CSE 252D: Advanced Computer Vision

Manmohan Chandraker

Lecture 12: Semantic Segmentation

Virtual classrooms

- Virtual lectures on Zoom
  - Only host shares the screen
  - Keep video off and microphone muted
  - But please do speak up (remember to unmute!)
  - Slides uploaded on webpage just before class
- Virtual interactions on Zoom
  - Ask and answer plenty of questions
  - “Raise hand” feature on Zoom when you wish to speak
  - Post questions on chat window
  - Happy to try other suggestions!
- Lectures recorded and upload on Canvas
  - Available under “My Media” on Canvas

Overall goals for the course

- Introduce the state-of-the-art in computer vision
- Study principles that make them possible
- Get understanding of tools that drive computer vision
- Enable one or all of several such outcomes
  - Pursue higher studies in computer vision
  - Join industry to do cutting-edge work in computer vision
  - Gain appreciation of modern computer vision technologies
- This is a great time to study computer vision!

Papers for Wed, May 12

- Deep High-Resolution Representation Learning for Human Pose Estimation
  - https://arxiv.org/abs/1902.09212
- Simple Baselines for Human Pose Estimation and Tracking
  - https://arxiv.org/abs/1804.06208
- OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields
  - https://arxiv.org/abs/1812.08008
- End-to-end Recovery of Human Shape and Pose
  - https://arxiv.org/abs/1712.06584

Papers for Fri, May 14

- ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation
  - https://arxiv.org/abs/1606.02147
- ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation
  - https://ieeexplore.ieee.org/abstract/document/8063438
- Fast-SCNN: Fast Semantic Segmentation Network
  - https://arxiv.org/abs/1902.04502
- Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation
  - https://arxiv.org/abs/1506.04924

Papers for Wed, May 19

- Context Encoding for Semantic Segmentation
  - https://arxiv.org/abs/1803.08904
- Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation
  - https://arxiv.org/abs/1802.02611
- High-Resolution Representations for Labeling Pixels and Regions
  - https://arxiv.org/abs/1904.04514
- MSeg: A Composite Dataset for Multi-Domain Semantic Segmentation
  - https://ieeexplore.ieee.org/document/9157628

Recap

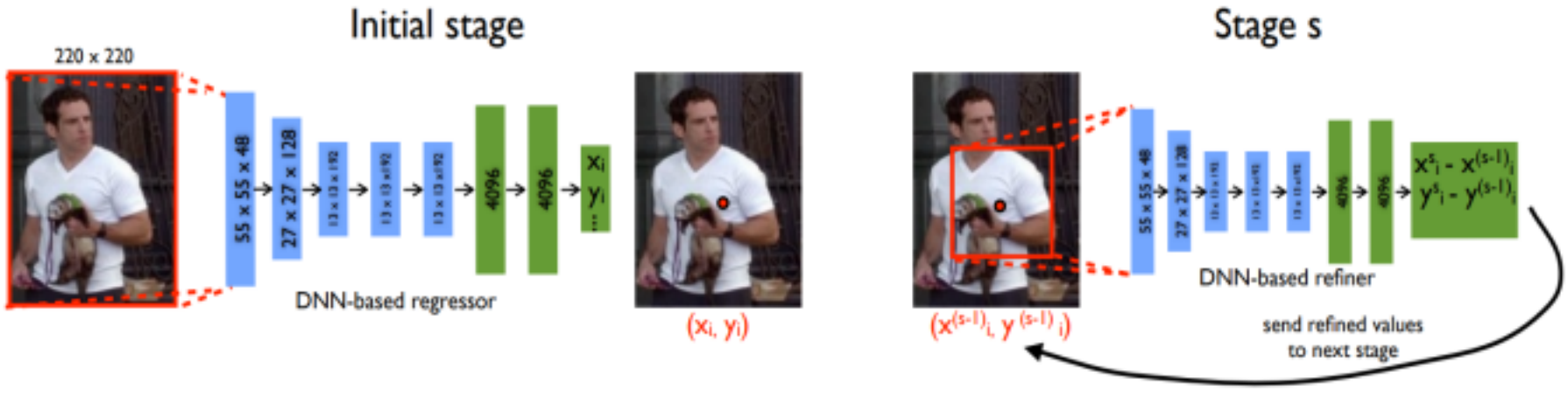
Local and Global Needed for Pose Estimation

- Local part-based detectors are not sufficient for pose estimation

Cascade of Regressors

- Advantages:
  - Simple, yet holistic
  - No need to define losses that capture interactions
  - Instead, all hidden layers are shared by joint regressors
  - Increasingly detailed prediction along cascade stages

- Disadvantages:
  - ~~Limited ability to consider details~~
  - ~~One prediction per image, no candidates~~
  - ~~Depends on quality of initial prediction~~

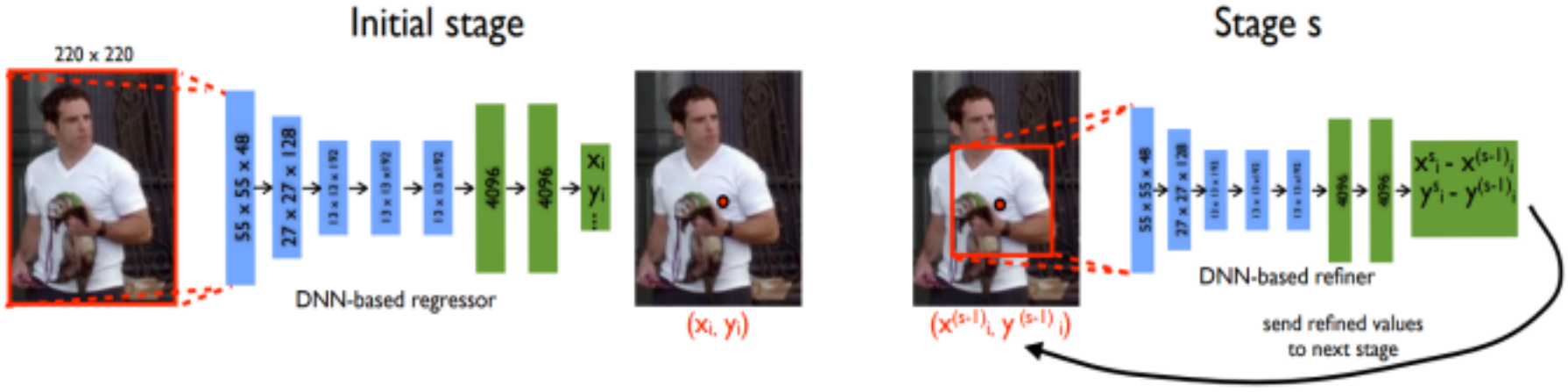


[Toshev and Szegedy, DeepPose]

Cascade of Regressors

- Advantages:
  - Simple, yet holistic
  - No need to define losses that capture interactions
  - Instead, all hidden layers are shared by joint regressors
  - Increasingly detailed prediction along cascade stages

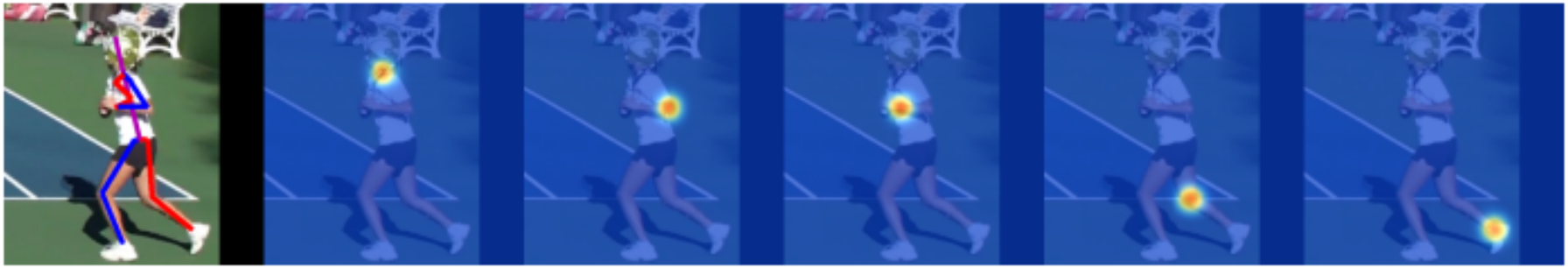
- Disadvantages:
  - ~~Limited ability to consider details~~
  - One prediction per image, no candidates
  - Depends on quality of initial prediction



[Toshev and Szegedy, DeepPose]

Cascade of Heat Maps

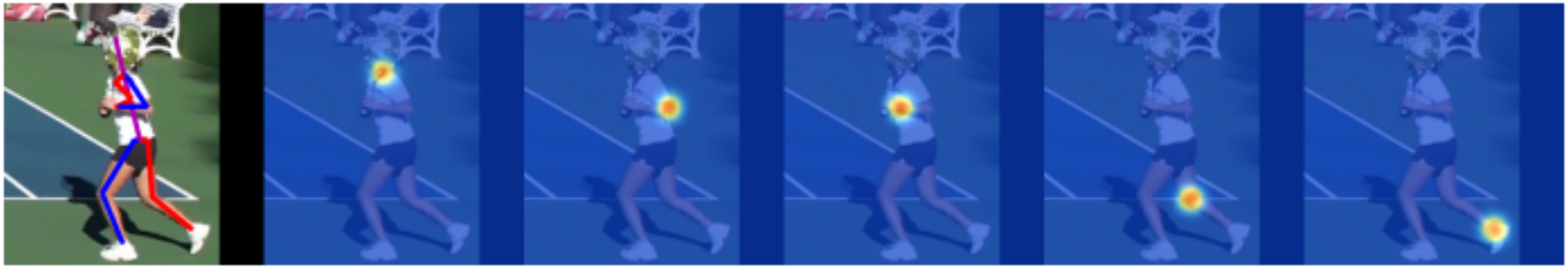
- Advantages:
  - Simple, yet holistic
  - No need to define losses that capture interactions
  - Instead, all hidden layers are shared by joint regressors
  - Increasingly detailed prediction along cascade stages
  - Encodes probability of joint appearing at a pixel
- Disadvantages:
  - ~~Limited ability to consider details~~
  - ~~One prediction per image, no candidates~~
  - Depends on quality of initial prediction
  - ~~Not enough modeling of spatial structure~~
  - ~~Cannot reason about occluded parts, or those outside window~~



[Tompson et al., Efficient Object Localization]

Cascade of Heat Maps

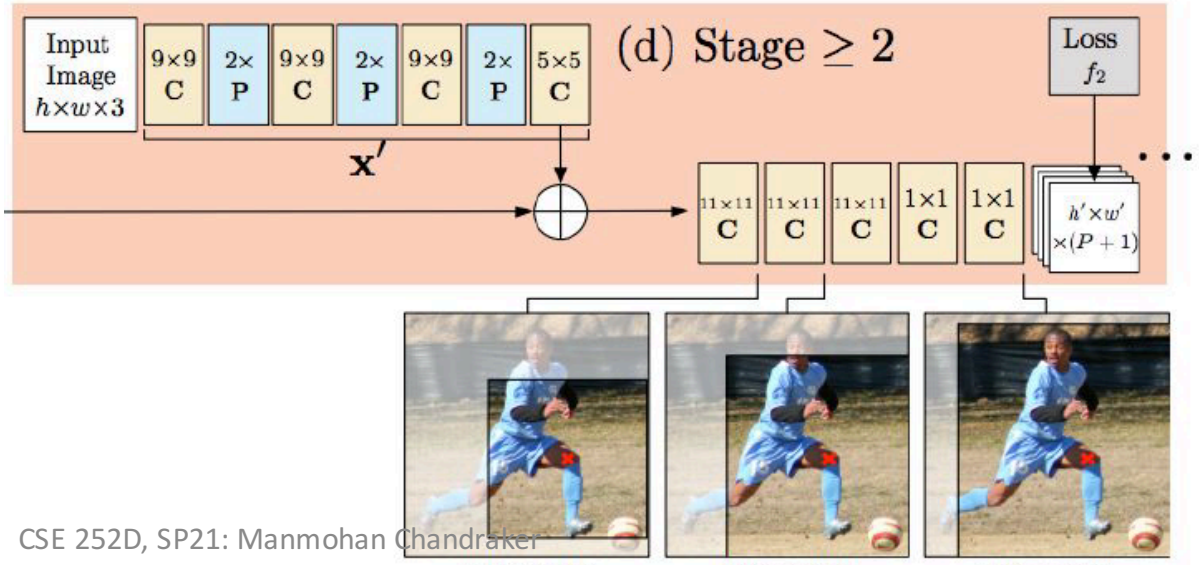
- Advantages:
  - Simple, yet holistic
  - No need to define losses that capture interactions
  - Instead, all hidden layers are shared by joint regressors
  - Increasingly detailed prediction along cascade stages
  - Encodes probability of joint appearing at a pixel
- Disadvantages:
  - ~~Limited ability to consider details~~
  - ~~One prediction per image, no candidates~~
  - Depends on quality of initial prediction
  - Not enough modeling of spatial structure
  - Cannot reason about occluded parts, or those outside window



[Tompson et al., Efficient Object Localization]

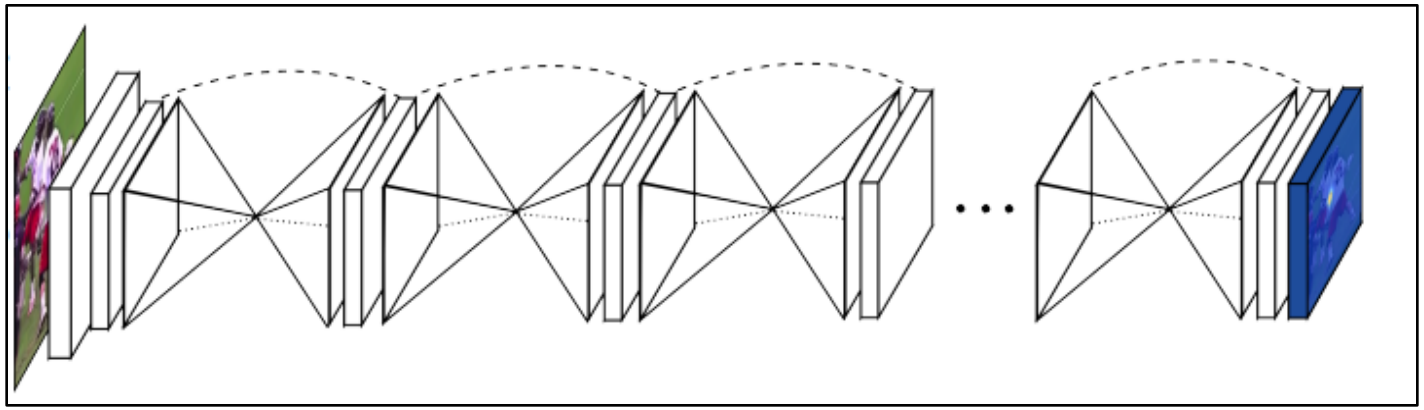
Convolutional Pose Machines: Next Stages

- Image features **x'** from the previous stage
- Context function ψ encodes landscape of belief maps around part locations
  - In practice, ψ is just the receptive field
  - Network decides how to combine features and learn higher relations
  - Previously: hand-defined potential functions in graphical model
- Three ways to increase size of receptive field
  - More pooling: lose local details
  - Larger filters: increase number of parameters
  - More layers: vanishing gradients ⇨ Intermediate supervision



Stacked Hourglass Network

- Multiple iterative stages allow refinement
- Each stage does full bottom-up and top-down processing (no weights shared)
- Repeated inference, with intermediate supervision for each stage
  - Network has had a chance to reason both locally and globally
  - Subsequent hourglass modules can reassess high-order spatial relations
  - Ask network to repeatedly reason across scales
- Intermediate supervision not straightforward for single hourglass module
  - Cannot apply before pooling since only local information available
  - After upsampling, cannot re-evaluate features globally



Recipes for Human Pose Estimation

- Local appearance: needed for accurate part detection
- Global reasoning: orientation of body, limb arrangement, part relationships

Recipes for Human Pose Estimation

- Local appearance: needed for accurate part detection
- Global reasoning: orientation of body, limb arrangement, part relationships

- CNNs are inherently well-suited for this task
  - Multiscale feature extraction
  - Shared hidden layers capture part interactions

- But mechanisms needed to coax local and global performance out of CNNs

Recipes for Human Pose Estimation

- Local appearance: needed for accurate part detection
- Global reasoning: orientation of body, limb arrangement, part relationships

- CNNs are inherently well-suited for this task
  - Multiscale feature extraction
  - Shared hidden layers capture part interactions

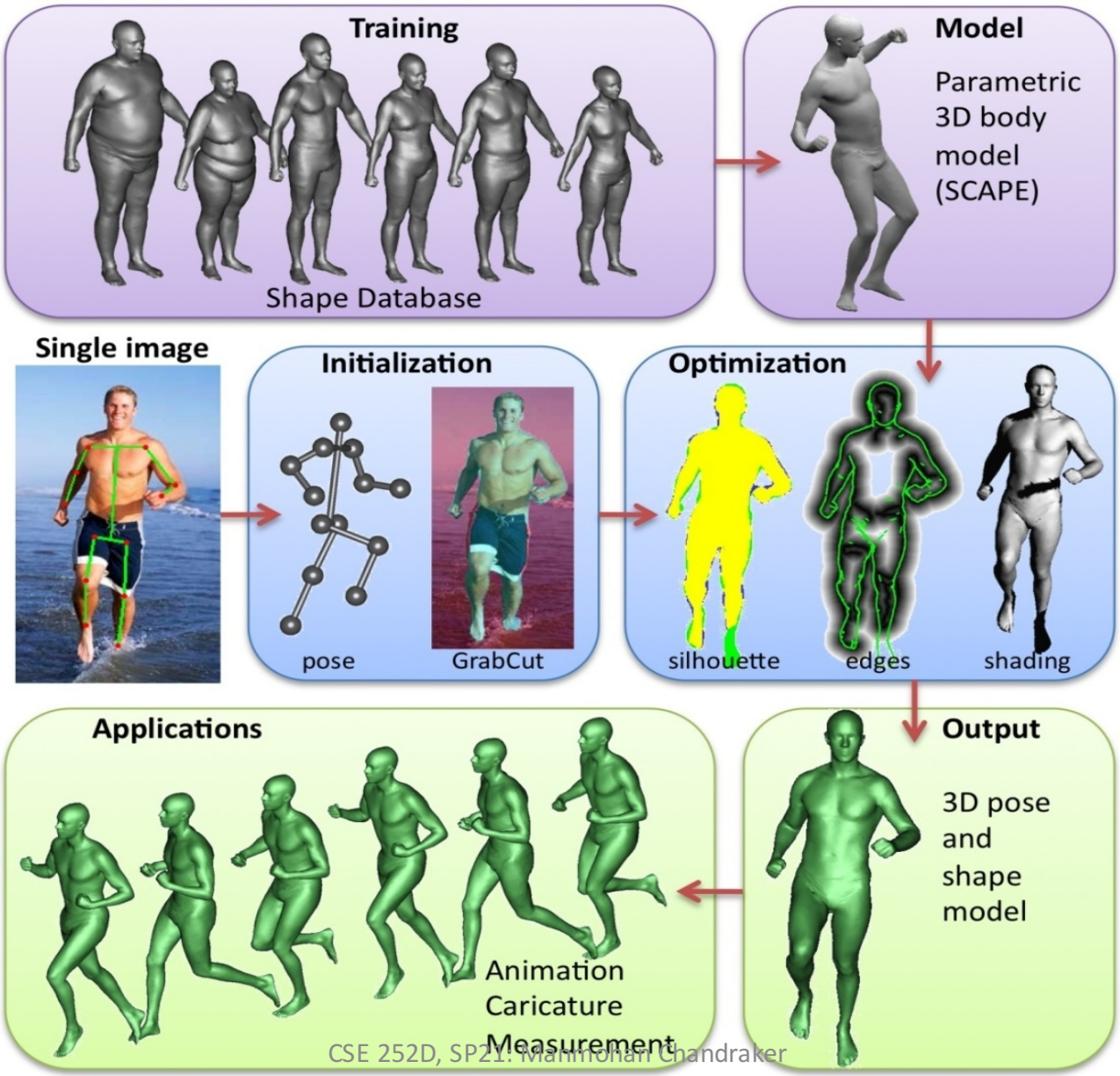
- But mechanisms needed to coax local and global performance out of CNNs

- Heat maps
  - Predict probability of joint appearing at a pixel

- Cascades
  - Iteratively refine predictions for fine localization
  - Long-range interactions through wider receptive fields

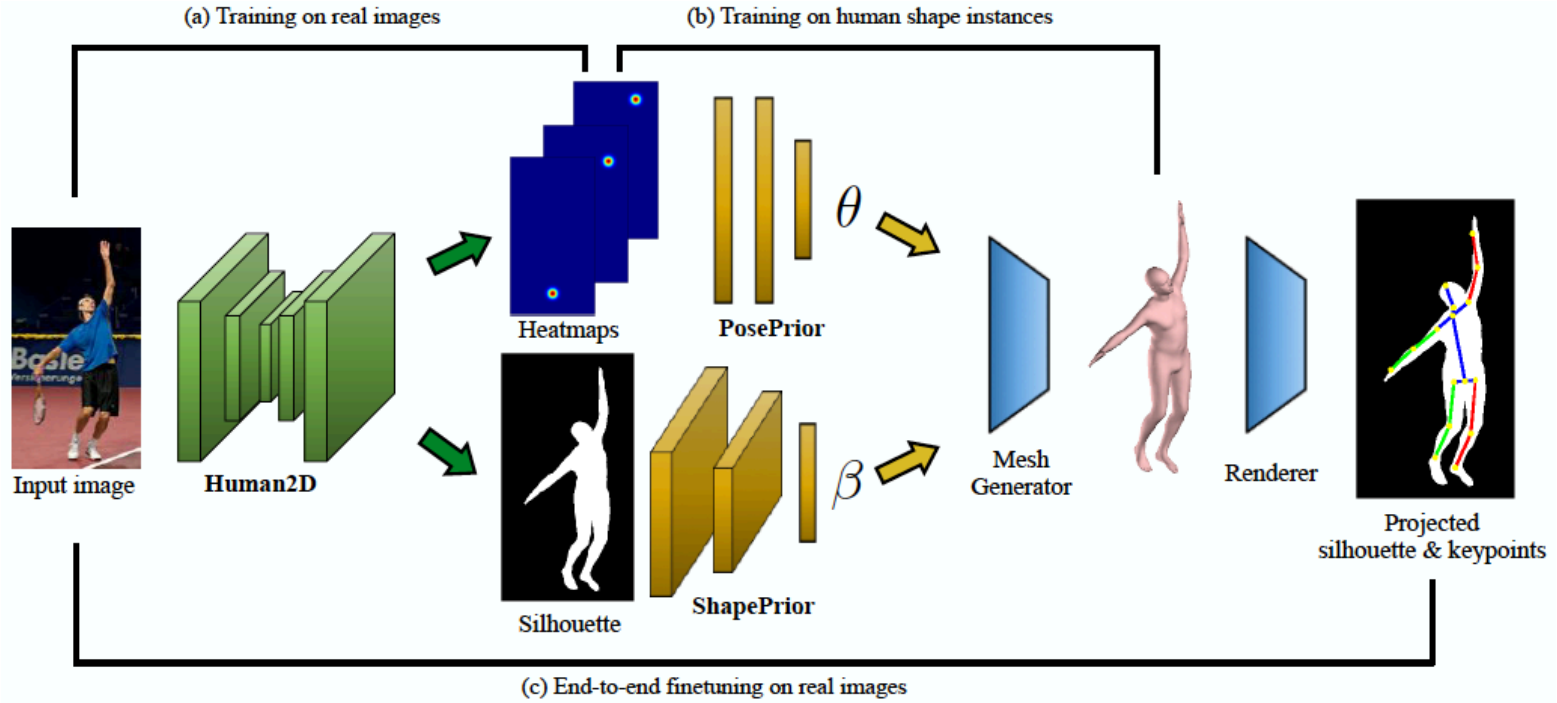
- Intermediate supervision
  - Prevent vanishing gradients through cascade stages
  - Allow cascade to repeatedly assess local and global information

Shape Fitting Pipeline



Predicting 3D Human Shape and Pose

- Use data-driven priors from deep network to avoid optimization difficulties
- Large-scale data exists for 2D keypoints and silhouettes
- Train a multi-task stacked hourglass network for heat maps and silhouette

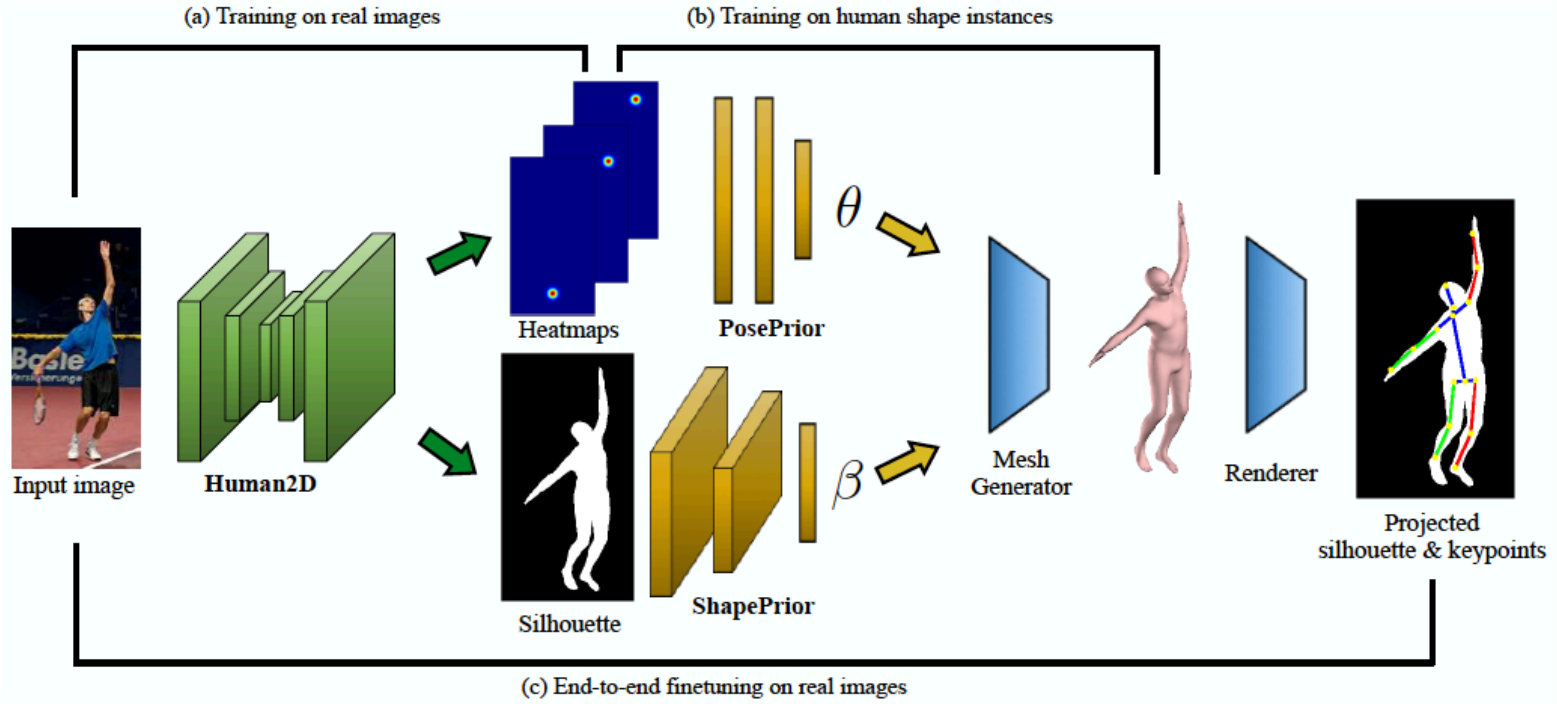


[Pavlakos et al., Learning to Estimate 3D Human Pose]

Predicting 3D Human Shape and Pose

- Use heat maps to predict pose, silhouette to predict shape parameters in SMPL
- Ignore inter-dependence, in favor of disentanglement
- Train based on data rendered from ground truth SMPL instances
- Objectives: 3D vertex loss and joint loss

$$\mathcal{L}_{\mathcal{M}} = \sum_{i=1}^{N} \|\hat{P}_i - P_i\|_2^2, \qquad \mathcal{L}_{\mathcal{J}} = \sum_{i=1}^{M} \|\hat{J}_i - J_i\|_2^2.$$



[Pavlakos et al., Learning to Estimate 3D Human Pose]

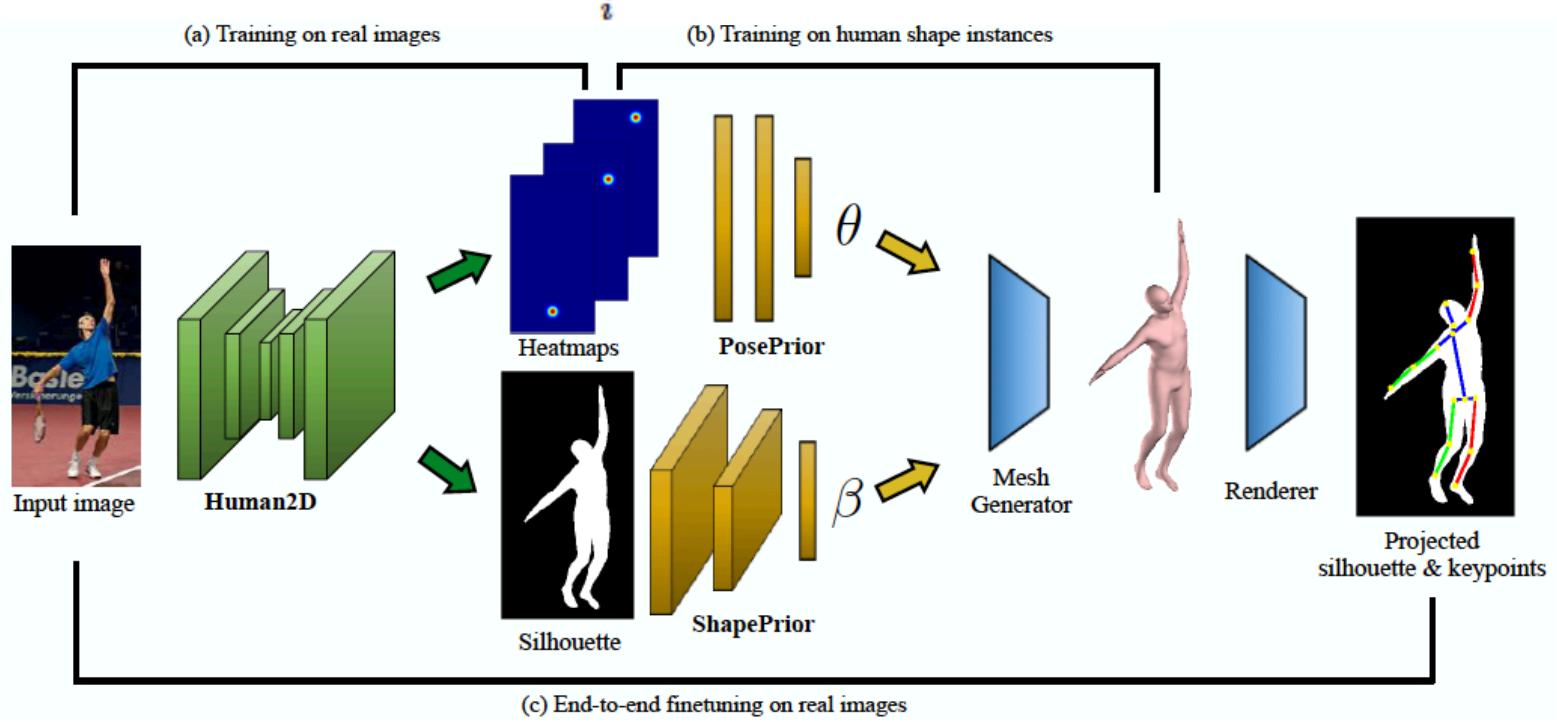
Predicting 3D Human Shape and Pose

- Fine-tune end-to-end using differentiable rendering
- Project vertex estimates to obtain silhouette, joint estimates for keypoints

$$\Pi(\hat{\boldsymbol{P}}) = \hat{\boldsymbol{S}}, \quad \Pi(\hat{\boldsymbol{J}}) = \hat{\boldsymbol{W}} \in \mathbb{R}^{M\times 2}.$$

- Supervised loss to minimize reprojection error:

$$\mathcal{L}_{\Pi} = \mu \sum_{i}^{M} \|\hat{W}_i - W_i\|_2^2 + \|\hat{S} - S\|_2^2$$



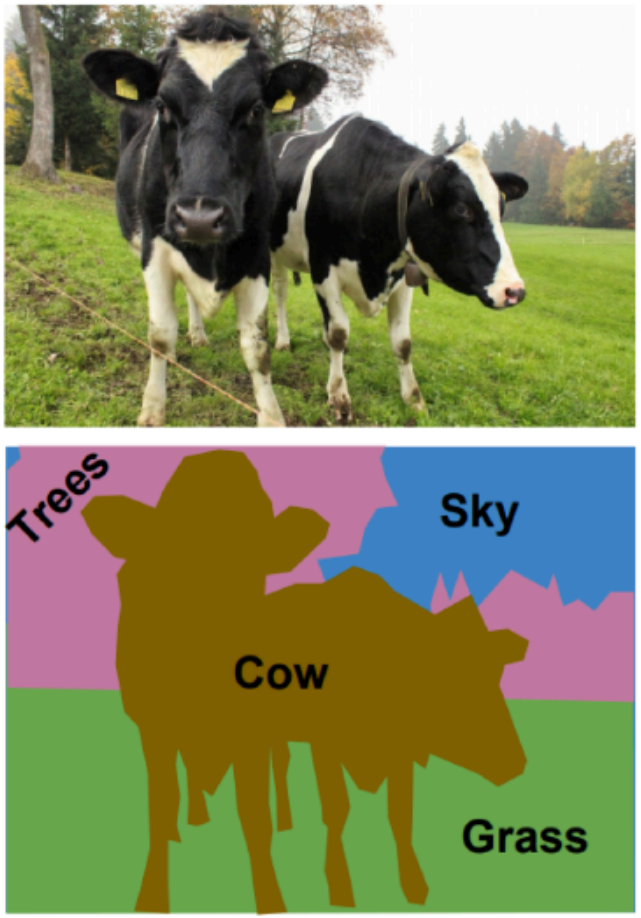
[Pavlakos et al., Learning to Estimate 3D Human Pose]

Predicting 3D Human Shape and Pose

Semantic Segmentation

Semantic Segmentation

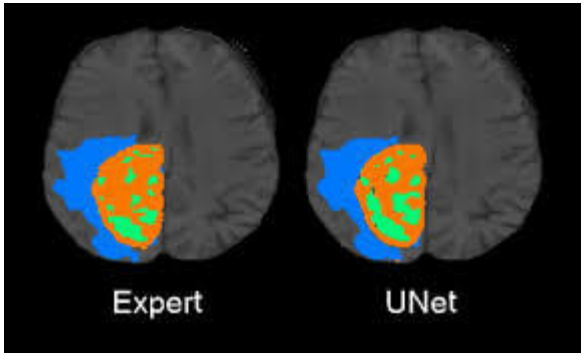
- Task: Assign each image pixel a class label

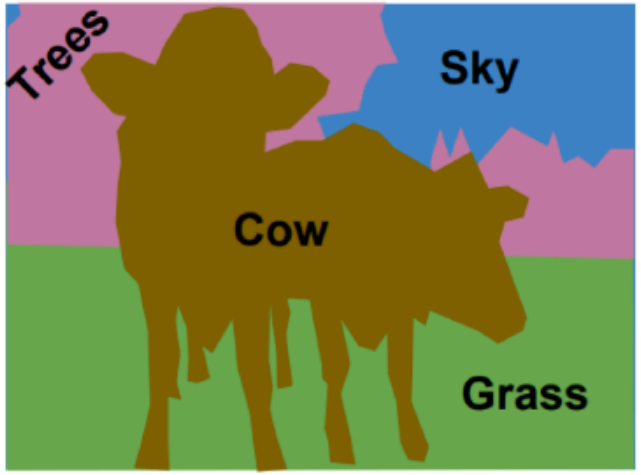


Semantic Segmentation

- Task: Assign each image pixel a class label

- Important applications:
  - Medicine
  - Self-driving





Semantic Segmentation

- Task: Assign each image pixel a class label

- Important applications:
  - Medicine
  - Self-driving
  - Selfies!!

[Bazarevsky and Tkachenka, Google AI blog]

Semantic Segmentation

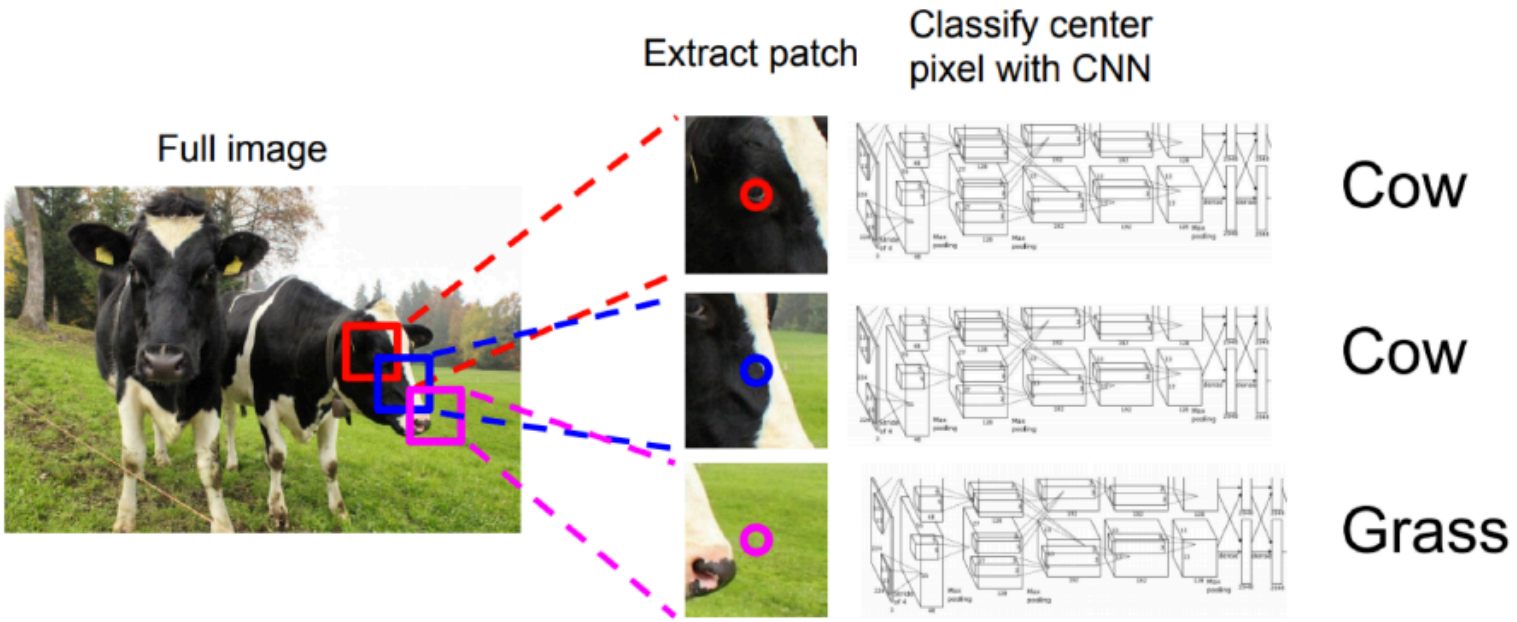
Global Reasoning ⟷ Locally accurate Boundaries

Figure from CityScapes Dataset

Per-Pixel Classification for Semantic Segmentation

Approaches

What could be a Problem?



Per-Pixel Classification for Semantic Segmentation

Approaches

What could be a Problem?
-Very inefficient
-No shared computation between overlapping patches

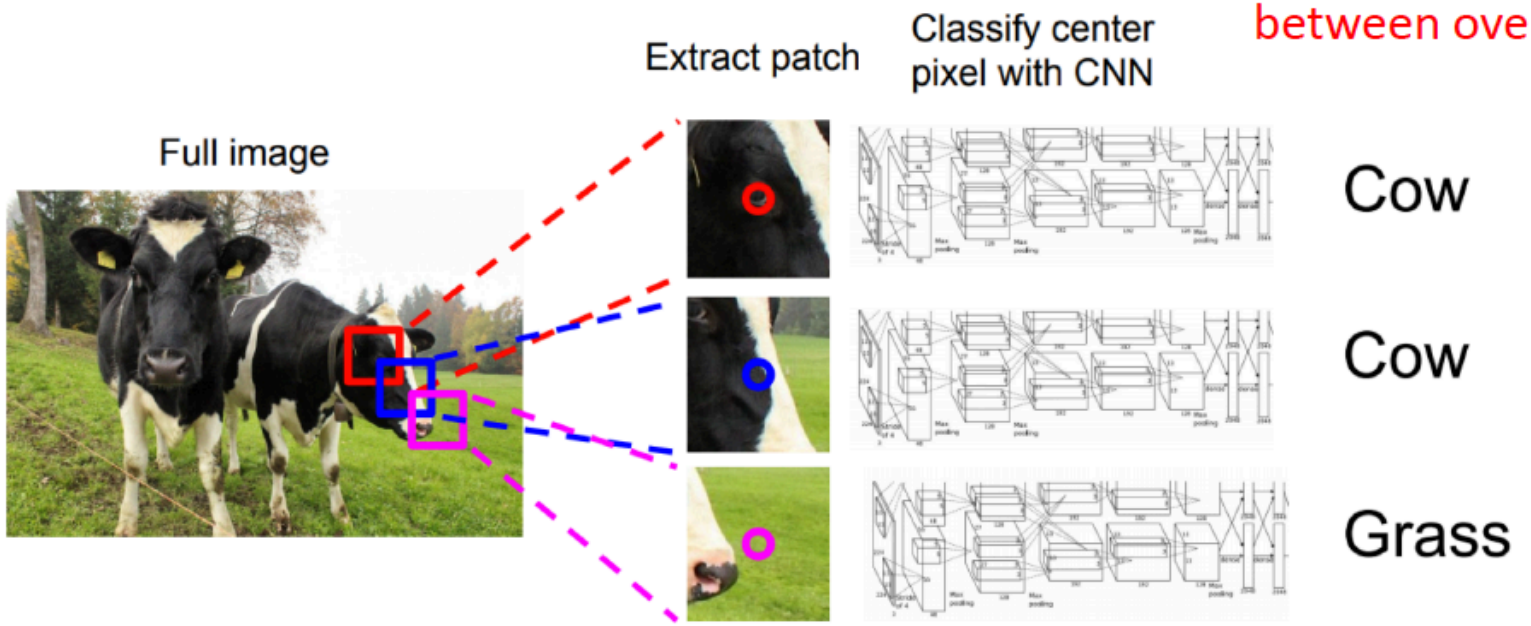
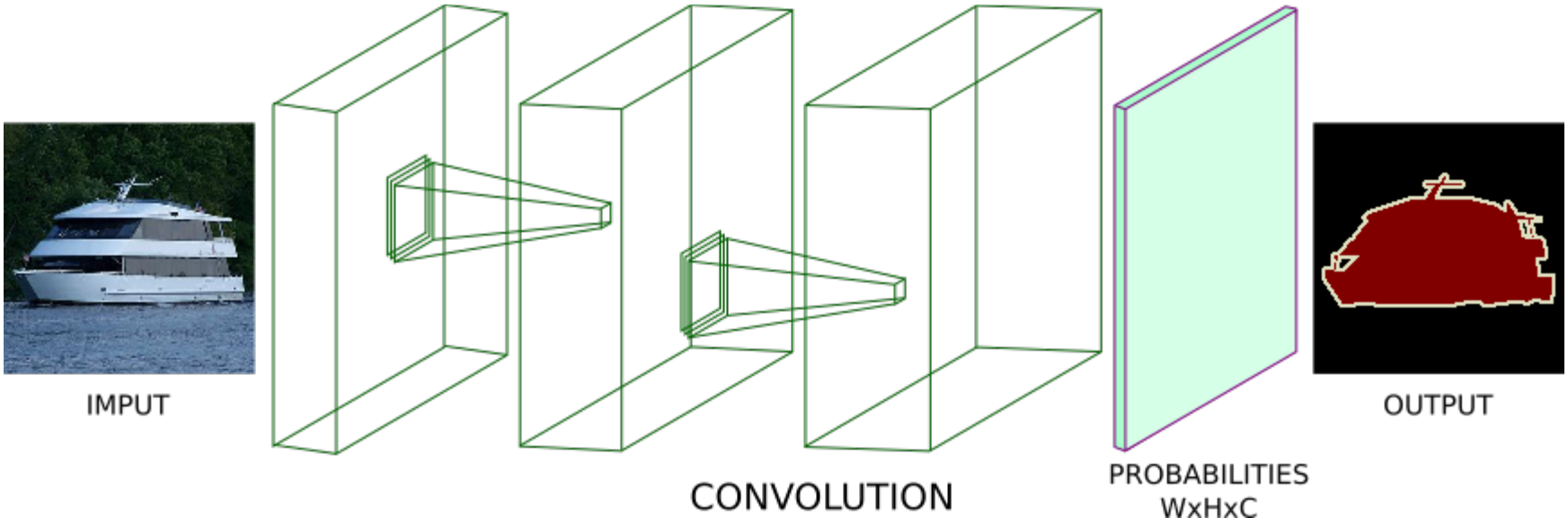


Image Resolution Convolutions



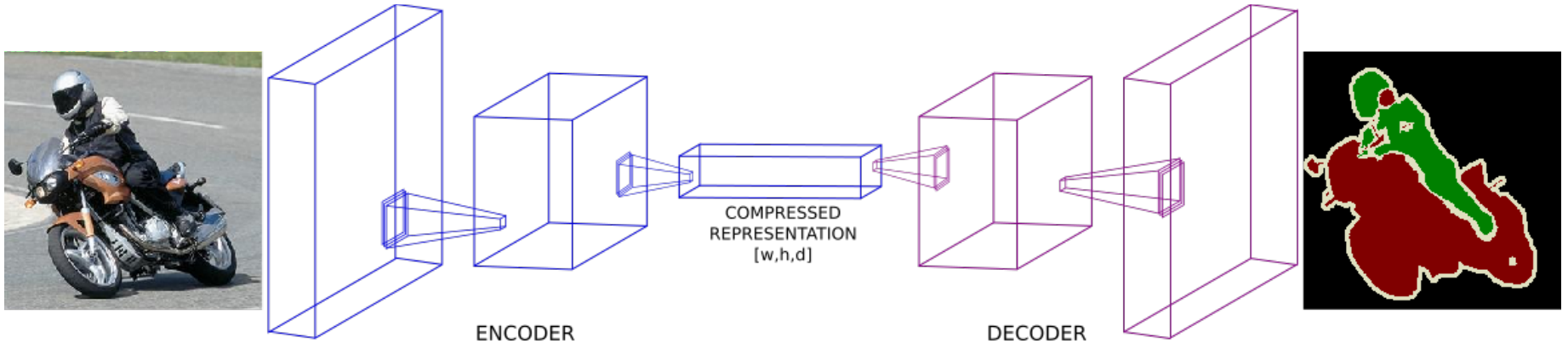
- Purely convolutions with stride 1, no pooling
- Maintains image resolution

- Drawbacks:
  - Expensive in memory and computation
  - Insufficient global reasoning

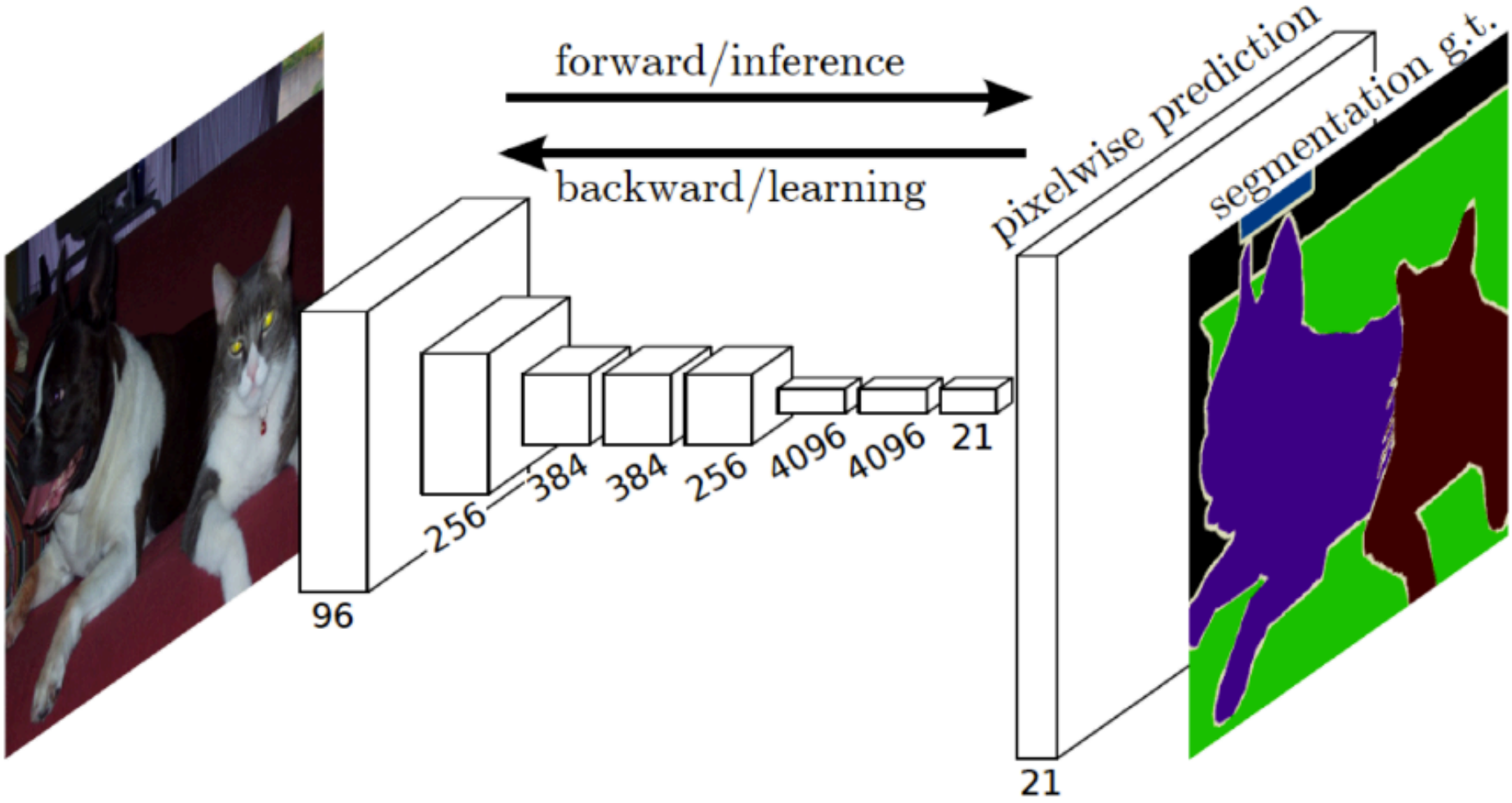
General Form of Segmentation Networks

- Most networks have similar encoders (inspired by classification networks)
  - Pooling and striding to downsample
  - Goal: do deeper convolutions without memory concerns

- Differences usually in decoder
  - Interpolation, transposed convolution, unpooling
  - Goal: recover spatial detail



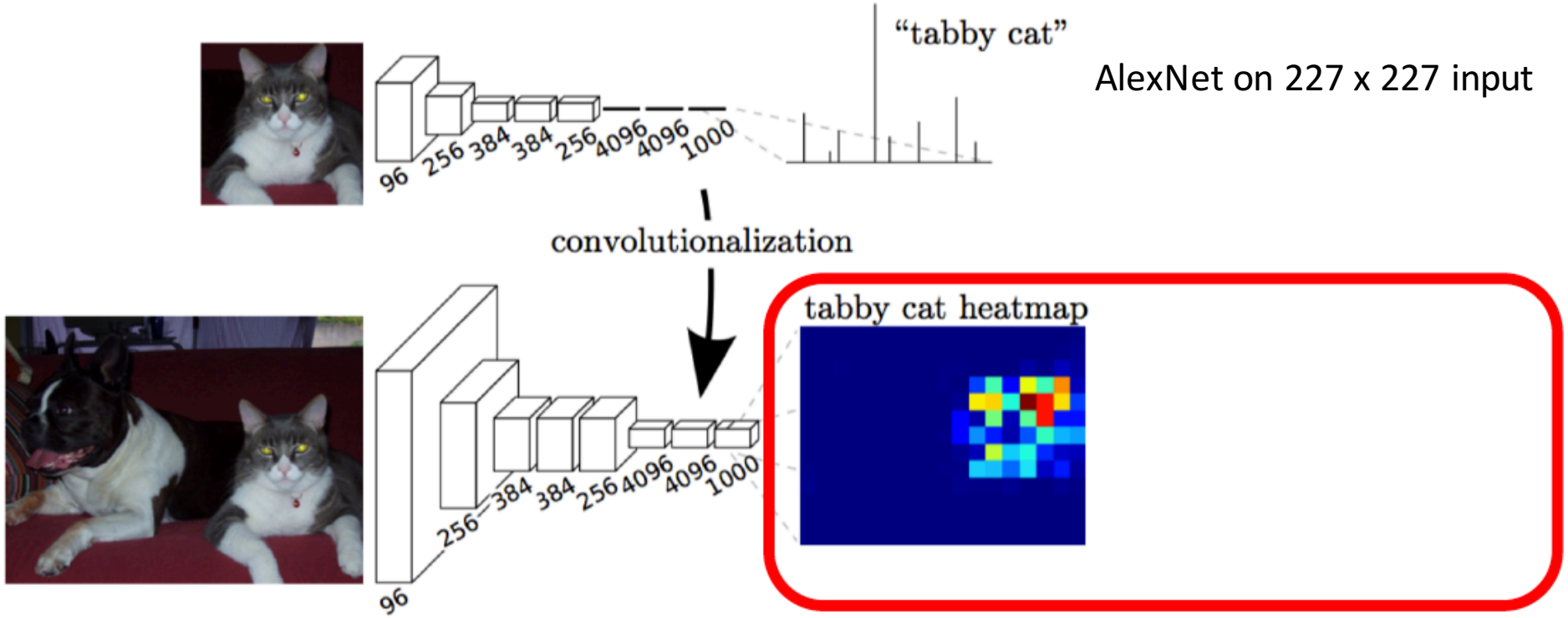
Fully Convolutional Network



[Long et al., FCNs for Semantic Segmentation]

Converting to Fully Convolutional Network

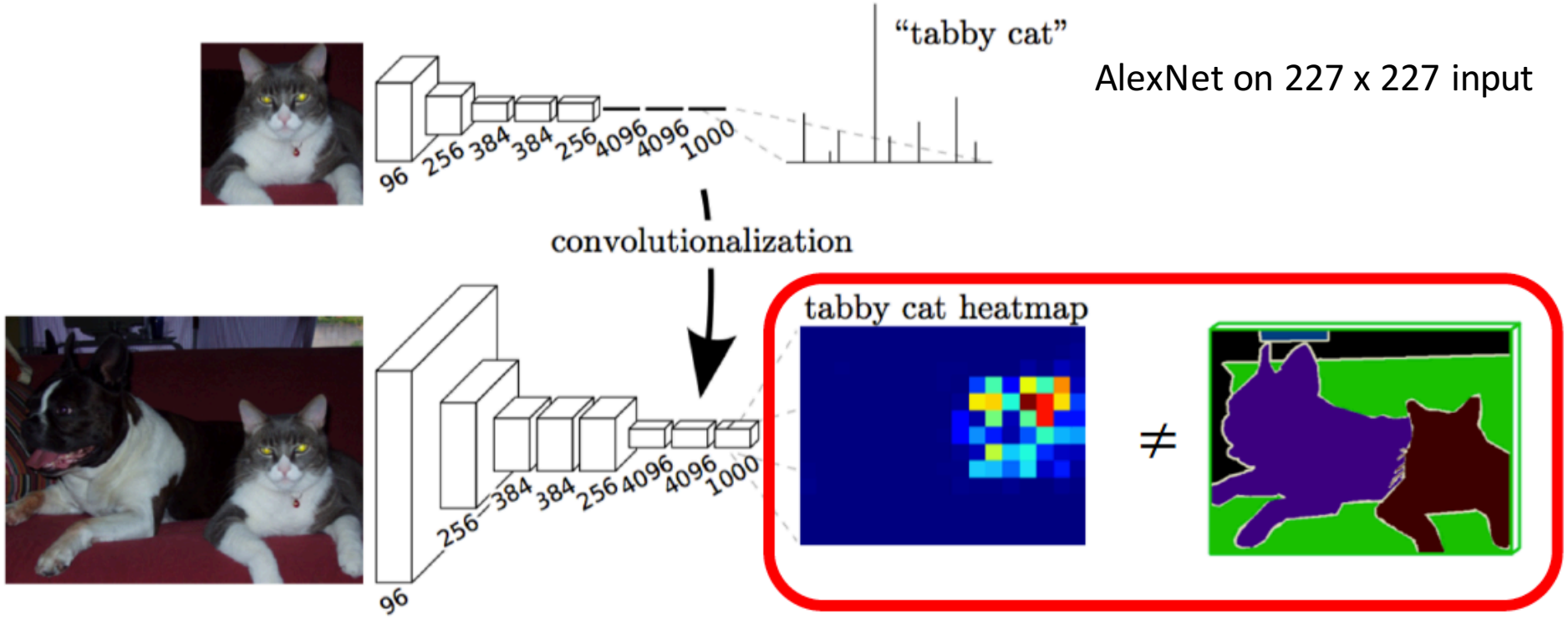
Fully-connected layer with *k* units = Convolution layer with *k* filters of size that covers input



Given 500 x 500 image, slide FCN with stride 32 to get 10 x 10 output.

Coarse Predictions

Fully-connected layer with *k* units = Convolution layer with *k* filters of size that covers input



AlexNet on 227 x 227 input

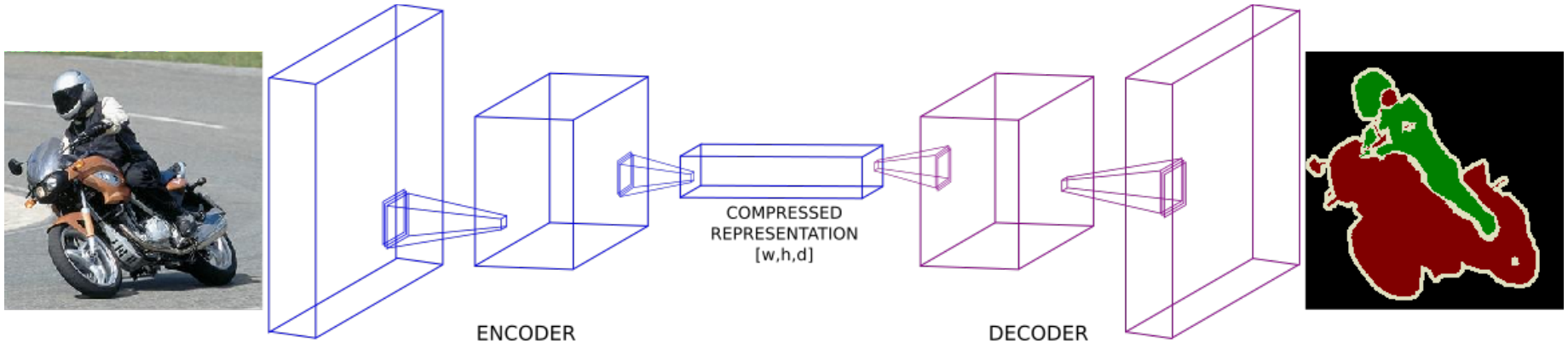
Given 500 x 500 image, slide FCN with stride 32 to get 10 x 10 output.

We want a segmentation output at image resolution.

Output Going to Image Resolution

- Encoder aggressively pools and subsamples image
- Necessary to capture context information which is necessary for segmentation
- But spatial detail is also necessary

- Goal for decoder: obtain output at image *resolution*
- Goal for decoder: recover *detail* in encoder feature maps before subsampling



Output Going to Image Resolution

- Encoder aggressively pools and subsamples image
- Necessary to capture context information which is necessary for segmentation
- But spatial detail is also necessary

- Goal for decoder: obtain output at image *resolution*
- Goal for decoder: recover *detail* in encoder feature maps before subsampling

- Option 1
  - Use transposed convolution to upsample to image resolution

Normal Convolutions

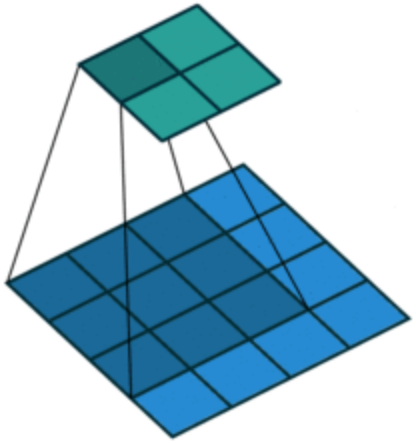
1	2
3	4

☆

1	2	3
4	5	6
7	8	9

=

37	47
67	77



Filter, a

Input, x

1	2	0	3	4	0	0	0	0
0	1	2	0	3	4	0	0	0
0	0	0	1	2	0	3	4	0
0	0	0	0	1	2	0	3	4

A

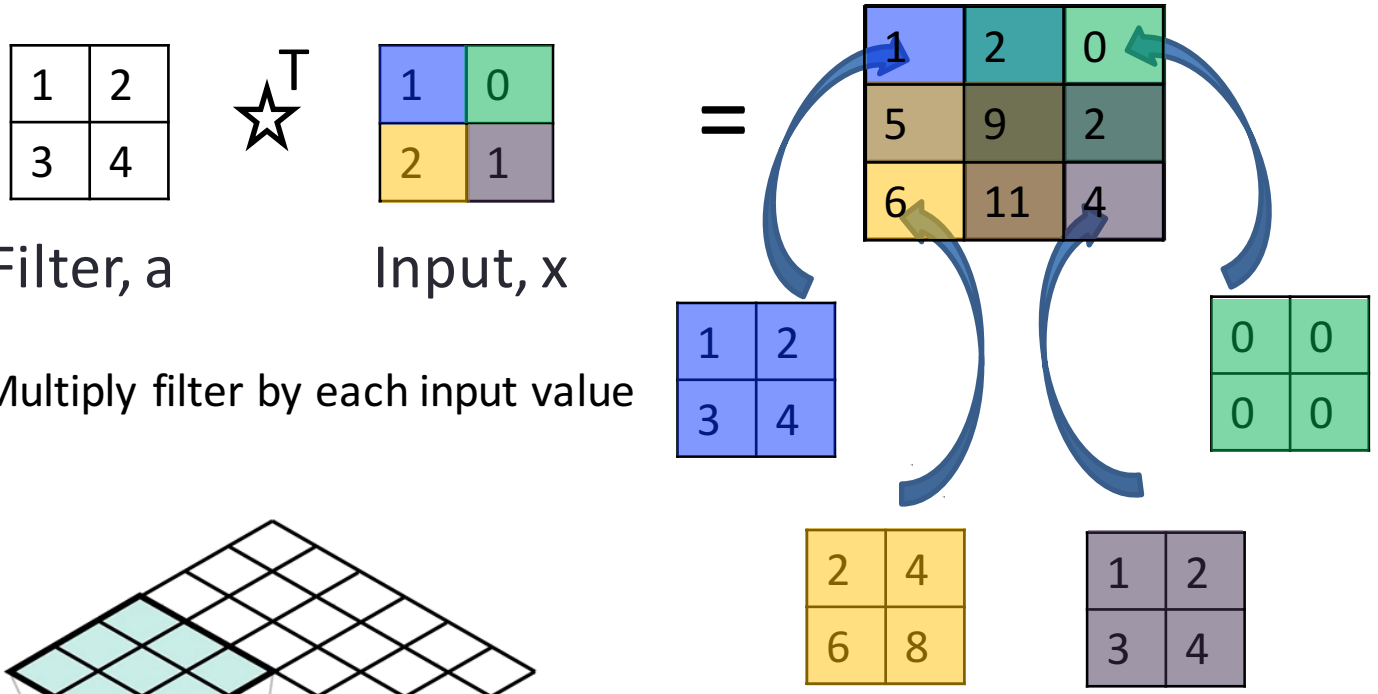
1
2
3
4
5
6
7
8
9

X

=

37
47
67
77

Transposed Convolutions

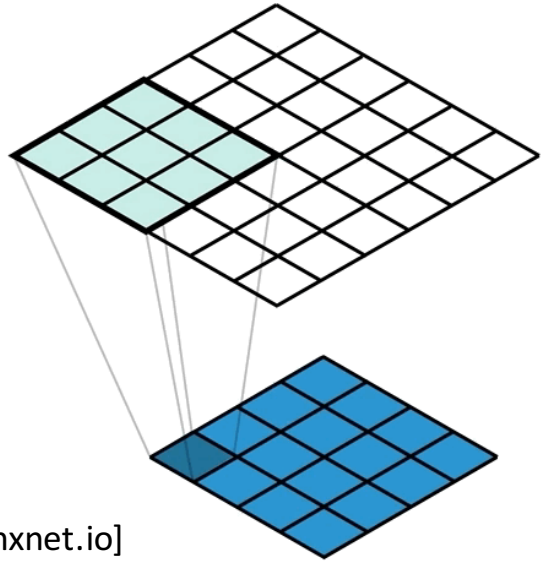


Filter, a

Input, x

Multiply filter by each input value

Tile the scaled filter at output locations
Add overlapping values



[Thom Lane, mxnet.io]

Transposed Convolutions

1	2
3	4

Filter, a

$\overset{T}{\star}$

1	0
2	1

Input, x

$=$

1	2	0
5	9	2
6	11	4

1	0	0	0
2	1	0	0
0	2	0	0
3	0	1	0
4	3	2	1
0	4	0	2
0	0	3	0
0	0	4	3
0	0	0	4

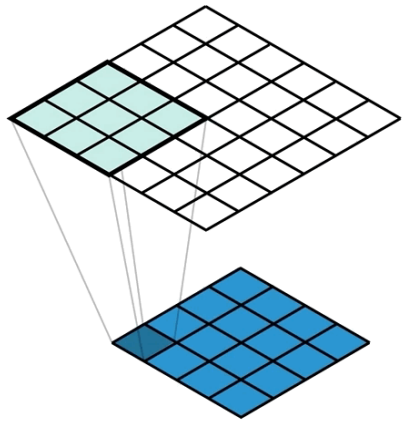
A^T

X

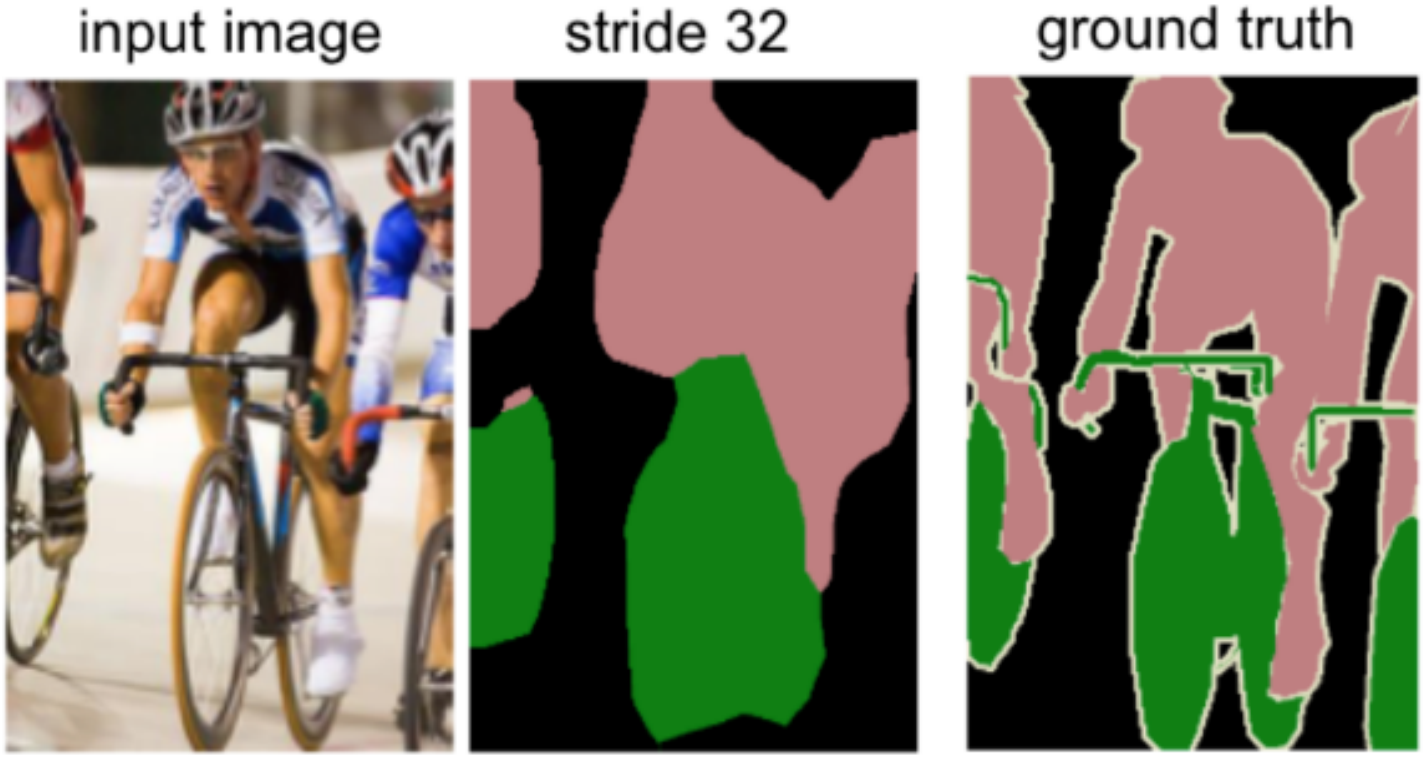
1
0
2
1

$=$

1
2
0
5
9
2
6
11
4



Output Going to Image Resolution



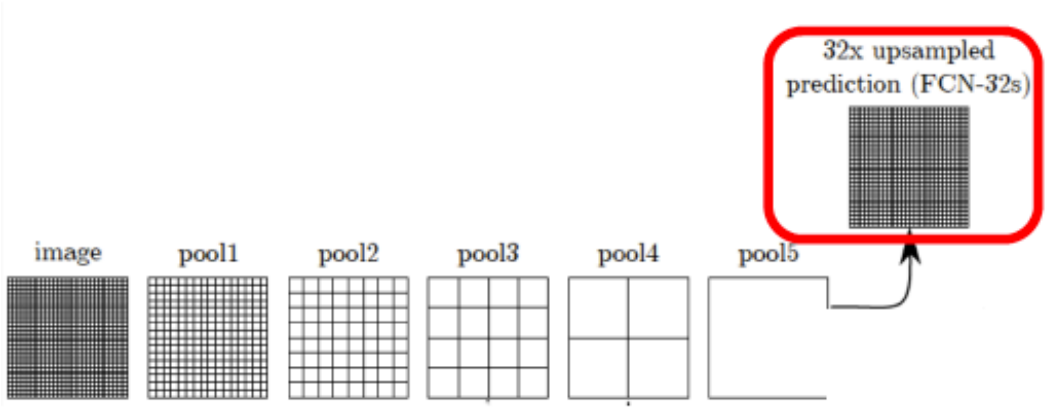
Output Going to Image Resolution

- Encoder aggressively pools and subsamples image
- Necessary to capture context information which is necessary for segmentation
- But spatial detail is also necessary

- Goal for decoder: obtain output at image *resolution*
- Goal for decoder: recover *detail* in encoder feature maps before subsampling

- Option 1
  - Use transposed convolution to upsample to image resolution
  - Concatenate encoder features to upsampled features during decoding

Combine Global and Local Information



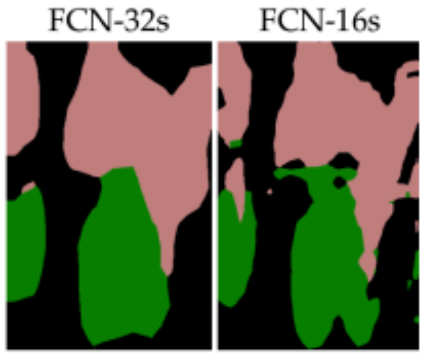
Combine Global and Local Information

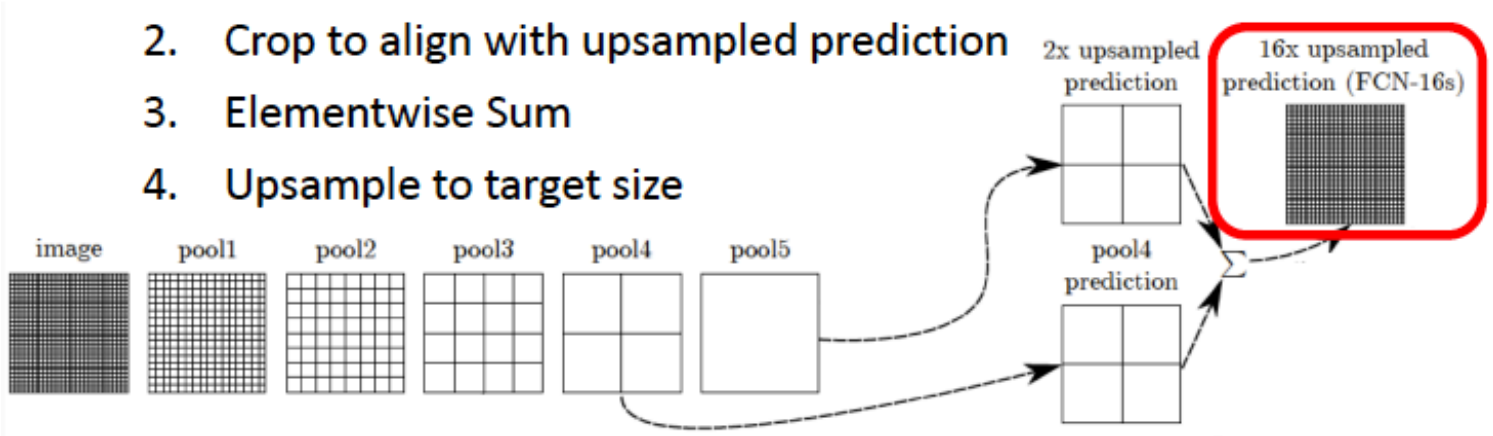
Fuse coarse semantic information with local appearance

Similar to: skip connections.

Process:

1. 1x1 Convolution on pool4 output
2. Crop to align with upsampled prediction
3. Elementwise Sum
4. Upsample to target size

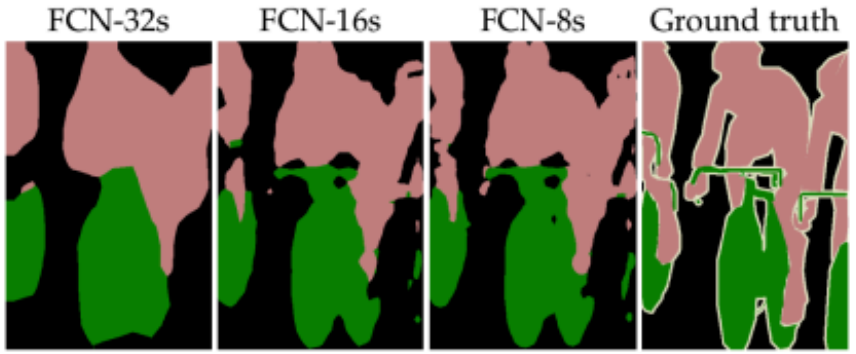


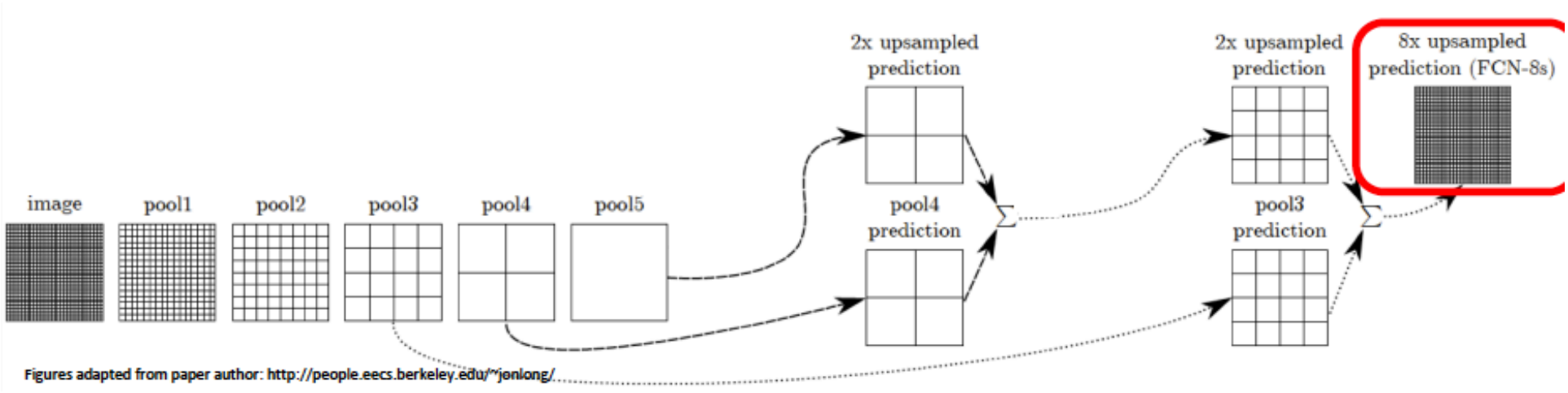


Combine Global and Local Information

Fuse coarse semantic information with local appearance

Similar to: skip connections.





Output Going to Image Resolution

- Encoder aggressively pools and subsamples image
- Necessary to capture context information which is necessary for segmentation
- But spatial detail is also necessary

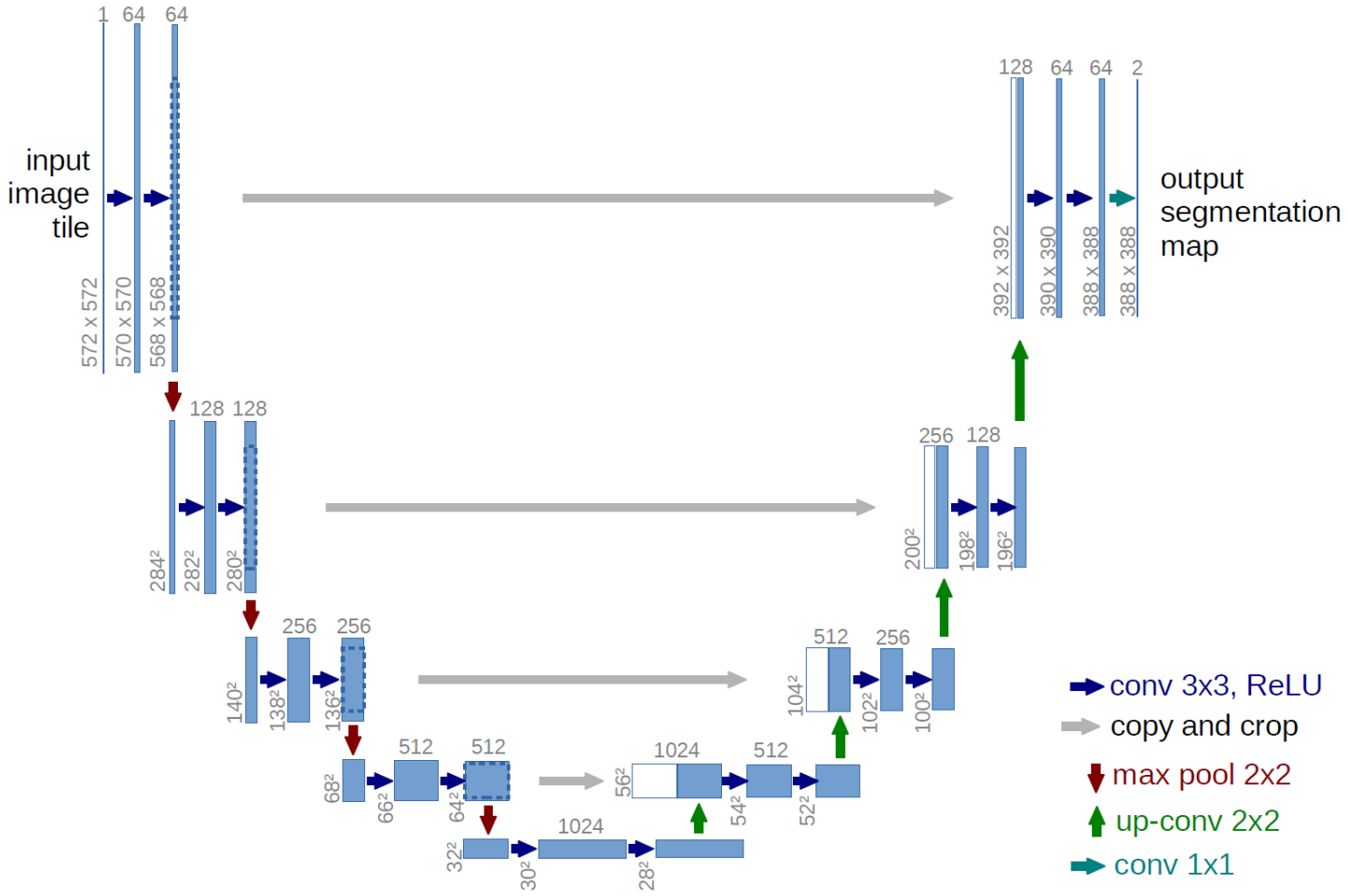
- Goal for decoder: obtain output at image *resolution*
- Goal for decoder: recover *detail* in encoder feature maps before subsampling

- Option 1
  - Use transposed convolution to upsample to image resolution
  - Concatenate encoder features to upsampled features during decoding

- Option 2
  - Similar as above, but more symmetric upsampling to image resolution

U-Net

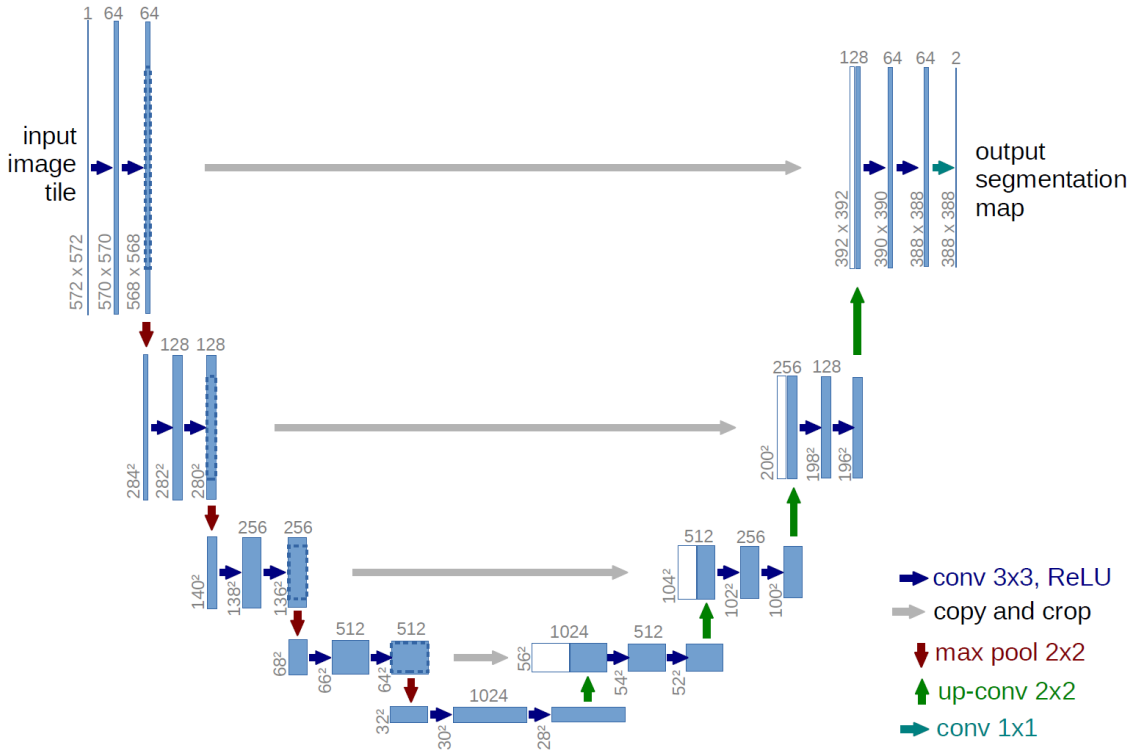
- Similar to FCN, but symmetric architecture



[Ronneberger et al., U-Net]

U-Net

- Encoder: Modules of two 3x3 convolution, followed by 2x2 pooling
- Decoder:
  - Upsample with 2x2 transposed convolution
  - Concatenate feature map from corresponding encoder level
  - Two 3x3 convolutions



[Ronneberger et al., U-Net]

Output Going to Image Resolution

- Encoder aggressively pools and subsamples image
- Necessary to capture context information which is necessary for segmentation
- But spatial detail is also necessary

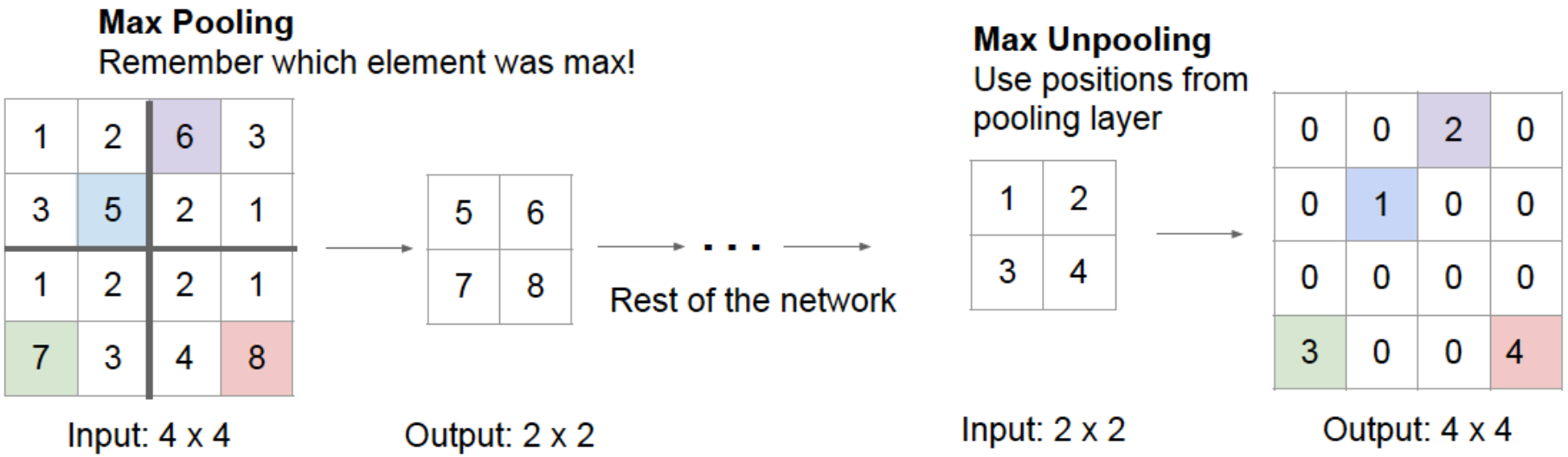
- Goal for decoder: obtain output at image *resolution*
- Goal for decoder: recover *detail* in encoder feature maps before subsampling

- Option 1
  - Use transposed convolution to upsample to image resolution
  - Concatenate encoder features to upsampled features during decoding

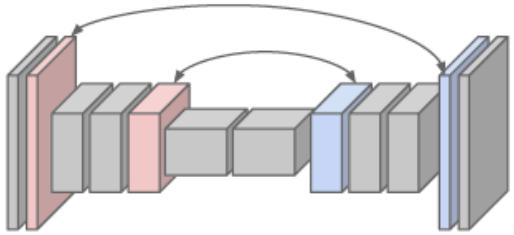
- Option 2
  - Similar as above, but more symmetric upsampling to image resolution

- Option 3
  - Unpool based on encoder locations and convolve to densify

Refinement: Unpooling followed by convolution

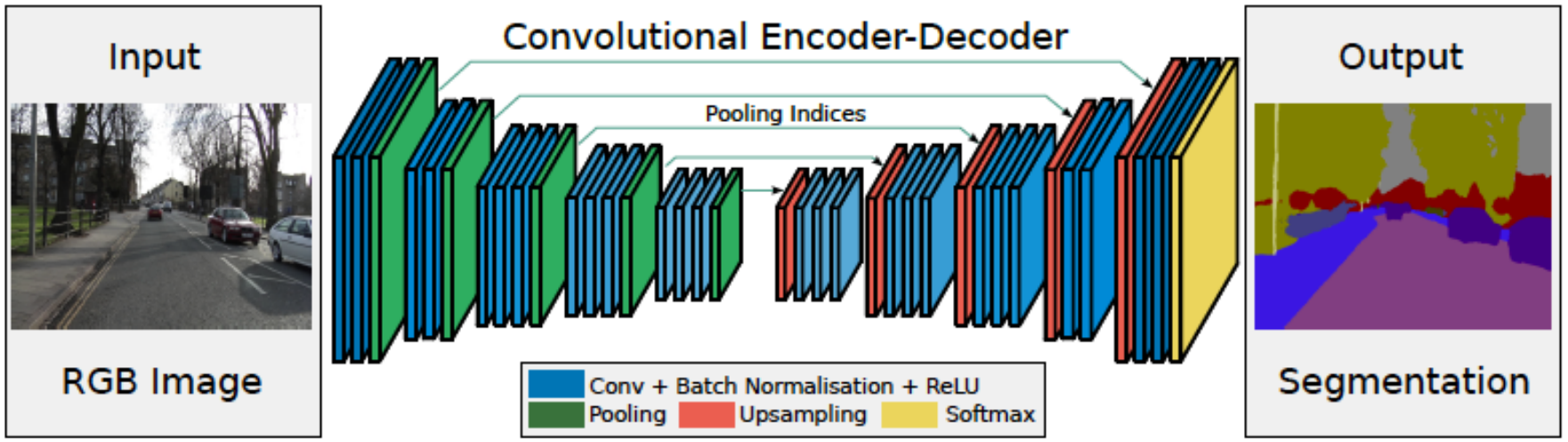


Corresponding pairs of downsampling and upsampling layers



[Figure from: L. Fei-Fei et al., CS 231n]

Output Going to Image Resolution

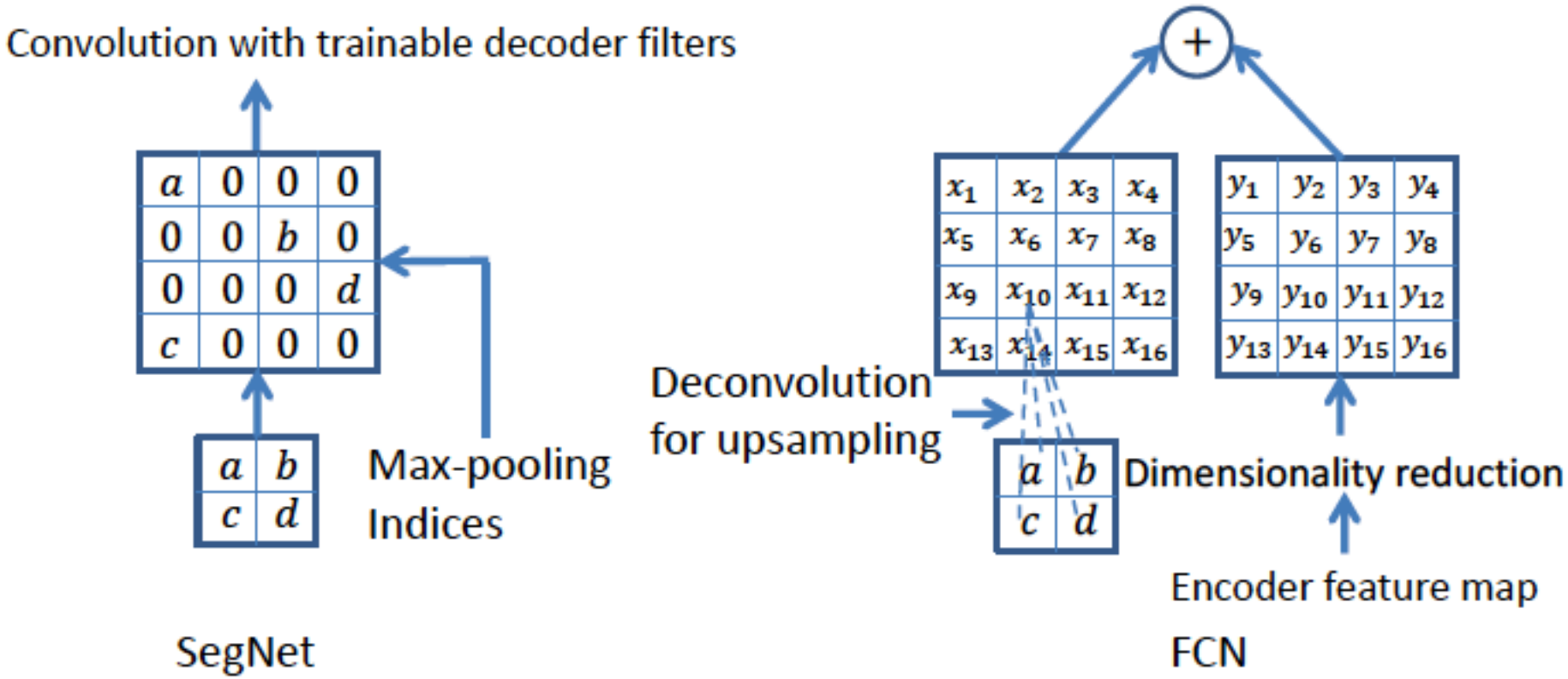


- In U-Net, encoder feature maps need to be stored in memory for concatenation
- In SegNet, only store the indices where max-pool is active
- Store 2x2 pooling index with 2 bits, as opposed to floating point feature map

- Significant memory reduction at inference time

[Badrinarayanan et al., SegNet]

Comparison of two approaches to upsample



[Badrinarayanan et al., SegNet]