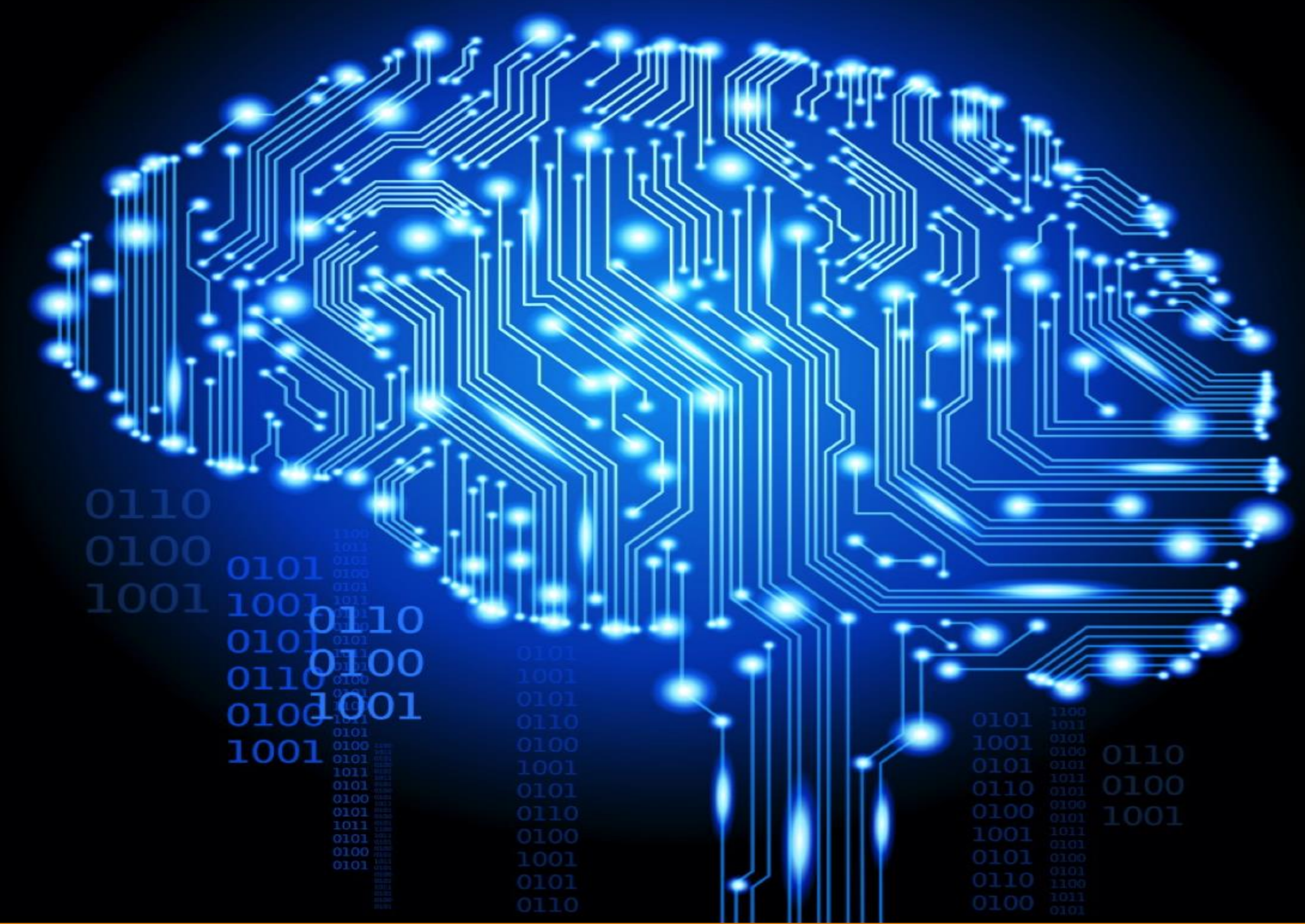


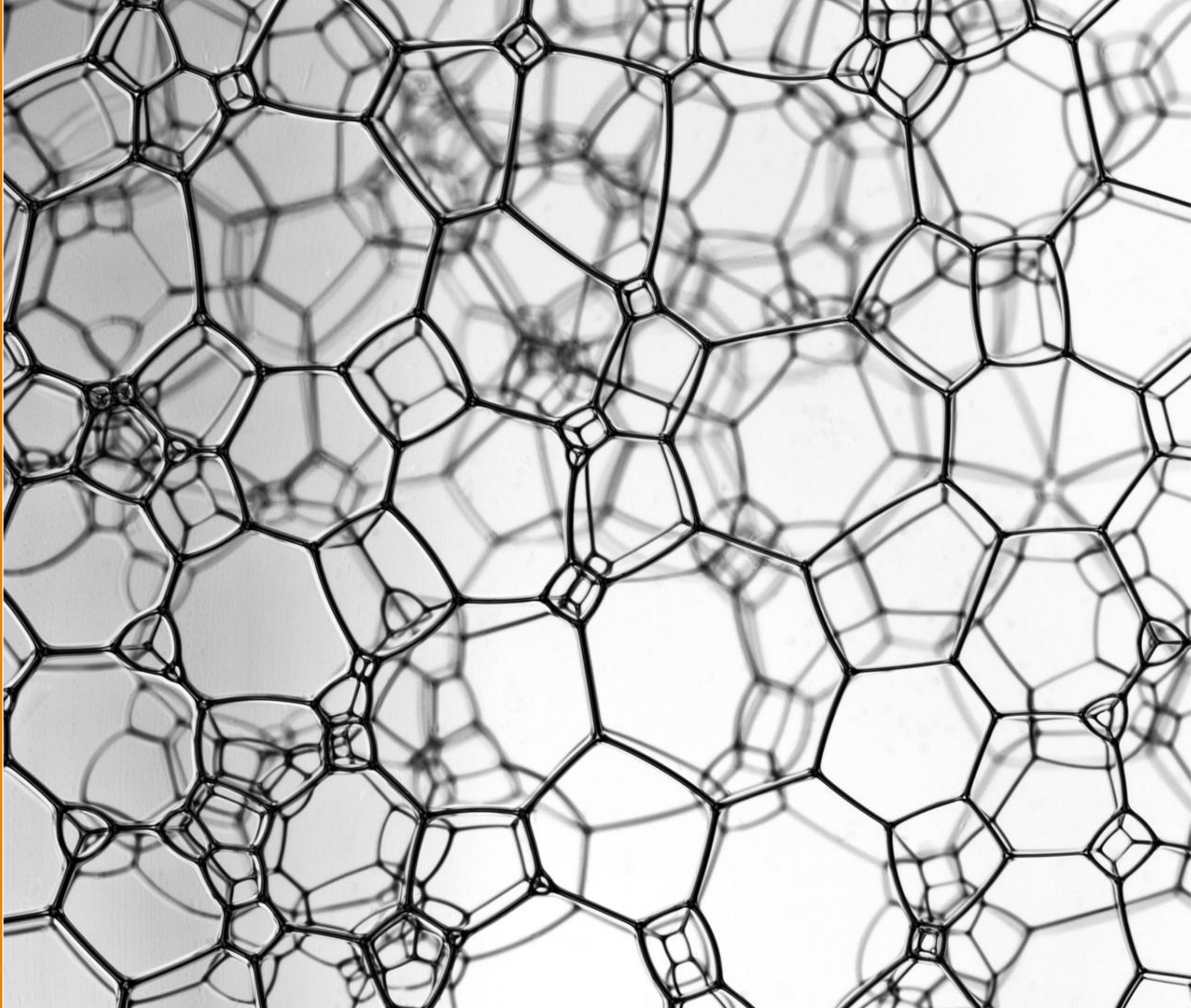
Lecture 11: Structured Deep Learning for Computer Vision

Deep Learning @ UvA

Lecture Overview

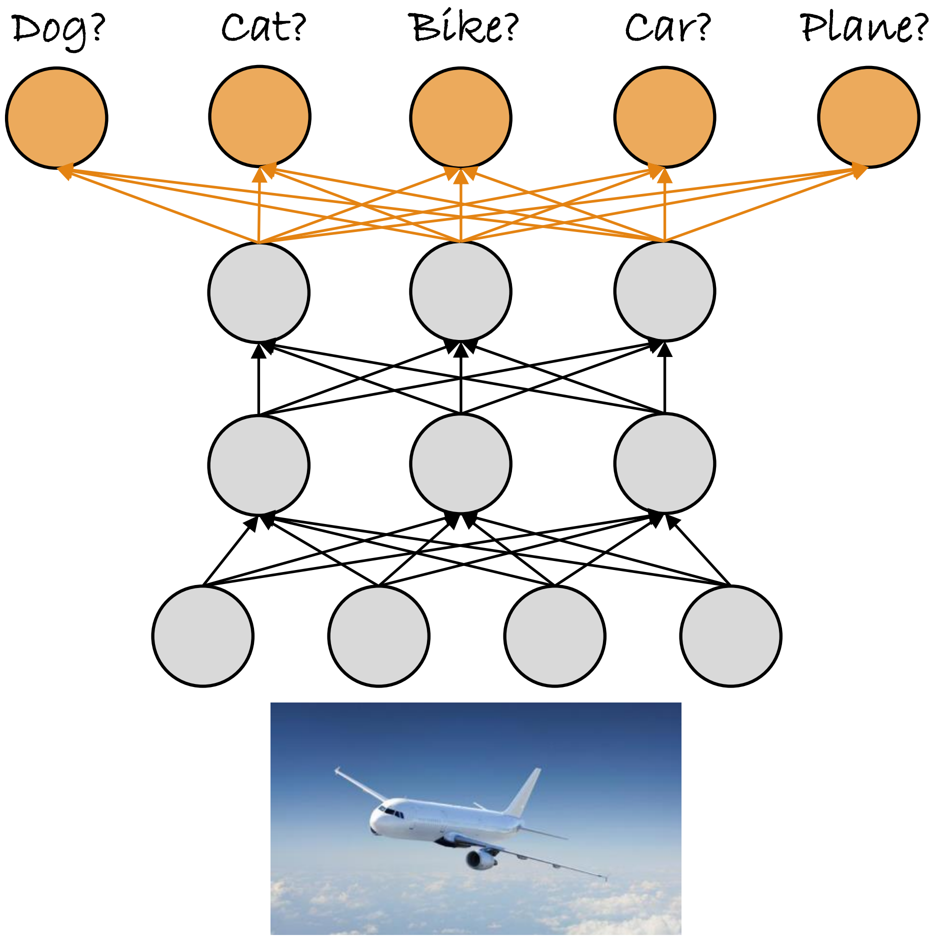
- What is structured prediction?
- Can we repurpose for structured prediction?
- Structured losses on ConvNets
- Multi-task learning with ConvNets

What is structured prediction?



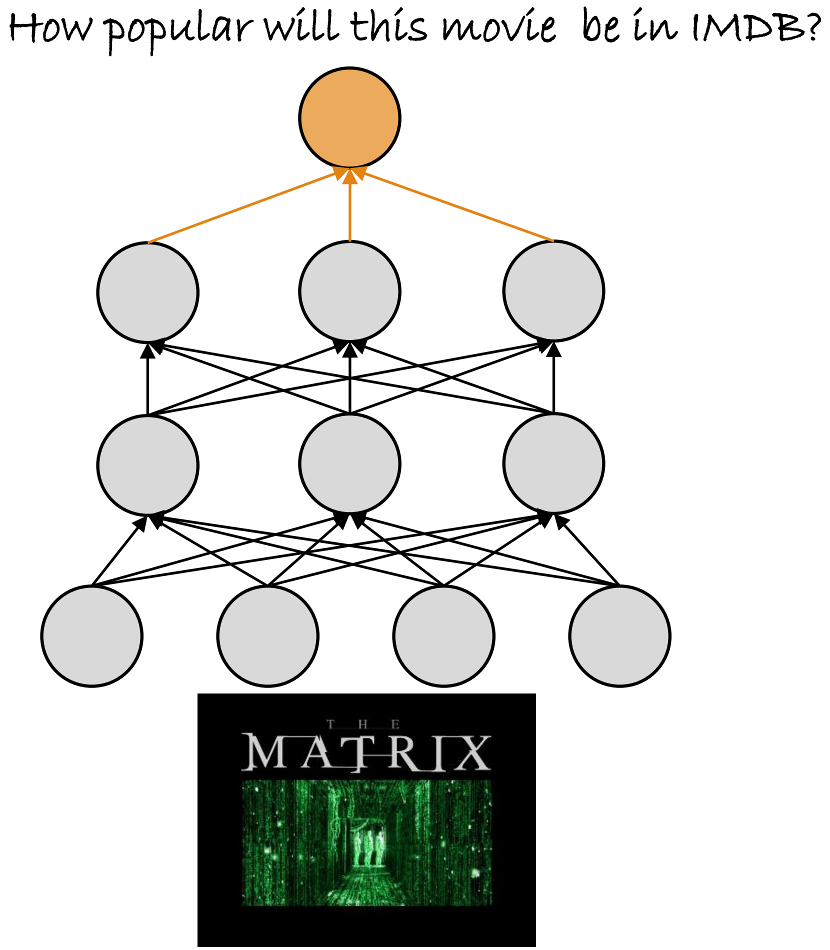
Standard inference

- N-way classification



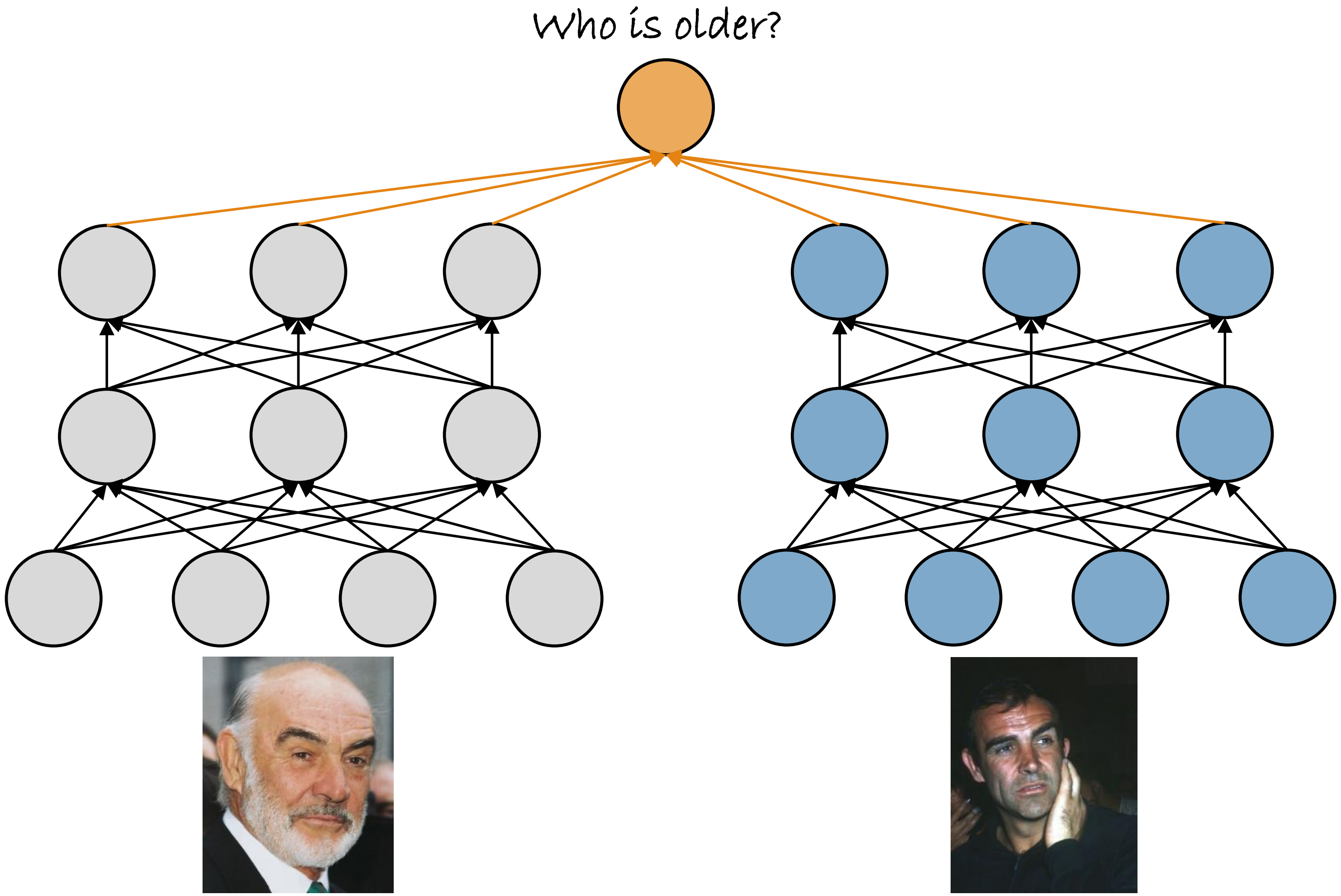
Standard inference

- N-way classification
- Regression



Standard inference

- N-way classification
- Regression
- Ranking
- …



What do they have in common?

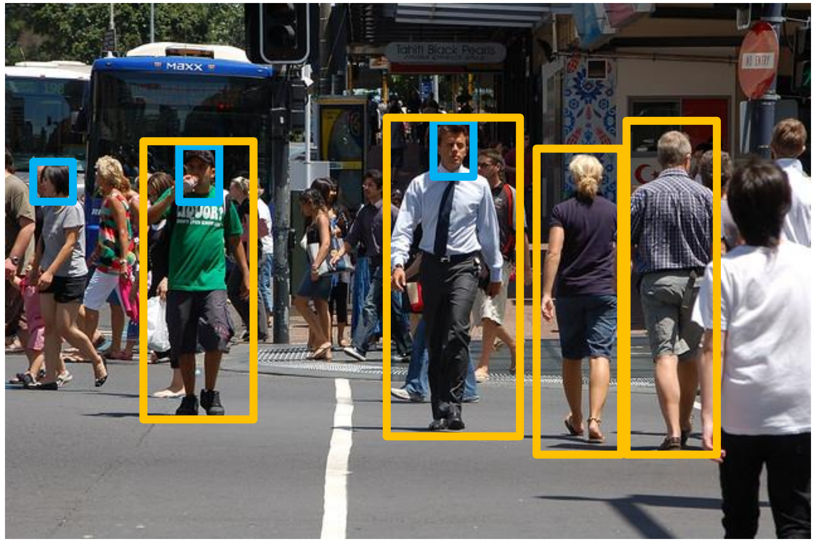
What do they have in common?

- They all make “single value” predictions
- Are there tasks where outputs are somehow correlated?
- Is there some structure in this output correlations?
- How can we predict such structures? → Structured prediction

Examples?

Object detection

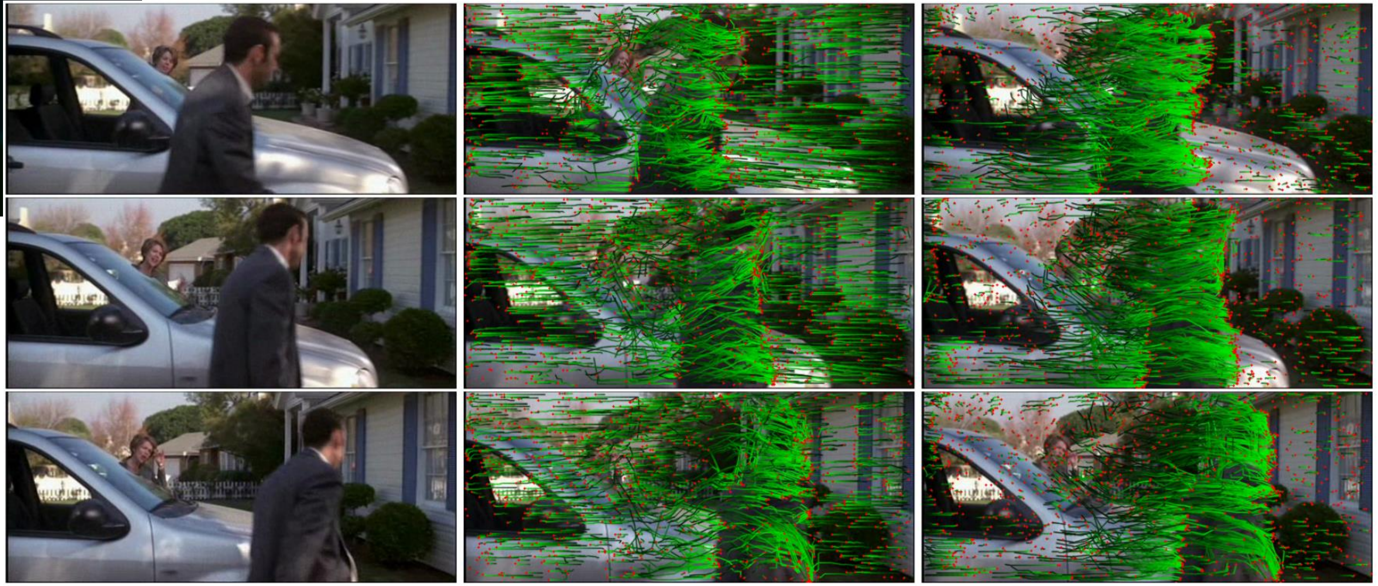
- Predict a box around an object
- Images
  - Spatial location → b(ounding) box
- Videos
  - Spatio-temporal location → bbox@t, bbox@t+1, …



Object Segmentation

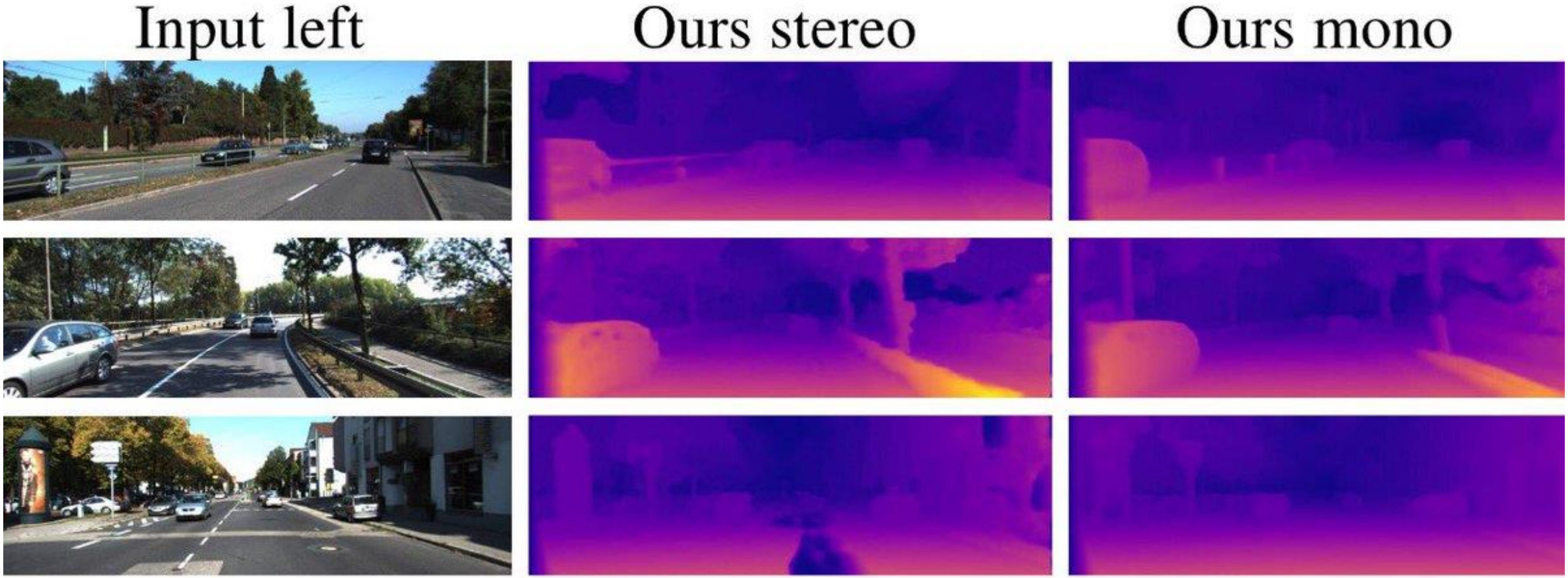
Image | Class map | Instance map | Part map | Part map (high level)

Optical flow & motion estimation



(a) Consecutive frames (b) Trajectories from Optical Flow (c) ω-trajectories

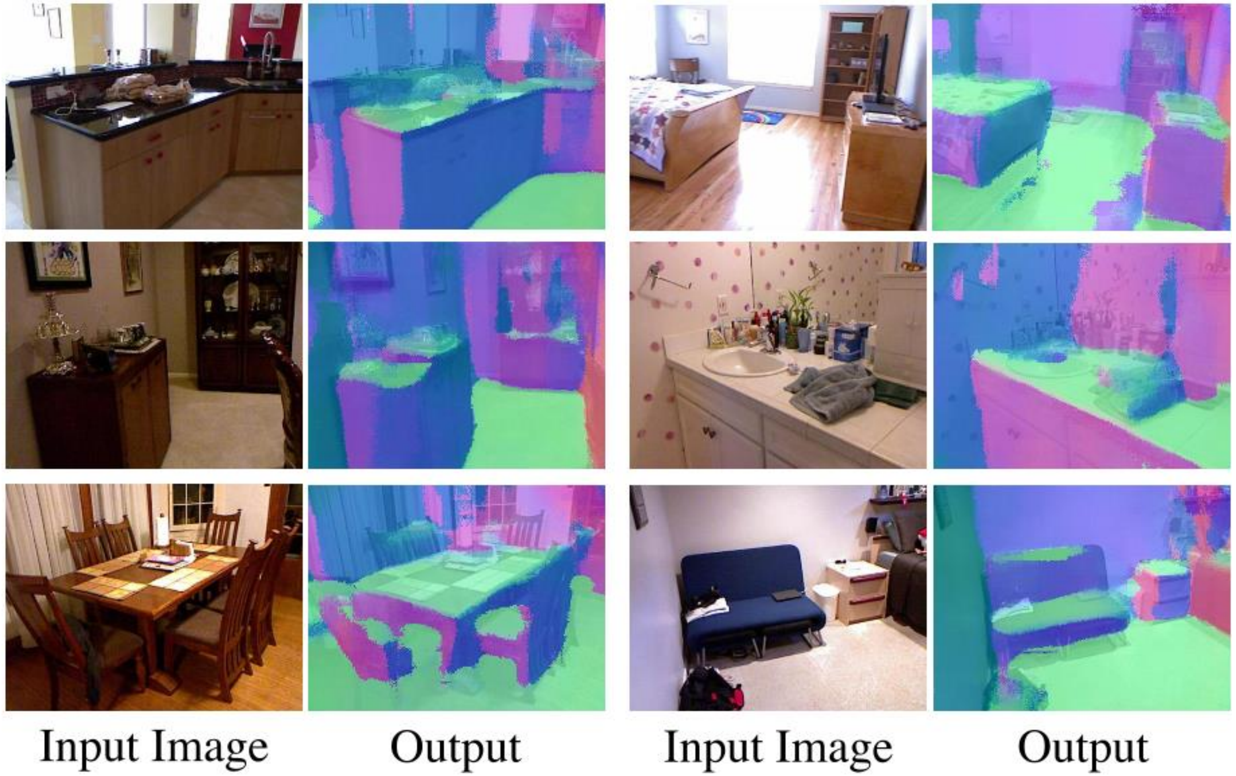
Depth estimation

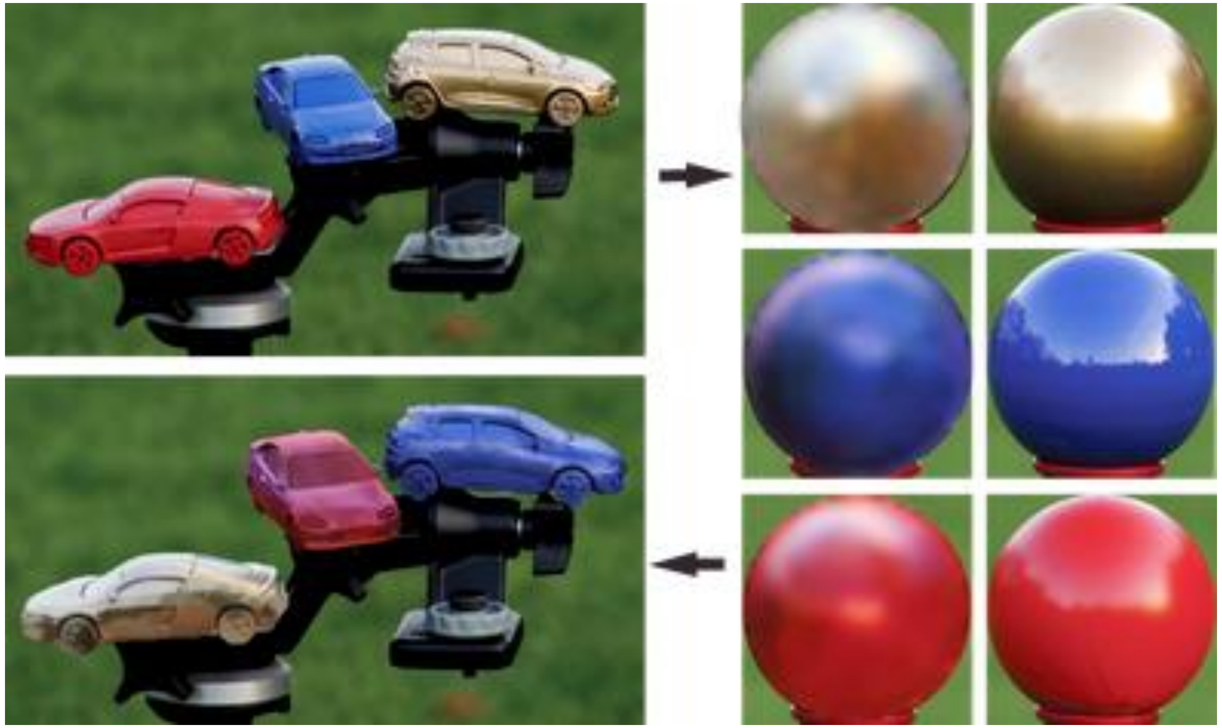


Godard et al., Unsupervised Monocular Depth Estimation with Left-Right Consistency, 2016

Normals and reflectance estimation

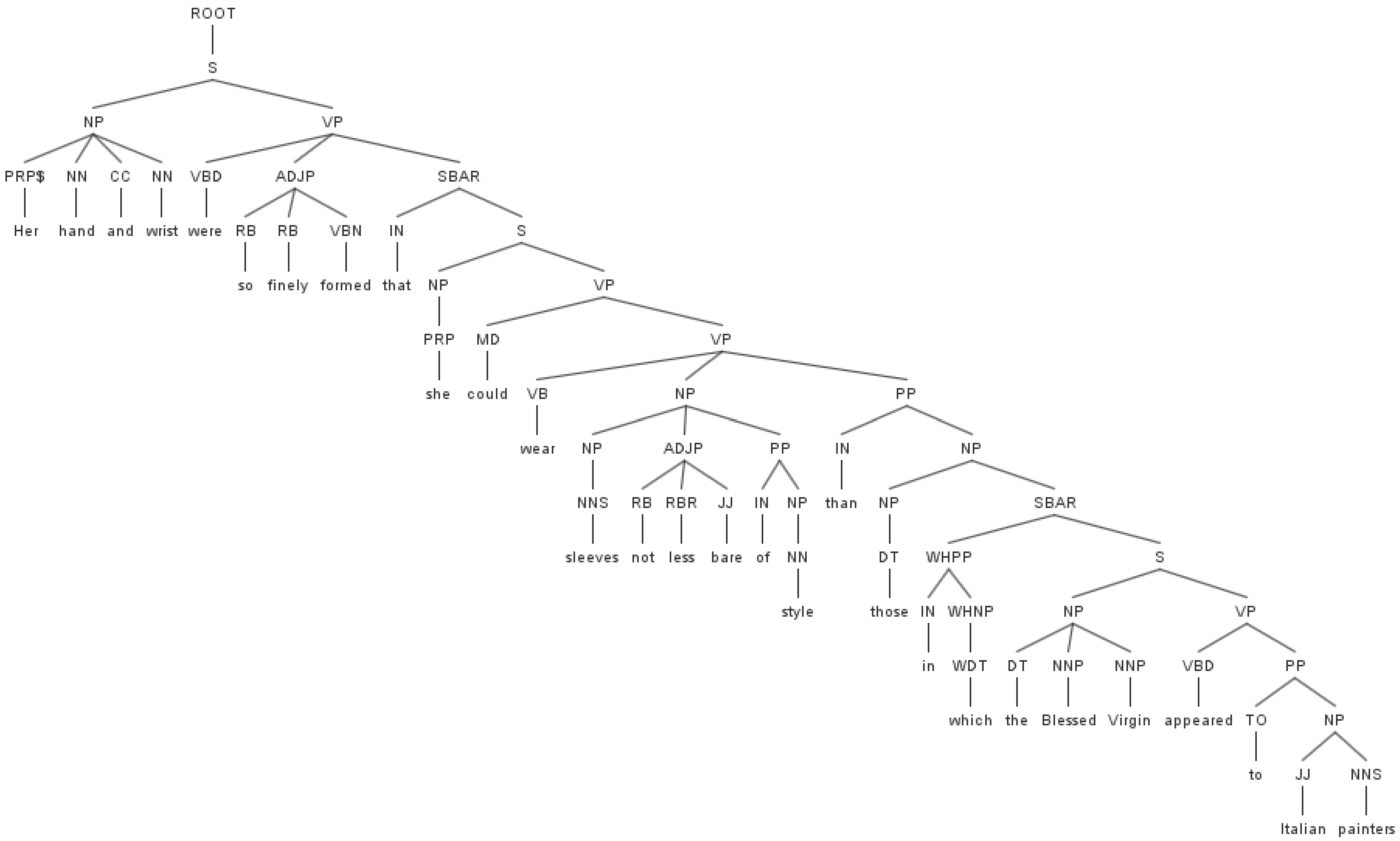
Wang et al., Designing deep networks for surface normal estimation, 2015



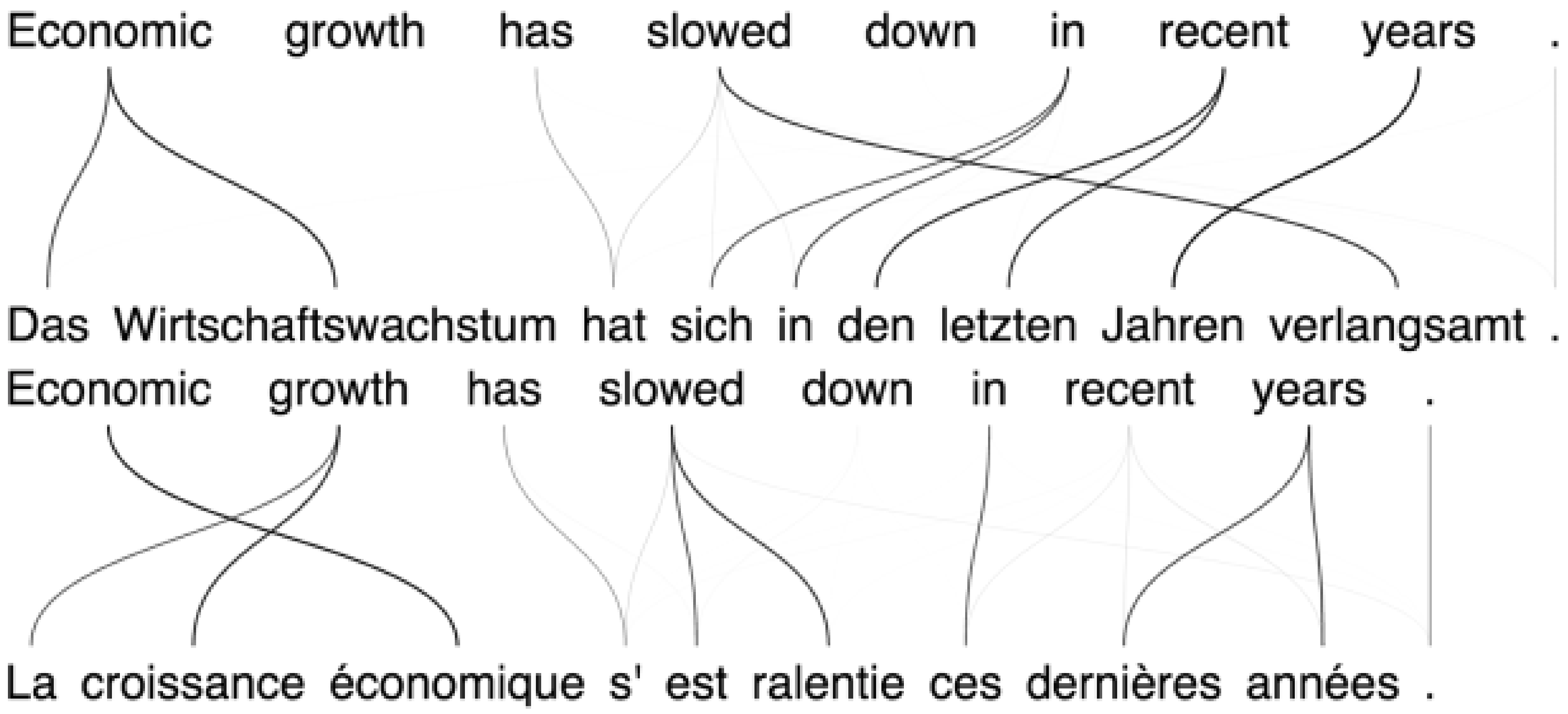


Rematas et al., Deep Reflectance Maps, 2016

Sentence parsing



Machine translation



And many more

- Speech synthesis
- Captioning
- Robot control
- Pose estimation
- …

What is common?

What is common?

- Prediction goes beyond asking for “single values”
- Outputs are complex and output dimensions correlated

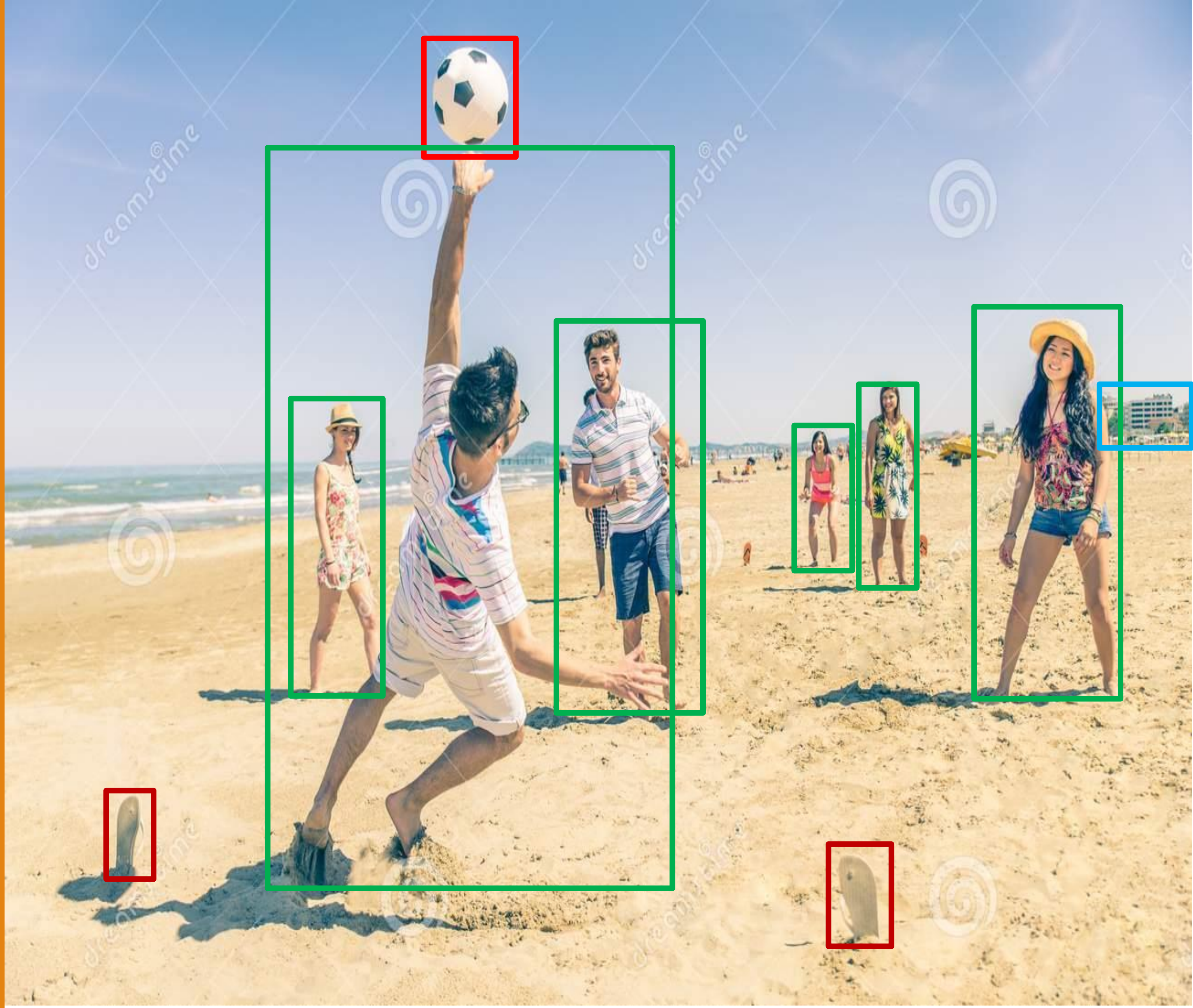
Structured prediction

- Prediction goes beyond asking for “single values”
- Outputs are complex and output dimensions correlated
- Output dimensions have latent structure
- Can we make deep networks to return **structured predictions?**

Structured prediction

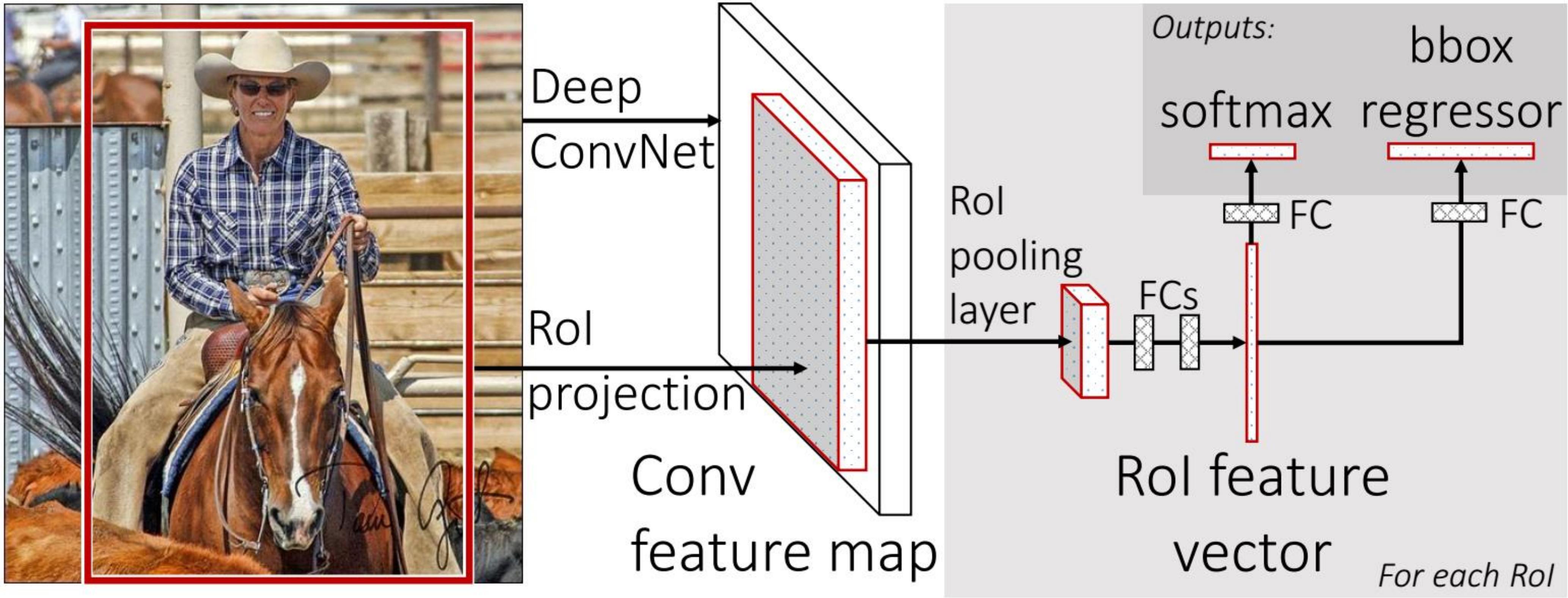
- Prediction goes beyond asking for “single values”
- Outputs are complex and output dimensions correlated
- Output dimensions have latent structure
- Can we make deep networks to return **structured predictions?**

ConvNets for structured prediction



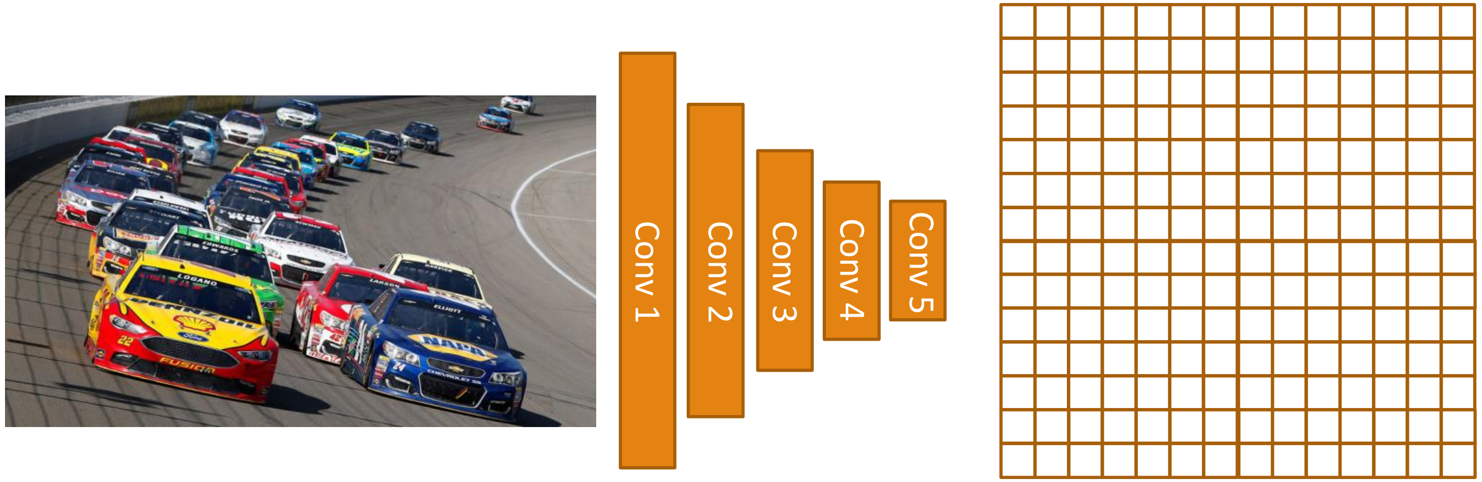
Sliding window on feature maps

- SPPnet [He2014]
- Fast R-CNN [Girshick2015]



Fast R-CNN: Steps

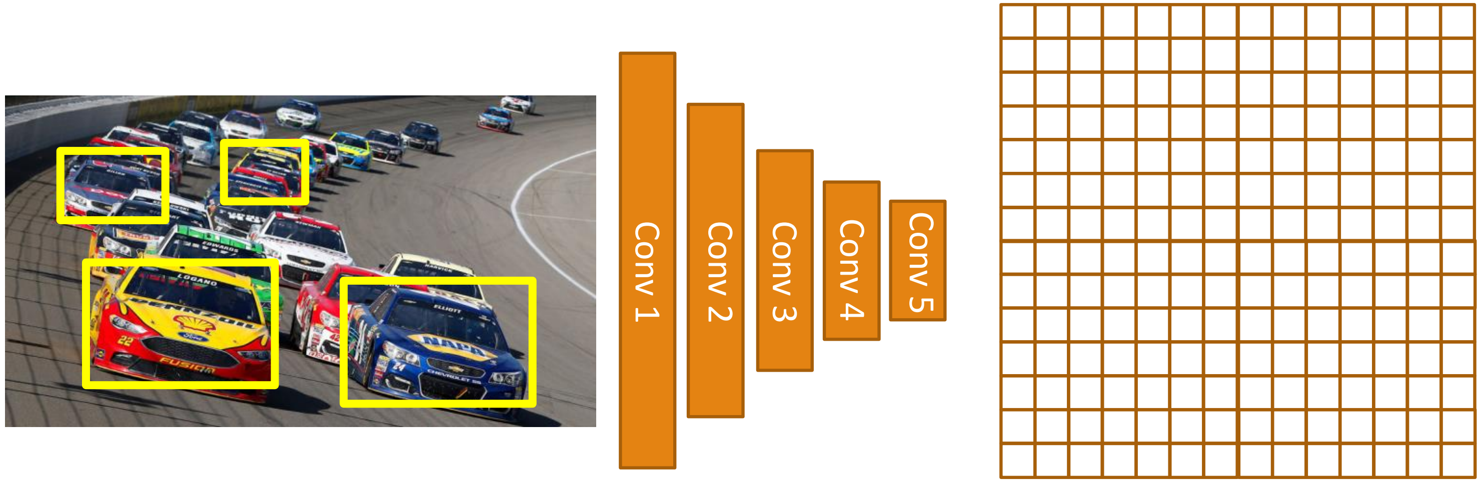
- Process the whole image up to conv5



Conv 5 feature map

Fast R-CNN: Steps

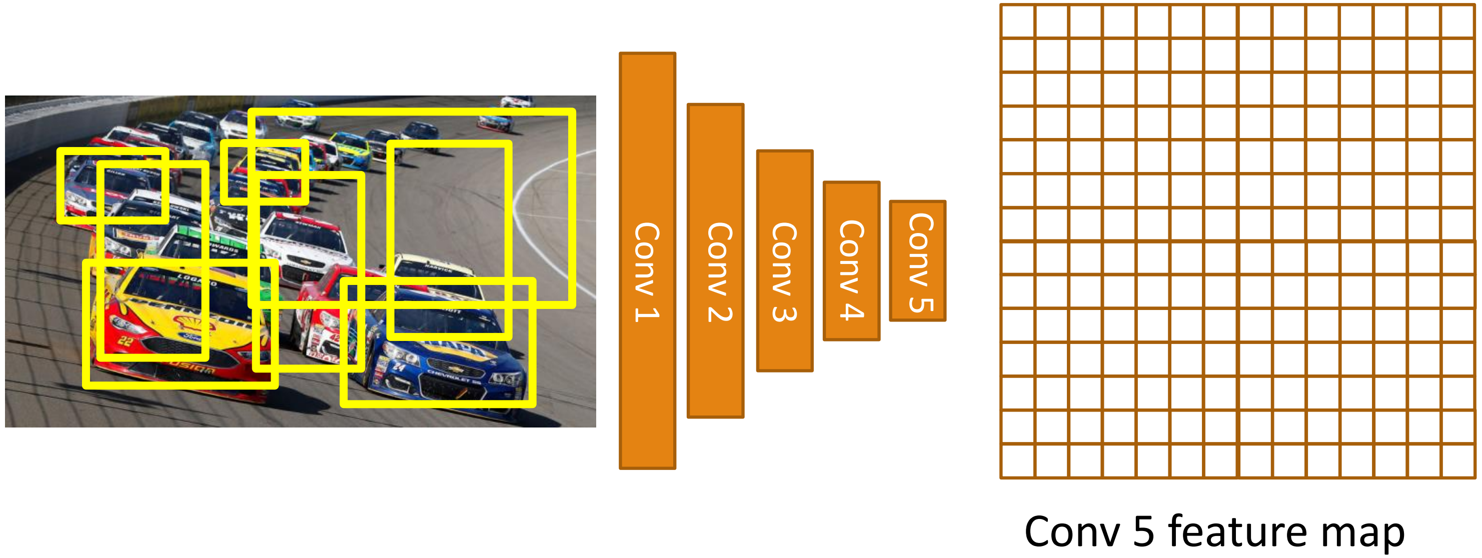
- Process the whole image up to conv5
- Compute possible locations for objects



Conv 5 feature map

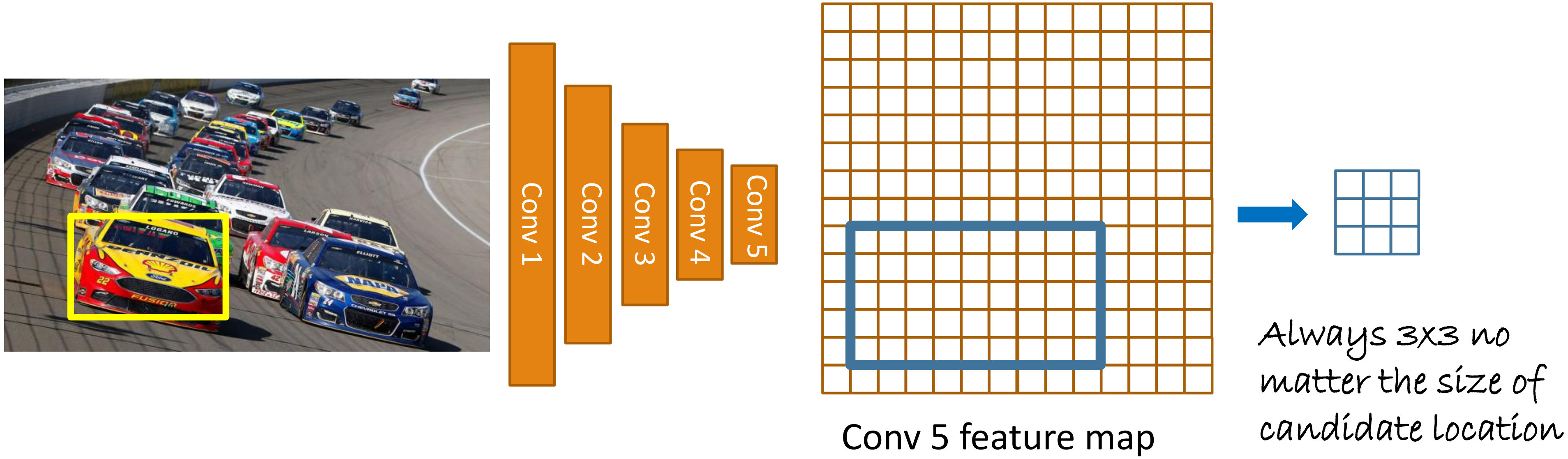
Fast R-CNN: Steps

- Process the whole image up to conv5
- Compute possible locations for objects (some correct, most wrong)



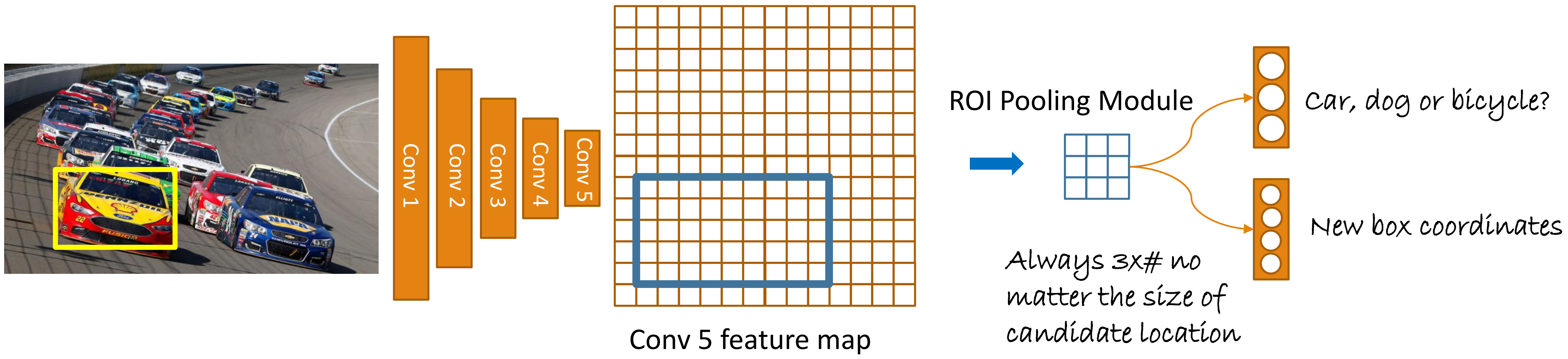
Fast R-CNN: Steps

- Process the whole image up to conv5
- Compute possible locations for objects
- Given single location → ROI pooling module extracts fixed length feature



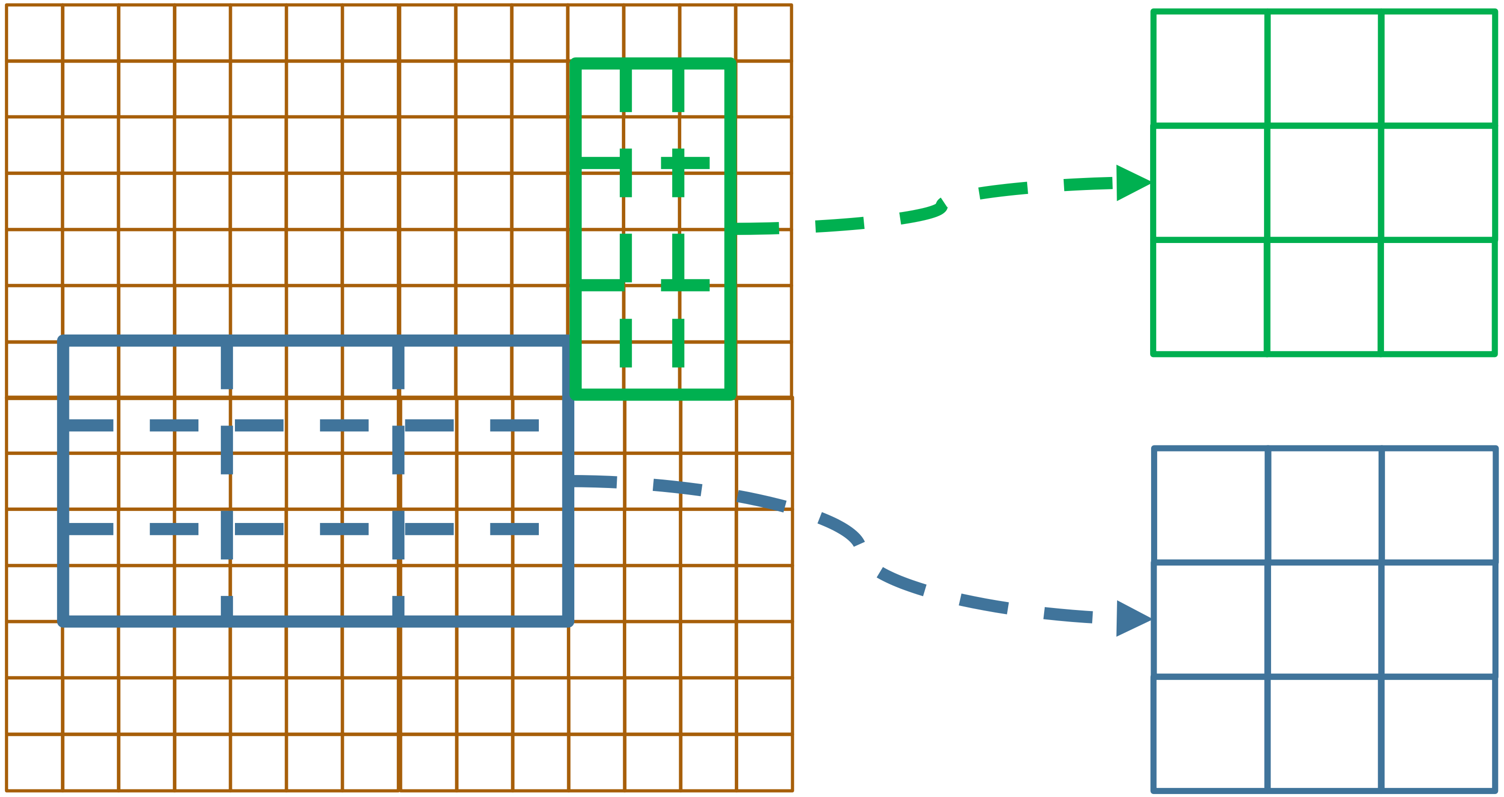
Fast R-CNN: Steps

- Process the whole image up to conv5
- Compute possible locations for objects
- Given single location → ROI pooling module extracts fixed length feature
- Connect to two final layers, 1 for classification, 1 for box refinement



Region-of-Interest (ROI) Pooling Module

- Divide feature map in TxT cells
  - The cell size will change depending on the size of the candidate location

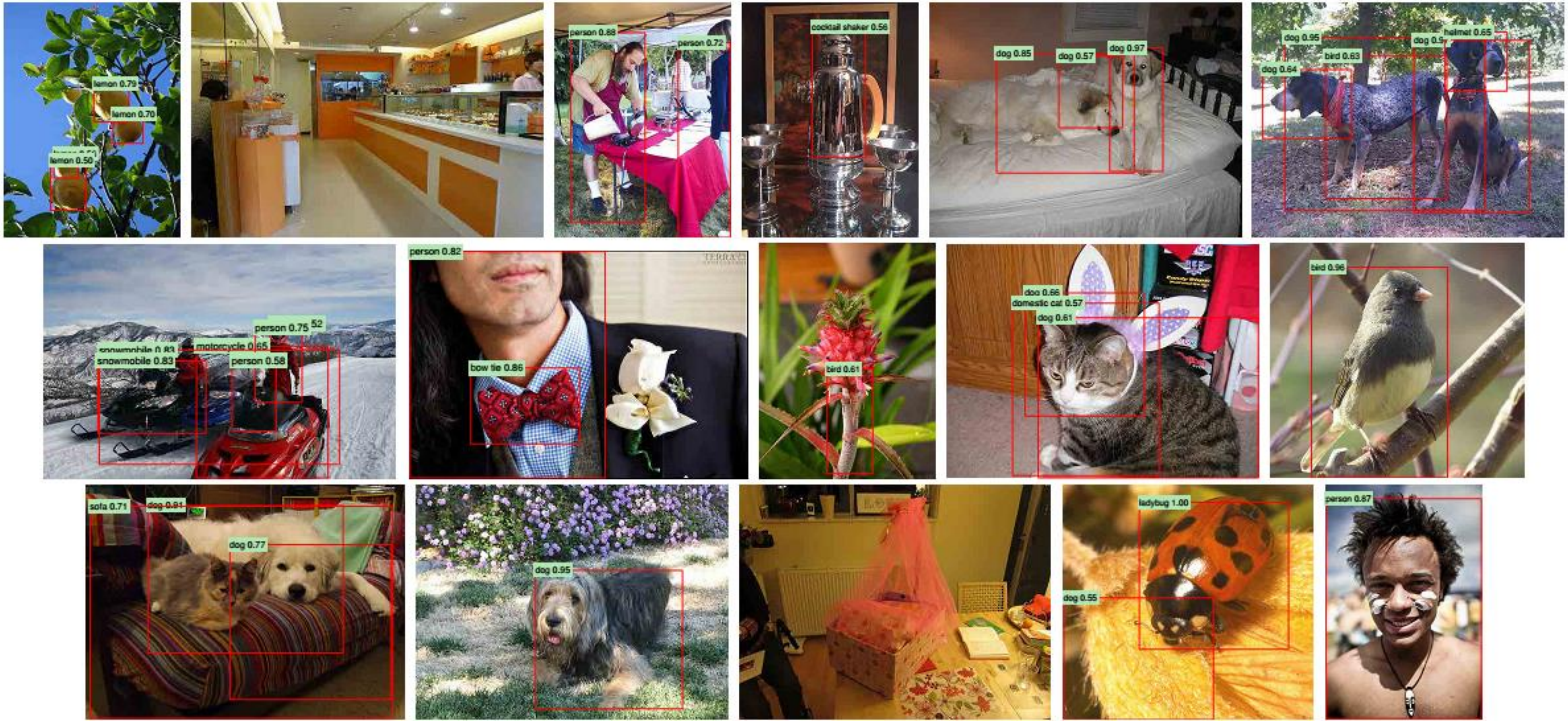


Always 3x3 no matter the size of candidate location

Smart fine-tuning

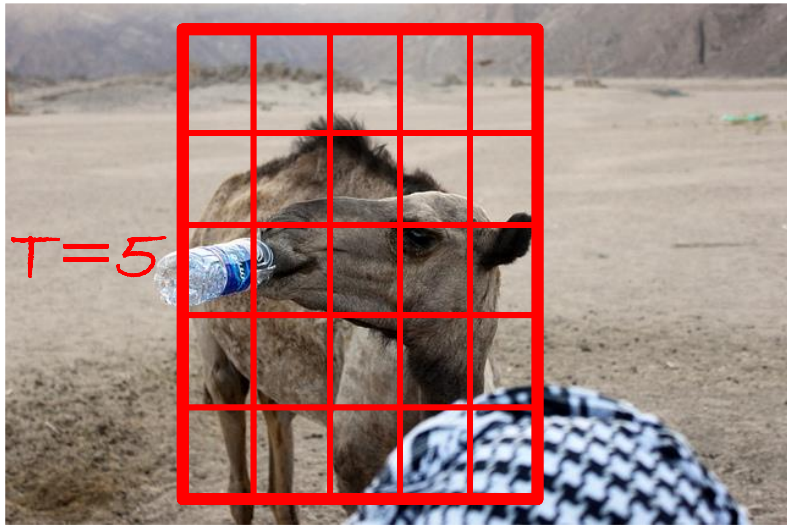
- Normally samples in a mini-batch completely random
- Instead, organize mini-batches by ROIs
- 1 mini-batch = N (images) $\times \frac{R}{N}$ (candidate locations)
- Feature maps shared → training speed-up by a factor of $\frac{R}{N}$
- Mini-batch samples might be correlated
  - In Fast R-CNN that was not observed

Some results



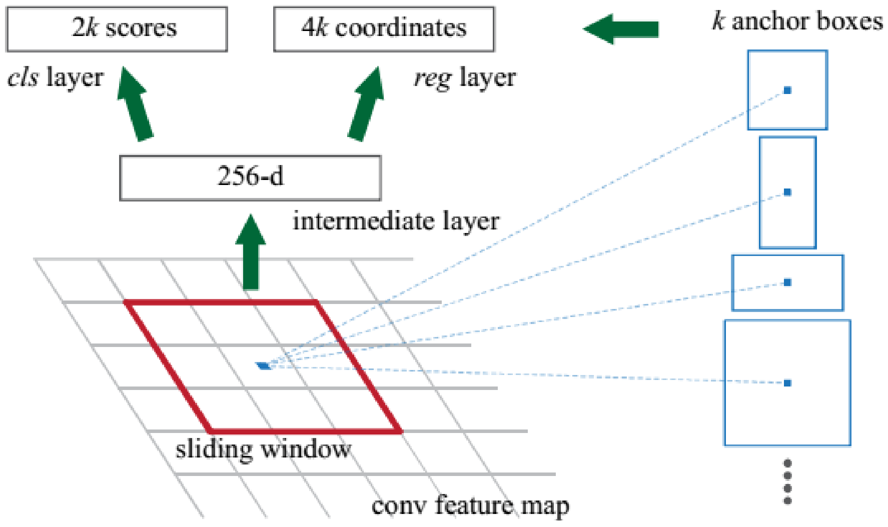
Fast-RCNN

- Reuse convolutions for different candidate boxes
  - Compute feature maps only once
- Region-of-Interest pooling
  - Define stride relatively → box width divided by predefined number of "poolings" T
  - Fixed length vector
- End-to-end training!
- (Very) Accurate object detection
- (Very) Faster
  - Less than a second per image
- External box proposals needed



Faster R-CNN [Girshick2016]

- Fast R-CNN → external candidate locations
- Faster R-CNN → deep network proposes candidate locations
- Slide the feature map → k anchor boxes per slide



Region Proposal Network

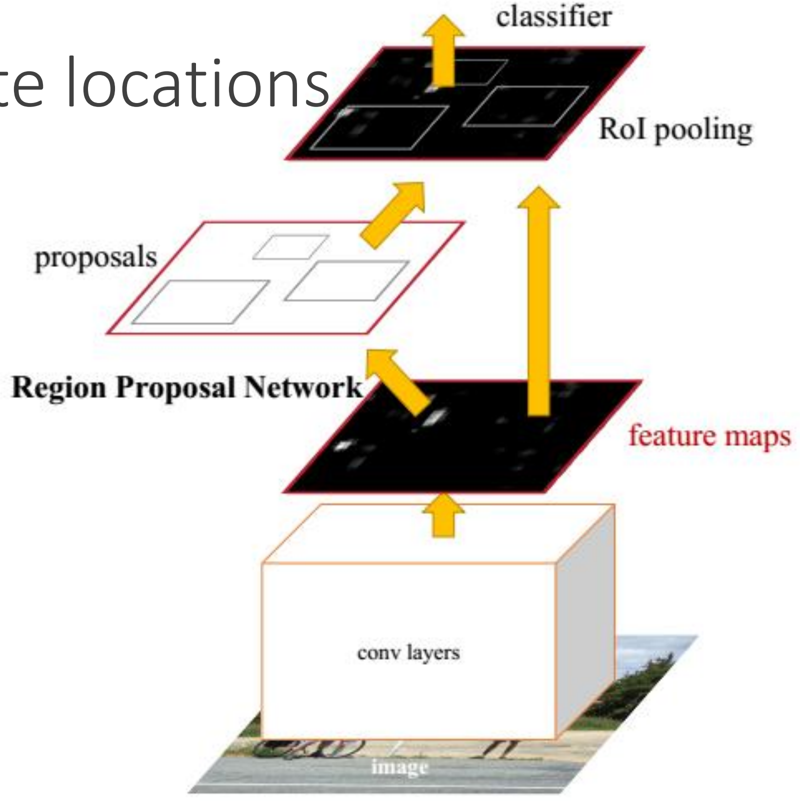
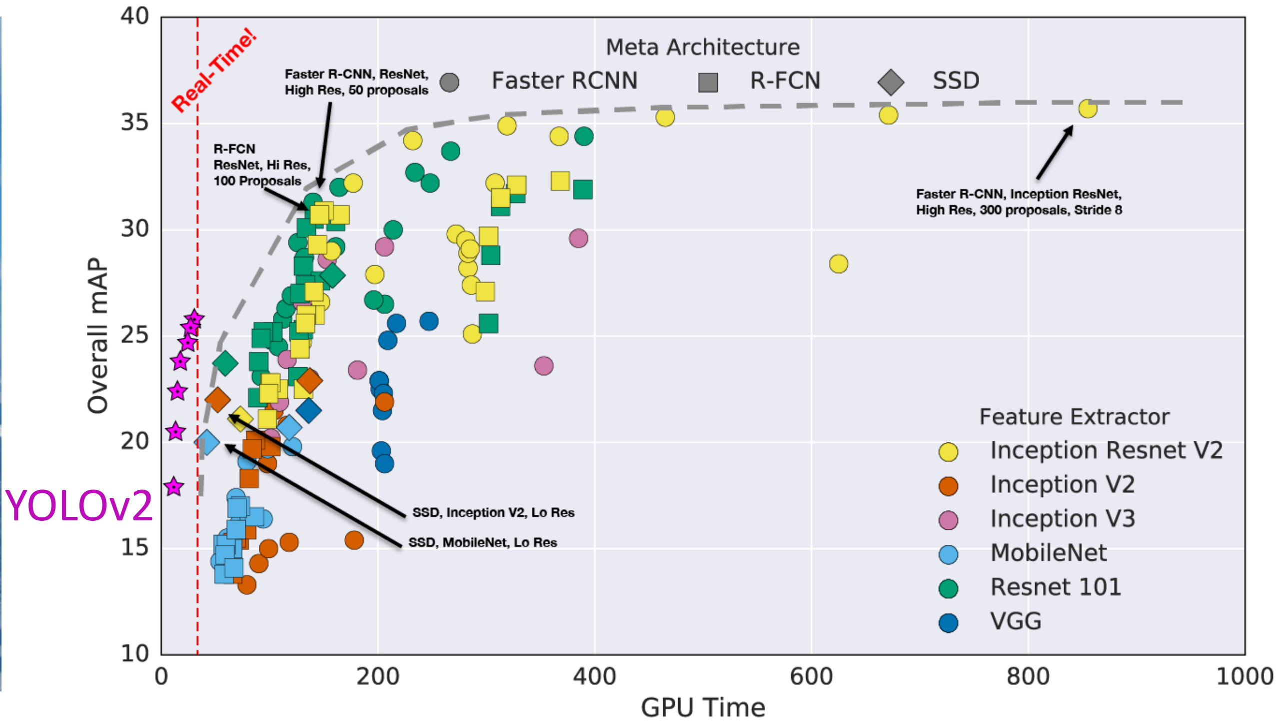


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

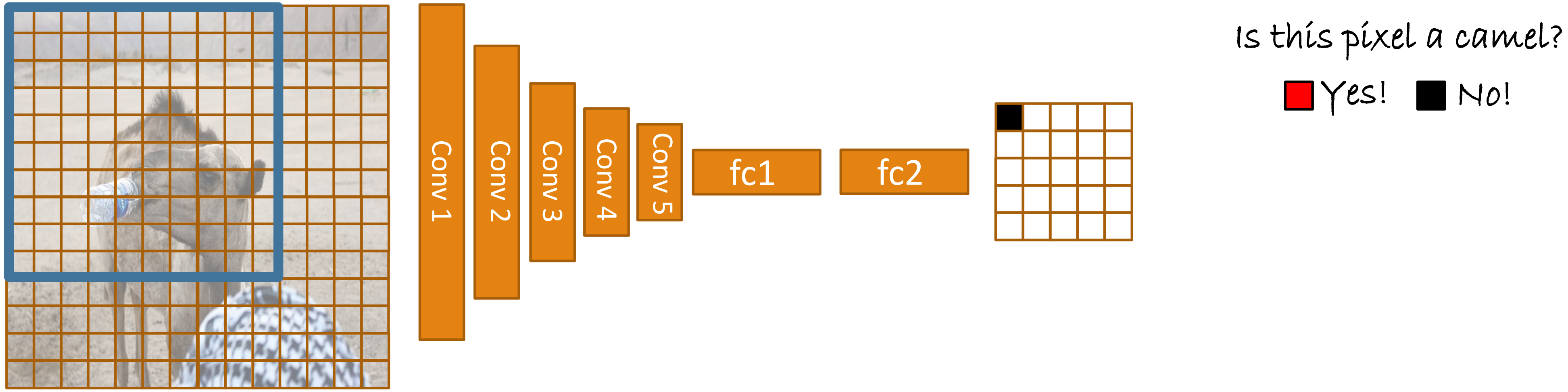
Even better

- YOLO9000: https://www.youtube.com/watch?v=yQwfDxBMtXg
- SSD detector



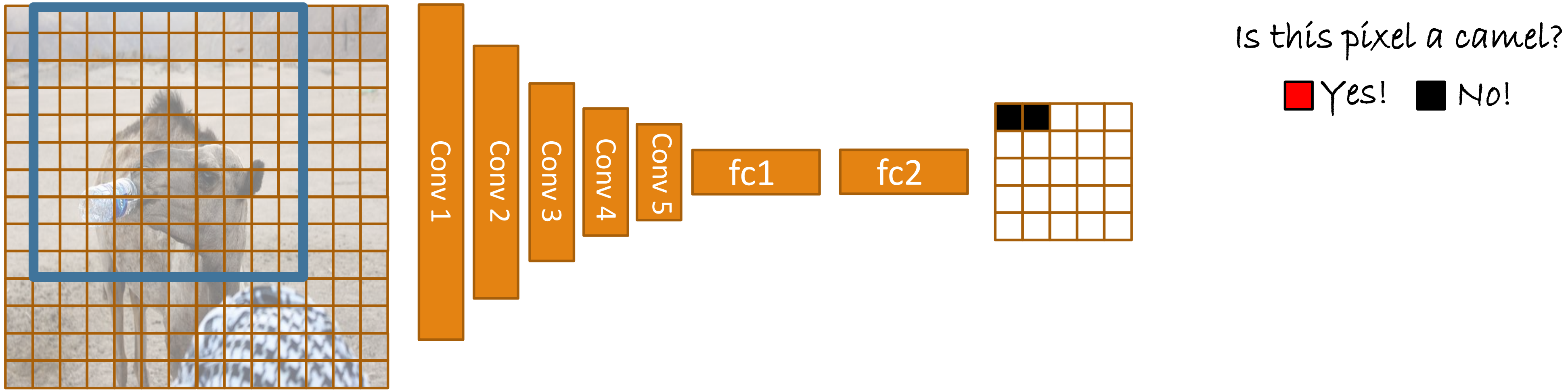
Going Fully Convolutional [LongCVPR2014]

- Image larger than network input → slide the network



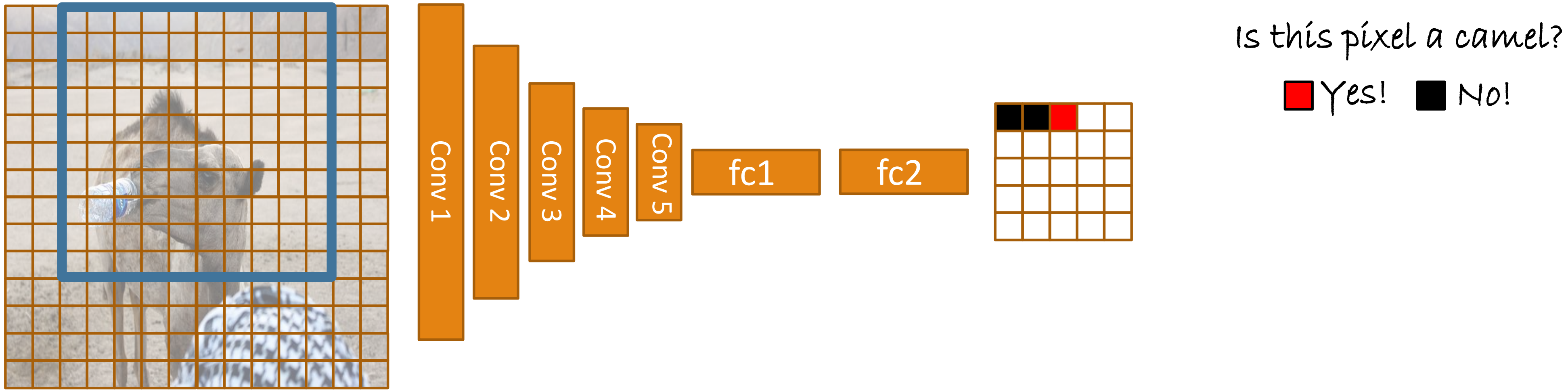
Going Fully Convolutional [LongCVPR2014]

- Image larger than network input → slide the network



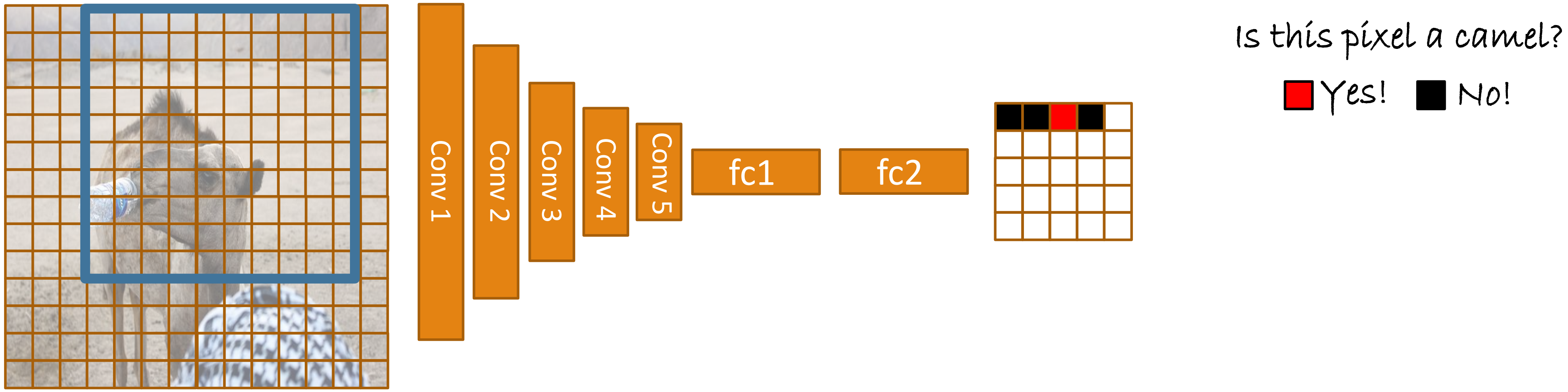
Going Fully Convolutional [LongCVPR2014]

- Image larger than network input → slide the network



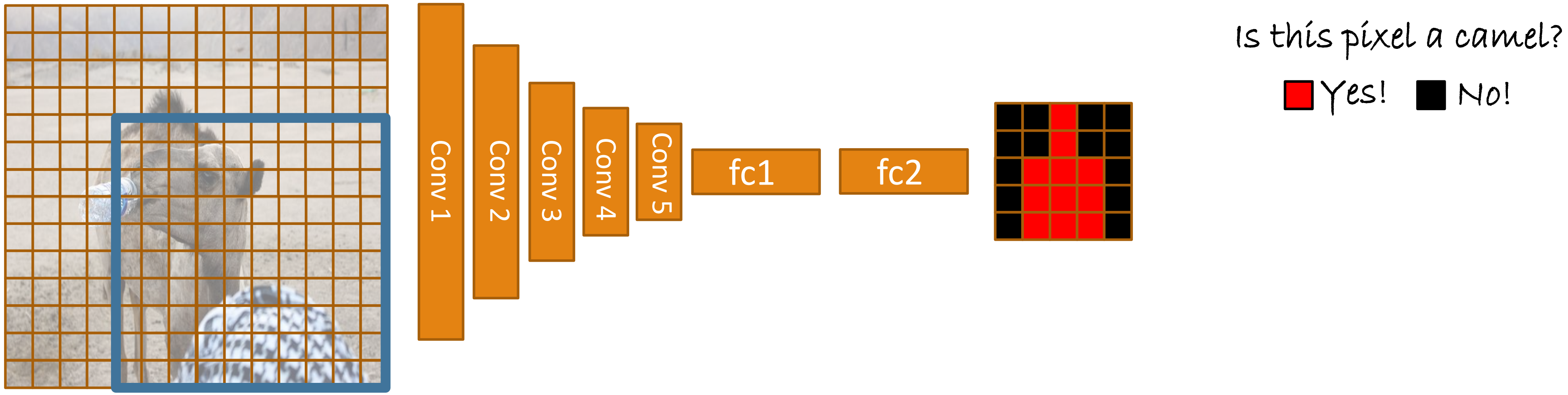
Going Fully Convolutional [LongCVPR2014]

- Image larger than network input → slide the network



Going Fully Convolutional [LongCVPR2014]

- Image larger than network input → slide the network



Fully Convolutional Networks [LongCVPR2014]

- Connect intermediate layers to output

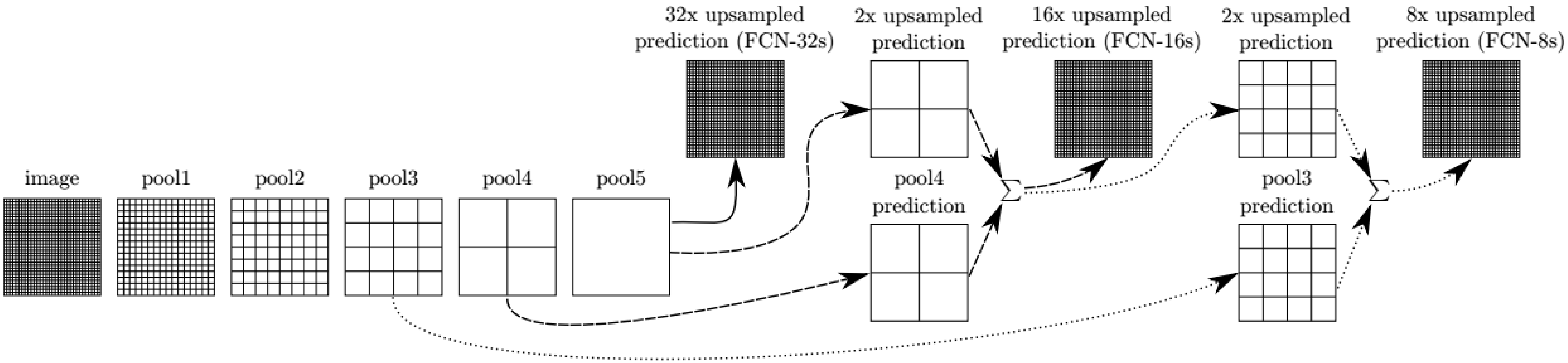
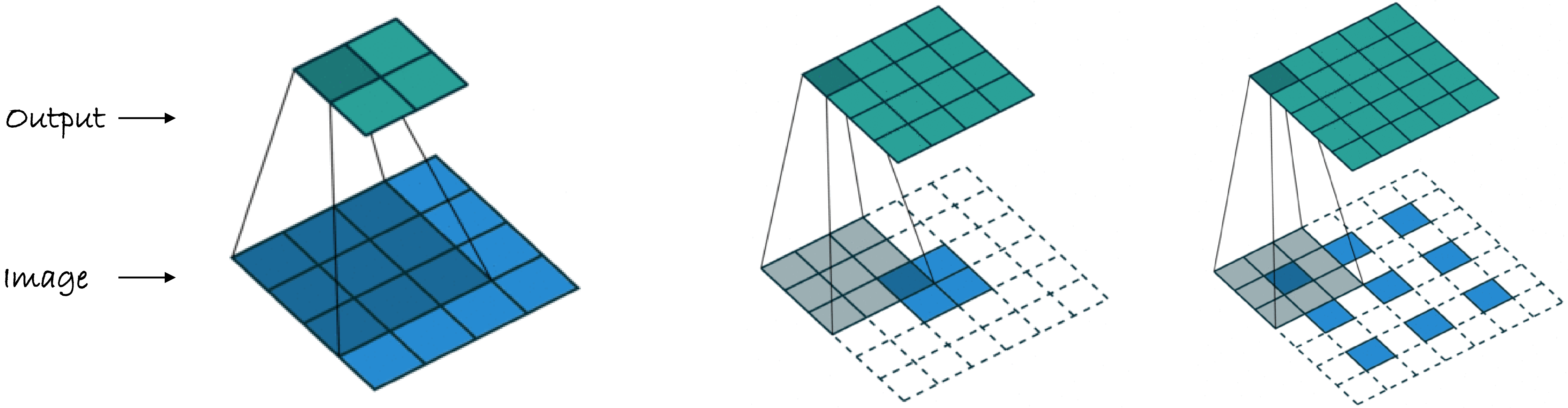


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the `pool4` layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from `pool3`, at stride 8, provide further precision.

Fully Convolutional Networks [LongCVPR2014]

- Output is too coarse
  - Image Size 500x500, Alexnet Input Size: 227x227 → Output: 10x10
- How to obtain dense predictions?
- Upconvolution
  - Other names: deconvolution, transposed convolution, fractionally-strided convolutions

Deconvolutional modules



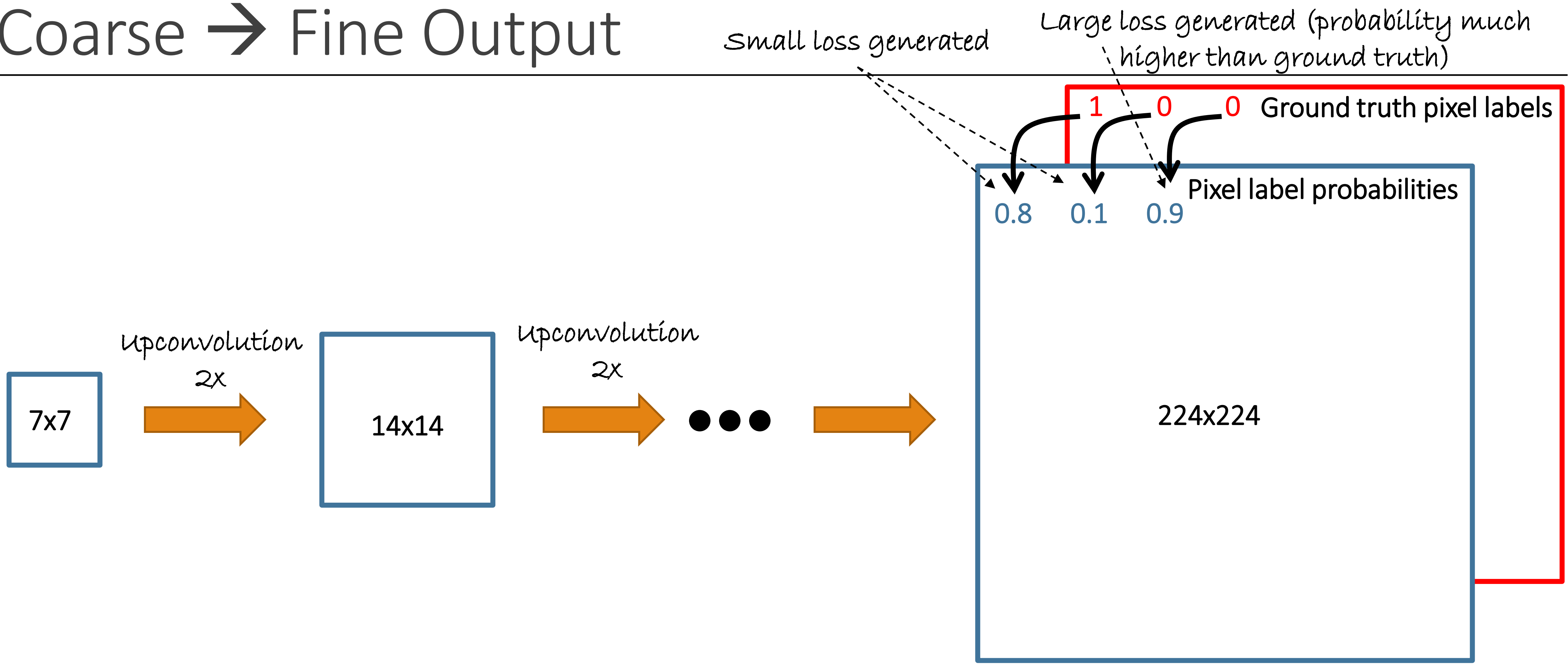
<u>Convolution</u>
No padding, no strides

<u>Upconvolution</u>
No padding, no strides

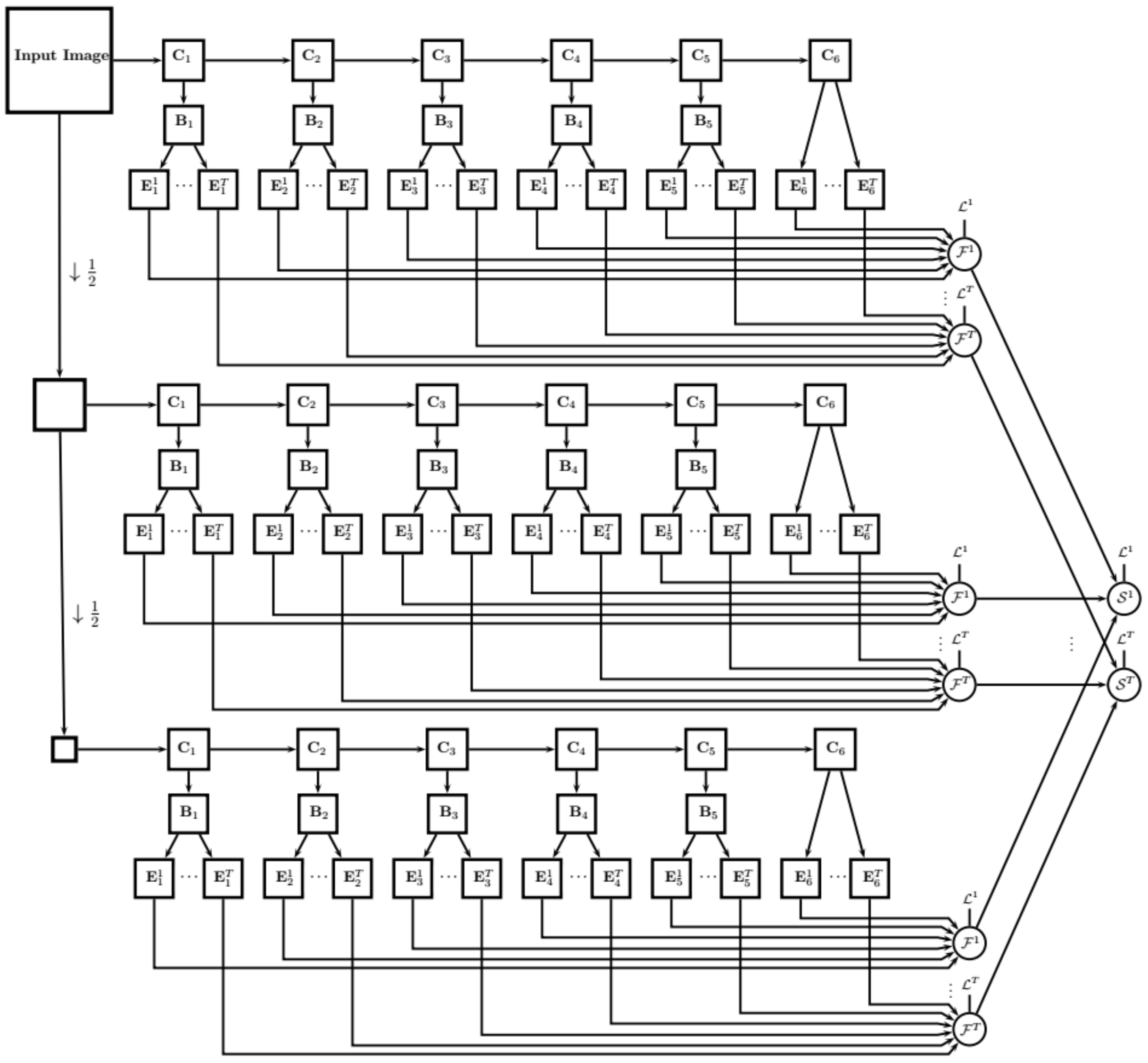
<u>Upconvolution</u>
Padding, strides

More visualizations: https://github.com/vdumoulin/conv_arithmetic

Coarse → Fine Output



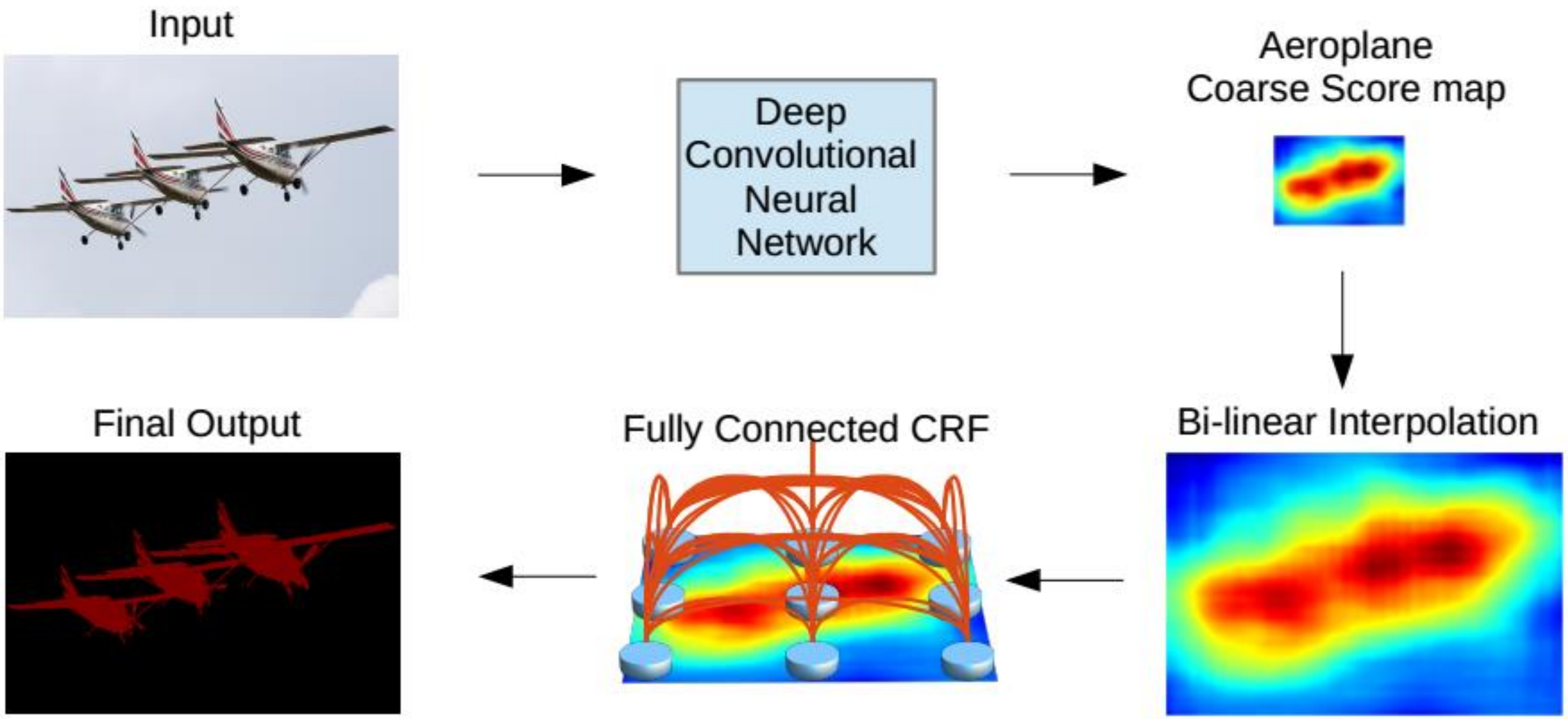
Structured losses



Deep ConvNets with CRF loss [Chen, Papandreou 2016]

- Segmentation map is good but not pixel-precise
  - Details around boundaries are lost
- Cast fully convolutional outputs as unary potentials
- Consider pairwise potentials between output dimensions

Deep ConvNets with CRF loss [Chen, Papandreou 2016]



Deep ConvNets with CRF loss [Chen, Papandreou 2016]

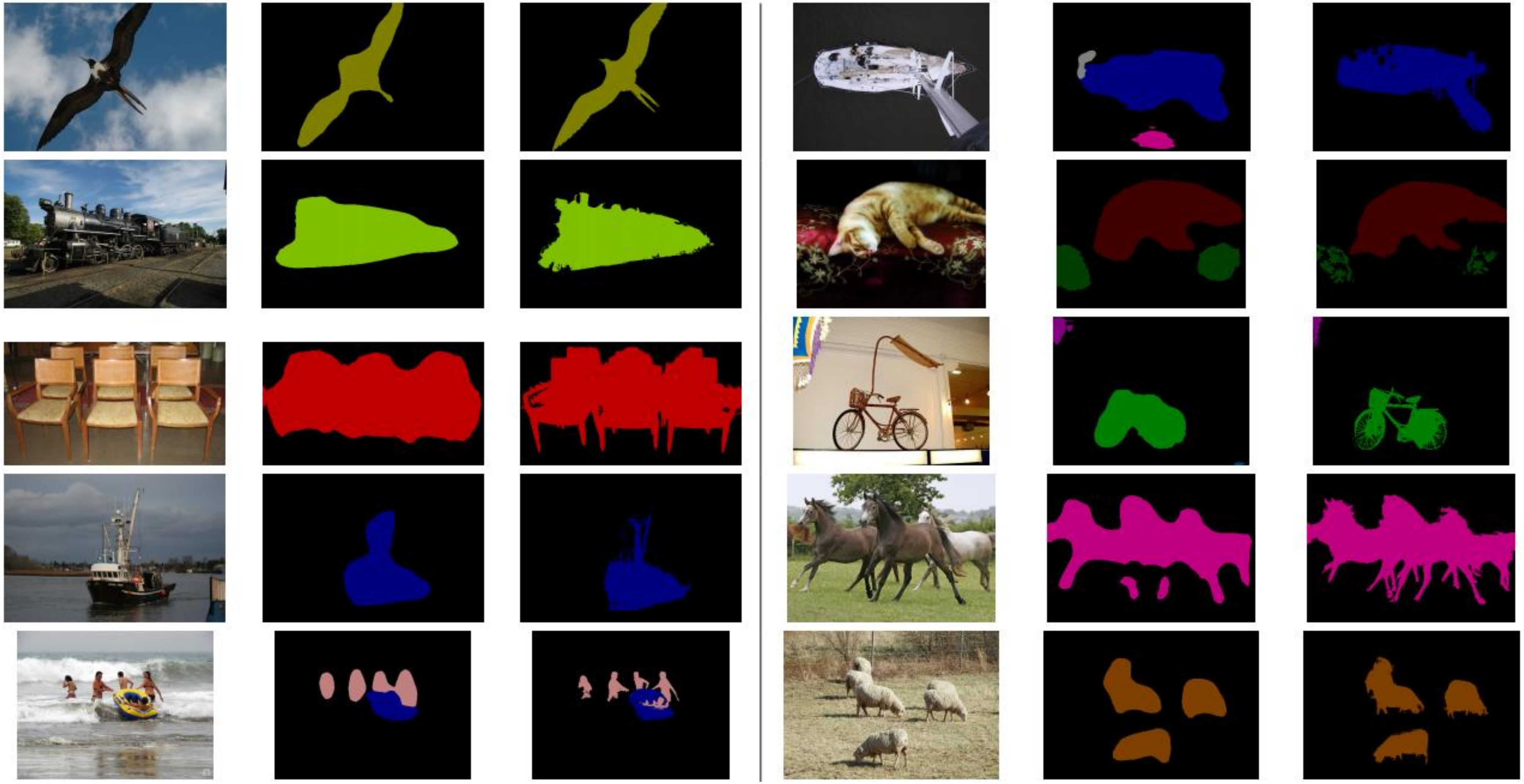
- Segmentation map is good but not pixel-precise
  - Details around boundaries are lost
- Cast fully convolutional outputs as unary potentials
- Consider pairwise potentials between output dimensions
- Include Fully Connected CRF loss to refine segmentation

$$E(x) = \sum \theta_i(x_i) + \sum \theta_{ij}(x_i, x_j)$$

Total loss — Unary loss — Pairwise loss

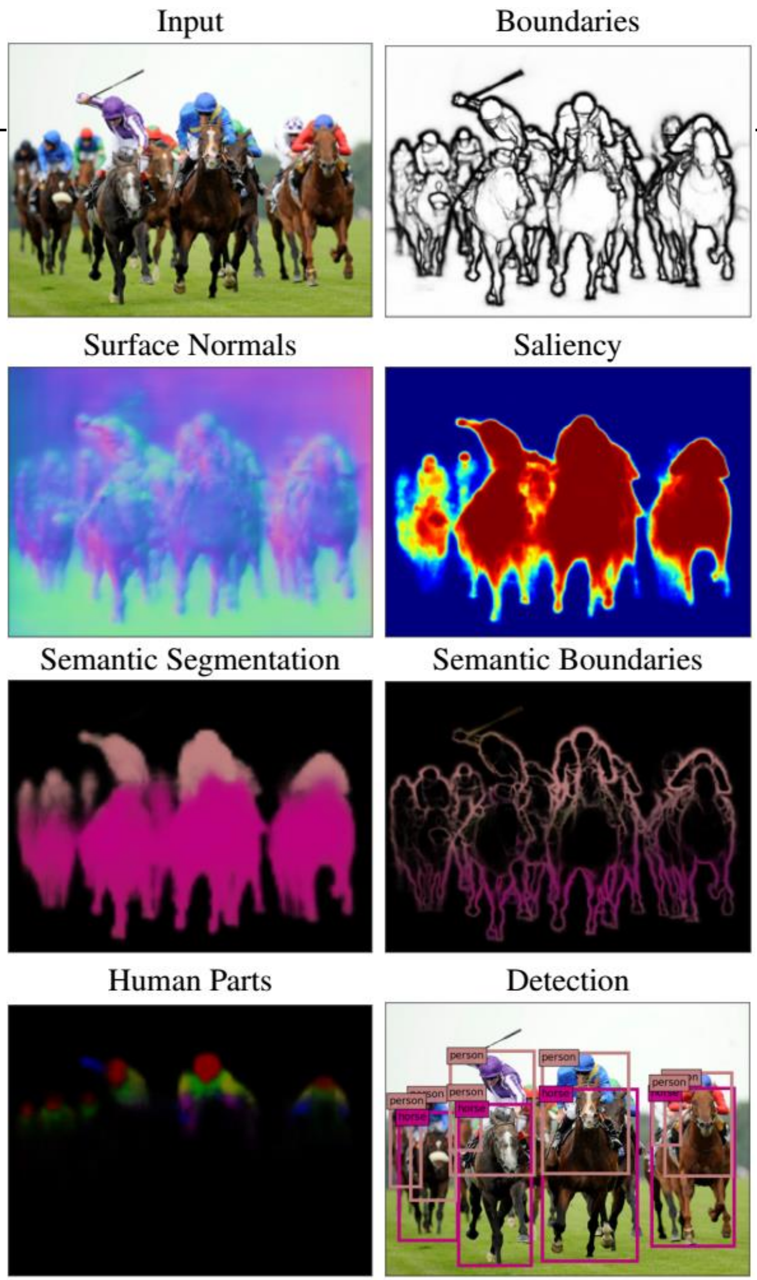
$$\theta_{ij}(x_i, x_j) \sim w_1 \exp\left(-\alpha |p_i - p_j|^2 - \beta |I_i - I_j|^2\right) + w_2 \exp(-\gamma |p_i - p_j|^2)$$

Deep ConvNets with CRF loss: Examples



One image → Several tasks

- Per image we can predict, boundaries, segmentation, detection, …
  - Why separately?
- Solve multiple tasks simultaneously
- One task might help learn another better
- One task might have more annotations
- In real applications we don't want 7 VGGnets
  - 1 for boundaries, 1 for normals, 1 for saliency, …



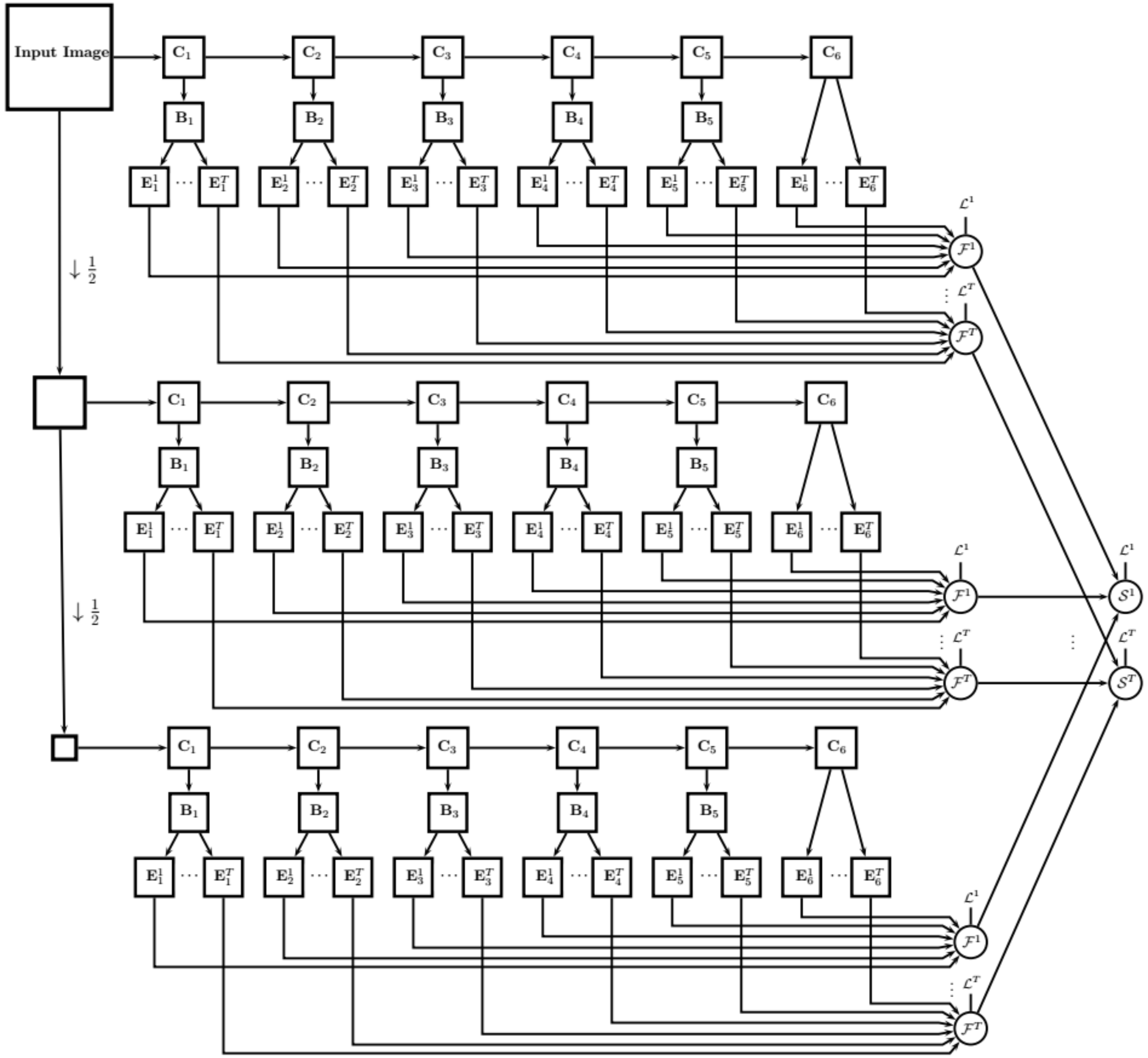
Multi-task learning

- The total loss is the summation of the per task losses
- The per task loss relies on the common weights (VGGnet) and the weights specialized for the task

$$\mathcal{L}_{total} = \sum_{task} \mathcal{L}_{task}(\theta_{common}, \theta_{task}) + \mathcal{R}(\theta_{task})$$

- One training image might contain specific only annotations
  - Only a particular task is “run” for that image
- Gradients per image are computed for tasks available for the image only

Ubernet [Kokkinos2016]



Ubernet: Backpropagation

Naïve backpropagation

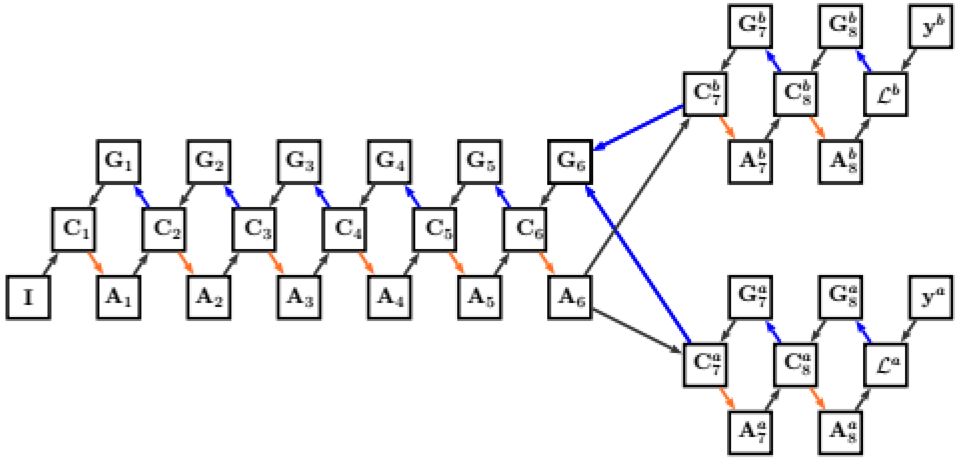
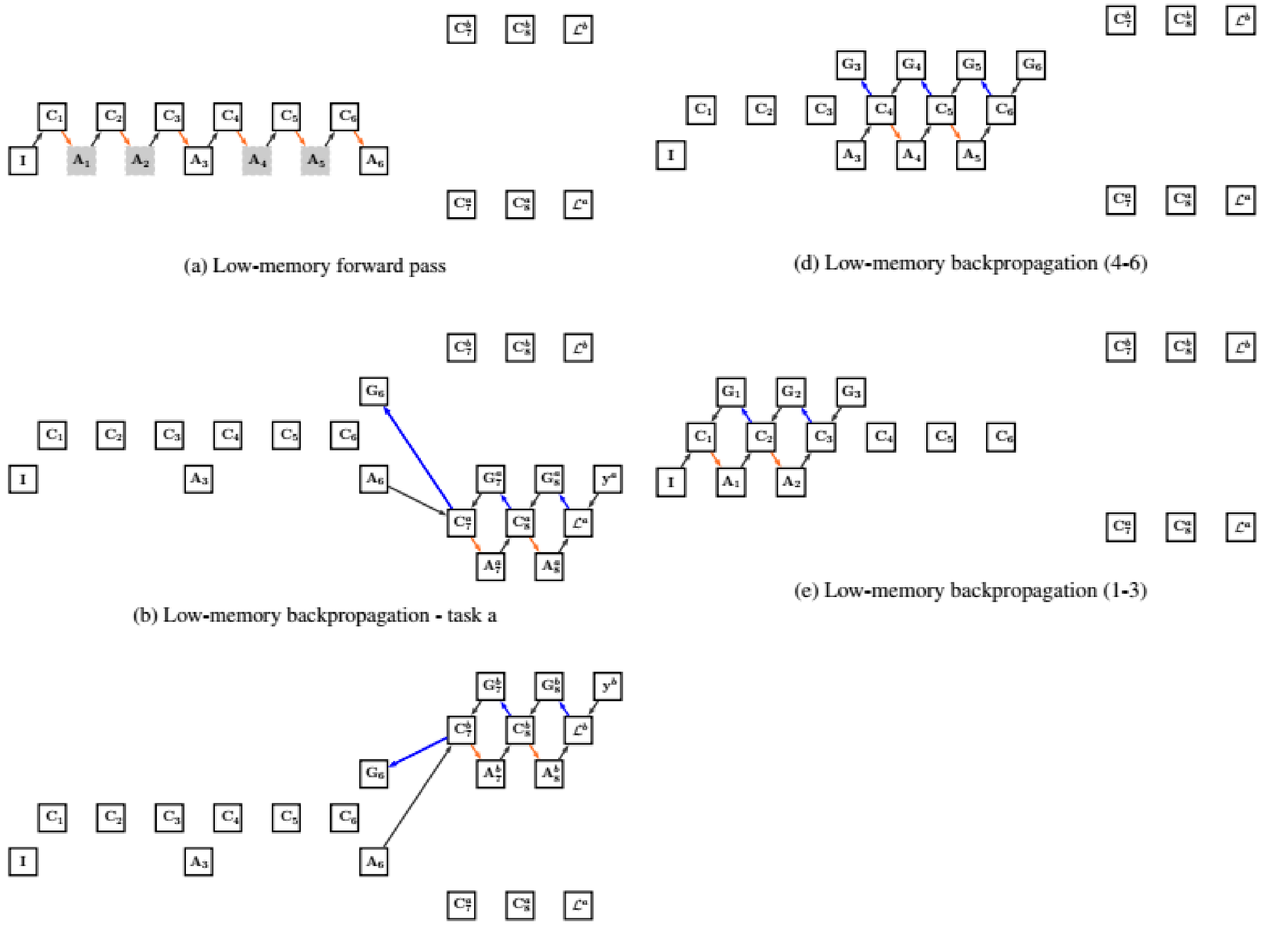


Figure 5: Vanilla backpropagation for multi-task training: a naive implementation has a memory complexity $2N(L_C + TL_T)$, where here $L_C = 6$ is the depth of the common CNN trunk, $L_T = 3$ is the depth of the task-specific branches and $T = 2$ is the number of tasks.

Ubernet backpropagation



(a) Low-memory forward pass

(d) Low-memory backpropagation (4-6)

(b) Low-memory backpropagation - task a

(e) Low-memory backpropagation (1-3)

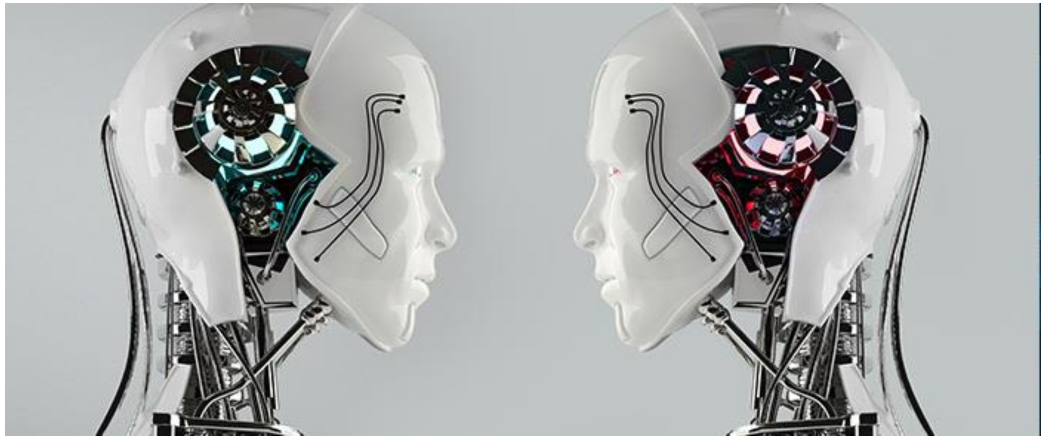
(c) Low-memory backpropagation - task b

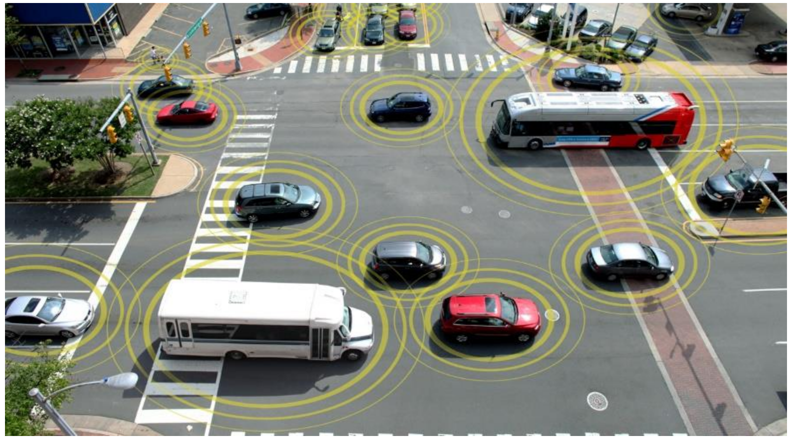
Question

- So far, what have you noticed?

Question

- So far, what have you noticed?
- Most works are done in the last 2-3 years
  - Very fast, very active, very volatile area of research that attracts lots of interest





Summary

- What is structured prediction?
- Can we repurpose for structured prediction?
- Structured losses on ConvNets
- Multi-task learning with ConvNets

Reading material & references

- http://www.deeplearningbook.org/
  - Part III: Chapter 16

[Kokkinos2016] Kokkinos, *UberNet: Training a `Universal' Convolutional Neural Network for Low-, Mid-, and High-Level Vision using Diverse Datasets and Limited Memory*, arXiv, 2016
[Rematas2016] Rematas, Ritschel, Fritz, Gavves, Tuytelaars. *Deep Reflectance Maps*, CVPR, 2016
[Ren2016] Ren, He, Girshick, Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, NIPS, 2015
[Girshick2015] Girshick. *Fast R-CNN*, ICCV, 2015
[Wang2015] Wang, Fouhey, Gupta. *Designing Deep Networks for Surface Normal Estimation*, arXiv, 2015
[Chen2014] Chen, Papandreou, Kokkinos, Murphy, Yuille. *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs*, arXiv, 2014
[He2014] He, Zhang, Ren, Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, ECCV, 2014

Next lecture

- Deep Learning and Natural Language
- Invited lecture given by Prof. Christof Monz