Region-Based Global Reasoning Networks

Chuanming Wang,[1] **Huiyuan Fu,**[1] **Charles X. Ling,**[2] **Peilun Du,**[1] **Huadong Ma**[1]
[1]Beijing University of Posts and Telecommunications, China
[2]Western University, Canada
{wcm, fhy, dupeilun1995, mhd}@bupt.edu.cn, charles.ling@uwo.ca

Abstract

Global reasoning plays a significant role in many computer vision tasks which need to capture long-distance relationships. However, most current studies on global reasoning focus on exploring the relationship between pixels and ignore the critical role of the regions. In this paper, we propose an novel approach that explores the relationship between regions which have richer semantics than pixels. Specifically, we design a region aggregation method that can gather regional features automatically into a uniform shape, and adjust theirs positions adaptively for better alignment. To achieve the best performance of global reasoning, we propose various relationship exploration methods and apply them on the regional features. Our region-based global reasoning module, named ReGr, is end-to-end and can be inserted into existing visual understanding models without extra supervision. To evaluate our approach, we apply ReGr to fine-grained classification and action recognition benchmark tasks, and the experimental results demonstrate the effectiveness of our approach.

Introduction

Recently, more and more studies focus on exploring relationships to perform global reasoning in visual understanding models (Yue et al. 2018; Wang et al. 2018) and it has achieved outstanding performance in many computer vision tasks like image classification, image generation, and action recognition. The relationship is calculated and distributed based on pixel-level features to make each of pixels contains the information of the whole input. However, these methods ignore the significance of regions in building relationships.

As described in (Wang and Gupta 2018), finding regions and establishing relationships are two key ingredients to understand the visual input. Humans can find the objects quickly and make global reasoning due to the prior knowledge and the ability of global perception. As shown in Fig. 1, when we apperceive images, we always find the objects first, and then we explore the relationships between them (like the person is *playing* football or *watching* television). Through the relationships, global reasoning is performed, and we can realize the role of each object played and understand the complete content expressed by the whole visual data ultimately. Therefore, modeling relationships between regions is significant for the understanding procedure.

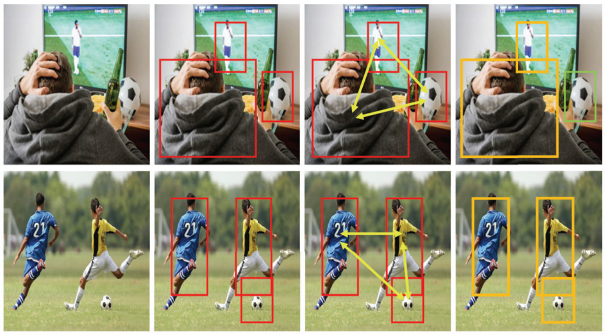


Figure 1: Two pictures both contain a football but have different meanings. When we see these pictures, we find the primary objects (red boxes in the second column), determine the relationships between them (arrows in the third column) and realize the role of each object played in the image, to achieve the purpose of understanding. The football in the yellow box is the protagonist and it is a decoration in the green box due to the relationship between person and football in the below is obviously stronger than that in the above.

Motivated by the observation, we hope to introduce regions into the relationship exploration to promote global reasoning. Compared to pixels, regions contain richer semantic information and more complete spatial-temporal information which can contribute to the understanding of the visual scene. However, global reasoning through the relationship between regions is difficult. On the one hand, regions with rich semantics always have different scales, so it is challenging to locate these regions and extract their features. On the other hand, the regions contain richer information than the pixels which makes it more complicated to explore the relationship between regions.

To overcome the above difficulties, in this work, we propose an elaborate approach which can enable the CNN architectures to perform region-based global reasoning. For generating meaningful regions, we first split the input into several multi-scale raw regions and aggregate these regions through a special aggregation process which can unify regions into fixed feature vector representations while preserving spatial information as much as possible. To solve the problem of misalignment, instead of using explicit regional offset learning which will result in difficulty of feature extraction, we implicitly exchange the feature of one region with its neighbors to achieve the purpose of moving the region in spatial space. To better explore relationship between regions, we design three methods which model the regional relationship from different perspective. There will be a corresponding descriptor existed in the produced relationship for each region, and by applying the descriptor to the region, it can obtain the information contained in other regions which may be far apart. Therefore, every region can realize whether it is the protagonist through the global reasoning. To avoid breaking the flow of information on the network, we then distribute the regional relationship into all pixels according to a group of local descriptors which are generated from the regional feature the pixel belongs to so that it can be easily embedded in modern CNN architectures.

We design an end-to-end trainable module named ReGr for region-based global reasoning, which can be embedded into many modern visual understanding models without extra supervision. We show the effectiveness and generality of our designed module in the tasks of fine-grained classification and action recognition. Extensive experiments demonstrate that given the regional reasoning module, the performance of state-of-the-art CNN models can be boosted with a clear-cut improvement, and it can bring more promotion than other global reasoning methods. Our contributions can be concluded as follows:

- We propose a novel regional aggregation method which can extract richer semantic regional features without extra supervision to reserve more spatial information and achieve better alignment. It can enable our proposed ReGr to embed into current CNN architectures seamlessly.
- We design three regional relationship exploration mechanisms to sufficiently explore the relationships between regions which usually have more complicated semantics than the pixels. More importantly, we give an insightful analysis on building these different regional relationships.
- We conduct extensive experiments on various datasets. Our proposed approach achieves superior performance compared with the state-of-the-art methods on the challenging fine-grained classification task and action recognition task.

Related Work

Global Reasoning. Conditional random fields are often used to build global relationship among the entire image for semantic segmentation (Lafferty, McCallum, and Pereira 2001; Chen et al. 2018a). These methods are generally used as post-processing processes and cannot be well embedded into the CNN architectures for optimization. Graph Convolutional Network(GCN) (Kipf and Welling 2017) is adopted for building the relationship between detected objects to extract rich semantic information for video classification (Wang and Gupta 2018), which needs extra annotations of regions and its performance is seriously affected by the capability of used detector. Recently, GCN also is applied to the action recognition task (Chen et al. 2019) in which the feature maps are projected to a special space to perform global reasoning. Bilinear pooling is also found to be able to explore the long-distance relationship to perform global reasoning (Wang et al. 2018; Chen et al. 2018c; Yue et al. 2018). However, there exists a definite gap in their produced relationship, which is built directly on pixels without the alleviation of regions. (Hu et al. 2018a; Chen et al. 2018b) also want to use regions to perform global reasoning, but they need extra annotations which limit their application. (Li and Gupta 2018) uses clustering to find regions, but the number of vertices can seriously affect its results. Compared to these methods, we bridge the gap via modeling the relationship between regions, and the designed module can be used in many modern visual understanding models without additional bounding-box annotations.

Regional Feature Representation. Regional feature representation is widely used in computer vision tasks, e.g., SIFT (Lowe 2004), Fisher Kernel (Perronnin, Sánchez, and Mensink 2010) and LBP (Ojala, Pietikäinen, and Harwood 1994). Recently, average pooling used in SENet (Hu, Shen, and Sun 2018) and GENet (Hu et al. 2018b) aggregates the regional feature to produce local attention for each pixel within the region. However, the attention without the relationship between regions is not enough, which lacks guidance from global information. SPP-Net (He et al. 2015) and Fast-RCNN (Girshick 2015) adopt a particular pooling method for object detection task, but it lacks the ability to recover spatial information. T-C3D (Liu et al. 2018) segments the video into many regions over time and summarizes their scores to the final score. ARG-Net (Liu and Ma 2019) designs a regional loss for exploring the effectiveness of regions in anomaly detection. The aggregation function in our approach can be applied to multi-scale regions and it can adjust the regional feature with its surrounding regions to achieve better alignment. Compared with the pooling methods mentioned above, the aggregated features of our approach can be recovered through matrix multiplication without losing lots of spatial information.

Deep Architecture Design. Designing a better module to enhance the performance of CNN has always been the concern of researchers. VGG (Simonyan and Zisserman 2015) combines multiple convolution layers as a basic module to explore the impact of the depth of CNNs. ResNet (He et al. 2016) and DenseNet (Huang et al. 2017) introduce the residual pathway to build modules which achieve non-locality of the network at the level of layers. STN (Jaderberg et al. 2015) and DCN (Dai et al. 2017) propose a new mapping function for convolutional operation and improve the ability of the network to model deformed objects. The series of InceptionNets (Ioffe and Szegedy 2015) continually explore the effects of different convolutional kernels to improve the capability of CNNs. These modules need to stack layers to model the relationship of disjoint regions, which will cause inefficiency of global perception and reasoning. Given our proposed approach, the weakness can be relieved and the

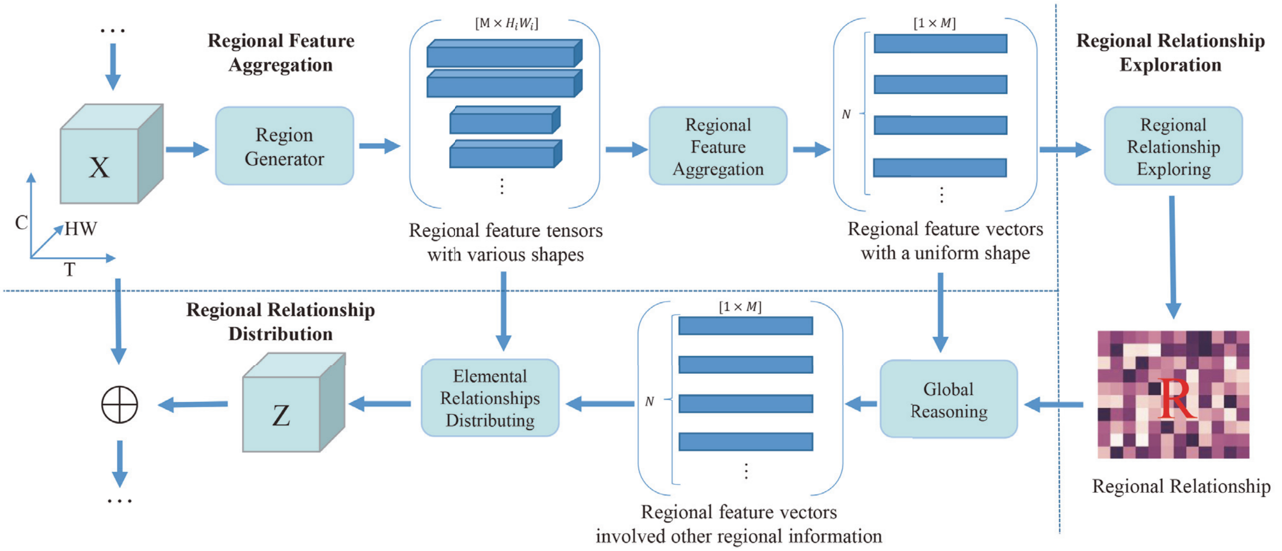

...
Regional Feature Aggregation
X
Region Generator
[M × H_iW_i]
Regional feature tensors with various shapes
Regional Feature Aggregation
[1 × M]
N
Regional feature vectors with a uniform shape
Regional Relationship Exploration
Regional Relationship Exploring
C
HW
T
Regional Relationship Distribution
⊕
Z
Elemental Relationships Distributing
[1 × M]
N
Regional feature vectors involved other regional information
Global Reasoning
R
Regional Relationship
...

Figure 2: The overview of our approach. In regional feature aggregation, we first split the input tensor into several raw regions and then aggregate them into a uniform shape with better alignment. In regional relationship exploration, we design and analyze the effects of various relationship building methods to produce a relationship matrix. Finally, we perform global reasoning on the regional relationships and distribute the relationship to pixels according to the local regional features.

performance of these networks will be boosted.

Approach

The overview of our approach is shown in Fig. 2, which can be split into three parts, regional feature aggregation, regional relationship exploration, and regional relationship distribution. By applying regional feature aggregation, we can gather the regional features into a uniform shape with better alignment. Then, the relationship exploration method is performed and we can obtain a regional relationship matrix. Finally, we perform global reasoning and distribute the relationship into every pixel.

Regional Feature Aggregation

To generate regions, we first split the whole input into several parts at spatial space with various scales. For one scale of region K_i with the shape of $\{H_i, W_i\}$ and input feature map $X \in \mathbb{R}^{C\times T\times H\times W}$($C$ denotes the channel dimension, H denotes height dimension, W denotes width dimension, T denotes temporal dimension, the dimension of T should be 1 for image), we formulate the process as:

$$X^{K_i} = \rho(X, K_i) \in \mathbb{R}^{T\times\frac{H}{H_i}\times\frac{W}{W_i}\times C\times H_iW_i}. \quad (1)$$

For the scale K_i, we can finally split the input into total $T\frac{HW}{H_iW_i}$ regions and each region has a feature tensor $X_j^{K_i} \in \mathbb{R}^{C\times H_iW_i}$ where j is its index in the resulted regions. We union all the regions to form a region set $X^K = \bigcup_{\forall j} X^{K_j}$ and let X_i^K denote the i-th region in the set.

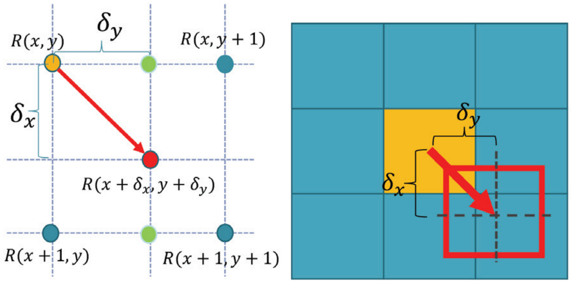
This method is easily implemented and widely adopted in feature extraction methods (Girshick 2015; Lowe 2004; Ojala, Pietikäinen, and Harwood 1994), but it has two disadvantages. First, the shape of the produced region varies depending on the scale K_i, and the inconstancy of shapes makes the calculation of the relationship between regions complicated. Average pooling, max pooling or RoI pooling (He et al. 2015; Girshick 2015) can regular these regions into same shape, but the spatial information contained in regions will be damaged and cannot be restored easily. Second, these handcrafted regions may not be aligned well to the objects contained in the input. Learning the offsets of the regional location directly is difficult and will cause fractional regional coordinates, which will hamper the extraction of the regional feature. To overcome the above difficulties, we design a special aggregation function to gather regional features. Compared with previous methods which extract regional feature after moving regional position, we first extract regional features and then implicitly move the regions by exchanging information with their adjacent regions to achieve better alignment.

For a region X_i^K, the formulation of aggregation function can be described as:

$$A_i = \mu(X_i^K; W_\mu) \odot \nu(X_i^K; W_\nu)^T. \quad (2)$$

In Eq.2, μ with parameter W_μ and ν with parameter W_ν compact regional information, $\odot$ denotes matrix multiplication. After reshaping, the output A_i can be represented as a feature vector whose length is denoted as M and is controlled by μ and ν. Therefore, we aggregate the multi-scale regional feature into a uniform shape. If we set the μ and ν to identity maps, the X_i^K can be recovered by matrix multiplication when it is full rank. Therefore,compared to the previous pooling operations, our method can reserve more spatial information.

To solve the problem of misalignment, we exchange the contents of a region with its neighbors which is called *refine* operation. We assume that the A_i is at the $\{x, y\}$ position



(a) (b)

Figure 3: Illustration of exchanging content of $R(x, y)$ with its neighbors. In 3a, the blue and orange nodes denote the regional features and the red node is the output of bilinear interpolation. We achieve the movement of regions in spatial space as shown in 3b by exchanging the regional feature values with its neighbors as shown in a.

in the spatial space and is denoted as $R(x, y)$, so the coordinates of its neighbors and its own can be described as $\mathcal{C}(x, y) = \{(x-1, y-1), (x-1, y), \cdots, (x+1, y+1)\}$. We learn a pair of offset parameters δ_x, δ_y and an amplitude parameter m from $R(x, y)$:

$$\delta_x, \delta_y, m = \eta(R(x, y); W_\eta), \tag{3}$$

then we exchange its content with its neighbors via bilinear interpolation:

$$R(x + \delta_x, y + \delta_y) = m * \sum_{\boldsymbol{q} \in \mathcal{C}(x,y)} B(\boldsymbol{q}, \boldsymbol{\delta}) \cdot R(\boldsymbol{q}), \tag{4}$$

where $B(\boldsymbol{q}, \boldsymbol{\delta}) = g(q_x, x + \delta_x) \cdot g(q_y, y + \delta_y)$, $g(a, b) = max(0, 1 - |a - b|)$. As shown in Fig. 3, the learnable parameters δ_x and δ_y can be seen as the direction in which the region is to be moved, and the m can be regarded as the weight of the region when the information is exchanged. We uniform the δ_x, δ_y into range (-1.0,1.0) and uniform m into range(0.0,1.0) to make sure that the region always exchange information with its neighbors.

Consequently, we aggregate the regional feature into a concise and unified representation with better alignment. In the next subsection, we will describe how to build long-distance relationships among these regions. We concatenate all the aggregated regional features as $V \in \mathbb{R}^{N \times M}$ where N denotes the total number of regions and use V_i to denote the i-th aggregated regional feature vector.

Regional Relationship Exploration

As mentioned above, it is the relationship between regions that gives the visual data a rich meaning. Therefore, the method of building long-distance regional relationships is a very significant component in our approach. Regions usually have more complicated semantics than pixels, so the capability of regional relationship exploration method affects the performance of global reasoning seriously. In this subsection, we design three regional relationship exploration mechanisms to sufficiently explore the relationships, providing a comprehensive description and insightful analysis for these methods which includes bilinear pooling, graph convolution network and attention.

Bilinear Pooling. Bilinear pooling (BP) is adopted in fine-grained classification taks (Lin, Roy Chowdhury, and Maji 2015) and then it has been modified and used in NonLocal-Net (Wang et al. 2018) to model non-local relationships. For input $X \in \mathbb{R}^{N \times M}$, the modified bilinear pooling that used for capturing long-distance relationship can be formulated as:

$$\mathbf{R} = X \odot X^\top \in \mathbb{R}^{N \times N}. \tag{5}$$

The output $\mathbf{R}$ is a two-dimensional matrix and the (i, j)-th element in the matrix can be seen as the relation between the i-th region and the j-th region. Following NonLocalNet net (Wang et al. 2018), we apply two convolution layers on V receptively and then use bilinear pooling on their outputs to produce relationship matrix $\mathbf{R}$:

$$\mathbf{R} = Conv(V) \odot Conv(V)^T \in \mathbb{R}^{N \times N}. \tag{6}$$

Graph Convolution Network. Graph convolution networks (GCN) have shown effecient ability of relationships reasoning in mutliply domains. It can take unregular data as input and learn the weight of nodes and edges through training. For a graph G with N nodes and its adjacency martix D_g, a single-layer GCN (Kipf and Welling 2017) can be defined as:

$$Z = GUW_g = ((I - D_g)\,U)\,W_g, \tag{7}$$

where I is the identity matrix, U denotes information diffusion of nodes and W_g denotes a linear transformation for updating the state of nodes.

The GCN can be trained with gradient decent and we use the regional features as basic nodes during training. Following GloRe-Net (Chen et al. 2019), we adopt two $1D$ convolution layers to implement the graph convolution:

$$\mathbf{R} = Conv1\mathrm{D}\left(Conv1\mathrm{D}(V^T)^T\right) \in \mathbb{R}^{N \times N}. \tag{8}$$

The convolution layers are applied on different dimensions to perform node-wise learning and channel-wise learning receptively. After the node-wise learning, each node has the information of other nodes and we modify the second convolution layer which expand the dimension of node feature from M to N to present the relation with other nodes.

Attention. Attention (Att) plays an important role in human perception. It helps humans selectively focus on salient parts in a sequence of glimpses instead of processing the whole scene, which can benefit us to capture visual structure better. Many researchers incorporate the attention mechanism into CNNs to imporve performance. Inspired by SENet (Hu, Shen, and Sun 2018), for the input V, we design a method which sequentially infers a 1D channel attention map $L_c(V^T) \in \mathbb{R}^{1 \times M}$ and a 1D regional attention map $L_s(V) \in \mathbb{R}^{N \times 1}$:

$$Z = L_s\left(\left(L_c(V^T) \otimes V\right)^T\right) \otimes V \in \mathbb{R}^{N \times M}, \tag{9}$$

where $\otimes$ denotes element-wise multiplication. Each attention function is composed by two $1D$ convolution layers and one $1D$ max-pooling layer:

$$L(V)_{c/s} = Conv1\mathrm{D}\left(Conv1\mathrm{D}(Pooling(V))\right). \tag{10}$$

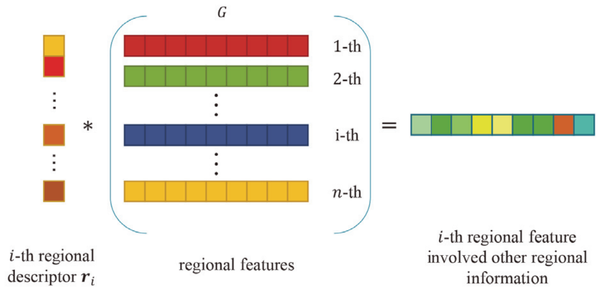

G
1-th
2-th
i-th
n-th
*
=
i-th regional descriptor r_i
regional features
i-th regional feature involved other regional information

Figure 4: Illustration of global reasoning on regional features. For a descriptor $\mathbf{r}_i$, it describes the degree of association of each region with the i-th region. By adding the weights of the regional features, the information contained in each region is fused and applied to the i-th region.

Distribution of Regional Relationship

Compared with previous global reasoning methods, the produced relationship of our approach is about regions which can not be applied on pixels directly. Therefore, we first distribute the produced relationship into each region to perform global reasoning, so that the region can realize the role it played in the input. Then, for each region, we distribute its feature into pixels within it depending on local descriptors.

For the regional features $V \in \mathbb{R}^{N \times M}$ and regional relationships $R \in \mathbb{R}^{N \times N}$, we first distribute relationship information into regions:

$$Z = \mathbf{R} \odot V \in \mathbb{R}^{N \times M}, \tag{11}$$

where $\mathbf{R}$ is the relationship matrix produced in the last step and Z is the regional features which have already involved other regional information. Look closer to see the details that how the global reasoning is performed. As we mentioned above, the i-th row $\mathbf{r}_i$ in $\mathbf{R}$ is a descriptor which indicates the influence of all regions on i-th regions. As shown in Fig. 4, when the descriptor is applied on the regional features, it gathers all regional information into one feature vector with different weights:

$$\mathbf{z}_i = \mathbf{r}_i \cdot V = \sum_j r_{i,j} \mathbf{v}_i \in \mathbb{R}^{1 \times M}. \tag{12}$$

Therefore, every region has an impact on the i-th region and the impacts of high related regions are deeper. Note that this method is not applied to the attention method, in which the global reasoning has already been performed when the attention is applied to the region features.

Then we describe how to distribute the i-th regional feature $\mathbf{z}_i$ into every pixel within it, which is called elemental distribution. For one region, we first learn a group of local descriptors from its original feature tensor $X_i^K \in \mathbb{R}^{M \times H_k \times W_k}$($k$ denotes the scale that produces this region). According to these descriptors, we distribute the regional information into pixels within it:

$$Y_i = \mathbf{z}_i \cdot \xi(X_i^K; W_\xi) \in \mathbb{R}^{C \times H_k \times W_k}, \tag{13}$$

where ξ denotes the process of learning descriptors. Two disjoint regions are associated together through the regional relationship and the information contained in the regions is distributed into every pixels within regions. Therefore, two distant pixels can perceive each others. We gather all the Y_is whose X_i^K are produced by the same scale, then we reshape them into the same size as the input tensor and add them together. After a BatchNorm layer (Ioffe and Szegedy 2015), we add them with the input tensor to construct a residual connection.

Experiments

We first describe the benchmark datasets and implementation details. Then we conduct extensive ablation studies to illustrate the impact of different components on the performance of our proposed ReGr module. Finally, we report our results on different tasks and compare the performance of our approach with the state-of-the-art approaches.

Datasets and Implementation Details

Datasets. We empirically evaluate our approach on two challenge tasks: fine-grained classification and action recognition. For fine-grained classification task, we adopt the Birds-200-2011 (CUB) dataset (Welinder et al. 2010) as the benchmark dataset. For action recognition task, we evaluate our approach on the UCF101 (Soomro, Zamir, and Shah 2012) and Kinetics (Carreira and Zisserman 2017). For action recognition task, all models are trained and tested on RGB input. We report the results of UCF101 according to its official *split1* file. For Kinetics, there are about 240k training videos can be downloaded for experiments due to the corruption of urls and we train our networks on these videos.
Baseline. For the fine-grained classification task, we use the standard 2DResNet (He et al. 2016) as the baseline. For action recogniton task, we use C2DResNet (Wang et al. 2018) to conduct ablation experiments on the UCF101. We combine I3DResNet (Carreira and Zisserman 2017) with our module and evaluate it on the Kinetics dataset. Their architectures can be found in the supplementary material.
Training. We use the models pretrained on ImageNet to initialize the weights and set the weight of BN layer in our module to zero. A dropout layer with ratio 0.5 is inserted after the pooling layer to avoid overfitting. For action recognition task, we randomly crop out 64 consecutive frames from the full-length video and then extract 8 frames with random interval. We resize the shorter side randomly in [256,320] and crop random 224 × 224 pixels as input. For fine-grained classification task, we crop a patch which is random in [0.08 ,1.25] of the original input, then we resize the patch to 448×448. We adopt SGD as the optimizer with a weight decay of 0.0001 and momentum of 0.9. The strategy of gradual warmup is used during training. We train our models for 100 epochs in total, starting with a learning rate of 0.01 and reducing it by a factor of 10 at 30^{th}, 60^{th} and 80^{th} epochs, receptively.
Testing. For fine-grained classification task, we resize the shorter side of the input to 512 and take a center crop of 448×448 pixels as the input. For action recognition task, we

Table 1: Results of different scales on CUB200. Scale (4,4) achieves the better results than scale (7,7). When we combine the scale (4,4) and scale (7,7), the accuracy is improved, but too many scales will cause the accuracy to drop.

Model	(1,1)	(4,4)	(7,7)	(4,7)	(7,4)	Acc. Top1
Res50						84.61%
	✓					85.40%
Res50		✓				85.69%
+			✓			85.55%
ReGr		✓	✓			86.19%
		✓	✓	✓	✓	85.47%

Table 2: Results of different scales on UCF101 dataset. We use smaller scales than fine-grained classification task due to the difference of input size.

Model	(1,1)	(3,3)	(5,5)	(3,5)	(5,3)	Acc. Top1
Res50						82.80%
	✓					83.67%
Res50		✓				83.79%
+			✓			83.50%
ReGr		✓	✓			83.98%
		✓	✓	✓	✓	83.95%

use spatially fully convolutional inference (Simonyan and Zisserman 2015) to report the video-level results. We evenly sample 10 clips from the whole video and resize its shorter side to 256. Multi-crop is used to cover the entire spatial space along the longer side. We average the softmax scores of all clips as the prediction.

Ablation Studies

To study the effects of each parts of our approach on the results, we conduct ablation experiments for regional scales, relationship exploration methods and types of architectures.

Regional Scales. We first investigate the impact of regional scales. We use (a, b) to represent the region of $a \times b$ size on the feature map and the results is present in Table 2. The scale (4, 4) can bring more improvement and (1, 1) and (7, 7). When there are many scales, the performance is further improved, but excessive regions cause the performance to decrease. The special scale $(1, 1)$ denotes that to build relationship between pixels which is commonly used in other methods, and the lower accuracy indicates that the long-distance relationship through the pixels is worse than that built through the regions. In addition, all the scales have the same amount of parameters and the difference in results is directly due to regional scale changes.

Relationship Exploration Methods. We investigate the effect of three relationship exploration methods proposed in Section. Table 3 summaries the results. We can observe that *Bilinear Pooling* is more suitable for learning the long-distance relationship than other two methods. The *Attention* method has a small amount of calculation due to the Pool operation and it achieves good results in fine-grained classification task. For the *GCN*, we assume that the fully connected graph structure may limit its reasoning ability and building a better graph structure may promote its performance.

Table 3: Results of different global reasoning methods. Bilinear pooling can obtain best performance on both CUB200 and UCF101 than the other global reasoning methods.

Model	Dataset	+ ReGr		
		BP	GCN	Att
Res50	CUB200	86.19%	85.87%	85.92%
	UCF101	83.98%	83.10%	82.81%

Table 4: Results of different module numbers. Five modules can obtain best performance on both CUB200 and UCF101, which shows that more modules can bring more performance gains.

Model	Num.	Dataset	
		CUB200	UCF101
Res50	None	84.61% / 96.36%	82.80% / 95.25%
	+1 ReGr	86.19% / 97.05%	83.98% / 96.94%
	+2 ReGr	86.00% / 96.98%	84.13% / 96.74%
	+5 ReGr	86.31% / 96.86%	84.28% / 97.12%

Module Number. We add 1, 2 and 5 units in ResNet50 receptively and study the impression of module numbers on performance. The result shown in Table 4 demonstrates that adding more modules can improve the results.

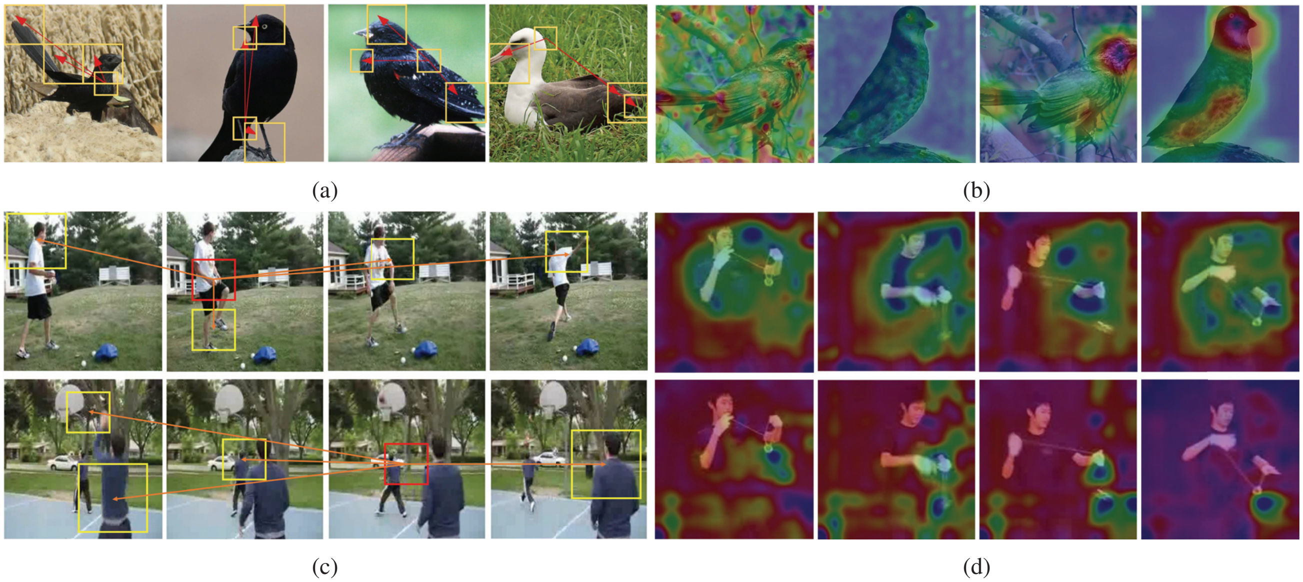
Aggregation Method. For traditional pooling methods, average pool and max pool, we conduct experiments on the CUB200 dataset. Our proposed aggregation method (86.19%) can surpass them by 1.02% (85.17%) and 1.12% (85.07%) receptively with the same settings. The results show the effectiveness of our aggregation method.

Results

In this subsection, we compare our approach with state-of-the-art global reasoning methods, e.g. NL Unit (Wang et al. 2018), CGNL Unit (Yue et al. 2018), GloRe Unit (Chen et al. 2019). The results are shown in Table 5 and Table 6. Our approach can surpasses other global reasoning methods on both CUB200 dataset and UCF101 dataset. And the result also shows that the refine operation can benefit the process of building regional relationships. We add five ReGr

Table 5: Results of different global reasoning units on the CUB200 dataset. Our approach surpasses other approaches even without the refining operations. The result is the average of multiple experiments.

	Unit	Acc. Top1	Acc. Top5
Res50	None	84.61%	96.36%
	NL (CVPR18)	85.45%	96.36%
	CGNL (NeurlIPS18)	85.12%	96.41%
	GloRe (CVPR19)	85.78%	96.27%
	ReGr w/o refine (our)	85.88%	96.65%
	ReGr w/ refine (our)	**86.19**%	**97.05**%



(a)

(b)

(c)

(d)

Figure 5: Visualization analysis. (a) and (c) show the high related regions. The regions pointed by the arrows have strong relationship with the region where the arrow starts. For birds, the high related regions locate at their heads, claws and wings which are usually the most recognizable places. For videos, the high related regions usually have obvious objects and discriminated behaviors. The heatmap shown in (b) and (d) present the feature maps before and after our module. After our module, high responses are gathered in rich semantic regions and some interference information is eliminated.

Table 6: Results of different global reasoning methods on the UCF101 dataset. Without refine operation, the result of our module is worse than other global reasoning methods and by adopting the refine operation, the perfomance of our module is boosted with a clear improvement.

	Unit	Acc. Top1	Acc. Top5
Res50	None	82.80%	95.25%
	NL (CVPR18)	83.40%	96.19%
	CGNL (NeurIIPS18)	83.33%	96.00%
	GloRe (CVPR19)	83.30%	96.78%
	ReGr w/o refine (our)	83.12%	95.89%
	ReGr w/ refine (our)	**83.98%**	**96.94%**

Table 7: Results on Kinetics dataset. For fair comparison, we reproduce the result of NL-Net which is the standard method to model relationship between pixels. Our approach surpasses it on both clip-level and video-level.

Method	Clip Top-1	Video Top-1
I3D-RGB (CVPR17)	-	71.1%
R(2+1)D-RGB (CVPR18)	-	72.0%
S3D-G (ECCV18)	-	74.7%
NL-Net (CVPR18)	67.02%	74.47%
ReGr-Net(Ours)	**67.57%**	**75.02%**

modules into I3DResNet (Carreira and Zisserman 2017) to turn it as ReGrNet and train ReGrNet on kinetics to see its performance on large datasets. Table 7 shows the results on Kinetics dataset. Compared with other methods, our approach can produce comparable result with the state of the art methods, showing the effectiveness of our approach on large datasets. We compare our approach with multiple state-of-the-art mehtods, e.g., I3D-RGB (Carreira and Zisserman 2017), R(2+1)D-RGB (Tran et al. 2018), S3D-G (Xie et al. 2018), NL-Nets (Wang et al. 2018) and GloRe-Nets (Chen et al. 2019).

To further understand the effects of our approach, the visualization analysis is shown in Fig. 5. We first present the high related regions according the produced relationship martix and we can see that the high related regions are gathered at the most recognizable places, e.g. the heads, claws and wings of birds and the key points of human. Then we show the feature map before and after our module, and we can see that the high responses are gathered in semantic regions, showing the effectiveness of our approach.

Conclusion

We have introduced an approach named ReGr that can enable the current CNN architectures to perform region-based global reasoning. It extracts regional features into a concise and unified representation with better alignment automatically through a well-designed aggregation process, explores regional relationships effectively by applying relationship exploration methods, and distributes the relationship to make it end-to-end. Ablation experiments illustrate that incorporating regional relationships into global reasoning can improve its performance. Extensive experiments show that our model produces competitive or better results on various benchmark datasets than other methods, showing the effectiveness of our approach.

Acknowledgement

The work is supported in part by the NSFC (61720106007, 61872047), the Funds for Creative Research Groups of China under Grant No. 61921003; the NSFC-Guangdong Joint Found under No.U1501254, the 111 Project (B18008).

References

Carreira, J., and Zisserman, A. 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 4724–4733.

Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40(4):834–848.

Chen, X.; Li, L.; Fei-Fei, L.; and Gupta, A. 2018b. Iterative visual reasoning beyond convolutions. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 7239–7248.

Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; and Feng, J. 2018c. A^2-nets: Double attention networks. In *NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, 350–359.

Chen, Y.; Rohrbach, M.; Yan, Z.; Yan, S.; Feng, J.; and Kalantidis, Y. 2019. Graph-based global reasoning networks. In *CVPR 2019*.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *ICCV 2017, Venice, Italy, October 22-29, 2017*, 764–773.

Girshick, R. B. 2015. Fast R-CNN. In *ICCV 2015, Santiago, Chile, December 7-13, 2015*, 1440–1448.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 37(9):1904–1916.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 770–778.

Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; and Wei, Y. 2018a. Relation networks for object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 3588–3597.

Hu, J.; Shen, L.; Albanie, S.; Sun, G.; and Vedaldi, A. 2018b. Gather-excite: Exploiting feature context in convolutional neural networks. In *NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, 9423–9433.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 7132–7141.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2261–2269.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML 2015, Lille, France, 6-11 July 2015*, 448–456.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and Kavukcuoglu, K. 2015. Spatial transformer networks. In *NeurIPS 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2017–2025.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001, Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, 282–289.

Li, Y., and Gupta, A. 2018. Beyond grids: Learning graph representations for visual recognition. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, 9245–9255.

Lin, T.; Roy Chowdhury, A.; and Maji, S. 2015. Bilinear CNN models for fine-grained visual recognition. In *ICCV 2015, Santiago, Chile, December 7-13, 2015*, 1449–1457.

Liu, K., and Ma, H. 2019. Exploring background-bias for anomaly detection in surveillance videos. In *Proceedings of the 27th ACM International Conference on Multimedia*, 1490–1499. ACM.

Liu, K.; Liu, W.; Gan, C.; Tan, M.; and Ma, H. 2018. T-c3d: Temporal convolutional 3d network for real-time action recognition. In *AAAI Conference on Artificial Intelligence*, 7138–7145.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.

Ojala, T.; Pietikäinen, M.; and Harwood, D. 1994. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *ICPR 1994, Jerusalem, Israel, 9-13 October, 1994, Volume 1*, 582–585.

Perronnin, F.; Sánchez, J.; and Mensink, T. 2010. Improving the fisher kernel for large-scale image classification. In *ECCV 2010, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, 143–156.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR* abs/1212.0402.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A closer look at spatiotemporal convolutions for action recognition. In *CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 6450–6459.

Wang, X., and Gupta, A. 2018. Videos as space-time region graphs. In *ECCV 2018, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, 413–431.

Wang, X.; Girshick, R. B.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 7794–7803.

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV 2018, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, 318–335.

Yue, K.; Sun, M.; Yuan, Y.; Zhou, F.; Ding, E.; and Xu, F. 2018. Compact generalized non-local network. In *NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, 6511–6520.