High-order Statistical Modeling based Deep CNNs

Part 3: Approximation and Extensions

Wangmeng Zuo

Harbin Institute of Technology

Outline

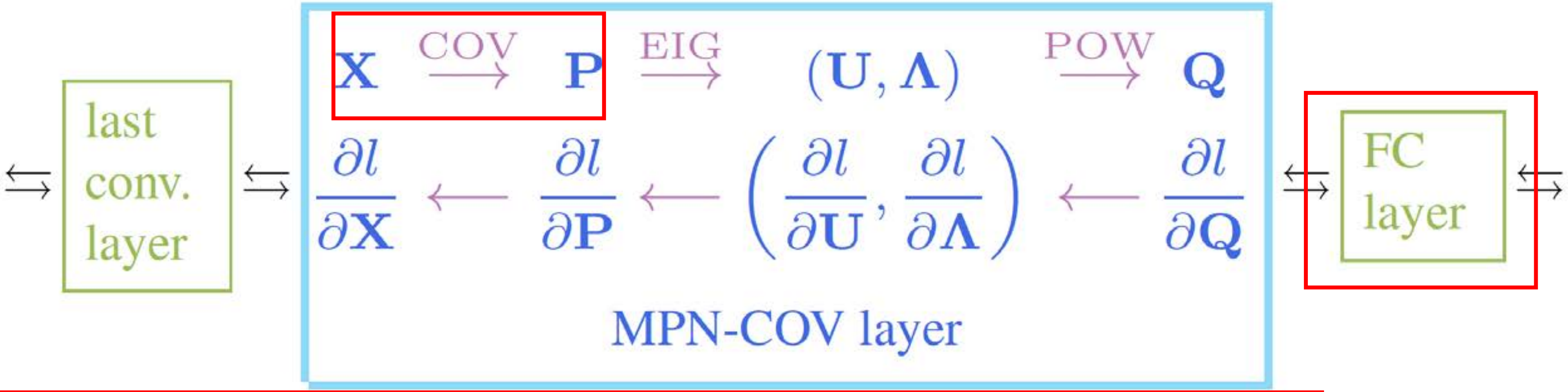
- ➢ Connect High-order with Metric Learning

- ➢ Extension to Higher-order

- ➢ Extension to Dense Higher-order

Revisit CNN with 2nd-order Info.

- MPN-COV



- 1. $\mathbf{X}\mapsto\mathbf{P}, \quad \mathbf{P}=\mathbf{X}\bar{\mathbf{I}}\mathbf{X}^T \quad \Longleftrightarrow \quad \tilde{\mathbf{P}}=\mathbf{X}\mathbf{X}^T$
- 2. $\mathbf{Q}\triangleq\mathbf{P}^{\alpha}=\mathbf{U}\mathbf{F}(\mathbf{\Lambda})\mathbf{U}^T \quad \Longleftrightarrow \quad \tilde{\mathbf{Q}}=\tilde{\mathbf{P}}$
- 3. $\langle\mathbf{W},\mathbf{Q}\rangle \quad \Longleftrightarrow \quad \langle\mathbf{W},\tilde{\mathbf{Q}}\rangle$

$$\left\langle \mathbf{W}, \sum_i \mathbf{x}_i\mathbf{x}_i^T \right\rangle = \sum_i \mathbf{x}_i^T\mathbf{W}\mathbf{x}_i$$

P. Li, J. Xie, Q. Wang, W. Zuo, Is Second-order Information Helpful for Large-scale Visual Recognition? ICCV 2017.

Revisit Metric Learning

- Mahalanobis distance

$$d_{\mathbf{M}}^{2}(\mathbf{x}_i,\mathbf{x}_j)=(\mathbf{x}_i-\mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i-\mathbf{x}_j)=\left\langle \mathbf{M},(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^T \right\rangle$$

- Euclidean distance in transformed space

$$\mathbf{M}=\mathbf{A}^T\mathbf{A}$$

$$d_{\mathbf{M}}^{2}(\mathbf{x}_i,\mathbf{x}_j)=\left\|\mathbf{A}\mathbf{x}_i-\mathbf{A}\mathbf{x}_j\right\|^2$$

- Connect High-order with Metric Learning $\mathbf{W}=\mathbf{A}^T\mathbf{A}$

$$\left\langle \mathbf{W},\sum_i \mathbf{x}_i\mathbf{x}_i^T \right\rangle=\left\langle \mathbf{1},(\mathbf{AX})\circ(\mathbf{AX})\right\rangle$$

F. Wang, W. Zuo, L. Zhang, D. Meng, and D. Zhang, A Kernel Classification Framework for Metric Learning, IEEE T-NNLS, 26(9): 1950 - 1962, 2015.

Revisiting Matrix Power Normalization with α=0.5

- Matrix Power Normalization

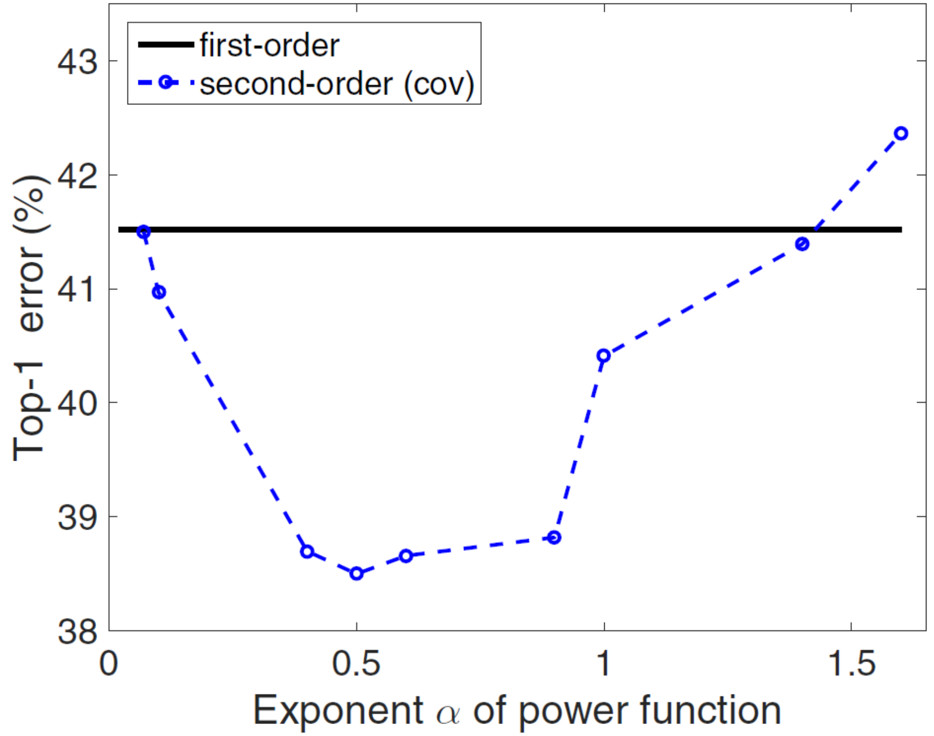
$$\mathbf{Q} \triangleq \mathbf{P}^{\alpha} = \mathbf{U}\mathbf{F}(\mathbf{\Lambda})\mathbf{U}^{T}$$

$$f(\lambda_i) = \lambda_i^{\alpha}$$

- Connect with spectral regularization

$$\begin{aligned} \min_{\mathbf{B}} \quad & L(\mathbf{B}) = \|\mathbf{B}\|_* + \tau' tr\left(\mathbf{B}^{-1}\mathbf{S}\right), \\ s.t. \quad & \mathbf{B} \succ 0. \end{aligned}$$

$$\hat{\mathbf{B}} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^{T}, \mathbf{\Sigma} = diag\{(\tau'\lambda_1)^{1/2}, \cdots, (\tau'\lambda_d)^{1/2}\}$$



X. Wu, W. Zuo, L. Lin, W. Jia, D. Zhang, F-SVM: Combination of Feature Transformation and SVM Learning via Convex Relaxation, IEEE T-NNLS, 29(11): 5185-5199, 2018

Going beyond (1)

- Batch Normalization on Feature Maps

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

- Orthogonal regularization on convolution filters

$$\|W^{\mathrm{T}}W - I\|_F^2$$

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML 2015, Lille, France, 6-11 July 2015*, pages 448–456, 2015.
K. Jia, D. Tao, S. Gao, and X. Xu, Improving training of deep neural networks via Singular Value Bounding, CVPR 2017

Going beyond (2)

- Weight normalization

$$\sigma_1(\bar{W}_{\mathrm{WN}})^2 + \sigma_2(\bar{W}_{\mathrm{WN}})^2 + \cdots + \sigma_T(\bar{W}_{\mathrm{WN}})^2 = d_o$$

- Spectral normalization

$$\bar{W}_{\mathrm{SN}}(W) := W/\sigma(W)$$

- Explicit Matrix Power Normalization? Implicit Matrix Regularization?

T. Salimans and D.P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In NIPS, pp. 901–909, 2016.
T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks, ICLR 2018

Limitation of MPN-COV

$$\tilde{\mathbf{P}} = \mathbf{X}\mathbf{X}^T$$

- Incapable of capturing higher-order (>2) statistics

- Loss of spatial information

$$\tilde{\mathbf{P}} = \sum_i \mathbf{x}_i \mathbf{x}_i^T$$

  - May be less important for image classification
  - But is crucial to object detection and localization

- Can its connection with metric learning be exploited to tackle these issues?

$$\left\langle \mathbf{W}, \sum_i \mathbf{x}_i \mathbf{x}_i^T \right\rangle = \left\langle \mathbf{1}, (\mathbf{A}\mathbf{X}) \circ (\mathbf{A}\mathbf{X}) \right\rangle$$

Outline

- Connect High-order with Metric Learning

- Extension to Higher-order

- Extension to Dense Higher-order

Intuitive Idea

$$\left\langle \mathbf{W}, \sum_i \mathbf{x}_i \mathbf{x}_i^T \right\rangle = \left\langle \mathbf{1}, (\mathbf{AX}) \circ (\mathbf{AX}) \right\rangle$$

- Generalize **1** to be any matrix
- Let the two **A**s be different
- Extension from 2nd-order to higher order

Aggregation of Pairwise Similarities

- Aggregation kernel

$$\mathcal{K}(\boldsymbol{\mathcal{X}}, \bar{\boldsymbol{\mathcal{X}}}) = Agg(\{k(\boldsymbol{x}_p, \bar{\boldsymbol{x}}_{\bar{p}})\}_{p \in \Omega, \bar{p} \in \bar{\Omega}}) = \psi(\boldsymbol{\mathcal{X}})^T \psi(\bar{\boldsymbol{\mathcal{X}}})$$

$$\psi(\boldsymbol{\mathcal{X}}) = g(\{\phi(\boldsymbol{x}_p)\}_{p \in \Omega}) = \sum_{p \in \Omega} \phi(\boldsymbol{x}_p)$$

- Explicit kernel representation/approximation

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle$$

S. Cai, W. Zuo, L. Zhang, Higher-order Integration of Hierarchical Convolutional Activations for Fine-grained Visual Categorization, ICCV 2017.

Predictor Based on Kernels

- Linear kernel

$$f(\mathbf{x}) = \sum_k w_k x_k$$

- Homogeneous polynomial kernel

$$f(\boldsymbol{x}) = \sum_{k_1,\ldots,k_r} \mathcal{W}^r_{k_1,\ldots,k_r} (\prod_{s=1}^{r} x_{k_s})$$

- Polynomial kernel

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} w_k x_k + \sum_{r=2}^{R} \sum_{k_1,\ldots,k_r} \mathcal{W}^r_{k_1,\ldots,k_r} (\prod_{s=1}^{r} x_{k_s})$$

Tensor Approximation

- Reformulation

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + \sum_{r=2}^{R} \langle \boldsymbol{\mathcal{W}}^r, \otimes_r \boldsymbol{x} \rangle$$

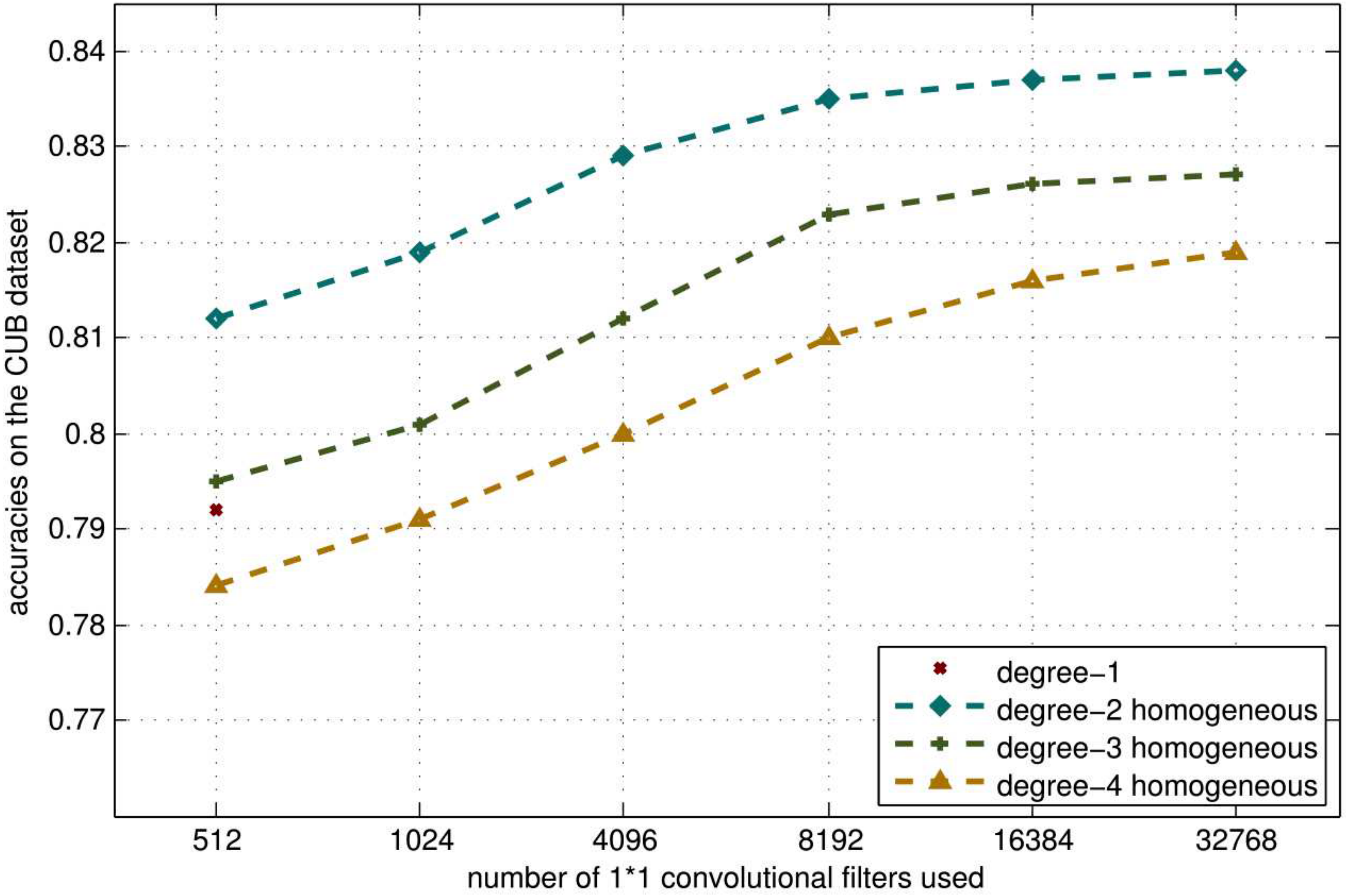
- Reminder

$$\mathbf{M} = \mathbf{A}^T \mathbf{A} \quad \Rightarrow \quad \mathbf{M} \approx \mathbf{A}^T \mathbf{A}$$

- Approximation

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + \sum_{r=2}^{R} \langle \sum_{d=1}^{D^r} \alpha^{r,d} \boldsymbol{u}_1^{r,d} \otimes \cdots \otimes \boldsymbol{u}_r^{r,d}, \otimes_r \boldsymbol{x} \rangle$$

Effect of Approximation



Trainable Polynomial Module

$$
\begin{aligned}
f(\boldsymbol{x}) &= \langle \boldsymbol{w}, \boldsymbol{x} \rangle + \sum_{r=2}^{R} \sum_{d=1}^{D^r} \alpha^{r,d} \prod_{s=1}^{r} \langle \boldsymbol{u}_s^{r,d}, \boldsymbol{x} \rangle \\
&= \langle \boldsymbol{w}, \boldsymbol{x} \rangle + \sum_{r=2}^{R} \langle \boldsymbol{\alpha}^r, \boldsymbol{z}^r \rangle
\end{aligned}
$$

- Back-propagation

$$
\begin{aligned}
\frac{\partial \ell}{\partial \boldsymbol{x}} &= \frac{\partial \ell}{\partial \boldsymbol{y}^r} \sum_{d=1}^{D^r} \sum_{s=1}^{r} \Big(\prod_{t \neq s} \langle \boldsymbol{u}_t^{r,d}, \boldsymbol{x} \rangle \Big) \boldsymbol{u}_s^{r,d} \\
\frac{\partial \ell}{\partial \boldsymbol{u}_s^{r,d}} &= \frac{\partial \ell}{\partial \boldsymbol{y}^r} \Big(\prod_{t \neq s} \langle \boldsymbol{u}_t^{r,d}, \boldsymbol{x} \rangle \Big) \boldsymbol{x}
\end{aligned}
$$

HIHCA: Higher-order Integration of Hierarchical Convolutional Activations

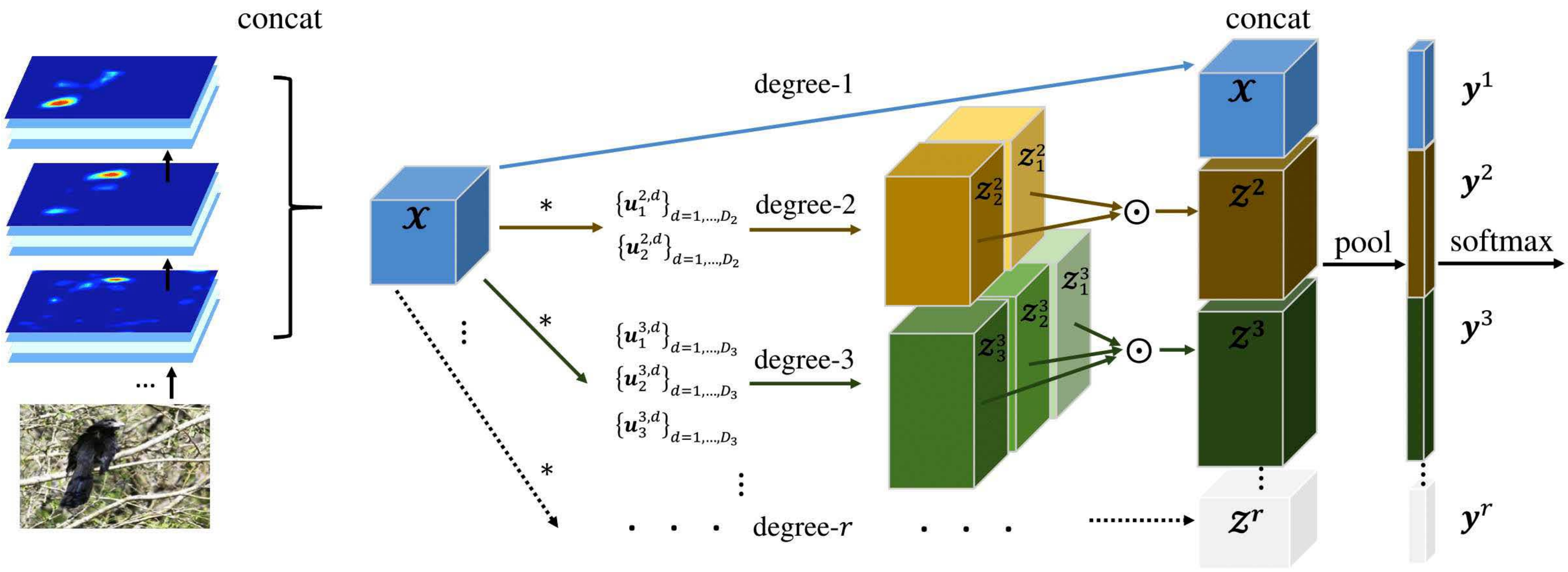
- Extend to multiple layers

$$\boldsymbol{\psi}_I \ : \ \{\boldsymbol{x}^l\}_{l=1}^L$$

- HIHCA

$$\begin{aligned} k(\boldsymbol{\psi}_{\boldsymbol{I}}, \boldsymbol{\psi}_{\bar{\boldsymbol{I}}}) &= \langle \phi(\{\boldsymbol{x}^l\}_{l=1}^L), \phi(\{\bar{\boldsymbol{x}}^l\}_{l=1}^L) \rangle \\ &= \sum_{l=1}^{L} \eta_l \langle \phi^l(\boldsymbol{x}^l), \phi^l(\bar{\boldsymbol{x}}^l) \rangle, \end{aligned}$$

Network Architecture



Effect of Polynomial Degree

- Accuracy

r	1	2	3	4	5	6
non-ft	75.7	**78.3**	76.4	74.6	72.4	71.2
ft	79.2	**83.7**	83.3	82.0	81.1	79.5

- FPS

r	2	3	4	5	6
Training	9.7	7.4	5.5	4.2	2.8
Testing	29.8	23.7	18.3	14.5	10.4

Effect of feature integration

	r5_3	*r5_3+ r5_2*	*r5_3+ r5_1*	*r5_2+ r5_1*	*r5_3+ r5_2+ r5_1*
degree-1					
non-ft	75.7	**77.2**	75.5	68.9	77.0
ft	79.2	80.4	79.3	71.1	**80.8**
degree-2 homogeneous					
non-ft	77.2	78.1	77.5	72.3	**78.4**
ft	83.5	**85.0**	83.3	76.0	84.9
degree-2 non-homogeneous					
non-ft	78.3	78.5	77.5	72.1	**78.6**
ft	83.7	**85.3**	83.6	76.5	85.1
degree-3 homogeneous					
non-ft	75.7	**76.9**	76.0	70.7	76.1
ft	82.3	**83.8**	81.5	74.1	83.3
degree-3 non-homogeneous					
non-ft	76.4	**78.2**	77.4	72.3	78.1
ft	83.3	**84.6**	82.1	75.4	84.5

Results on Fine-grained Visual Classification

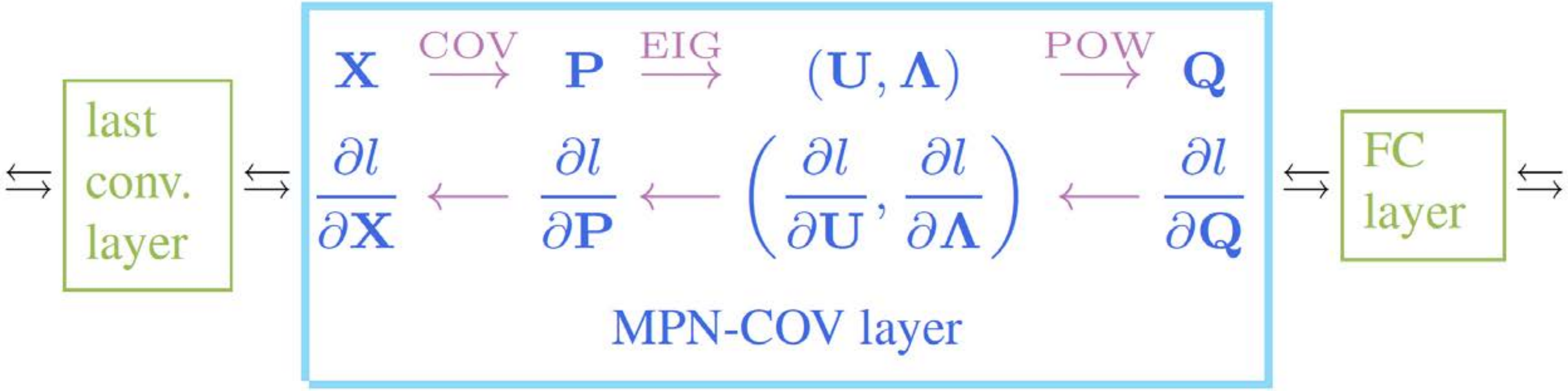
- CUB

methods	train anno.	test anno.	acc.
PB R-CNN [47]	bbox+parts	n/a	73.9
FG-Without [20]	bbox	bbox	82.0
SPDA-CNN [46]	bbox+parts	bbox+parts	85.1
STN [17]	n/a	n/a	84.1
B-CNN [24]	n/a	n/a	84.1
PDFS [48]	n/a	n/a	84.5
BoostCNN [30]	n/a	n/a	**85.6**
Ours	n/a	n/a	85.3

- Aircraft and Cars

methods	acc. (Aircraft)	acc. (Cars)
Symbiotic [6]	72.5	78.0
FV-FGC [14]	80.7	82.7
B-CNN [24]	84.1	91.3 (90.6)
Ours	**88.3**	**91.7**

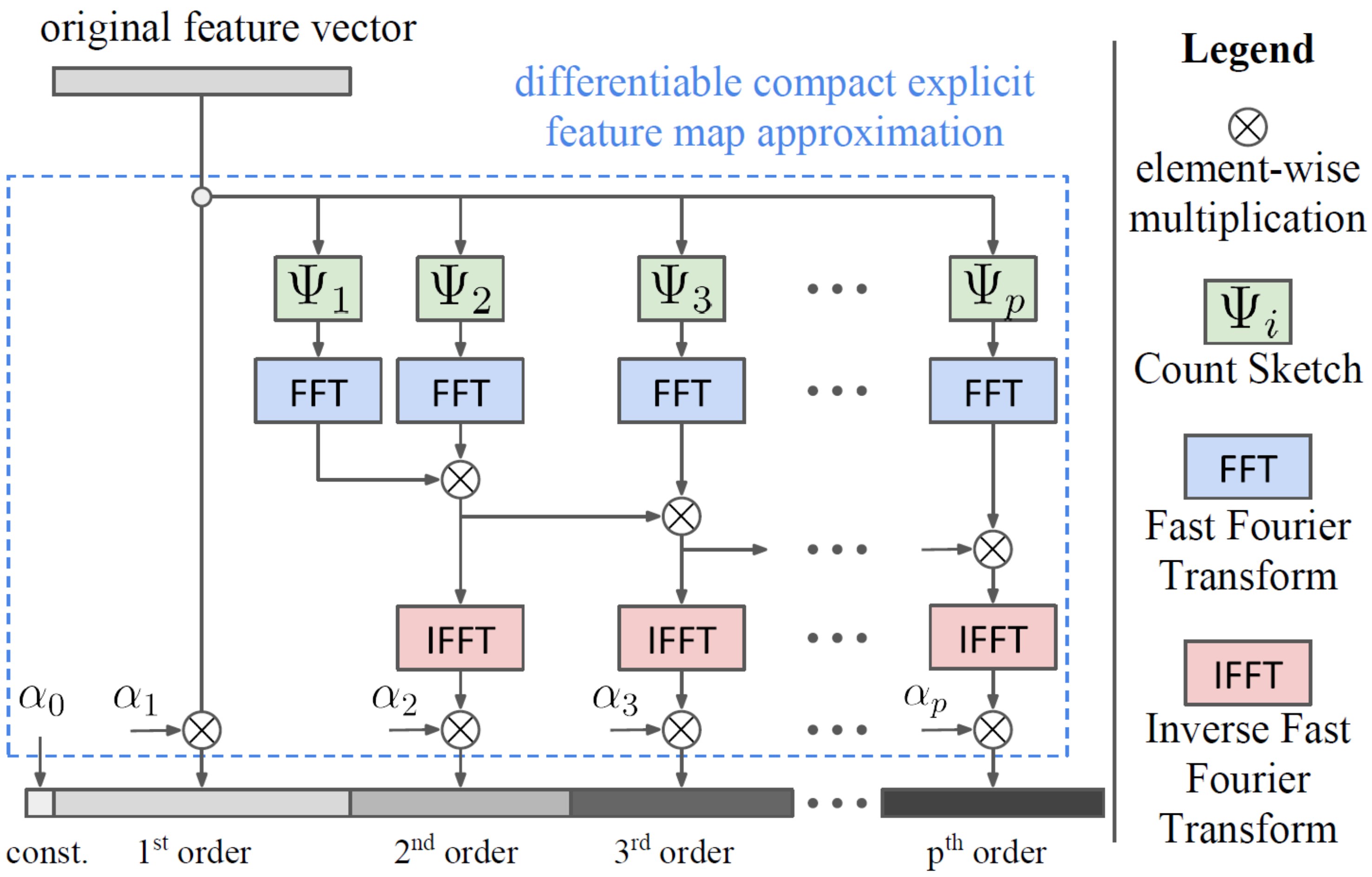
Brief Summary (1)



$$\left\langle \mathbf{W}, \sum_i \mathbf{x}_i \mathbf{x}_i^T \right\rangle = \left\langle \mathbf{1}, (\mathbf{AX}) \circ (\mathbf{AX}) \right\rangle$$

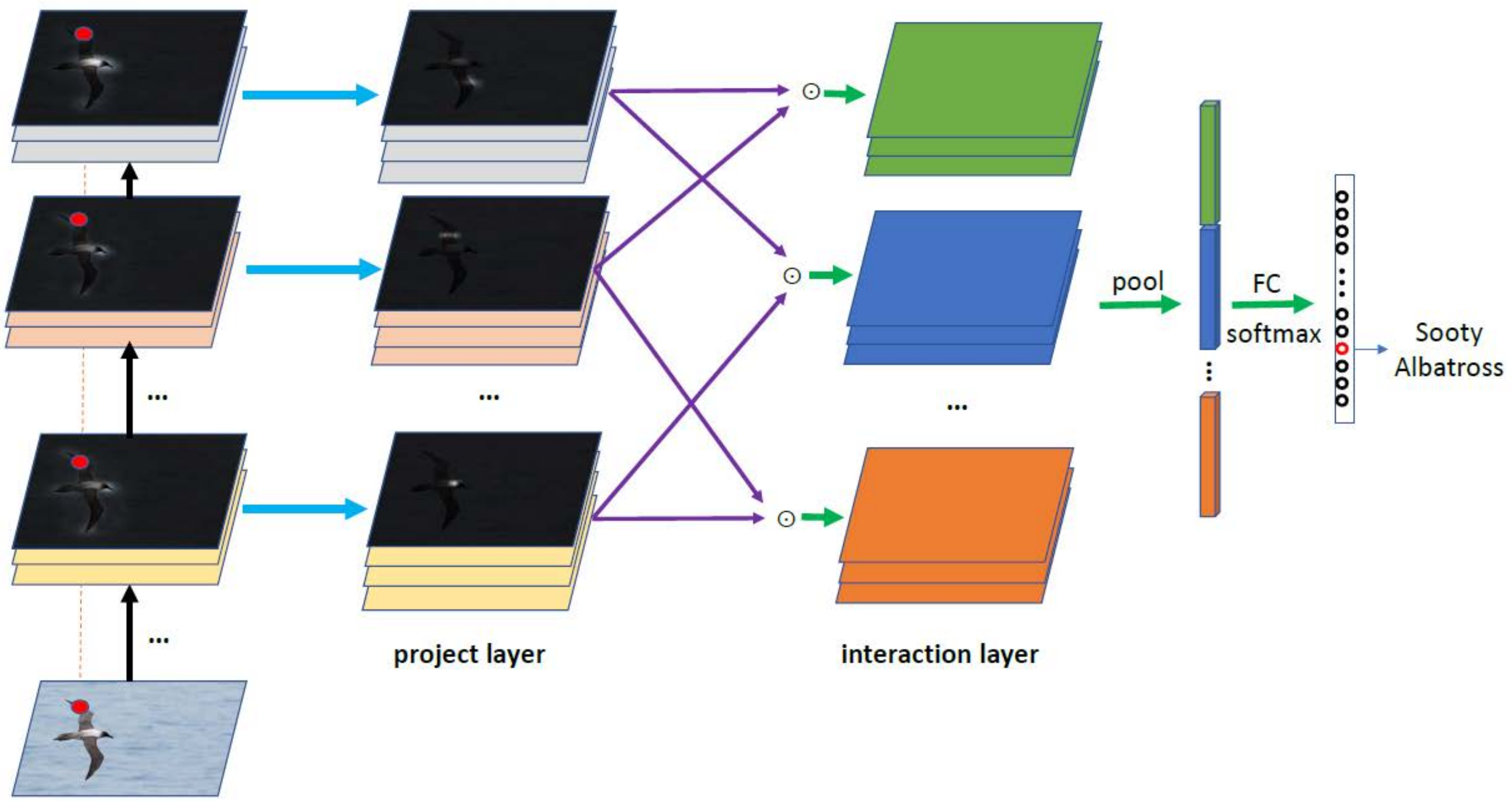
$$\begin{aligned} f(\boldsymbol{x}) &= \langle \boldsymbol{w}, \boldsymbol{x} \rangle + \sum_{r=2}^{R} \sum_{d=1}^{D^r} \alpha^{r,d} \prod_{s=1}^{r} \langle \boldsymbol{u}_s^{r,d}, \boldsymbol{x} \rangle \\ &= \langle \boldsymbol{w}, \boldsymbol{x} \rangle + \sum_{r=2}^{R} \langle \boldsymbol{\alpha}^r, \boldsymbol{z}^r \rangle \end{aligned}$$

Note (1): Kernel Pooling



Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, S. Belongie, Kernel Pooling for Convolutional Neural Networks, CVPR 2017

Note (2): Hierarchical Bilinear Pooling

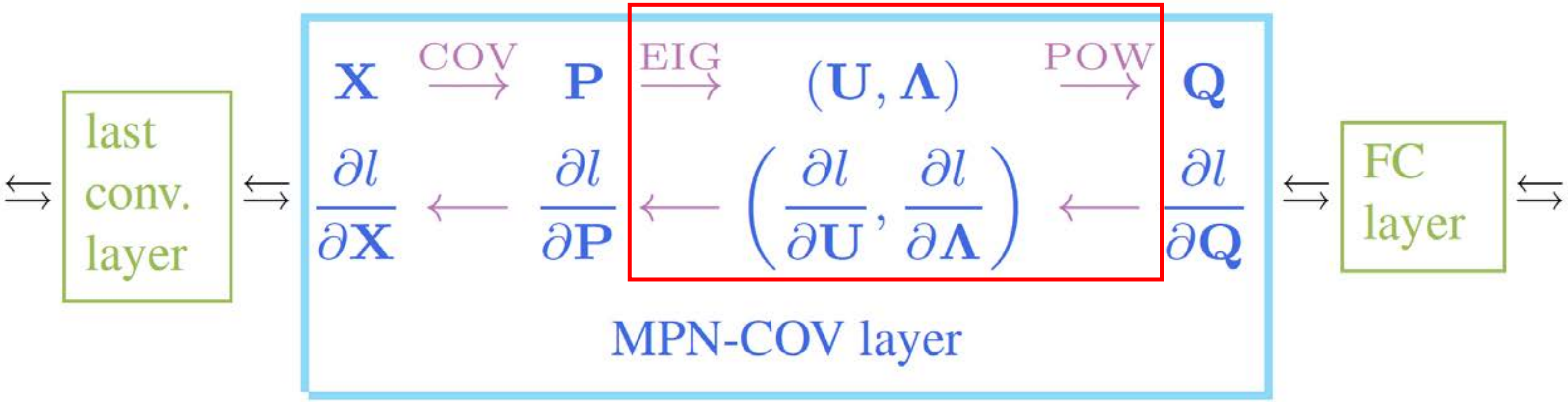


C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, Hierarchical Bilinear Pooling for Fine-Grained Visual Recognition, ECCV 2018

Outline

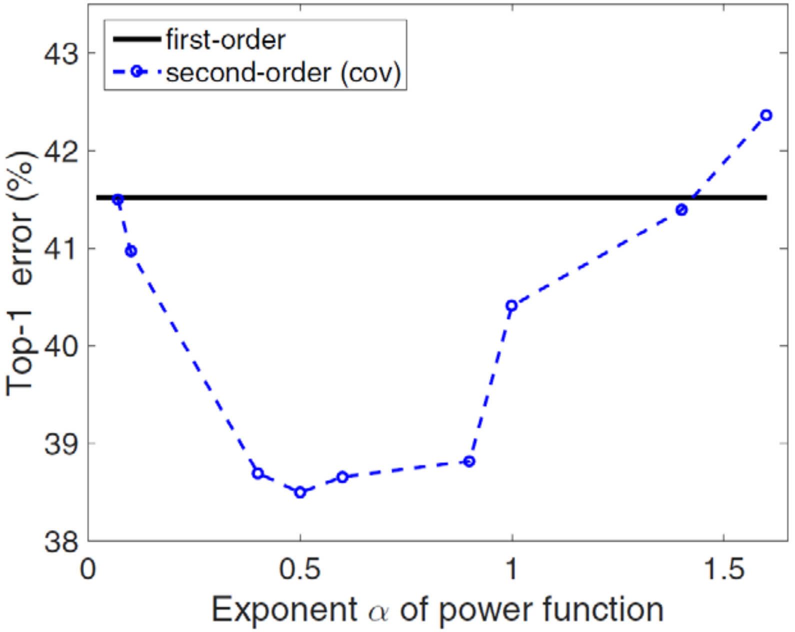
- Connect High-order with Metric Learning
- Extension to Higher-order
- Extension to Dense Higher-order

Pros and Cons of MPN-COV

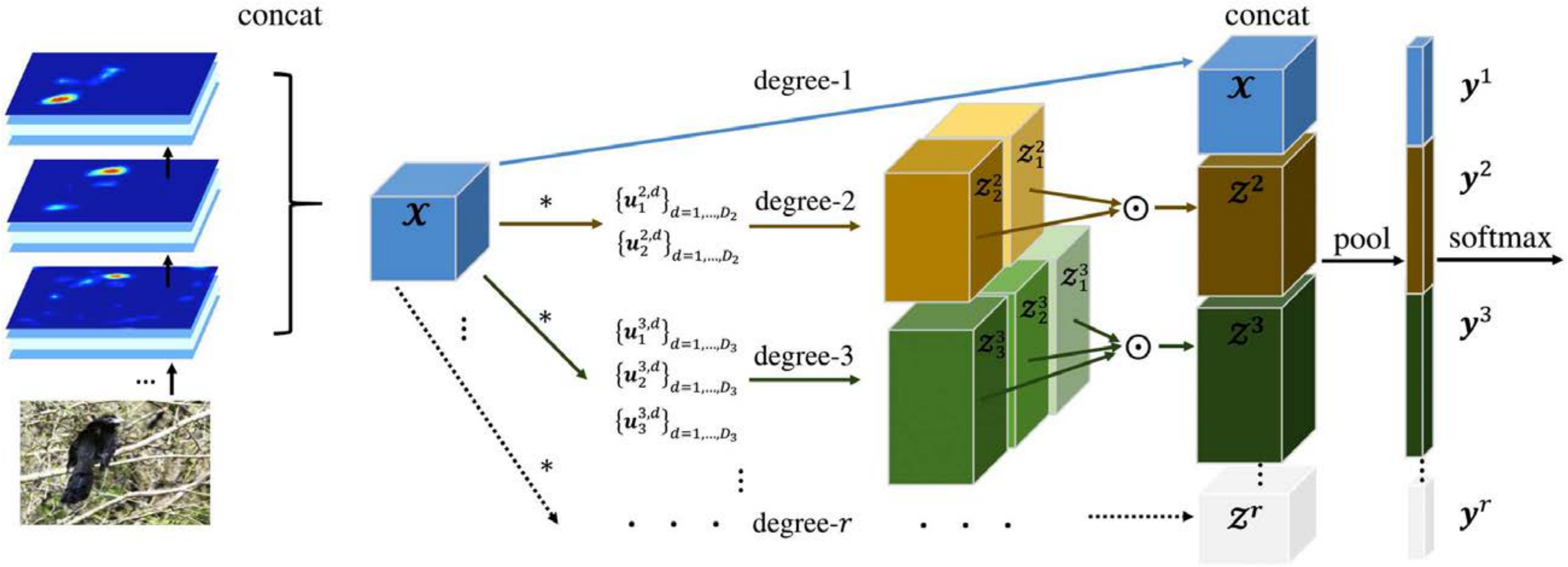


- Pros
  - Matrix power normalization

- Cons
  - Loss of spatial information



Turn to HIHCA



- Pros
  - Higher-order info.
- Cons
  - Cannot exploit matrix power normalization
  - Loss of spatial information
    - Sure?

Approximating MPN with Attention

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \qquad \mathbf{\Sigma} = \frac{1}{N}\sum_i \mathbf{x}_i \mathbf{x}_i^T$$

$$\mathbf{y} = w(\mathbf{x})\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Xi}) \qquad \mathbf{\Xi} = \frac{1}{N}\sum_i w^2(\mathbf{x}_i)\mathbf{x}_i \mathbf{x}_i^T$$

$$w(\mathbf{x}) = \frac{p(\mathbf{y})}{p(\mathbf{x})} \approx Attend(\mathbf{x})$$

Attention module

Using Object Detection as an Example

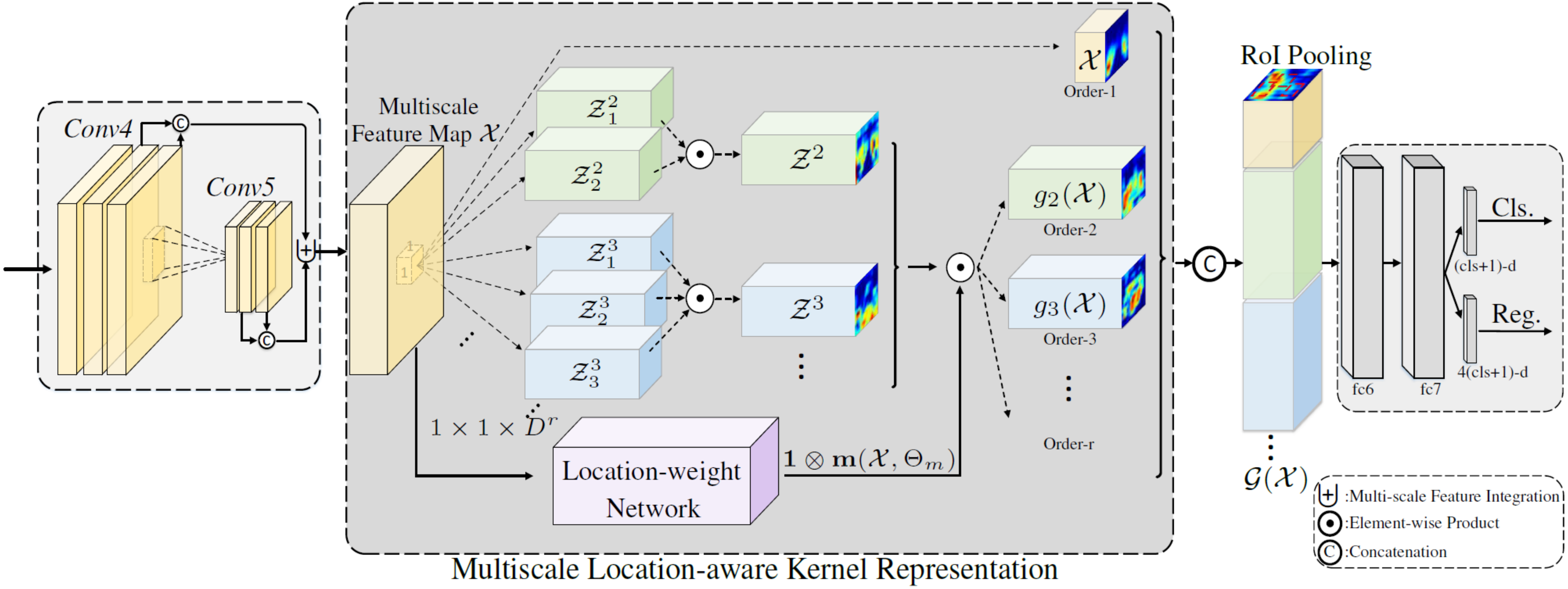
- Spatial info. is critical
  - Bounding box regression

- High or higher-order may be helpful
  - But not investigated yet

- MPN may be useful
  - Approximate with attention

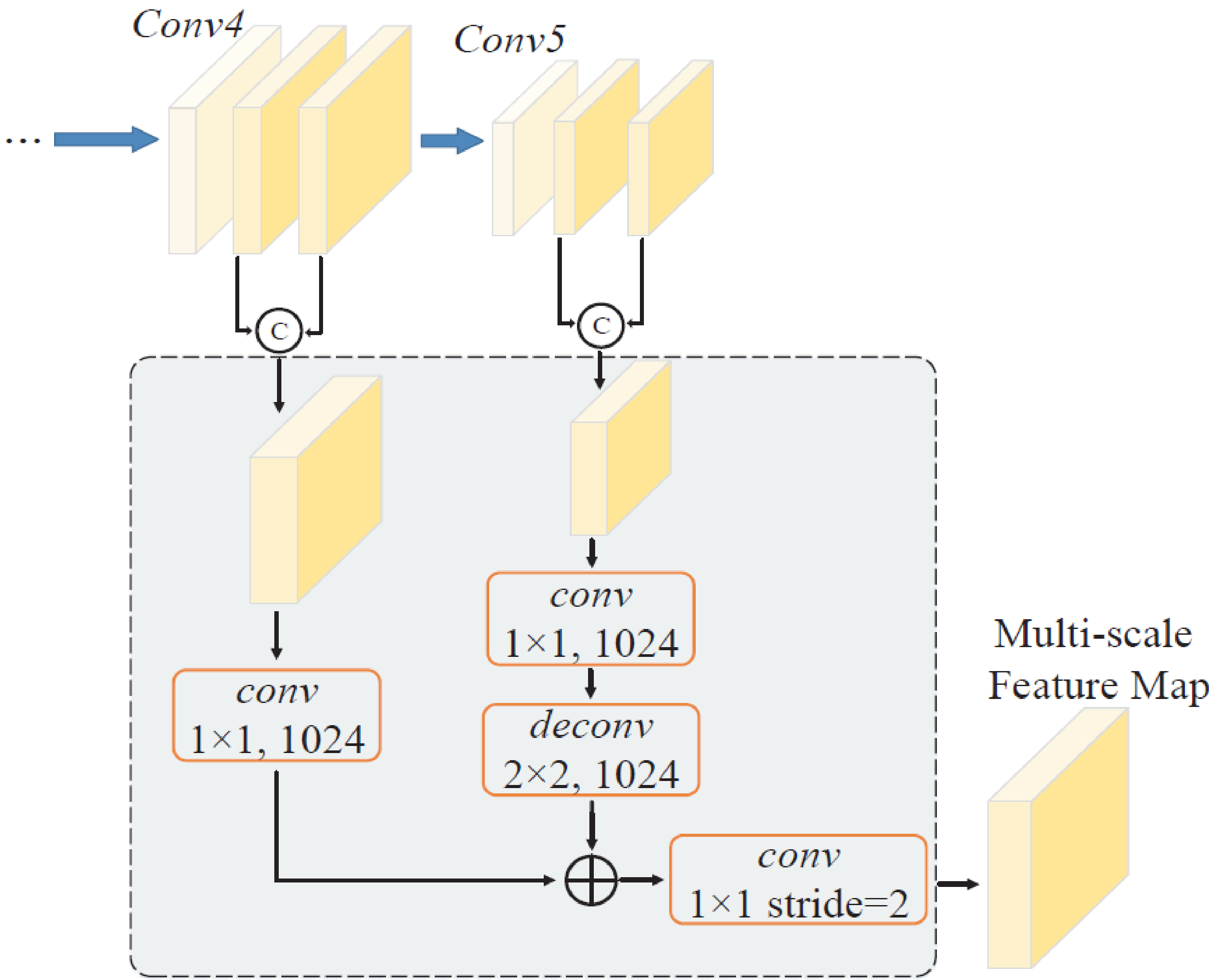
H. Wang, Q. Wang, M. Gao, P. Li, W. Zuo, Multi-scale Location-aware Kernel Representation for Object Detection, CVPR 2018.

Multi-scale Location-aware Kernel Representation



Multiscale Feature Map

- Multiple layers in each convolution block
- Upsampling
- Element-wise sum



Location-aware Kernel Representation

$$
\begin{aligned}
f(\mathcal{X}) &= \sum_{\mathbf{x}\in\mathcal{X}} \left\{ \langle \mathbf{w}^1, \mathbf{x} \rangle + \sum_{r=2}^{R} \sum_{d=1}^{D^r} a^{r,d} \prod_{s=1}^{r} \langle \mathbf{u}_s^{r,d}, \mathbf{x} \rangle \right\} \\
&= \left\{ \left\langle \mathbf{w}^1, \sum_{\mathbf{x}\in\mathcal{X}} \mathbf{x} \right\rangle + \sum_{r=2}^{R} \left\langle \mathbf{a}^r, \sum_{\mathbf{z}^r\in\mathcal{Z}^r} \mathbf{z}^r \right\rangle \right\} \quad (3)
\end{aligned}
$$

- Incorporating learnable weights

$$g_r(\mathcal{X}) = \mathcal{Z}^r \odot (\mathbf{1} \otimes \mathbf{m}(\mathcal{X}, \Theta_m))$$

- Higher orders

$$\mathcal{G}(\mathcal{X}) = [\mathcal{X}, g_2(\mathcal{X}), \ldots, g_r(\mathcal{X})]^\top$$

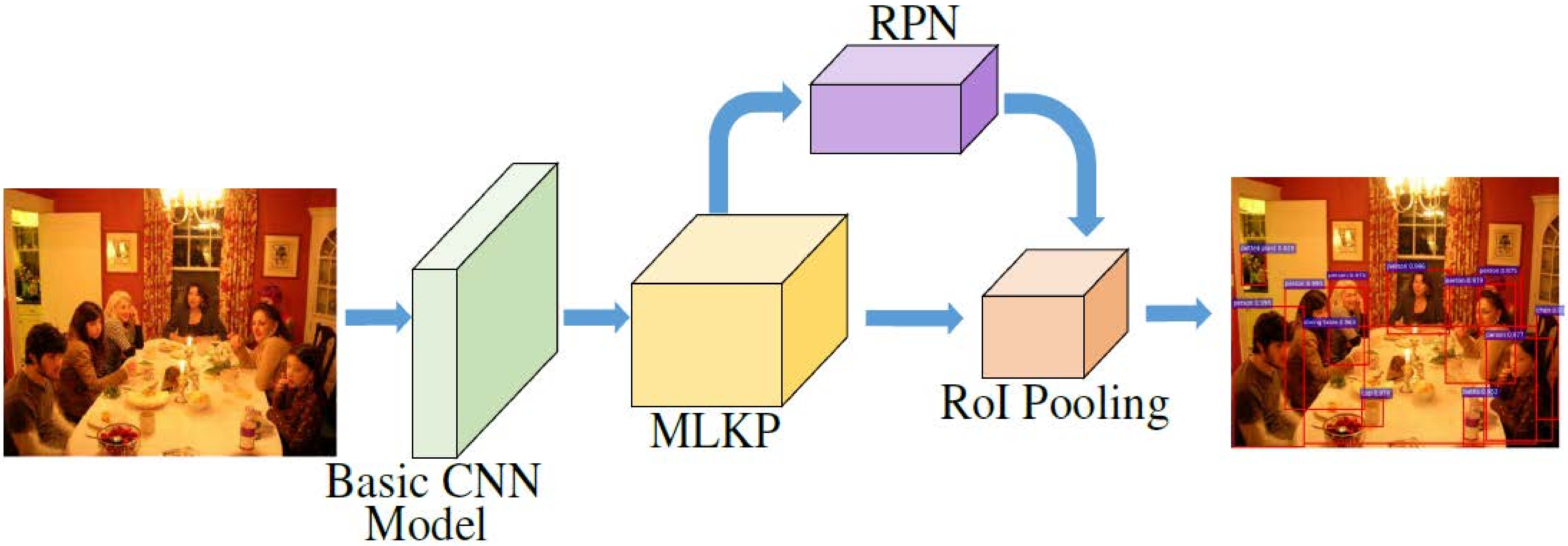
Gradient descent

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial \mathcal{L}}{\partial g_r}\left(\frac{\partial \mathcal{Z}^r}{\partial \mathbf{x}} + \sum_{d=1}^{D^r} \mathcal{Z}^r \frac{\partial \mathbf{m}}{\partial \mathbf{x}}\right)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}_s^{r,d}} = \frac{\partial \mathcal{L}}{\partial g_r} \frac{\partial \mathcal{Z}^r}{\partial \mathbf{u}_s^{r,d}} \odot (\mathbf{1} \otimes \mathbf{m}(\mathbf{x}, \Theta_m))$$

$$\frac{\partial \mathcal{L}}{\partial \Theta_m} = \frac{\partial \mathcal{L}}{\partial g_r} \mathcal{Z}^r \sum_{d=1}^{D^r} \frac{\partial \mathbf{m}}{\partial \Theta_m}$$

Whole Framework



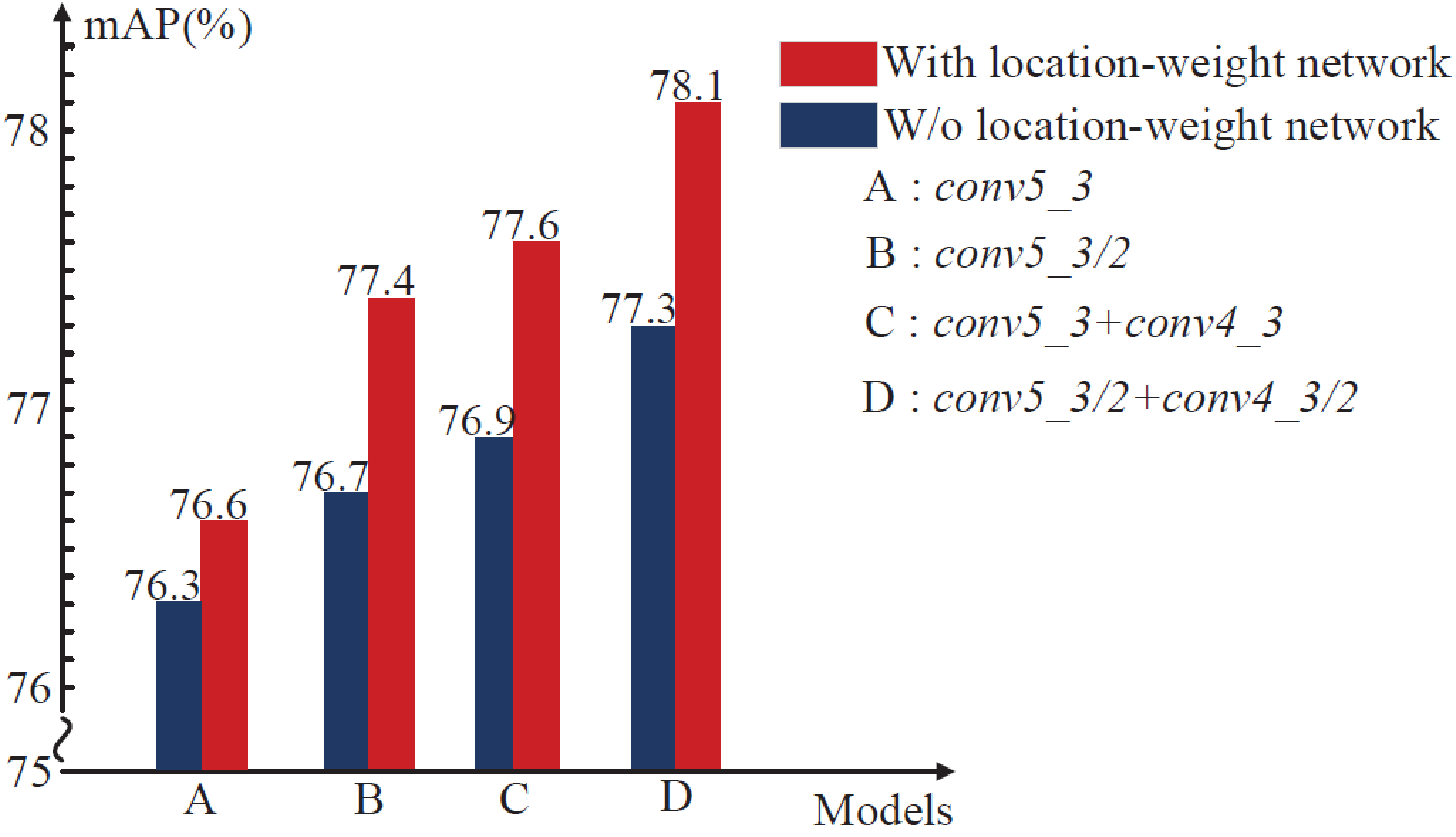
Effect of Multi-scale Feature Map

Method	mAP	Inference Time (FPS)
conv5_3	73.2	15
conv5_3+conv4_3	76.2	11
conv5_3+conv4_3+conv3_3	76.3	6
conv5_3/2	76.0	14
conv5_3/2+conv4_3/2	**76.5**	**11**

Effect of Higher-order Kernel Representation

Order	Dimension	mAP / Inference Time(FPS)	
		conv5_3	*conv5_3/2+conv4_3/2*
1	-	73.2 / 15	76.5 / 11
2	2048	76.4 / 14	77.7 / 10
	4096	76.5 / 14	77.5 / 10
3	2048	76.6 / 13	77.8 /10
	4096	**76.6 / 12**	**78.1 / 10**
	8192	76.2 /10	77.7 / 8

Effect of Location-weight Network



PASCAL VOC 2007

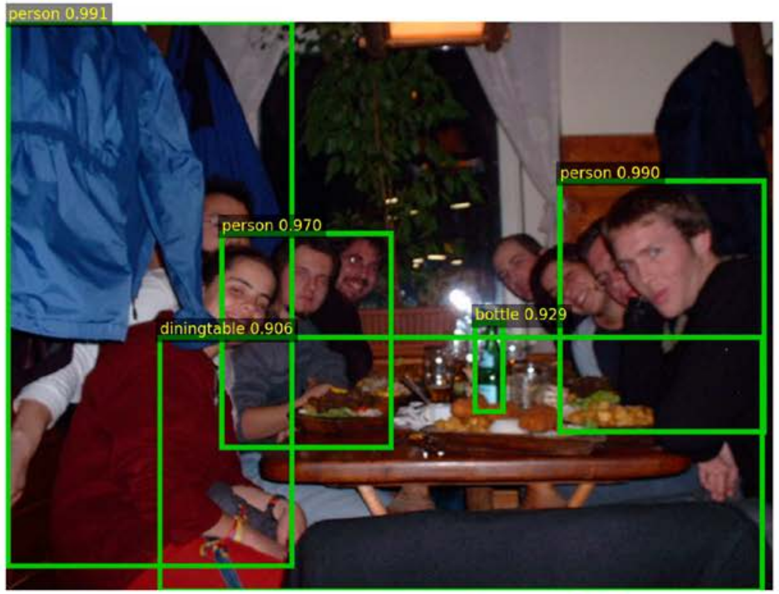
Method	Data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Faster R-CNN [33]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	75.5	76.7	38.8	73.6	73.9	83.0	72.6
HyperNet [24]	07+12	76.3	77.4	83.3	75.0	69.1	62.4	83.1	87.4	87.4	57.1	79.8	71.4	85.1	85.1	80.0	79.1	51.2	**79.1**	75.7	80.9	76.5
ION-Net [3]	07+12+s	76.5	79.2	79.2	77.4	69.8	55.7	85.2	84.3	**89.8**	57.5	78.5	73.8	**87.8**	85.9	81.3	75.3	49.7	76.9	74.6	85.2	**82.1**
SSD300 [29]	07+12	77.5	79.5	**83.9**	76.0	69.6	50.5	**87.0**	85.7	88.1	60.3	81.5	**77.0**	86.1	**87.5**	**83.9**	79.4	52.3	77.9	**79.5**	**87.6**	76.8
RON384++ [23]	07+12	77.6	**86.0**	82.5	76.9	69.1	59.2	86.2	85.5	87.2	59.9	81.4	73.3	85.9	86.8	82.2	**79.6**	52.4	78.2	76.0	86.2	78.0
MLKP (Ours)	07+12	**78.1**	78.7	83.1	**78.8**	**71.3**	**64.4**	86.1	**88.0**	87.8	**64.6**	**83.2**	73.6	85.7	86.4	81.9	79.3	**53.1**	77.2	76.7	85.0	76.1
Faster R-CNN [33]*	07+12	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	**89.8**	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0
SSD321 [29]*	07+12	77.1	76.3	84.6	79.3	64.6	47.2	85.4	84.0	88.8	60.1	82.6	76.9	86.7	87.2	85.4	79.1	50.8	77.2	82.6	**87.3**	76.6
DSSD321 [10]*	07+12	78.6	81.9	84.9	80.5	68.4	53.9	85.6	86.2	88.9	61.1	83.5	**78.7**	86.7	88.7	**86.7**	79.7	51.7	78.0	80.9	87.2	79.4
R-FCN [7]*	07+12	80.5	79.9	**87.2**	**81.5**	72.0	69.8	86.8	**88.5**	**89.8**	67.0	**88.1**	74.5	**89.8**	**90.6**	79.9	81.2	53.7	81.8	81.5	85.9	**79.9**
MLKP (Ours)*	07+12	**80.6**	**82.2**	83.2	79.5	**72.9**	**70.5**	**87.1**	88.2	88.8	**68.3**	86.3	74.5	88.8	88.7	82.0	**81.6**	**56.3**	**84.2**	**83.3**	85.3	79.7

PASCAL VOC 2012

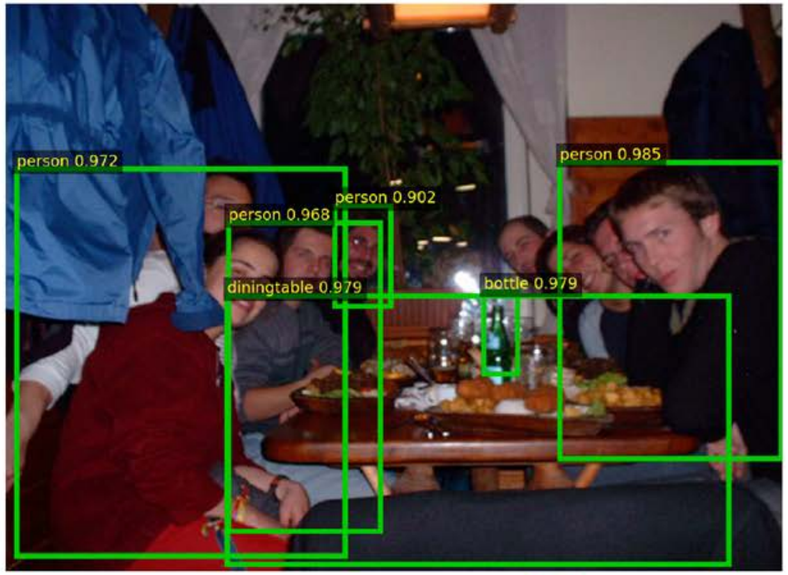
Method	Data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Faster R-CNN [33]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
HyperNet [24]	07++12	71.4	84.2	78.5	73.5	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7
SSD512 [29]	07++12	74.9	**87.4**	82.3	75.8	59.0	52.6	**81.7**	**81.5**	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	**86.3**	**72.0**
RON384++ [23]	07++12	75.4	86.5	82.9	76.6	**60.9**	55.8	**81.7**	80.2	91.1	**57.3**	**81.1**	**60.4**	87.2	**84.8**	**84.9**	81.7	51.9	**79.1**	**68.6**	84.1	70.3
MLKP (Ours)	07++12	**75.5**	86.4	**83.4**	**78.2**	60.5	**57.9**	80.6	79.5	**91.2**	56.4	81.0	58.6	**91.3**	84.4	84.3	**83.5**	**56.5**	77.8	67.5	83.9	67.4
Faster R-CNN [33]*	07++12	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	**93.2**	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
SSD321 [29]*	07++12	75.4	**87.9**	82.9	73.7	61.5	45.3	81.4	75.6	92.6	57.4	78.3	**65.0**	90.8	**86.8**	85.8	81.5	50.3	78.1	75.3	**85.2**	72.5
DSSD321 [10]*	07++12	76.3	87.3	83.3	75.4	**64.6**	46.8	**82.7**	76.5	92.9	**59.5**	78.3	64.3	91.5	86.6	**86.6**	82.1	53.3	79.6	**75.7**	**85.2**	**73.9**
MLKP(Ours)*	07++12	**77.2**	87.1	**85.1**	**79.0**	64.2	**60.3**	82.1	**80.6**	92.3	57.4	**81.8**	61.6	**92.1**	86.3	85.3	**84.3**	**59.1**	**81.7**	69.5	85.0	70.1

MS COCO *2017 test-dev*

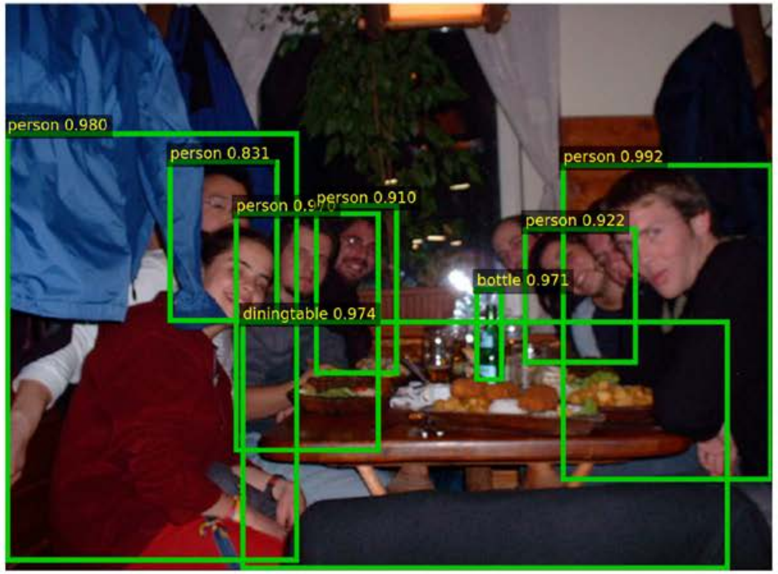
Method	Training set	Avg.Precision, IOU:			Avg.Precision, Area:			Avg.Recall, #Det:			Avg.Recall, Area:		
		0.5:0.95	0.50	0.75	Small	Med.	Large	1	10	100	Small	Med.	Large
Faster R-CNN [33]	trainval	21.9	42.7	23.0	6.7	25.2	34.6	22.5	32.7	33.4	10.0	38.1	53.4
ION [3]	train+s	24.9	44.7	25.3	7.0	26.1	40.1	23.9	33.5	34.1	10.7	38.8	54.1
SSD300 [29]	trainval35	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
SSD512 [29]	trainval35	26.8	46.5	**27.8**	**9.0**	28.9	**41.9**	24.8	37.5	**39.8**	14.0	43.5	59.0
MLKP (Ours)	trainval35	**26.9**	**48.4**	26.9	8.6	**29.2**	41.1	**25.6**	**37.9**	38.9	**16.0**	**44.1**	**59.0**
DSSD321 [10]*	trainval35	28.0	45.4	29.3	6.2	28.3	**49.3**	25.9	37.8	39.9	11.5	43.3	**64.9**
SSD321 [10]*	trainval35	28.0	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42.0	62.6
MLKP (Ours)*	trainval35	**28.6**	**52.4**	**31.6**	**10.8**	**33.4**	45.1	**27.0**	**40.9**	**41.4**	**15.8**	**47.8**	62.2



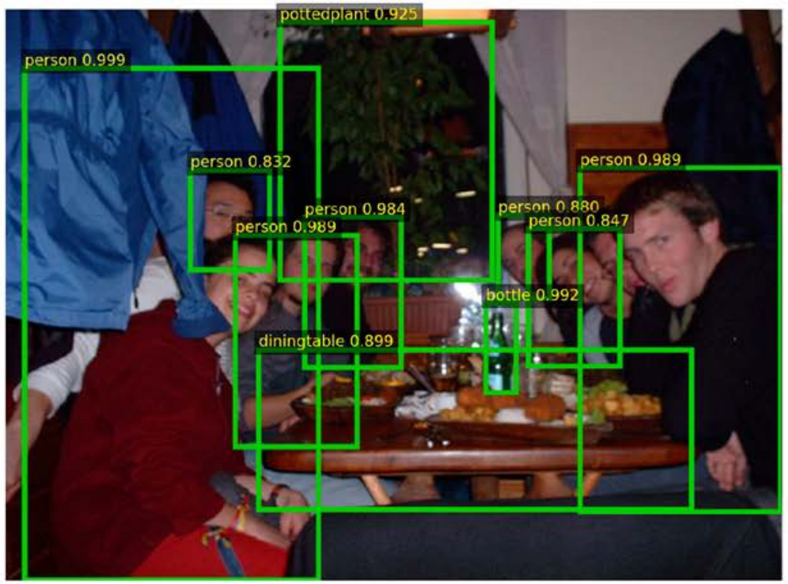
(a) Faster R-CNN [33]



(b) HyperNet [24]



(c) RON [23]

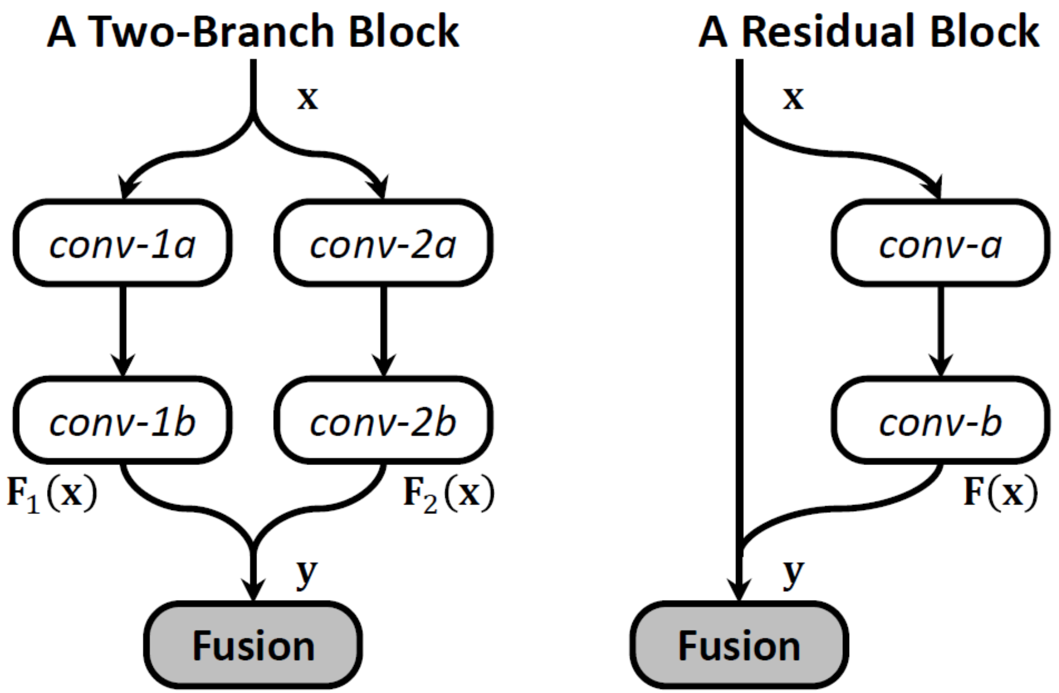


(d) Our MLKP

Brief Summary (2)

- Preserving spatial information
- Connecting MPN with Attention
- Application to Object Detection

Note (1): Second-Order Response Transform

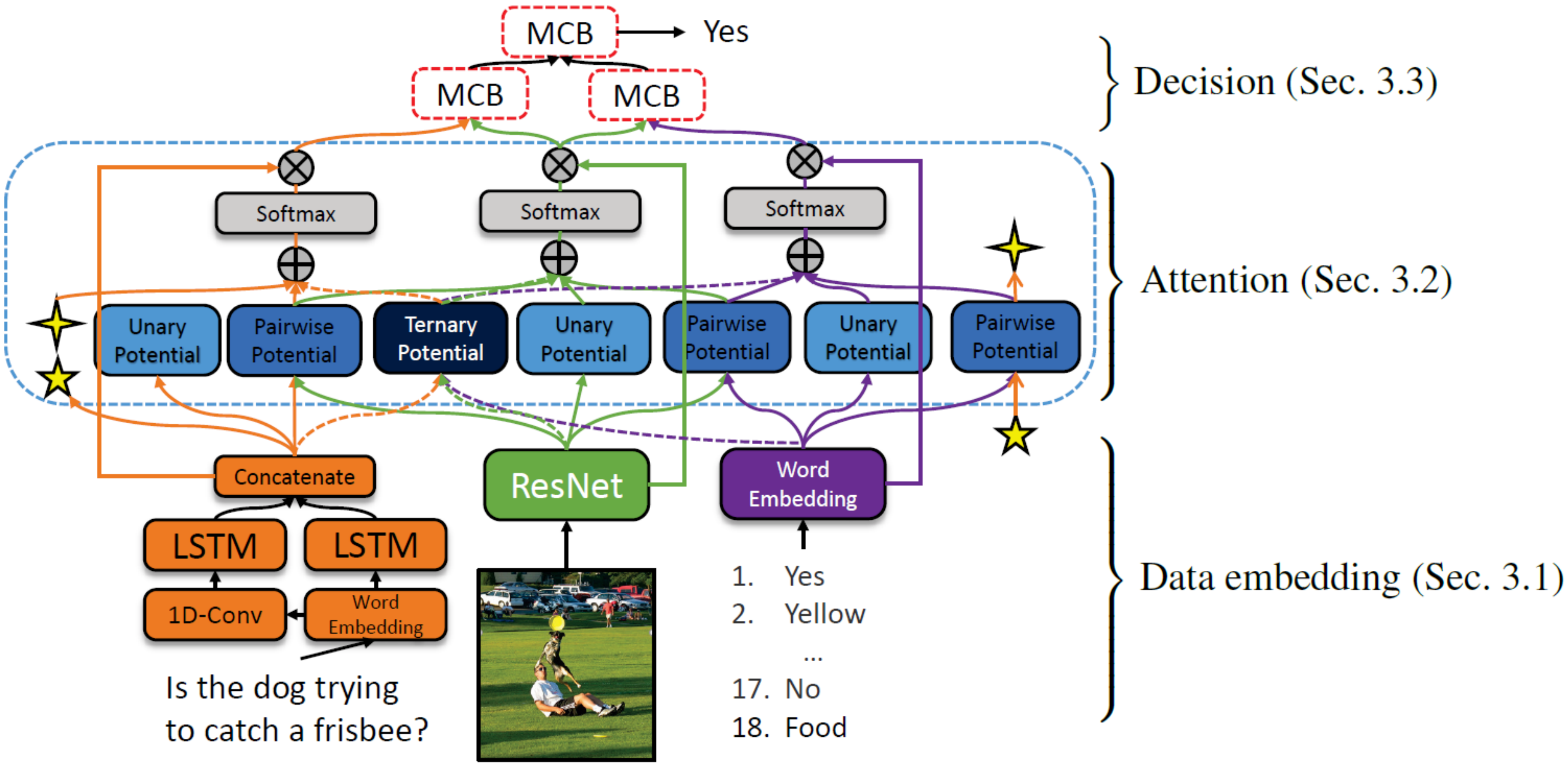


$\mathbf{y}^{\mathrm{R}} = \mathbf{F}_1(\mathbf{x}) + \mathbf{F}_2(\mathbf{x})$ **ORIGINAL** $\mathbf{y}^{\mathrm{R}} = \mathbf{x} + \mathbf{F}(\mathbf{x})$

$\mathbf{y}^{\mathrm{S}} = \mathbf{F}_1(\mathbf{x}) + \mathbf{F}_2(\mathbf{x}) + \mathbf{F}_1(\mathbf{x}) \odot \mathbf{F}_2(\mathbf{x})$ **SORT** $\mathbf{y}^{\mathrm{S}} = \mathbf{x} + \mathbf{F}(\mathbf{x}) + \sqrt{\mathbf{x} \odot \mathbf{F}(\mathbf{x})}$

Y. Wang, L. Xie, C. Liu, Y. Zhang, W. Zhang, A. Yuille, SORT: Second-Order Response Transform for Visual Recognition, ICCV 2017

Note (2): High-Order Attention



I. Schwartz, A. G. Schwing, T. Hazan, High-Order Attention Models for Visual Question Answering, arXiv:1711.04323

Code

- HIHCA
  - https://github.com/cssjcai/hihca

- MLKR
  - https://github.com/Hwang64/MLKP

References

- P. Li, J. Xie, Q. Wang, W. Zuo, Is Second-order Information Helpful for Large-scale Visual Recognition? ICCV 2017.
- F. Wang, W. Zuo, L. Zhang, D. Meng, and D. Zhang, A Kernel Classification Framework for Metric Learning, IEEE T-NNLS, 26(9): 1950 - 1962, 2015.
- X. Wu, W. Zuo, L. Lin, W. Jia, D. Zhang, F-SVM: Combination of Feature Transformation and SVM Learning via Convex Relaxation, IEEE T-NNLS, 29(11): 5185-5199, 2018.
- S. Cai, W. Zuo, L. Zhang, Higher-order Integration of Hierarchical Convolutional Activations for Fine-grained Visual Categorization, ICCV 2017.
- H. Wang, Q. Wang, M. Gao, P. Li, W. Zuo, Multi-scale Location-aware Kernel Representation for Object Detection, CVPR 2018.

Thank you!