Deep learning for dense per-pixel prediction

Chunhua Shen

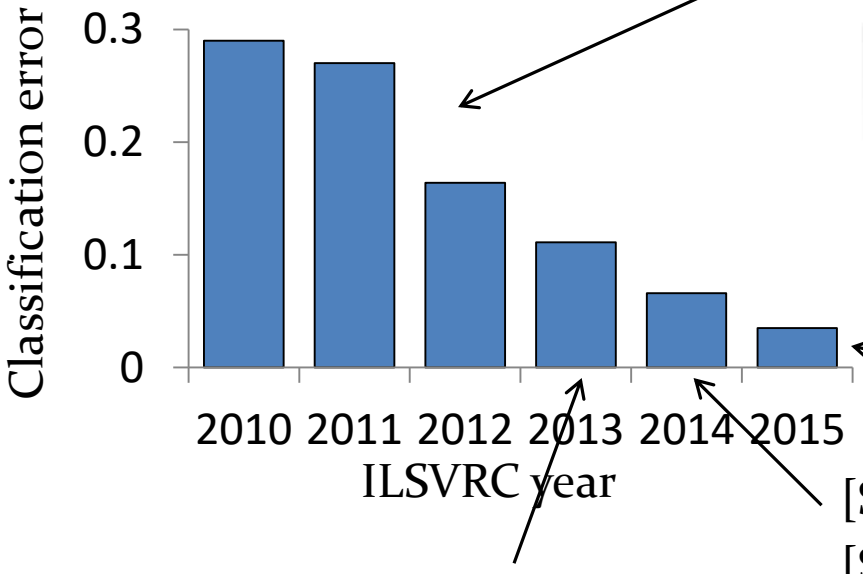
The University of Adelaide, Australia

THE UNIVERSITY of ADELAIDE

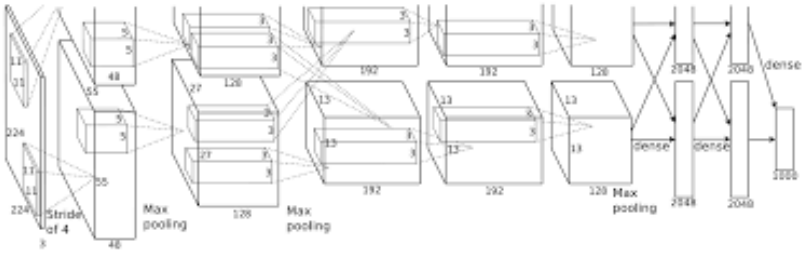
Image understanding

Convolution Neural Networks

Image Classification



[Krizhevsky et al., 2012] 16.4% error (AlexNet)



[He et al., 2015] 3.6% error (ResNet)

[Szegedt et al., 2014] 6.6% error (GoogLeNet)

[Simonyan et al., 2014] 7.3% error (VGGNet)

[Zeiler et al., 2013] 11.1% error

Google's best reported results 2016

method	depth	tr. input	top-1	top-5	speed
VGG16 [28], 10 crops	16	224	28.1	9.3	–
ResNet-50 [12], our tested	50	224	23.5	6.8	75.2
ResNet-101 [12], our tested	101	224	22.1	6.1	56.8
ResNet-152 [12], our tested	152	224	21.8	5.8	41.8
ResNet-152 [13]	152	224	21.3	5.5	–
ResNet-152 [13], pre-act.	152	224	21.1	5.5	–
ResNet-200 [13], pre-act.	200	224	20.7	5.3	–
Inception-v4 [30]	76	299	20.0	5.0	–
Inception-ResNet-v2 [30]	96	299	19.9	4.9	–
56-1-1-1-1-9-1-1, Model F	34	56	25.2	7.8	113.5
112-1-1-1-1-5-1-1, Model E	26	112	22.3	6.2	97.3
112-1-1-1-1-9-1-1, Model D	34	112	22.1	6.0	81.2
112-1-1-1-1-13-1-1, Model C	42	112	21.8	5.9	69.2
224-0-1-1-1-1-1-1	16	224	22.0	5.8	55.3
224-0-1-1-1-3-1-1, Model B	20	224	21.0	5.5	43.3
224-0-3-3-6-3-1-1, Model A	38	224	**19.2**	**4.7**	15.7

Table 1. Comparison of networks by top-1 (%) and top-5 (%) errors on the ILSVRC 2012 validation set [27] with 50k images, obtained using a single crop. Testing speeds (images/second) are evaluated with ten images/mini-batch using cuDNN 4 on a GTX 980 card. Input sizes during training are also listed. Note that a smaller size often leads to faster training speed.

"Wider or Deeper: Revisiting the ResNet Model for Visual Recognition", arXiv:1611.10080

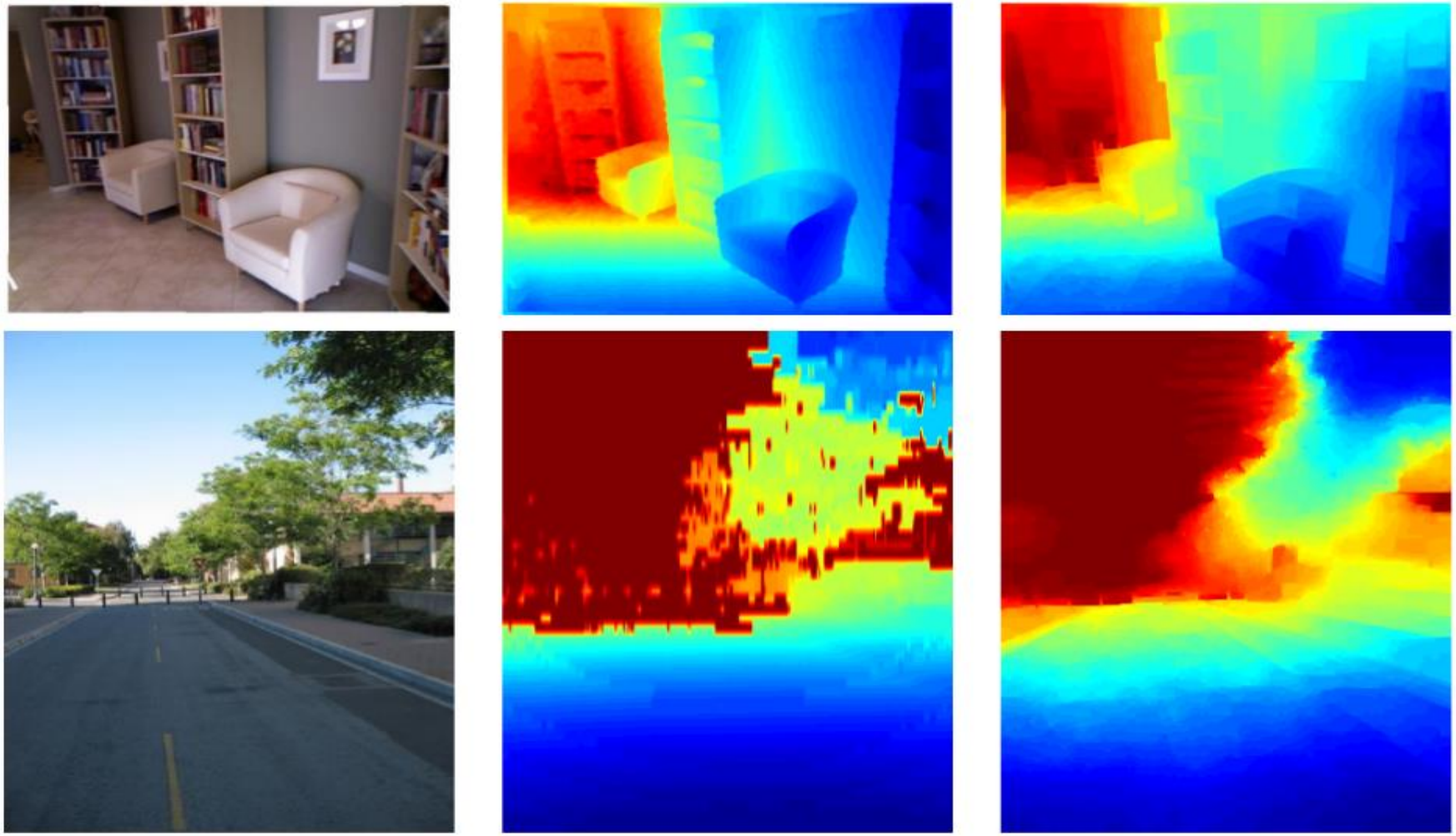
Image understanding

Image understanding

Depth estimation from single monocular images

- Depth acquisition:
  - Depth sensors, e.g., Kinect
  - Machine learning methods
- Most vision datasets are still RGB images
- Estimate depth from single RGB images
  - Ill-posed problem

Depth Estimation From Single Monocular Images



Test image | Ground-truth | Our prediction

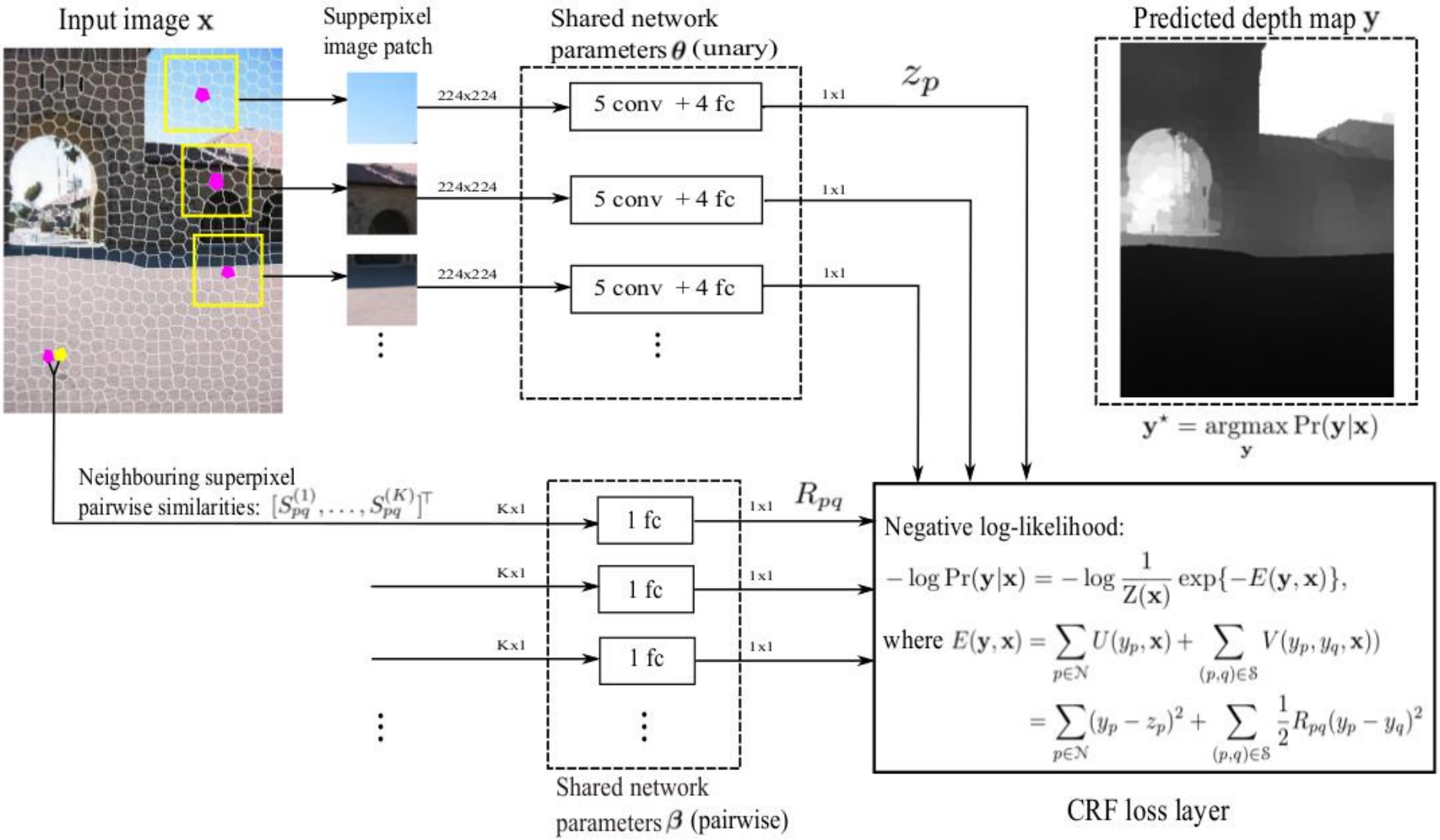
Depth Estimation From Single Monocular Images

- Useful
  - Scene understanding
  - 3D modelling
  - Benefit other vision tasks
    - e.g., semantic labellings, pose estimations
- Challenging
  - No reliable depth cues
    - e.g., stereo correspondence, motion information

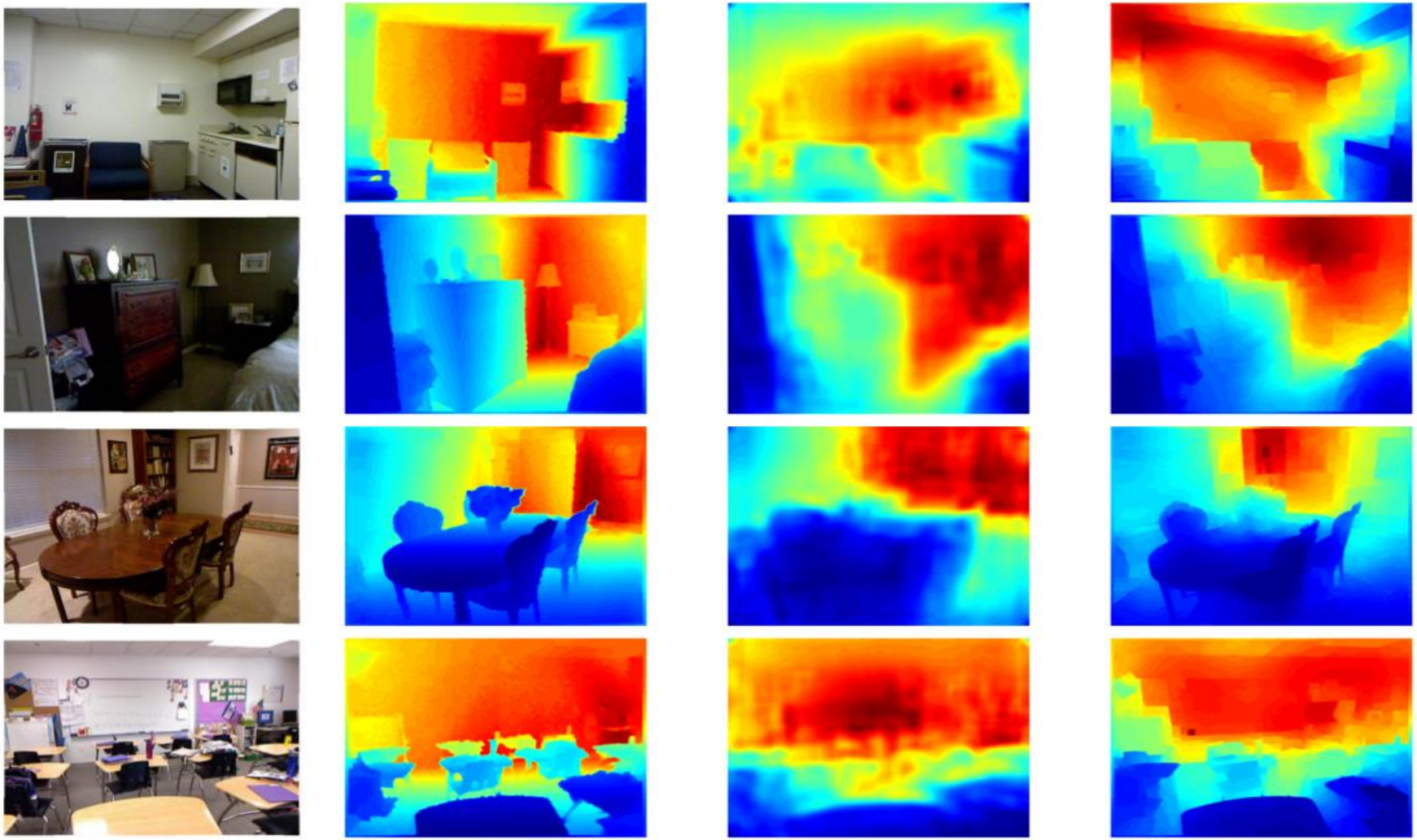
Our method

- Joint learning: Continuous CRF + deep CNN
- Exact maximization of log-likelihood
- Closed form solution for MAP inference

Deep convolutional neural fields

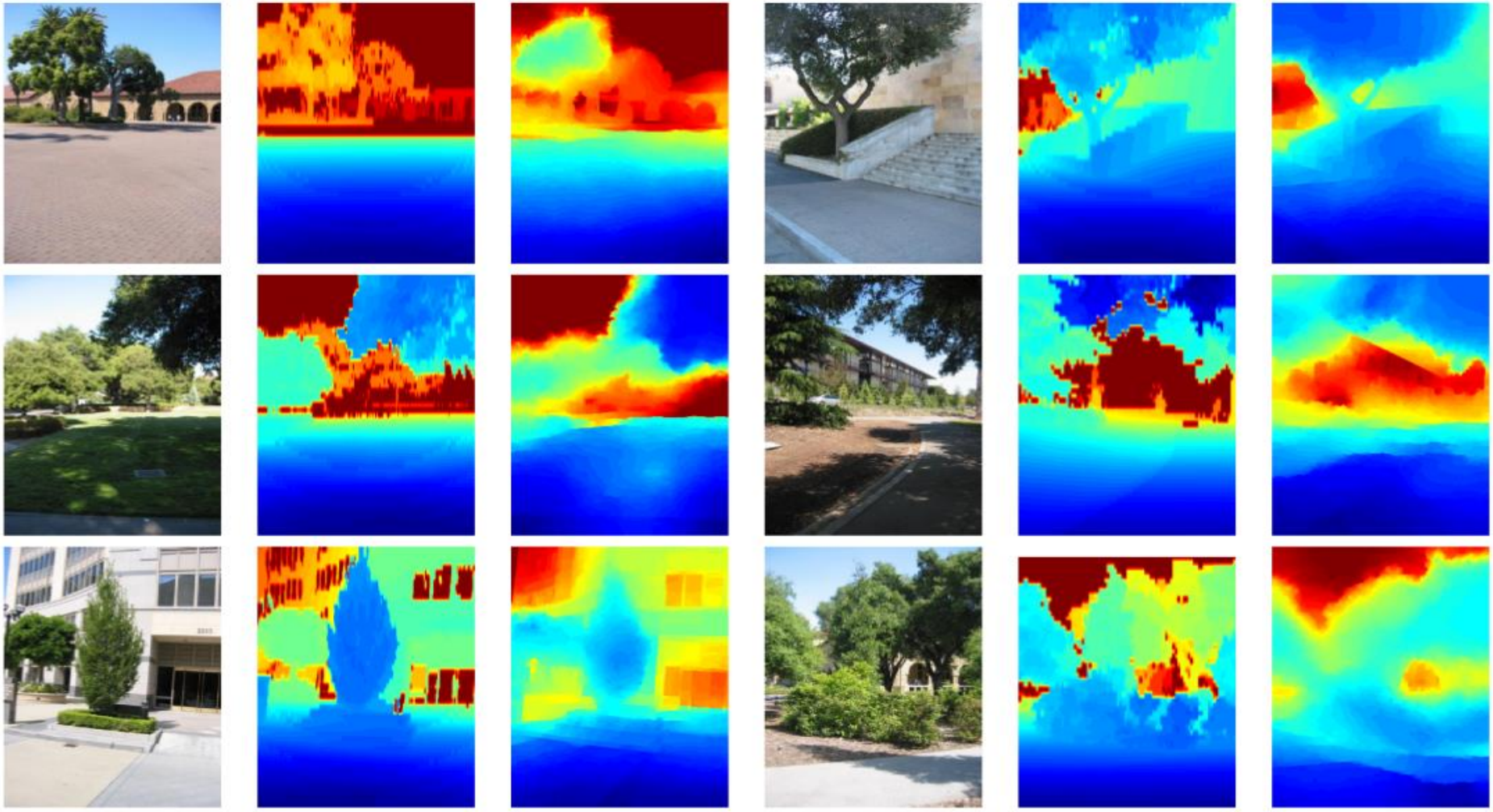


Deep convolutional neural fields



Test image | Ground-truth | Eigen etal. (NIPS2014) | Ours

Deep convolutional neural fields



Conclusion

- Deep convolutional neural fields for monocular image depth estimations
- Combine deep CNN and continuous CRF
- General learning framework

Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields
Fayao Liu, Chunhua Shen, Guosheng Lin
CVPR2015
http://arxiv.org/abs/1502.07411

Monocular Depth Estimation with Augmented Ordinal Depth Relationships

Motivation:

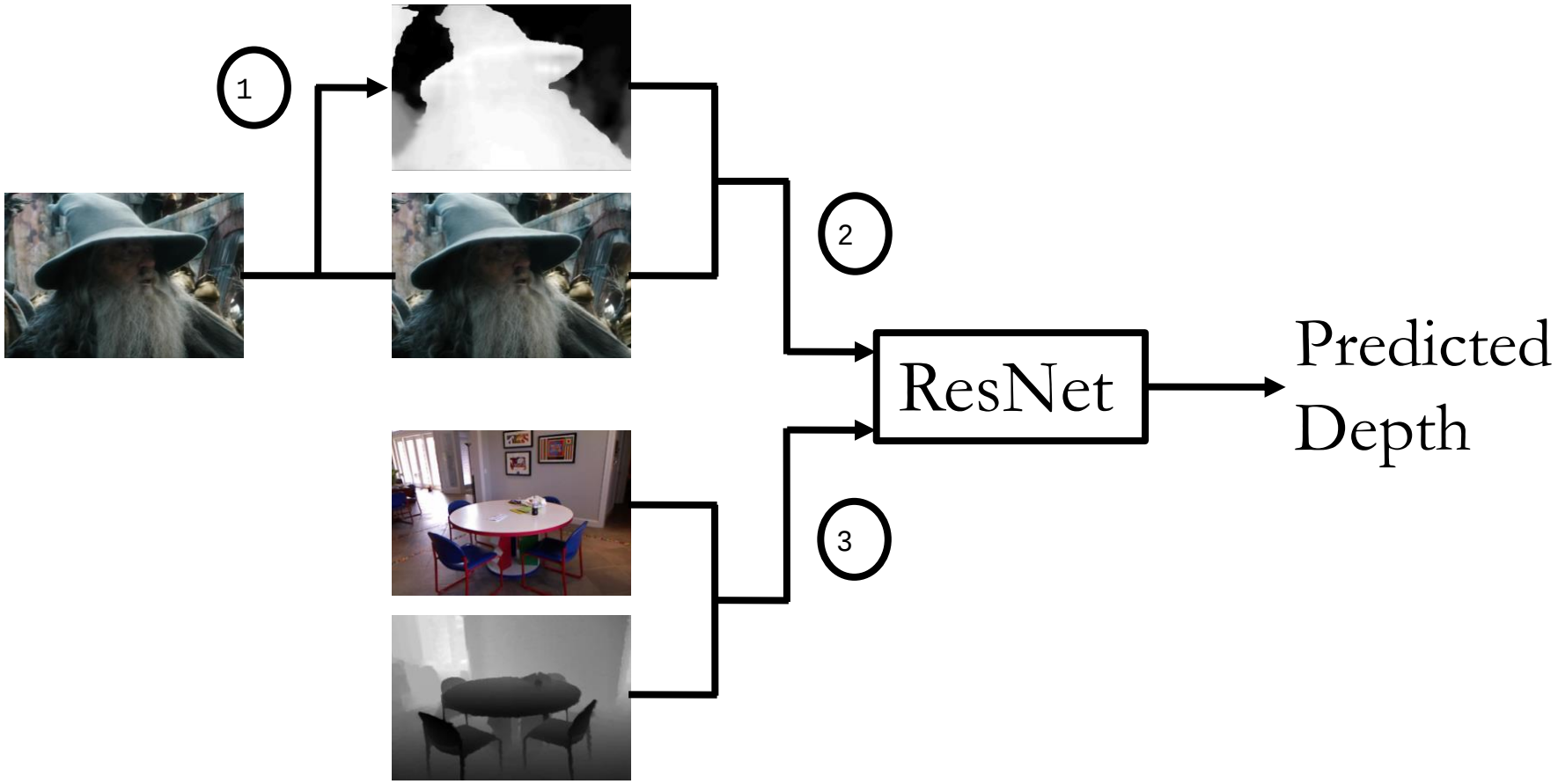
- Limited metric RGB-D data in diversity and quantity.
- Relative depth has been proven to be an informative cue.
- Relative depth can be easily acquired from vast stereo videos.

Highlights:

- A new Relative Depth in Stereo (RDIS) dataset is proposed.
- Densely labelled relative depth using existing stereo matching methods.
- State-of-the-art results on benchmark Depth Estimation datasets.

Overview

1. Acquire relative depth from stereo videos.
2. Pretrain a deep ResNet with relative depths.
3. Finetune the ResNet with metric depths.



Relative Depth Generation

1. Use the absolute difference (AD) matching cost and the semi-global matching (SGM) method to generate the initial disparity maps.

2. Post-process the disparity maps: Correct vague or missing boundaries of objects, and smooth disparities within objects and background. This is done by experienced workers from movie production companies.

Results

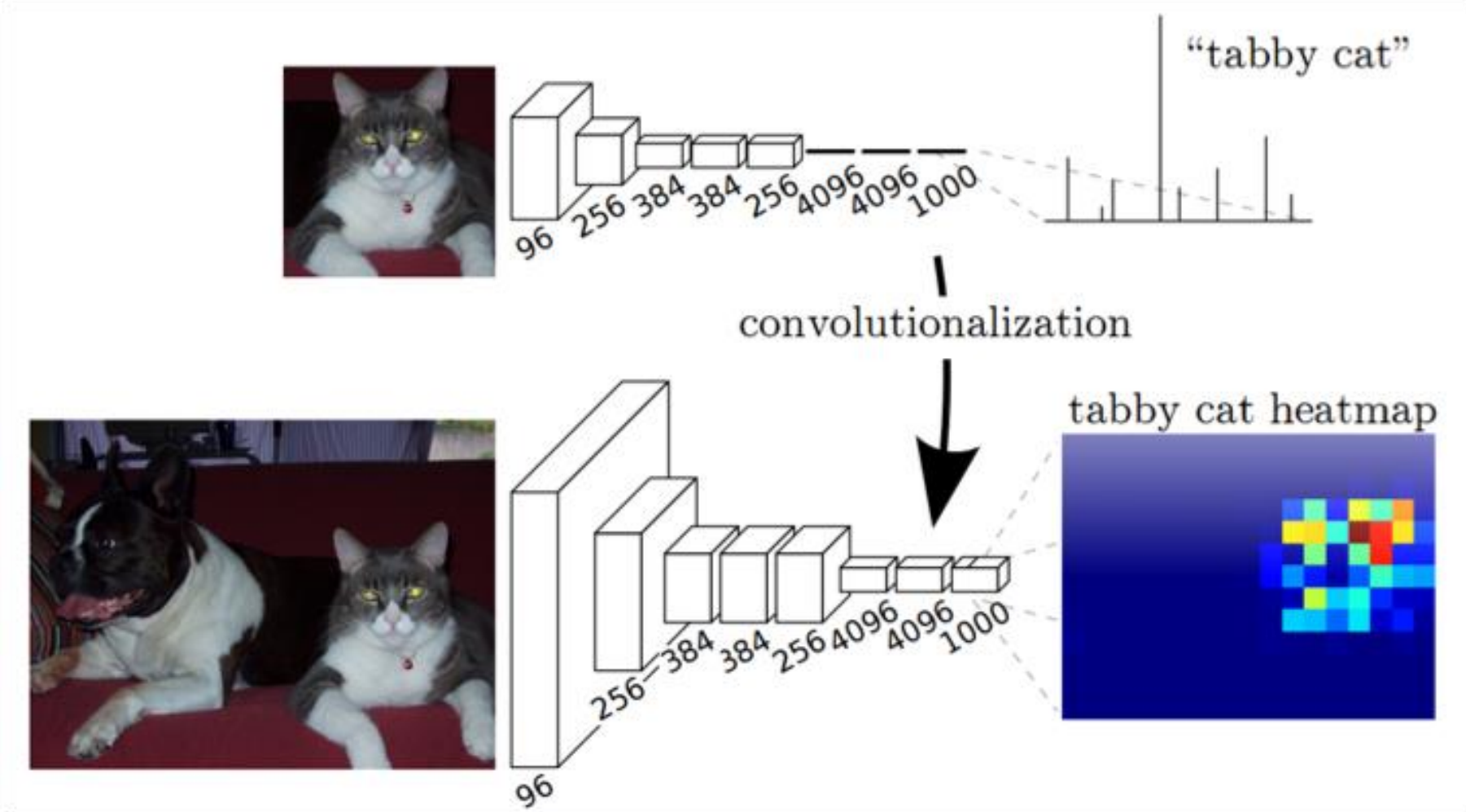
State-of-the-art results on NYUD2

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
Wang et al. [28]	60.5%	89.0%	97.0%	0.210	0.094	0.745
Liu et al. [20]	65.0%	90.6%	97.6%	0.213	0.087	0.759
Eigen et al. [5]	76.9%	95.0%	98.8%	0.158	-	0.641
Laina et al. [17]	81.1%	95.3%	98.8%	**0.127**	**0.055**	0.573
Ours	**83.1%**	**96.2%**	**98.8%**	0.132	0.057	**0.538**

State-of-the-art results on KITTI

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	rmslog	rms
Cap 80 meters						
Liu et al. [20]	65.6%	88.1%	95.8%	0.217	-	7.046
Eigen et al. [6]	69.2%	89.9%	96.7%	0.190	0.270	7.156
Godard et al. [10]	81.8%	92.9%	96.6%	0.141	0.242	5.849
Godard et al. CS [10]	83.6%	93.5%	96.8%	0.136	0.236	5.763
Ours	**89.0%**	**96.7%**	**98.4%**	**0.120**	**0.192**	**4.533**
Cap 50 meters						
Garg et al. [8]	74.0%	90.4%	96.2%	0.169	0.273	5.104
Godard et al. [10]	84.3%	94.2%	97.2%	0.123	0.221	5.061
Godard et al. CS [10]	85.8%	94.7%	97.4%	0.118	0.215	4.941
Ours	**89.7%**	**96.8%**	**98.4%**	**0.117**	**0.189**	**3.753**

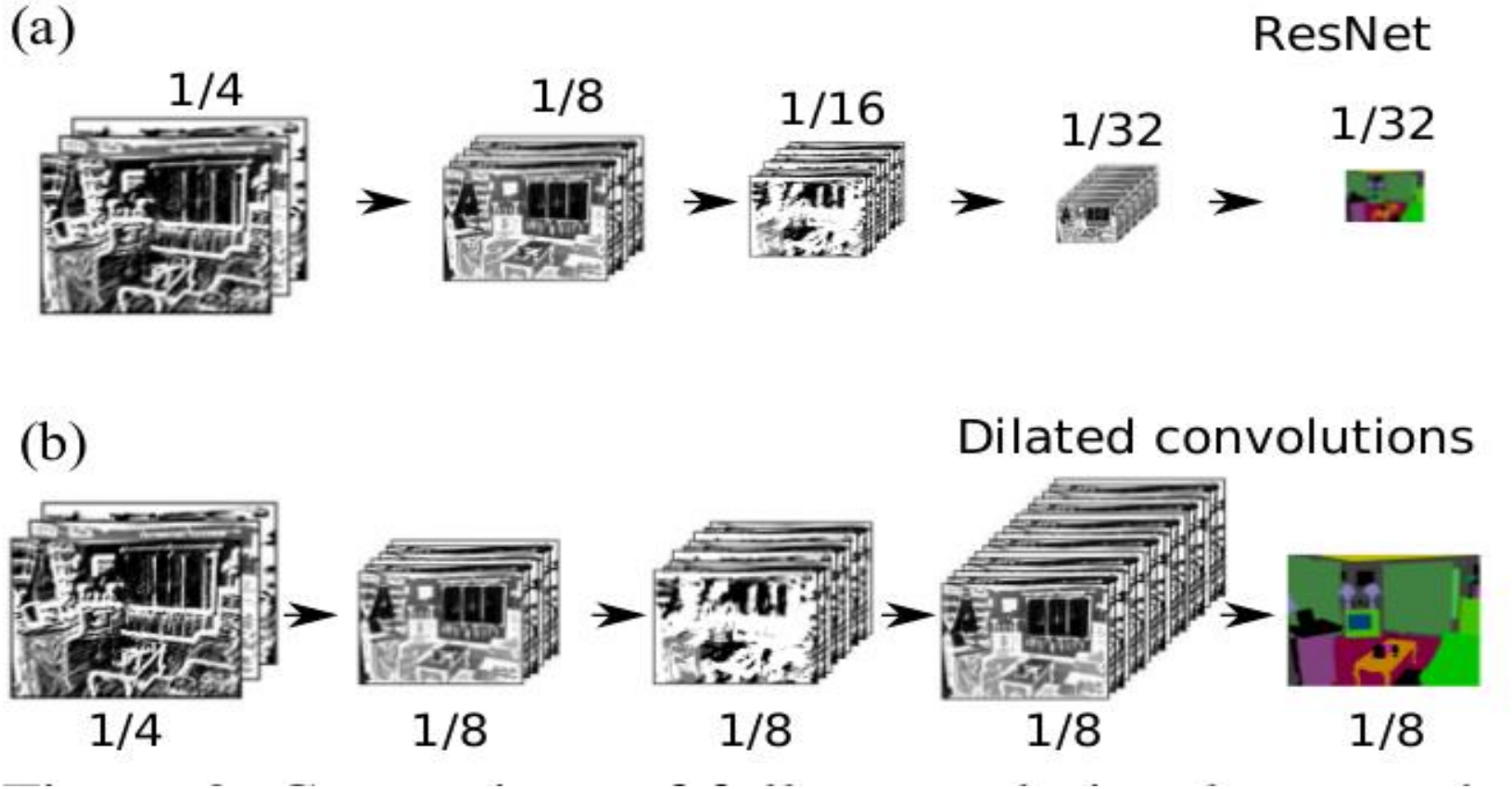
Semantic pixel labelling using FCN



RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation

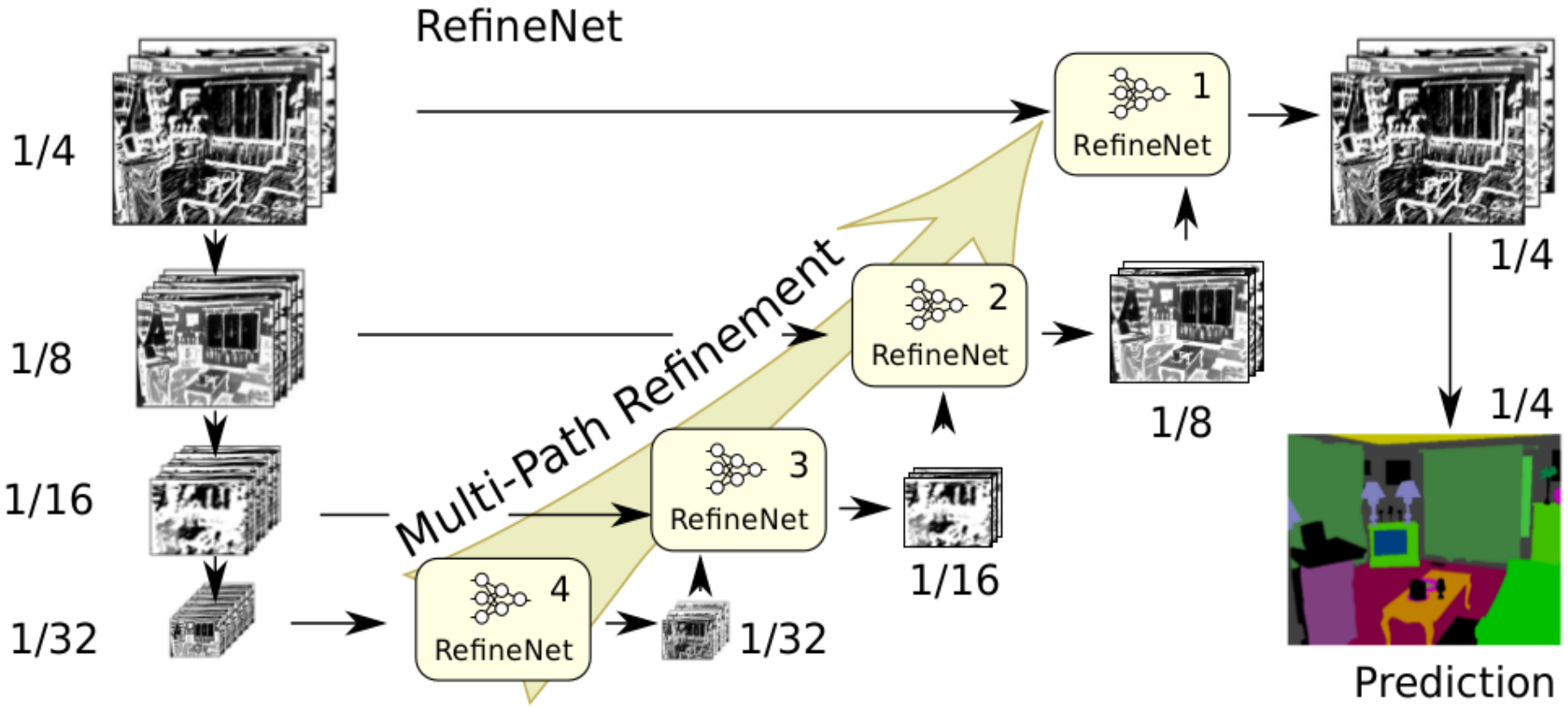
Figure 1. Example results of our method on the task of object parsing *(left)* and semantic segmentation *(right)*.

Existing approaches



1. Standard multi-layer CNNs, such as ResNet (a):
producing low-resolution (down-sampled) feature maps; fine structures/details are lost.
2. Dilated convolutions (b):
Resulting high-resolution and high-dimension feature maps;
computationally expensive and huge memory consumption if generating large resolution output.

Our approach



Exploits various levels of detail at different stages of convolutions and fuses them to obtain a high-resolution prediction without the need to maintain large intermediate feature maps

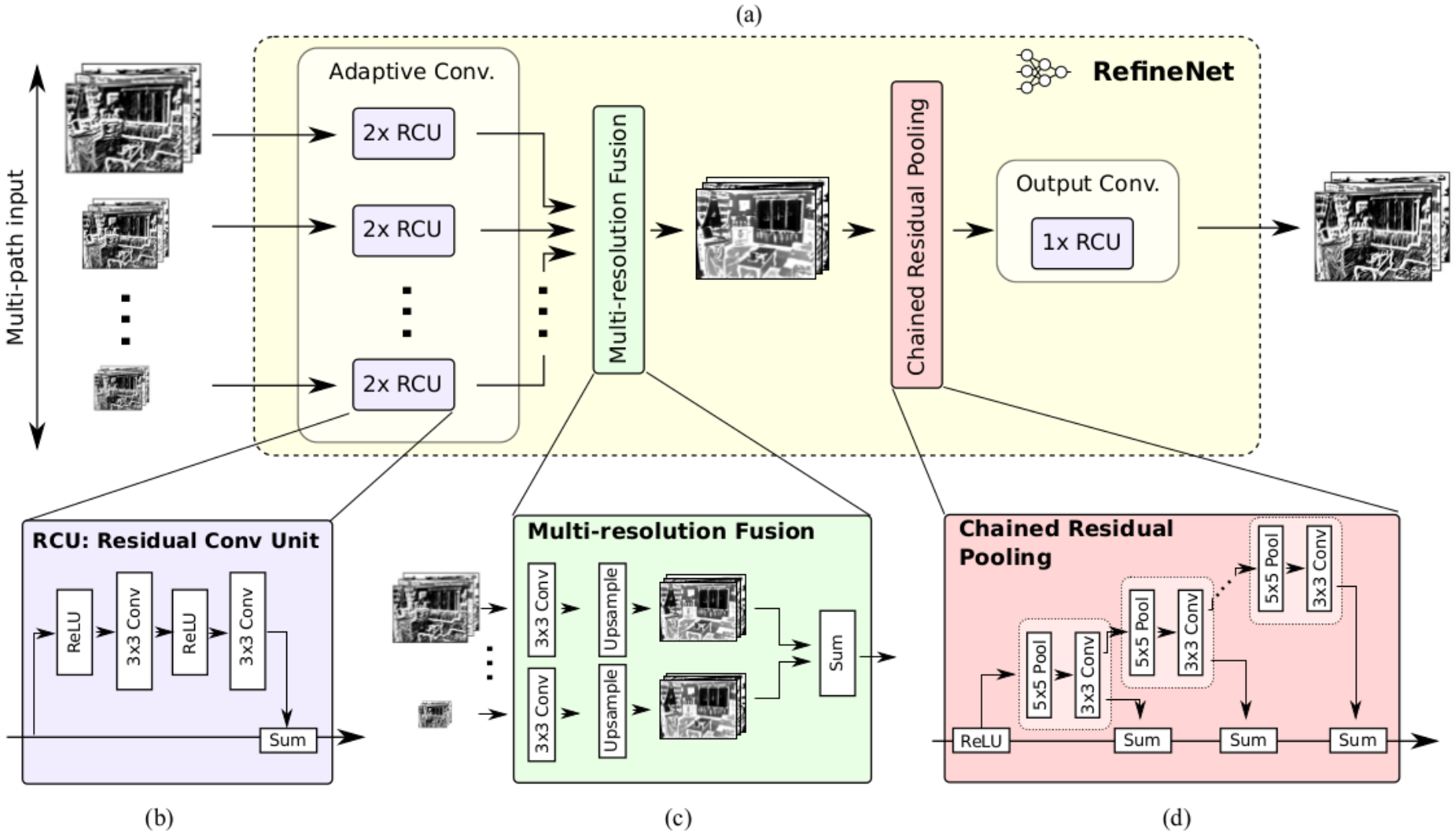


Figure 3. The individual components of our multi-path refinement network architecture RefineNet. Components in RefineNet employ residual connections with identity mappings. In this way, gradients can be directly propagated within RefineNet via local residual connections, and also directly propagate to the input paths via long-range residual connections, and thus we achieve effective end-to-end training of the whole system.

Highlights

1. Exploits features at multiple levels of abstraction for high-resolution output.

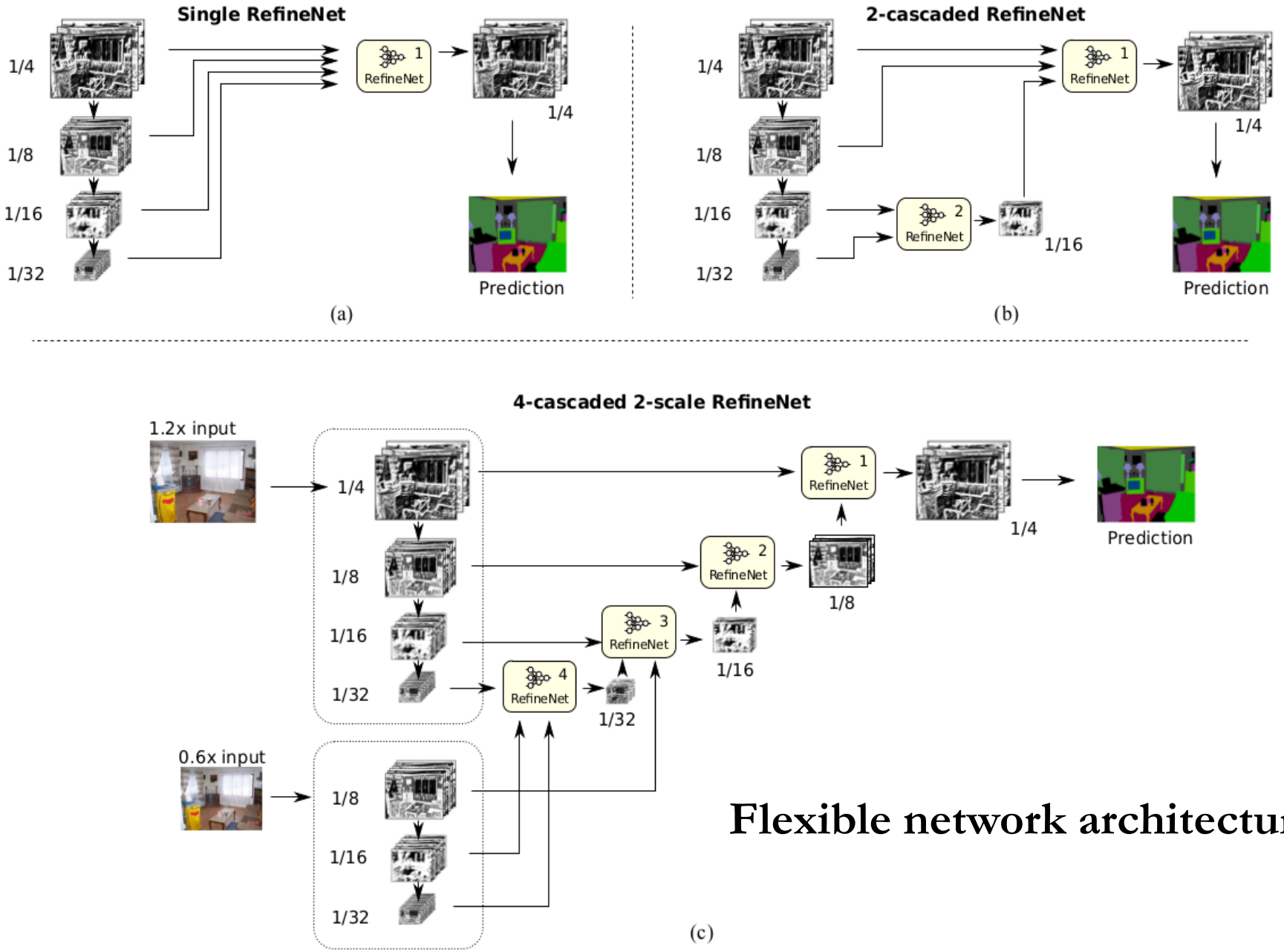
RefineNet refines low-resolution (coarse) semantic features with fine-grained low-level features in a recursive manner to generate high-resolution semantic feature maps. Our model is flexible in that it can be cascaded and modified in various ways.

2. Effective gradient propagation with identity mappings through short and long range connections

Our cascaded RefineNets can be effectively trained end-to-end, which is crucial for best prediction performance. All components in RefineNet employ residual connections with identity mappings, such that gradients can be directly propagated through short-range and long-range residual connections allowing for both effective and efficient end-to-end training.

3. Chained residual pooling

We propose a new network component we call "chained residual pooling" which is able to capture background context from a large image region. It does so by efficiently pooling features with multiple window sizes and fusing them together with residual connections and learnable weights.



Flexible network architectures

Figure 7. Illustration of 3 variants of our network architecture: (a) single RefineNet, (b) 2-cascaded RefineNet and (c) 4-cascaded RefineNet with 2-scale ResNet. Note that our proposed RefineNet block can seamlessly handle different numbers of inputs of arbitrary resolutions and dimensions without any modification.

Experiments

Our source code is available at:
https://github.com/guosheng/refinenet

Table 1. Object parsing results on the Person-Part dataset. Our method achieves the best performance (bold).

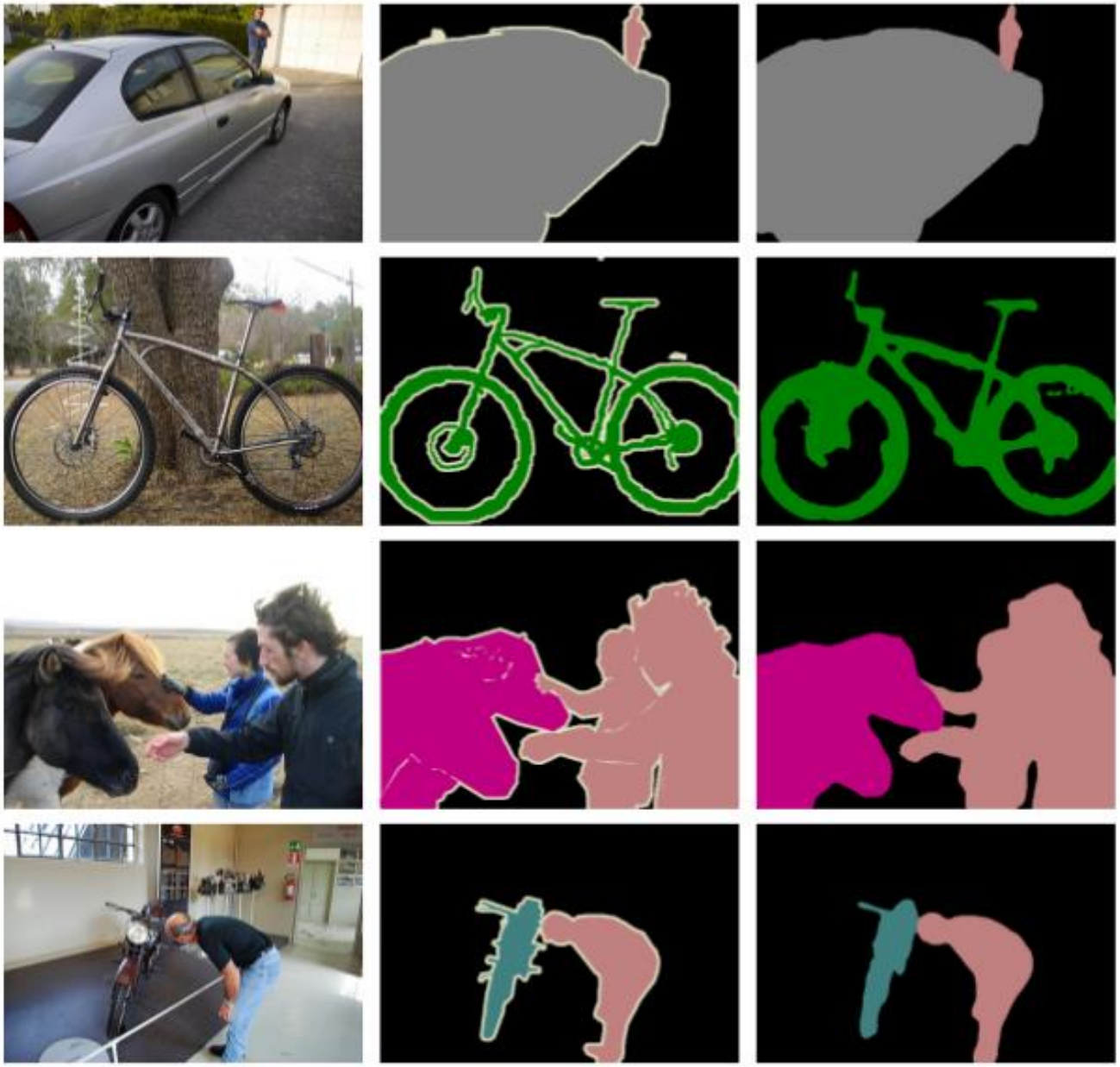
method	IoU
Attention [7]	56.4
HAZN [45]	57.5
LG-LSTM [29]	58.0
Graph-LSTM [28]	60.2
DeepLab [5]	62.8
DeepLab-v2 (Res101) [6]	64.9
RefineNet-Res101 (ours)	**68.6**

(a) Test Image **(b)** Ground Truth **(c)** Prediction

Figure 4. Our prediction examples on Person-Parts dataset.

Table 5. Results on the PASCAL VOC 2012 test set (IoU scores). Our RefineNet archives the best performance (IoU 83.4).

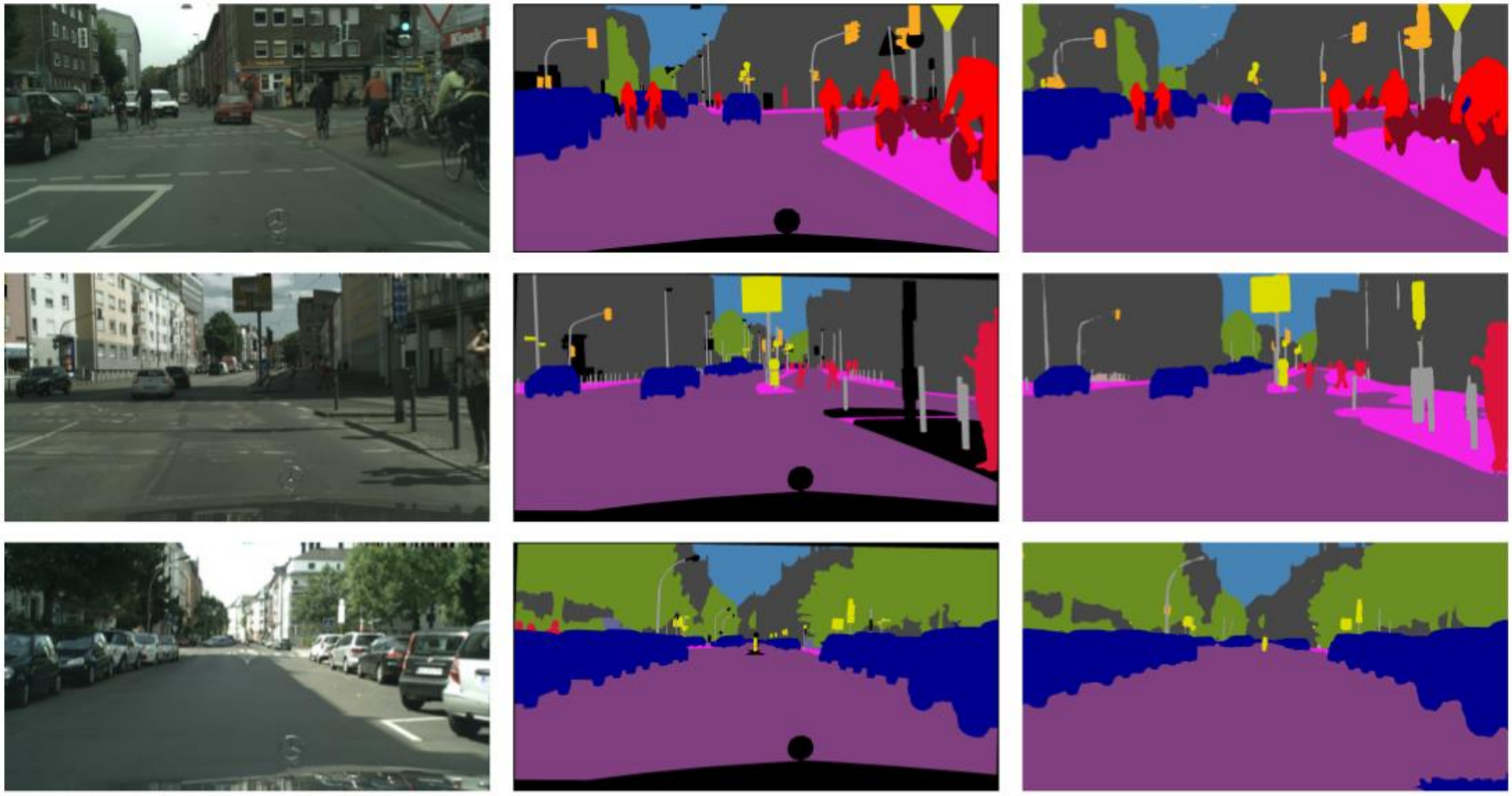
Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	potted	sheep	sofa	train	tv	**mean**
FCN-8s [36]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeconvNet [38]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
CRF-RNN [47]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
BoxSup [10]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
DPN [35]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
Context [30]	94.1	40.7	84.1	67.8	75.9	93.4	84.3	88.4	42.5	86.4	64.7	85.4	89.0	85.8	86.0	67.5	90.2	63.8	80.9	73.0	78.0
DeepLab [5]	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
DeepLab2-Res101 [6]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
CSupelec-Res101 [4]	92.9	61.2	91.0	66.3	77.7	**95.3**	**88.9**	92.4	33.8	88.4	69.1	89.8	**92.9**	87.7	87.5	62.6	89.9	59.2	87.1	74.2	80.2
RefineNet-Res101	**94.9**	60.2	92.8	**77.5**	81.5	95.0	87.4	93.3	39.6	89.3	73.0	**92.7**	92.4	85.4	**88.3**	69.7	92.2	65.3	84.2	**78.7**	82.4
RefineNet-Res152	94.7	**64.3**	**94.9**	74.9	**82.9**	95.1	88.5	**94.7**	**45.5**	**91.4**	**76.3**	90.6	91.8	**88.1**	88.0	**69.9**	**92.3**	**65.9**	**88.7**	76.8	**83.4**



(a) Test Image **(b)** Ground Truth **(c)** Prediction

Figure 5. Our prediction examples on VOC 2012 dataset.

15 FPS with 720P input on a single GPU



(a) Test Image **(b)** Ground Truth **(c)** Prediction


GiGi
BEST BURGER IN TOWN

Low-level image processing with very deep FCN

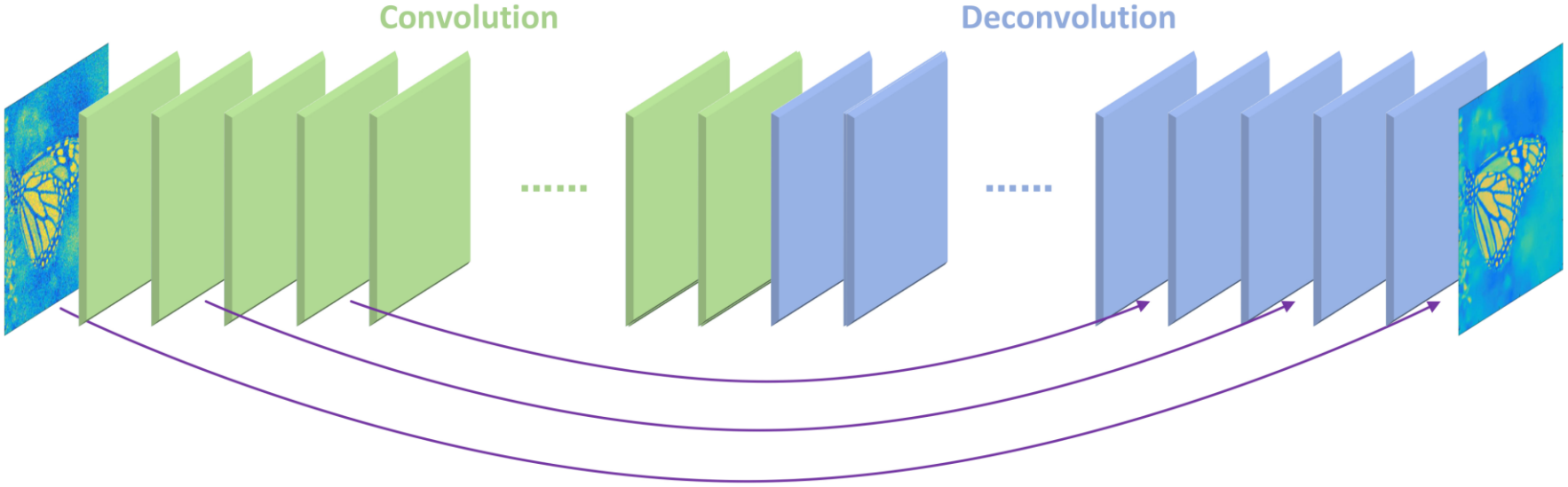


Figure 1: The overall architecture of our proposed network. The network contains layers of symmetric convolution (encoder) and deconvolution (decoder). Skip shortcuts are connected every a few (in our experiments, two) layers from convolutional feature maps to their mirrored deconvolutional feature maps. The skip connections divide the network into several blocks, the size of which is 4, i.e., 2 convolutional and deconvolutional layer respectively. The response from a convolutional layer is directly propagated to the corresponding mirrored deconvolutional layer, both forwardly and backwardly.

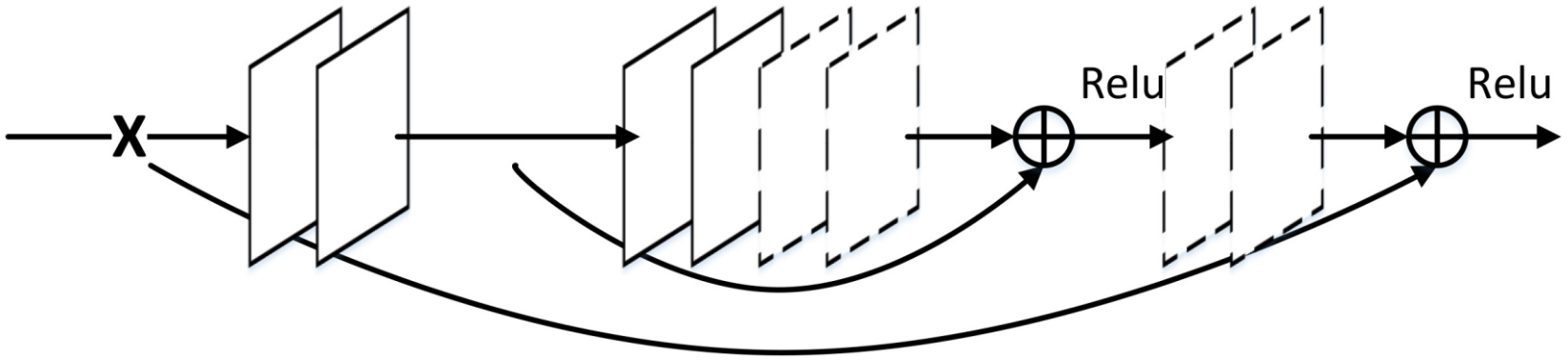


Figure 5: An example of a building block in the proposed framework. The rectangle in solid and dotted lines denote convolution and deconvolution respectively. $\oplus$ denotes element-wise sum of feature maps.

Table 1: Configurations of the 20 and 30 layer networks. "conv3" and "deconv3" stand for convolution and deconvolution kernels of size 3×3. 128, 256 and 512 is the number of feature maps after each convolution and deconvolution. "c" is the number of channels of input and output image. In this work, we test on gray-scale images, i.e., $c = 1$. However, it is straightforward to apply to color images.

RED-Net20	RED-Net30
(conv3-128)×4	(conv3-128)×6
(conv3-256)×3	(conv3-256)×6
(conv3-512)×3	(conv3-512)×3
(deconv3-512)×2	(deconv3-512)×2
(deconv3-256)×3	(deconv3-512)×6
(deconv3-128)×4	(deconv3-512)×6
(deconv3-c)	(deconv3-c)

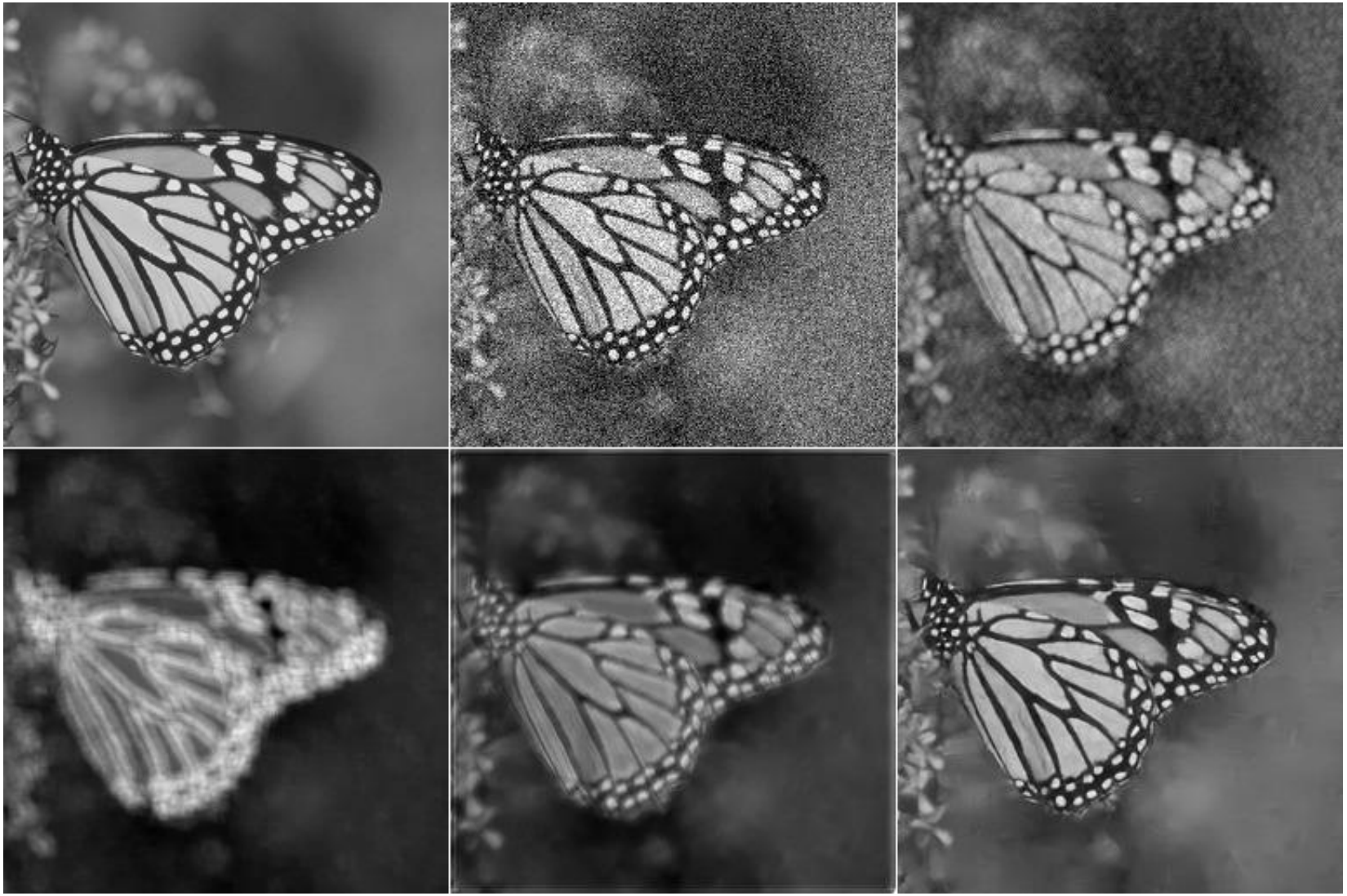


Figure 4: Visualization of the 10-layer convolutional and deconvolutional network. The images from top-left to bottom-right are: clean image, noisy image, output of conv-2, output of conv-5, output of deconv-3 and output of deconv-5, where "conv-i" and "deconv-i" stand for the i-th convolutional and deconvolutional layer respectively.

Denoise


SMITH
21
SMITH


SIX SHOOTER

Super-resolution


RESPECTFUL VISITORS
WELCOME
NO
STUDENTS
ALLOWED

Deblur

Enhancing JPEG images

Inpainting

Inpainting

. Superior results on Denoising, & Super resolution

. Many other low-level image processing tasks:

. Deblur

. Dehaze

Image Restoration Using Very Deep Fully Convolutional Encoder-Decoder Networks with Symmetric Skip Connections, X. Mao, C. Shen, Y. Yang, NIPS 2016.

Thanks.

Questions?