边缘检测年度进展概述

Ming-Ming Cheng

Media Computing Lab, Nankai University

Email: cmm@nankai.edu.cn

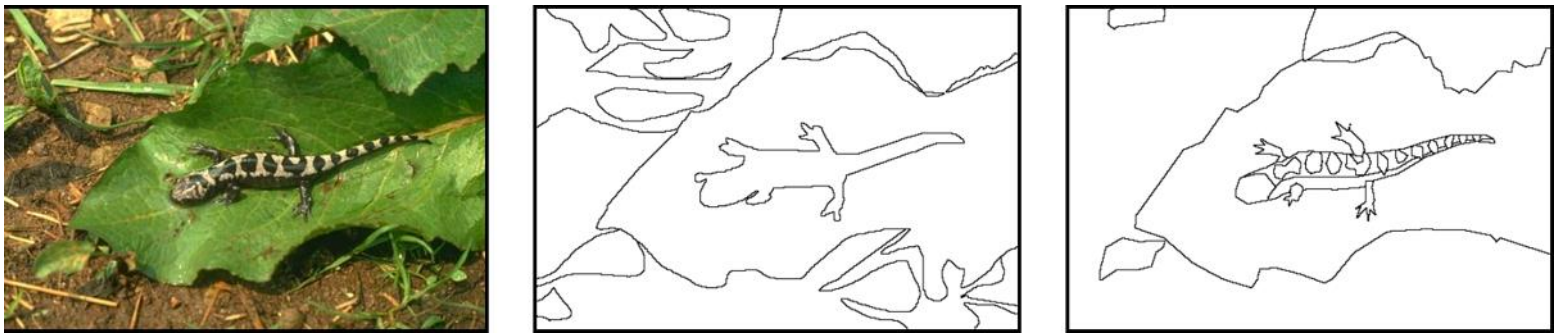
URL: http://mmcheng.net/

Early pioneering methods

- Locate sharp changes in the intensity function
  - Sobel, Canny, Laplacian, …

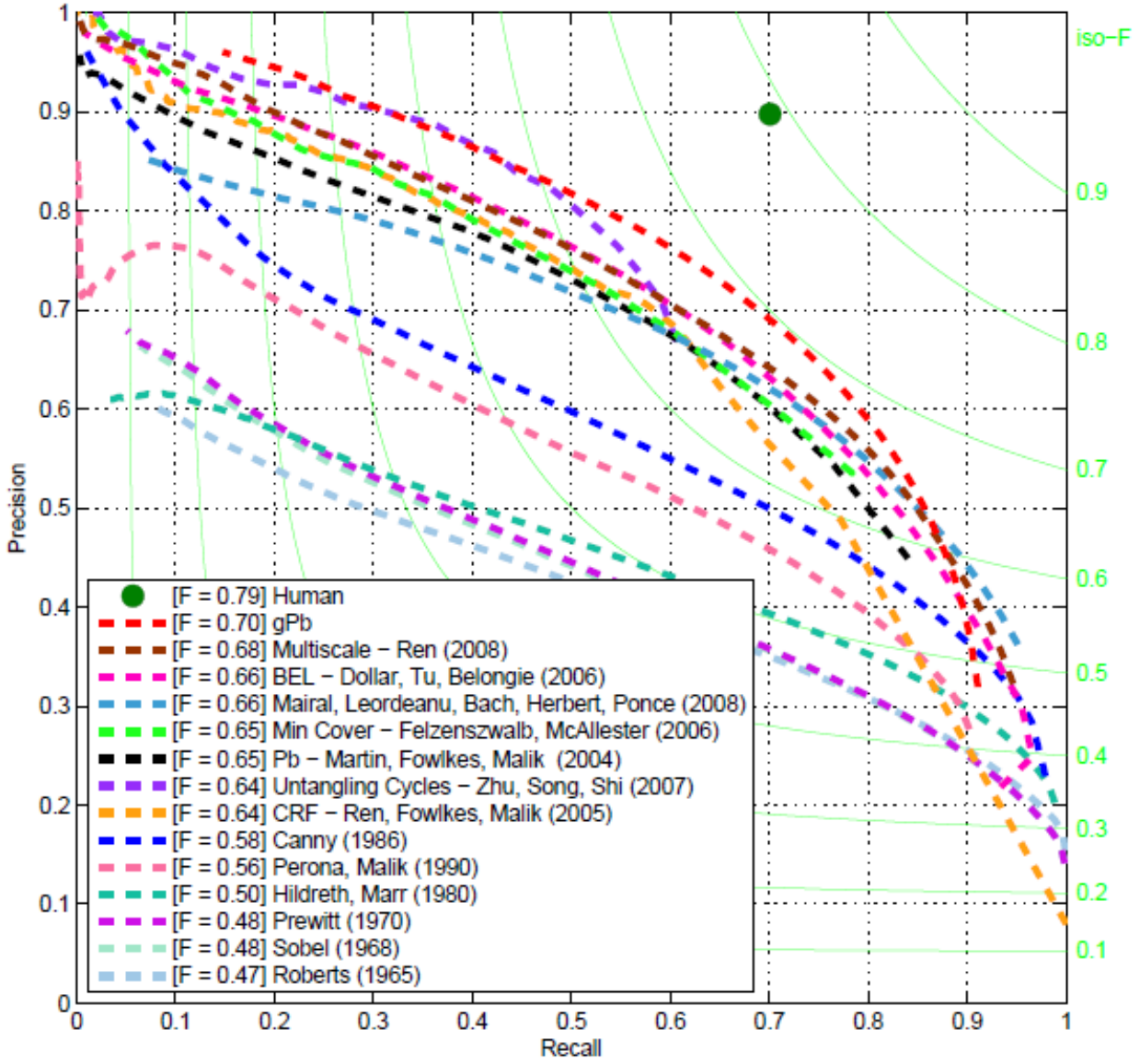
Data driven methods

- Learning the probability distributions of features
  - Konishi *et al.* [1] proposed the first data-driven method
  - Low-level cues such as color, intensity, gradient, texture, etc.
- Employ various classification paradigm
  - Popular method: Pb, gPb, StructuredEdge
- Berkeley segmentation Dataset



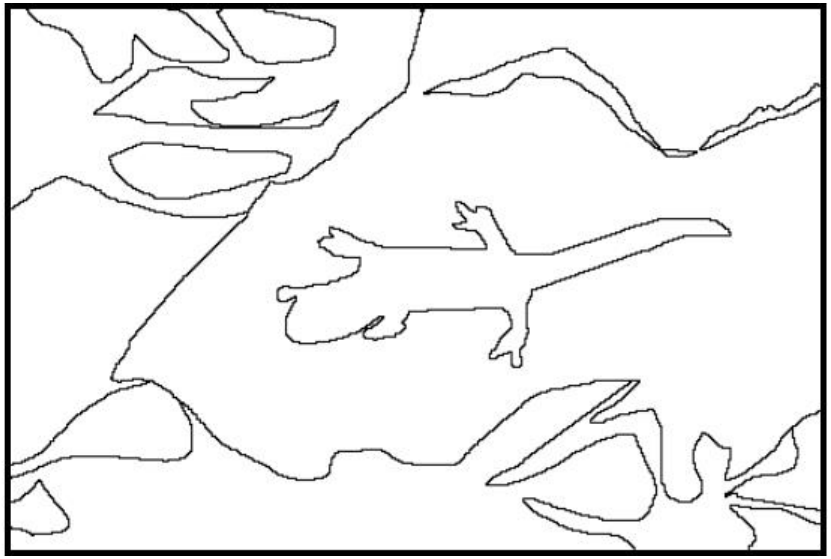
[1] Konishi S, Yuille A L, Coughlan J M, et al. Statistical edge detection: Learning and evaluating edge cues[J]. IEEE TPAMI, 2003, 25(1): 57-74.

45 years of boundary detection



Limitations of previous methods

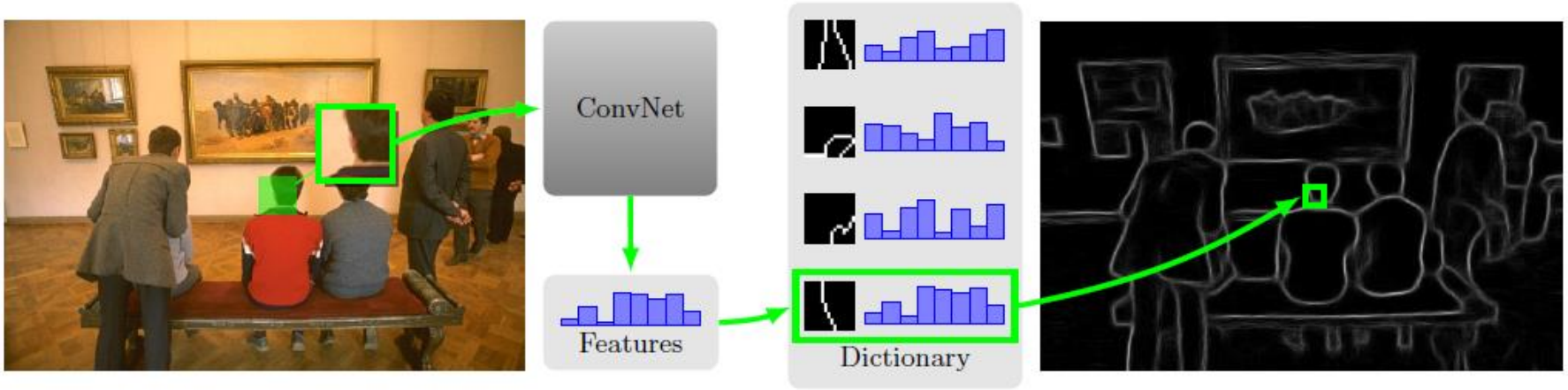
- Semantically meaningful edge detection
  - Requires object-level (high level) information



Recent CNN based methods

- N^4-Fields, ACCV 2014
- DeepEdge, CVPR 2015
- DeepContour, CVPR 2015
- HFL, ICCV 2015
- HED, ICCV 2015
- RCF, CVPR 2017

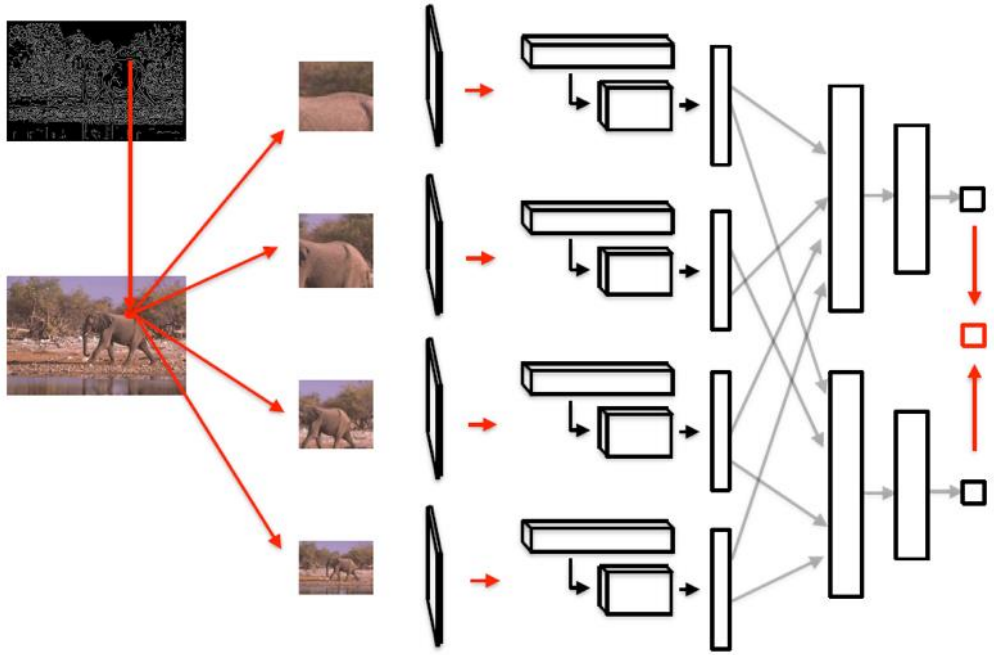
N^4-Fields



Ganin Y, Lempitsky V. N^4-fields: Neural network nearest neighbor fields for image transforms[C]//ACCV. Springer International Publishing, 2014: 536-551.

DeepEdge

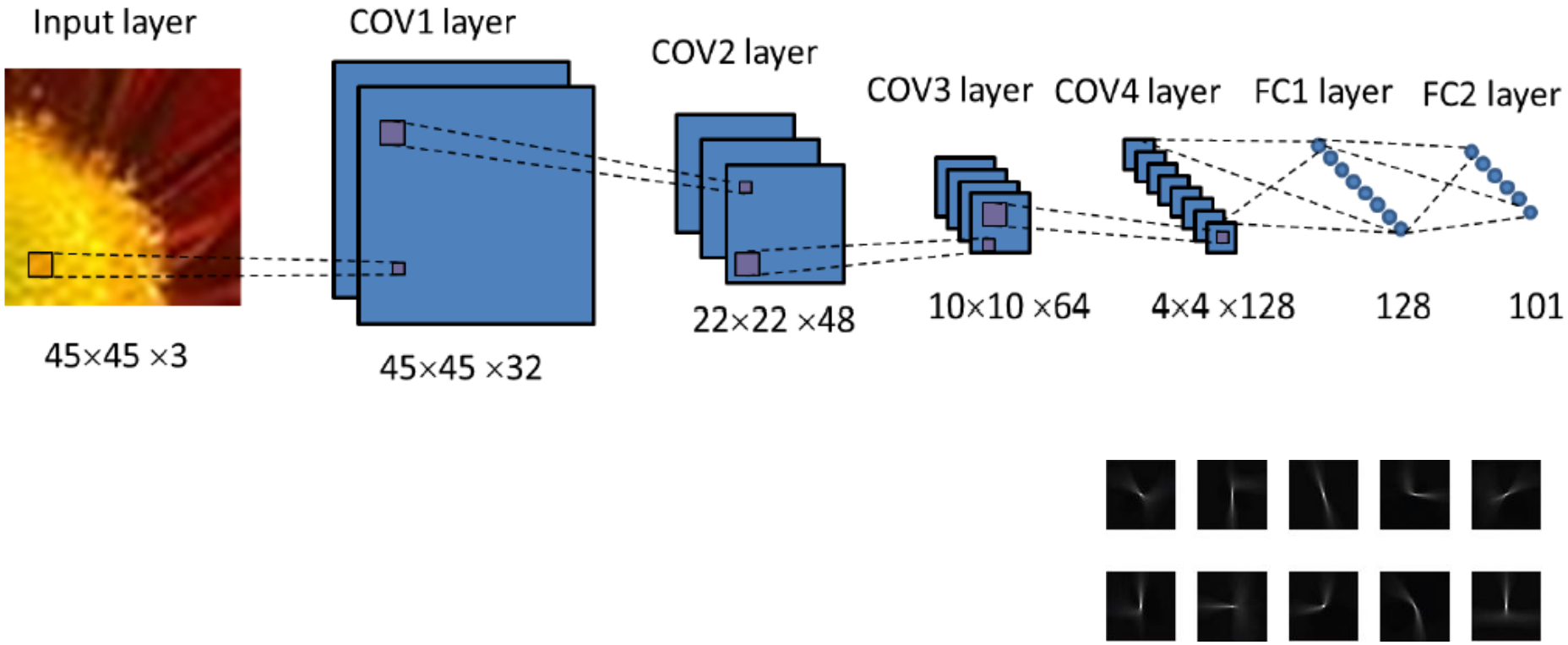
- Candidate contour points ← Canny edge
- Patches at 4 scales run through the CNN
- Two branches for: classification & regressor.



Bertasius G, Shi J, Torresani L. Deepedge: A multi-scale bifurcated deep network for top-down contour detection[C]//CVPR. 2015: 4380-4389.

DeepContour

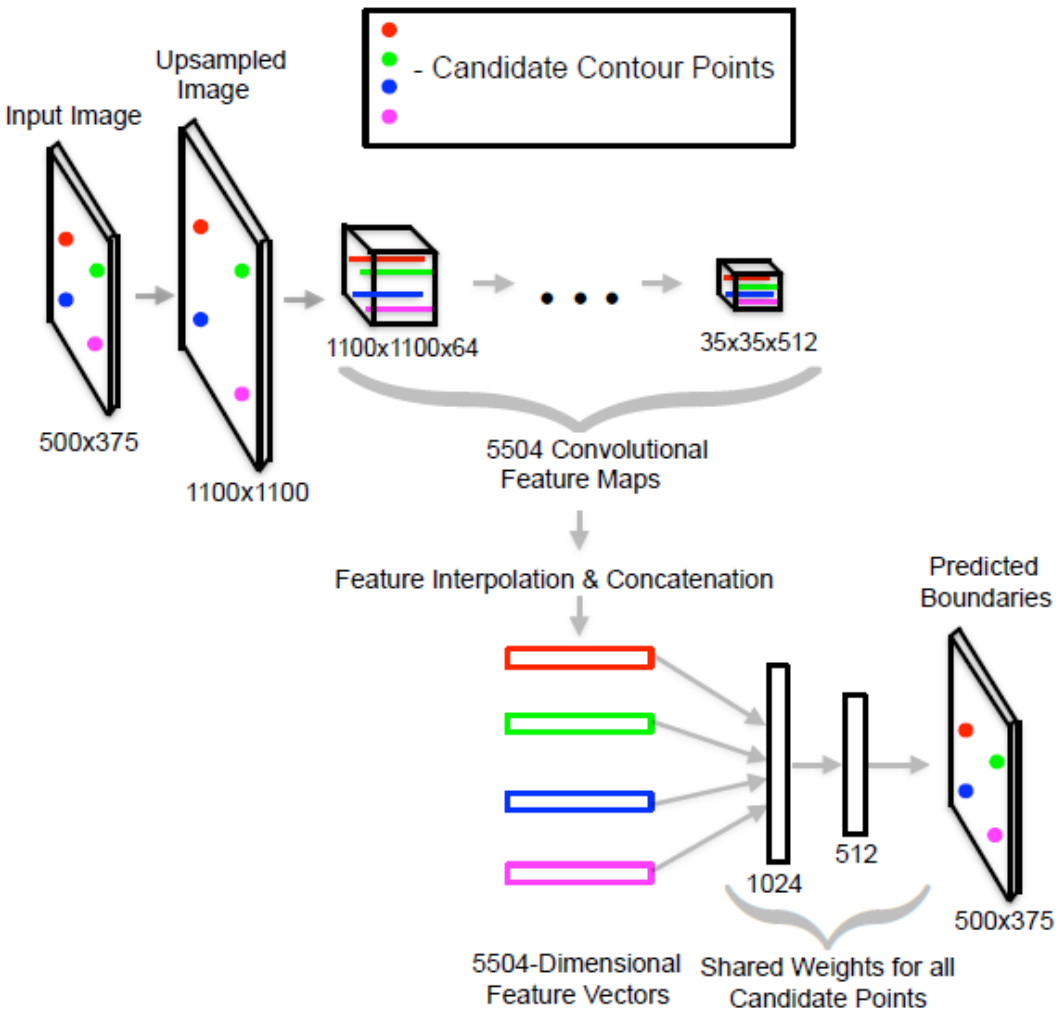
- Multi-class shape classification
- New CNN loss for positive vs. negative



Shen W, Wang X, Wang Y, et al. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection[C]//CVPR. 2015: 3982-3991.

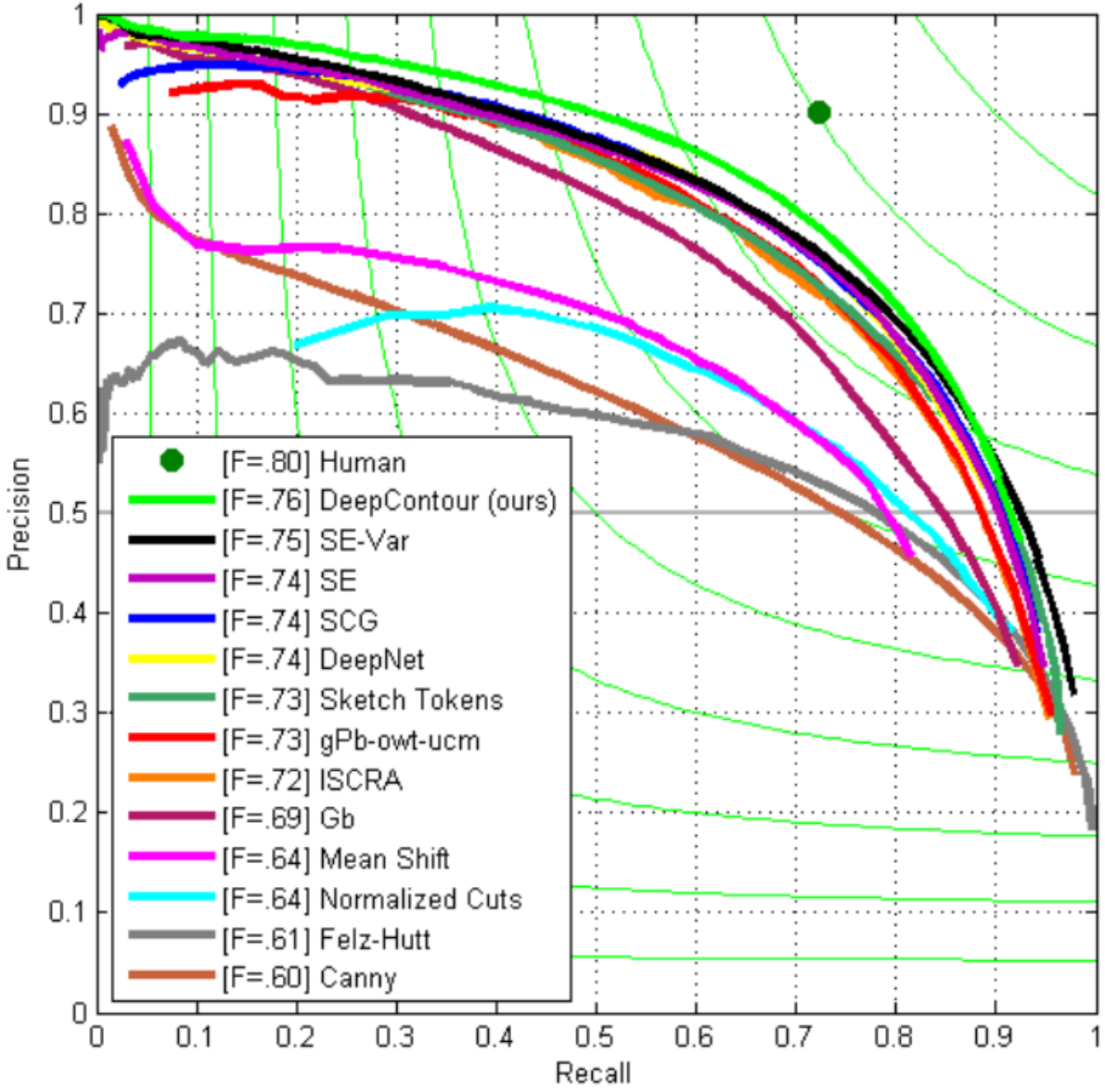
HFL

- Candidate contour points
- Unsample image to feed through VGG net, which was pre-trained for high level task



Bertasius G, Shi J, Torresani L. High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision[C]//ICCV. 2015: 504-512.

Results of making local decisions



HED

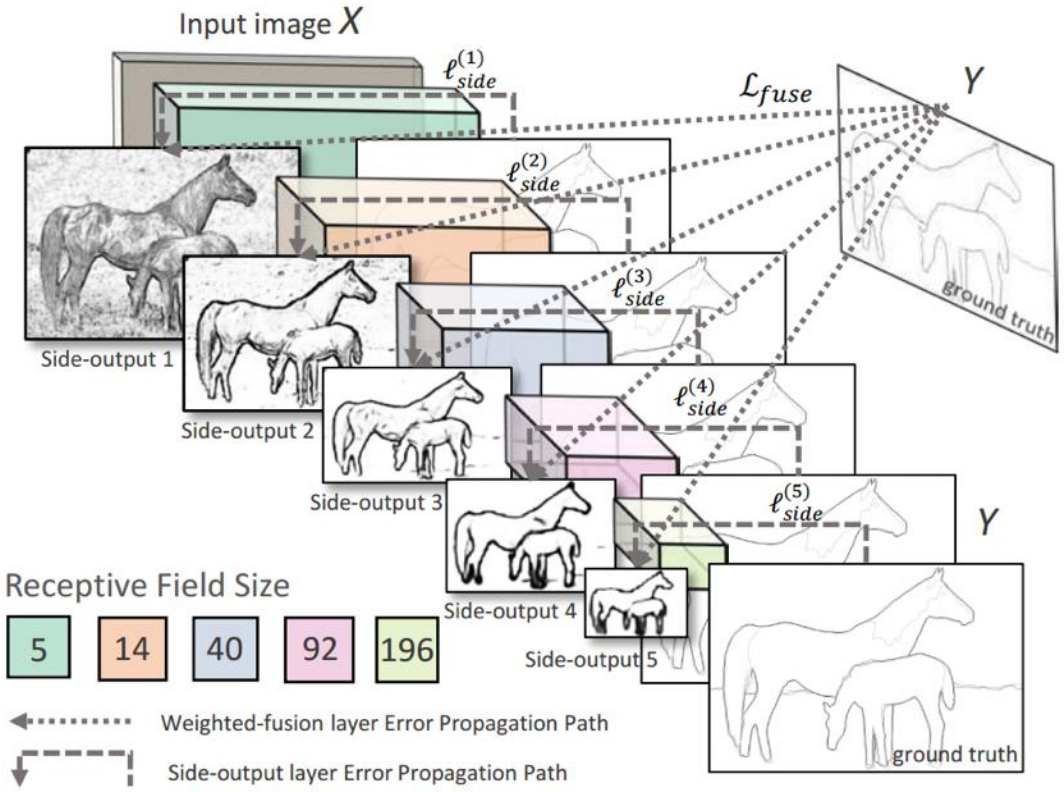
- Holistically-Nested Edge Detection
  - *Holistic: image to image fashion*

(a) original image (b) ground truth (c) HED: output

Xie S, Tu Z. Holistically-nested edge detection[C]//ICCV. 2015: 1395-1403.

HED

- Holistically-Nested Edge Detection
  - *Holistic: image to image fashion*
  - *Multi-scale and multi-level feature learning*



HED

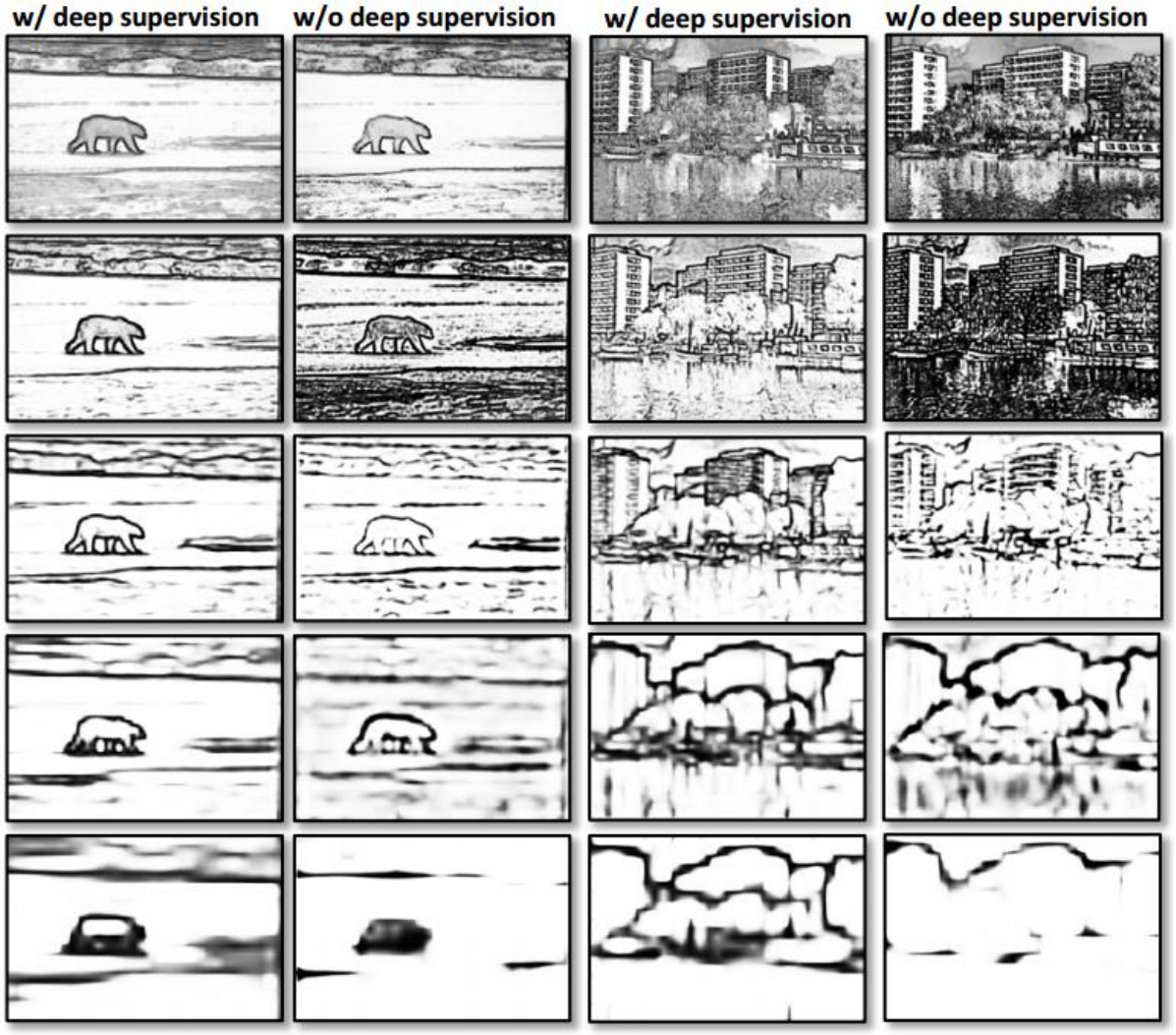
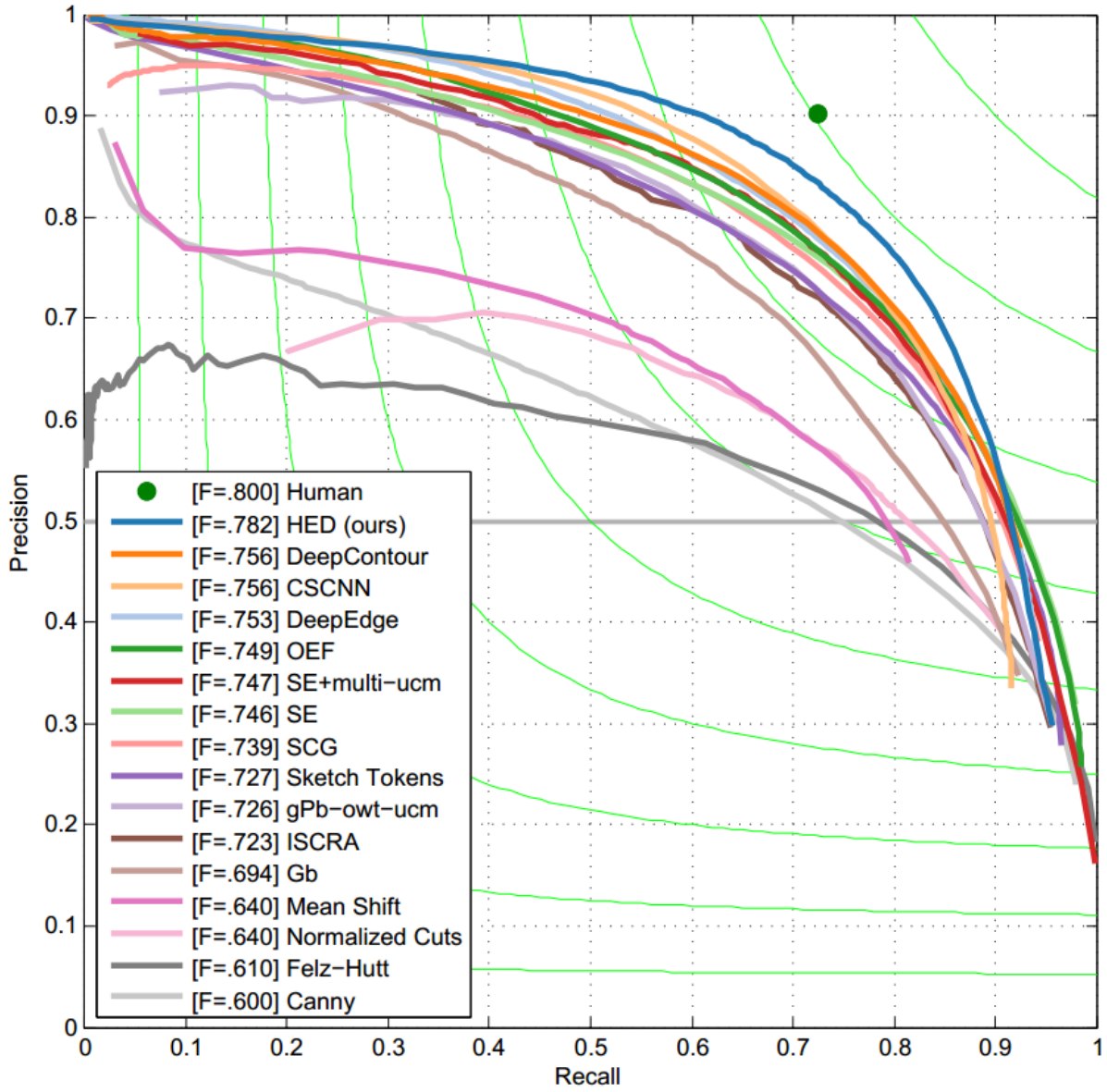
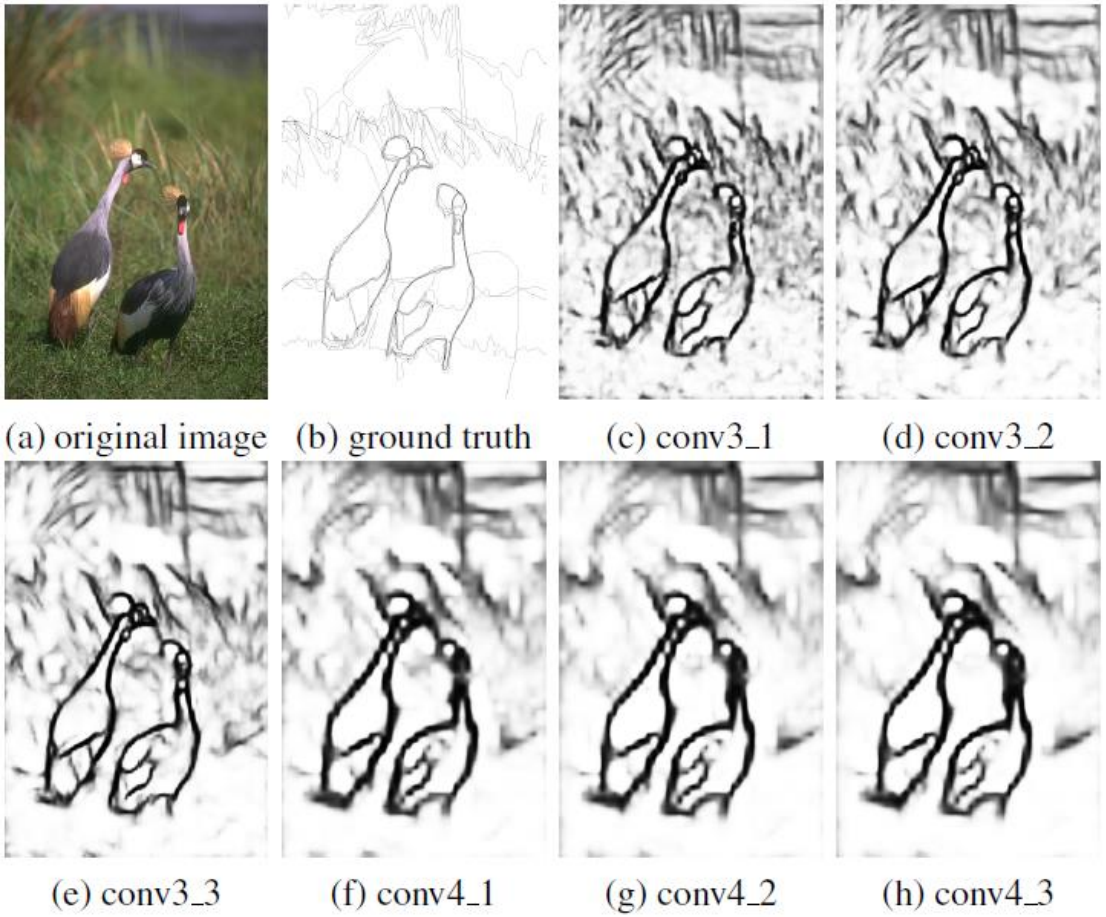


Illustration: how deep supervision helps side-output layers to produce multi-scale dense predictions.

Results


Precision
Recall
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
[F=.800] Human
[F=.782] HED (ours)
[F=.756] DeepContour
[F=.756] CSCNN
[F=.753] DeepEdge
[F=.749] OEF
[F=.747] SE+multi-ucm
[F=.746] SE
[F=.739] SCG
[F=.727] Sketch Tokens
[F=.726] gPb-owt-ucm
[F=.723] ISCRA
[F=.694] Gb
[F=.640] Mean Shift
[F=.640] Normalized Cuts
[F=.610] Felz-Hutt
[F=.600] Canny

RCF: Richer Convolutional Features

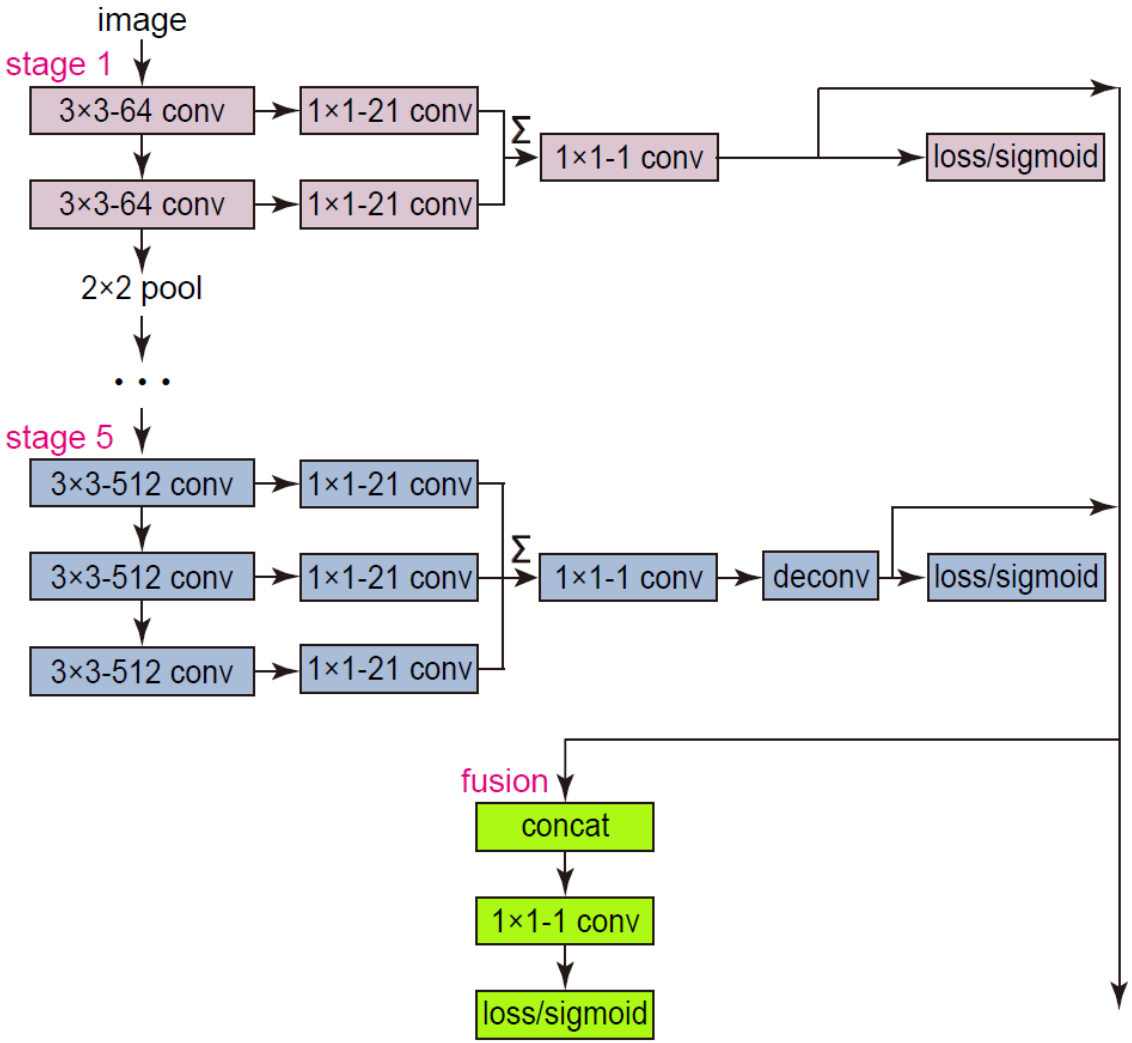


Motivation: intermediate layers contain lots of useful fine details

Key idea: combining ALL the meaningful convolutional features

Liu Y, Cheng M M, Hu X, et al. Richer Convolutional Features for Edge Detection[C]//CVPR. 2017.

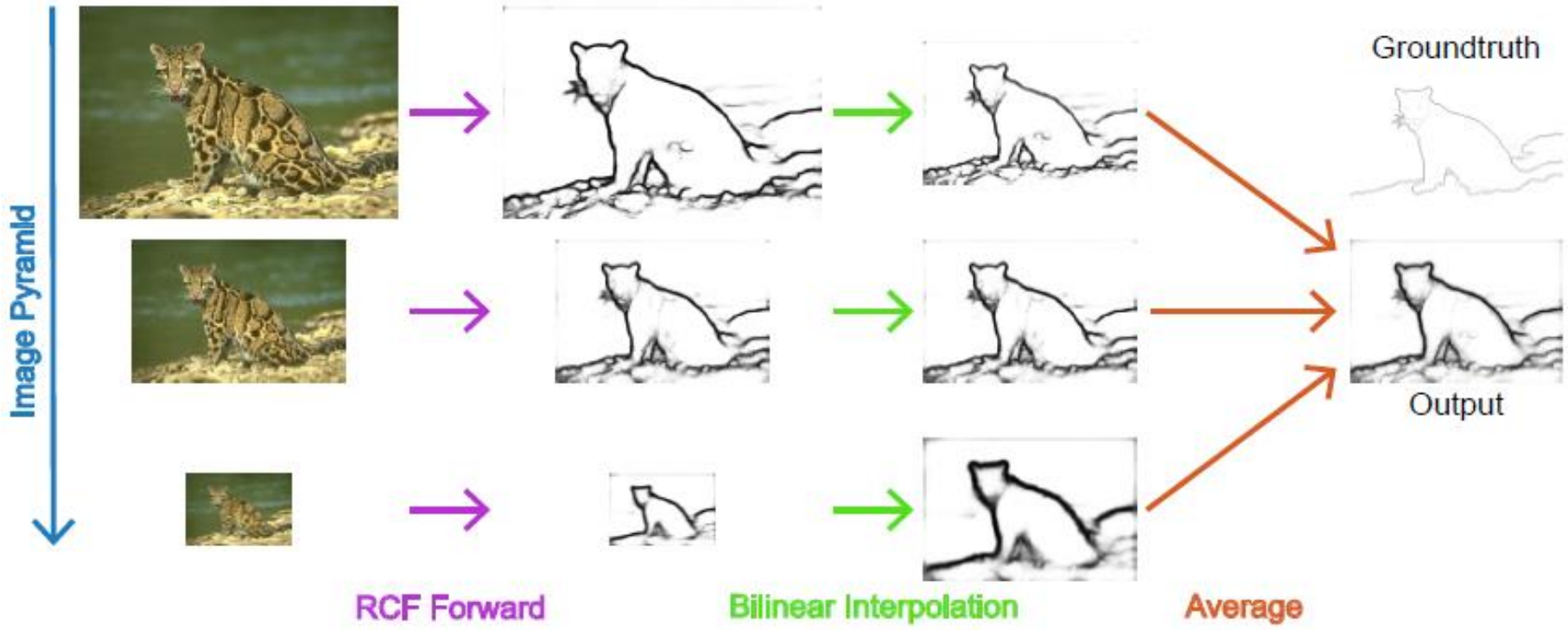
RCF: Richer Convolutional Features



Use **ALL** convolutional layer, instead of last layer of each stage.

Liu Y, Cheng M M, Hu X, et al. Richer Convolutional Features for Edge Detection[C]//CVPR. 2017.

Multiscale Fusion



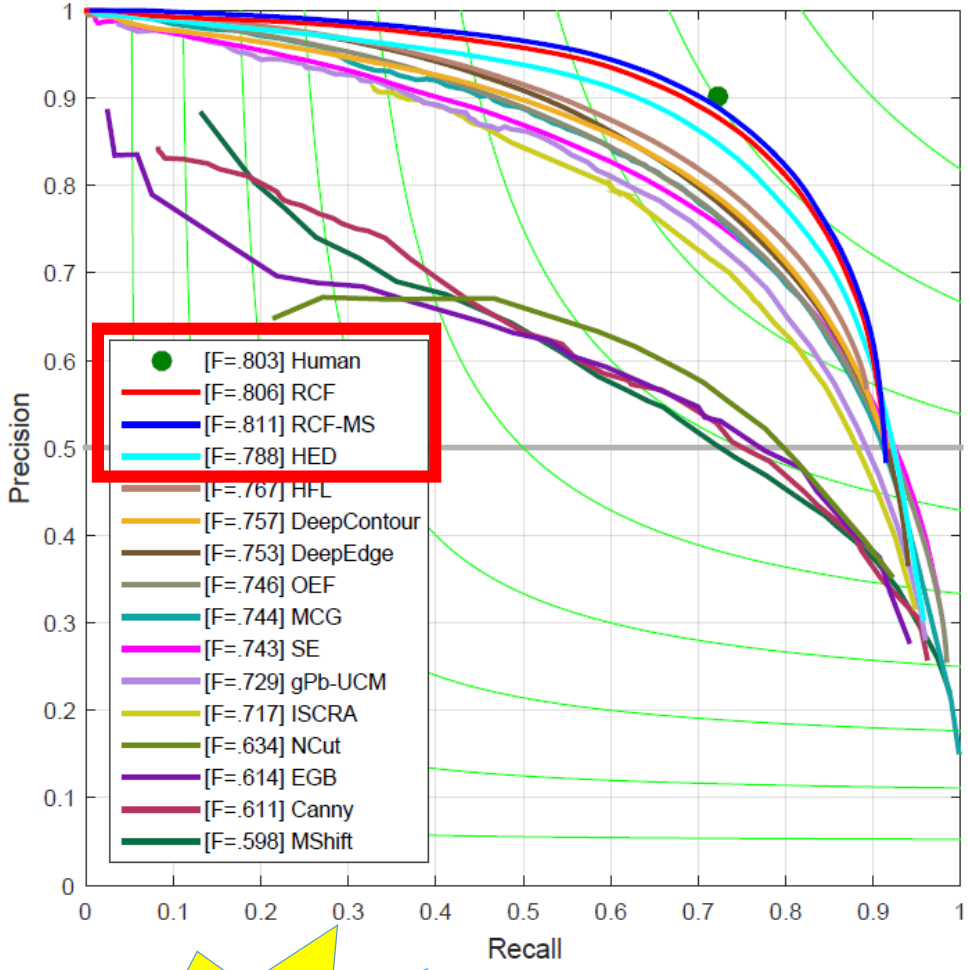
Explicit multi-scale processing.

Liu Y, Cheng M M, Hu X, et al. Richer Convolutional Features for Edge Detection[C]//CVPR. 2017.

Samples

image G-Truth results

Evaluation on BSDS500



Method	ODS	OIS	FPS
Canny	.611	.676	28
EGB	.614	.658	10
MShift	.598	.645	1/5
gPb-UCM	.729	.755	1/240
Sketch Tokens	.727	.746	1
MCG	.744	.777	1/18
SE	.743	.763	2.5
OEF	.746	.770	2/3
DeepContour	.757	.776	1/30†
DeepEdge	.753	.772	1/1000†
HFL	.767	.788	5/6†
N^4-Fields	.753	.769	1/6†
HED	.788	.808	**30**†
RDS	.792	.810	**30**†
CEDN	.788	.804	10†
RCF	.806	.823	**30**†
RCF-MS	**.811**	**.830**	8†

Open Source

Source code: https://github.com/yun-liu/rcf