Representation Learning with Context-aware Feature Augmentation for Visual Recognition

Sonagyang Zhang
sy.zhangbuaa@gmail.com

School of Information Science and Technology
ShanghaiTech University

MSRA, Beijing, China
2019/07/16

Outline

Outline

What is Visual Scene Context?

Frome *Antonio Torralba*

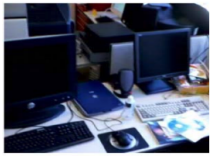
- A specific scene category
  (a coffeemaker is usually in a kitchen)
- The structure of the scene background
  (a chair is on the ground, not the ceiling)
- A combination of objects of shapes
  (TV+sofa+rug+bookshelf = living-room)
- Spatial relationships between shapes

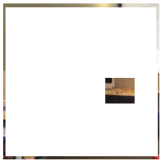
Image Credit: Advance in Computer Vision-MIT

Why is Context Important?[1]

- Changes the interpretation of an object(or its function)

- Reduce the search space.

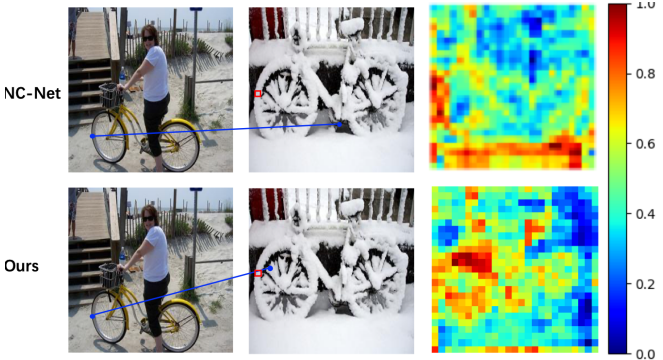




[1] Image Credit: Advance in Computer Vision-MIT

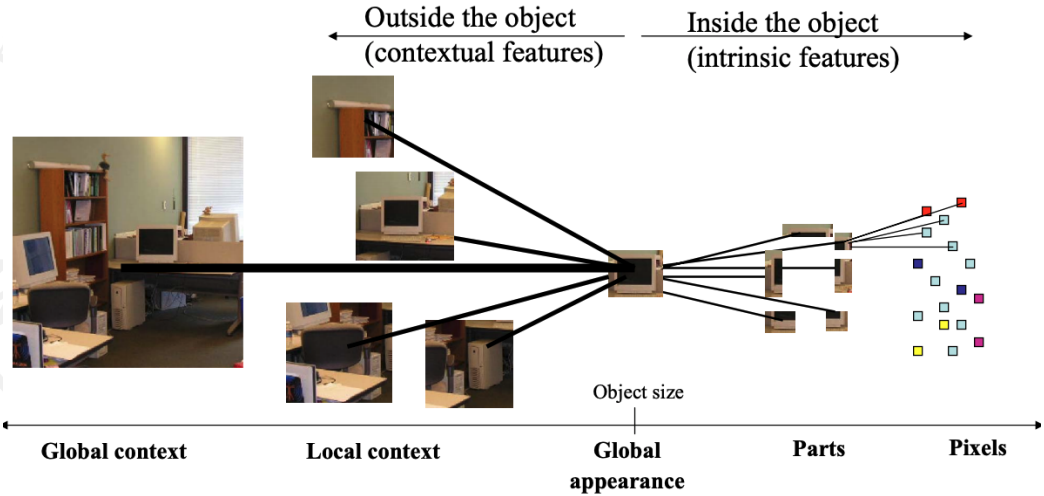
Why is Context Important?[2]

- Reduce the ambiguity.



[2] Image Credit: Dynamic Context Correspondence Network for Semantic Alignment

Looking Outside the Box[3]



[3] Image Credit: Advance in Computer Vision-MIT

Outline

Prior Works(Probabilistic Graphical Models)

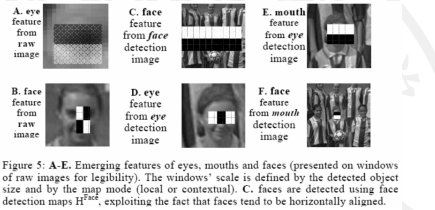
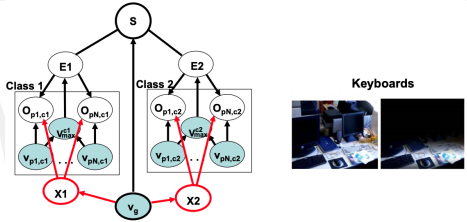
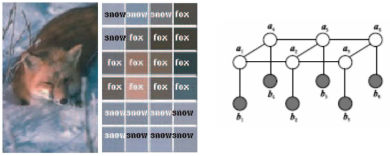


Figure 5: **A-E.** Emerging features of eyes, mouths and faces (presented on windows of raw images for legibility). The windows' scale is defined by the detected object size and by the map mode (local or contextual). **C.** faces are detected using face detection maps H^{Face}, exploiting the fact that faces tend to be horizontally aligned.

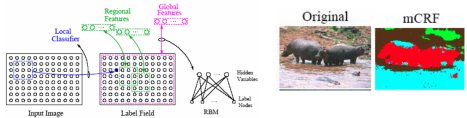
(a) Fink & Perona(03)



(b) Murphy, Torralba & Freeman(03)



(c) Carbonetto, de Freitas & Barnard(04)



(d) He, Zemel & Carreira-Perpinan(04)

Outline

LatentGNN: Learning Efficient Non-local Relations for Visual Recognition

Songyang Zhang, Shipeng Yan, Xuming He

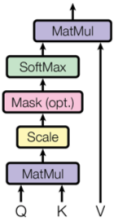
ICML 2019

Goal

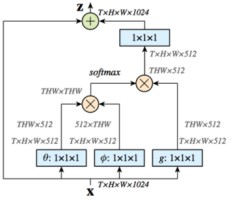
Learning efficient feature augmentation with Non-local relations for visual recognitions.

Motivation

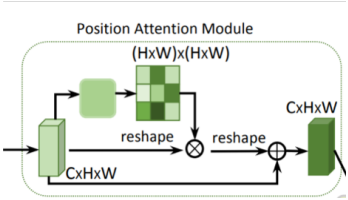
- To model the non-local feature context by a **Graph Neural Network (GNN)**.
  - **Self-attention Mechanism, Non-local network** as special examples of **Graph Neural Network** with truncated inference.
- To reduce the complexity of a fully-connected GNN by introducing a **latent representation**.



Self-attention(Vaswani et al)



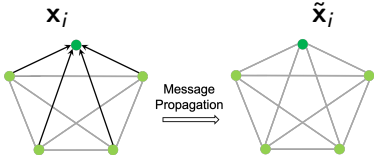
Non-local Network(Wang et al)



Dual Attention Network(Fu et al)

Non-local Features with GNN

- **Input**: Grid/Non-grid Conv-feature,
$$\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^\mathsf{T}, \mathbf{x}_i \in \mathbb{R}^c$$
- **Output**: Context-aware Conv-feature,
$$\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_N]^\mathsf{T}, \tilde{\mathbf{x}}_i \in \mathbb{R}^c$$
- **Each Location**:
$$\tilde{\mathbf{x}}_i = h\left(\frac{1}{Z_i(\mathbf{X})}\sum_{j=1}^{N} g\left(\mathbf{x}_i, \mathbf{x}_j\right)\mathbf{W}^\top \mathbf{x}_j\right) \quad (1)$$
- **Matrix Form**:
$$\tilde{\mathbf{X}} = h\left(\mathbf{A}(\mathbf{X})\mathbf{X}\mathbf{W}\right), \quad \mathbf{X}_{\text{aug}} = \lambda \cdot \tilde{\mathbf{X}} + \mathbf{X} \quad (2)$$
  - $g(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\mathsf{T}\mathbf{x}_j$: Pair-wise relations function
  - h: Element-wise activation function(ReLU)
  - $Z_i(\mathbf{X})$: Normalization factor
  - $\mathbf{W} \in \mathbb{R}^{c \times c}$: Weight matrix of the linear mapping
  - λ: Scaling parameter

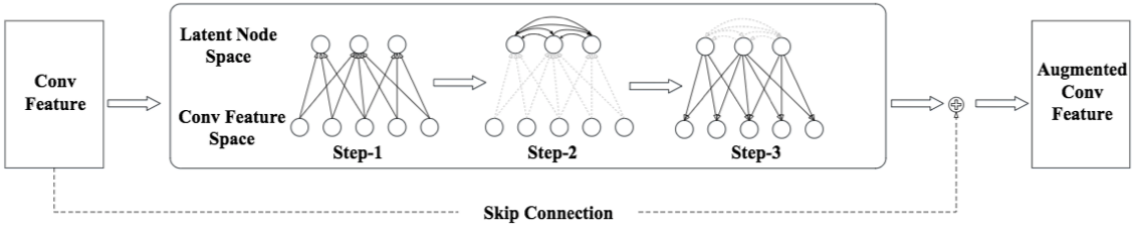


If $N = 500 \times 500$, A requires 500GB of storage!!!

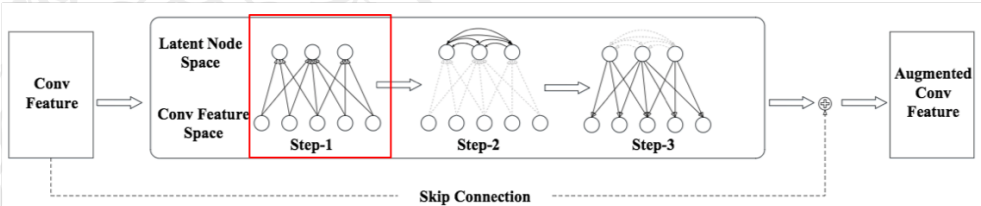
Latent Graph Neural Network

LatentGNN

- **Key Idea**: Introduce a latent space for efficient global context encoding
- Conv-feature Space: $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]^\top, \mathbf{x}_i \in \mathbb{R}^c$
- Latent Space: $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_d]^\top, \mathbf{z}_i \in \mathbb{R}^c, d \ll N$



Latent Graph Neural Network



Step-1: **Visible-to-Latent** Propagation(Bipartite Graph)

- **Each Latent Node:**

$$\mathbf{z}_k = \sum_{j=1}^{N} \frac{1}{m_k(\mathbf{X})} \psi(\mathbf{x}_j, \theta_k) \mathbf{W}^{\mathsf{T}} \mathbf{x}_j, \quad 1 \le k \le d \tag{3}$$

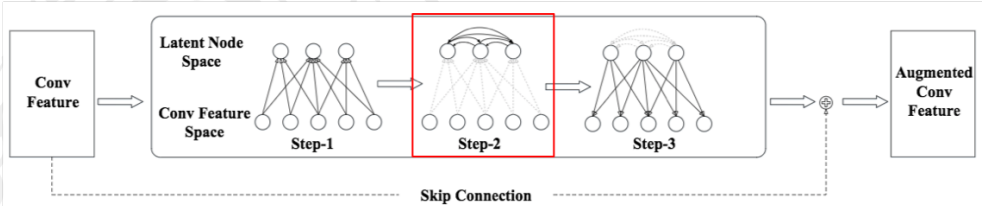
- **Matrix Form**:

$$\mathbf{Z} = \mathbf{\Psi}(\mathbf{X})^{\mathsf{T}} \mathbf{X} \mathbf{W} \tag{4}$$

$$\mathbf{\Psi}(\mathbf{X}) = [\psi(\mathbf{x}_1), \cdots, \psi(\mathbf{x}_N)]^{\mathsf{T}} \in \mathbb{R}^{N \times d}, \quad \psi(\mathbf{x}_i) = [\frac{\psi(\mathbf{x}_i, \theta_1)}{m_1(\mathbf{X})}, \cdots, \frac{\psi(\mathbf{x}_i, \theta_d)}{m_d(\mathbf{X})}]^{\mathsf{T}} \tag{5}$$

- $\psi(\mathbf{x}_j, \theta_k)$: : encode the affinity between node $\mathbf{x}_j$ and node $\mathbf{z}_k$
- $m_k(\mathbf{X})$: the normalization factor

Latent Graph Neural Network



Step-2: **Latent-to-Latent** Propagation(Fully-connected Graph)

- **Each Latent Node:**

$$\tilde{\mathbf{z}}_k = \sum_{j=1}^{d} f(\phi_k, \phi_j, \mathbf{X})\mathbf{z}_j, \quad 1 \le k \le d \tag{6}$$

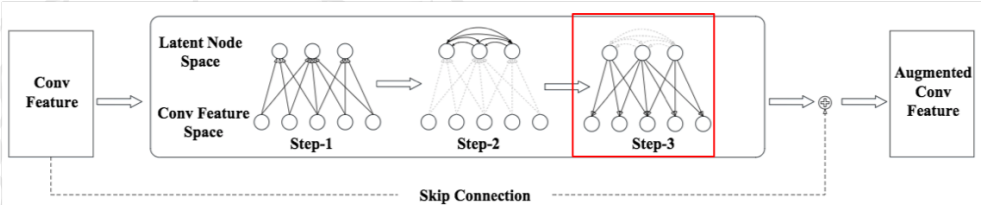
- **Matrix Form**:

$$\mathbf{F}_{\mathbf{X}} = [f(\phi_k, \phi_j, \mathbf{X})]_{d\times d} \tag{7}$$

$$\tilde{\mathbf{Z}} = \mathbf{F}_{\mathbf{X}}\mathbf{Z} \tag{8}$$

  - $f(\phi_k, \phi_j, \mathbf{X})$: data-dependent pair-wise relations between two latent nodes

Latent Graph Neural Network



Step-3: **Latent-to-Visible** Propagation(Bipartite Graph)

- **Each Visible Node:**

$$\tilde{\mathbf{x}}_i = h\left(\sum_{k=1}^{d} \psi(\mathbf{x}_i, \theta_k)\tilde{\mathbf{z}}_k\right), \quad 1 \le i \le N \tag{9}$$

- **Matrix Form**:

$$\tilde{\mathbf{X}} = h\left(\boldsymbol{\Psi}(\mathbf{X})\tilde{\mathbf{Z}}\right) \tag{10}$$

LatentGNN vs. GNN

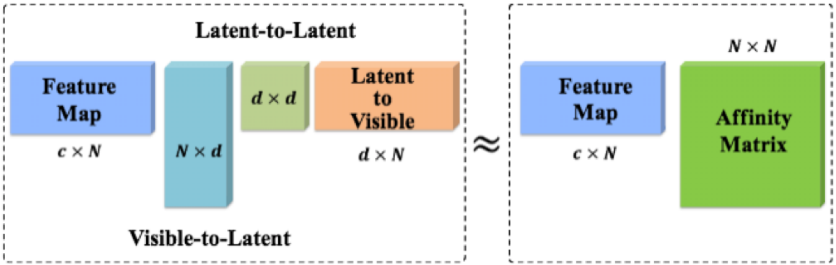
Overall Process

LatentGNN

- $\tilde{\mathbf{X}} = h\left(\mathbf{\Psi}(\mathbf{X})\mathbf{F}_{\mathbf{X}}\mathbf{\Psi}(\mathbf{X})^{\top}\mathbf{X}\mathbf{W}\right)$
- $\mathbf{X}_{\text{aug}} = \lambda \cdot \tilde{\mathbf{X}} + \mathbf{X}$
- $\mathbf{A}(\mathbf{X}) = \mathbf{\Psi}(\mathbf{X})\mathbf{F}_{\mathbf{X}}\mathbf{\Psi}(\mathbf{X})^{\top}$

GNN

- $\tilde{\mathbf{X}} = h\left(\mathbf{A}(\mathbf{X})\mathbf{X}\mathbf{W}\right)$
- $\mathbf{X}_{\text{aug}} = \lambda \cdot \tilde{\mathbf{X}} + \mathbf{X}$
- $\mathbf{A}_{i,j} = \frac{1}{Z_i(\mathbf{X})} g(\mathbf{x}_i, \mathbf{x}_j), \mathbf{A}(\mathbf{X}) \in \mathbb{R}^{N \times N}$



$O(N \cdot d)$ $O(N \cdot N)$

Relations with Recent Works

Relation with GCNet[a]

[a] Yue Cao et al. "GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond". In: *arXiv preprint arXiv:1904.11492* (2019).

GCNet can be considered as the special example of LatentGNN

- Latent space only have one latent node
- No message propagation within the latent space
- Latent-to-visible propagation is implemented in a non-parametric way.

Relation with CCNet[a]

[a] Zilong Huang et al. "CCNet: Criss-Cross Attention for Semantic Segmentation". In: *arXiv preprint arXiv:1811.11721* (2018).

- Propose to build a spare graph for the grid data(image)
- Strong assumption of the message propagation

Experimental Results

Grid Data: Object Detection/Instance Segmentation on MSCOCO

- **+NLBlock**: insert the non-local block in the last stage of the backbone.
- **+LatentGNN**: Integrate LatentGNN with the backbone at different stages.

Model	Stage	Kernels	AP_{box}	AP^{50}_{box}	AP^{75}_{box}	AP_{sem}	AP^{50}_{sem}	AP^{75}_{sem}	FLOPS	#Params
ResNet-50[1]	-	-	38.0	59.6	41.5	34.6	56.4	36.5	-	-
+NL Block[1]	Stage4	1	39.0	61.1	41.9	35.5	58.0	37.4	+10.67G	+ 2.09M
ResNet-50(1x)[2]	-	-	37.8	59.1	41.2	34.2	55.8	36.3	-	-
+ NL Block[2]	Stage4	1	38.7	60.2	42.2	35.0	57.0	37.1	+10.67G	+ 2.09M
+ LatentGNN	Stage3	1	38.2	59.7	41.7	34.7	56.3	36.8	+1.48G	**+ 0.06M**
+ LatentGNN	Stage4	1	39.0	60.7	42.6	35.2	57.6	37.4	+1.11G	+ 0.20M
+ LatentGNN	Stage5	1	38.8	61.0	42.0	35.0	57.6	37.0	**+0.97G**	+ 0.81M
+ LatentGNN	Stage345	1	**39.5**	**61.6**	**43.2**	**35.6**	**58.3**	**37.7**	+3.59G	+1.07M
ResNet-101(1x)	-	-	39.9	61.3	43.8	35.9	58.2	38.1	-	-
+ LatentGNN	Stage4	1	41.0	63.2	45.0	36.9	59.6	39.4	**+1.11G**	**+ 0.20M**
+ LatentGNN	Stage345	1	**41.4**	**63.7**	**45.2**	**37.2**	**60.1**	**39.5**	+3.59G	+1.07M
ResNeXt-101(1x)	-	-	42.1	64.1	45.9	37.8	60.3	39.5	-	-
+ LatentGNN	Stage4	1	43.0	65.3	46.9	38.5	61.9	40.9	**+1.11G**	**+ 0.20M**
+ LatentGNN	Stage345	1	**43.2**	**65.6**	**47.2**	**38.8**	**62.1**	**41.0**	+3.59G	+1.07M

Experimental Results

Grid Data: Ablation Study on MSCOCO

- Effects of different backbone networks.
- A mixture of low-rank matrices.

Model	Stage	Kernels	AP_{box}	AP^{50}_{box}	AP^{75}_{box}	AP_{sem}	AP^{50}_{sem}	AP^{75}_{sem}	FLOPS	#Params
ResNet-50(1x)	-	-	37.8	59.1	41.2	34.2	55.8	36.3	-	-
+LatentGNN	Stage4	1	39.0	60.7	42.6	35.2	57.6	37.4	**+1.11G**	**+ 0.20M**
	Stage4	2	39.0	60.7	42.7	35.3	57.6	37.6	+1.30G	+ 0.29M
	Stage4	3	**39.2**	**61.0**	**42.8**	**35.4**	**57.6**	**37.7**	+1.48G	+0.38M
+LatentGNN	Stage345	1	39.5	61.6	43.2	35.6	58.3	37.7	**+3.59G**	**+1.07M**
	Stage345	3	**39.5**	**61.7**	**43.3**	**35.7**	**58.4**	**37.8**	+5.13G	+1.89M

Non-grid Data: Point Cloud Semantic Segmentation on ScanNet

Model	Kernels	Scale	Pixel Accuracy	Voxel Accuracy	Class Pixel Accuracy	Class Voxel Accuracy	FLOPS	#Params
3DCNN(Dai et al., 2017a)	-	-	-	73.0	-	-	-	-
PointNet(Qi et al., 2017a)	-	-	-	73.9	-	-	-	-
PointCNN(Li et al., 2018)	-	-	85.1	-	-	-	-	
PointNet++(Qi et al., 2017b)	-	Single Scale	81.5	83.2	51.7	53.1	-	-
PointNet++(Qi et al., 2017b)	-	Multi Scale	-	84.5	-	-	-	-
+NL Block	1	Single Scale	82.3	84.0	53.1	54.5	+31M	+0.70M
+LatentGNN	1	Single Scale	82.6	84.2	53.2	54.6	**+15M**	**+0.31M**
+LatentGNN	3	Single Scale	**83.7**	**85.2**	**56.0**	**57.6**	+30M	+0.54M

Issues of LatentGNN

- Lack of the interpretation of the Latent Space.
- Ignore the spatial layout information.
- Size of latent space is sensitive to different tasks.
- Computation cost.

Context-aware Latent Graph Decoder for Dense Visual Recognition

Songyang Zhang, Shipeng Yan, Xuming He

Submitted to NIPS 2019

Goal & Motivation

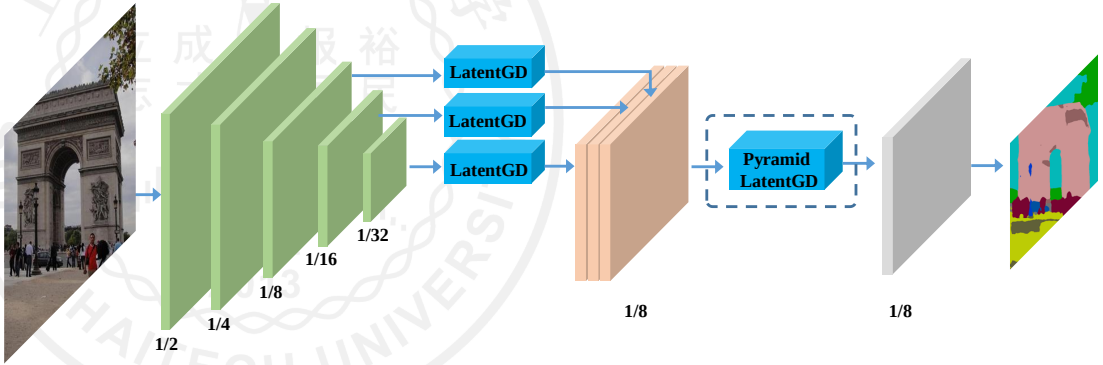
Goal

Efficiently generate a context-aware and pixel-wise representation for dense visual recognition.

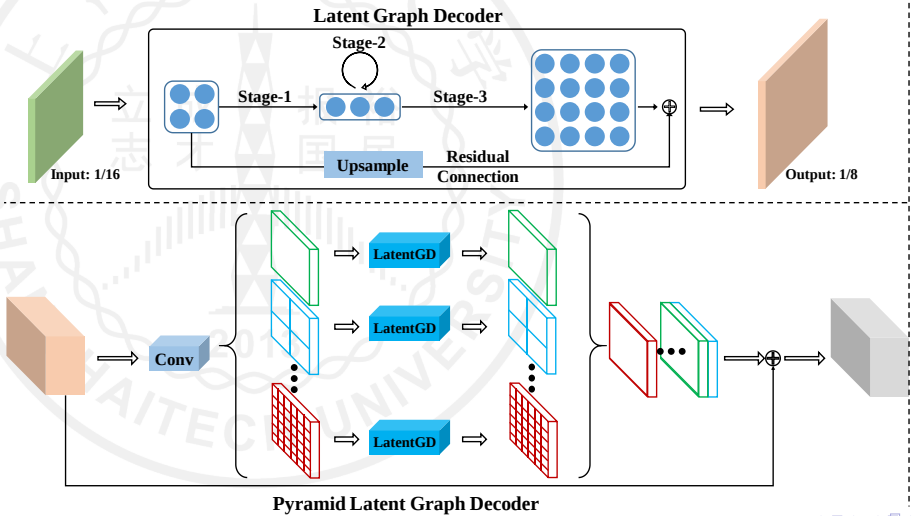
Motivation

- Reduce the heavy computation introduced by the dilated convolution
- Replace the normal decoder with a context-aware decoder.
- Introduce the spatial layout information to enhance the LatentGNN.

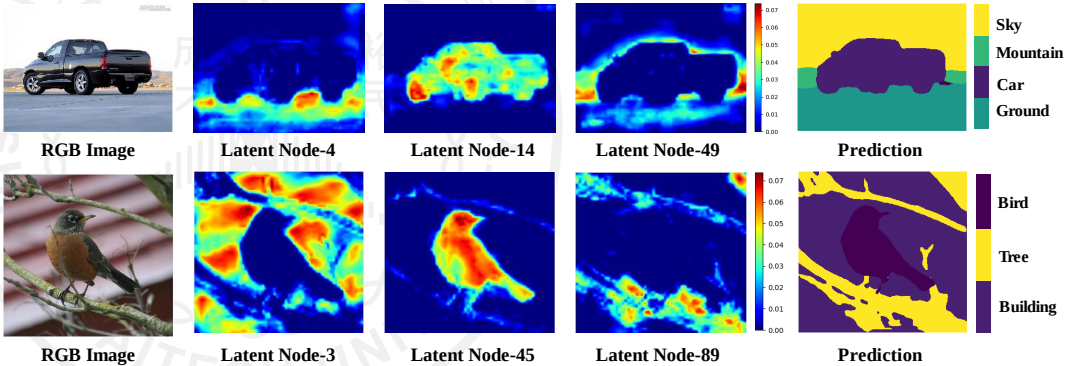
LatentFCN


LatentGD
LatentGD
LatentGD
Pyramid
LatentGD
1/2
1/4
1/8
1/16
1/32
1/8
1/8

LatentFCN


Latent Graph Decoder
Stage-2
Stage-1
Stage-3
Upsample
Residual Connection
Input: 1/16
Output: 1/8
Conv
LatentGD
LatentGD
LatentGD
Pyramid Latent Graph Decoder
RGB Image
Sky
Mountain
Sea
Boat
Prediction
Latent Node-10
Latent Node-35
Latent Node-56

Visualization Results



Outline

Dual Attention Network with Semantic Embedding for Few-shot Learning

Shipeng Yan*, Songyang Zhang*, Xuming He
(* means equal contribution)

AAAI 2019

training data

Braque | Cezanne

test datapoint

By Braque or Cezanne?

Image Credit: Chelsea Finn & Sergey Levine

How did you accomplish this?

Through previous experience.

How Might You Get a Machine to Accomplish This Task?

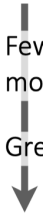
Modeling image formation

Geometry

SIFT features, HOG features + SVM

Fine-tuning from ImageNet features
Domain adaptation from other painters

???



Fewer human priors,
more data-driven priors

Greater success.

Can we explicitly learn priors from previous experience
that lead to efficient downstream learning?

Few-shot Image Classification

Each Task: $T \in \mathcal{T}$

Task-test Image/Label

$$T = (\mathbf{L}, \mathbf{D}^{tr}, \mathbf{x}^{ts}, \mathbf{y}^{ts})$$

Task-train Set(Support Set)

$$D^{tr} = \{(\mathbf{x}^{(1)}, y^{(1)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

Supervised-Learning	Train	Test	One data point
Meta-learning	Meta-training	Meta-testing	One task

Image Credit: Ravi & Larochelle 2017

Problems & Solutions

Problems

- Sensitive the the background clutter
- Difficult to interpret
- Complex architecture
- Slow Convergence

Solutions

- A dual attention architecture
  - Spatial attention to localize the foreground object
  - Task attention to encode the task context
  - Recurrent structure for refinement

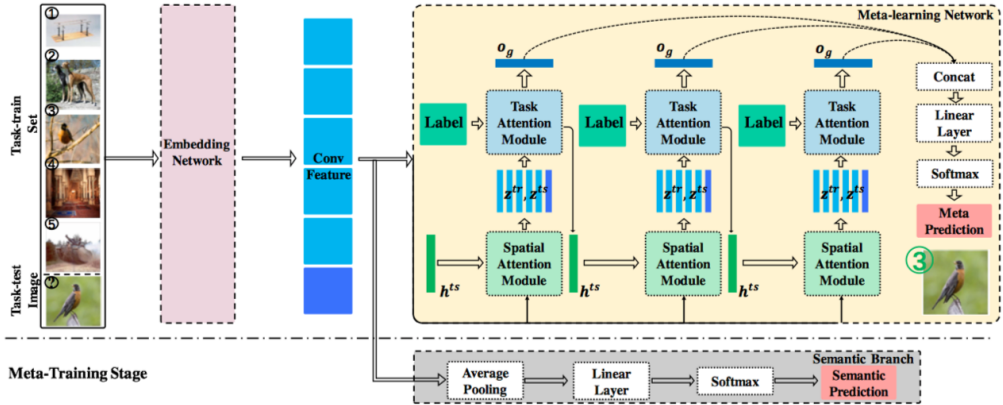
Few-shot Image Classification

z^{tr} : Task–train feature maps

z^{ts} : Task–test feature maps

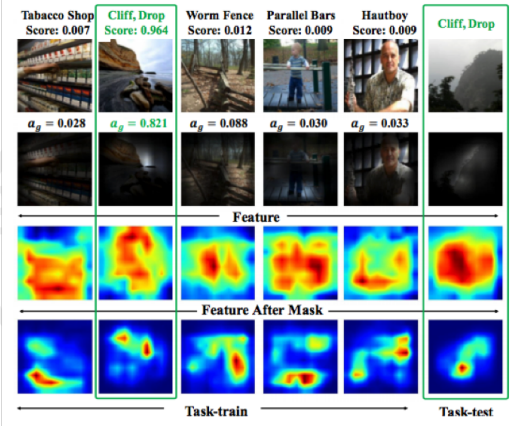
o_g : Task–test label prediction score

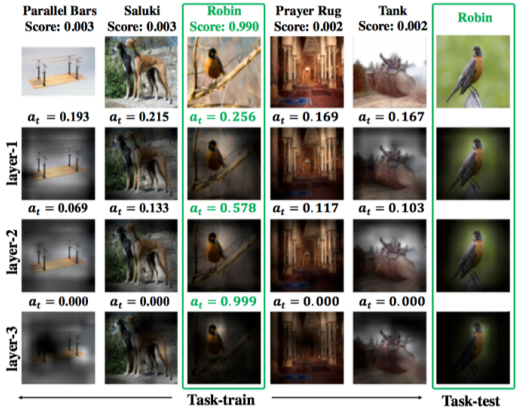
h^{ts} : Task–test spatial aware representation



Framework of Spatial-Task Attention Network

Visualization Results





Dual Attention Modules

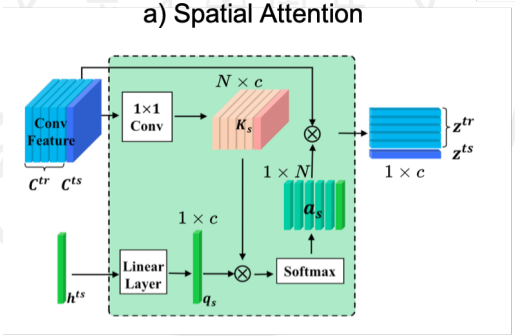
z^{ts} : Task-test feature maps

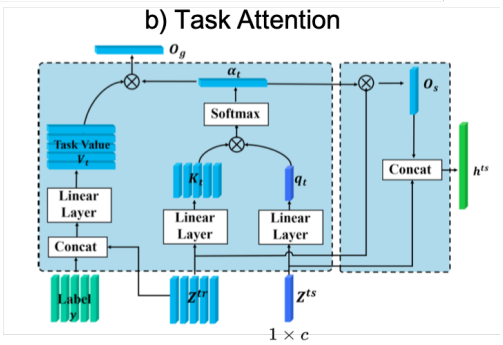
z^{tr} : Task-train feature maps

o_g : Task-test label prediction score

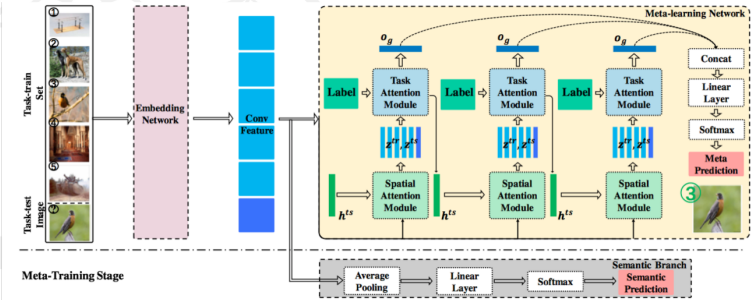
h^{ts} : Task-test spatial aware representation

K : Key q : Query V : Value





Semantic Regularization



$$L_{full} = L_{task}(\Theta) + \lambda L_{sem}(\Theta)$$

Task-train Set(Support Set)

Model Parameters

$$L_{task}(\Theta) = \sum_{T \in \mathcal{S}_{tr}^{meta}} \frac{-\log P_{\text{meta}}(y_T^{ts} | \mathbf{D}_T^{tr}, \mathbf{x}_T^{ts}; \Theta)}{|\mathcal{S}_{tr}^{meta}|}$$

$$L_{sem}(\Theta) = \sum_{T \in \mathcal{S}_{tr}^{meta}} \sum_{(\mathbf{x}_T, y_T)} \frac{-\log P_{\text{sem}}(y_T | \mathbf{x}_T; \Theta)}{|\mathcal{S}_{tr}^{\text{meta}}|(|\mathbf{D}^{tr}|+1)}$$

Task-test Image

Quantitative Results

a) Results on miniImageNet

Method	# Params	Feature Extractor	5-way Accuracy 1-shot	5-way Accuracy 5-shot
Matching Net (Vinyals et al. 2016)	0.1M	Conv64	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
Prototypical Net(Snell et al. 2017)	0.1M	Conv64	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$
MAML (Finn et al. 2017)	0.1M	Conv64	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
RelationNet(Sung et al. 2018)	0.23M	Conv64	$50.44 \pm 0.82\%$	$65.32 \pm 0.70\%$
(Gidaris et al. 2018)	0.24M	Conv64	$56.20 \pm 0.86\%$	$72.81 \pm 0.62\%$
GNN (Satorras et al. 2018)	1.6M	Conv64	$50.33 \pm 0.36\%$	$66.41 \pm 0.63\%$
STANet-S(1-Layer)	0.24M	Conv64	$50.38 \pm 0.65\%$	$65.67 \pm 0.66\%$
STANet-S(3-Layer)	0.24M	Conv64	$\mathbf{53.11 \pm 0.60\%}$	$\mathbf{67.16 \pm 0.66\%}$
SNAIL (Mishra et al. 2018)	6.1M	ResNet-12	$55.71 \pm 0.99\%$	$68.88 \pm 0.92\%$
(Gidaris et al. 2018)	2.6M	ResNet-12	$55.45 \pm 0.86\%$	$70.13 \pm 0.68\%$
(Qiao et al. 2018)	40.5M	WRN-28	$\mathbf{59.60 \pm 0.41\%}$	$\mathbf{73.74 \pm 0.19\%}$
STANet(1-Layer)	2.6M	ResNet-12	$57.25 \pm 0.40\%$	$69.45 \pm 0.50\%$
STANet(3-Layer)	2.6M	ResNet-12	$\mathbf{58.35 \pm 0.57\%}$	$\mathbf{71.07 \pm 0.39\%}$

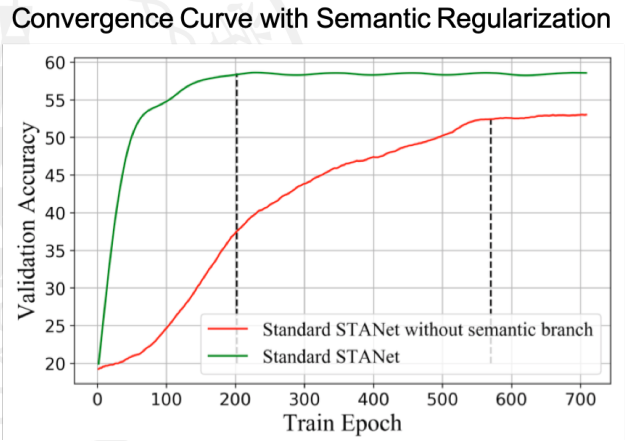
STANet-S: Shallow embedding network

b) Ablation Study

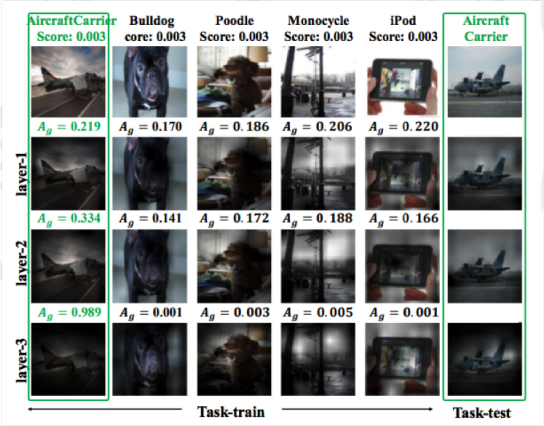
Components			5-way(Normal)	
SR.	SA.	TA.	1-shot	5-shot
✗	Uniform	✓	$53.41 \pm 0.61\%$	$64.32 \pm 0.57\%$
✗	Gaussian	✓	$54.29 \pm 0.66\%$	$65.41 \pm 0.55\%$
✗	✓	✓	$55.52 \pm 0.64\%$	$66.75 \pm 0.62\%$
✓	✓	✓	$\mathbf{58.35 \pm 0.57\%}$	$\mathbf{71.07 \pm 0.39\%}$

SR: Semantic Regularization
SA: Spatial Attention
TA: Task Attention

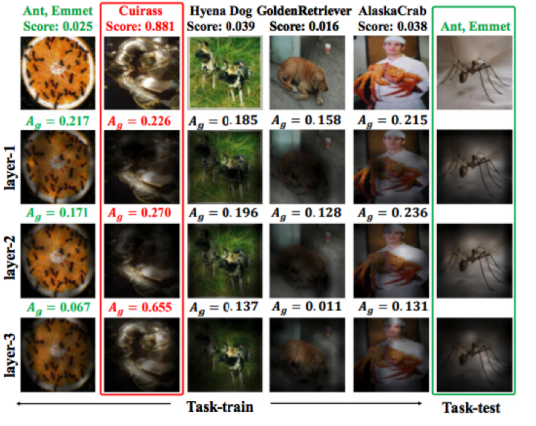
Quantitative Results



Visualization Results



Failure Case:
Large Difference in Scale



Outline

Context Modeling with GNN

How to design an appropriate graph to model the context?

- Graphical model and latent variable model
- Dense(Non-local) or sparse(CCNet)
- Efficient message propagation(LatentGNN)
- Kernel function to measure the pair-wise relation

Efficient and effective GNN for context modeling

- One node v.s. all nodes
- Adaptive latent space for different tasks
- Balance the complexity and effectiveness

Outline

Feature Augmentation Module to Basic Operator

Why we need new basic operator

- In order to use X, X is used?
- Faults of the standard convolution?
- Difference between the novel operator with the standard convolution
  - Pixel-adaptive CNN[a]
  - Stand-alone Self-attention[b]
  - Local Relation Network[c]
- A uniform framework for grid/non-grid data?

[a] Hang Su et al. "Pixel-Adaptive Convolutional Neural Networks". In: *Proceedings of CVPR*. 2019.

[b] Irwan Bello et al. "Attention augmented convolutional networks". In: *arXiv preprint arXiv:1904.09925* (2019).

[c] Han Hu et al. "Local relation networks for image recognition". In: *arXiv preprint arXiv:1904.11491* (2019).

Thanks!