Local Relation Networks for Image Recognition

Han Hu[1] Zheng Zhang[1] Zhenda Xie[1,2*] Stephen Lin[1]
[1]Microsoft Research Asia [2]Tsinghua University
{hanhu, zhez, v-zhxia, stevelin}@microsoft.com

Abstract

The convolution layer has been the dominant feature extractor in computer vision for years. However, the spatial aggregation in convolution is basically a pattern matching process that applies fixed filters which are inefficient at modeling visual elements with varying spatial distributions. This paper presents a new image feature extractor, called the local relation layer, that adaptively determines aggregation weights based on the compositional relationship of local pixel pairs. With this relational approach, it can composite visual elements into higher-level entities in a more efficient manner that benefits semantic inference. A network built with local relation layers, called the Local Relation Network (LR-Net), is found to provide greater modeling capacity than its counterpart built with regular convolution on large-scale recognition tasks such as ImageNet classification.

1. Introduction

Humans have a remarkable ability to "see the infinite world with finite means" [27, 2]. From perceiving a limited set of low-level visual primitives, they can productively compose unlimited higher-level visual concepts, from which an understanding of a viewed scene can be formed.

In computer vision, this compositional behavior may be approximated by the building of hierarchical representations in a convolutional neural network, where different layers represent different levels of visual elements. At lower layers, basic elements such as edges are extracted. These are combined at middle layers to form object parts, and then finally at higher layers, whole objects are represented [35].

Although a series of convolutional layers can construct a hierarchical representation, its mechanism for composing lower-level elements into higher-level entities can be viewed as highly inefficient in regards to conceptual inference. Rather than recognizing how elements can be meaningfully joined together, convolutional layers act as templates, where input features are spatially aggregated according to convolutional filter weights. For an effective composition of features, suitable filters would need to be learned and applied. This requirement is problematic when trying to infer visual concepts that have significant spatial variability, such as from geometric deformation as illustrated in Fig. 1, since filter learning could potentially face a combinatorial explosion of different valid compositional possibilities [24, 34, 23].

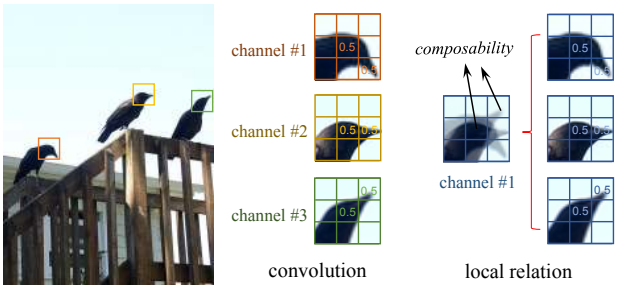


Figure 1. Illustration of the 3×3 convolution layer and the 3×3 local relation layer. While 3 channels are required by convolution to represent the spatial variability between bird eye and beak, the local relation layer requires only 1 channel.

In this paper, we present a new computational network layer, called the local relation layer, in which meaningful compositional structure can be adaptively inferred among visual elements in a local area. In contrast to convolution layers which employ fixed aggregation weights over spatially neighboring input features, our new layer adapts the aggregation weights based on the composability of local pixel pairs. Inspired by recent works on relation modeling [1], composability is determined by the similarity of two pixels' feature projections into a learned embedding space. This embedding may additionally account for geometric priors, which have proven to be useful in visual recognition tasks[1]. By learning how to adaptively compose

*This work was done when Zhenda Xie was interns at Microsoft Research Asia.

[1]For example, geometric priors are intrinsically encoded in the convolution layer, as its aggregation weights are parameterized on relative positions. This is an important property leading to its success in visual recognition.

pixels in a local area, a more effective and efficient compositional hierarchy can be built.

Local relation layers can be used as a direct replacement of convolutional layers[2] in deep networks, with little added overhead. Using these layers, we have developed a network architecture called Local Relation Network (LR-Net) that follows the practice in ResNet [9] of stacking residual blocks to enable optimization of very deep networks. Given the same computation budget, LR-Net with 26 layers and bottleneck residual blocks surpasses the regular 26-layer ResNet by an absolute 3% in top-1 accuracy on the ImageNet image classification task [7]. Improved accuracy is also achieved with basic residual blocks and on deeper networks (50 and 101 layers).

Besides strong image classification performance, we demonstrate several favorable properties of local relation networks. One of them is their greater effectiveness in utilizing large kernel neighborhoods compared to regular convolution networks. While regular ConvNets mainly employ 3×3 kernels due to saturation at larger sizes, LR-Net is found to benefit from kernels of 7×7 or even larger. We additionally show that the network is more robust to adversarial attacks, likely due to its compositional power in the spatial domain.

We note that while deep neural networks all form a bottom-up hierarchy of image features, they generally aggregate feature based on static convolution weights, which can be regarded as a top-down manner. By contrast, our compositional approach computes the weights adaptively based on composability of local pixels pairs, referred to as a bottom-up manner. There exist a few recent methods [24, 10, 28] that also do so, but they are either not applicable to large-scale recognition tasks [24, 10] or act in only a complementary role to regular convolution, rather than as a replacement [28]. Moreover, these methods do spatial aggregation over the whole input feature map and do not consider geometric relationships between pixels, while our network demonstrates the importance of *locality* and *geometric priors*. With this work, it is shown that a bottom-up approach to determining feature aggregation weights can be both practical and effective.

2. Related Works

Convolution Layers and Extensions The convolution layer has existed for several decades [8, 18]. Its recent popularity started with the impressive performance of AlexNet [17] in classifying objects on ImageNet [7]. Since then, the convolution layer has been almost exclusively used in extracting basic visual features.

[2]Since 1×1 convolutions do not involve filtering over neighboring pixels, we do not treat them as convolutions in this paper and refer to them as channel transformations [20]. Nevertheless, in some figures/tables, we use 1×1 to denote a channel transformation layer for notation convenience.

Extensions to the regular convolution layer have been proposed. In one direction, a better accuracy-efficiency tradeoff is obtained by limiting the scope of aggregated input channels. Representative works include group convolution [17, 31] and depthwise convolution [5, 11]. Another direction is to modify the spatial scope for aggregation. This has been done to enlarge the receptive field, such as through atrous/dilated convolution [4, 33], and to enhance the ability to model geometric deformation, via active [14] and deformable convolution [6, 36].

Some works relax the requirement of sharing aggregation weights/scopes across positions. A straightforward approach is taken with the locally connected layer [25], which learns independent aggregation weights for different positions. Its application is limited due to the loss of important properties from regular convolution, including translation invariance and knowledge transfer from one position to others. In other works along this direction, convolution layers are proposed which generate position-adaptive aggregation weights [15] or an adaptive aggregation scope [6, 36].

We note that regular convolution and the above extensions all operate in a top-down manner, determining their convolution behavior based on image appearance or spatial positions within a receptive field. In contrast, the proposed layer determines aggregation weights in a bottom-up fashion based on composability of local pixel pairs, which we believe provides a more efficient encoding of spatial composition in the visual world. At the same time, the proposed layer follows and adapts several favorable design principles from these convolution variants, such as *locality*, use of *geometric priors*, and weight/meta-weight sharing across positions, which have been found to be crucial in effectively extracting visual features.

Capsule Networks To address some shortcomings of convolution layers, there have been recent works that determine the aggregation weights in a bottom-up manner based on the composability of pixel pairs. A representative work is Capsule Networks [24, 10], in which composability is computed by an iterative routing process. In each routing step, the aggregation weights are enlarged if the vectors before and after aggregation are close to each other, and they are reduced otherwise. This self-strengthening process in capsule networks is similar to the process of a *filtering bubble*, a popular phenomenon in social networks where the connection between agents with the same interests becomes stronger, while the connections become weaker when interests are dissimilar.

Although the routing method is inspiring, the computation is not well aligned with current learning infrastructures such as back-propagation and multi-layer networks. In contrast, the composability of pixel pairs in the local relation layer is computed by the similarity of pixel pairs in an em-

bedding space with learnt embedding parameters, which is more friendly to the current learning infrastructure. The local relation layer is also differentiated from capsule networks by its aggregation computation process, including its spatial scope (*local* vs. *global*) and geometric priors (*with vs. without*). With these differences, local relation networks are significantly more practical than existing methods based on bottom-up aggregation.

Self-Attention / Graph Networks The proposed local relation layer is also related to self-attention models [26] used in natural language processing, and to graph networks applied on non-grid data [3]. These works share a basic structure similar to general relation modeling [1], which naturally introduces compositionality in the networks.

Due mainly to their powerful composition modeling ability, these methods have become the dominant approaches in their respective fields. However, in computer vision, there are few works involving such compositionality in their network architecture [12, 28, 32, 19, 29, 13, 30] . In [12], relationships between object proposals are modeled, which leads to improved accuracy as well as the first fully end-to-end object detector. The relation modeling in that work is applied to non-grid data. In [28], relationships are modeled between pixels, as in our work. However, the goal is different. [28] extracts long-range context as complementary to the convolution layer, we pursue a basic image feature extractor with more representation power for spatial composition than the convolution layer. In [13], the channel-wise attention are explored and [30] further integrated the channel-wise attention and spatial-wise attention.

In this sense, our work bridges the general philosophy of introducing compositionality into representation, which has proven effective in processing sequential and non-grid data, and applicability as a basic feature extractor for computer vision. Such a goal is non-trivial and requires adaptations from both sides.

3. A General Formulation

In this section, we describe a general formulation for basic image feature extractors, based on which the proposed local relation layer will be presented. Denote the input and output of a layer by $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{y} \in \mathbb{R}^{C'\times H'\times W'}$, with C, C' being the channels of input/output features and H, W, H', W' the input/output spatial resolution. Existing basic image extraction layers generally produce the output feature by a weighted aggregation of input features,

$$\mathbf{y}(c', \mathbf{p}') = \sum_{c\in\Omega_{c'}, \mathbf{p}\in\Omega_{\mathbf{p}'}} \omega(c', c, \mathbf{p}', \mathbf{p}) \cdot \mathbf{x}(c, \mathbf{p}), \qquad (1)$$

where c, c' and $\mathbf{p} = (h, w), \mathbf{p}' = (h', w')$ index the input and output channels and feature map positions, respectively; $\Omega_{c'}$ and $\Omega_{\mathbf{p}'}$ denote the scope for channel and spatial aggregation of input features in producing the output feature value at channel c' and position $\mathbf{p}'$, respectively; $\omega(c', c, \mathbf{p}', \mathbf{p})$ denotes the aggregation weight from $c, \mathbf{p}$ to $c', \mathbf{p}'$. Existing basic image feature extraction layers are differentiated mainly by three aspects: parameterization method, aggregation scope, and aggregation weights.

Parameterization method defines the model weights to be learnt. The most common parameterization method is to directly learn the aggregation weights ω [18]. There are also some methods that learn a meta-network $\{\theta\}$ on input features to generate adaptive aggregation weights [15] or an adaptive aggregation scope across spatial positions [6], or learn a fixed prior about spatial aggregation scope (Ω) [14]. In general, the parameterization is shared across spatial position to enable translation invariance.

Aggregation scope defines the range of channels and spatial positions involved in aggregation computation. For channel scope, regular convolution includes all input channels in computing each channel output. For greater efficiency, some methods consider only one or a group of input channels in producing one channel of the output feature [17, 5]. Recently, there have been methods where multiple or all output channels share the same aggregation weights [28, 24]. For spatial scope, most methods constrain the aggregation computation in a local area. Restricting aggregation to a local area can not only significantly reduce computation but also help introduce an information bottleneck that facilitates learning of visual patterns. Nevertheless, recent non-convolution methods [28, 24] mostly adopt a full spatial scope for aggregation computation.

Aggregation weights are typically learned as network parameters or are computed from them. Almost all variants of convolution obtain their aggregation weights in a *top-down* manner, where they are either fixed across positions or determined by a meta-network on the input features at the position. There are also some non-convolution methods [28, 24] that compute the aggregation weights in a *bottom-up* fashion, with the weights determined by the composability of a pixel pair. In contrast to convolution variants whose aggregation weights depend heavily on geometric priors, such priors are seldom used in recent non-convolution methods.

Table 1 presents a summary of existing basic image feature extractors.

4. Local Relation Layer

In this section, we introduce the local relation layer. Expressed within the general formulation of Eqn. (1), its ag-

Table 1. A summary of basic image feature extractors. The "parameterization" column indicates the model weights to be learnt. The symbols $\omega, \{\theta\}, \Omega$ denote aggregation weights, weights of meta-networks, and spatial sampling points, respectively. "share" indicates whether the parameterized weights are shared across position. The aggregation scope is given over both the channel and spatial domains. The "aggregation weight" column covers three aspects: how aggregation weights are computed from parameterized weights ("computation" sub-column); inclusion of geometric priors ("geo." sub-column); type of computation ("type" sub-column).

method		parameterization		aggregation scope		aggregation weight		
		param.	share	channel (in/out/share)	spatial	computation	geo.	type
conv.	regular	ω	✓	all/one/no	local	ω	✓	top-down
	group [17, 31]	ω	✓	group/one/no	local	ω	✓	top-down
	depthwise [5, 11]	ω	✓	one/one/no	local	ω	✓	top-down
	dilated [4, 33]	ω	✓	all/one/no	atrous	ω	✓	top-down
	active [14]	ω, Ω	✓	all/one/no	Ω	ω	✓	top-down
	local connected [25]	ω	✗	all/one/no	local	ω	✓	top-down
	dynamic filters [15]	θ	✓	all/one/no	local	$f_\theta(\mathbf{x}_{\mathbf{p}'})$	✓	top-down
	deformable [6, 36]	ω, θ	✓	all/one/no	$\Omega(\theta)$	ω	✓	top-down
	non-local [28]	θ_k, θ_q	✓	one/one/all	full	$\Phi(f_{\theta_q}(\mathbf{x}_{\mathbf{p}'}), f_{\theta_k}(\mathbf{x}_{\mathbf{p}}))$	✗	bottom-up
	capsule [24, 10]	θ	✗	one/one/group	full	$\text{route}(\mathbf{y}_{\mathbf{p}'}, f_\theta(\mathbf{x}_{\mathbf{p}}))$	✗	bottom-up
	local relation (our)	$\theta_k, \theta_q, \theta_g$	✓	one/one/group	local	$\text{softmax}_\Omega(\Phi(f_{\theta_q}(\mathbf{x}_{\mathbf{p}'}), f_{\theta_k}(\mathbf{x}_{\mathbf{p}})) + f_{\theta_g}(\mathbf{p}-\mathbf{p}'))$	✓	bottom-up

gregation weights are defined as[3]

$$\omega(\mathbf{p}', \mathbf{p}) = \text{softmax}(\Phi(f_{\theta_q}(\mathbf{x}_{\mathbf{p}'}), f_{\theta_k}(\mathbf{x}_{\mathbf{p}})) + f_{\theta_g}(\mathbf{p}-\mathbf{p}')), \quad (2)$$

where the term $\Phi(f_{\theta_q}(\mathbf{x}_{\mathbf{p}'}), f_{\theta_k}(\mathbf{x}_{\mathbf{p}}))$ is a measure of composability between the target pixel $\mathbf{p}'$ and a pixel $\mathbf{p}$ within its position scope, based on their appearance after transformations f_{θ_q} and f_{θ_k}, following recent works on relation modeling [1]. The term $f_{\theta_g}(\mathbf{p}-\mathbf{p}')$ defines the composability of a pixel pair $(\mathbf{p}, \mathbf{p}')$ based on a geometric prior. The geometric term adopts the relative position as input and is translationally invariant.

This new layer belongs to the class of bottom-up methods, as indicated in Table 1, as it determines composability based on the properties of the two visual elements. In the following, we present its design and discuss its differences from existing bottom-up methods. These differences lead to significantly higher accuracy on image recognition benchmarks. Its performance also is comparable to or surpasses state-of-the-art top-down convolution methods.

Locality The bottom-up methods typically aggregate input features from over the full image. In contrast, the local relation layer limits the aggregation computation to a local area, e.g., a 7×7 neighborhood. We find that constraining the aggregation scope to a local neighborhood is crucial for feature learning in visual recognition (see Table 3).

Compared with the convolution variants which also constrain the aggregation computation to a spatial neighborhood, the local relation layer proves more effective in utilizing larger kernels. While convolution variants usually exhibit performance saturation with neighborhoods larger than 3×3, the local relation layer yields steady improvements in accuracy when increasing the neighborhood size from 3×3 to 7×7 (see Table 3). This difference may be due to the representation power of convolution layer being bottlenecked by the number of fixed filters, hence there is no benefit from a larger kernel size. In contrast, the local relation layer composes local pixel pairs in a flexible bottom-up manner that allows it to effectively model visual patterns of increasing size and complexity. We use a 7×7 kernel size by default.

Appearance composability We follow a general approach for relation modeling [1] to compute appearance composability $\Phi(f_{\theta_q}(\mathbf{x}_{\mathbf{p}'}), f_{\theta_k}(\mathbf{x}_{\mathbf{p}}))$, where $\mathbf{x}_{\mathbf{p}'}$ and $\mathbf{x}_{\mathbf{p}}$ are projected to a *query* (by a channel transformation layer f_{θ_q}) and *key* (by a channel transformation layer f_{θ_k}) embedding space, respectively. While in previous works the *query* and *key* are vectors, in the local relation layer, we use scalars to represent them so that the computation and representation are lightweight. We find that scalars work also well and have better speed-accuracy trade-off compared to vectors (see Table 4).

We consider the following instantiations of function Φ, which we later show to work similarly well (see Table 6):

a) squared difference:

$$\Phi(q_{\mathbf{p}'}, k_{\mathbf{p}}) = -(q_{\mathbf{p}'} - k_{\mathbf{p}})^2. \quad (3)$$

b) absolute difference:

$$\Phi(q_{\mathbf{p}'}, k_{\mathbf{p}}) = -|q_{\mathbf{p}'} - k_{\mathbf{p}}|, \quad (4)$$

c) multiplication:

$$\Phi(q_{\mathbf{p}'}, k_{\mathbf{p}}) = q_{\mathbf{p}'} \cdot k_{\mathbf{p}}, \quad (5)$$

We use Eqn. (3) by default.

[3] Since one output channel strictly uses one input channel in aggregation computation, we omit the c, c' for notational convenience.

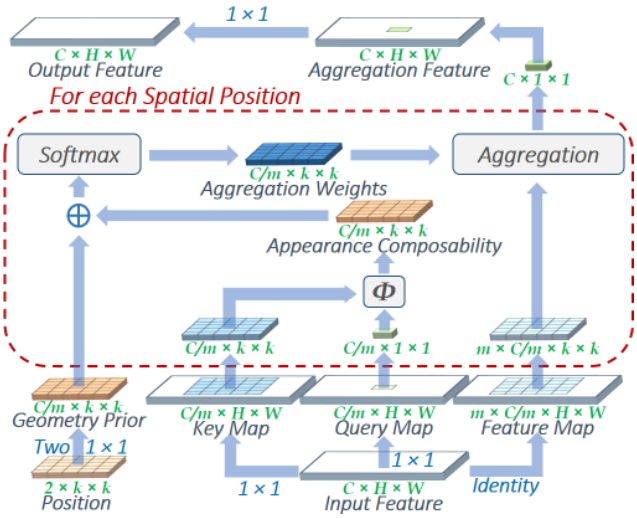


Figure 2. The local relation layer.

Geometric priors Another important aspect differentiating the local relation layer from other bottom-up methods is the inclusion of geometric priors.

The geometric prior is encoded by a small network on the relative position of $\mathbf{p}$ to $\mathbf{p}'$. The small network consists of two-channel transformation layers, with a ReLU activation in between. We find that using a small network to compute the geometric prior values is better than directly learning the values, especially when the neighborhood size is large (see Table 3). This is possible because a small network on relative position treats relative positions as vectors in metric space, while the direct method treats different relative positions as independent identities.

Note that the inference process with using a small network is the same as that of directly learning the geometric priors. In fact, during inference, the fixed learnt weights θ_g will induce fixed geometric prior values $f_{\theta_g}(\Delta\mathbf{p})$ for a relative position $\Delta\mathbf{p}$. We use these fixed geometric prior values instead of the original model weights θ_g for more convenient inference.

Weight normalization We use SoftMax normalization over the spatial scope Ω to compute the final aggregation weights. Such normalization is found to be crucial in balancing the contributions of the appearance composability and geometric prior terms (see Table 6).

Channel sharing Following [24], the local relation layer uses *channel sharing* in aggregation computation, where multiple channels share the same aggregation weights. *Channel sharing* can generally reduce the model size and facilitate GPU memory scheduling for efficient implementation. We observe no accuracy drop with up to 8 channels (default) sharing the same aggregation (see Table 5), while achieving more than 3× actual speed-up than that of 1 channel per aggregation in our CUDA kernel implementation.

Complexity and implementation The local relation layer is summarized in Figure 2. Given an $H \times W$ input feature map, $k \times k$ spatial neighborhood, C channels, and m channels per aggregation computation, the total computational complexity (in FLOPs) of a local relation layer with stride s is

$$\mathcal{C} = \mathcal{O}\left((\frac{1+s^2}{m}+1)C(C+k^2)\frac{HW}{s^2}\right). \quad (6)$$

In our experiments, a naive implementation by a CUDA kernel is used, which is several times slower than regular convolution with the same FLOPs[4]. Note that convolution has a highly optimized implementation with careful memory scheduling. Optimization of memory scheduling for the local relation layer will be a focus of our future work.

5. Local Relation Networks

Local relation layers can be used to replace spatial convolution layers in deep neural networks. In this section, we describe layer replacement in the ResNet architecture [9], where residual blocks with the same topology are stacked.

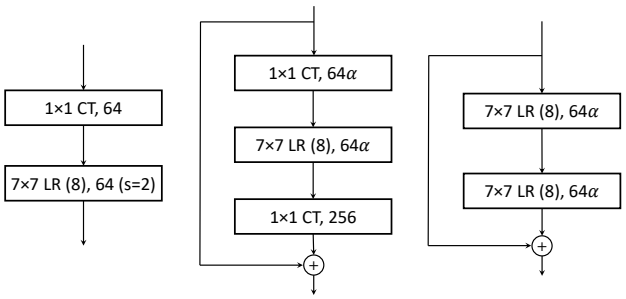
Figure 3 illustrates the replacement of the 3×3 convolution layer in the bottleneck/basic residual blocks and the first 7×7 convolution layer in ResNet. For residual blocks, we keep the FLOPs the same by adopting the expansion ratio (α) of the layer to be replaced. For the first 7×7 convolution layer, we transform the $3 \times H \times W$ input to a feature map of $64 \times H \times W$ by a channel transformation layer and follow this with a 7×7 local relation layer. The replacement of the 7×7 convolution layer consumes similar FLOPs and has comparable accuracy on ImageNet recognition. In the experiments, we will mainly ablate the effects of replacing 3×3 convolution layers in residual blocks.

After replacing all convolution layers in ResNet, we obtain a network which we call the Local Relation Network (LR-Net). Table 2 shows a comparison of ResNet-50 and LR-Net-50 (with default hyper-parameters of 7×7 kernel size and $m = 8$ channels per aggregation). LR-Net-50 uses similar FLOPs but has a slightly smaller model size because of its *channel sharing* in aggregation.

6. Experiments

We perform an ablation study on the ImageNet-1K image classification task. To facilitate the study given limited GPU resources, we conduct the study using LR-Net-26, which is a 26 layer local relation network adapted from ResNet-26. The networks have 8 bottleneck residual

[4]The LR-Net-26 network introduced in Section 5 is about 3× slower than that of a regular ResNet-26 model on a Titan Xp GPU.



(a) entry layers (b) bottleneck block (c) basic block

Figure 3. Illustration of replacing the first 7×7 convolution layer (a) and the bottleneck/basic residual blocks (b)(c) in the ResNet architecture. "CT" denotes the channel transformation layer and "LR" denotes the local relation layer. "7×7 (8), 64" represents kernel size of 7×7, channel sharing of $m = 8$ and output channel of 64. "$s = 2$" represents a stride of 2. All layers are followed by a batch normalization layer and a ReLU activation layer.

stage	output	ResNet-50	**LR-Net-50 (7×7, m=8)**
res1	112×112	7×7 conv, 64, stride 2	**1×1, 64** **7×7 LR, 64, stride 2**
res2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		[1×1, 64 3×3 conv, 64 1×1, 256] ×3	[1×1, 100 **7×7 LR, 100** 1×1, 256] ×3
res3	28×28	[1×1, 128 3×3 conv, 128 1×1, 512] ×4	[1×1, 200 **7×7 LR, 200** 1×1, 512] ×4
res4	14×14	[1×1, 256 3×3 conv, 256 1×1, 1024] ×6	[1×1, 400 **7×7 LR, 400** 1×1, 1024] ×6
res5	7×7	[1×1, 512 3×3 conv, 512 1×1, 2048] ×3	[1×1, 800 **7×7 LR, 800** 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params		**25.5**×10^6	**23.3**×10^6
FLOPs		**4.3**×10^9	**4.3**×10^9

Table 2. (**Left**) ResNet-50. (**Right**) LR-Net-50 with 7×7 kernel size and $m = 8$ channels per aggregation computation. Inside the brackets are the shape of a residual block, and outside the brackets is the number of stacked blocks in a stage. *LR-Net-50 requires similar FLOPs as ResNet-50 and a slightly smaller number of parameters.*

blocks, with $\{2, 2, 2, 2\}$ blocks for *res2*, *res3*, *res4*, *res5*, respectively. We also report results on networks stacked by basic residual blocks (LR-Net-18) and with a larger depth of layers (LR-Net-50, LR-Net-101). The robustness of LR-Nets to adversarial attacks is examined as well.

Our experimental settings and hyper-parameters mostly follow [31]. Please see the appendix for details.

6.1. Ablation Study

Impact of spatial scope Table 3 presents the impact of varying aggregation spatial scope for the proposed local relation networks, as well as the regular ResNet-26 network and its variant, ResNet-DW-26 [21], where the regular convolution layer is replaced by depthwise convolution. We have the following observations.

a) Importance of locality Existing bottom-up methods typically compute spatial aggregation over the entire input feature map [28, 24]. We first compare the proposed local relation networks, which enforces a *locality* constraint on spatial aggregation scope, to the equivalent method without this constraint (the "full image" column in Table 3)[5].

Without encoding any geometric priors (noted as "NG" in the table), we observe a huge improvement by changing the aggregation computation from using the whole input feature map to just a 7×7 neighborhood (from 50.7 to 71.9). Surprisingly, while the effectiveness of convolution networks is ascribed to the explicit modeling of geometric priors, we obtain competitive accuracy on ImageNet classification *purely by applying a locality constraint to a geometric-free aggregation method (71.9 vs. 72.8), demonstrating the effectiveness of the locality constraint.*

For the LR-Net-26 models which encode the geometric prior term described in Section 4, we also observe significant accuracy improvement, from 68.4 to 75.7. Noting that geometric priors can also act as a method to limit the aggregation scope (positions with smaller geometric prior values will contribute little to the final aggregation computation), the locality constraint further constrains the aggregation scope.

The locality constraint may also provide an information bottleneck to the network, which aids representation learning.

b) LR-Net Benefits from large kernel

The regular ResNet-26 model has similar accuracy with 3×3 and 5×5 kernels and loses accuracy when kernel size is larger than 5×5. For ResNet-DW-26 models, the accuracy is almost unchanged when moving from 3×3 to 9×9.

In contrast, both LR-Net-26 variants (with/without geometric prior terms) obtain steadily improved accuracy when the kernel size grows from 3×3 to 7×7: 70.8→ 71.5→71.9 for LR-Net-26 (NG) which has no geometric prior term, and 73.6 → 74.9 → 75.7 for LR-Net-26 which includes both the appearance composability and geometric prior terms. The results demonstrate the superiority of the proposed LR-Net in harnessing large kernels.

[5]We follow [28] to reduce the computation complexity of the "full image" method, by adopting downsampled *key* feature maps at high resolution stages: 4× for *res2*, 2× for *res3* and 2× for *res4*. Without this, the accuracy of "full image" methods would be even lower.

Table 3. Recognition performance of different architectures with varying spatial aggregation scope and different geometric prior terms on ImageNet classification. Top-1 and top-5 accuracy (%) is reported. "NG" denotes local relation networks without the geometric prior term. "G*" represents the method that directly learns the geometric prior values as described in Section 4. For fair comparison, we set all the architectures to have similar FLOPs with the regular ResNet-26 model, by adapting their bottleneck ratio α. For ResNet-(DW)-26 networks, we omit the "full image" column due to implementation difficulty.

network	geo. prior	aggregation spatial scope									
		3×3		5×5		7×7		9×9		full image	
		top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
ResNet-26	✓	72.8	91.0	73.0	91.1	72.3	90.7	71.4	90.3	-	-
ResNet-DW-26	✓	73.7	91.5	73.9	91.6	73.8	91.6	73.8	91.6	-	-
LR-Net-26 (NG)	✗	70.8	89.8	71.5	90.1	71.9	90.4	70.2	89.3	50.7	74.7
LR-Net-26 (G*)	✓	73.2	91.1	74.1	91.7	73.6	91.2	72.3	90.7	60.3	82.1
LR-Net-26	✓	73.6	91.6	74.9	92.3	**75.7**	**92.6**	75.4	92.4	68.4	88.0

Table 4. Ablation on *query/key* dimension (top-1 acc %).

query/key dim	1	2	4	8	16
LR-Net-26	75.7	75.4	75.1	74.7	73.7

Table 5. Ablation on channel sharing (top-1 acc %).

chn. sharing m	1	2	4	8	16	#chn.
LR-Net-26	75.3	75.5	75.5	75.7	75.3	70.9

Table 6. Ablation on appearance composability term and the normalization method (top-1 acc %).

method	app. comp. Eqn.			normalization	
	(3)	(4)	(5)	none	softmax
LR-Net-26	75.7	75.5	75.7	74.8	75.7

Table 7. Comparison with non-local neural networks.

method	top-1	top-5	# params	FLOPs
ResNet-26	72.8	91.0	16.0M	2.6G
NL-26	47.7	72.1	17.3M	2.6G
ResNet-26-NL	73.4	91.2	38.2M	5.6G
LR-Net-26	75.7	92.6	14.7M	2.6G
LR-Net-26-NL	76.0	92.8	37.1M	5.6G

Table 8. Applied on Different Architectures. For LR-Net-18, α balances increasing # params and decreasing FLOPs.

method	top-1	top-5	# params	FLOPs
ResNet-18	70.1	89.4	11.7M	3.1G
LR-Net-18	74.6	92.0	14.4M	2.5G
ResNet-50	76.3	93.2	25.5M	4.3G
LR-Net-50	77.3	93.6	23.3M	4.3G
ResNet-101	77.9	94.0	44.4M	8.0G
LR-Net-101	78.5	94.3	42.0M	8.0G

Effect of geometric prior In the last three rows of Table 3, the encoding of geometric priors is ablated. Both geometric prior embedding methods perform better than that without geometric priors for all spatial scopes, demonstrating their usefulness in visual feature learning.

Comparing the two geometric prior encoding methods, applying a small network on relative positions (the last row) performs better than directly learning independent geometric prior values. The gap between them is larger when the kernel size is larger (0.4 at 3×3 and 3.1 at 9×9), showing that it is crucial to additionally account for relative positions, especially when the neighborhood is large.

Figure 4 shows the learnt 7×7 geometric prior values after softmax at four stages of LR-Net-26. In general, for lower layers, the priors are sharper, indicating a preference for stronger constraints in the learning of appearance composability. For higher layers, the priors are smoother, indicating a preference for greater freedom.

Table 9. Comparison of robustness to *white-box* adversarial attacks for different architectures on ImageNet (top-1 acc %).

network	adversarial train			regular train
	clean	*targeted*	*untargeted*	clean
ResNet-26	44.9	37.9	14.4	72.8
ResNet-50	52.0	43.0	22.5	**76.3**
LR-Net-26	**52.1**	**44.2**	**26.8**	75.7

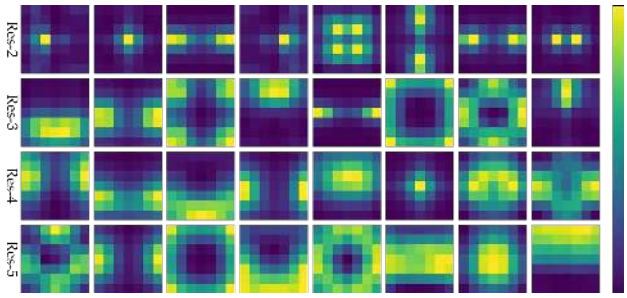


Figure 4. Illustration of learnt geometric prior values.

Other designs We also ablate various design elements.

a) Effect of query/key dim

Table 4 ablates the accuracy of the proposed LR-Net-26 model with varying *key*/*query* dimensions. We follow [26] to compute the appearance composablity between *key* and *query* vectors. We find decreased accuracy with increasing *key*/*query* dimension, indicating the superiority of *scalars* over typically-used vectors, as well as a better speed-accuracy tradeoff.

b) Effect of channel sharing

Table 5 ablates the LR-Net-26 model with varying numbers of shared channels per aggregation (m). The accuracy of LR-Net-26 is maintained when m is as large as 8, while being 3× faster than not sharing ($m = 1$).

c) Composability term

Table 6 ablates over different appearance composabilty terms: Eqn. (3), Eqn. (4) and Eqn. (5). They are found to work comparably well. Figure 5 exhibits representative examples of *key* and *query* maps learnt using the default term of Eqn. (3), which indicate that composability between semantic visual elements are learnt (girl and dog, tennis ball and racket).

d) Softmax normalization

Table 6 shows that including the softmax normalization in Eqn. (2) improves accuracy by 0.9, indicating the importance of normalization in balancing the two terms.

Comparison with other bottom-up methods Table 7 compares LR-Net with other bottom-up methods, i.e. non-local neural networks [28]. By directly replacing the 3×3 convolution layer in the ResNet-26 model by non-local modules, the model (NL-26) achieves an accuracy of 47.7, far lower than its regular counterpart. By applying the non-local modules after every residual block, top-1 accuracy of 73.4 is obtained, which is 0.6 higher than its regular counterpart, with about 2× more computation.

The local relation layer is designed to replace convolution layers for better representation power. It achieves a 2.9 gain over the regular ResNet counterpart with a similar computation load. We note that the non-local module is complementary to local relation networks, bringing a 0.3 gain when applied after every local relation block (see the last row).

On different/deeper networks In Table 8, we evaluate LR-Net with different/deeper network architectures, including ResNet-18 which consists of 8 *basic* residual blocks and ResNet-50/101 which use the same type of *bottleneck* residual blocks but have more layers (50 and 101 layers). The proposed networks are also effective on these architectures.

6.2. Robustness to adversarial attacks

We test the ability of LR-Net to withstand adversarial attacks using the *white-box* multi-step PGD attack method [22, 16], under both *targeted* and *untargeted* attacks. *Targeted* attacks randomly choose one wrong class as the target, while *untargeted* attacks succeed as long as the model produces wrong predictions. We utilize the hyper-parameters from [16] of the attacking methods, and employ the *targeted* multi-step PGD adversarial method for training with the same hyper-parameters except for the number of attack steps, set to 16 due to limited GPU resources.

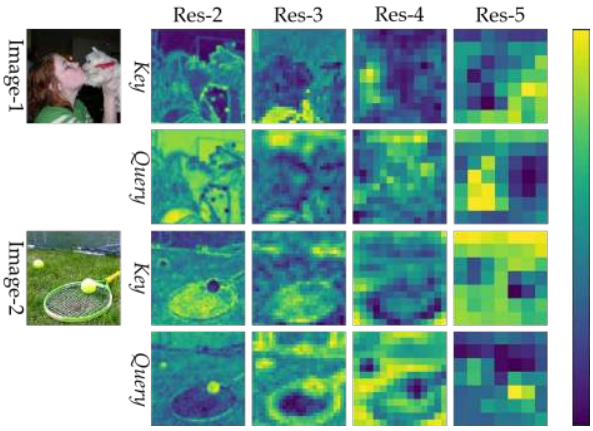


Figure 5. Illustration of learnt *key* and *query*.

Table 9 compares the robustness of LR-Net-26 and the regular ResNet-26/ResNet-50 models against *white-box* adversarial attacks on ImageNet. The LR-Net-26 model performs significantly better than ResNet-26 model against both the *targeted* (+6.3) and *untargeted* attacks (+12.4). The LR-Net-26 model also performs better than the ResNet-50 model (+0.8 for *targeted* attacks and +4.3 for *untargeted* attacks), which uses about 2× more FLOPs and has better top-1 accuracy in regular training (see the last column of Table 9). These results indicate that the superior performance of LR-Net in adversarial robustness is not purely due to larger capacity but also because of the architecture itself.

7. Conclusion and Future Works

This paper presents the local relation layer, a basic image feature extractor following the general philosophy of introducing compositionality into representation. A deep network composed by this new layer demonstrates strong results on ImageNet classification, significantly expanding the practicality of bottom-up methods, which are long believed to be more fundamental in representation than top-down methods such as convolution.

We note that the study of this new layer is still at an early stage. Future directions includes: 1) better GPU memory scheduling for faster implementation; 2) better designs to outperform advanced convolution methods such as deformable convolution [6, 36]; 3) exploring other properties and the applicability on other vision tasks.

References

[1] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 1, 3, 4

[2] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 1

[3] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 3

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 2, 4

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 2, 3, 4

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 2, 3, 4, 8

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009. 2

[8] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 5

[10] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018. 2, 4

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2, 4

[12] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 3

[13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[14] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4201–4209, 2017. 2, 3, 4

[15] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pages 667–675, 2016. 2, 3, 4

[16] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018. 8

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2, 3, 4

[18] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 2, 3

[19] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–351, 2018. 3

[20] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 2

[21] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 6

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 8

[23] Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018. 1

[24] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017. 1, 2, 3, 4, 5, 6

[25] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 2, 4

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 3, 8

[27] Wilhelm von Humboldt. *On Language: On the Diversity of Human Language Construction and Its Influence on the Mental Development of the Human Species*. Cambridge Texts in the History of Philosophy. Cambridge University Press, 1999/1836. 1

[28] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 2, 3, 4, 6, 8

[29] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. In *Advances in Neural Information Processing Systems*, pages 560–570, 2018. 3

[30] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In

So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 3

[31] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017. 2, 4, 6

[32] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685, 2018. 3

[33] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2, 4

[34] Alan L Yuille and Chenxi Liu. Deep nets: What have they ever done for vision? *arXiv preprint arXiv:1805.04025*, 2018. 1

[35] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1

[36] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018. 2, 4, 8