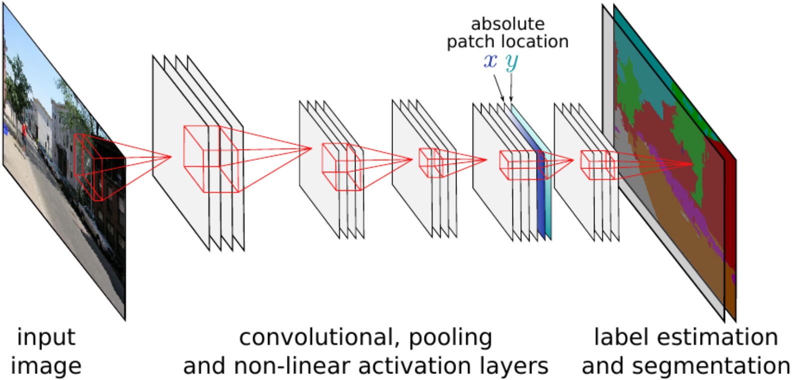
Fully Convolutional Neural Networks and Conditional Random Fields

Issam Laradji

Deep Convolutional Neural Networks

State of the art for many vision applications!



Deep Convolutional Neural Networks

State of the art for many vision applications!
Object Detection:

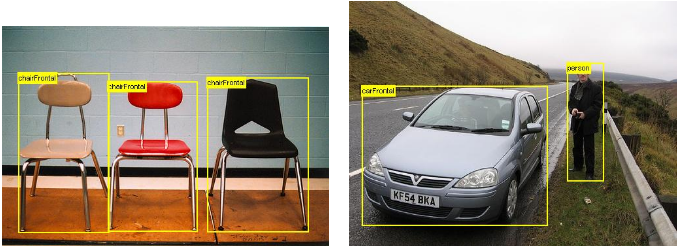


Figure 1: Detect the object

Deep Convolutional Neural Networks

State of the art for many vision applications!
Object Detection:

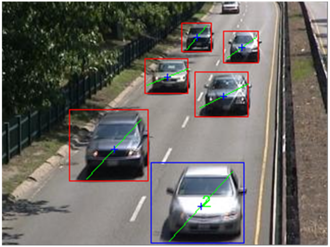
 

Figure 2: Detect the object

Deep Convolutional Neural Networks

State of the art for many vision applications!
Object Detection:

Figure 3: Detect the object

Deep Convolutional Neural Networks

State of the art for many vision applications!
Object Counting:

Figure 4: Number of objects in the image

Deep Convolutional Neural Networks

State of the art for many vision applications!
Object Segmentation:

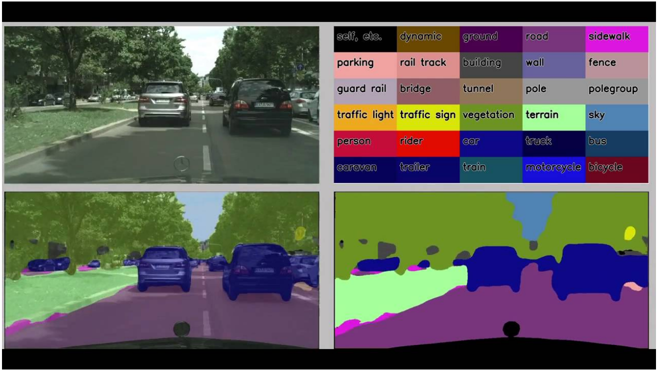


Figure 5: Predict the class of each pixel label

Problem Statement

Image segmentation using deep convolutional neural networks (CNN)

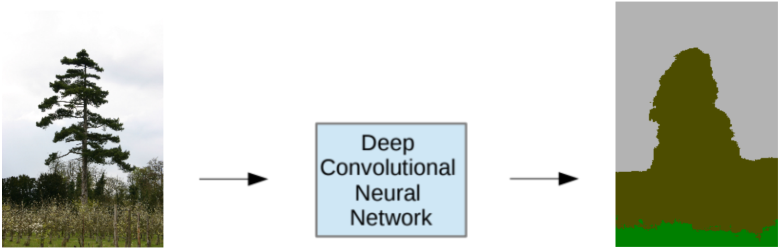


Figure 6: Using DNN to segment image

Problem Statement

Image segmentation using deep convolutional neural networks (CNN)

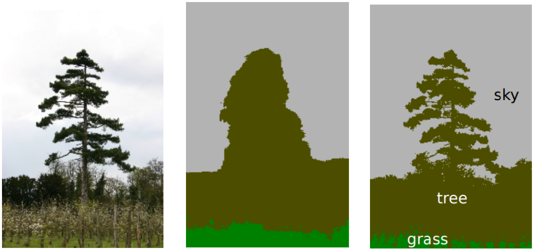


Figure 7: From left to right: (1) Original Image; (2) CNN Prediction; (3) Groundtruth

- Segmentation map: $\hat{Y} = \max_j \frac{\exp(f_j(X|\theta))}{\sum_k \exp(f_k(X|\theta))}$

Problem Statement

Image segmentation using deep convolutional neural networks (CNN)

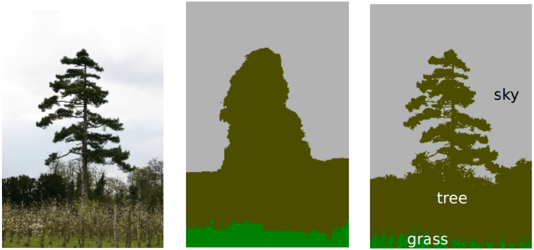


Figure 8: From left to right: (1) Original Image; (2) CNN Prediction; (3) Groundtruth

Segmentation map: $\hat{Y} = \max_j \frac{\exp(f_j(X|\theta))}{\sum_k \exp(f_k(X|\theta))}$

Prediction output can be made more accurate

- Take into account change of color pixel intensity (smoothness)

Problem Statement

Use conditional random fields:

- Smoothens noisy segmentation maps
- Energy function to minimize:

$$E(y) = \sum_i \underbrace{\phi_u(y_i)}_{\text{unary potential}} + \sum_{ij} \underbrace{\phi_p(y_i, y_j)}_{\text{pairwise potential}}$$

Problem Statement

Use conditional random fields:

- Smoothens noisy segmentation maps
- Energy function to minimize:

$$E(y) = \sum_i \underbrace{\phi_u(y_i)}_{\text{unary potential}} + \sum_{ij} \underbrace{\phi_p(y_i, y_j)}_{\text{pairwise potential}}$$

Unary Potentials:

$$\phi_u(y_i) = -\log P(y_i|X_i, \theta) = -\log \frac{\exp(f_{y_i}(X_i|\theta))}{\sum_k \exp(f_k(X_i|\theta))} \tag{1}$$

Pairwise Potentials:

$$\begin{aligned}\phi_p(y_i, y_j) = (y_i \neq y_j)[w_1 \exp(-\frac{||L_i - L_j||}{\sigma_L} - \frac{||I_i - I_j||}{\sigma_I}) + \\ w_2 \exp(-\frac{||L_i - L_j||}{\sigma_L})]\end{aligned} \tag{2}$$

Problem Statement

Use conditional random fields:

- Smoothens noisy segmentation maps
- Energy function to minimize:

$$E(y) = \sum_i \underbrace{\phi_u(y_i)}_{\text{unary potential}} + \sum_{ij} \underbrace{\phi_p(y_i, y_j)}_{\text{pairwise potential}}$$

- Encourages neighboring nodes to have the same label

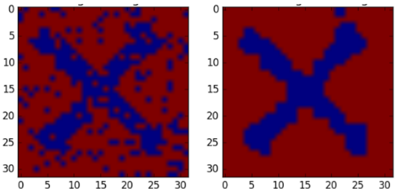


Figure 9: (Left) Original noisy image; (Right) Smooth output from minimizing E(Y)

Problem Statement

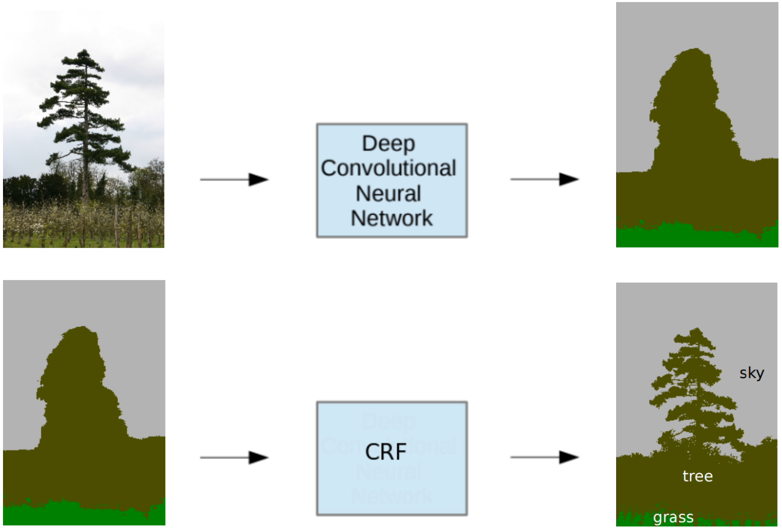


Figure 10: Penalty: $\phi_p(y_i, y_j) = (y_i \neq y_j)\exp\left(-\frac{||I_i - I_j||^2}{\sigma_I} - \frac{||L_i - L_j||^2}{\sigma_L}\right)$

Problem Statement

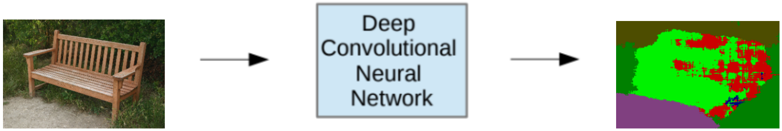
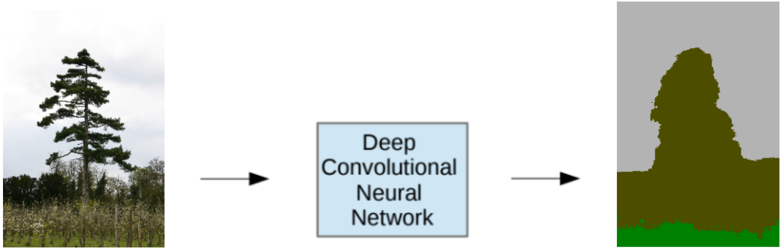


Figure 11: Fitting term:
$\phi_p(y_i) = -\log P(\hat{Y} = y_i) = -\log P(f(X|\theta) = y_i)$

Figure 12: Penalty: $\phi_p(y_i, y_j) = (y_i \neq y_j)\exp(-\frac{||I_i - I_j||^2}{\sigma_I} - \frac{||L_i - L_j||^2}{\sigma_L})$

Step 1: Optimizing the DNN step



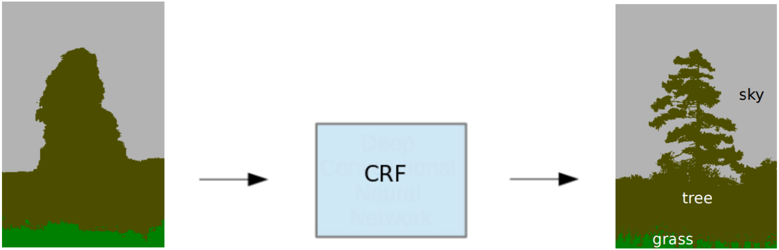
Optimize DNN on the training set:

$$\min_{\theta} ||f(X_i|\theta) - y_i||^2 \quad (X_i \text{ is an image, } Y_i \text{ is the image label}) \quad (3)$$

Equivalent to minimizing the unary potentials:

$$\hat{E}(\theta) = \sum_i ||f(X_i|\theta) - y_i||^2 = \sum_i \phi_u(y_i|\theta) \quad \text{(y here is the true label)} \quad (4)$$

Step 2: Optimizing the CRF step



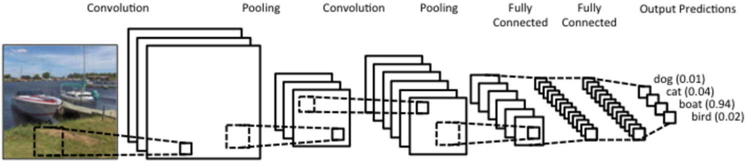
Step 1 optimize DNN on train images:

$$\hat{E}(\theta) = \sum_i ||f(X_i|\theta) - y_i||^2 = \sum_i \phi_u(y_i|\theta) \qquad \text{(y here is the true label)} \tag{5}$$

Step 2 optimize CRF on test images,

$$E(y) = \sum_i \phi_u(y_i) + \sum_{ij} \phi_p(y_i, y_j) \qquad \text{(y here is a parameter)} \tag{6}$$

Convolutional Neural Network for Classification



Objective function:

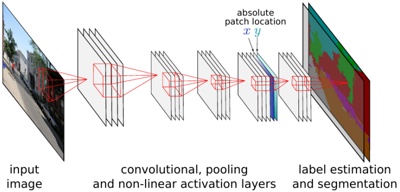
- Maximize

$$P(y_i = j|X_i, \theta) = \frac{\exp(f(X_i|\theta))}{\sum_k \exp(f(X_i|\theta))} \quad \forall i \qquad (7)$$

- Alternatively optimize (uncostrained optimization)

$$\min_{\theta} -\sum_i \log P(y_i = j|X_i, \theta) \quad \forall i \qquad (8)$$

Fully Convolutional Network For Segmentation



Objective function:

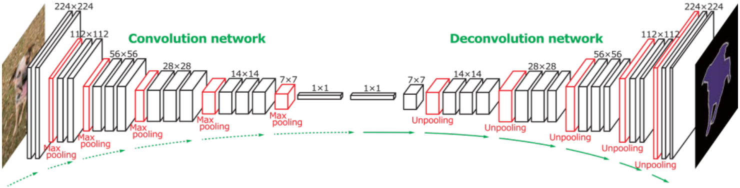
- Maximize

$$P(Y_{ip} = j | X_i, \theta) = \frac{\exp(f_p(X_i|\theta))}{\sum_k \exp(f_p(X_i|\theta))} \quad \forall i, p \qquad (9)$$

- Alternatively optimize (uncostrained optimization)

$$\min_\theta - \sum_i \sum_p \log P(Y_{ip} = j | X_i, \theta) \quad \forall i, p \qquad (10)$$

Fully Convolutional Network For Segmentation



Objective function:

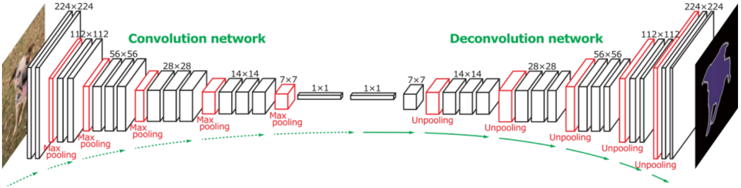
- Maximize

$$P(Y_{ip} = j|X_i) = \frac{\exp(f_p(X_i|\theta))}{\sum_k \exp(f_p(X_i|\theta))} \quad \forall i, p \tag{11}$$

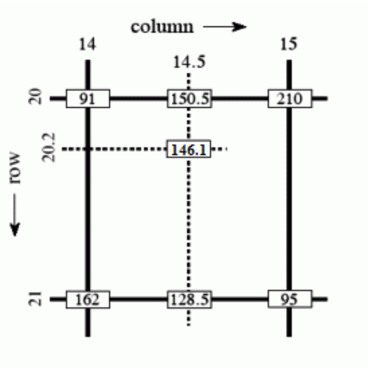
- Alternatively optimize (uncostrained optimization)

$$\min_{\theta} - \sum_i \sum_p \log P(Y_{ip} = j|X_i) \quad \forall i, p \tag{12}$$

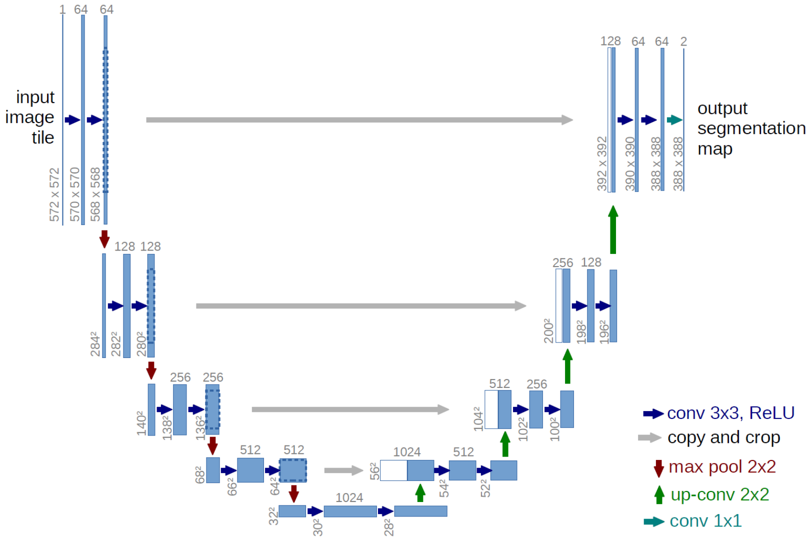
Fully Convolutional Network For Segmentation



Interpolation:

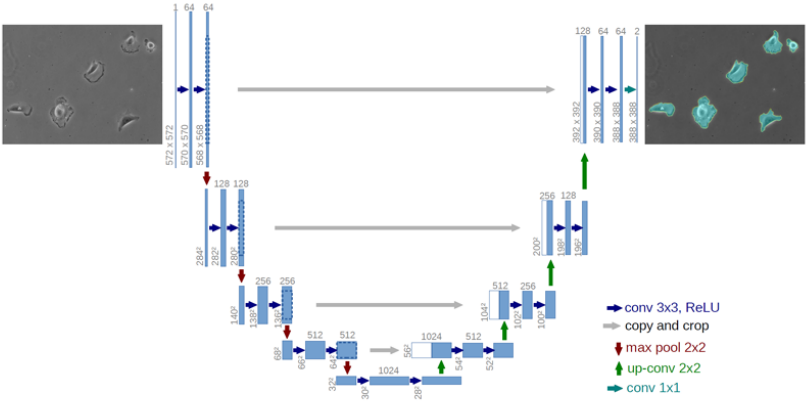


U-Net



1
64
64
input
image
tile
572 x 572
570 x 570
568 x 568
128
128
284^2
282^2
280^2
256
256
140^2
138^2
136^2
512
512
68^2
66^2
64^2
1024
32^2
30^2
28^2
1024
512
56^2
54^2
52^2
512
256
104^2
102^2
100^2
256
128
200^2
198^2
196^2
128
64
64
2
392 x 392
390 x 390
388 x 388
388 x 388
output
segmentation
map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

U-Net


1 64 64
572 x 572
570 x 570
568 x 568
128 128
256 256
512 512
1024
1024
512
512 256
256 128
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

U-Net

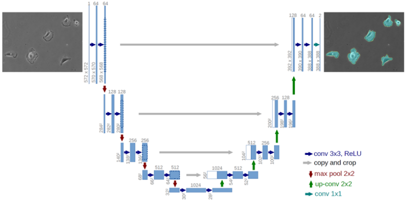
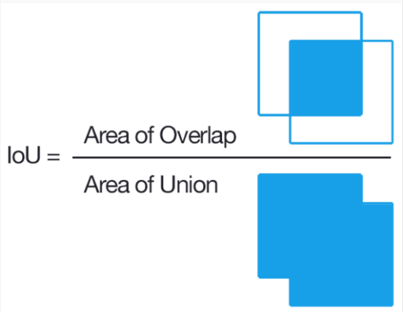


Figure 13: State of the art on the ISBI cell tracking challenge 2015.

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**

Figure 14: Segmentation results (IOU) on the ISBI cell tracking challenge 2015.

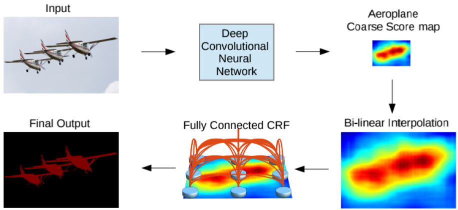
U-Net



Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**

Figure 15: Segmentation results (IOU) on the ISBI cell tracking

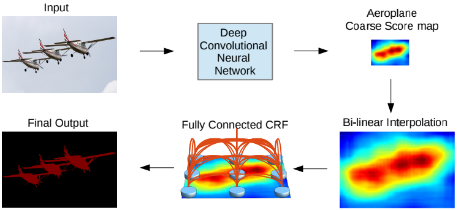
FCN + Conditional Random Fields



Objective function:

$$\min_{y} E(y) = \min \sum_{i} \phi_u(y_i) + \sum_{ij} \phi_p(y_i, y_j) \qquad (13)$$

FCN + Conditional Random Fields



Objective function:

$$\min_y E(y) = \min \sum_i \phi_u(y_i) + \sum_{ij} \phi_p(y_i, y_j)$$
$$\phi_u(y_i) = -\log P(y_i|X_i, \theta) = -\log \frac{\exp(f_{y_i}(X_i|\theta))}{\sum_k \exp(f(X_i|\theta))} \tag{14}$$

FCN + Conditional Random Fields

Energy Term:

$$\min_{y} E(y) = \min \sum_{i} \phi_u(y_i) + \sum_{ij} \phi_p(y_i, y_j) \tag{15}$$

Unary Potentials:

$$\phi_u(y_i) = -\log P(y_i|X_i, \theta) = -\log \frac{\exp(f_{y_i}(X_i|\theta))}{\sum_k \exp(f(X_i|\theta))} \tag{16}$$

Pairwise Potentials:

$$\begin{aligned}\phi_p(y_i, y_j) = (y_i \neq y_j)[w_1 \exp(-\frac{||L_i - L_j||}{\sigma_L} - \frac{||I_i - I_j||}{\sigma_I})+ \\ w_2 \exp(-\frac{||L_i - L_j||}{\sigma_L})]\end{aligned} \tag{17}$$

Conditional Random Field

CRF formulation:

$$\min_{y \in \{0,1\}^n} E(y) = \min_{y \in \{0,1\}^n} \sum_i \phi_u(y_i) + \sum_{ij} \phi_p(y_i, y_j) \tag{18}$$

Mean Field Approximation

- Do not compute the exact distribution $P(Y)$ of the CRF
- Compute a distribution $Q(Y)$ where,
  - Q minimizes $KL(Q||P)$; and
  - $Q(Y) = \prod_i Q_i(Y_i)$ (independent marginals)

Conditional Random Field

CRF formulation:

$$\min_{y\in\{0,1\}^n} E(y) = \min_{y\in\{0,1\}^n} \sum_i \phi_u(y_i) + \sum_{ij} \phi_p(y_i, y_j) \tag{19}$$

$$\begin{aligned}\min_{y\in\{0,1\}^n} E(y) = \min_{y\in\{0,1\}^n} \sum_i - \log \frac{\exp(f(X_i|\theta))}{\sum_k \exp(f(X_i|\theta))} + \\ \sum_{ij}(y_i \neq y_j) w \cdot k(y_i, y_j)\end{aligned} \tag{20}$$

$$k(y_i, y_j) = \exp(-\frac{||L_i - L_j||}{\sigma_L} - \frac{||I_i - I_j||}{\sigma_I}) + w_2 \exp(-\frac{||L_i - L_j||}{\sigma_L}) \tag{21}$$

Conditional Random Field

CRF formulation:

$$\min_{y \in \{0,1\}^n} E(y) = \min_{y \in \{0,1\}^n} \sum_i \phi_u(y_i) + \sum_{ij} \phi_p(y_i, y_j) \tag{22}$$

The iterative step from minimizing the KL divergence w.r.t. Q:

$$Q_i(c) = \frac{1}{Z_i}(-\phi_u(y_i) - \sum_{c'}(c \neq c')w \sum_j k(y_i, y_j)Q_j(c')) \tag{23}$$

Consteraints

- $Q_i(y_i)$ is a valid distribution (positive and adds up to one)
- $Q(y) = \prod_i Q_i(y_i)$ (independent marginals)

Conditional Random Field

The iterative step from minimizing the KL divergence w.r.t. Q:

$$Q_i(c) = \frac{1}{Z_i}(-\phi_u(y_i) - \sum_{c'}(c \neq c')w \sum_j k(y_i, y_j)Q_j(c')) \qquad (24)$$

Data: Image I, DNN output segmentation maps y
Result: $Q(Y)$, an approximation of the CRF distribution
Initialize Q;
while *while not converged do* **do**
    $\bar{Q}_i(y_i = c) \leftarrow \sum_j k(y_i, y_j)Q_j(y_i = c)$ Message passing $O(n^2)$;
    $\hat{Q}_i(y_i) \leftarrow \sum_{c \in C}(y_i \neq c)w\bar{Q}_i(y_i = c)$ Compat. transform $O(n)$;
    $Q_i(y_i) \leftarrow \exp\{-\phi_u(y_i) - \hat{Q}_i(y_i)\}$ Local update $O(n)$;
    normalize $Q_i(y_i)$
end

Algorithm 1: Mean field in fully connected CRFs

Example

CRF formulation:

$$\min_{y\in\{0,1\}^n} E(y) = \min_{y\in\{0,1\}^n} \sum_i \phi_u(y_i) + \sum_{ij} \phi_p(y_i, y_j) \tag{25}$$

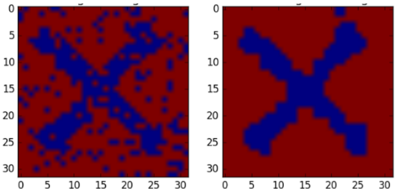


Figure 16: (Left) Original noisy image; (Right) Smooth output from minimizing E(Y)

Example

CRF formulation:

$$\min_{y \in \{0,1\}^n} E(y) = \min_{y \in \{0,1\}^n} \sum_i -\log P(y_i | X_i, \theta) + \sum_{ij} (y_i \neq y_j) w \cdot k(y_i, y_j) \tag{26}$$

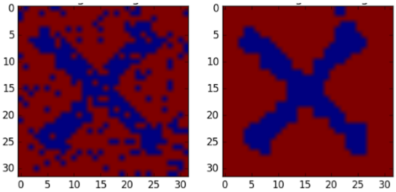


Figure 17: (Left) Original noisy image; (Right) Smooth output from minimizing E(Y)

PASCAL Dataset for Segmentation

	train		val	
	img	**obj**	**img**	**obj**
Aeroplane	88	108	90	110
Bicycle	65	94	79	103
Bird	105	137	103	140
Boat	78	124	72	108
Bottle	87	195	96	162
Bus	78	121	74	116
Car	128	209	127	249
Cat	131	154	119	132
Chair	148	303	123	245
Cow	64	152	71	132
Diningtable	82	86	75	82
Dog	121	149	128	150
Horse	68	100	79	104
Motorbike	81	101	76	103
Person	442	868	445	865
Pottedplant	82	151	85	171
Sheep	63	155	57	153
Sofa	93	103	90	106
Train	83	96	84	93
Tvmonitor	84	101	74	98
Total	1464	3507	1449	3422

PASCAL Dataset for Segmentation

PASCAL Dataset for Segmentation

Deep Convolutional Neural Networks

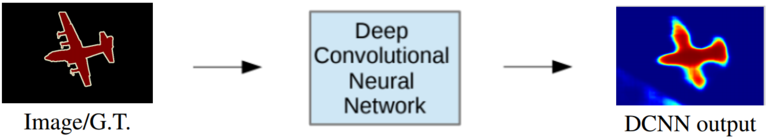


Figure 18: DNN is based on VGG16, a popular CNN architecture [1]

[1] Liang-Chieh, Chen, et al. "Semantic image segmentation with deep convolutional nets and fully connected crfs."

Deep Convolutional Neural Networks with CRF

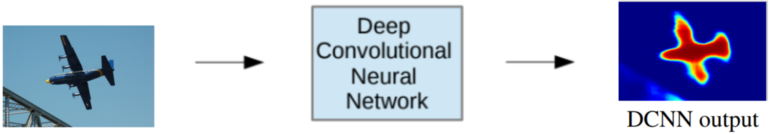


Figure 19: DNN is based on VGG16, a popular CNN architecture

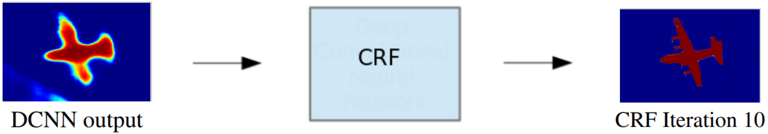
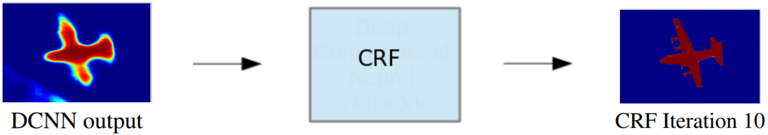


Figure 20: After CRF 10 iterations

Results



Method	mean IOU (%)
DeepLab	59.80
DeepLab-CRF	63.74

Figure 21: Results - IOU

Conclusion

- Deep convolutional neural networks
  - State-of-the-art for many vision tasks
- Conditional random fields helps refine the output segmentation map
  - Use mean-field approximation to estimate the CRF distribution
  - Every iteration naively has $O(n^2)$ time complexity
- State-of-the-art results on image segmentation in PASCAL VOC 2012 dataset