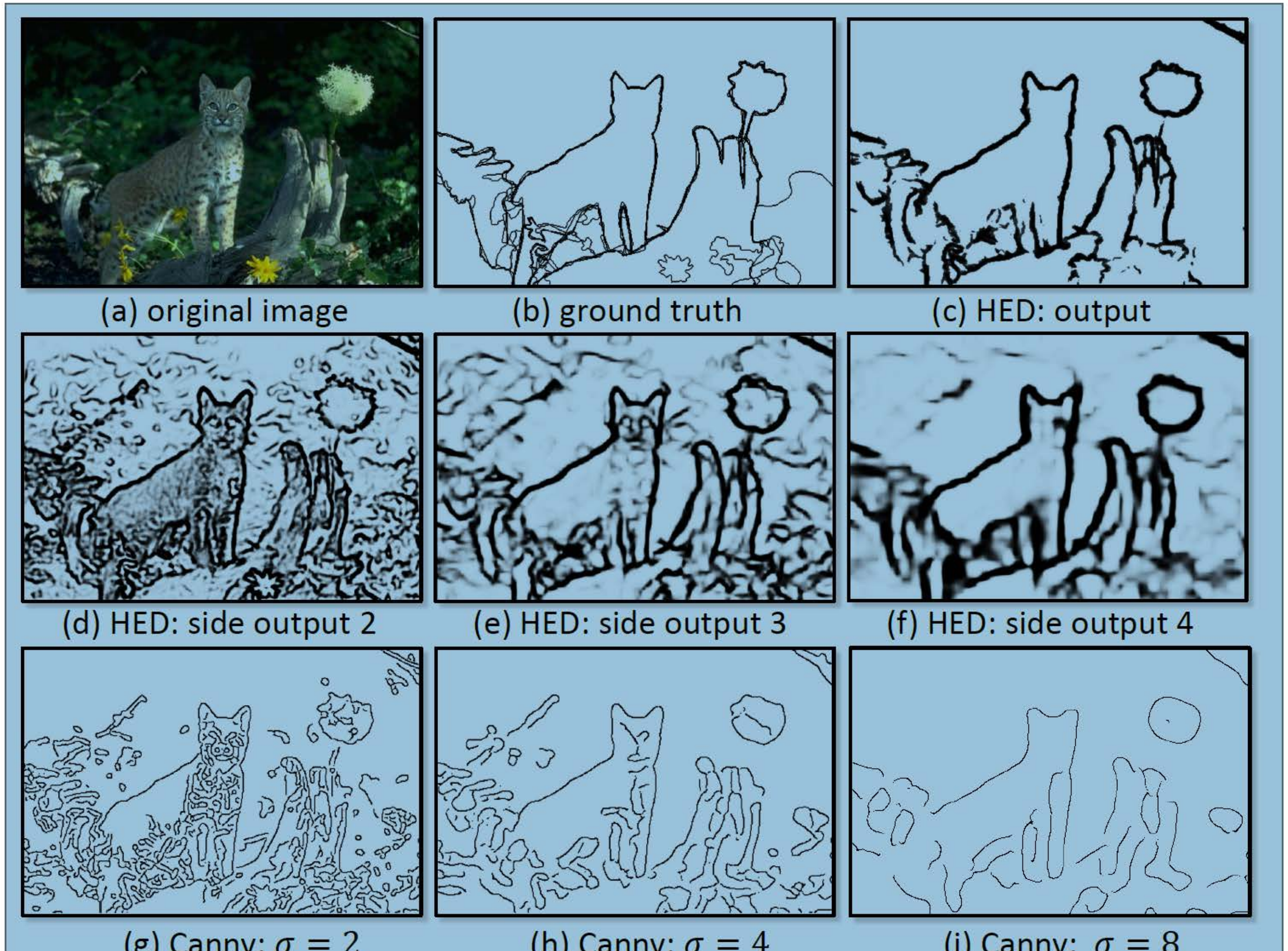
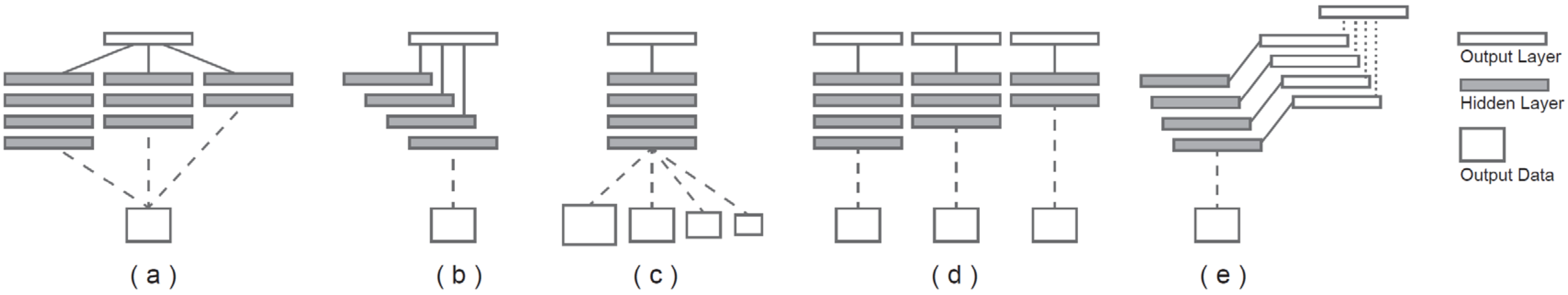
The HED Edge Detector:
Holistically-Nested Edge Detector

Basic Idea: Edge Detection at Multiple Scale



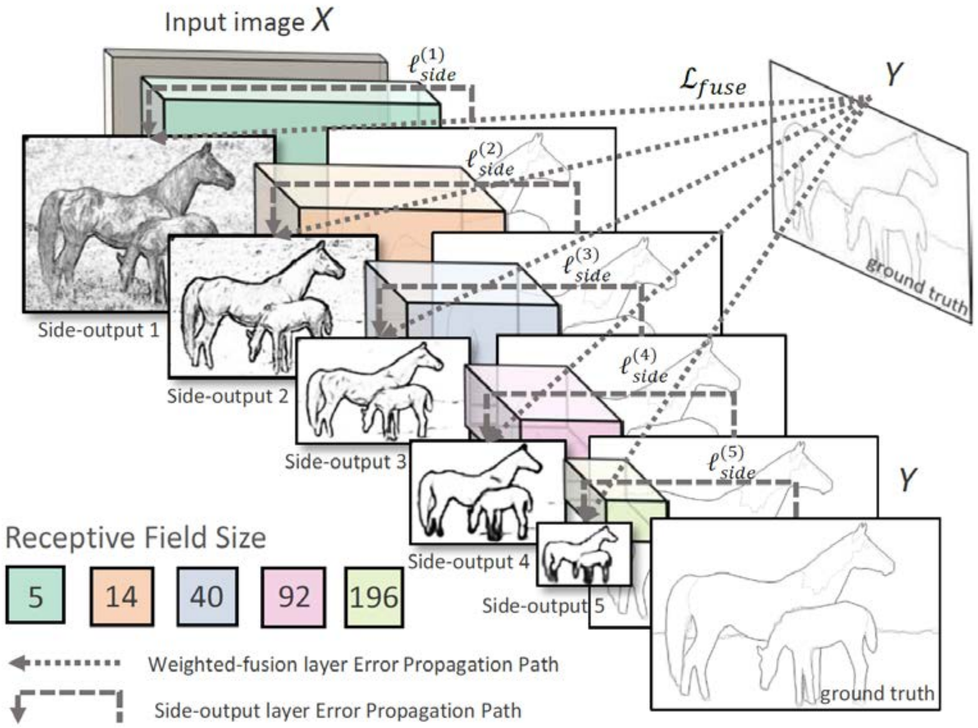
Varieties of Deep Network Architectures

- HED uses architecture (e). Exploits the fact that there are cues for edges at different scales.



Network Architecture for Edge Detection

- Deep Supervision at each side-output layer.



We consider the objective function

$$\mathcal{L}_{\text{side}}(\mathbf{W}, \mathbf{w}) = \sum_{m=1}^{M} \alpha_m \ell_{\text{side}}^{(m)}(\mathbf{W}, \mathbf{w}^{(m)}), \tag{1}$$

where ℓ_{side} denotes the image-level loss function for side-outputs. In our image-to-image training, the loss function is computed over all pixels in a training image $X = (x_j, j = 1, \ldots, |X|)$ and edge map $Y = (y_j, j = 1, \ldots, |X|), y_j \in$

Balancing the Loss between Edges and Non-Edges

- Roughly 90% of pixels in images and non-edges. Must balance this to reward true positives (edges).

$$\ell_{\text{side}}^{(m)}(\mathbf{W}, \mathbf{w}^{(m)}) = -\beta \sum_{j \in Y_+} \log \Pr(y_j = 1 | X; \mathbf{W}, \mathbf{w}^{(m)}) - (1 - \beta) \sum_{j \in Y_-} \log \Pr(y_j = 0 | X; \mathbf{W}, \mathbf{w}^{(m)}) \qquad (2)$$

where $\beta = |Y_-|/|Y|$ and $1 - \beta = |Y_+|/|Y|$. $|Y_-|$ and $|Y_+|$