Instance Segmentation

Jun Gao

Outline

- Motivation
- Basic Pipeline
- Recurrent paradigm
- Energy based methods
- Instance-aware FCN

Motivation

- Both object detection and semantic segmentation
- Assign category label and instance label to each pixel

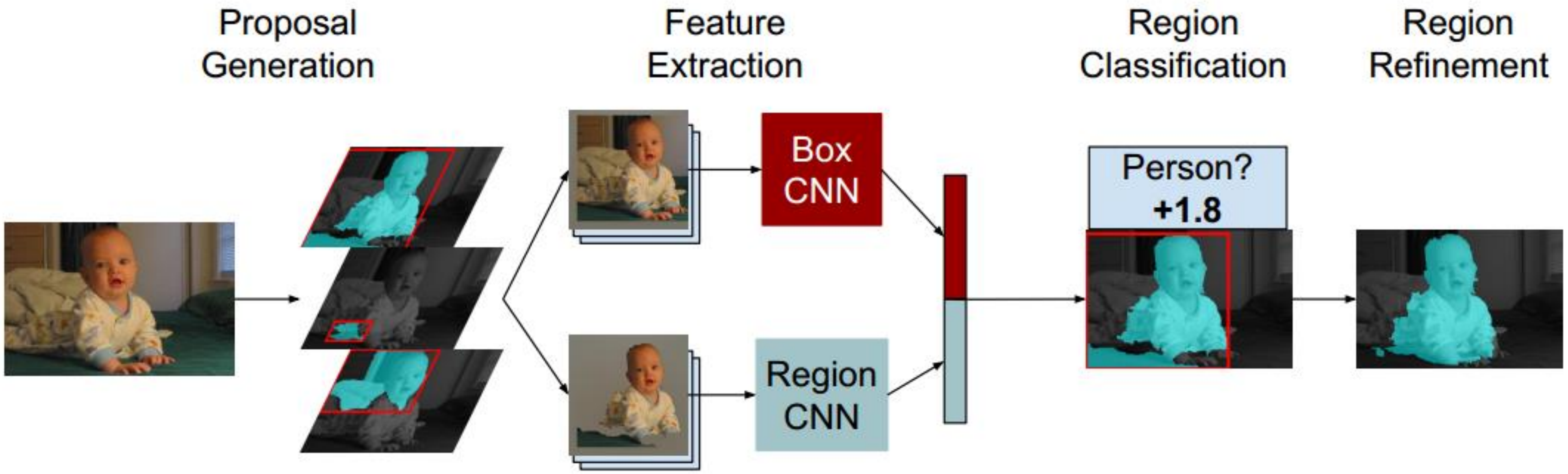
(a) Object Detection (b) Semantic Segmentation (c) Instance Segmentation

Simultaneous Detection and Segmentation

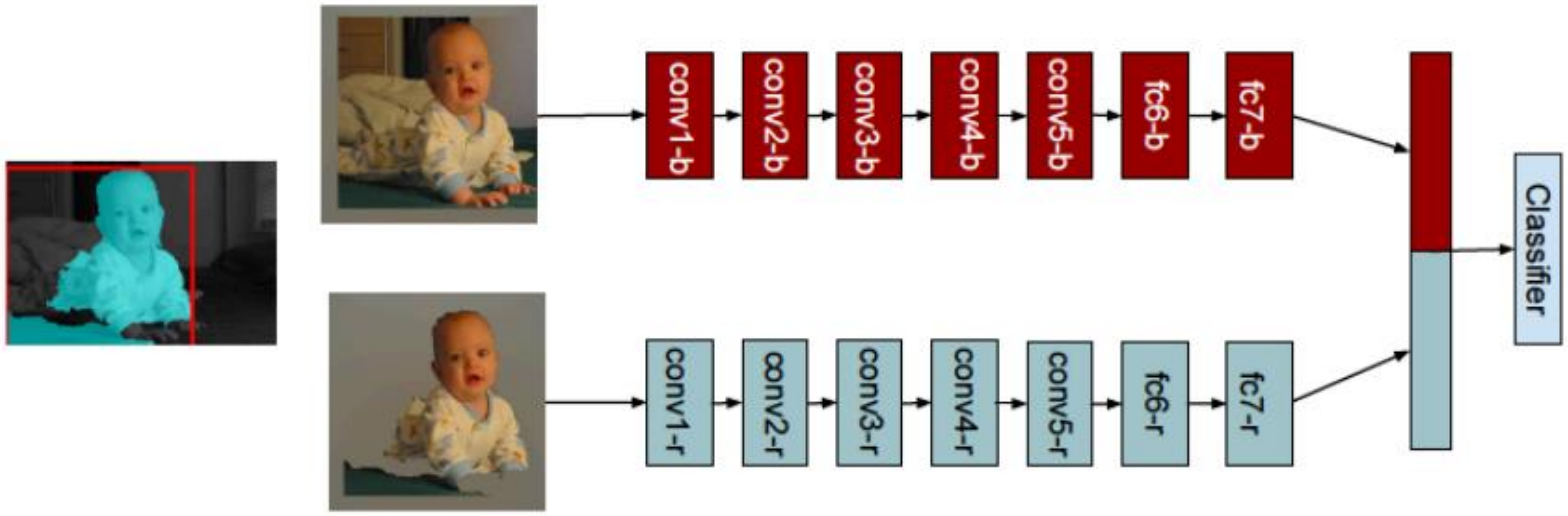
Bharath Hariharan, Pablo Arbel aez, Ross Girshick, Jitendra Malik
UC Berkely
In ECCV 2014, cited by 286

Basic Pipeline

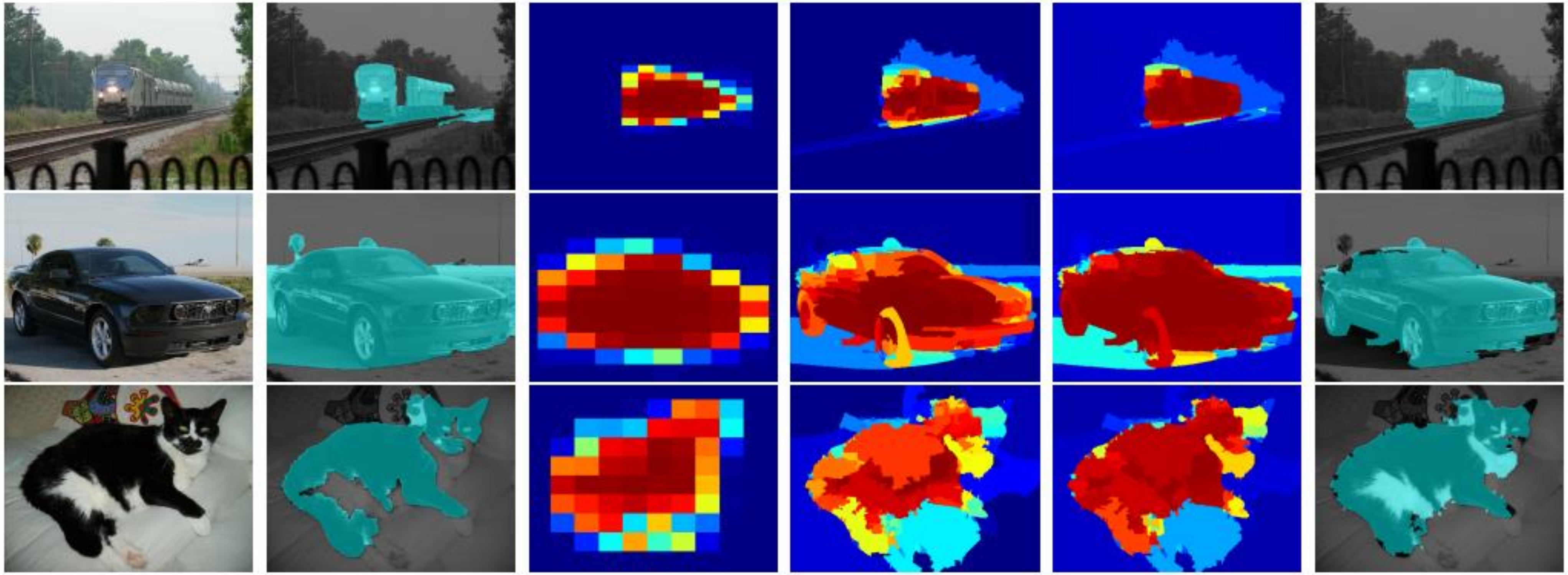
- **High level idea:** Detection → Segmentation
- **Pipeline**:



Feature extraction



Region Refinement



Contributions

- Dual way for feature extraction
- Region refinement

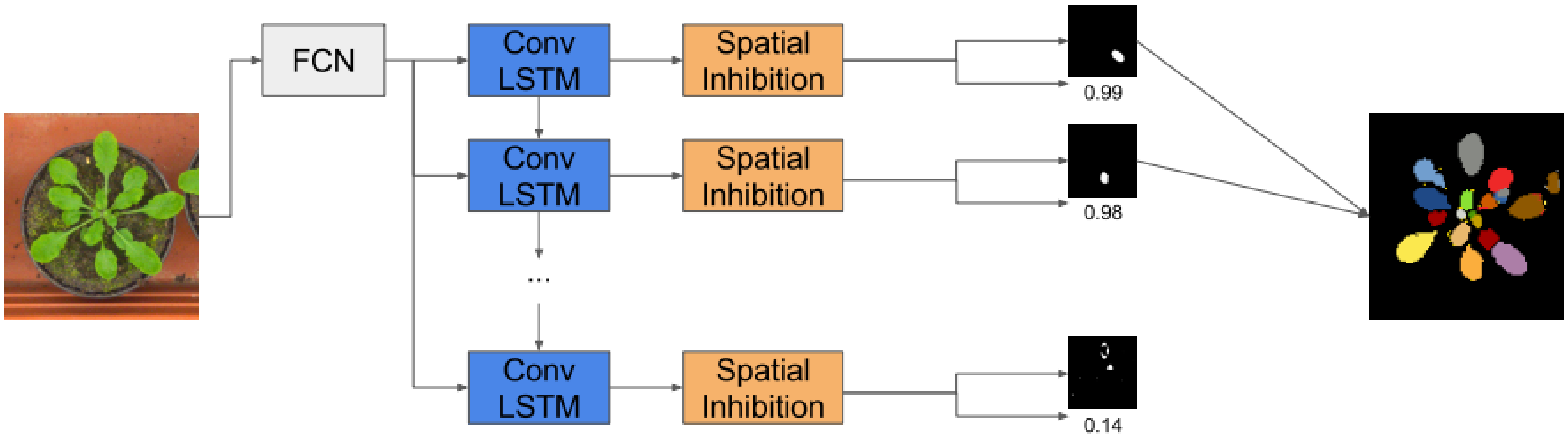
Recurrent paradigm

- Recurrent over one instance
  - Recurrent Instance Segmentation
- Recurrent over segmentation
  - Iterative Instance Segmentation
- Recurrent over detection
  - End-to-End Instance Segmentation with Recurrent Attention

Recurrent Instance Segmentation

Bernardino Romera-Paredes and Philip H.S Torr
University of Oxford
In ECCV 2016, cited by 35

Pipeline

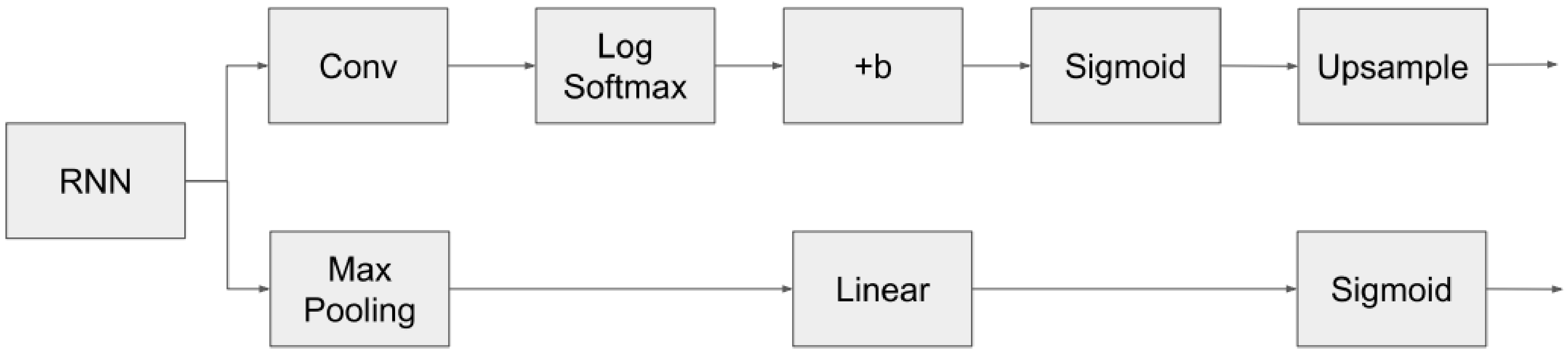


FCN
Conv LSTM
Conv LSTM
...
Conv LSTM
Spatial Inhibition
Spatial Inhibition
Spatial Inhibition
0.99
0.98
0.14

Attention by Spatial Inhibition

- Conv-LSTM
  - Both input and output is a map
  - Replace linear layer with convolutional layer in LSTM
- Attention (serve as segmentor)

$$\mathbb{R}^{h' \times w' \times d} \rightarrow \{[0,1]^{h \times w}, [0,1]\}$$



Loss Function

$$\ell(\hat{\mathbf{Y}}, \mathbf{s}, \mathbf{Y}) = \min_{\delta \in \mathcal{S}} - \sum_{\hat{t}=1}^{\tilde{n}} \sum_{t=1}^{n} f_{\mathrm{IoU}} \left(\hat{\mathbf{Y}}_{\hat{\mathbf{t}}}, \mathbf{Y}_{\mathbf{t}} \right) \delta_{\hat{t},t} + \lambda \sum_{t=1}^{\hat{n}} f_{\mathrm{BCE}} \left([t \leq n], s_t \right),$$

- Hungarian Algorithm
- Predict n+2 while training, and terminate if score less than 0.5

Summary

- Prons:
  - Recurrently generation, suitable for variable number of instances
- Cons
  - Segmentation result maybe too coarse.
  - Only considered binary class (foreground and background)

Iterative Instance Segmentation

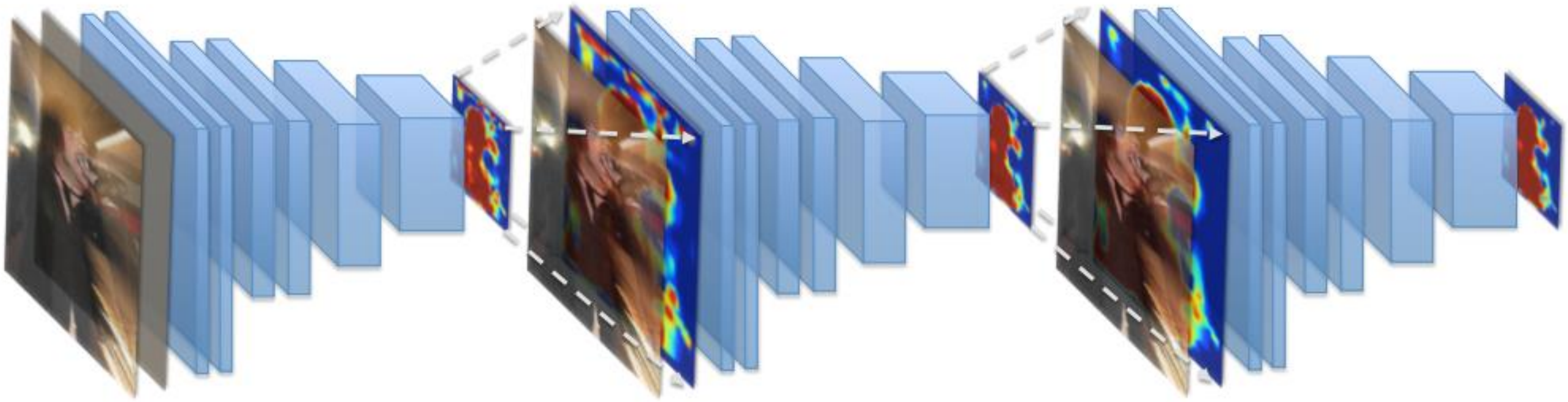
Ke Li, Bharath Hariharan, Jitendra Malik
UC Berkeley
In CVPR 2016, cited by 24

Intuition

- Incorporating underlying structure of labellings into segmentation.
- Learning object structure automatically.
- Iterative error feedback.
  - Improve previous outputs
  - Correct previous mistakes

Pipeline

- Fast R-CNN to get bounding boxes.
- Hyper-colmumn Net to get segmentation results.



Pseudocode

Algorithm 1 Training Procedure

Require: D is a training set consisting of (x, y) pairs, where x and y denote the instance and the ground truth labelling respectively, and f is the model

function TRAIN(D, f)
  // $p_x^{(t)}$ *is the predicted labelling of* x *in the* t^{th} *stage*
  $p_x^{(0)} \leftarrow \begin{pmatrix} 1/2 & \cdots & 1/2 \end{pmatrix}^T \ \forall (x, y) \in D$
  for $t = 1$ **to** N **do**
    // *Training set for the current stage*
    $T \leftarrow \left\{ \left(\begin{pmatrix} x \\ p_x^{(i)} \end{pmatrix}, y \right) \middle| (x, y) \in D, i < t \right\}$
    Train model f on T starting from the current parameters of f
    $p_x^{(t)} \leftarrow f \begin{pmatrix} x \\ p_x^{(t-1)} \end{pmatrix} \ \forall (x, y) \in D$
  end for
  return f
end function

Algorithm 2 Testing Procedure

Require: f is the model and x is an instance

function TEST(f, x)
  // $\hat{y}^{(t)}$ *is the predicted labelling of* x *after* t *iterations*
  $\hat{y}^{(0)} \leftarrow \begin{pmatrix} 1/2 & \cdots & 1/2 \end{pmatrix}^T$
  for $t = 1$ **to** M **do**
    $\hat{y}^{(t)} \leftarrow f \begin{pmatrix} x \\ \hat{y}^{(t-1)} \end{pmatrix}$
  end for
  return $\hat{y}^{(M)}$
end function

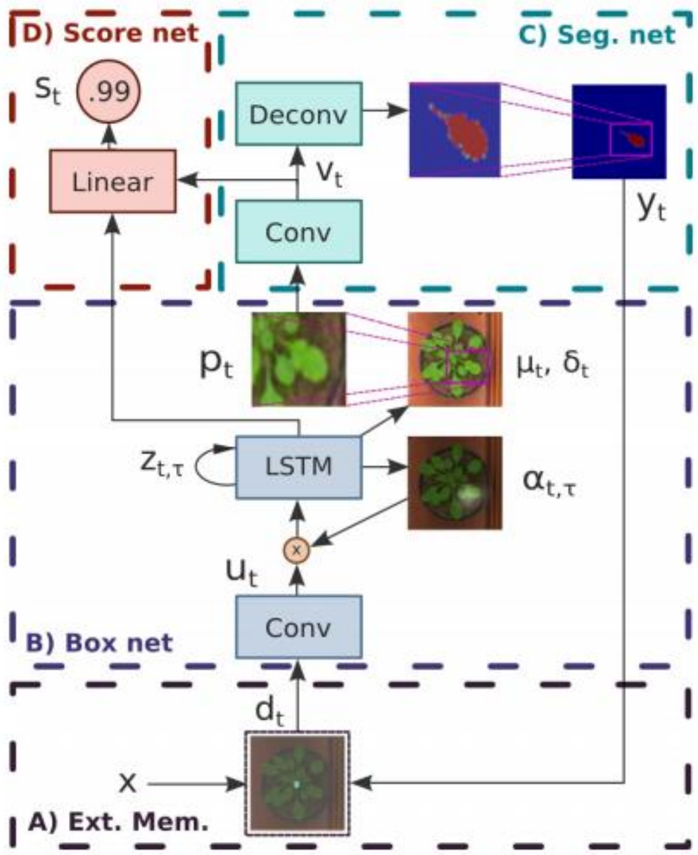
Summary

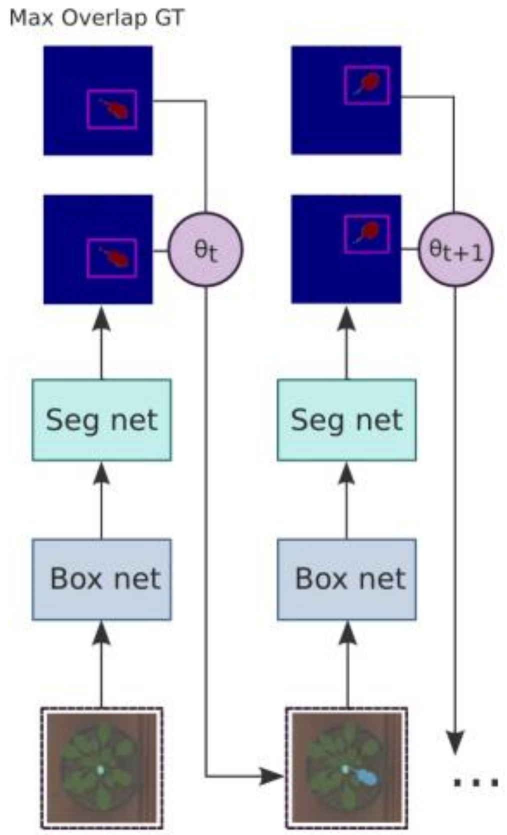
- Pros
  - Recurrently refine the segmentation
  - Learn the object structure constrains
- Cons
  - Only for instance segmentation, not semantic
  - Maybe, different stage needs different networks?

End-to-End Instance Segmentation with Recurrent Attention

Mengye Ren, Richard S. Zemel
University of Toronto
In CVPR 2017, cited by 16

Pipeline





A) External Memory: Image representation and current canvas
B) Box Net: Recurrently generate bounding boxes
C) Seg. Net: Get segmentation
D) Score Net: Get the probability.

θ_t is the probability for scheduled sampling.
Terminate if score is less than a threshold.

External Memory

- Preprocess Images:
  - Pixel-level foreground segmentation
  - Angles relative to centroid of objects (discretize into 8 categories)
- Memory:
  - What pixels we have already segmented out.

$$\mathbf{c}_t = \begin{cases} \mathbf{0}, & \text{if } t = 0 \\ \max(\mathbf{c}_{t-1}, \mathbf{y}_{t-1}), & \text{otherwise} \end{cases} \tag{1}$$

$$\mathbf{d}_t = [\mathbf{c}_t, \mathbf{x}] \tag{2}$$

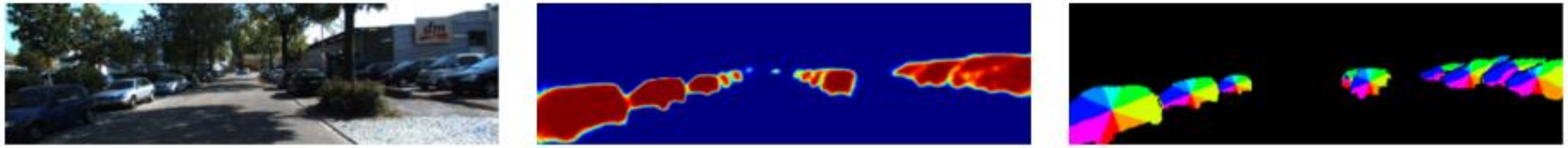
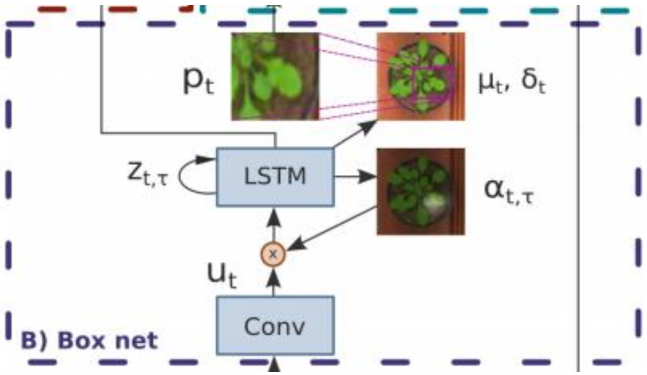


Figure 3: Illustration of the output of the pretrained FCN. Left: input image. Middle: predicted foreground. Right: predicted angle map.

Box Net



- z_t is used to predict the position of bounding box

$$\mathbf{u}_t = \mathrm{CNN}(\mathbf{d}_t) \tag{3}$$

$$\mathbf{z}_{t,\tau} = \begin{cases} \mathbf{0}, & \text{if } \tau = 0 \\ \mathrm{LSTM}(\mathbf{z}_{t,\tau-1}, \sum_{h,w} \alpha_{t,\tau-1}^{h,w} u_t^{h,w,l}) & \text{otherwise} \end{cases} \tag{4}$$

$$\alpha_{t,\tau}^{h,w} = \begin{cases} 1/(H' \times W'), & \text{if } \tau = 0 \\ \mathrm{MLP}(\mathbf{z}_{t,\tau}), & \text{otherwise} \end{cases} \tag{5}$$

$$[\tilde{g}_{X,Y}, \log \tilde{\delta}_{X,Y}, \log \sigma_{X,Y}, \gamma] = \mathbf{w}_b^\top \mathbf{z}_{t,\mathrm{end}} + w_{b0} \tag{6}$$

$$g_X = (\tilde{g}_X + 1)W/2 \tag{7}$$

$$g_Y = (\tilde{g}_Y + 1)H/2 \tag{8}$$

$$\delta_X = \tilde{\delta}_X W \tag{9}$$

$$\delta_Y = \tilde{\delta}_Y H \tag{10}$$

- α is the soft attention over pixels.
- Extract Image patch using Gaussian Interpolation.

Others

- Seg. Net:
  - Deconv Network

$$\mathbf{v}_t = \text{CNN}(\mathbf{p}_t) \quad (17)$$

$$\tilde{\mathbf{y}}_t = \text{D-CNN}(\mathbf{v}_t) \quad (18)$$

$$\mathbf{y}_t = \text{sigmoid}\left(\gamma \cdot \text{Extract}(\tilde{\mathbf{y}}_t, F_Y^\top, F_X^\top) - \beta\right) \quad (19)$$

- Score Net:

$$s_t = \text{sigmoid}(\mathbf{w}_{zs}^\top \mathbf{z}_{t,\text{end}} + \mathbf{w}_{vs}^\top \mathbf{v}_t + w_{s0}) \quad (20)$$

- Loss Function:

$$\mathcal{L}(\mathbf{y}, \mathbf{b}, \mathbf{s}) = \mathcal{L}_y(\mathbf{y}, \mathbf{y}^*) + \lambda_b \mathcal{L}_b(\mathbf{b}, \mathbf{b}^*) + \lambda_s \mathcal{L}_s(\mathbf{s}, \mathbf{s}^*) \quad (21)$$

Summary

- Two recurrent pathways
  - Recurrent inside each instance (multi glimpse)
  - Recurrent outside each instance
- End-to-end optimization
- One instance at a time

Summary - Recurrent paradigm

- Recurrent over detection
  - End-to-End Instance Segmentation with Recurrent Attention
- Recurrent over segmentation
  - Iterative Instance Segmentation
- Recurrent over one instance
  - Recurrent Instance Segmentation

Energy Based

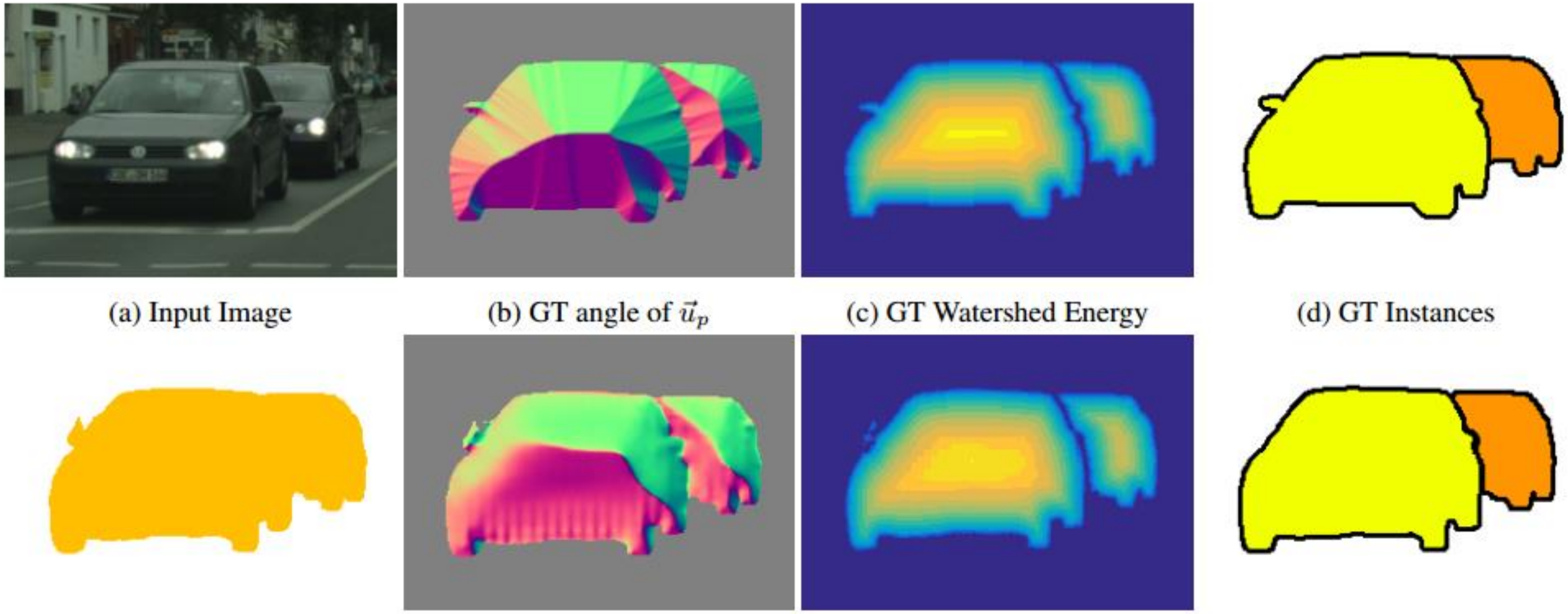
- Deep Watershed Transform for Instance Segmentation
- Boundary-aware Instance Segmentation
- Pixel wise Instance Segmentation with a Dynamically Instantiated Network

Deep Watershed Transform for Instance Segmentation

Min Bai Raquel Urtasun
University of Toronto
In CVPR 2017, cited by 6

Intuition

- Another view of instance segmentation
  - Energy map: the boundary and background in the basin

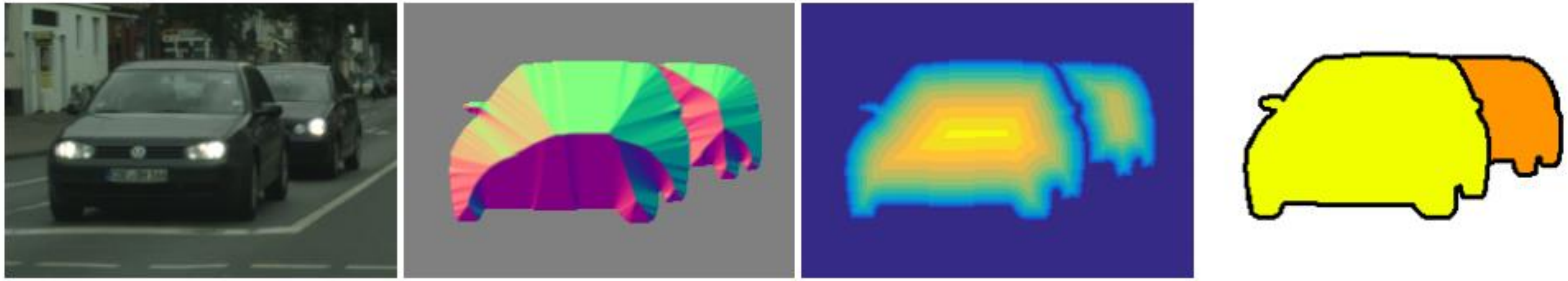


(a) Input Image (b) GT angle of $\vec{u}_p$ (c) GT Watershed Energy (d) GT Instances

(e) Sem. Segmentation of [34] (f) Pred. angle of $\vec{u}_p$ (g) Pred. Watershed Transform (h) Pred. Instances

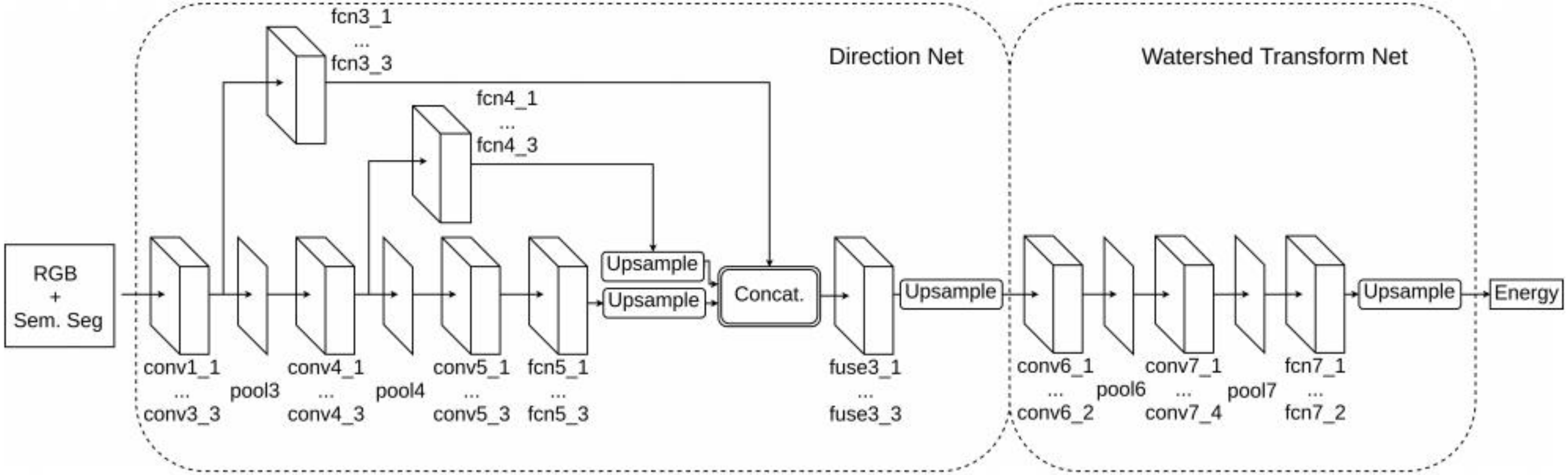
Two parts

- Energy map:
  - Distance from the pixel to the nearest boundary point
  - K discrete energy bins
- Direction of the descent of energy.
  - the direction from the pixel to the nearest boundary point
  - Unit vector



(a) Input Image (b) GT angle of $\vec{u}_p$ (c) GT Watershed Energy (d) GT Instances

Networks



$$l_{\text{direction}} = \sum_{p \in \mathcal{P}_{\text{obj}}} w_p \| \cos^{-1} < \vec{u}_{p,\text{GT}}, \vec{u}_{p,\text{pred}} > \|^2$$

$$l_{\text{watershed}} = \sum_{p \in \mathcal{P}_{\text{obj}}} \sum_{k=1}^{K} w_p c_k (\bar{t}_{p,k} \log \bar{y}_{p,k} + t_{p,k} \log y_{p,k})$$

Comments

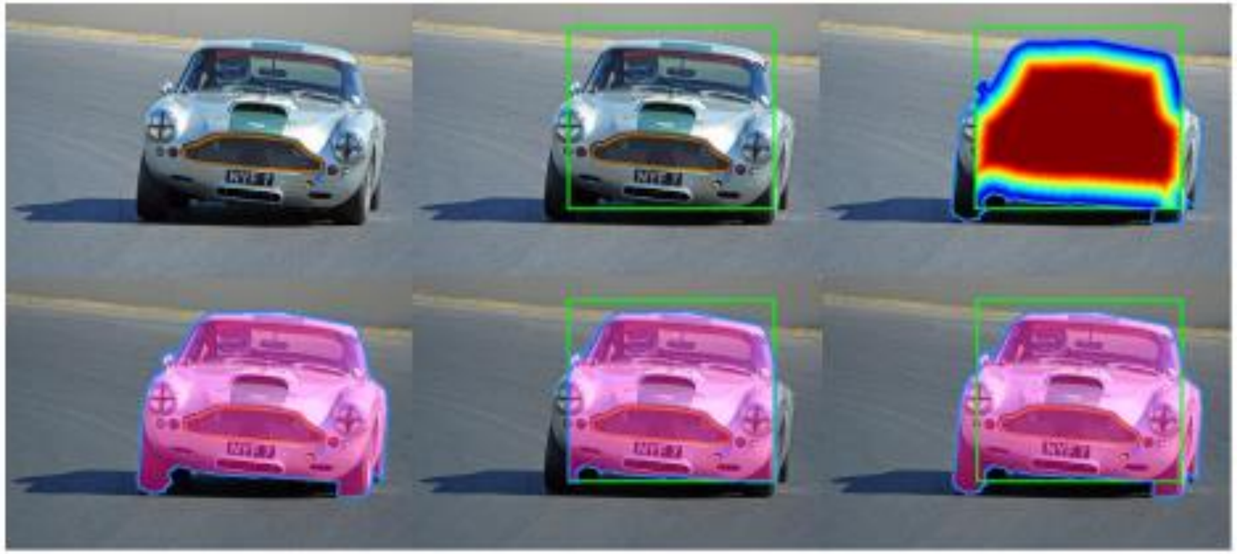
- Energy view of instance segmentation
- Divide into two subtasks
- For spine curve:
  - Treat the interpolation value as energy?
  - Discuss later.

Boundary-aware Instance Segmentation

Zeeshan Hayder, Xuming He, Mathieu Salzmann
Australian National University & 2Data61/CSIRO 3CVLab, EPFL, Switzerland
In CVPR 2017

Intuition

- Recover error from object detection part
- Representation of segmentation results must have the ability to extend to the outside of bounding boxes
- Boundary ware mask representation.



Methods

- Boundary aware mask representation
  - Distance from pixel to the boundary and background

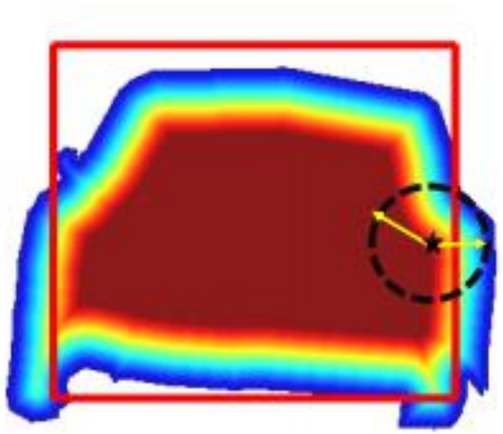
$$D(p) = \min\left(\min_{\forall q \in Q} \lceil d(p,q) \rceil, R\right)$$

  - Binary view-point (for classification)

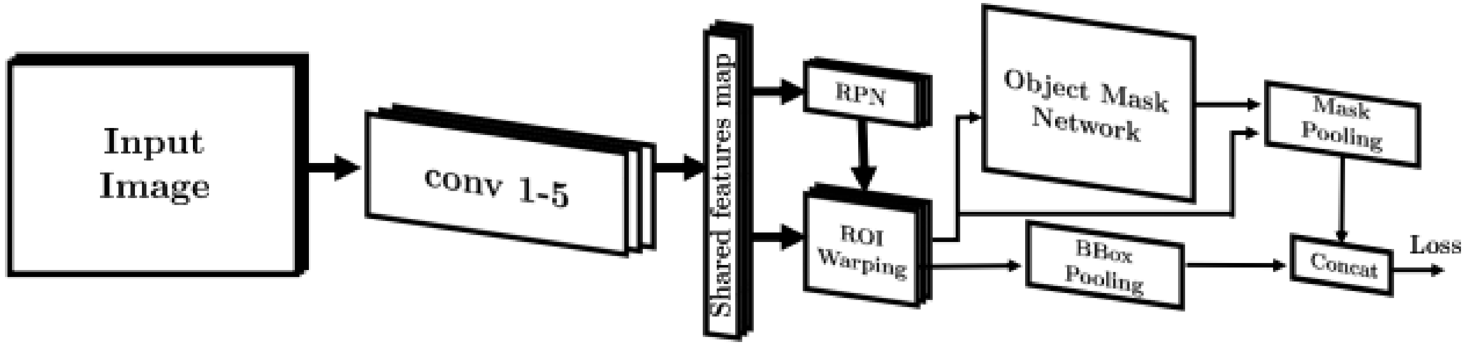
$$D(p) = \sum_{n=1}^{K} r_n \cdot b_n(p), \quad \sum_{n=1}^{K} b_n(p) = 1$$

  - Mask could be represent as

$$M = \bigcup_{p} T(p, D(p)) = \bigcup_{p} T(p, \sum_{n=1}^{K} r_n \cdot b_n(p))$$
$$= \bigcup_{n=1}^{K} \bigcup_{p} T(p, r_n \cdot b_n(p)) = \bigcup_{n=1}^{K} T(\cdot, r_n) * B_n$$



Networks



- Loss Function:
  - segmentation loss (only binary cross entropy)
  - bounding box loss
  - classification loss

Discussion

- Redefine the concept of segmentation map
- For spine curve:
  - Treat the interpolation value as energy?
  - Current idea:
    - Ambiguity of start&end point: only consider [0,0.5] interval.
    - Discretize into K bins. (classification might be better than regression)
    - For connection pixels: multi-label classification.
    - After get the interpolation value, then only needs post processing
  - Advantages:
    - Both local and global, might solve local minimum problem
    - Could be extended to multi-spline curves
  - Disadvantages:
    - Might too hard to learn? Needs intermediate representation?

Thank You

Supplementary

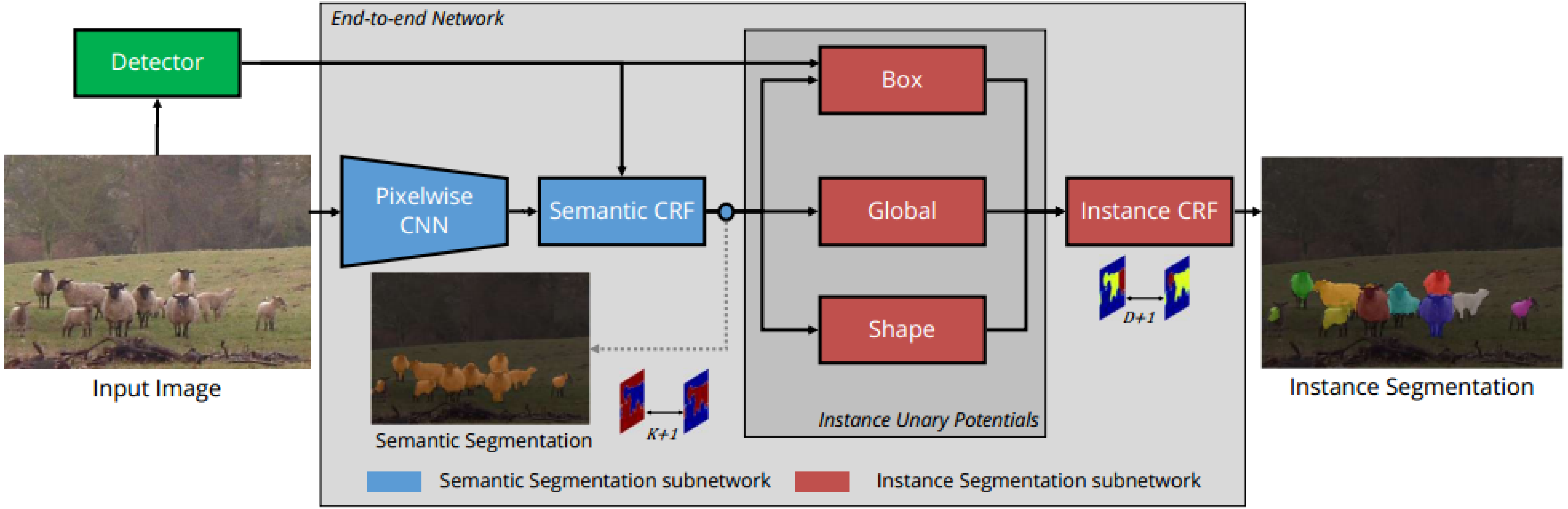
Pixelwise Instance Segmentation with a Dynamically Instantiated Network

Anurag Arnab and Philip H.S Torr
University of Oxford
In CVPR 2017, cited by 3

Intuition

- From the entire semantic segmentation map instead of bounding box
- How to assign each semantic pixels to instance identity?
- Assume each box is an instance.

Models



End-to-end Network
Detector
Box
Pixelwise CNN
Semantic CRF
Global
Instance CRF
Shape
Input Image
Semantic Segmentation
K+1
D+1
Instance Unary Potentials
Instance Segmentation
Semantic Segmentation subnetwork
Instance Segmentation subnetwork

Conditional Random Field

- Formulate the instance identity assignment as a CRF.
- During training:
  - Maximum Posterior (MAP) to get the predicted instance label. (Inference)
  - Losses are used to optimize CRF's parameters
  - The inference of CRF could be unrolled as a RNN, thus fully differentiable.
- During Testing:
  - Maximum Posterior (MAP) to get the predicted instance label.

$$E(\mathbf{V} = \mathbf{v}) = \sum_i U(v_i) + \sum_{i<j} P(v_i, v_j).$$

$$U(v_i) = -\ln[w_1 \psi_{Box}(v_i) + w_2 \psi_{Global}(v_i) + w_3 \psi_{Shape}(v_i)],$$

Energy Terms

- Box term
- Global term
- Shape term
- Pairwise term

R-FCN: Object Detection via Region-based Fully Convolutional Networks

Jifeng Dai, Yi Li, Kaiming He, Jian Sun
Microsoft Research, Tsinghua University
In Arxiv.org. 21 June 2016.

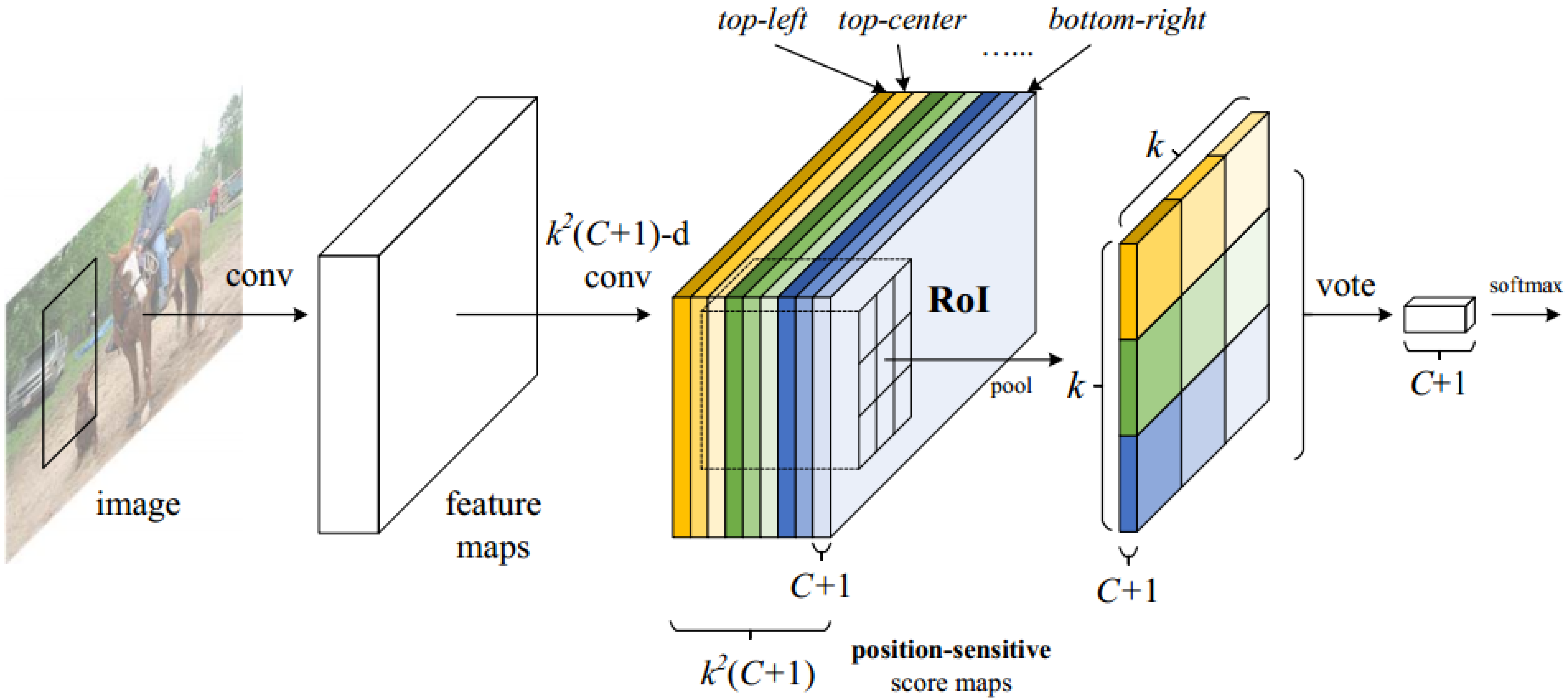
Motivation

- Translation invariance for classification V.S. translation variance for detection
- Unshared per-RoI computation (Fast R-CNN/Faster R-CNN)

Table 1: Methodologies of *region-based* detectors using **ResNet-101** [9].

	R-CNN [7]	Faster R-CNN [19, 9]	R-FCN [ours]
depth of shared convolutional subnetwork	0	91	101
depth of RoI-wise subnetwork	101	10	**0**

Key idea of R-FCN



$$r_c(i,j \mid \Theta) = \sum_{(x,y)\in \text{bin}(i,j)} z_{i,j,c}(x+x_0, y+y_0 \mid \Theta)/n.$$

Visualization of R-FCN

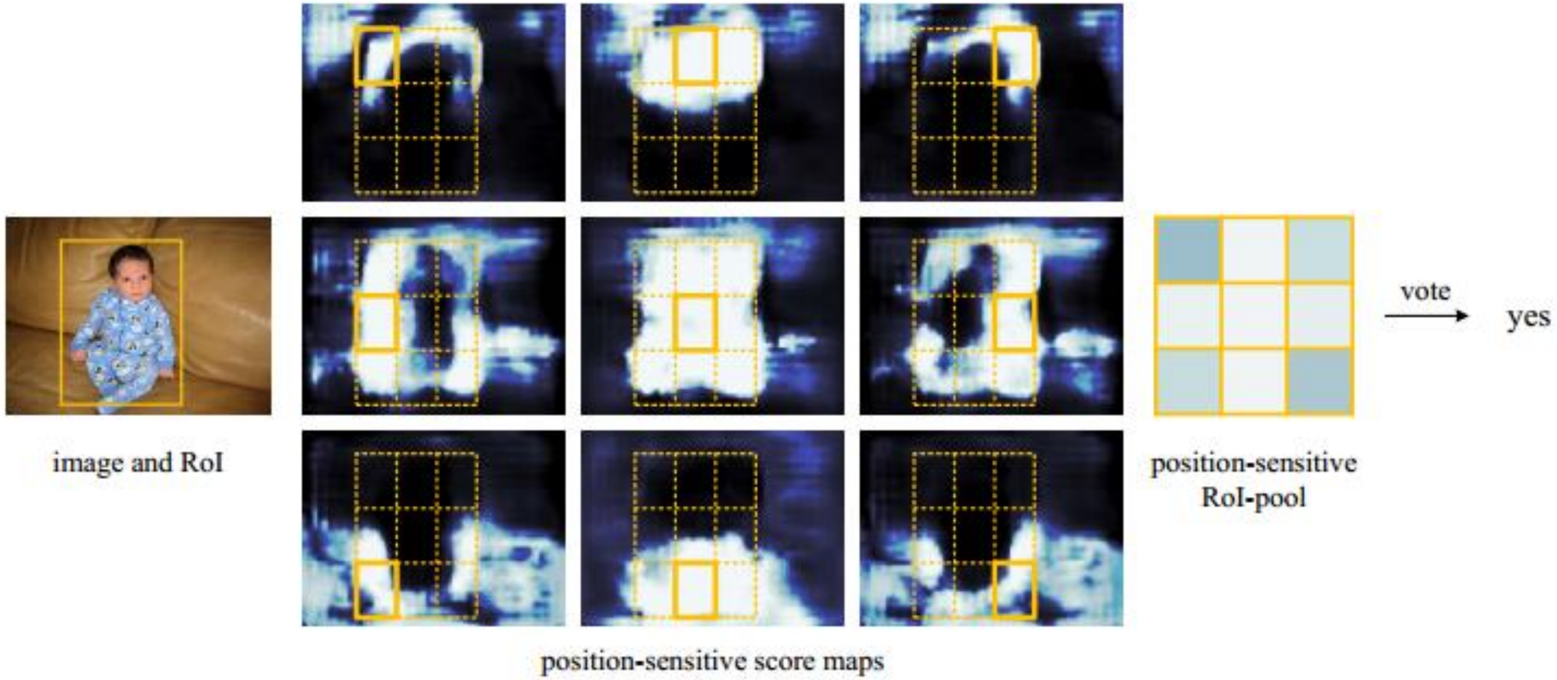


Figure 3: Visualization of R-FCN ($k \times k = 3 \times 3$) for the *person* category.

Visualization of R-FCN

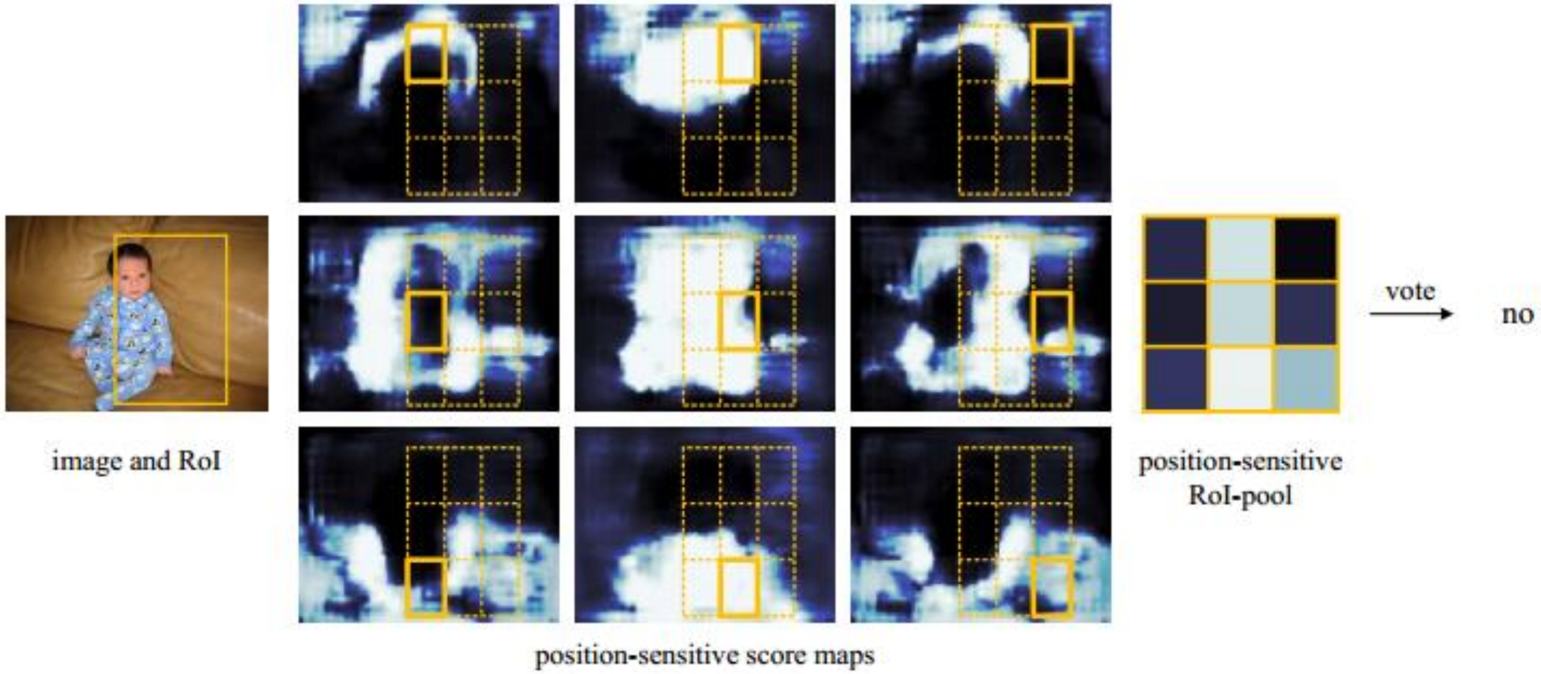
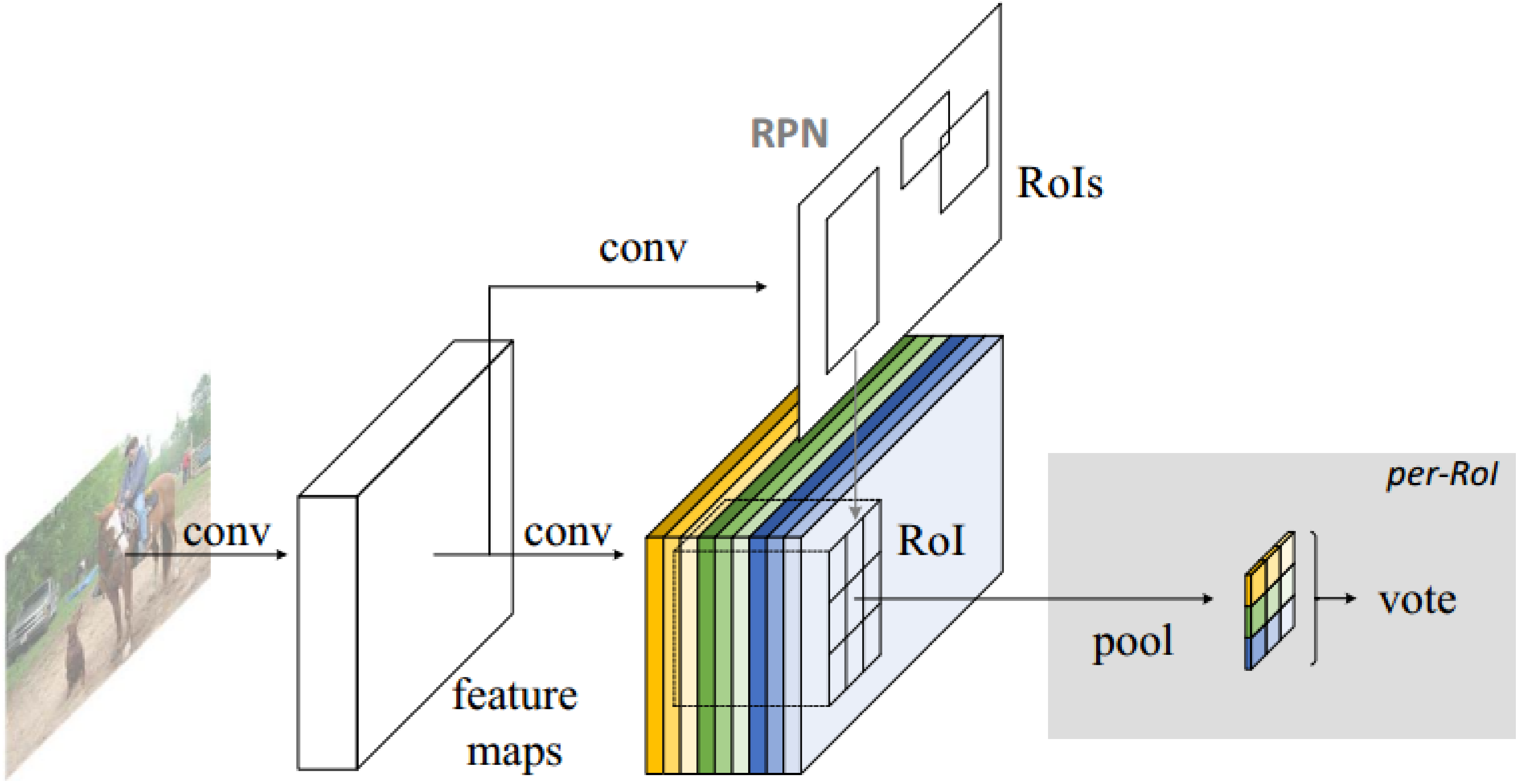


Figure 4: Visualization when an RoI does not correctly overlap the object.

Network Architecture



Results

- Test on PASCAL VOC 2007

	training data	mAP (%)	test time (sec/img)
Faster R-CNN [9]	07+12	76.4	0.42
Faster R-CNN +++ [9]	07+12+COCO	**85.6**	3.36
R-FCN	07+12	79.5	0.17
R-FCN multi-sc train	07+12	80.5	0.17
R-FCN multi-sc train	07+12+COCO	**83.6**	0.17

- Test on PASCAL VOC 2012

	training data	mAP (%)	test time (sec/img)
Faster R-CNN [9]	07++12	73.8	0.42
Faster R-CNN +++ [9]	07++12+COCO	**83.8**	3.36
R-FCN multi-sc train	07++12	77.6[†]	0.17
R-FCN multi-sc train	07++12+COCO	**82.0**[‡]	0.17

Contributions

- A simple but accurate and efficient framework for object detection
- Accuracy competitive with Faster R-CNN, but much faster.