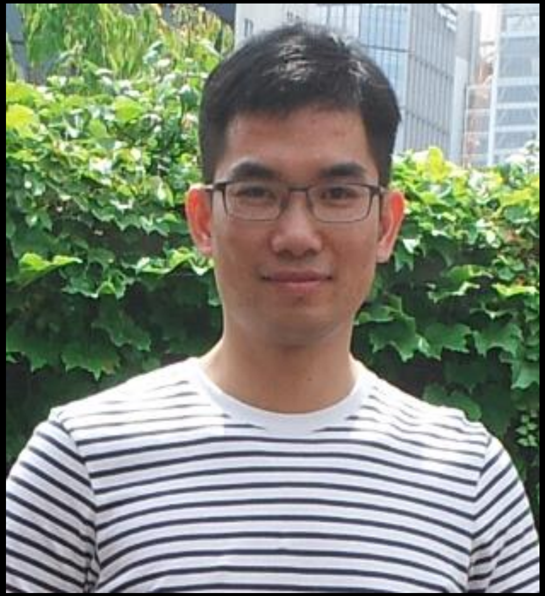
Recurrent Neural Networks for Scene Labelling and Saliency Detection

Wang Gang

Nanyang Technological University

Main PhD Collaborators

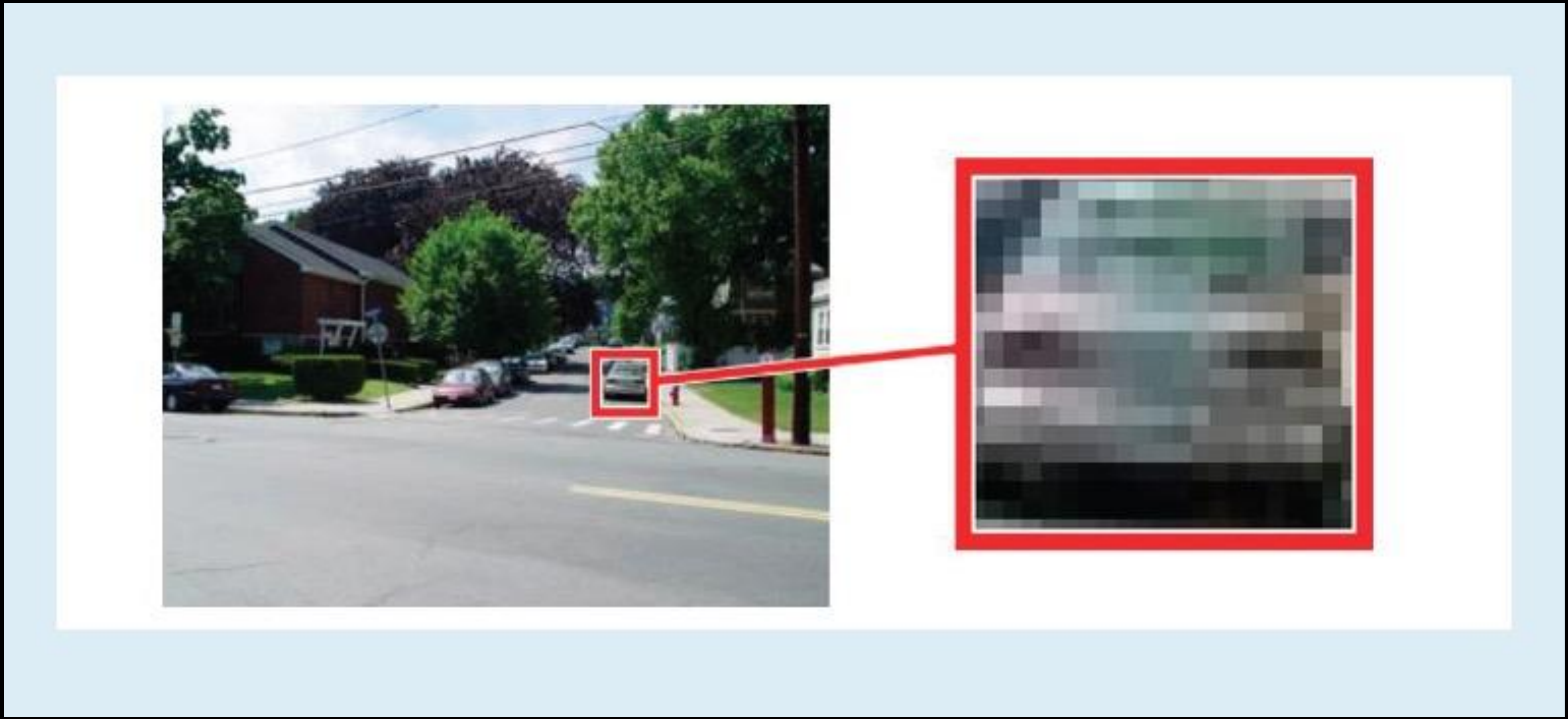


Shuai Bing

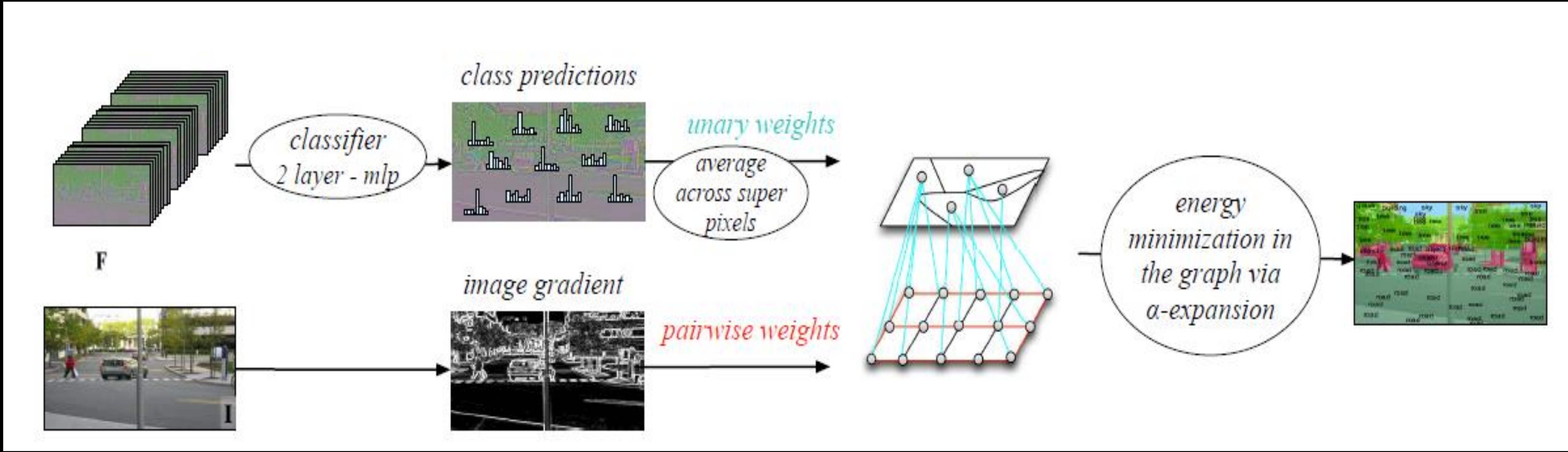
Zuo Zhen

Jason Kuen

Context is Important in Computer Vision

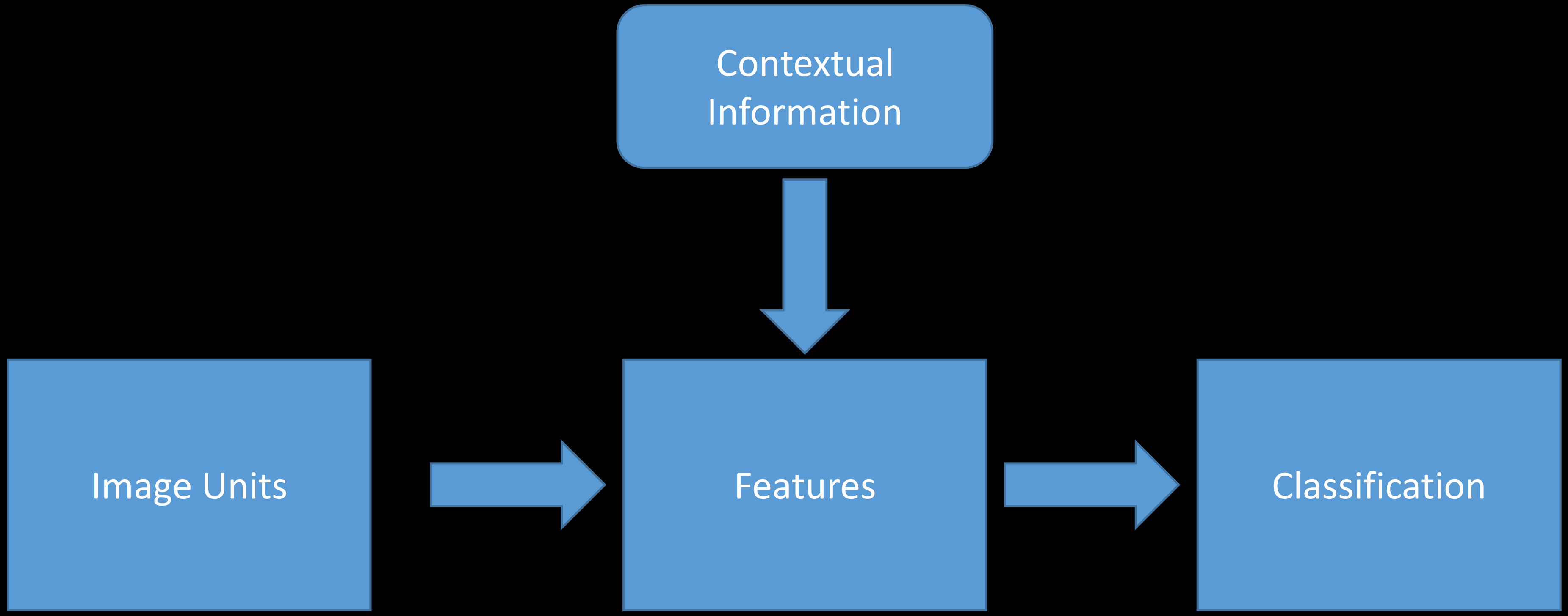


Existing Methods: Graphical Models

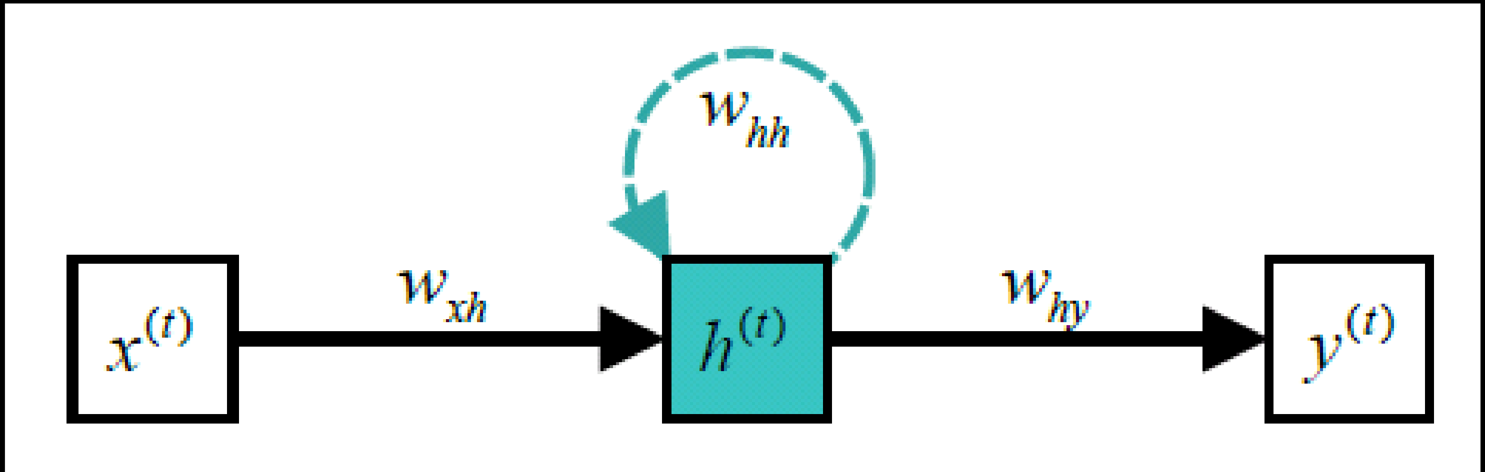


- Farabet et al. "Learning hierarchical features for scene labeling." *TPAMI 2013.*
- Zheng, Shuai, et al. "Conditional random fields as recurrent neural networks.“ *Proceedings of the IEEE International Conference on Computer Vision*. 2015.
- Liu, Ziwei, et al. "Semantic image segmentation via deep parsing network.“ *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

Our Method: Learning Local Features to Encode Contextual Information



Recurrent Neural Networks for Dependency Modelling



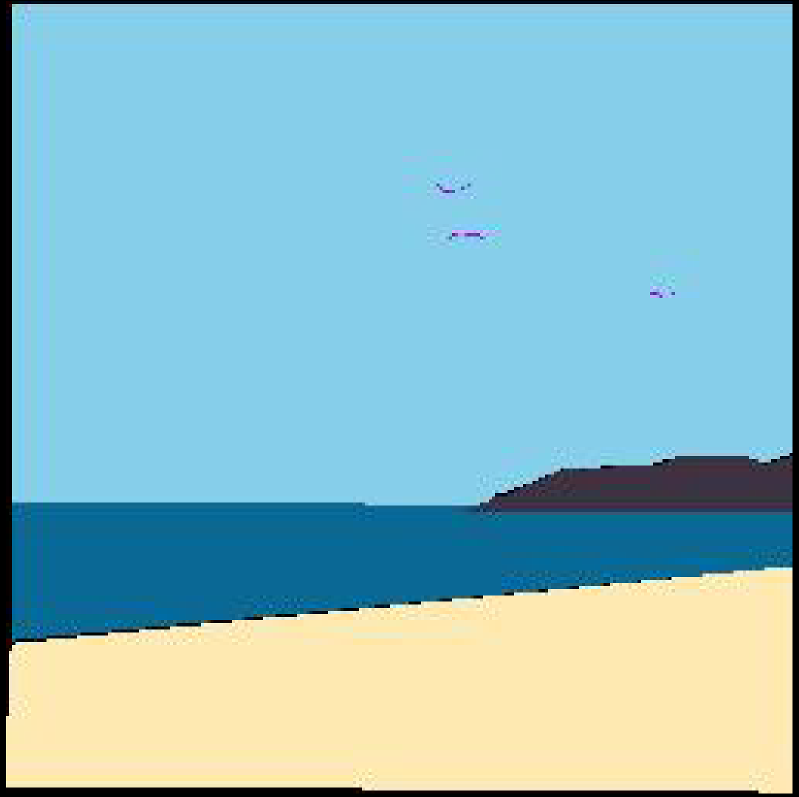
$$h^{(t)} = \psi_h \left(W_{hh} h^{(t-1)} + W_{xh} x^{(t)} + b_h \right)$$

$$y^{(t)} = \psi_y \left(W_{hy} h^{(t)} + b_y \right)$$

Advantages of Recurrent Neural Networks

- Have internal memory to remember the past
- Can model the temporal dynamic behavior of data
- Can deal with arbitrary length of data
- Fast inference speed

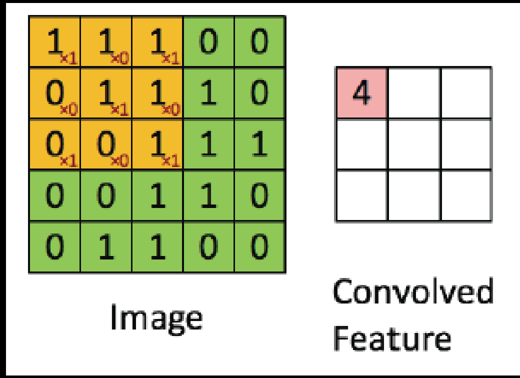
DAG-RNNs for Scene Labelling

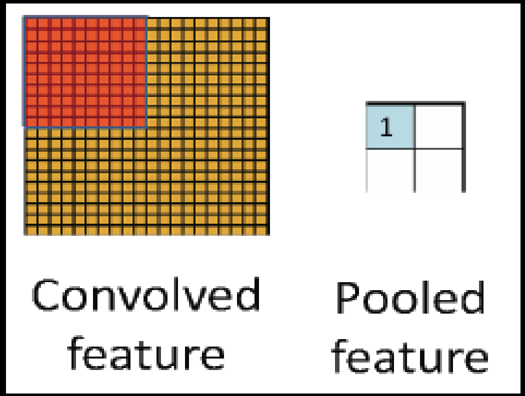


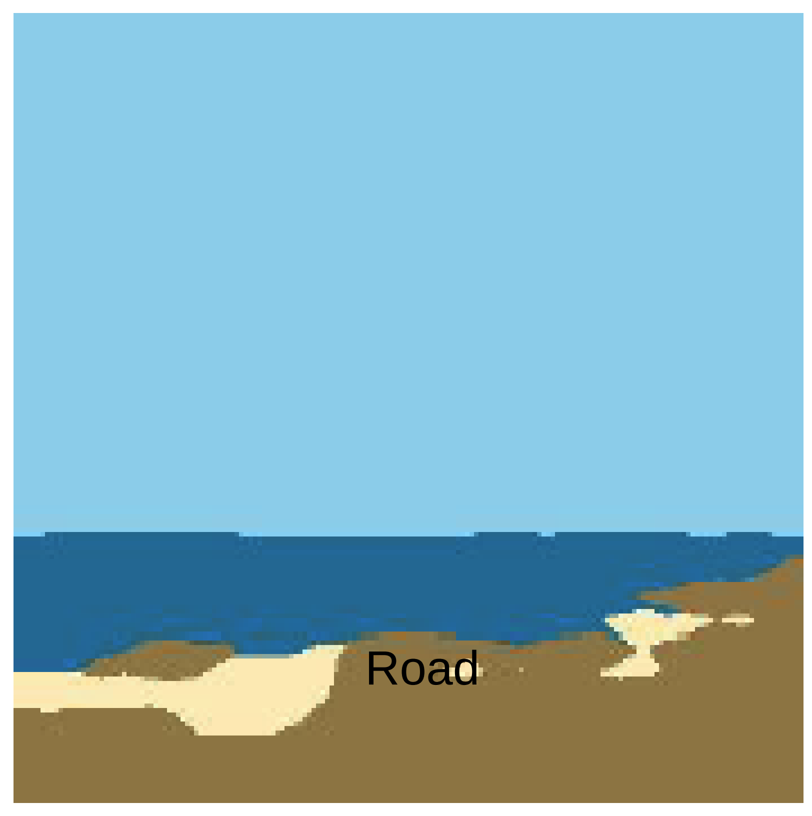
Bing Shuai, Zhen Zuo, Gang Wang and Bing Wang, "DAG-Recurrent Neural Networks For Scene Labeling," CVPR, 2016.

Motivation

Both convolution and pooling are
locally performed in CNN







Learning Extra Recurrent Layers to Encode Contextual Information

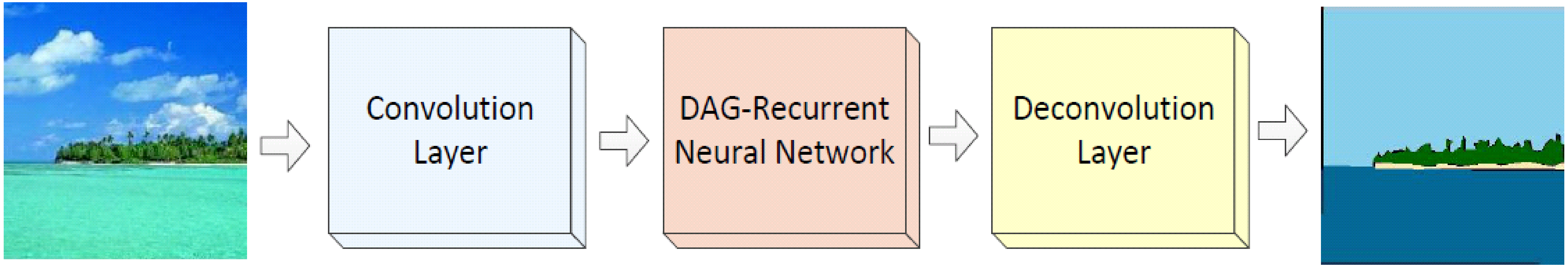
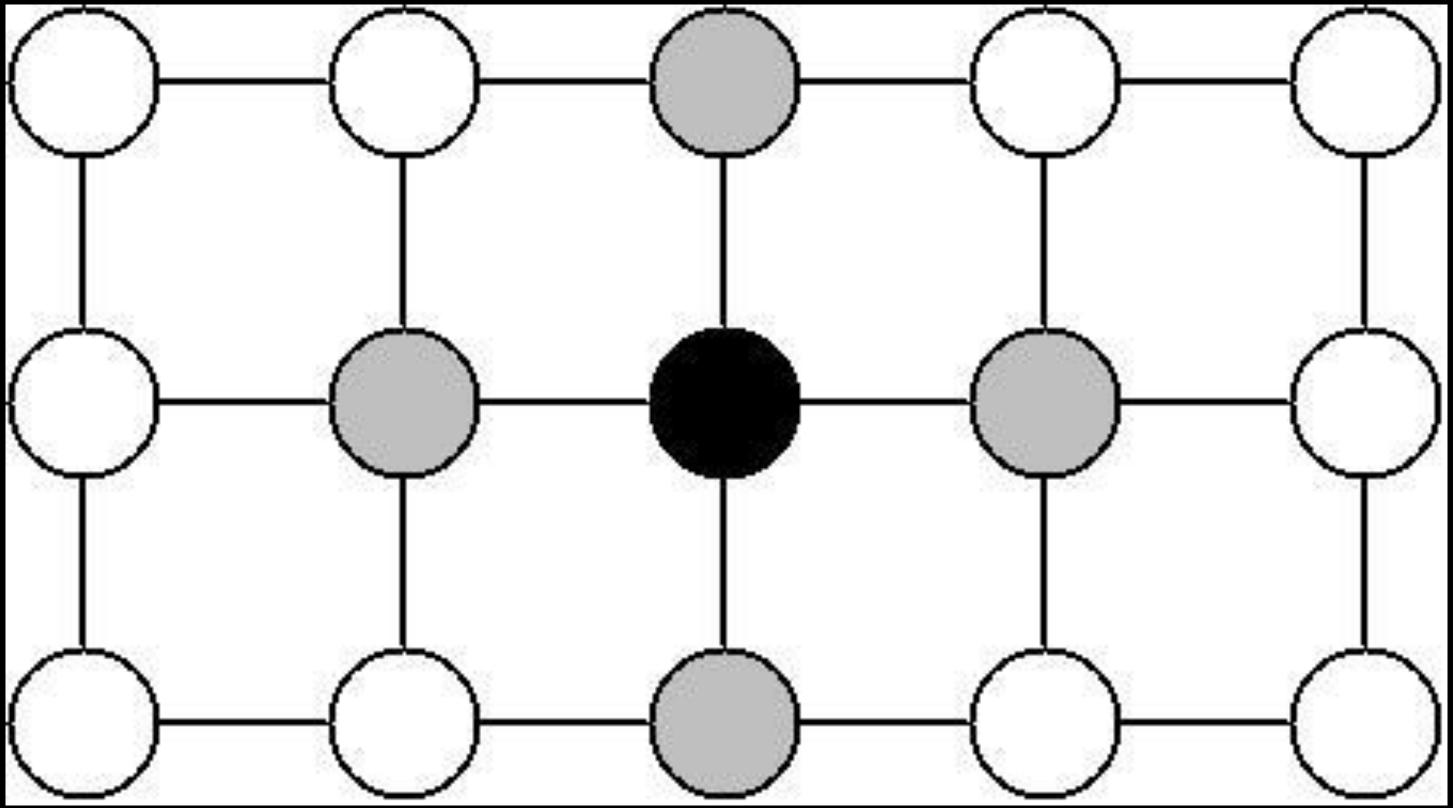


Figure 3: The architecture of the full labeling network, which consists of three functional layers: (1), convolution layer: it produces discriminative feature maps; (2), DAG-RNN: it models the contextual dependency among elements in the feature maps; (3), deconvolution layer: it upsamples the feature maps to output the desired sizes of label prediction maps.

Generalization RNNs to 2-D images is Non-trivial

- Traditional RNN can only deal with 1-D sequence
- Images are naturally 2-D undirected graphs

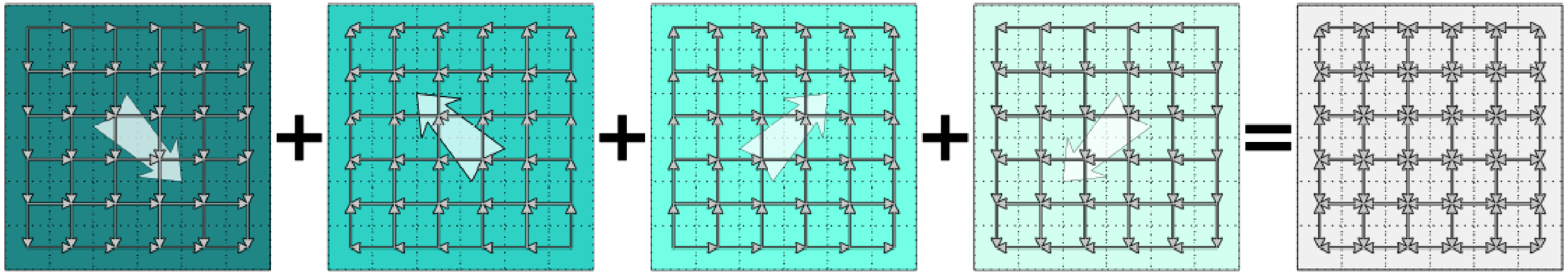


DAG-RNN

- Approximate UCG as a set of DAGs
- Develop DAG-RNNs for each DAG
- Aggregate the hidden features of different DAG-RNNs to feed the output layer

$$\mathbf{h}_d^{(v_i)} = f(U_d \mathbf{x}^{(v_i)} + \sum_{v_j \in \mathcal{P}_{\mathcal{G}_d}(v_i)} W_d \mathbf{h}_d^{(v_j)} + b_d)$$

$$\mathbf{o}^{(v_i)} = g(\sum_{\mathcal{G}_d \in \mathcal{G}^{\mathcal{U}}} V_d \mathbf{h}_d^{(v_i)} + c)$$



(a) (b) (c) (d) (e)

$$h_{\searrow}^{(r,c)} = \psi_h\Big(W_{hh\searrow}^{r} h_{\searrow}^{(r-1,c)} + W_{hh\searrow}^{c} h_{\searrow}^{(r,c-1)} + W_{xh\searrow} x^{(r,c)} + b_{h\searrow}\Big)$$

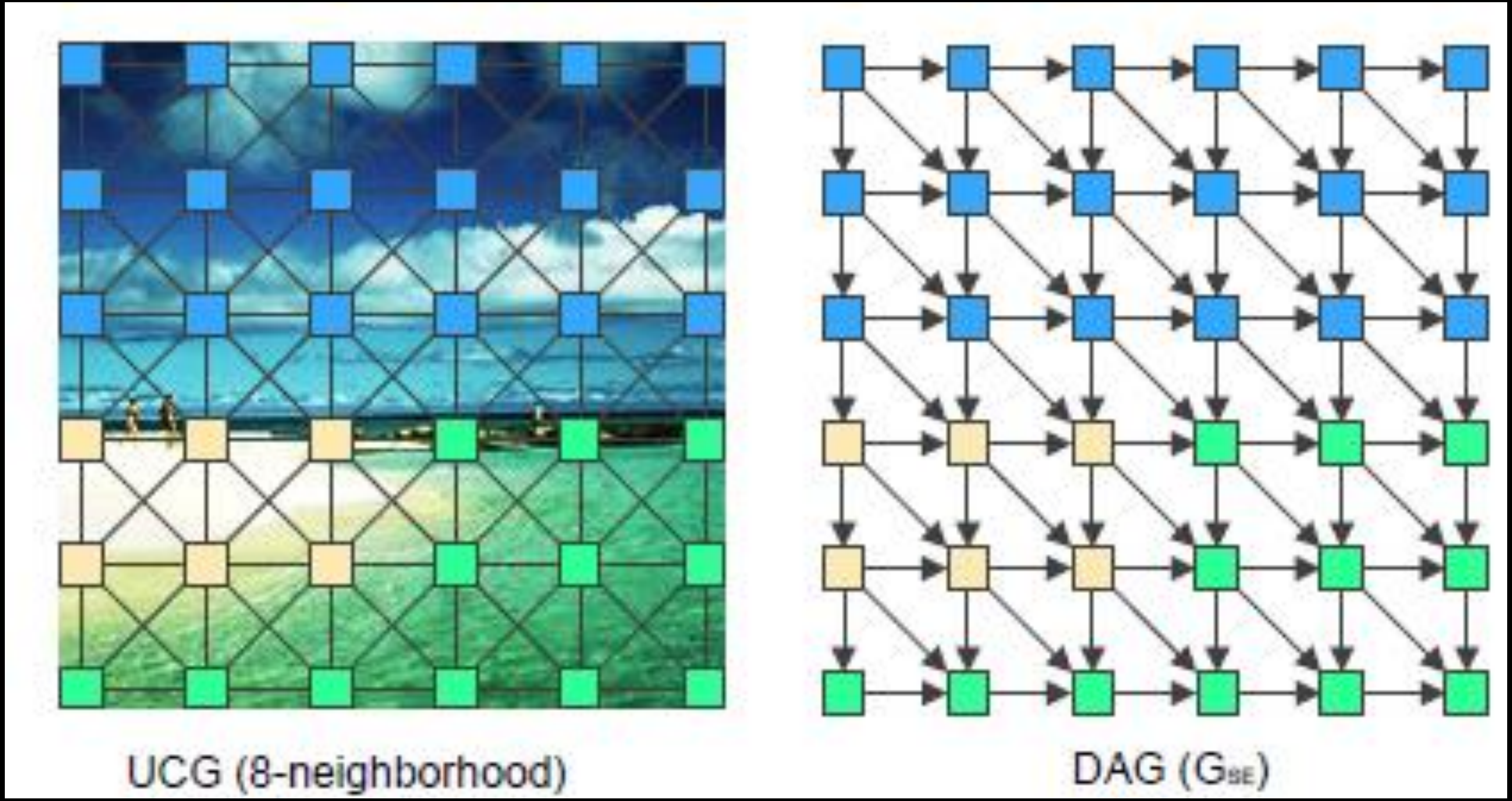
$$h_{\nwarrow}^{(r,c)} = \psi_h\Big(W_{hh\nwarrow}^{r} h_{\nwarrow}^{(r+1,c)} + W_{hh\nwarrow}^{c} h_{\nwarrow}^{(r,c+1)} + W_{xh\nwarrow} x^{(r,c)} + b_{h\nwarrow}\Big)$$

$$h_{\nearrow}^{(r,c)} = \psi_h\Big(W_{hh\nearrow}^{r} h_{\nearrow}^{(r+1,c)} + W_{hh\nearrow}^{c} h_{\nearrow}^{(r,c-1)} + W_{xh\nearrow} x^{(r,c)} + b_{h\nearrow}\Big)$$

$$h_{\swarrow}^{(r,c)} = \psi_h\Big(W_{hh\swarrow}^{r} h_{\swarrow}^{(r-1,c)} + W_{hh\swarrow}^{c} h_{\swarrow}^{(r,c+1)} + W_{xh\swarrow} x^{(r,c)} + b_{h\swarrow}\Big)$$

$$h^{(r,c)} = h_{\searrow}^{(r,c)} + h_{\nwarrow}^{(r,c)} + h_{\nearrow}^{(r,c)} + h_{\swarrow}^{(r,c)}$$

Modeling More Complicated Contextual Structures



Implementation details

- CNN features are extracted from each local patch; then RNN layers are further learned based on CNN features; CNNs and RNNs are trained in an end-to-end fashion
- Backpropagation through time is used for RNN training
- 4 and 8 neighborhood systems are used respectively
- Tested on 3 popular benchmark scene datasets, and achieved state-of-the-art performance on all of them

Methods	Global	Class
Byeon *et al.* [4]	70.1%	22.6%
Liu *et al.*[13]	74.8%	N/A
Farabet *et al.* [7]	78.5%	29.4%
Pinheiro *et al.* [17]	77.7%	29.8%
Tighe *et al.*[27]	79.2%	39.2%
Sharma *et al.*[19]	79.6%	33.6%
Shuai *et al.*[21]	**80.1%**	39.7%
Yang *et al.*[31]	79.8%	**48.7%**
CNN-65	76.1%	32.5%
CNN-65-ENN	76.1%	37.0%
CNN-65-DAG-RNN(4)	80.5%	42.6%
CNN-65-DAG-RNN(8)	**81.1%**	**48.2%**
Long *et al.*[14]	85.2%	51.7%
VGG-conv5-ENN	84.0%	48.8%
VGG-conv5-DAG-RNN(8)	**85.3%**	**55.7%**

Table 1: Quantitative performance of our method on the siftFlow dataset. The numbers (in brackets) following the DAG-RNN denote the neighborhood system of the UCG.

The SiftFlow dataset has 2688 images generally captured from 8 typical outdoor scenes. Every image has 256*256 pixels, which belong to one of the 33 semantic classes.

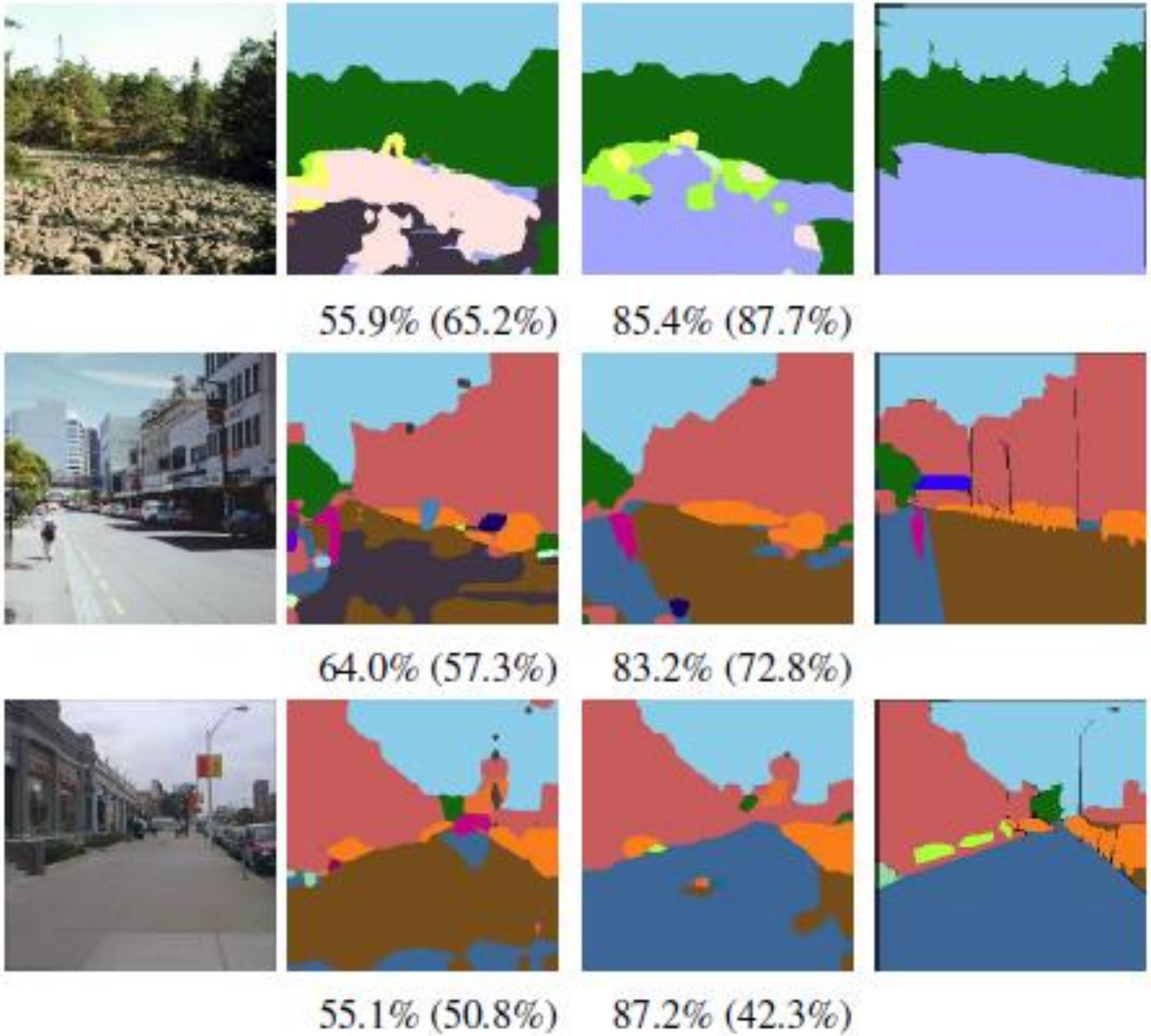
Methods	Global	Class
Tighe[30]	78.6%	43.8%
Sturgess[28]	83.8%	59.2%
Zhang[39]	82.1%	55.4%
Bulo[3]	82.1%	56.1%
Ladicky[16]	83.8%	**62.5%**
Tighe[31]	**83.9%**	**62.5%**
baseline-CNN	84.3%	53.2%
baseline-ENN	84.4%	56.7%
DAG-RNN (4)	88.6%	63.2%
DAG-RNN (8)	**89.4%**	**66.1%**
DAG-RNN$_3$ (8)	**89.8%**	**66.3%**

Table 1: Performance of our method on the CamVid dataset. The numbers (in brackets) following the DAG-RNN denote the neighborhood system of the UCG. The subscript 3 in the last line indicates that the results are obtained by averaging over 3 different DAG-RNN (8) models.

- The CamVid dataset contains 701 high-resolution images (960*720 pixels) from 4 driving videos
- Images are densely labeled with 32 semantic classes.
- We follow the usual split protocol (468/233) to obtain training/ testing images.

	Frequency	CNN	DAG-RNN(8)
bicyclist	0.76	17.0	**49.5** (+32.5)
building	24.7	89.7	**91.2** (+1.50)
car	4.3	81.5	**90.1** (+8.60)
pole	0.96	16.2	**30.7** (+14.5)
fence	1.77	28.5	**41.6** (+13.1)
pedestrian	0.75	33.5	**63.3** (+29.8)
road	32.0	96.1	**97.2** (+1.10)
sidewalk	5.52	49.1	**76.1** (+27.1)
sign	0.07	0.23	**7.35** (+7.12)
sky	17.2	**95.4**	94.7 (-0.70)
tree	12.1	78.3	**85.6** (+7.30)
global	-	84.3	**89.4** (+5.10)
class	-	53.2	**66.1** (+12.9)

Table 4: Per-class accuracy comparison on the CamVid dataset. All the numbers are scaled to the percentage range. The statistics for class frequency is obtained in test images.


55.9% (65.2%)
85.4% (87.7%)
64.0% (57.3%)
83.2% (72.8%)
55.1% (50.8%)
87.2% (42.3%)

Code will be released soon!

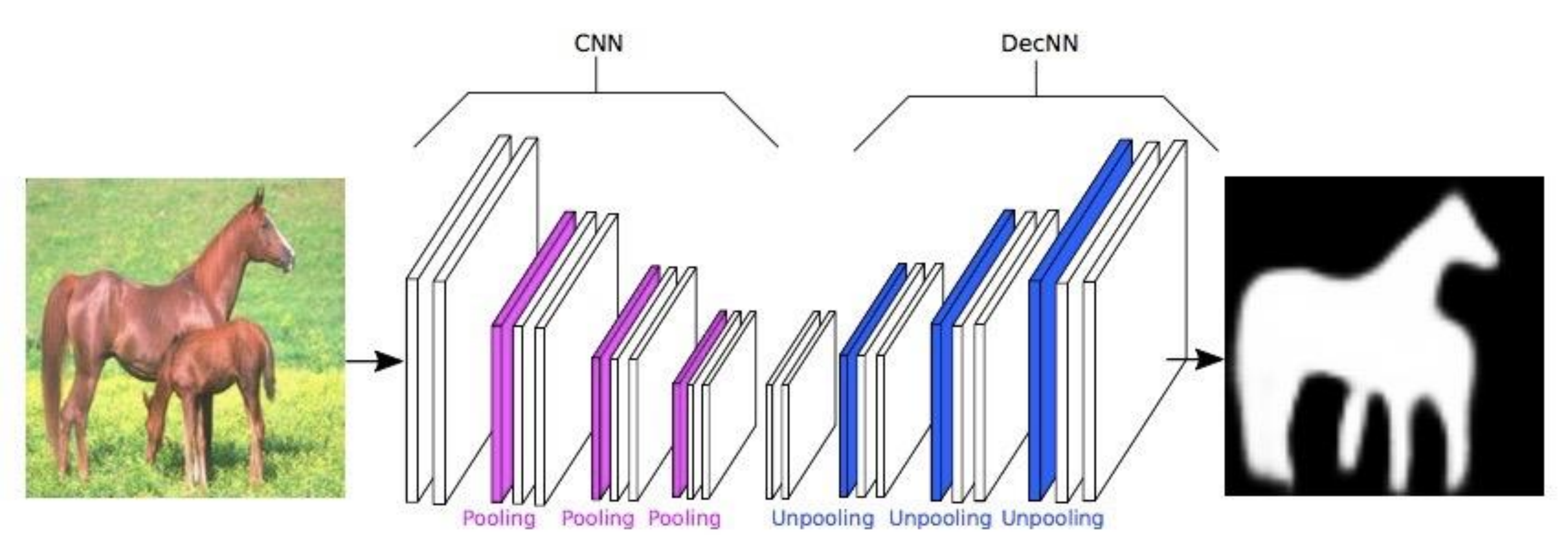
Recurrent Attentional Networks for Saliency Detection

- Humans do not tend to process a whole scene at once.
- Humans focus attention selectively on parts of the visual space to acquire information.
- The dynamic behavior of the sequence can be modelled by RNNs hence the processing of current part can leverage contextual information from previous parts

Jason Kuen, Zhenhua Wang, and Gang Wang, "Recurrent Attentional Networks for Saliency Detection," CVPR, 2016.

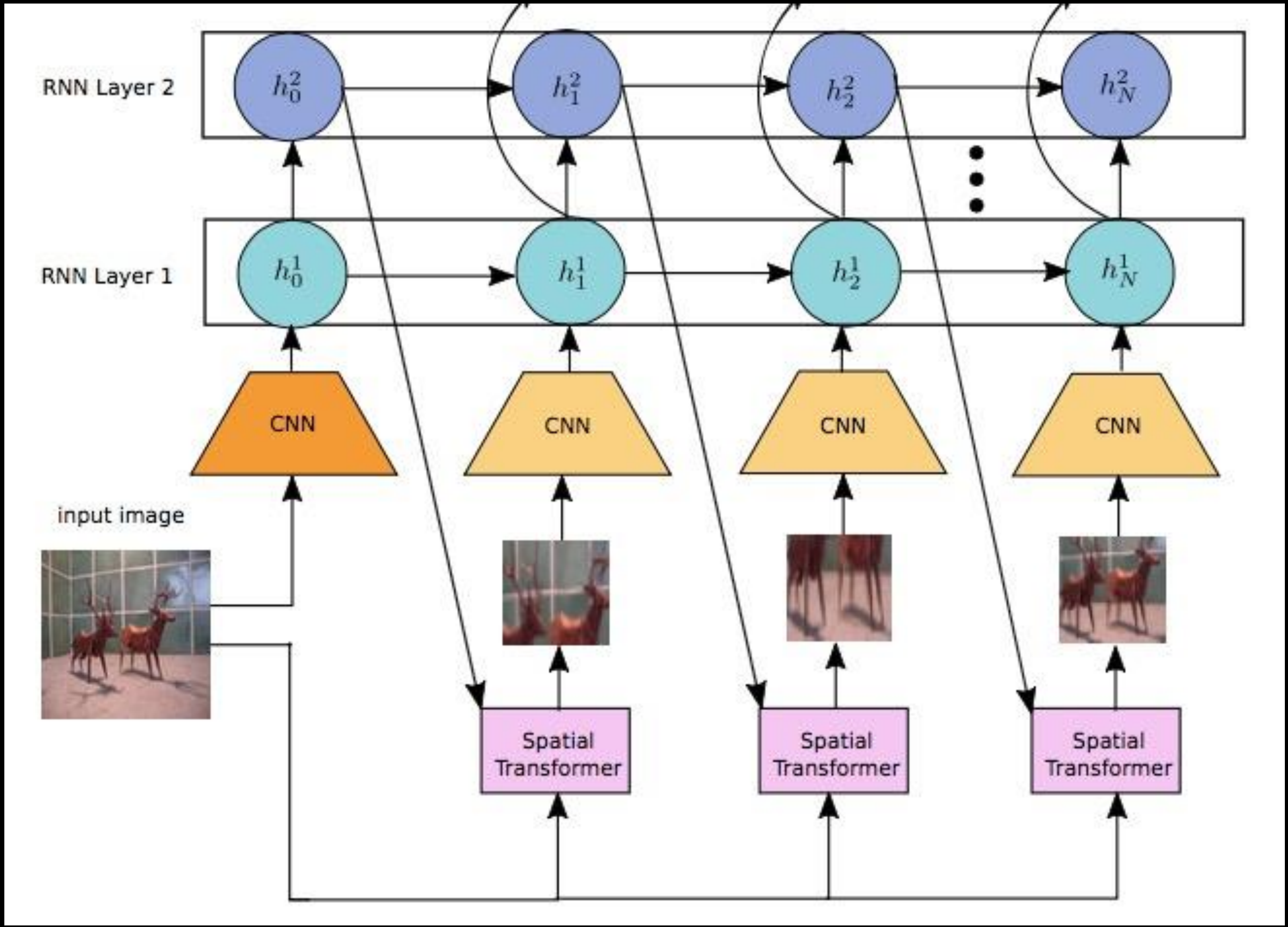
Salient Object Detection and Segmentation

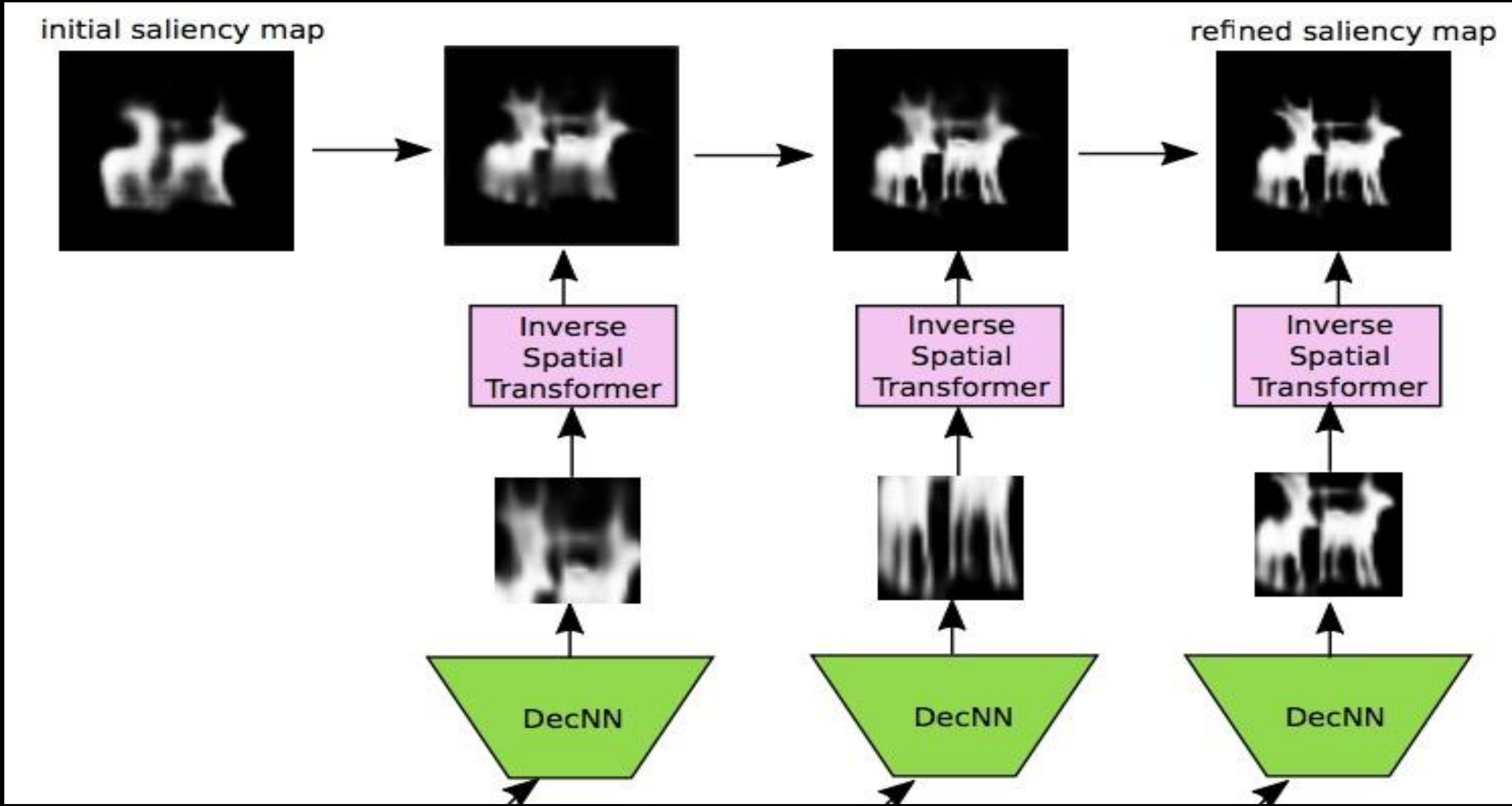
A Baseline Saliency Detection Framework



Limitation: cannot well detect and segment objects at multiple scales

Our Method


RNN Layer 2
h_0^2
h_1^2
h_2^2
h_N^2
RNN Layer 1
h_0^1
h_1^1
h_2^1
h_N^1
CNN
CNN
CNN
CNN
input image
Spatial Transformer
Spatial Transformer
Spatial Transformer



initial saliency map
refined saliency map
Inverse Spatial Transformer
Inverse Spatial Transformer
Inverse Spatial Transformer
DecNN
DecNN
DecNN

Implementation details

- For initial saliency detection, we use a CNN-DecNN independent from the CNN-DecNN used in the saliency refinement stage
- We train the deep models (initial CNN-DecNN and RADCNN) in our proposed method on saliency datasets different from the datasets used for experimental evaluations. The training datasets we use have a total number of 12,430 images.
- Experiments on 5 benchmark datasets

Comparison with Baselines

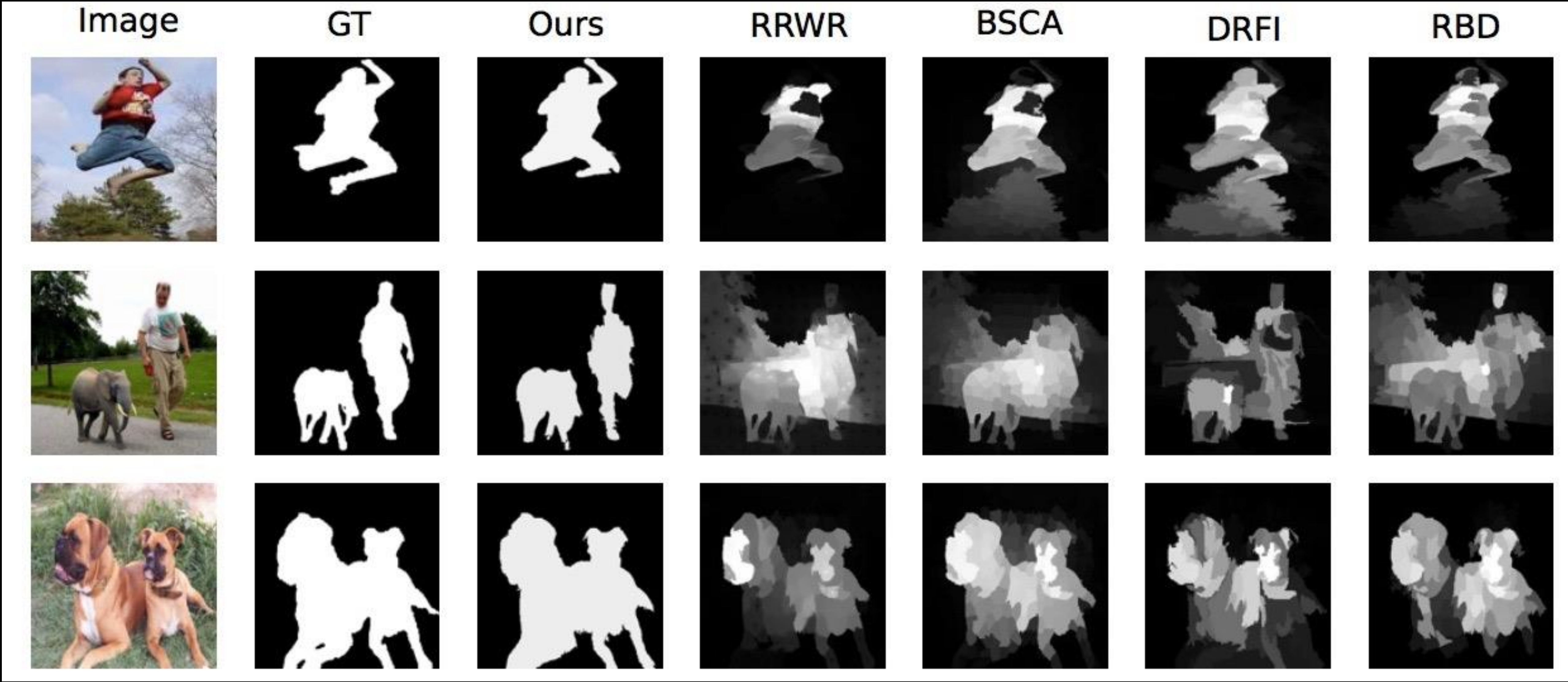
in %	MSRA10K	THUR15K	HKUIS	ECSSD	SED2
CNN-DecNN	$87.91 \pm 0.03, 7.03 \pm 0.03$	$69.28 \pm 0.01, 10.42 \pm 0.04$	$82.48 \pm 0.03, 8.10 \pm 0.01$	$85.72 \pm 0.03, 8.72 \pm 0.02$	$82.79 \pm 0.05, 9.29 \pm 0.01$
NRACDNN	$88.62 \pm 0.06, 6.85 \pm 0.04$	$70.39 \pm 0.14, 10.46 \pm 0.08$	$83.74 \pm 0.08, 7.88 \pm 0.02$	$86.65 \pm 0.07, 8.43 \pm 0.01$	$83.99 \pm 0.06, 9.30 \pm 0.03$
RACDNN	$89.98 \pm 0.04, 6.02 \pm 0.03$	$71.12 \pm 0.03, 9.04 \pm 0.05$	$85.57 \pm 0.04, 7.03 \pm 0.03$	$87.81 \pm 0.03, 8.12 \pm 0.01$	$85.35 \pm 0.12, 9.29 \pm 0.01$

CNN-DecNN： the convolutional and deconvolutional network
NRACDNN: no RNN between different image regions
RACDNN: our full model

Result Comparison

	MSRA10K	THUR15K	HKUIS	ECSSD	SED2
RRWR [27]	0.850	0.600	0.713	0.747	0.780
BSCA [36]	0.859	0.609	0.719	0.760	0.783
DRFI [23]	0.881	0.670	0.773	0.787	0.839
RBD [47]	0.856	0.596	0.723	0.718	0.830
DSR [29]	0.835	0.611	0.735	0.737	0.789
MC [22]	0.848	0.610	0.723	0.742	0.771
HS [38]	0.845	0.585	0.708	0.740	0.784
CNN-DecNN + RACDNN	0.902	0.711	0.856	0.878	0.854

Qualitative Results



Conclusions

- Recurrent neural networks are powerful for modelling contextual dependency between different image units

- Our works on scene labelling and saliency detection have shown the effectiveness of RNNs

- Future works can be done to improve the memory capacity

Our group is continuously looking for self-motivated PhD students, researchers, and interns to join us!

wanggang@ntu.edu.sg