Holistically-Nested Edge Detection (HED)

Saining Xie, Zhuowen Tu

Presented by Yuxin Wu

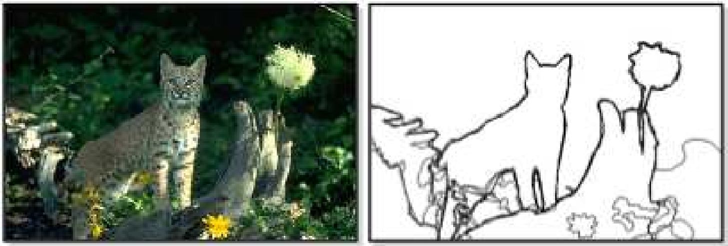
February 10, 2016

What is an Edge?

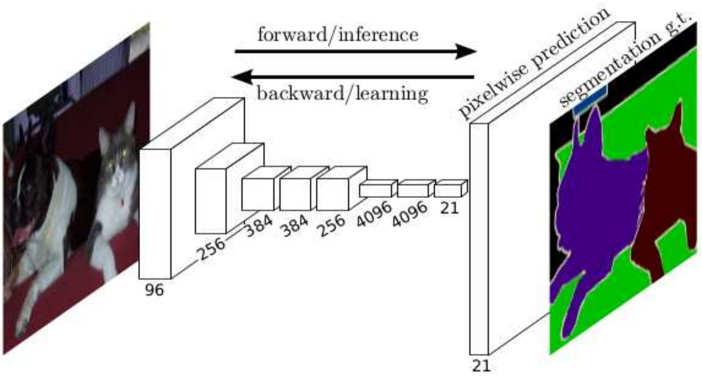
- **Local** intensity change? Used in traditional methods: Canny, Sobel, etc.
- **Learn it!**

What is an Edge?

- **Local** intensity change? Used in traditional methods: Canny, Sobel, etc.
- **Learn it!**

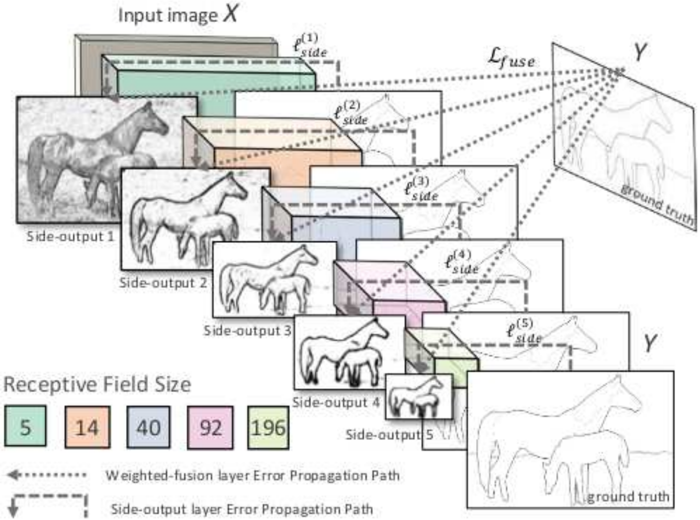


Fully Convolutional Network (FCN)

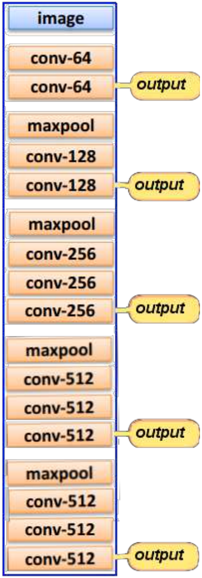


- Concept originally brought out for semantic segmentation
- No fully-connected layers (can be converted)
- Allow inputs of any sizes

Holistically-Nested architecture

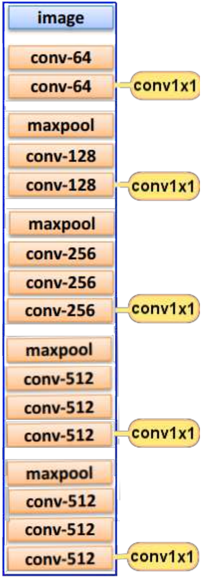


Multiple Supervision Signals



- Single output, multiple cost
- Learn earlier, learn better
- Alleviate gradient vanishing

Convolutional Layers



Fine-tuning from VGG16:

- Lots of people do fine-tuning on top of VGG16.
- 5 stage. 3x3 convolution only.
- HED adds a side output (conv1x1) after each stage.

Upsampling by Deconvolution

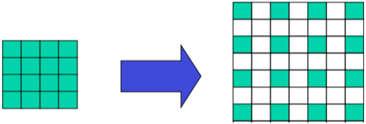
Upsampling by a factor of $k \in \mathbb{N}^+$ is implemented by a **deconvolution** with a $2k \times 2k$ kernel and output stride k.

An mathematically equivalent explanation (assume $k = 2$):

1. Input image with shape n
2. Zero-filled upsample as above, by a factor of 2. Shape becomes $2n - 1$
3. Convolve with a filter $\begin{bmatrix} \frac{1}{16} & \frac{3}{16} & \frac{3}{16} & \frac{1}{16} \\ \frac{3}{16} & \frac{9}{16} & \frac{9}{16} & \frac{3}{16} \\ \frac{3}{16} & \frac{9}{16} & \frac{9}{16} & \frac{3}{16} \\ \frac{1}{16} & \frac{3}{16} & \frac{3}{16} & \frac{1}{16} \end{bmatrix}$ with padding $= 3$, shape becomes $(2n - 1) + 3 = 2n + 2$. Then center-crop to $2n$

Upsampling by Deconvolution

Upsampling by a factor of $k \in \mathbb{N}^+$ is implemented by a **deconvolution** with a $2k \times 2k$ kernel and output stride k.
An mathematically equivalent explanation (assume $k = 2$):



1. Input image with shape n
2. Zero-filled upsample as above, by a factor of 2. Shape becomes $2n - 1$
3. Convolve with a filter $\begin{bmatrix} \frac{1}{16} & \frac{3}{16} & \frac{3}{16} & \frac{1}{16} \\ \frac{3}{16} & \frac{9}{16} & \frac{9}{16} & \frac{3}{16} \\ \frac{3}{16} & \frac{9}{16} & \frac{9}{16} & \frac{3}{16} \\ \frac{1}{16} & \frac{3}{16} & \frac{3}{16} & \frac{1}{16} \end{bmatrix}$ with padding = 3, shape becomes $(2n - 1) + 3 = 2n + 2$. Then center-crop to $2n$

Class-Balanced Sigmoid Cross Entropy Loss

Sigmoid Cross Entropy Loss

For each pixel, loss $L = -[y^\star \log(y) + (1 - y^\star)\log(1 - y)]$
where ground truth label $y^\star \in \{0, 1\}, y = \dfrac{1}{1 + e^{-z}}$

In images, 90% pixels are not edge, cost function is dominated by negative labels.
To avoid this, re-weight the terms:

Class-Balanced Sigmoid Cross Entropy Loss

$L = -[\beta y^\star \log(y) + (1 - \beta)(1 - y^\star)\log(1 - y)]$
where β is the ratio of **negative** ground truth labels in this batch of data

This loss function is computed for $\ell_{1..5}$ as well as $\ell_{fuse} = \sum_{i=1}^{5} \alpha_i \ell_i$

Class-Balanced Sigmoid Cross Entropy Loss

Sigmoid Cross Entropy Loss

For each pixel, loss $L = -[y^\star \log(y) + (1 - y^\star) \log(1 - y)]$
where ground truth label $y^\star \in \{0, 1\}, y = \dfrac{1}{1 + e^{-z}}$

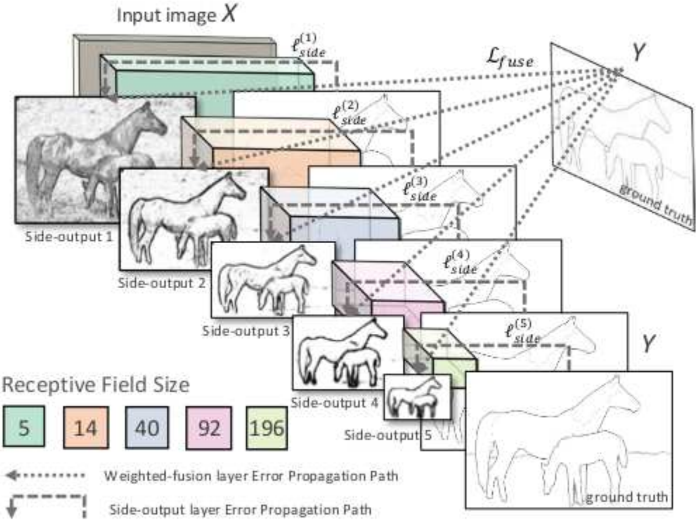
In images, 90% pixels are not edge, cost function is dominated by negative labels.
To avoid this, re-weight the terms:

Class-Balanced Sigmoid Cross Entropy Loss

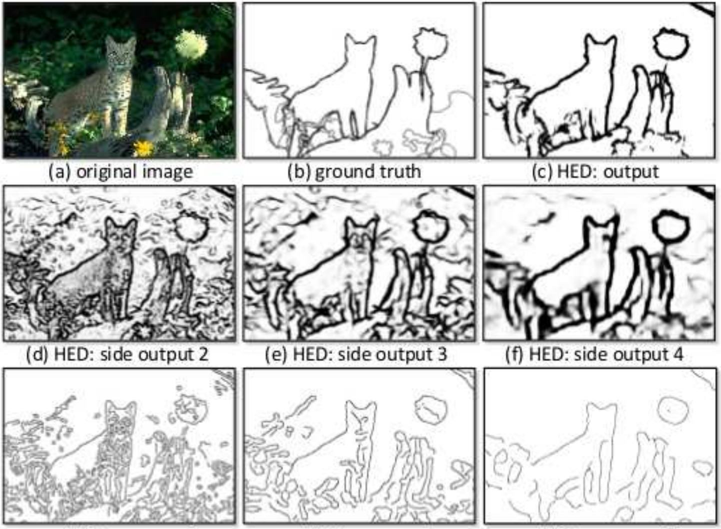
$L = -[\beta y^\star \log(y) + (1 - \beta)(1 - y^\star) \log(1 - y)]$
where β is the ratio of **negative** ground truth labels in this batch of data

This loss function is computed for $\ell_{1..5}$ as well as $\ell_{fuse} = \sum_{i=1}^{5} \alpha_i \ell_i$

Holistically-Nested architecture



Outputs



(a) original image
(b) ground truth
(c) HED: output
(d) HED: side output 2
(e) HED: side output 3
(f) HED: side output 4
(g) Canny: $\sigma = 2$
(h) Canny: $\sigma = 4$
(i) Canny: $\sigma = 8$

Qualitative Results

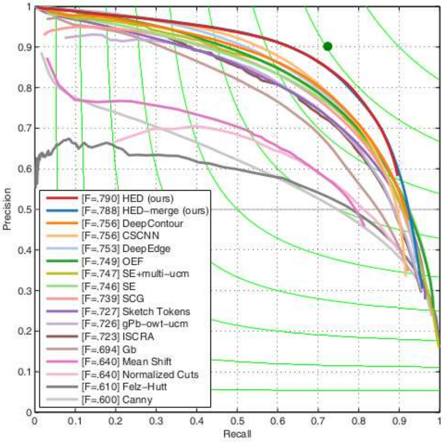
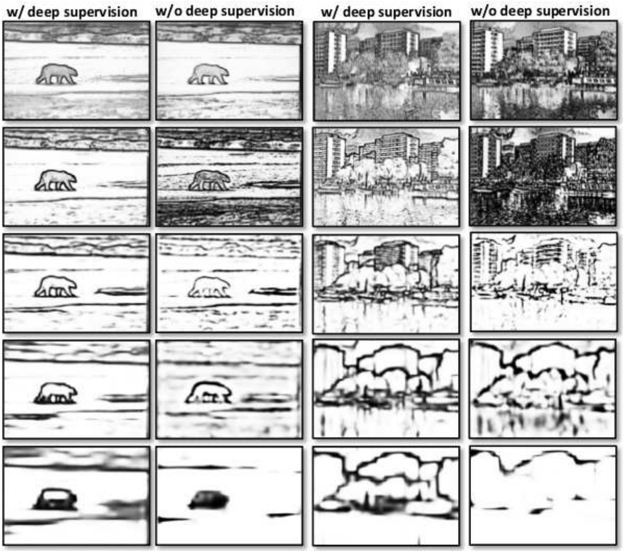


Figure: Results on BSD500 (a small dataset)

Effect of Supervision



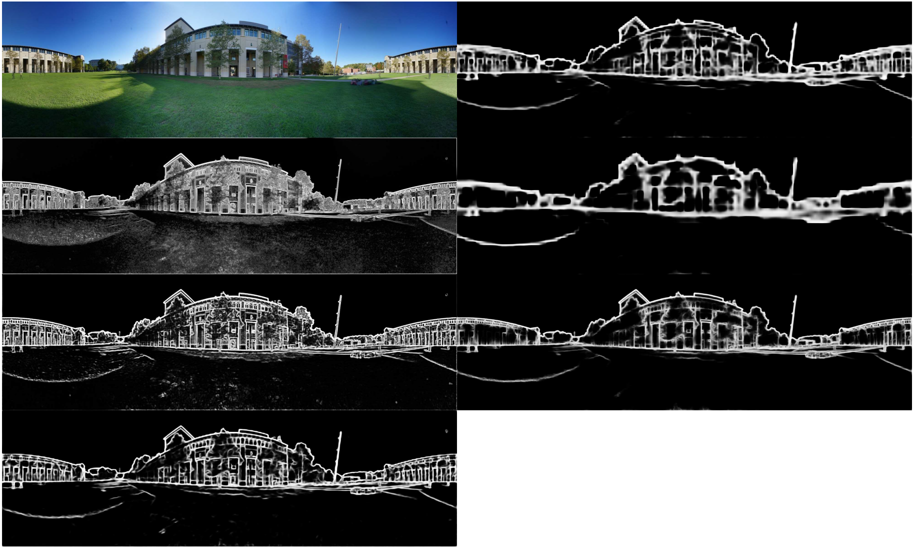
Effect of Supervision

Figure: Output of 2nd stage with(left) and without(right) extra supervision

Misc.

- Rotation/flip/scaling as data augmentation
- Using depth information (in NYUD dataset) gives better performance
- Pure FCN / HED without multiple supervision don't work as good
- 2.5 fps on K40 for 320×480 input

CMU Pano



Thanks!

Yuxin Wu