

Visual Concept Reasoning Networks

Taesup Kim*^{1,2}, Sungwoong Kim², Yoshua Bengio¹

¹Mila, Université de Montréal

²Kakao Brain

Abstract

A split-transform-merge strategy has been broadly used as an architectural constraint in convolutional neural networks for visual recognition tasks. It approximates sparsely connected networks by explicitly defining multiple branches to simultaneously learn representations with different visual concepts or properties. Dependencies or interactions between these representations are typically defined by dense and local operations, however, without any adaptiveness or high-level reasoning. In this work, we propose to exploit this strategy and combine it with our *Visual Concept Reasoning Networks* (VCRNet) to enable reasoning between high-level visual concepts. We associate each branch with a visual concept and derive a compact concept state by selecting a few local descriptors through an attention module. These concept states are then updated by graph-based interaction and used to adaptively modulate the local descriptors. We describe our proposed model by split-transform-attend-interact-modulate-merge stages, which are implemented by opting for a highly modularized architecture. Extensive experiments on visual recognition tasks such as image classification, semantic segmentation, object detection, scene recognition, and action recognition show that our proposed model, VCRNet, consistently improves the performance by increasing the number of parameters by less than 1%.

Introduction

Convolutional neural networks have shown notable success in visual recognition tasks by learning hierarchical representations. The main properties of convolutional operations, which are local connectivity and weight sharing, are the key factors that make it more efficient than fully-connected networks for processing images. The local connectivity particularly comes up with a fundamental concept, receptive field, that defines how far the local descriptor can capture the context in the input image. In principle, the receptive field can be expanded by stacking multiple convolutional layers or increasing the kernel size of them. However, it is known that the effective receptive field only covers a fraction of the theoretical size of it (Luo et al. 2016). This eventually restricts convolutional neural networks to capture the global context based on long-range dependencies. On the

other hand, most convolutional neural networks are characterized by dense and local operations that take the advantage of the weight sharing property. It hence typically lacks an internal mechanism for high-level reasoning based on abstract semantic concepts such as those humans manipulate with natural language and inspired by modern theories of consciousness (Bengio 2017). It is related to system 2 cognitive abilities, which include things like reasoning, planning, and imagination, that are assumed to capture the global context from interactions between a few abstract factors and accordingly give feedback to the local descriptor for decision-making.

There have been approaches to enhance capturing long-range dependencies such as non-local networks (Wang et al. 2018). The main concept of it, which is related to self-attention (Vaswani et al. 2017), is to compute a local descriptor by adaptively aggregating other descriptors from all positions, regardless of relative spatial distance. In this setting, the image feature map is plugged into a fully-connected graph neural network, where all local positions are fully connected to all others. It is able to capture long-range dependencies and extract the global context, but it still works with dense operations and lacks high-level reasoning. Both LatentGNN (Zhang, He, and Yan 2019) and GloRe (Chen et al. 2019) alleviate these issues by introducing compact graph neural networks with some latent nodes designed to aggregate local descriptors.

In this work, we propose *Visual Concept Reasoning Networks* (VCRNet) to enable reasoning between high-level visual concepts. We exploit a modularized multi-branch architecture that follows a split-transform-merge strategy (Xie et al. 2017). While it explicitly has multiple branches to simultaneously learn multiple visual concepts or properties, it only considers the dependencies or interactions between them by using dense and local operations. We extend the architecture by split-transform-attend-interact-modulate-merge stages, and this allows the model to capture the global context by reasoning with sparse interactions between high-level visual concepts from different branches.

The main contributions of the paper are:

- We propose Visual Concept Reasoning Networks (VCR-Net) that efficiently capture the global context by reasoning over high-level visual concepts.

*Now at Amazon Web Services (taesup@amazon.com).
Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

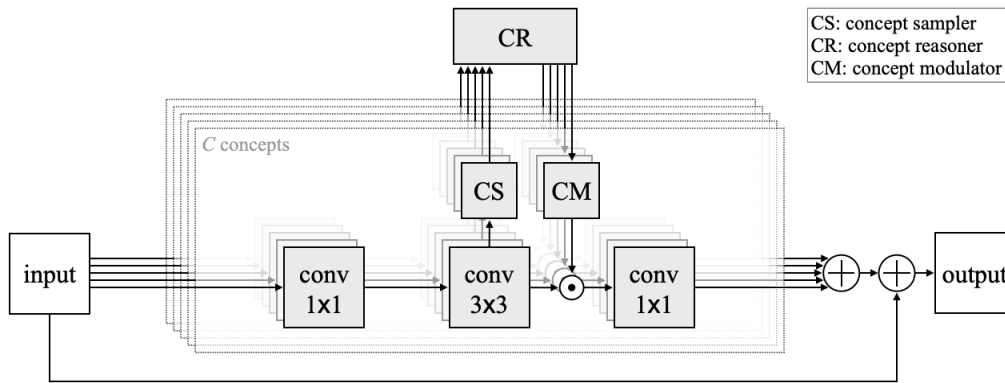


Figure 1: A residual block with visual concept reasoning modules.

- We compactly implement our proposed model by exploiting a modularized multi-branch architecture composed of split-transform-attend-interact-modulate-merge stages.
- We showcase that our proposed model improves the performance more than other models by increasing the number of parameters by less than 1% on multiple visual recognition tasks.

Related Works

Multi-branch architectures are carefully designed with multiple branches characterized by different dense operations, and split-transform-merge stages are used as the building blocks. The Inception models (Szegedy et al. 2015) are one of the successful multi-branch architectures that define branches with different scales to handle multiple scales. ResNeXt (Xie et al. 2017) is another version of ResNet (He et al. 2016) having multiple branches with the same topology in residual blocks, and it is efficiently implemented by grouped convolutions. In this work, we utilize this residual block and associate each branch of it with a visual concept.

There have been several works to adaptively modulate the feature maps based on the external context or the global context of input data. Squeeze-and-Excitation networks (SE) (Hu, Shen, and Sun 2018) use a gating mechanism to do channel-wise re-scaling in accordance with the channel dependencies based on the global context. Gather-Excite networks (GE) (Hu et al. 2018) further re-scale locally that it is able to finely redistribute the global context to the local descriptors. Convolutional block attention module (CBAM) (Woo et al. 2018) independently and sequentially has channel-wise and spatial-wise gating networks to modulate the feature maps. All these approaches extract the global context by using the global average pooling that equally attends all local positions. Dynamic layer normalization (DLN) (Kim, Song, and Bengio 2017) and Feature-wise Linear Modulation (FiLM) (Perez et al. 2018) present a method of feature modulation on normalization layers by conditioning on the global context and the external context, respectively.

Content-based soft-attention mechanisms (Bahdanau, Cho, and Bengio 2015) have been broadly used on neural networks to operate on a set of interchangeable objects

and aggregate it. Particularly, Transformer models (Vaswani et al. 2017) have shown impressive results by using multi-head self-attention modules to improve the ability to capture long-range dependencies. Non-local networks (NL) (Wang et al. 2018) use this framework in pixel-level self-attention blocks to implement non-local operations. There are some additional related works that one augments the self-attention modules to convolutional operations (Bello et al. 2019a), and another replaces all of them with a form of self-attention (Ramachandran et al. 2019). Global-context networks (GC) (Cao et al. 2019) simplify the non-local networks by replacing the pixel-level self-attention with an attention module having a single fixed query that is globally shared and learned. Attention-augmented convolutional networks (Bello et al. 2019b) similarly augment convolutional operators with self-attention modules as the non-local networks, but concatenate feature maps from convolution path and self-attention path. LatentGNN (Zhang, He, and Yan 2019) and Global reasoning module (GloRe) (Chen et al. 2019) differently simplifies the non-local networks that they first map local descriptors into latent nodes, where the number of nodes is relatively smaller than the number of local positions, and capture the long-range dependencies from interactions between the latent nodes. Our proposed model is similar to these two models, but we take the advantage of the multi-branch architecture and the attention mechanism to efficiently extract a set of distinct visual concept states from the input data.

Methods

In this section, we introduce our proposed model, Visual Concept Reasoning Network (VCRNet), and describe the overall architecture and its components in detail. The proposed model is designed to reason over high-level visual concepts and accordingly modulate feature maps based on its result. In the following, we assume the input data $X \in \mathbb{R}^{H \times W \times d}$ is a 2D tensor as an image feature map, where H , W , and d refer to the height, width, and feature size of X , respectively. Moreover, for simplicity, we denote all modules by a function $F_{\text{func}}(\cdot; \theta)$, where θ is a learnable parameter and the subscript func briefly explains the functionality of it.

Modularized Multi-Branch Residual Block

Residual blocks are composed of a skip connection and multiple convolutional layers (He et al. 2016). We especially take advantage of using a residual block of ResNeXt (Xie et al. 2017) that operates by grouped convolutions. This block is explicable by a *split-transform-merge* strategy and a highly modularized multi-branch architecture. It has an additional dimension "cardinality" to define the number of branches used in the block. The branches are defined by separate networks, which are based on the same topology and implemented by grouped convolutions, processing non-overlapping low-dimensional feature maps. In this work, we use this block by regarding each branch as a network learning representation of a specific *visual concept* and, therefore, refer to the cardinality as the number of visual concepts C . The split-transform-merge strategy can be described by visual concept processing as the following. Each concept c has a compact concept-wise feature map $Z_c \in \mathbb{R}^{HW \times p}$, where p is a lot smaller than d . It is initially extracted from the input data X by *splitting* it into a low-dimensional feature map $\tilde{X}_c \in \mathbb{R}^{HW \times p}$ with a 1×1 convolution $F_{\text{split}}(X; \theta_c^{\text{split}})$. Afterward, it is followed by a *concept-wise transformation* based on a 3×3 convolution $F_{\text{trans}}(\tilde{X}_c; \theta_c^{\text{trans}})$ while keeping the feature size compact. The extracted concept-wise feature maps $\{Z_c\}_{c=1}^C$ are then projected back into the input space to be *merged* as $Y = X + \sum_{c=1}^C F_{\text{merge}}(Z_c; \theta_c^{\text{merge}})$. This overall multi-branch procedure interestingly can be highly modularized and parallelized by grouped convolutions. However, it lacks the ability of reasoning over the high-level visual concepts that captures both local and global contexts. We propose to extend this approach by introducing additional modules to enable visual concept reasoning. Our proposed model is based on a new strategy with split-transform-attend-interact-modulate-merge stages. The new stages completely work into the residual block with the following modules: (a) concept sampler, (b) concept reasoner, and (c) concept modulator. The overall architecture is depicted in Figure 1 showing how it is highly modularized by sharing the topology between different concepts. We refer to networks having residual blocks with these modules, as Visual Concept Reasoning Networks (VCRNet).

Concept Sampler

The concept-wise feature maps $\{Z_c\}_{c=1}^C$ are composed of all possible pixel-level local descriptors, which contain spatially local feature information, as sets of vectors. To do efficient reasoning over the visual concepts, it first requires a set of abstract feature vectors representing the visual concepts. Therefore, a form of aggregation mechanism is necessary to derive a set of visual concept states, where each state is a vector, from the concept-wise feature maps. We implement this by presenting a *concept sampler* (CS) module. Each concept c has a separate concept sampler $F_{\text{CS}}(Z_c; \theta_c^{\text{CS}})$ that aggregates the set of local descriptors in Z_c and converts it into a concept state $h_c \in \mathbb{R}^{1 \times \tilde{p}}$, where we set $\tilde{p} = \min(p/4, 4)$. We introduce two types of concept samplers that are based on pooling and attention operations, respectively. Global average pooling is one of the simplest

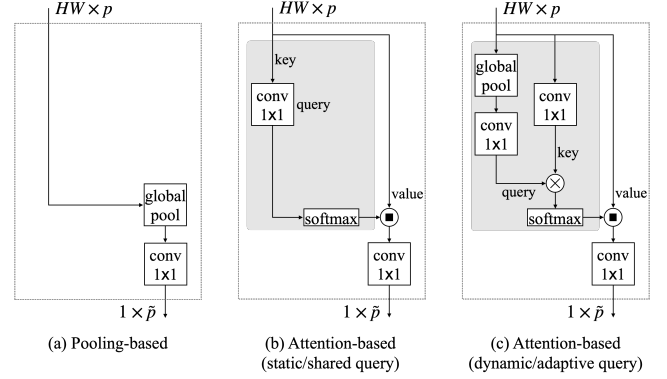


Figure 2: Concept samplers with different approaches (\otimes is a weighted-sum operation).

ways to extract the global context from a feature map without explicitly capturing long-range dependencies. It equally and densely attends all local positions to aggregate the local descriptors. Our pooling-based sampler adopts this operation to compute the concept state h_c as shown in Figure 2.a, and it is formulated as:

$$h_c = F_{\text{GAP}}(Z_c)W_c^v = \left(\frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W Z_c[i, j] \right) W_c^v, \quad (1)$$

where $Z_c[i, j] \in \mathbb{R}^{1 \times p}$ is a local descriptor at position (i, j) , and $W_c^v \in \mathbb{R}^{p \times \tilde{p}}$ is a learnable projection weight. In comparison with the attention-based sampler, it is simple and compact having a small number of parameters, but there is no data-adaptive process. Due to its simplicity, similar approaches have been broadly used in the previous works such as SENet (Hu, Shen, and Sun 2018) and CBAM (Woo et al. 2018). The attention mechanism operates by mapping a query vector and a set of interchangeable key-value vector pairs into a single vector, which is a weighted sum of value vectors. It allows us to aggregate a set of local descriptors by sparsely and adaptively selecting them. We hence apply this approach to our concept sampler. For each concept c , the query vector $q_c \in \mathbb{R}^{1 \times \tilde{p}}$ describes what to focus on during aggregation. The concept-wise feature map Z_c converts into a set of key-value vector pairs that we separately project it into a key map $K_c = Z_c W_c^k$ and a value map $V_c = Z_c W_c^v$, where $W_c^k, W_c^v \in \mathbb{R}^{p \times \tilde{p}}$ are learnable projection weights. The concept state h_c is derived by computing the dot products of the query vector q_c with the key map K_c and subsequently applying a softmax function to obtain the attention weights over the value map V_c as:

$$h_c = \left(\text{softmax} \left(\frac{q_c (Z_c W_c^k)^\top}{\sqrt{\tilde{p}}} \right) Z_c \right) W_c^v. \quad (2)$$

The query vector q_c can be either learned as a model parameter or computed by a function of the feature map Z_c . The former approach defines a static query that is shared globally over all data. GCNet (Cao et al. 2019) uses this approach, instead of global average pooling, to extract the global context.

It can be simplified and implemented by replacing the term $q_c (Z_c W_c^k)^\top$ in Equation 2 with a 1×1 convolution as depicted in Figure 2.b. The latter approach, in contrast, uses a dynamic query that varies according to Z_c . We set the query as an output of the function as $q_c = F_{\text{GAP}}(Z_c) W_c^q$, which is equal to the pool-based sampler, as shown in Equation 1.

The concept samplers can be viewed as multi-head attention modules in Transformer models (Vaswani et al. 2017) that we set each concept to be operated by a single-head attention module. However, our concept samplers don't process the same input feature map as they do. Each concept is only accessible to its corresponding feature map, and this encourages the concept samplers to attend and process different features. Moreover, we explicitly define concept-wise queries to aggregate pixel-wise (low-level) descriptors and obtain global descriptors (high-level concepts) rather than only capturing long-range dependencies as non-local networks (Wang et al. 2018) work with pixel-level dense self-attention operations.

Concept Reasoner

The visual concept states are derived independently from separate branches in which no communication exists. Therefore, we introduce a reasoning module, *Concept Reasoner* (CR), to make the visual concept states to interact with the others and accordingly update them. We opt for using a graph-based method by defining a fully-connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes \mathcal{V} and directional edges \mathcal{E} . The node $v_c \in \mathcal{V}$ corresponds to a single visual concept c and is described by the visual concept state h_c . The edge $e_{cc'} \in \mathcal{E}$ defines the relationship or dependency between visual concepts c and c' . It is further specified by an adjacency matrix $A \in \mathbb{R}^{C \times C}$ to represent edge weight values in a matrix form. Based on this setting, we describe the update rule of the visual concept states as:

$$\tilde{h}_c = \text{ReLU} \left(\text{BN} \left(h_c + \sum_{c'=1}^C A[c, c'] h_{c'} \right) \right), \quad (3)$$

where $A[c, c'] \in \mathbb{R}$ is a edge weight value, and batch normalization (BN) and ReLU activation are used. This can also be implemented in a matrix form as $\tilde{H} = \text{ReLU}(\text{BN}(H + AH))$, where $H = [h_1; h_2; \dots; h_C] \in \mathbb{R}^{C \times \tilde{p}}$ is vertically stacked concept states. The adjacency matrix A can be treated as a module parameter that is learned during training. This sets the edges to be static that all relationships between visual concepts are consistently applied to all data. However, we relax this constraint by dynamically computing the edge weights based on the concept states. A function $A[c, :] = F_{\text{edge}}(h_c; W^{\text{edge}}) = \tanh(h_c W^{\text{edge}})$, where $W^{\text{edge}} \in \mathbb{R}^{\tilde{p} \times C}$ is a learnable projection weight, is used to get all edge weights $A[c, :]$ related to the concept c as shown in Figure 3. This function learns how each concept adaptively relates to the others based on its state.

Concept Modulator

The updated concept states are regarding not only a single concept, but also the others as a result of reasoning based

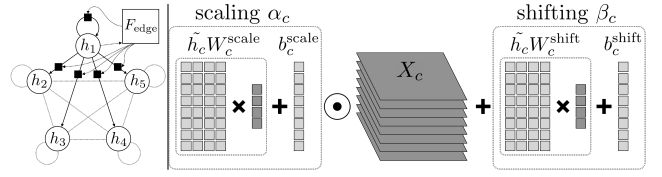


Figure 3: (Left) Concept reasoner and (right) modulator

on interactions. This information has to be further propagated to local concept features, which are extracted from the mainstream of the network. However, this is a non-trivial problem due to dimensional mismatch that the concept states are vectors not containing any explicit spatial information. We alleviate this issue by implementing a module, *Concept Modulator* (CM), which is based on a feature modulation approach. It modulates the concept-wise feature maps by channel-wise scaling and shifting operations. These operations are conditioned on the updated concept states to fine-tune the feature maps based on the result of reasoning. We design this module based on DLN (Kim, Song, and Bengio 2017) and FiLM (Perez et al. 2018). Both models use feature-wise affine transformations on normalization layers by dynamically generating the affine parameters instead of learning them. In this way, we define separate modules for the visual concepts as shown in Figure 3. Each concept-wise feature map X_c is modulated as:

$$\tilde{X}_c = F_{\text{CM}}(\tilde{h}_c, X_c; \theta^{\text{CM}}) = \text{ReLU}(\alpha_c \odot X_c + \beta_c),$$

$$\alpha_c = \tilde{h}_c W_c^{\text{scale}} + b_c^{\text{scale}}, \beta_c = \tilde{h}_c W_c^{\text{shift}} + b_c^{\text{shift}},$$

where \odot indicates channel-wise multiplication. $\alpha_c, \beta_c \in \mathbb{R}^{1 \times p}$ are scaling and shifting parameters, respectively, which are adaptively computed by linearly mapping the updated concept state \tilde{h}_c .

We further implement *pixel-level* concept modulators to propagate the global context adaptively and differently into local descriptors. Each concept state h_c is derived by computing the attention map $M_c \in \mathbb{R}^{HW \times 1}$ from the concept sampler as shown in Equation 2, and we assume it contains the spatial information related to the concept c . Therefore, we utilize this attention map for the pixel-level concept modulator. We first re-normalize the attention map by its maximum value:

$$\tilde{M}_c = \frac{M_c}{\max(M_c)}, M_c = \text{softmax} \left(\frac{q_c (Z_c W_c^k)^\top}{\sqrt{\tilde{p}}} \right).$$

Without this re-normalization, the learning doesn't work properly. The re-normalized attention map \tilde{M}_c is used to project the updated concept state \tilde{h}_c into all local positions by projection $\tilde{M}_c \tilde{h}_c \in \mathbb{R}^{HW \times \tilde{p}}$. Based on this projection, we are able to do pixel-level feature modulation as:

$$\tilde{X}_c = F_{\text{CM}}(\tilde{h}_c, \tilde{M}_c, X_c; \theta^{\text{CM}}) = \text{ReLU}(\alpha_c \cdot X_c + \beta_c)$$

$$\alpha_c = (\tilde{M}_c \tilde{h}_c) W_c^{\text{scale}} + b_c^{\text{scale}}, \beta_c = (\tilde{M}_c \tilde{h}_c) W_c^{\text{shift}} + b_c^{\text{shift}},$$

where \cdot is an element-wise multiplication. Both α_c and β_c are having the same size as the feature map X_c so that all local positions have separate scaling and shifting parameters.

| Model | Error (%) | | # of Params | GFLOPs |
|---|-----------|-------|-------------|--------|
| | Top-1 | Top-5 | | |
| ResNeXt-50 (Xie et al. 2017) | 21.10 | 5.59 | 25.03M | 4.24 |
| ResNeXt-50 + SE (Hu, Shen, and Sun 2018) | 20.79 | 5.38 | 27.56M | 4.25 |
| ResNeXt-50 + CBAM (Woo et al. 2018) | 20.73 | 5.36 | 27.56M | 4.25 |
| ResNeXt-50 + GC (Cao et al. 2019) | 20.44 | 5.34 | 27.58M | 4.25 |
| ResNeXt-50 + GloRe (Chen et al. 2019) | 20.15 | 5.14 | 30.79M | 5.86 |
| ResNeXt-50 + VCR (ours) | 19.97 | 5.03 | 25.26M | 4.26 |
| ResNeXt-50 + VCR (ours, pixel-level) | 19.94 | 5.18 | 25.26M | 4.29 |
| ResNeXt-101 (Xie et al. 2017) | 19.82 | 4.96 | 44.18M | 7.99 |
| ResNeXt-101 + SE (Hu, Shen, and Sun 2018) | 19.39 | 4.73 | 48.96M | 8.00 |
| ResNeXt-101 + CBAM (Woo et al. 2018) | 19.60 | 4.87 | 48.96M | 8.00 |
| ResNeXt-101 + GC (Cao et al. 2019) | 19.52 | 5.03 | 48.99M | 8.00 |
| ResNeXt-101 + GloRe (Chen et al. 2019) | 19.56 | 4.85 | 49.93M | 9.61 |
| ResNeXt-101 + VCR (ours) | 18.84 | 4.48 | 44.60M | 8.01 |

Table 1: Results of image classification on ImageNet validation set

| Backbone Network | AP ^{bbox} | AP ₅₀ ^{bbox} | AP ₇₅ ^{bbox} | AP ^{mask} | AP ₅₀ ^{mask} | AP ₇₅ ^{mask} | # Params |
|--|--------------------|----------------------------------|----------------------------------|--------------------|----------------------------------|----------------------------------|----------|
| ResNeXt-50 (Xie et al. 2017) | 40.70 | 62.02 | 44.49 | 36.75 | 58.89 | 39.03 | 43.94M |
| ResNeXt-50 + SE (Hu, Shen, and Sun 2018) | 41.04 | 62.61 | 44.45 | 37.13 | 59.53 | 39.79 | 46.47M |
| ResNeXt-50 + CBAM (Woo et al. 2018) | 41.69 | 63.54 | 45.17 | 37.48 | 60.27 | 39.71 | 46.47M |
| ResNeXt-50 + GC (Cao et al. 2019) | 41.66 | 63.76 | 45.29 | 37.58 | 60.36 | 39.92 | 46.48M |
| ResNeXt-50 + GloRe (Chen et al. 2019) | 42.31 | 64.18 | 46.13 | 37.83 | 60.63 | 40.17 | 49.71M |
| ResNeXt-50 + VCR (ours) | 41.81 | 63.93 | 45.67 | 37.71 | 60.36 | 40.25 | 44.18M |
| ResNeXt-50 + VCR (ours, pixel-level) | 42.02 | 64.15 | 45.87 | 37.75 | 60.62 | 40.22 | 44.18M |

Table 2: Results of object detection and instance segmentation on COCO 2017 validation set

Experiments

In this section, we run experiments on visual recognition tasks such as image classification, object detection/segmentation, scene recognition, and action recognition with large-scale datasets. In all experiments, we set ResNeXt (Xie et al. 2017), which performs better than ResNet (He et al. 2016) with less parameters, as a base architecture with cardinality = 32 and base width = 4d. As our main contribution is to exploit the *multi-branch* architecture to enable high-level concept reasoning by implementing *split-transform-attend-interact-modulate-merge* stages, we only use the ResNeXt as a backbone network that it is the only one already having the split-transform-merge stages allowing us to seamlessly implement our VCRNet. Furthermore, our proposed model, VCRNet, is also defined by $C = 32$ concepts in all residual blocks. We also compare VCRNet against other networks (modules), which have a form of attention or reasoning modules, such as Squeeze-and-Excitation (SE) (Hu, Shen, and Sun 2018), Convolutional Block Attention Module (CBAM) (Woo et al. 2018), Global Context block (GC) (Cao et al. 2019), and Global Reasoning unit (GloRe) (Chen et al. 2019). All networks are implemented in all residual blocks in the ResNeXt except GloRe, which is partially adopted in the second and third residual stages. In all experiments, we mainly set VCRNet with using (1) attention-based concept samplers with dynamic queries, (2) concept reasoners with dynamic edge weights, and (3) concept modulators with channel-level fea-

ture map scaling and shifting.

Image Classification

We conduct experiments on a large-scale image classification task on the ImageNet dataset (Russakovsky et al. 2015). The dataset consists of 1.28M training images and 50K validation images from 1000 different classes. All networks are trained on the training set and evaluated on the validation set by reporting the top-1 and top-5 errors with single center-cropping. Our training setting is explained in detail in Appendix. The overall experimental results are shown in Table 1, where all results are reproduced by our training setting for a fair comparison. For evaluation, we always take the final model, which is obtained by exponential moving average (EMA) with the decay value 0.9999. VCRNet consistently outperforms than other networks in both ResNeXt-50 and ResNeXt-101 settings. Moreover, it is more compact than the others as it only increases the number of parameters by less than 1% ($\approx 0.95\%$). In contrast, GloRe (Chen et al. 2019), which also does high-level reasoning as our model, requires more parameters than ours, although it is partially applied in the ResNeXt architecture. In addition, we test the pixel-level concept modulators to reuse the attention maps extracted from the concept samplers to modulate local descriptors at *pixel-level* as GloRe has a pixel-level re-projection mechanism. The modification slightly improves the top-1 performance by using the same number of parameters, but it increases the computational cost (GFLOPs).

| Model | Error (%) | | # of Params |
|--|-----------|-------|-------------|
| | Top-1 | Top-5 | |
| ResNeXt-50 (Xie et al. 2017) | 43.49 | 13.54 | 23.73M |
| ResNeXt-50 + SE (Hu, Shen, and Sun 2018) | 43.18 | 13.41 | 26.26M |
| ResNeXt-50 + CBAM (Woo et al. 2018) | 43.18 | 13.45 | 26.26M |
| ResNeXt-50 + GC (Cao et al. 2019) | 43.07 | 13.34 | 26.28M |
| ResNeXt-50 + GloRe (Chen et al. 2019) | 42.94 | 13.22 | 29.48M |
| ResNeXt-50 + VCR (ours) | 42.92 | 12.96 | 23.96M |

Table 3: Results of scene recognition on Places-365

| Backbone network (Slow-only pathway) | Error (%) | | # of Params |
|--|-----------|-------|-------------|
| | Top-1 | Top-5 | |
| ResNeXt-50 (Xie et al. 2017) | 26.41 | 9.43 | 40.07M |
| ResNeXt-50 + SE (Hu, Shen, and Sun 2018) | 25.06 | 8.70 | 42.58M |
| ResNeXt-50 + CBAM (Woo et al. 2018) | 24.87 | 8.81 | 42.59M |
| ResNeXt-50 + GC (Cao et al. 2019) | 25.31 | 9.32 | 42.60M |
| ResNeXt-50 + GloRe (Chen et al. 2019) | 25.52 | 9.23 | 45.81M |
| ResNeXt-50 + VCR(ours) | 24.73 | 8.39 | 40.28M |

Table 4: Results of action recognition on Kinetics-400

| Model (ResNeXt-50) | Top-1 Error (%) | # of Params |
|-----------------------|--------------------|----------------|
| pool | 20.21 | 25.17M |
| static attn | 20.18 | 25.17M |
| dynamic attn | 19.97 | 25.26M |

(a) Concept Sampler

| Model (ResNeXt-50) | Top-1 Error (%) | # of Params |
|-----------------------|--------------------|----------------|
| no edge | 20.23 | 25.26M |
| static edge | 20.02 | 25.28M |
| dynamic edge | 19.97 | 25.26M |

(b) Concept Reasoner

| Model (ResNeXt-50) | Top-1 Error (%) | # of Params |
|-----------------------|--------------------|----------------|
| only scale | 20.13 | 25.22M |
| only shift | 20.05 | 25.22M |
| scale + shift | 19.97 | 25.26M |

(c) Concept Modulator

Table 5: Ablation study on VCRNet

Object Detection and Segmentation

We further do some experiments on object detection and instance segmentation on the MSCOCO 2017 dataset (Lin et al. 2014). MSCOCO dataset contains 115K images over 80 categories for training, 5K for validation. Our experiments are based on the Detectron2¹. All backbone networks are based on the ResNeXt-50 and pre-trained on the ImageNet dataset by default. We employ and train the Mask R-CNN with FPN (He et al. 2017). We follow the training procedure of the Detectron2 and use the 1× schedule setting. Furthermore, synchronized batch normalization is used instead of freezing all related parameters. For evaluation, we use the standard setting of evaluating object detection and instance segmentation via the standard mean average-precision scores at different boxes and the mask IoUs, respectively. Table 2 is the list of results by only varying the backbone network. It shows similar tendencies to the results of ImageNet. However, GloRe (Chen et al. 2019) is showing the best performance. We assume that this result is from two factors. One is the additional capacity, which is relatively larger than other models, used by GloRe. The other is that GloRe uses pixel-level re-projection mechanism that applies the result of reasoning by re-computing all local descriptors. Especially, the task requires to do prediction on pixel-level so that it would be beneficial to use it. Therefore, we also make our model to use pixel-level feature modulation. It further improves the performance without requiring additional parameters.

Scene and Action Recognition

Places365 (Zhou et al. 2017) is a dataset labeled with scene semantic categories for the scene recognition task. This task is challenging due to the ambiguity between classes that several scene classes may share some similar objects causing confusion among them. We specifically use the Places365-Standard setting that the train set has up to 1.8M images from 365 scene classes, and the validation set has 50 images

per each class. All networks are trained from random initialization and evaluated on the validation set by following the setting used in our ImageNet experiments. Additionally, we insert Dropout (Srivastava et al. 2014) layers in residual blocks with $p = 0.02$ to avoid some over-fitting. The human action recognition task is another task appropriate to demonstrate how the network can generalize well not only to 2D image data, but also to 3D video data. We use the Kinetics-400 dataset (Kay et al. 2017) including 400 human action categories with 235K training videos and 20K validation videos. We follow the slow-only experiment setting used in (Feichtenhofer et al. 2019) that simply takes the ImageNet pre-trained model with a parameter inflating approach (Carreira and Zisserman 2017). Both tasks are classification tasks similar to the ImageNet image classification, and the results shown in Table 3 and 4 explain that our approach are generally performing better than other baselines in various visual classification tasks. Moreover, action recognition results prove that our model can be generally applied to all types of data.

Ablation Study

(a) *Concept Sampler*: We have proposed different approaches for the concept sampler (pooling-based and attention based samplers). To compare these approaches, we train our proposed networks (ResNeXt-50 + VCR) by having different concept samplers and keeping all other modules fixed. Table 5.(a) compares the performance of these approaches on the ImageNet image classification task. The attention-based approach with dynamic queries (dynamic attn) outperforms the others, and we assume that this is due to having more adaptive power than the others. Furthermore, the results interestingly show that our models consistently perform better than other baseline networks except a network with GloRe, which are shown in Table 1, regardless of the type of concept sampler. (b) *Concept Reasoner*: To investigate the effectiveness of reasoning based on interactions between concepts, we conduct some experiments by modifying the concept reasoner. We first remove the concept in-

¹<https://github.com/facebookresearch/detectron2>

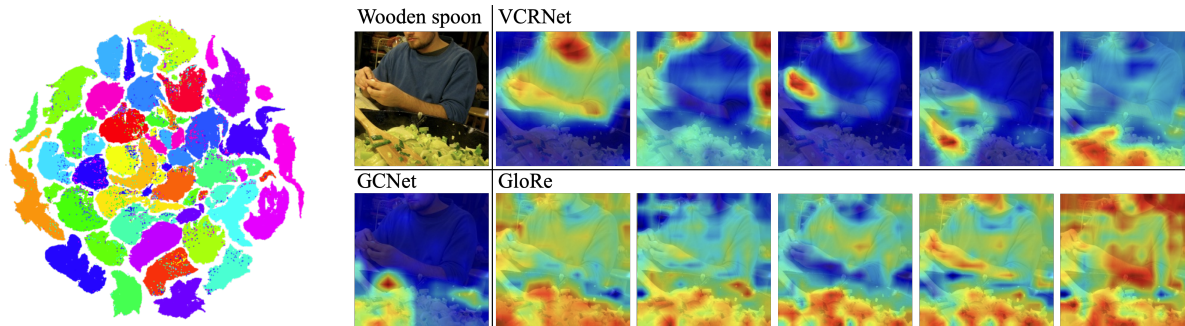


Figure 4: (Left) t-SNE plots of visual concept states. $C = 32$ concepts are distinguished by 32 colors. (Right) Visualization of attention (projection) maps from VCRNet, GCNet, and GloRe

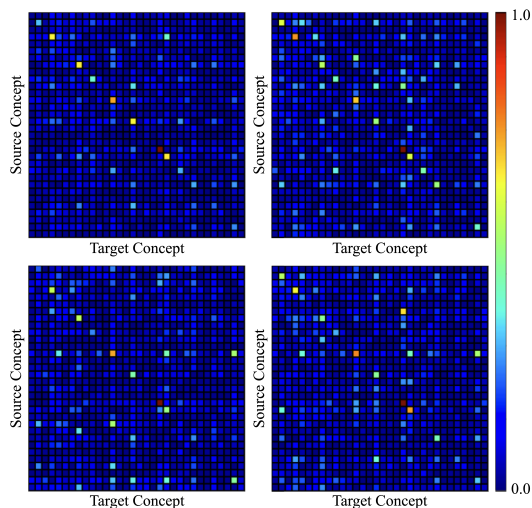


Figure 5: Visualization of interactions between concepts.

interaction term in Equation 3 and evaluate it to measure the effectiveness of reasoning. Moreover, we also compare the performance between learned static edges and computed dynamic edges. In Table 5.(b), the results show that the reasoning module is beneficial in terms of the performance. Notably, it also reveals that using dynamic edges can improve the reasoning and reduce the number of parameters. (c) *Concept Modulator*: Our feature modulation consists of both channel-wise scaling and shifting operations. Previous works have shown to use only scaling (gating) (Hu, Shen, and Sun 2018; Woo et al. 2018; Hu et al. 2018) or only shifting (Cao et al. 2019). We compare different settings of the feature modulation as shown in Table 5.(c). Using only shifting performs better than using only scaling, and combining both operations can be recommended as the best option.

Visualization

We use t-SNE (van der Maaten and Hinton 2008; Chan et al. 2019) to visualize how visual concept states are existing in the feature space. We collect a set of concept states, which are all extracted from the same concept sampler, by doing inference with the ImageNet validation set. In Figure 4, it

is shown that the concept states are clustered and separated by concepts. This result can be further explained by observing the attention maps computed from the concept samplers. Interestingly, they reveal the fact that the concept samplers sparsely attend different regions or objects, and this would result in clustered concept states. This also convinces that our proposed architecture is able to learn distinct concepts without any supervision that explicitly associates branches with certain labeled concepts. We also visualize attention (projection) maps from other networks such as GCNet (Cao et al. 2019) and GloRe (Chen et al. 2019) in Figure 4. GCNet only produces a single attention map, and it tends to sparsely attend foreground objects. GloRe similarly computes projection maps as our approach, but the maps are densely attending regions with some redundancies between them. We furthermore extract edge absolute values (interactions) between concepts from different images and visualized them in Figure 5. It shows that each image has different interactions between concepts, and concepts are interacting sparsely that most edge values are near zero.

Conclusion

In this work, we propose Visual Concept Reasoning Networks (VCRNet) that efficiently capture the global context by reasoning over high-level visual concepts. Our proposed model precisely fits to a modularized multi-branch architecture by having split-transform-attend-interact-modulate-merge stages. The experimental results shows that it consistently outperforms other baseline models on multiple visual recognition tasks and only increases the number of parameters by less than 1%. We strongly believe research in these approaches will provide notable improvements on more difficult visual recognition tasks in the future. As future works, we are looking forward to remove dense interactions between branches as possible to encourage more specialized concept-wise representation learning and improve the reasoning process. Moreover, we expect to have consistent and specialized visual concepts that are shared and updated over all stages in the network by removing dense interactions between different branches as possible. Explicitly associating branches in the networks with labeled concepts will also improve the learning of tasks requiring high-level reasoning.

References

- Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL <http://arxiv.org/abs/1409.0473>.
- Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; and Le, Q. V. 2019a. Attention Augmented Convolutional Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; and Le, Q. V. 2019b. Attention Augmented Convolutional Networks. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Bengio, Y. 2017. The Consciousness Prior. *CoRR* abs/1709.08568. URL <http://arxiv.org/abs/1709.08568>.
- Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 1971–1980.
- Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4724–4733.
- Chan, D. M.; Rao, R.; Huang, F.; and Canny, J. F. 2019. GPU accelerated t-distributed stochastic neighbor embedding. *Journal of Parallel and Distributed Computing* 131: 1–13.
- Chen, Y.; Rohrbach, M.; Yan, Z.; Shuicheng, Y.; Feng, J.; and Kalantidis, Y. 2019. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 433–442.
- Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. SlowFast Networks for Video Recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6201–6210.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hu, J.; Shen, L.; Albanie, S.; Sun, G.; and Vedaldi, A. 2018. Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 9401–9411. Curran Associates, Inc. URL <http://papers.nips.cc/paper/8151-gather-excite-exploiting-feature-context-in-convolutional-neural-networks.pdf>.
- Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-Excitation Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950. URL <http://arxiv.org/abs/1705.06950>.
- Kim, T.; Song, I.; and Bengio, Y. 2017. Dynamic Layer Normalization for Adaptive Neural Acoustic Modeling in Speech Recognition. In Lacerda, F., ed., *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2411–2415. ISCA. URL http://www.isca-speech.org/archive/Interspeech/_2017/abstracts/0556.html.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *European Conference Computer Vision ECCV*.
- Luo, W.; Li, Y.; Urtasun, R.; and Zemel, R. 2016. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*, 4898–4906. Curran Associates, Inc. URL <http://papers.nips.cc/paper/6203-understanding-the-effective-receptive-field-in-deep-convolutional-neural-networks.pdf>.
- Perez, E.; Strub, F.; de Vries, H.; Dumoulin, V.; and Courville, A. C. 2018. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI*.
- Ramachandran, P.; Parmar, N.; Vaswani, A.; Bello, I.; Levskaya, A.; and Shlens, J. 2019. Stand-Alone Self-Attention in Vision Models. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 68–80. Curran Associates, Inc. URL <http://papers.nips.cc/paper/8302-stand-alone-self-attention-in-vision-models.pdf>.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*.
- Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15(56): 1929–1958. URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going Deeper with Convolutions. In *Computer Vision and Pattern Recognition (CVPR)*.
- van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9: 2579–2605. URL <http://www.jmlr.org/papers/v9/vandermaaten08a.html>.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. At-

tention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 5998–6008. Curran Associates, Inc. URL <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7794–7803.

Woo, S.; Park, J.; Lee, J.; and Kweon, I. S. 2018. CBAM: Convolutional Block Attention Module. In *European Conference Computer Vision ECCV*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, S.; He, X.; and Yan, S. 2019. LatentGNN: Learning Efficient Non-local Relations for Visual Recognition. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 7374–7383. Long Beach, California, USA: PMLR. URL <http://proceedings.mlr.press/v97/zhang19f.html>.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.