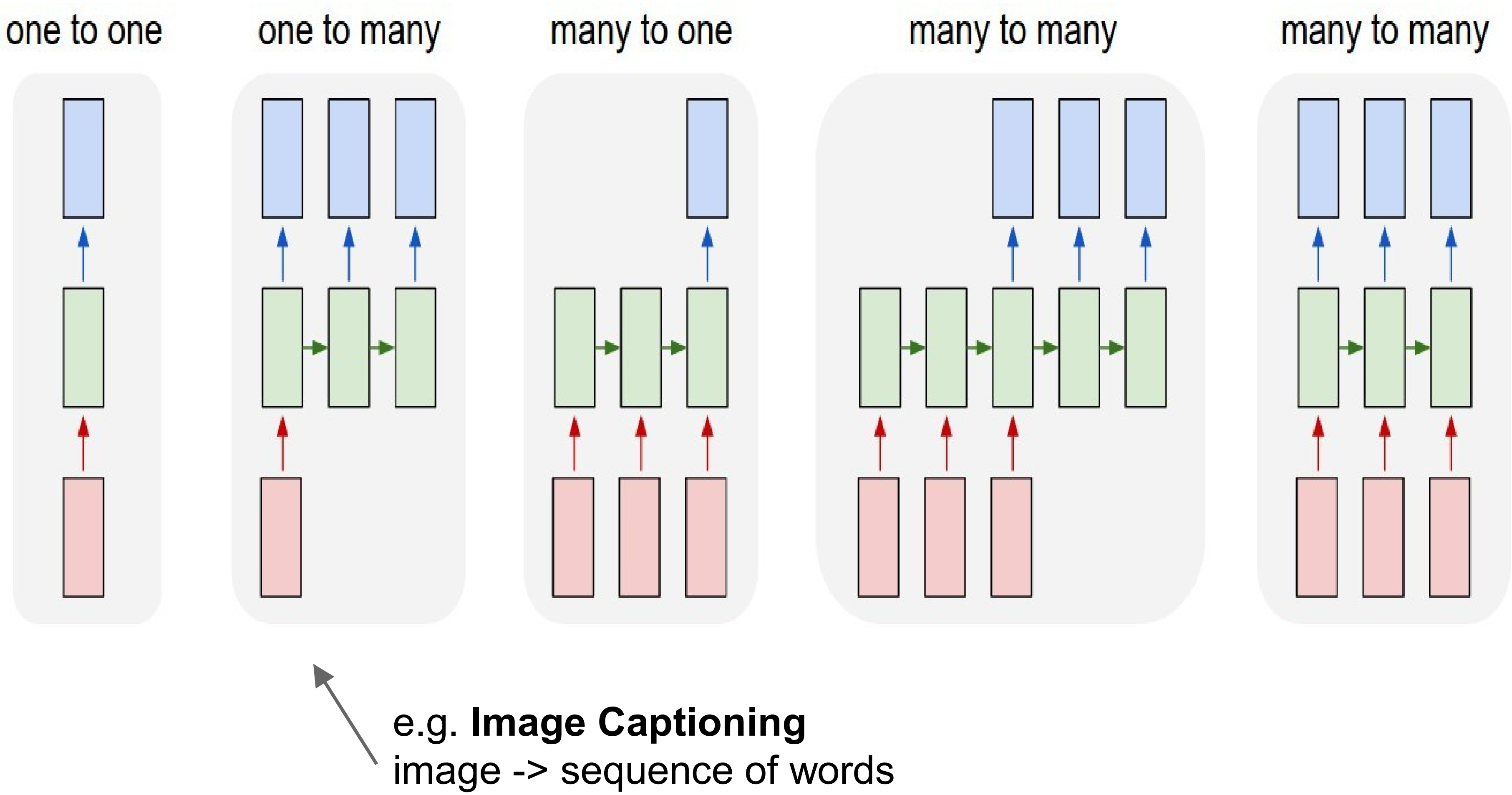
Recurrent Neural Networks (RNN)



Vanilla Neural Networks

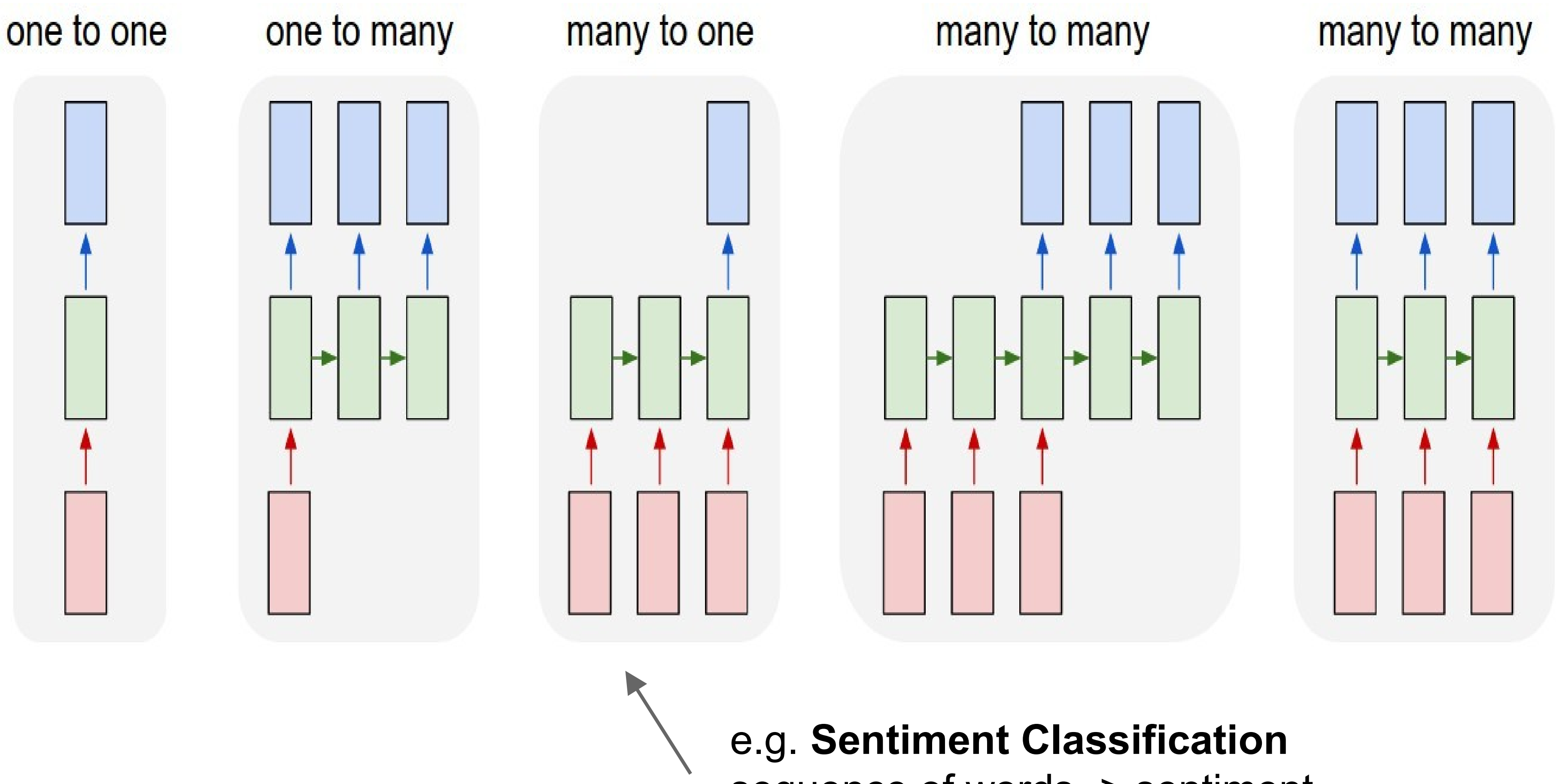
Slide Credit:

Recurrent Neural Networks (RNN)



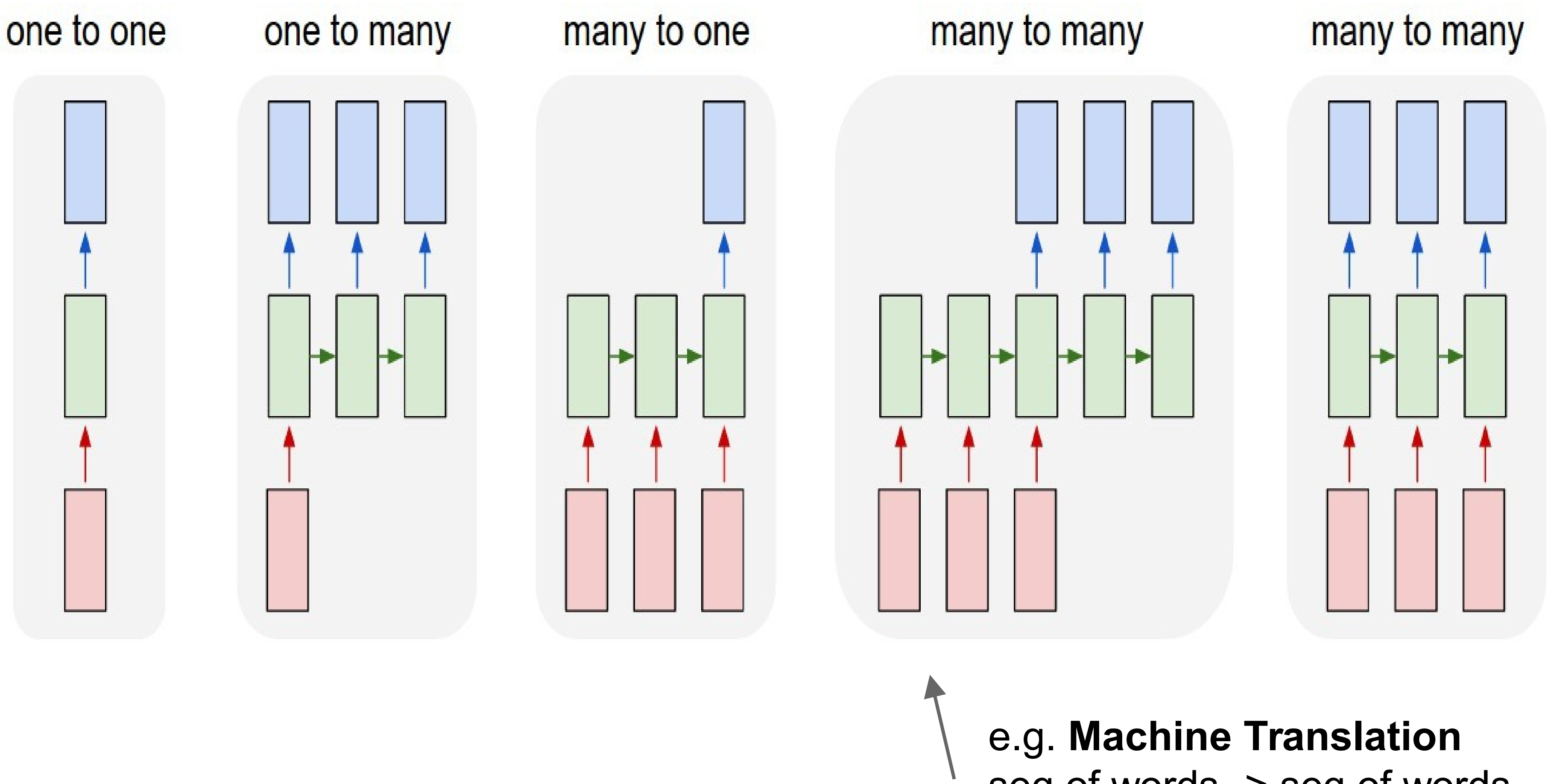
e.g. **Image Captioning**
image -> sequence of words

Recurrent Neural Networks (RNN)



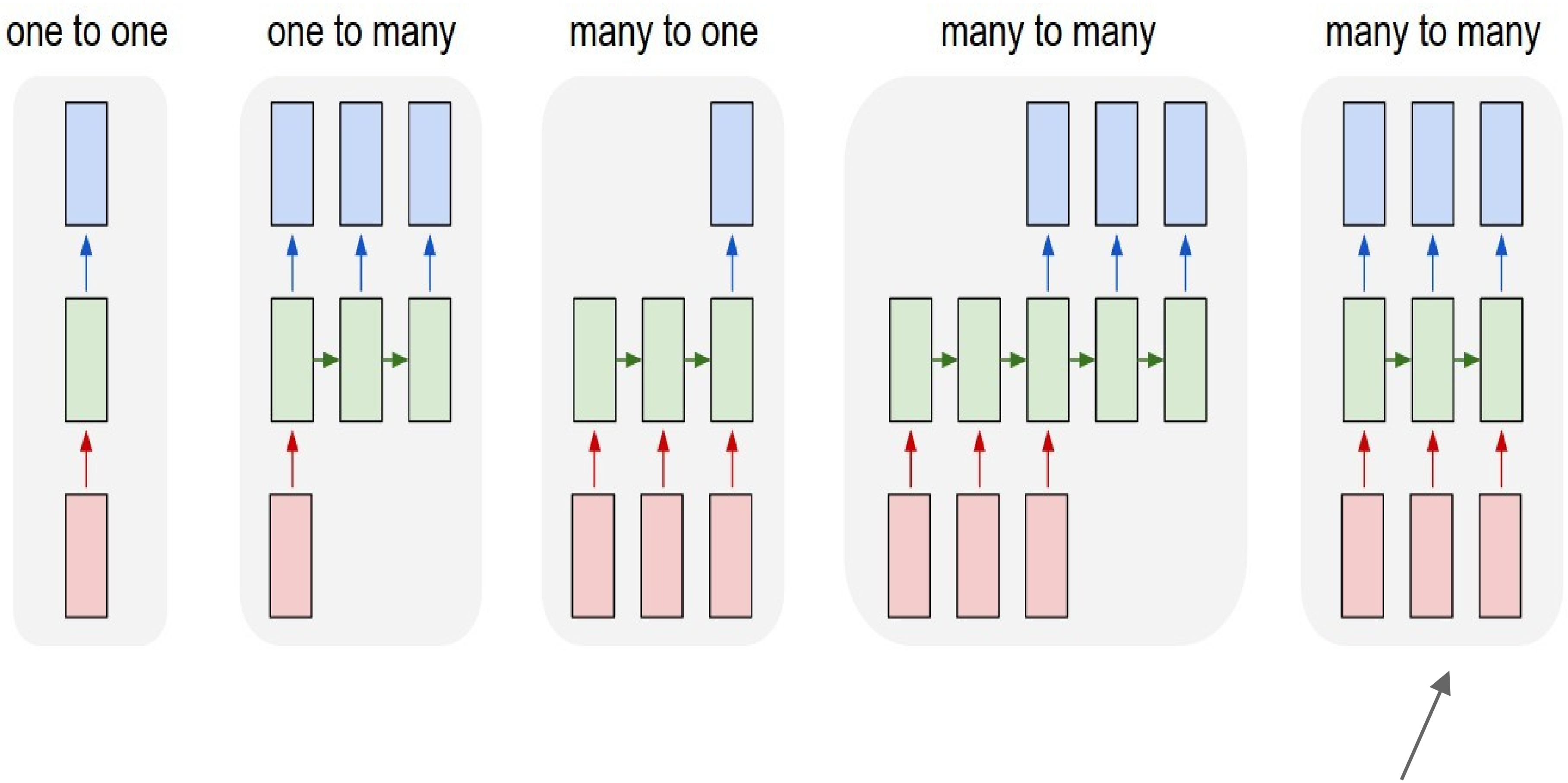
e.g. **Sentiment Classification**
sequence of words -> sentiment

Recurrent Neural Networks (RNN)



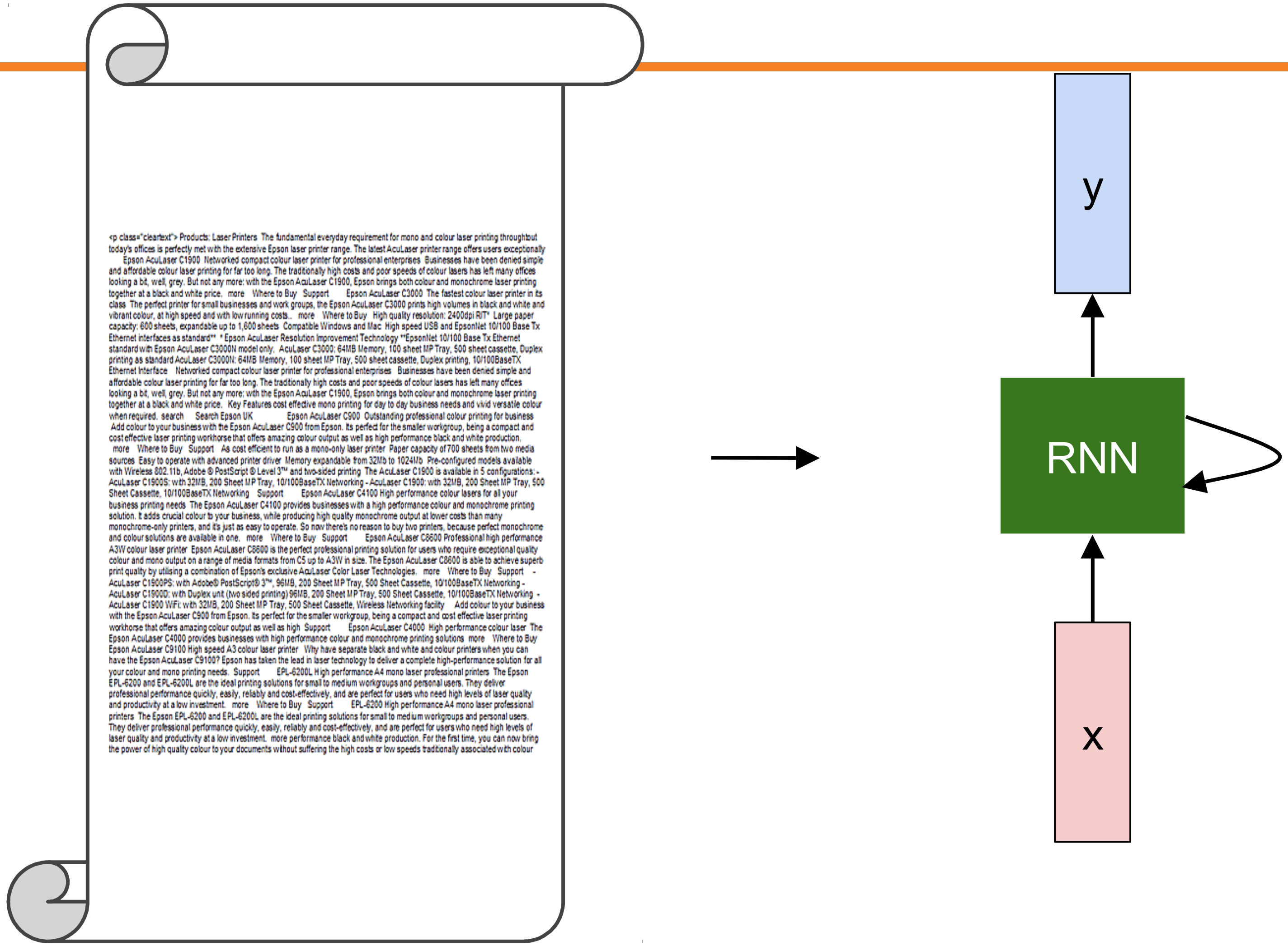
e.g. **Machine Translation**
seq of words -> seq of words

Recurrent Neural Networks (RNN)



e.g. **Video classification on frame level**

Slide Credit:



Slide Credit:

Character RNN during training

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

↓ train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

↓ train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

↓ train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```


Slide Credit:

PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.

VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

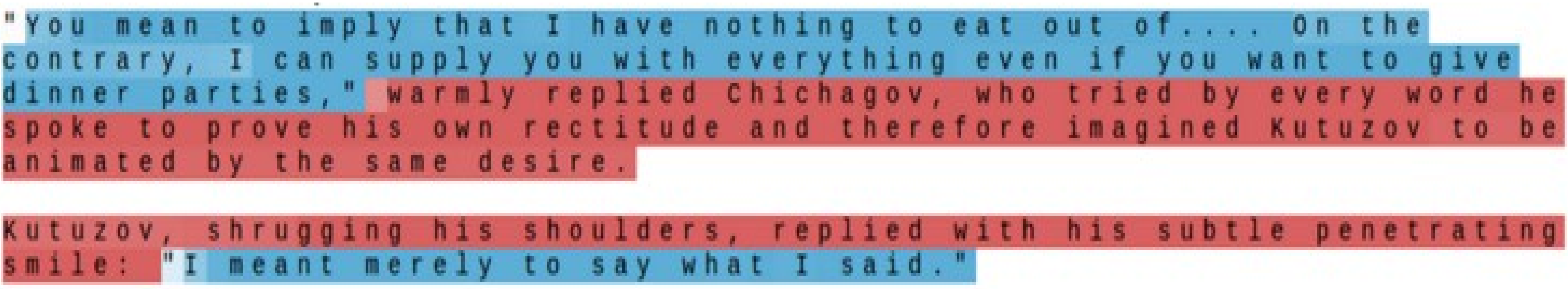
KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.

```c
static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```

Generated C code

Slide Credit:

Searching for interpretable cells

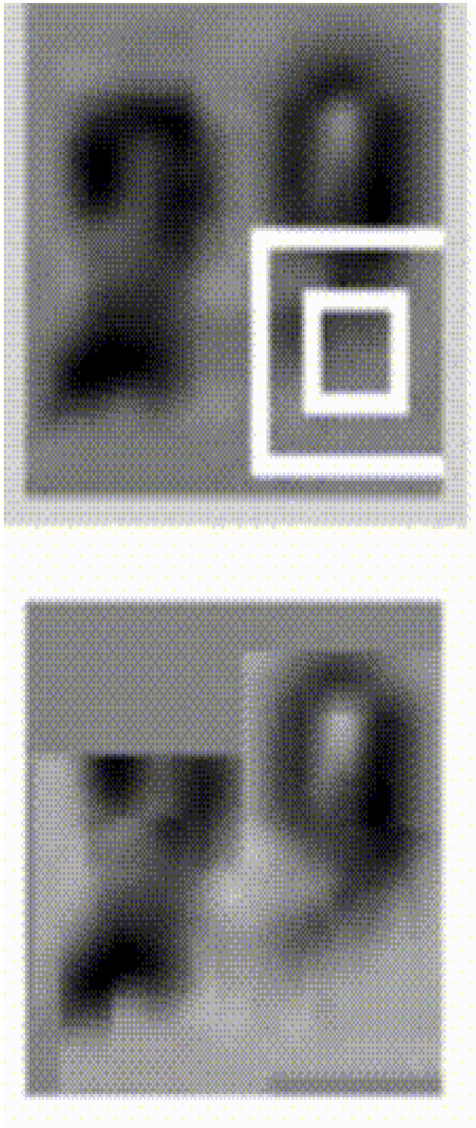


quote detection cell

Slide Credit:

Sequential Processing of fixed inputs

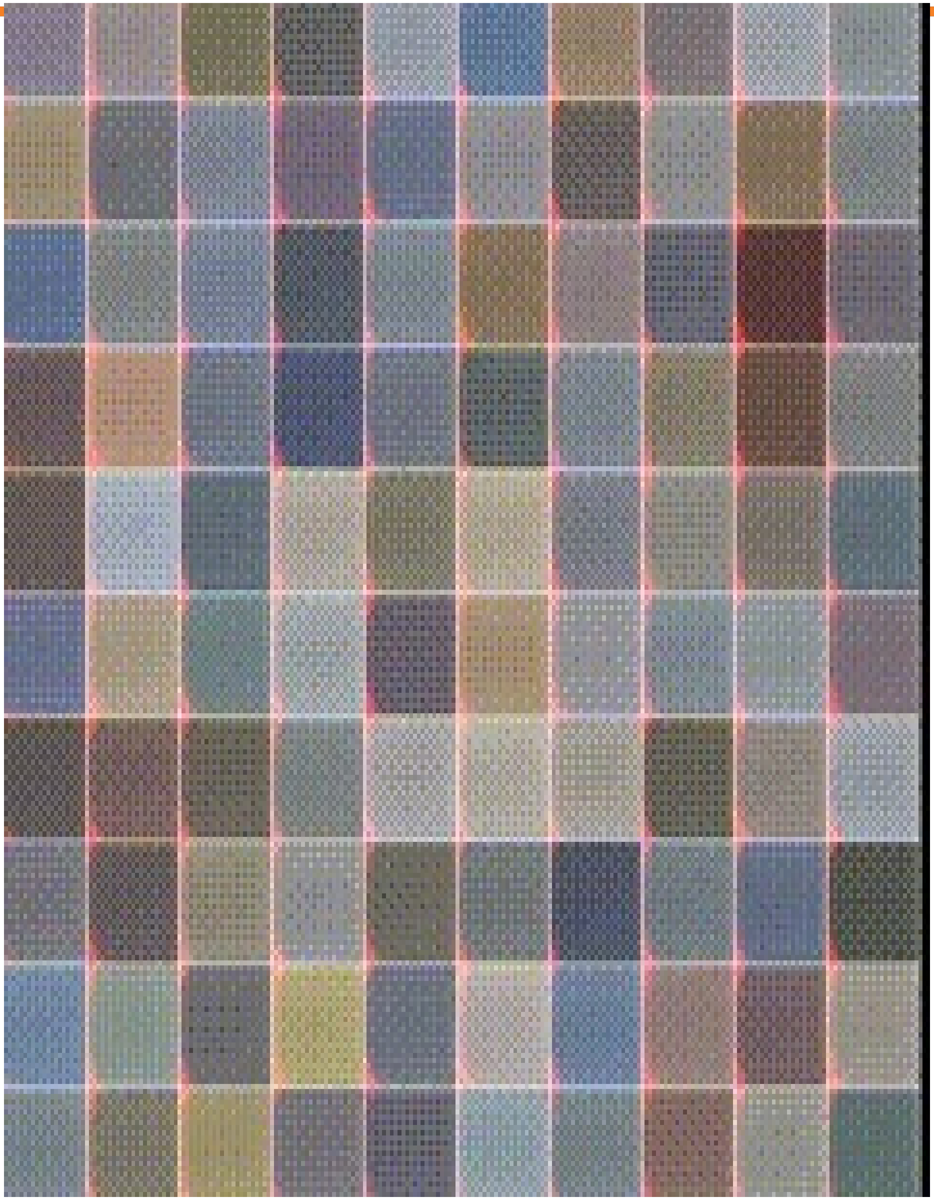
Multiple Object Recognition with Visual Attention, Ba et al.



Slide Credit:

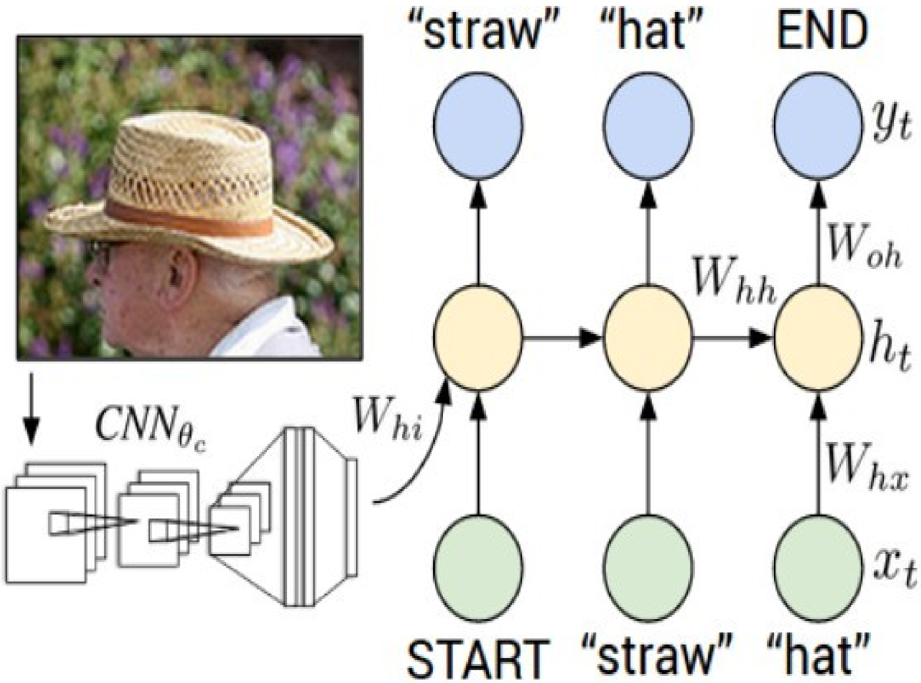
Sequential Processing of fixed outputs

DRAW: A Recurrent Neural Network For Image Generation, Gregor et al.



Slide Credit:

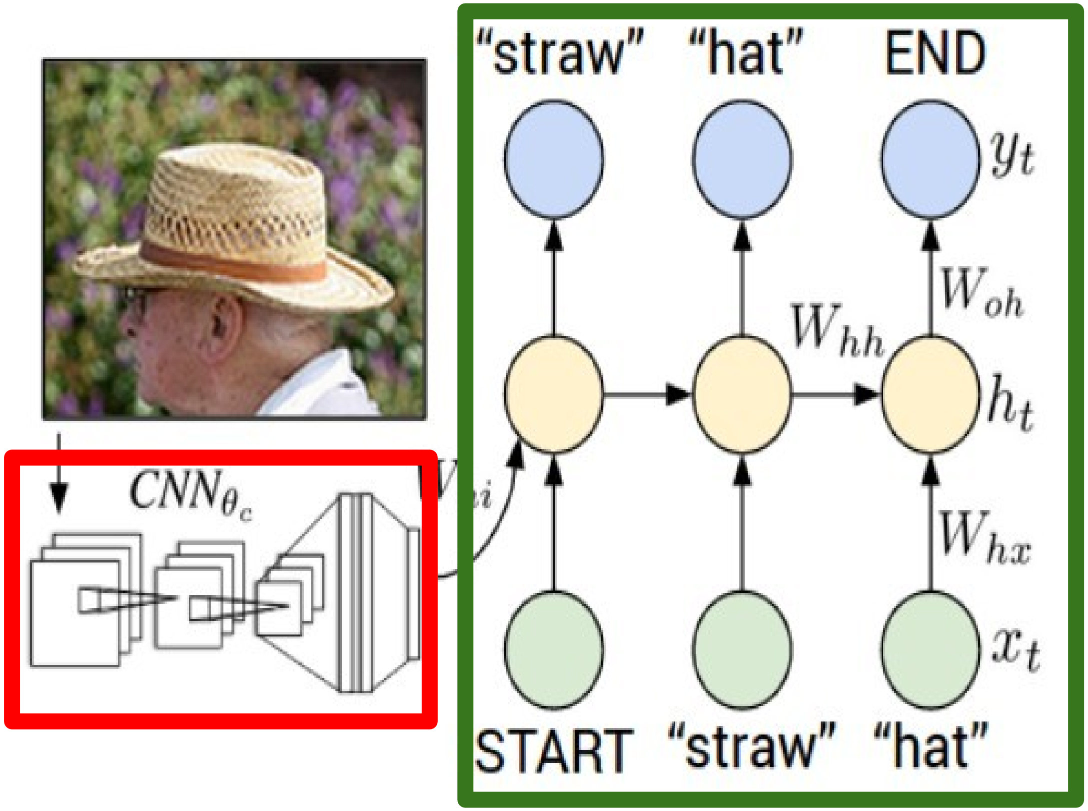
Image Captioning



Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

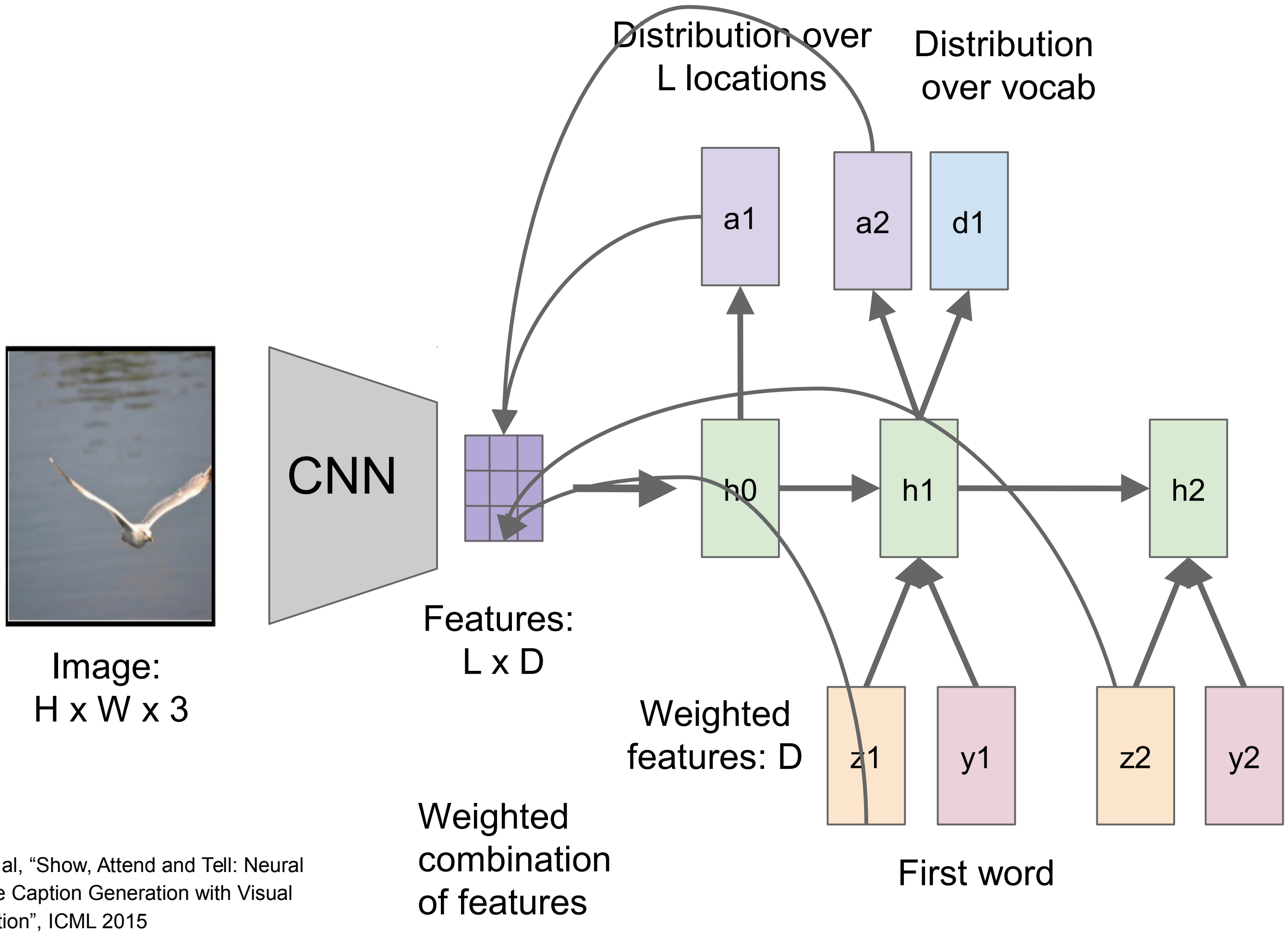
Slide Credit:

Recurrent Neural Network



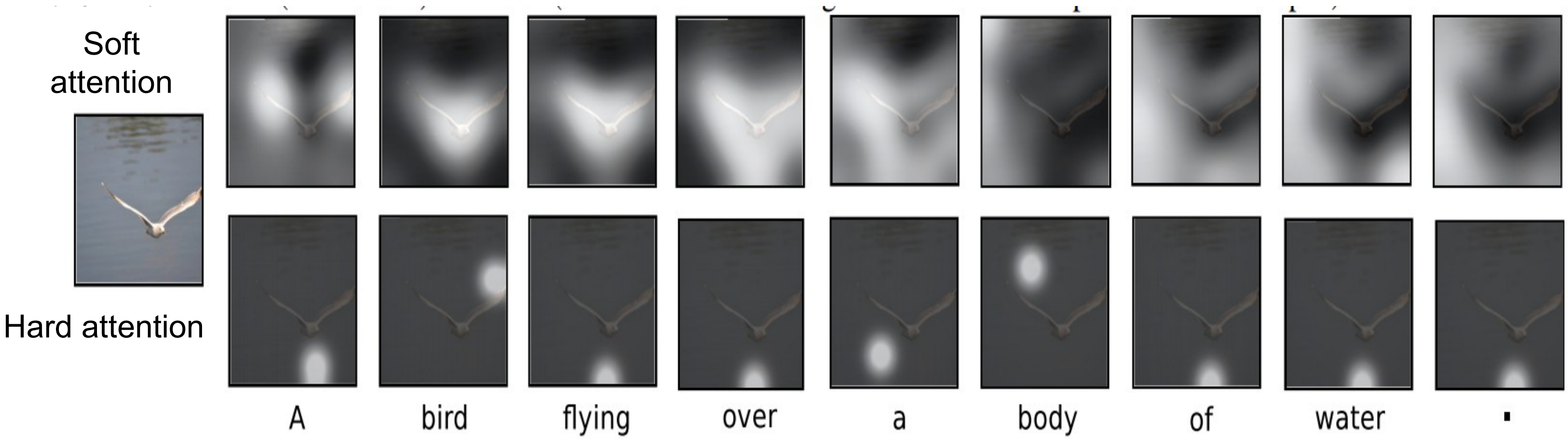
Convolutional Neural Network

Soft Attention for Captioning



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Soft Attention for Captioning

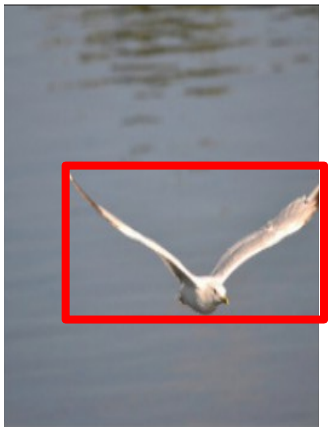


Xu et al, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, ICML 2015

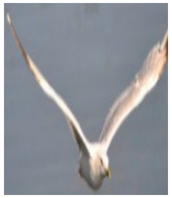
Soft Attention for Captioning

Xu et al, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, ICML 2015

Spatial Transformer Networks



Can we make this
function differentiable?



Input image:
H x W x 3

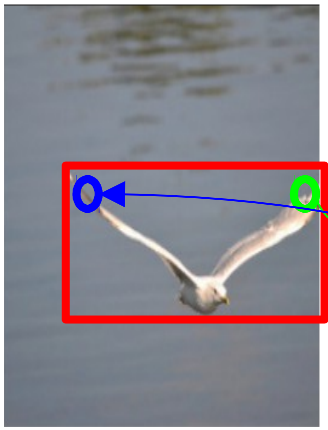
Cropped and
rescaled image:
X x Y x 3

Box Coordinates:
(xc, yc, w, h)

Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

Slide Credit:

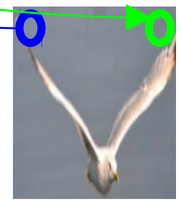
Spatial Transformer Networks



Input image:
H x W x 3

Box Coordinates:
(xc, yc, w, h)

Can we make this function differentiable?

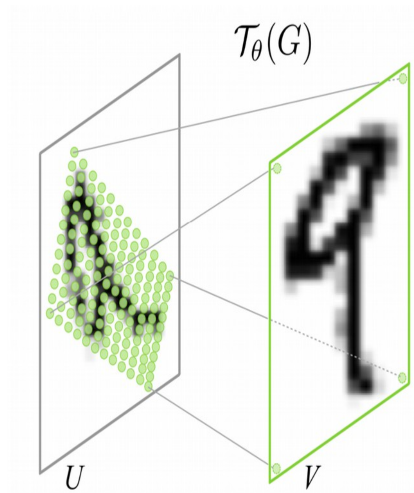


Cropped and rescaled image:
X x Y x 3

Idea: Function mapping *pixel coordinates* (xt, yt) of output to *pixel coordinates* (xs, ys) of input

Network attends to input by predicting θ

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



Repeat for all pixels in *output* to get a **sampling grid**

Then use **bilinear interpolation** to compute output

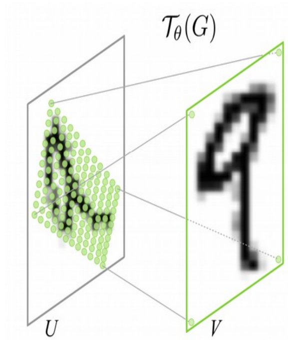
Jaderberg et al, “Spatial Transformer Networks”, NIPS 2015

Slide Credit:

Spatial Transformer Networks

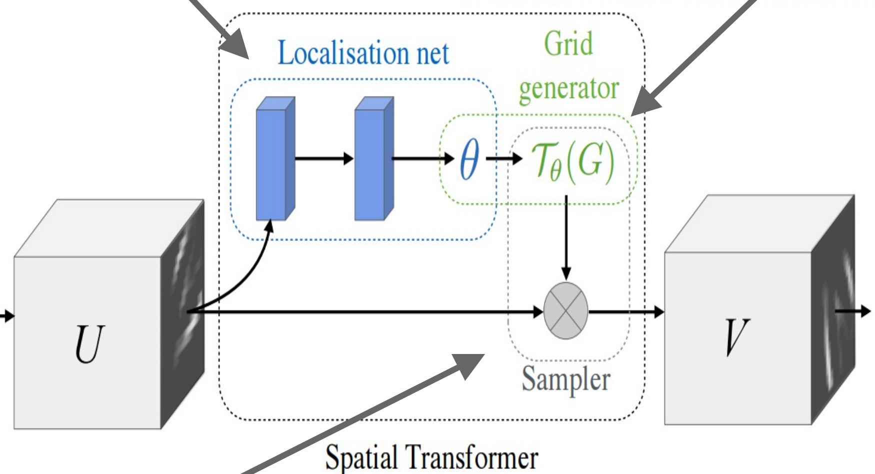
Grid generator uses θ to compute sampling grid

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



A small **Localization network** predicts transform θ

Input: Full image



Output: Region of interest from input

Sampler uses bilinear interpolation to produce output

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|)$$

Slide Credit:

Spatial Transformer Networks

Insert spatial transformers into a classification network and it learns to attend and transform the input

Differentiable “attention / transformation” module